MEDSKILL-E^2: EVALUATING AND EVOLVING SKILLS FOR MEDICAL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) agents increasingly use skills to access domain knowledge and complete real-world tasks. We observe that even when a skill specifies the rules and procedures needed to complete a task, agents may still fail to apply them effectively during autonomous execution, a phenomenon we call skill Enable But Underuse (EBU). This issue is particularly consequential in medical tasks, where overlooking even a single critical constraint can lead to a plausible but incorrect result. To systematically evaluate and address this phenomenon, we introduce MedSkill-E^2, a framework for evaluating and evolving skills for medical agents. We construct EBU-Bench, a challenging benchmark covering 12 medically relevant task groups and their associated skills, to evaluate agents' ability to use available skills autonomously. We quantify the EBU gap as the difference in skill-use accuracy between autonomous use and explicitly guided reading and application of the same skill. On EBU-Bench, all evaluated model–harness combinations exhibit EBU, with a mean gap of 43.94 percentage points (pp). For the same model, skill-use accuracy under autonomous use differs by up to 38.96 pp across harnesses. Motivated by these observations, we propose EBU-Evolve, which uses execution feedback from autonomous and explicitly guided skill use to distill effective guidance into reusable skill instructions. Relative to their pre-evolution versions, the revised skills improve skill-use accuracy during autonomous execution by 33.33–55.17 pp and continue to yield improvements when transferred across multiple models and harnesses. Our results show that skill-use accuracy under autonomous execution varies substantially across harnesses even when the model and skill are held fixed, and model rankings also change; EBU can persist even after a skill is opened. EBU-Evolve distills effective guidance into reusable skill instructions, improving skill-use accuracy under autonomous execution on unseen tasks within the same skill, and still yields improvements when the evolved skills are transferred to other models and harnesses.

1 INTRODUCTION

Large language model (LLM) agents use skills to access domain knowledge and apply it to specific tasks. A skill organizes the methods, tool-use instructions, and operating procedures needed for a class of tasks into a reusable resource, so that agents need not work out a solution procedure anew for each problem. This support is particularly important in medicine and related research: clinical calculations require accurate application of formulas and their constraints, while evidence review requires adherence to explicit assessment criteria. Skills offer a direct way to standardize these operations. Skills have also begun to emerge for healthcare and the life sciences (Xu et al., 2026).

However, on medical benchmarks and practical medical research tasks, we observe that agents sometimes fail to apply the methods in a skill effectively, even when an applicable skill is provided and its purpose and conditions of use are specified in its description. In other cases, changing the execution framework (harness) while holding the model and skill fixed reduces or even eliminates the performance gains previously obtained from the skill. These observations raise a question: in real medical applications, how much assistance can an available skill provide, and how much of that benefit can an agent realize when solving tasks autonomously? We further find that some previously unsuccessful tasks can be completed when the agent is directed to use a particular skill and

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

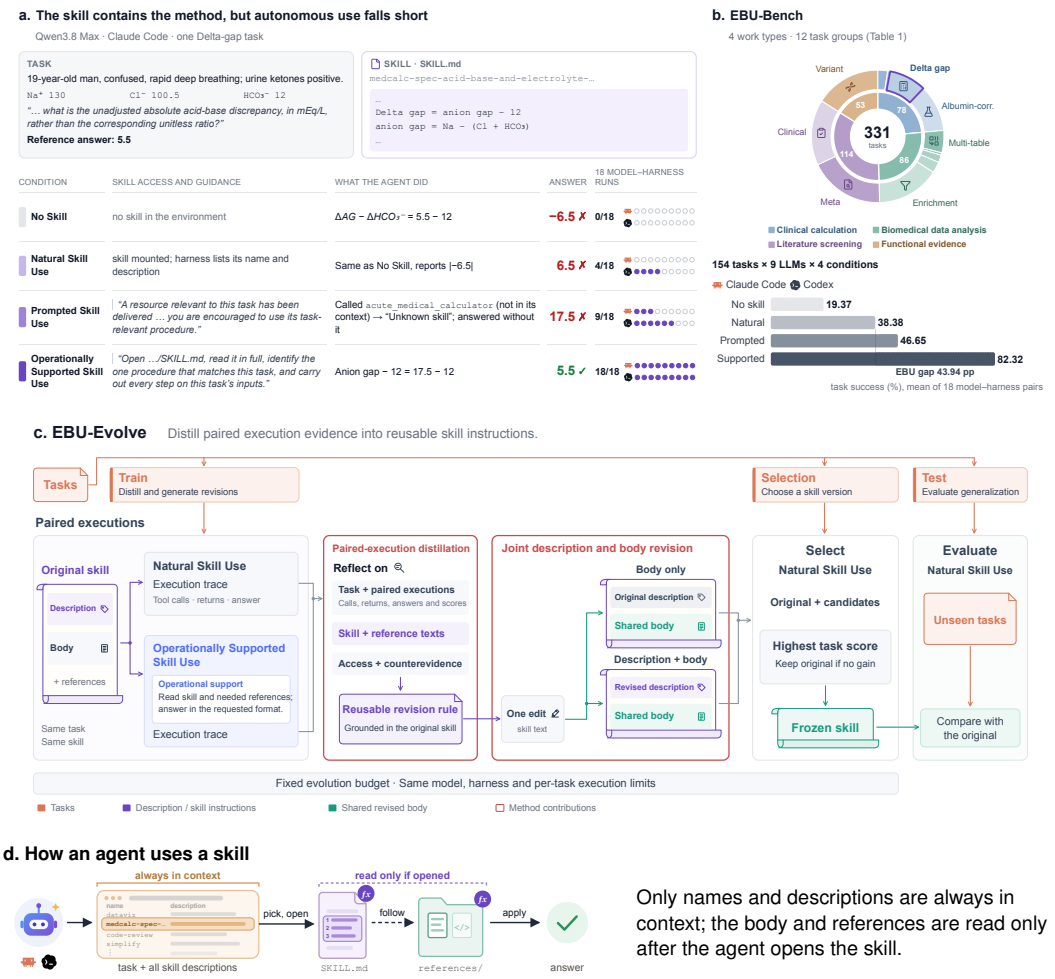


Figure 1: An example of skill underuse, the benchmark, and the evolution method. (a) Answers and behavior on the same task under four conditions; the dots on the right summarize the number of correct answers by harness, and their positions do not correspond to model order. (b) Task composition and mean skill-use accuracy under each condition on EBU-Bench Lite. (c) EBU-Evolve distills reusable instructions from paired executions of the same task, generates two candidates sharing the same body, selects the best version by Natural Skill Use scores on the validation set, and evaluates it on test tasks excluded from revision and selection. Red boxes indicate the methodological contributions. (d) How an agent uses a skill.

given corresponding execution requirements. We call this additional guidance "operational support" and use task performance under this condition as an empirical reference for the benefits achievable with the skill. We term the phenomenon in which the same skill supports task completion under operational support but its corresponding benefits remain unrealized during autonomous execution skill Enable But Underuse (EBU). EBU may not be widespread across medical tasks, but when it occurs, agents produce plausible results that do not comply with the specified procedures, warranting dedicated study. We therefore identify tasks exhibiting EBU during benchmark construction and investigate two questions: how underuse on these tasks varies across models and harnesses, and whether revising reusable skill instructions can improve autonomous execution.

To study this problem systematically, we introduce MedSkill-E^2, a framework for evaluating and evolving skills for medical agents that comprises EBU-Bench and EBU-Evolve. EBU-Bench contains 331 tasks and their associated skills, combining tasks from public benchmarks with tasks authored around research questions. Evaluation is organized around reusable skills, each intended for a class of problems. We select task–skill pairs from this collection and evaluate nine ad-

vanced language models under two harnesses. Mean skill-use accuracy is 38.38% when models use skills autonomously and 82.32% with operational support; on EBU-Bench, all evaluated model–harness combinations exhibit EBU. Even with the model and skill held fixed, performance under autonomous use varies substantially across harnesses.

Operational support externally identifies a skill to use and provides general execution requirements for the task, but such guidance is usually unavailable in advance in practical settings. Notably, these supplementary instructions do not disclose any domain knowledge or operational steps specific to the task. We therefore propose EBU-Evolve to distill effective guidance from operational support into reusable skill instructions. The method revises a skill using paired execution feedback from the same task under autonomous use and operational support, and selects a revised version based on task performance under autonomous use. The revised skill can then be used directly for autonomous task solving without additional operational support for individual tasks. Experiments show that the revised skills improve overall skill-use accuracy under autonomous use on test tasks excluded from revision and selection, and continue to yield improvements when transferred across models or harnesses.

2 EBU-BENCH

2.1 BENCHMARK CONSTRUCTION

EBU-Bench is a targeted challenge benchmark for skill underuse: it evaluates whether agents put available methods into practice. It combines tasks from public benchmarks with tasks authored around research questions; the public-benchmark task–skill pairs pass through three screening stages.

Stage 1: task collection and capability screening. We examine 11 public benchmarks and evaluation resources covering clinical calculation, medical image understanding, literature processing, biological data analysis, and clinical system operations (Khandekar et al., 2024; Wang et al., 2026c; Fallahpour et al., 2025; Mitchener et al., 2025; Jiang et al., 2025; Zhu et al., 2026; Liu et al., 2026a; Nair et al., 2026; Fa et al., 2026; Workman et al., 2026; Wang et al., 2026b). Tasks with the necessary data and tools are solved without a skill, and those not fully solved form the candidate pool.

Stage 2: skill pairing and effectiveness validation. For each task we retrieve and screen candidate skills from public skill libraries and methodological resources, organizing public methodological material into reusable procedural documents where necessary. Each candidate is then provided on its own with a prompt to perform its task-relevant operations, and pairs that can be completed under Operationally Supported Skill Use are retained.

Stage 3: screening for an autonomous-use gap. With the skill description provided alongside the task, the agent decides on its own whether and how to use the skill. For the same task and skill version, we retain tasks that are not completed under autonomous use but are completed under Operationally Supported Skill Use.

After the three screening stages, the retained public-benchmark tasks come from MedCalc-Bench and BixBench. To broaden coverage, we additionally author and expand tasks for literature screening, variant functional evidence, and enrichment review from common research questions, relevant domain skills, and public research materials; these require applying the skill's method under the specific conditions of the problem, such as screening studies against inclusion criteria, judging the applicability of functional evidence, or choosing the background for enrichment analysis. Varying problem conditions and resources yields 218 task pairings. Together with the 113 public-benchmark tasks, the two pathways yield 331 tasks after expansion, deduplication, and version freezing (Table X), each with its inputs, a skill, and scoring rules for the final output; sources, construction, and screening protocols are detailed in Appendix A.

Table 1: Task composition of EBU-Bench.

Work type	Task group	Tasks
Clinical calculation	HAS-BLED	8
	PSI	2
	Delta gap	30
	Albumin-corrected Delta gap	38
Biomedical data analysis	Multi-table integration and association analysis	18
	Paired matrix analysis	2
	Cohort and variant aggregation	6
	Expression–feature association	9
	Enrichment review	51
Literature screening	Abstract screening	61
	Clinical evidence screening	53
Functional evidence assessment	Variant functional evidence	53
Total	**12 task groups**	**331**

2.2 Task Definition and Evaluation Metrics

Task inputs and objectives. Each task includes a problem description, the required data, the associated skill, and the tool environment needed for completion; the agent performs the requested calculations, analyses, or judgments and submits a final answer. Under Natural Skill Use, the associated skill is provided through the harness's standard mechanism, and the agent decides on its own whether to open it and how to use it (Figure 1d).

Task scoring. Each task has a fixed scoring program: numerical answers are checked against predefined tolerances or intervals, categorical and structured answers against their rules. An answer scores 1 if it passes and 0 otherwise, and the mean score over the evaluated tasks is the skill-use accuracy. Scoring depends only on the final result; the agent chooses its own procedure. We also record from the execution trace whether the agent opens the associated skill.

The EBU gap. Operationally Supported Skill Use uses the same task and skill as Natural Skill Use but explicitly requires the agent to read the skill and perform its task-relevant operations, i.e., the operational support introduced above. With $P_{\text{natural}}(m, h)$ and $P_{\text{supported}}(m, h)$ the skill-use accuracies of model m under harness h in the two conditions, we define $\Delta_{\text{EBU}}(m, h) = P_{\text{supported}}(m, h) - P_{\text{natural}}(m, h)$. Using Operationally Supported Skill Use as an empirical reference, the EBU gap measures the task gains not yet realized under Natural Skill Use for a given model and harness. We report both accuracies and retain any negative differences; across harnesses, the gaps of the same model are averaged with equal weight. Skill-opening records and cross-harness differences support further analysis but do not enter the gap.

Organizing tasks around skills. Each associated skill corresponds to multiple tasks that apply the same method to different inputs and problem conditions (Table X), so that a method is evaluated across multiple instances; for skills with enough tasks, the tasks can be split into training, validation, and test sets for EBU-Evolve (§3).

EBU-Bench Lite. Because running the full benchmark across models, harnesses, and conditions is costly, we use a 154-task subset that keeps all 12 task groups: all 113 public-benchmark tasks plus one representative task per source-question–skill combination among the 218 authored tasks (41 tasks), which reduces the repeated weight of variants of the same source question. Representatives are selected independently of the evaluation results reported here (rules in Appendix A.4). Unless otherwise stated, the benchmark experiments use EBU-Bench Lite.

3 EBU-Evolve

Figures 1a and 1b show that general encouragement to use skills (Prompted Skill Use) remains insufficient to fully realize the benefits of an available skill. We propose EBU-Evolve, which derives concrete, generalizable revisions from the differences between executions of the same task under Natural Skill Use and Operationally Supported Skill Use, and distills them into persistent skill instructions. This addresses EBU on the individual task and, at the same time, enables agents to use the revised skill effectively on similar tasks that exhibit the same form of underuse, thereby realizing its value under Natural Skill Use.

Figure 1c presents the overall EBU-Evolve workflow. When evolving a skill, we divide its tasks into three sets: the training set provides evidence for revision, the validation set determines the skill version, and the test set evaluates the selected version on tasks that took part in neither this revision nor the selection. Specifically, we use paired executions to determine what to revise, jointly revise the description and the body, and select the version with the highest mean task score under Natural Skill Use. Evolution leaves the model parameters and the skill's accompanying reference files, code, and data unchanged; only the skill's instruction text is updated.

Distilling skill instructions from paired executions. A single failure is insufficient to determine the direction of revision, whereas a successful execution of the same task under Operationally Supported Skill Use provides a reference. We therefore collect, for the same training task, the execution traces and scores under both conditions, using the same skill version, model, and harness. Building on trace-based reflective optimization (Agrawal et al., 2026), we pass these paired executions, together with the original skill and its reference materials, to a reflection model, which judges whether the agent accessed the relevant rules during autonomous execution and whether it applied

them under the current task conditions, and proposes revision rules accordingly. Only revision rules supported by the original skill are used to generate candidates (Appendix B.1). We call this process **intervention-to-instruction distillation**: the improvements brought by operational support are written persistently into the skill.

Jointly revising the description and body. Skills are loaded progressively: the description is always visible, while the body and reference materials are read on demand, so a rule written only into the body may never be read. Accordingly, a single edit produces two candidates that share the same body: one revises only the body, and the other revises the description as well. Because the two bodies are identical, the Natural Skill Use scores on the validation set indicate whether the skill's entry point also needs to change (Appendices B.2–B.4).

4 Experimental Setup

4.1 Models and Execution Environments

We evaluate nine models: GPT-5.5, Claude Opus 4.8, Gemini 3.8 Flash, DeepSeek V4 Pro, GLM-5.2, Atria-Dawn-Preview (Atria), Kimi K2.7 Code, Qwen3.8 Max, and Qwen3.8 27B FP8, each under two harnesses, Claude Code and Codex, which differ in resource discovery and tool interaction (request identifiers and harness versions in Appendix C).

Benchmark evaluations are executed through BenchFlow: each run uses an independent container and session with the data, computational tools, and file-operation interfaces required by the task, and external retrieval and data downloads are disabled to prevent answer leakage. All nine models are evaluated on the same 154 tasks described in §2.2; results of three models on the full benchmark of 331 tasks are reported in Appendix F.

4.2 Skill Provision and Evaluation Conditions

We ask not only whether a skill is provided but how much additional prompting an agent needs before using it, and define four conditions per task (Figure 1a). The EBU gap is the difference between **Natural Skill Use** and **Operationally Supported Skill Use** (§2.2); **No Skill** and **Prompted Skill Use** measure performance without the associated skill and the change induced by a general reminder.

Under the conditions that provide a skill, the skill's name and description enter the context through the harness's native mechanism, and the body and reference files are read on demand. Natural Skill Use adds no task-level prompting. Prompted Skill Use adds only general encouragement to use the skill, without specifying solution steps, and tests whether merely reminding the agent suffices to realize the skill's benefits. Operationally Supported Skill Use further specifies the reading and execution requirements, providing a reference for the gains that remain unrealized. The three conditions with a skill use the same resource version and differ only in the support appended to the task prompt.

Operationally Supported Skill Use reuses a prompt fixed for each task group: for tasks derived from public benchmarks it requires reading the associated skill in full and applying its steps to the current inputs, and for authored tasks it requires reading the entry file and the reference materials it points to. Neither supplies the answer to the current task; because the two differ in specificity, we also report results by task source (Appendix C.4).

4.3 Execution Protocol and Result Aggregation

For each model, harness, and condition we keep the score of one run per task (11,088 runs; all accuracies use 154 as the denominator). The initial time limit is 900 or 3,600 seconds depending on the batch; runs that time out at 900 seconds are rerun with up to 7,200 seconds, and the rerun replaces the original record. Runs that time out or submit no valid answer, including answers in a format the scorer cannot parse, score 0 (configuration, rerun scope, and invalid-run distribution in Appendix C).

For each model, harness, and condition, we retain the score of one run on each of the 154 tasks, 11,088 runs in total, and all skill-use accuracies use 154 as the denominator. The initial time limit of a run is either 900 or 3,600 seconds depending on the batch; runs that time out under the 900-second limit are rerun with a budget of up to 7,200 seconds, and the rerun replaces the original record. Runs that time out or fail to submit a valid answer, including answers whose format does not meet the scoring requirements, are scored 0. The detailed configuration, the scope of reruns, and the distribution of invalid runs are given in Appendix C.

4.4 Evaluation Setup for EBU-Evolve

EBU-Evolve requires multiple tasks under the same skill to provide revision feedback and test the revisions. Because the task–skill pairs retained from public benchmarks do not support this split, we expand research questions from the literature into four skill groups: Abstract screening, Clinical evidence screening, Variant functional evidence, and Enrichment review. Training, validation, and test sets follow §3, with disjoint task IDs (splits and execution settings in Appendix B.7).

We evolve skills separately from the execution feedback of GLM-5.2, GPT-5.5, and Atria under Claude Code, with Claude Opus 4.8 performing reflection, editing, and review; each validation task is repeated three times and the version is selected by mean task score. At test time, the selected version is compared with the original skill under Natural Skill Use without operational support. To test reuse across configurations, each source's frozen version is also given to the other two source models and to DeepSeek V4 Pro, and tested with the same source model under Codex; all gains are relative to the original skill in the target configuration, and overall accuracies are aggregated over tasks.

The method comparison evaluates, under a common GLM-5.2/Claude Code configuration, the artifacts of the prompt-optimization method GEPA, the skill-optimization method SkillOpt, and EBU-Evolve; the first two are run with standard feedback (Std) and with paired evidence from the same task (Paired), under the same budget and editing permissions.

In addition, we design two ablations (Appendix B.8). First, the presentation-component analysis decomposes the frozen GLM-5.2 artifact, comparing replacement of the description alone, the body alone, and both. Second, to test the contribution of EBU-Evolve's overall EBU-driven design, we add a generic trace-reflection rewriting control that receives the same original skill, reference materials, and autonomous execution traces with scores, and uses the same candidate construction, review, and selection; it receives only basic rewriting instructions, whereas EBU-Evolve additionally uses paired executions under operational support, example selection based on score changes, and utilization diagnosis.

5 Experimental Results

5.1 EBU-Bench Results

We first compare the skill-use accuracies of the nine models under Claude Code and Codex across the four conditions (Figure 2; per-cell values in Appendix Table D.1). Averaged with equal weight over the 18 model–harness combinations, the accuracies under No Skill, Natural Skill Use, and Operationally Supported Skill Use are 19.37%, 38.38%, and 82.32%. All 18 combinations exhibit a positive EBU gap, from 8.44 to 73.38 pp with a mean of 43.94 pp (95% CI: [39.57, 48.09], bootstrap over tasks). The largest gap is GLM-5.2/Claude Code (73.38 pp); as task screening used GLM-5.1 from the same family, excluding GLM-5.2 leaves a largest gap of 71.43 pp (Qwen3.8 27B FP8/Claude Code). Even the best Natural Skill Use combination, Gemini 3.8 Flash/Codex, rises from 68.18% to 76.62% under Operationally Supported Skill Use.

Autonomous skill access depends on the model–harness combination. Cross-harness differences are concentrated in Natural Skill Use: the mean accuracies of Claude Code and Codex are 26.48% and 50.29%, a difference of 23.81 pp, whereas under No Skill and Operationally Supported Skill Use they differ by only 0.94 and 1.59 pp. For the same model, Natural Skill Use accuracy differs by up to 38.96 pp across harnesses (DeepSeek V4 Pro), and the Spearman correlation between the harnesses' model rankings drops from 0.90 under No Skill to 0.51 under Natural Skill Use, consistent with the ranking changes Skill-Use observed for skill-use scores (Han et al., 2026a).

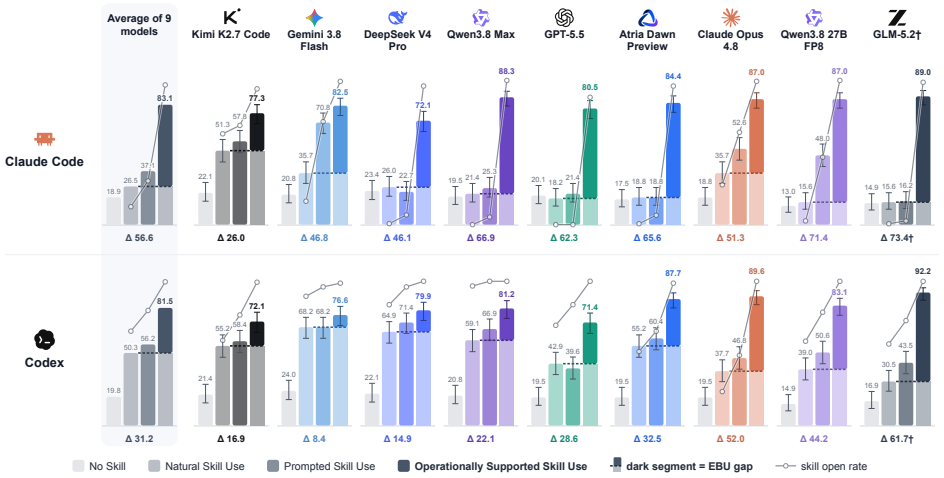


Figure 2: Skill-use accuracy (%) of nine models under two harnesses and four conditions. Models are ordered by increasing EBU gap averaged over the two harnesses; Δ below each group of bars is the EBU gap (pp). Error bars are Wilson 95% intervals (n = 154). †: task screening used GLM-5.1 from the same model family.

The same difference appears in resource access. A skill counts as opened once the target skill is read or loaded. Under Natural Skill Use the opening rates of Claude Code and Codex are 12.45% and 65.42%, whereas both approach saturation under Operationally Supported Skill Use (96.90% and 99.64%; Appendix G). Across the 18 combinations, opening rate and Natural Skill Use accuracy correlate at 0.91 (Pearson); Kimi K2.7 Code and Claude Opus 4.8, with similar opening rates under both harnesses, also show the smallest accuracy differences (3.90 and 1.95 pp). Thus, **autonomous skill utilization is a property of the model–harness combination; results and rankings under one harness do not transfer directly to another.**

Opening a skill does not eliminate underuse. Pairing runs by model, harness, and task, we decompose the net EBU gap by the change in opening status from Natural Skill Use to Operationally Supported Skill Use. Of the 56.64 pp gap of Claude Code, 54.69 pp comes from pairs that go from unopened to opened and only 1.95 pp from pairs opened under both conditions; for Codex the parts are 18.33 and 13.13 pp, the latter 42.0% of its 31.24 pp gap. Yet on same-task pairs in which the skill is opened under both conditions, Operationally Supported Skill Use still raises accuracy from 69.23% to 85.21% for Claude Code and from 63.97% to 84.15% for Codex (counts in Appendix G), so opening records alone cannot tell whether a skill's benefit has been realized.

An Atria/Codex trace on a Delta gap task shows this directly: the model reads the skill's rule AG − 12, calls it "unambiguous", yet submits an answer computed from the memorized $\Delta AG - \Delta HCO_3$; under Operationally Supported Skill Use the same model follows the resource and answers correctly (Appendix H). In medical tasks governed by a specification, a model may thus read the given rule and still substitute a familiar definition, producing a plausible result that does not comply with the provided specification.

Prompted Skill Use closes only a small part of the EBU gap. Prompted Skill Use encourages the model to use relevant resources without naming them. Relative to Natural Skill Use, accuracy rises from 38.38% to 46.65%, only 8.26 pp, closing 18.8% of the mean EBU gap (by aggregate net successes) and leaving 35.68 pp to the 82.32% of Operationally Supported Skill Use, which is higher in all 18 combinations. The traces also reveal a localized failure mode: under Prompted Skill Use, Qwen3.8 Max/Claude Code calls unregistered skill names in 60/154 runs and in 56 of them never opens the target skill yet submits an answer (Figure 1a; Appendix E).

5.2 EBU-EVOLVE RESULTS

Evolved skills improve Natural Skill Use performance and transfer to other models and harnesses. EBU-Evolve raises Natural Skill Use accuracy on three source models spanning frontier open-weight, proprietary, and science-oriented models (Figure 3a): on the 87 test tasks of the four skill groups (tasks of the same skill that took part in neither revision nor selection), skill-use accuracy improves by 55.17, 33.33, and 40.23 pp for GLM-5.2, GPT-5.5, and Atria. Distilling overlooked or ambiguously scoped rules into the skill thus improves Natural Skill Use on unseen tasks of the same skill.

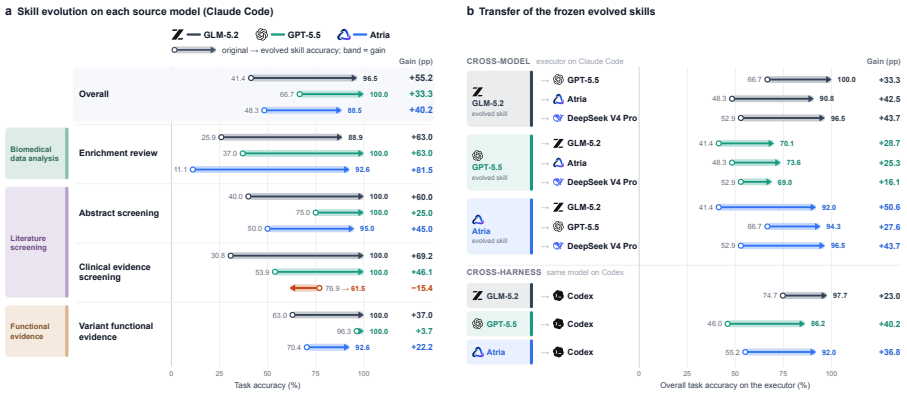


Figure 3: Skill evolution on source models and transfer of frozen skills. Orange marks a decrease in accuracy after evolution; in (b), the baseline of each target configuration is its own original skill.

The improvement transfers (Figure 3b): the evolved skills can be frozen and used as-is elsewhere, with Natural Skill Use gains of 16.09–50.57 pp across models and 22.99–40.23 pp when the same model switches from Claude Code to Codex; all 12 transfer configurations gain, so one evolution can be reused across agents and harnesses without per-task guidance.

Under the same budget and editing permissions, the skill revisions produced by EBU-Evolve are more effective than existing prompt-optimization and skill-optimization methods. In the common GLM-5.2 test configuration, EBU-Evolve raises Natural Skill Use accuracy from 41.38% to 96.55%, a gain of 55.17 pp, against 2.30 and 18.39 pp for the best configurations of GEPA (Agrawal et al., 2026) and SkillOpt (Yang et al., 2026), across all task types (Figure B.1a). Supplying the same paired feedback to the reflection modules of GEPA and SkillOpt (Paired) adds 2.30 and 10.34 pp over their standard settings (Std), so the Paired configuration also enriches the revision evidence of existing methods.

Two ablations test the EBU-Evolve design: decomposing the frozen GLM-5.2 artifact to separate the contributions of the description and body revisions, and comparing against generic trace-reflection rewriting to test the overall EBU-driven design. **Revising the description yields a larger gain with less added text, while jointly revising the description and body gives the highest skill-use accuracy** (Appendix B.8.1, Figure B.1b), and **EBU-Evolve attains a higher overall skill-use accuracy than generic trace-reflection rewriting** (96.55% vs. 88.51%; Appendix B.8.2, Figure B.1c). The gains of EBU-Evolve thus indeed come from the joint revision of the description and body and from its overall EBU-driven design based on paired executions.

Qualitative analysis reveals two EBU mechanisms. Comparing traces of the same training task under the two conditions, one mechanism is failing to acquire the rule, e.g., answering after reading only the main SKILL.md without opening the reference file needed for the judgment; the other is recognizing the task condition but not carrying it into the computation, e.g., excluding assay-eligible but unannotated entities in Enrichment review (Appendix B.9, Figure B.2).

The evolved skills state the applicability conditions and execution order of existing rules more concretely, at a limited context cost of 227 added tokens per skill on average. Rewriting also corrects the object of computation rather than merely inducing more reading: on a held-out Enrich-

ment review test task, the original and evolved skills read the same files, yet only the evolved skill keeps all 12 eligible entities and answers correctly. As in §5.1, reading a skill does not guarantee following it; problems diagnosed in training are corrected on tasks outside the revision.

6 Related Work

Agent evaluation and skill benchmarks. Evaluation of LLMs and LLM agents covers interactive environments (Liu et al., 2024b; Merrill et al., 2026) and specialized domains such as code repair, scientific data analysis, medical calculation, bioinformatics, and electronic health records (Jimenez et al., 2024; Chen et al., 2025; Khandekar et al., 2024; Mitchener et al., 2025; Jiang et al., 2025; Fa et al., 2026). Skill benchmarks instead evaluate the resources an agent is given: SkillsBench quantifies the benefit of a skill by paired evaluation of the same task with and without it and reports that this benefit varies across harnesses, a design we follow (Li et al., 2026); SWE-Skills-Bench uses the same comparison (Han et al., 2026b), and SkillLearnBench tests whether generated skills transfer across instances of the same task (Zhong et al., 2026). Unlike these with/without designs, EBU-Bench holds the task and skill fixed and contrasts Natural Skill Use with Operationally Supported Skill Use, quantifying the part of a skill's benefit the agent does not realize on its own.

From resource availability to effective use. Providing information does not guarantee that a model uses it: performance depends on where the information sits in the context (Liu et al., 2024a), and tool-use research separates whether a tool is invoked from how it is used (Ross et al., 2025; Chen et al., 2026). For skills, controlled comparisons of loading, selection, and retrieval conditions show that explicit invocation instructions raise the reading rate while prescribed steps may still be skipped (Liu et al., 2026b; Ying et al., 2026); finer analyses decompose skill use into triggering, adherence, and boundaries, distinguish tested from satisfied constraints, or compare outcomes on tasks where the skill is actually retrieved (Han et al., 2026a; Tan et al., 2026; Cho & Park, 2026). The harness also affects performance and cost (Yao et al., 2026), and agents may follow repository context files without gaining accuracy (Gloaguen et al., 2026). Access, adherence, and task-level benefit therefore need separate measurement; EBU-Bench measures underuse by the difference in skill-use accuracy rather than by reading or adherence, and examines its variation across harnesses with the model and skill fixed.

Improving skills from execution feedback. Retaining execution experience as an external resource is one route to continual improvement, from verbal reflections, cross-task rules, and programmatic skill libraries to prompt and context optimization with natural-language feedback (Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2024; Yuksekgonul et al., 2025; Agrawal et al., 2026; Zhang et al., 2026); tool documentation and skill descriptions can likewise be rewritten from interaction problems or routing errors (Qu et al., 2025; Zhou et al., 2026). For skills themselves, execution feedback has been used to discover new skills, summarize lessons, and select revised versions under validation feedback (Alzubi et al., 2026; Ni et al., 2026; Yang et al., 2026; Liu et al., 2026c), contrasting successes with failures under the initial skill, runs without a skill, or traces with and without the skill (Wang et al., 2026a; Ma et al., 2026; Gao et al., 2026). EBU-Evolve instead contrasts two use conditions of the same skill: successful executions under operational support locate the methods omitted or misapplied during autonomous use, and revisions are selected by Natural Skill Use outcomes.

7 Conclusion

We present MedSkill-E^2 to evaluate and mitigate the underuse of provided skills (EBU) by medical agents. On EBU-Bench Lite, whose tasks were selected for a gap in autonomous use, all 18 model–harness combinations show an EBU gap (43.94 pp on average); the harnesses are similar under Operationally Supported Skill Use but differ by 23.81 pp under Natural Skill Use, with changed rankings. Opening a skill is not the same as using it well, and a general reminder closes only 18.8% of the gap. EBU-Evolve distills operational requirements grounded in the original materials into the skill from paired executions: without task-level support, it improves Natural Skill Use accuracy on held-out tasks of the same skill for three source models, and the frozen skills still improve on other models and harnesses. Skill evaluation should therefore test, under the target model and harness, whether an agent realizes a skill's benefit on its own; skill authoring should attend to both the description's applicability conditions and the body's execution steps.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic data, to clean, reformat, and de-identify datasets, to assist with implementing and running experiments, and to translate the manuscript from Chinese into English. We have not used generative AI tools to develop theoretical models or conceptual frameworks, formulate mathematical claims, provide ingredients for or write proofs, propose or refine hypotheses, design research methodology or experiments, support qualitative or thematic data analysis, or interpret results. Additionally, we used generative AI tools to write and edit software code, to summarize existing literature, to search for information, to identify and verify relevant literature and format references, to draft and revise parts of the related-work section, and to edit the paper to improve readability. We have reviewed all AI-assisted work. Every cited reference was checked against its official publication record and cross-checked against at least one independent bibliographic source, and each claim attributed to a cited work was verified against the original text. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI. Large language models are also the subject of this study and serve as components of EBU-Evolve (the reflection and editing models); these uses are part of the method and are described in Sections 3 and 4.

ETHICS STATEMENT

This work evaluates LLM agents on tasks derived from publicly released benchmarks and evaluation resources, together with tasks authored by the authors around research questions; each external resource was used under its license and access terms. Some of these research questions were drawn from research workflows within the authors' group; they were de-identified and manually reviewed before use, and the benchmark contains no identifiable personal information. Several source benchmarks involve clinical calculations or electronic health record environments; performance on EBU-Bench is not evidence that an agent or skill is safe or effective for clinical use, and the skills studied here should not be deployed in patient care without independent clinical validation.

REPRODUCIBILITY STATEMENT

Section 2 and Appendix A describe how EBU-Bench tasks and their associated skills are constructed and how final outputs are scored. Section 4 and Appendix C specify the evaluation configuration and run protocol, including model request identifiers and harness versions. Appendix D reports per-cell results for the main evaluation, Appendix F reports results on the full benchmark, and Appendix G gives the counts underlying the skill-opening and paired analyses. Section 3 and Appendix B detail EBU-Evolve, including the data splits used for revision, selection, and testing.

REFERENCES

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alex Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RQm2KQTM5r.

Salaheddin Alzubi, Noah Provenzano, Jaydon Bingham, Christian Alexander Calvo, Weiyuan Chen, and Tu Vu. Evoskill: Automated skill discovery for multi-agent systems. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=tLvlDr1GWD.

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, 2026. URL https://arxiv.org/abs/2606.00510.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific dis-

covery. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=6z4YKr0GK6.

Seonghyeon Cho and Chanjun Park. Skill following: Evaluating actual skill use in retrieval-enabled llm agents, 2026. URL https://arxiv.org/abs/2609.00549.

Dionizije Fa, Marko Čuljak, Bruno Pandža, and Mateo Čupić. Bioagent bench: An AI agent evaluation suite for bioinformatics. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=2iFauBXg7Y.

Adibvafa Fallahpour, Jun Ma, Alif Munim, Hongwei Lyu, and Bo Wang. MedRAX: Medical reasoning agent for chest x-ray. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 15661–15676. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/fallahpour25a.html.

Haowen Gao, Haoran Chen, Can Wang, Shasha Guo, Liang Pang, Zhaoyang Liu, Huawei Shen, and Xueqi Cheng. Skillaudit: Ground-truth-free skill evolution via paired trajectory auditing, 2026. URL https://arxiv.org/abs/2606.14239.

Thibaud Gloaguen, Niels Mündler, Mark Müller, Veselin Raychev, and Martin Vechev. Evaluating agents.md: Are repository-level context files helpful for coding agents?, 2026. URL https://arxiv.org/abs/2602.11988.

Jinyi Han, Yuanjian Xu, Ying Liao, Xinyi Wang, Zishang Jiang, Zixiang Di, Fanyang Lu, Zhichao Hu, and Yanghua Xiao. Skill-use: Can llms actually use skills in agentic harnesses?, 2026a. URL https://arxiv.org/abs/2608.04828.

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?, 2026b. URL https://arxiv.org/abs/2603.15401.

Yixing Jiang, Kameron C. Black, Gloria Geng, Danny Park, James Zou, Andrew Y. Ng, and Jonathan H. Chen. Medagentbench: A virtual ehr environment to benchmark medical llm agents. *NEJM AI*, 2(9), August 2025. ISSN 2836-9386. doi: 10.1056/aidbp2500144. URL http://dx.doi.org/10.1056/AIdbp2500144.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Nikhil Khandekar, Qiao Jin, Guangzhi Xiong, Soren Dunn, Serina S Applebaum, Zain Anwar, Maame Sarfo-Gyamfi, Conrad W Safranek, Abid A Anwar, Andrew Zhang, Aidan Gilson, Maxwell B Singer, Amisha Dave, Andrew Taylor, Aidong Zhang, Qingyu Chen, and Zhiyong Lu. Medcalc-bench: Evaluating large language models for medical calculations. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 84730–84745. Curran Associates, Inc., 2024. doi: 10.52202/079017-2690. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/99e81750f3fdfcaf9613db2dbf4bd623-Paper-Datasets_and_Benchmarks_Track.pdf.

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, 2026. URL https://arxiv.org/abs/2602.12670.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024a. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Ruoqi Liu, Imran Q. Mohiuddin, Austin J. Schoeffler, Kavita Renduchintala, Ashwin Nayak, Prasantha L. Vemu, Shivam C. Vedak, Kameron C. Black, John L. Havlik, Isaac Ogunmola, Stephen P. Ma, Roopa Dhatt, and Jonathan H. Chen. Physicianbench: Evaluating llm agents in real-world ehr environments, 2026a. URL https://arxiv.org/abs/2605.02240.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating LLMs as agents. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=zAdUB0aCTQ.

Yujian Liu, Jiabao Ji, Li An, Tommi Jaakkola, Yang Zhang, and Shiyu Chang. How well do agentic skills work in the wild: Benchmarking LLM skill usage in realistic settings. In *Third Conference on Language Modeling*, 2026b. URL https://openreview.net/forum?id=axGWHkyD1z.

Yuxuan Liu, Zhaochen Su, Lingyun Xie, Yuhao Zhang, Qing Zong, Jiahe Guo, Zhongwei Xie, Yiyan Ji, Yauwai Yim, Hongyu Luo, Xiyu Ren, Chenyu Ruan, Haoran Li, and Yangqiu Song. Skillrevise: Improving llm-authored agent skills via trace-conditioned skill revision, 2026c. URL https://arxiv.org/abs/2606.01139.

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

Mike A Merrill, Alexander Glenn Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, E. Kelly Buchanan, Junhong Shen, Guanghao Ye, Haowei Lin, Jason Poulos, Maoyu Wang, Marianna Nezhurina, Di Lu, Orfeas Menis Mastromichalakis, Zhiwei Xu, Zizhao Chen, Yue Liu, Robert Zhang, Leon Liangyu Chen, Anurag Kashyap, Jan-Lucas Uslu, Jeffrey Li, Jianbo Wu, Minghao Yan, Song Bian, Vedang Sharma, Ke Sun, Steven Dillmann, Akshay Anand, Andrew Lanpouthakoun, Bardia Koopah, Changran Hu, Etash Kumar Guha, Gabriel H. S. Dreiman, Jiacheng Zhu, Karl Krauth, Li Zhong, Niklas Muennighoff, Robert Kwesi Amanfu, Shangyin Tan, Shreyas Pimpalgaonkar, Tushar Aggarwal, Xiangning Lin, Xin Lan, Xuandong Zhao, Yiqing Liang, Yuanli Wang, Zilong Wang, Changzhi Zhou, David Heineman, Hange Liu, Harsh Trivedi, John Yang, Junhong Lin, Manish Shetty, Michael Yang, Nabil Omi, Negin Raoof, Shanda Li, Terry Yue Zhuo, Wuwei Lin, Yiwei Dai, Yuxin Wang, Wenhao Chai, Shang Zhou, Dariush Wahdany, Ziyu She, Jiaming Hu, Zhikang Dong, Yuxuan Zhu, Sasha Cui, Ahson Saiyed, Arinbjörn Kolbeinsson, Jesse Hu, Christopher Michael Rytting, Ryan Marten, Yixin Wang, Jenia Jitsev, Alex Dimakis, Andy Konwinski, and Ludwig Schmidt. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=a7Qa4CcHak.

Ludovico Mitchener, Jon M Laurent, Alex Andonian, Benjamin Tenmann, Siddharth Narayanan, Geemi P Wellawatte, Andrew White, Lorenzo Sani, and Samuel G Rodriques. Bixbench: a comprehensive benchmark for llm-based agents in computational biology, 2025. URL https://arxiv.org/abs/2503.00096.

Surag Nair, Laura Gunsalus, Brian Orcutt-Jahns, Jordan Rossen, Avantika Lal, Carlo De Donno, Muhammed Hasan Çelik, Kipper Fletez-Brant, Xiaoman Xie, Hector Corrada Bravo, and Gokcen Eraslan. Agentic systems are adept at solving well-scoped, verifiable problems in computational

biology. April 2026. doi: 10.64898/2026.04.06.716850. URL http://dx.doi.org/10.64898/2026.04.06.716850.

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, 2026. URL https://arxiv.org/abs/2603.25158.

Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. From exploration to mastery: Enabling LLMs to master tools via self-driven interactions. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=QKBu1BOAwd.

Hayley Ross, Ameya Sunil Mahabaleshwarkar, and Yoshi Suhara. When2Call: When (not) to call tools. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3391–3409, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.174. URL https://aclanthology.org/2025.naacl-long.174/.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652. Curran Associates, Inc., 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1b44b878bb782e6954cd888628510e90-Paper-Conference.pdf.

Boyin Tan, Xiaowei Huang, and Youcheng Sun. Skill coverage: A test adequacy metric for agent skills, 2026. URL https://arxiv.org/abs/2606.20659.

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. ISSN 2835-8856. URL https://openreview.net/forum?id=ehfRiF0R3a.

Hanyu Wang, Yifan Lan, Bochuan Cao, Lu Lin, and Jinghui Chen. Skillgrad: Optimizing agent skills like gradient descent, 2026a. URL https://arxiv.org/abs/2605.27760.

Qing Wang, Bo Li, Jialu Liang, Daling Shi, Bob Zhang, and Qianqian Song. Drugclaw and drugaudit: A primary-source-grounded agent and authority-aware benchmark for drug-information question answering, 2026b. URL https://arxiv.org/abs/2606.01434.

Zifeng Wang, Jiacheng Lin, Qiao Jin, Junyi Gao, Jathurshan Pradeepkumar, Pengcheng Jiang, Zhiyong Lu, and Jimeng Sun. Benchmarking and developing large language models using one million clinical trials. *npj Digital Medicine*, July 2026c. ISSN 2398-6352. doi: 10.1038/s41746-026-02933-7. URL http://dx.doi.org/10.1038/s41746-026-02933-7.

Kenny Workman, Zhen Yang, Harihara Muralidharan, Aidan Abdulali, and Hannah Le. scbench: Evaluating ai agents on single-cell rna-seq analysis, 2026. URL https://arxiv.org/abs/2602.09063.

Gelei Xu, Ningzhi Tang, Xueyang Li, Toby Jia-Jun Li, Zhi Zheng, Wei Jin, and Yiyu Shi. An empirical study of agent skills for healthcare: Practice, gaps, and governance, 2026. URL https://arxiv.org/abs/2605.02709.

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, 2026. URL https://arxiv.org/abs/2605.23904.

Yilun Yao, Xinyu Tan, Chao-Hsuan Liu, Yaoming Li, Zhengyang Wang, Wenhan Yu, Zhewen Tan, Yuxuan Tian, Guangxiang Zhao, Lin Sun, Xiangzheng Zhang, and Tong Yang. Harness-bench: Measuring harness effects across models in realistic agent workflows, 2026. URL https://arxiv.org/abs/2605.27922.

Jiahao Ying, Boxian Ai, Wei Tang, Siyuan Liu, and Yixin Cao. Openskilleval: Automatically auditing the open skill ecosystem for llm agents, 2026. URL https://arxiv.org/abs/2605.23657.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. Optimizing generative ai by backpropagating language model feedback. *Nature*, 639(8055):609–616, March 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-08661-4. URL http://dx.doi.org/10.1038/s41586-025-08661-4.

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 *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=eC4ygDs02R.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19632–19642, March 2024. ISSN 2159-5399. doi: 10.1609/aaai.v38i17.29936. URL http://dx.doi.org/10.1609/aaai.v38i17.29936.

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. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=ulrZEJrNXA.

Yangqiaoyu Zhou, Mohammad Alqudah, Kwei-Herng Lai, Aaron Halfaker, Yingqi Xiong, and Yaar Harari. A single rewrite suffices: Empirical lessons from production skill description optimization, 2026. URL https://arxiv.org/abs/2606.30775.

Yakun Zhu, Yutong Huang, Shengqian Qin, Zhongzhen Huang, Shaoting Zhang, and Xiaofan Zhang. MedMCP-calc: Benchmarking LLMs for realistic medical calculator scenarios via MCP integration. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4845–4873, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.221. URL https://aclanthology.org/2026.acl-long.221/.

A Construction and Scoring of EBU-Bench

The 331 tasks of EBU-Bench come from two pathways: 113 tasks screened from public benchmarks (A.1) and 218 tasks authored and expanded around research questions (A.2). A.3 gives the scoring rules of each task group, and A.4 describes the selection of representative tasks for EBU-Bench Lite.

A.1 Tasks Derived from Public Benchmarks

Sources. We registered 11 public benchmarks and evaluation resources with 9,646 task entries in total (Table A.1). Some entries lack the data or tools needed to run them; the first round actually evaluated 9,305 records from 5 benchmarks.

Stage 1: No Skill screening. This stage has two rounds. The first round evaluates all runnable entries without providing a skill, using DeepSeek V4 Pro (GPT-4o for chest X-ray tasks), and records 2,966 tasks that do not reach full marks. The second round re-evaluates these candidates with GPT-5.5 (Codex, reasoning effort medium), removes tasks that are fully solved, and keeps 1,293. For non-binary scores such as F1, "not fully solved" means a score below full marks.

Stage 2: skill pairing and effectiveness validation. Candidate skills come from a locally curated skill set, GitHub repository search, the Bioconductor index, and skill marketplace listings, 15,368 entries after merging. Two rounds of filtering by domain relevance, documentation completeness, and maintenance status retain 1,996 entries, of which 1,154 are native agent skills; the retrieval pool actually used for pairing contains 1,478 entries. Pairing proceeds in three steps: the model

Table A.1: Registered public benchmark sources.

Source	Registered entries	Records evaluated in round 1
MedCalc-Bench	1,100	1,100
TrialPanorama	5,200	5,200
ChestAgentBench	2,500	2,500
BixBench	205	205
MedAgentBench	300	300
MedMCP-Calc	118	—
PhysicianBench	100	—
CompBioBench	100	—
BioAgentBench	10	—
scBench	6	—
DrugClaw	7	—
Total	9,646	9,305

first proposes 3–6 required capabilities from the task (without solving it), lexical retrieval over skill names and descriptions then keeps at most 25 candidates, and finally the model re-ranks them and returns 1–3 skills. The retriever is not given reference answers at this stage. Each candidate skill is then provided on its own, the agent is asked to read it and perform the relevant operations, and pairs that can be completed under Operationally Supported Skill Use are kept. Some BixBench skills were organized into procedural documents from public methodological material and re-validated for effectiveness.

Stage 3: screening for an autonomous-use gap. For the same task and skill version, we compare Natural Skill Use with Operationally Supported Skill Use and keep tasks that are not completed under the former but completed under the latter. This stage mainly used GLM-5.1 for screening. The final 113 tasks correspond to 163 comparisons meeting this condition; 74 tasks have one observation and 39 have several.

Final composition. The retained tasks derived from public benchmarks come from MedCalc-Bench and BixBench.

- **MedCalc-Bench (78 tasks).** These cover four calculators: HAS-BLED (8), PSI (2), Delta gap (30), and albumin-corrected Delta gap (38). The prompts are rewritten so that they do not name the calculator or the skill directly, while keeping the medical information needed for the calculation. Numerical variants are generated with the official task-expansion program. One of the 78 tasks is a neutrally worded version of an existing acid–base task.
- **BixBench (35 tasks).** These are spread over 4 data capsules: multi-table integration and association analysis (bix-10, 18 tasks), paired matrix analysis (bix-16, 2), cohort and variant aggregation (bix-20, 6), and expression–feature association (bix-22, 9). Five are official original tasks; the other 30 construct new analysis slices, parameter conditions, or analysis conventions on the real data of the same capsule. Input data are mounted read-only, and the reference notebooks and answers are not visible to the agent.

A.2 Authored and Expanded Tasks

Motivation and materials. These tasks are inspired by the patterns of real user questions on a research-agent platform, but contain no original user records. The task materials come from public literature and data, on which the problem conditions are varied. The four task groups are constructed as follows (Table A.2).

Literature screening tasks. The 31 representative tasks of the two screening groups correspond to 20 studies and 18 systematic reviews, with some studies shared between the groups. The typical construction modifies the inclusion criteria of the review. For example, the original review allowed quasi-randomized studies and the target study used alternate allocation; once this allowance is removed, the study should be excluded. The reference answer is therefore the judgment under the

Table A.2: Materials and construction of the authored and expanded tasks. "Representative" is the number of tasks in EBU-Bench Lite.

Task group	Tasks	Representative	Materials and construction
Abstract screening	61	21	Real study abstracts and systematic-review settings from TrialPanorama; the inclusion criteria are changed and the target study is re-judged under the new criteria
Clinical evidence screening	53	10	As above
Variant functional evidence	53	6	Based on the ClinGen SVI PS3/BS3 functional-evidence framework, two tasks set against real functional-assay papers; given the validation panel, controls, and calibration conditions, the evidence code and strength are determined
Enrichment review	51	4	One task uses a real GEO gene signature (GSE10586, obtained via Enrichr) with GO Biological Process 2025 annotations; the assay panel and experimental design are specified, and the correct background set must be rebuilt before computing

modified criteria, not the label in the original dataset. Other versions change the abstract wording, joint eligibility conditions, or resource versions.

Variant functional evidence and Enrichment review tasks. The background knowledge of these two groups comes from public frameworks and data, while the validation records, assay panels, and experimental conditions in the tasks are controlled synthetic material that does not represent real patients or the experimental results of the original papers. Reference answers are computed by frozen rules: the variant tasks give the evidence code, strength, and number of usable validation variants; the enrichment tasks give the size and hash of the background set and, for each term, K, n, k, and the hypergeometric right-tail p value.

Screening. The authored tasks also pass through the autonomous-use gap screening.

A.3 Scoring

All tasks are scored only on the agent's final answer, without regard to whether the skill was opened or which tools were called. A passing answer scores 1 and any other answer 0; a missing answer or one whose format does not meet the scoring requirements scores 0.

- **MedCalc-Bench**: the official scoring function `check_correctness` with frozen numerical bounds.
- **BixBench**: the original scorer; 33 tasks are judged by numerical intervals and 2 by string matching.
- **Authored and expanded tasks**: the 41 representative tasks share one scoring codebase, with the rules in Table A.3.

A.4 Selection of Representative Tasks for EBU-Bench Lite

EBU-Bench Lite contains 154 tasks: all 113 tasks derived from public benchmarks, plus one representative task from each "source question–associated skill" group of the 218 authored and expanded tasks, 41 in total (Table A.4). This keeps all 12 task groups while reducing the repeated weight of multiple variants of the same source question and skill in aggregate results.

Grouping. A "source question" is the original problem identifier used in the construction records to trace rewriting and expansion. Each task is keyed by its source question and the name of its associated skill: the same source question paired with different skills falls into different groups, whereas rewrites of the question or revisions of the resources under the same source question and skill fall into the same group. Grouping follows the existing construction relations, not text similarity

Table A.3: Scoring rules of the authored and expanded tasks.

Task group	What the agent submits	Condition for a correct answer
Abstract screening, Clinical evidence screening	An `included` or `excluded` judgment for the target study (identified by PMID)	The target PMID appears exactly once, the judgment matches the reference, and the submitted PMID set equals the expected set
Variant functional evidence	Evidence code (PS3/BS3/none), strength, number of usable validation variants	All three fields match the reference; integers must be exactly equal
Enrichment review	Size and hash of the background set; K, n, k, and p value for each term	All fields match; integers must be exactly equal, and floating-point values allow a relative error of 1e-5

Table A.4: Task composition of the full benchmark and EBU-Bench Lite.

Source or task group	Full benchmark	EBU-Bench Lite	Selection
MedCalc-Bench-derived tasks	78	78	All retained
BixBench-derived tasks	35	35	All retained
Abstract screening	61	21	One per source question–skill combination
Clinical evidence screening	53	10	One per source question–skill combination
Variant functional evidence	53	6	One per source question–skill combination
Enrichment review	51	4	One per source question–skill combination
Total	**331**	**154**	—

or task scores. The 218 tasks form 41 groups of 1–40 tasks each; merging identical source questions across skills gives 31 source questions.

Representative selection. Each group selects one task by the following rules in order, stopping at the first rule that discriminates:

1. By the source role in the construction records: tasks marked positive first, then tasks without a role label, and control tasks last.
2. Within the same priority, the task that comes first in the version order of the source index.
3. If still tied, the task whose directory name comes first in lexicographic order.

B Method Details of EBU-Evolve

This appendix gives the reflection inputs, candidate construction, and selection rules of EBU-Evolve as shown in Figure 1c.

B.1 Paired Evidence and Reflection Inputs

For the original skill s_0, we run Natural Skill Use and Operationally Supported Skill Use under the same model, harness, and execution limits. Both sides receive the same task inputs and skill resources; the latter additionally appends a predetermined operational-support prompt. The prompt requires reading or applying the existing resources and does not supply the answer to the current task. For the tasks authored around research questions, the prompt requires reading the skill entry file and consulting the necessary reference materials it points to.

The reflection model receives the following materials:

When selecting complete pairs, we prioritize cases whose Natural Skill Use score improves, include unchanged or degraded cases when the input limit allows, and fill the remainder in a predetermined order. If the context budget is exceeded, the number of cases is reduced while every selected trace is kept in full; the overview of all training pairs is still supplied alongside the complete cases. The overview may be cited for scores and access facts, whereas specific revisions must be supported by complete cases and the original materials. The reflection model maps the differences observed in

Material	Content and use
Original skill and reference materials	Provide the rules, applicability conditions, and operational basis for the revision
Overview of all training pairs	Records both sides' task scores and access facts, showing improved, unchanged, and degraded pairs
Selected complete pairs	Keep the prompt, tool calls and returns, final answers, task scores, and access records for analyzing specific changes
Training access diagnosis	Distinguishes instructions not read, instructions read but misapplied, and access status unknown

the paired executions to existing rules; pairs without improvement are used to test the scope of the proposed revisions.

Access status is judged from explicit operations in the execution records, not inferred from answer correctness or the appearance of file names; it is left unknown when records are missing. If the agent did not read the relevant instructions, the reflection model considers whether the description needs adjusting; if it did, the model further examines how the rules were applied.

The reflection output includes the pairs supporting the change, counterexamples to consider, citations to the original materials, reusable revision rules, and suggested locations. Revision rules state which applicability conditions or execution steps should be clarified, moved earlier, or reorganized, covering a rule's applicability conditions, execution steps, and how its results are used. Citations must be locatable in the original skill package. Neither reflection nor editing receives per-task prompts, answers, traces, or scores from the validation or test sets.

B.2 Candidate Construction with a Shared Body

Revision rules take effect only if the agent reads and correctly applies them during autonomous execution; candidates therefore consider both the description and the body.

Let the original instructions be $s_0 = (d_0, b_0)$, with description d_0 and body b_0. From the original text, the revision rules, and the training access diagnosis, the editing model produces a new body b' and a new description d', from which we construct

$$s_{\text{body}} = (d_0, b'), \qquad s_{\text{joint}} = (d', b').$$

The body candidate keeps all original metadata; the joint candidate changes only the description. We check that the bodies of the two candidates are identical and that the other metadata are preserved. The two proposals are then reviewed separately; the review may keep both, one, or neither.

The description tells the executing agent when the skill applies and which requirements are essential, while the body organizes the complete rules and steps. Added content must preserve the applicability conditions of the original method and must not store training-task identifiers, specific values, or answers.

B.3 Candidate Review

Editing is limited to the body of `SKILL.md` and the description where permitted; other metadata, reference files, code, and data remain unchanged. Every candidate is subject to an edit-size cap, computed as the number of lines added plus lines deleted relative to the parent version, where replacing a line counts as one deletion and one addition. Passages designated for preservation must remain locatable in the candidate.

A source check verifies newly added identifiers, existing dependencies, and links to reference files. An independent semantic review further checks whether the changes are supported by the original materials, whether the applicability conditions of a rule have been altered incorrectly, and whether the description still addresses the executing agent. Proposals that fail the review are not evaluated.

A candidate that fails the review may be repaired once according to the review comments; the repair does not use feedback from the validation or test sets.

B.4 Validation-Set Selection

Selection is performed under Natural Skill Use, reproducing how a skill is read and executed by the agent on its own in practice.

The original skill and the eligible candidates are evaluated under Natural Skill Use on the same validation set, with the same model, harness, and per-task execution limits. A candidate must complete all scheduled tasks and repetitions with a valid score each time; candidates with incomplete evaluations do not enter the selectable set, and missing scores are not filled with zeros.

To stabilize selection, each task is repeated K times ($K = 3$ in this paper); scores are first averaged within a task and then across tasks:

$$J(s) = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \left[\frac{1}{K} \sum_{k=1}^{K} y_{i,k}(s) \right].$$

Versions are ranked by $J(s)$. A tie with the original keeps the original; a tie between revisions prefers the one with the smaller edit relative to the original, then a fixed candidate order. A revision is adopted only if its score strictly exceeds the original. The original always takes part in the comparison, so that no revision is adopted without an actual improvement. Skill-opening rate, process predicates, and the EBU gap do not enter the selection score.

B.5 Budget and Algorithm

Each search fixes in advance the call limits for execution, reflection, editing, and review, and records the actual usage of each. One edit yields two proposals, and the task evaluations of both candidates count against the execution budget. The final test is accounted separately, and test feedback is never used to generate or select candidates. EBU-Evolve performs one round of paired diagnosis and candidate selection on the original skill. The algorithm is as follows:

```
Input: original skill s0; training set T, validation set V,
       test set U; fixed operational-support prompt; search budget

Evaluate s0 on V under Natural Skill Use.
Collect paired executions of s0 on T under Natural Skill Use
    and Operationally Supported Skill Use.
Reflect over the complete pairs, the training overview, and
    the original materials to obtain reusable revision rules.
If the evidence does not support a revision, select s0 and
    go to the final test.

Generate a new body b' and a new description d' from the rules.
Construct (d0, b') and (d', b'); check structure, sources,
    and semantics.
Evaluate the eligible candidates on V under Natural Skill Use;
    keep the versions whose evaluation is complete.

Select the version with the highest task score; keep s0
    unless a candidate strictly exceeds it.
Freeze the selected skill and compare it with s0 on U;
    test feedback does not re-enter revision or selection.
```

B.6 Final Evaluation and Information Use

Training, validation, and test sets are split in advance by task ID and used, respectively, to generate revisions, select the version, and evaluate the final skill. The selected version is handed to the agent as an ordinary skill; later tasks carry no training traces, reflection records, or operational-support prompts. Cross-model and cross-harness evaluations use the same text and compare against the original skill in the target setting; target test results are not used to re-edit or re-select the version.

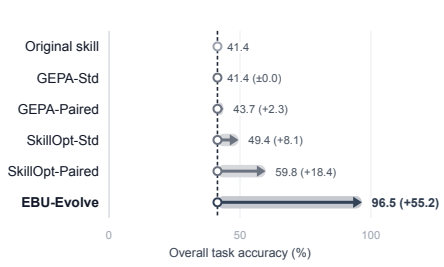
B.7 Data Splits and Execution Settings

The numbers of training, validation, and test tasks of each skill are given in Table B.1; the three source models use the same split, except that the Enrichment review training and validation sets of GPT-5.5 contain 14 and 10 tasks. Within a target configuration, the original and evolved skills use the same execution settings; the output limit of a single run is 49,152 tokens (64,000 for Atria). Runs interrupted by infrastructure failures are rerun according to the protocol.

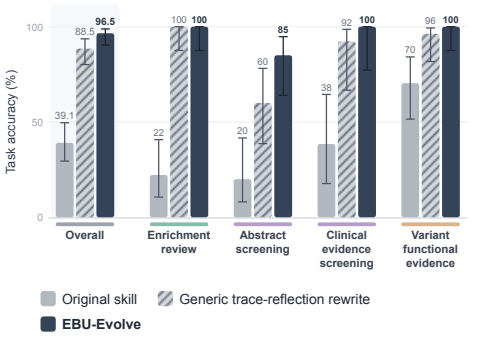
Table B.1: Task splits of each skill (number of tasks).

Skill	Training	Validation	Test
Abstract screening	16	16	20
Clinical evidence screening	22	18	13
Enrichment review	11	13	27
Variant functional evidence	13	13	27
Total	62	60	87

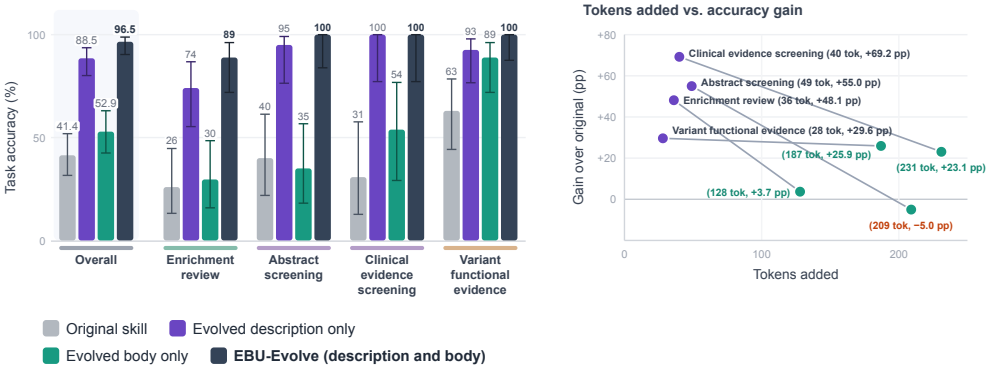
B.8 Ablations



(a) Comparison with GEPA and SkillOpt



(c) Comparison with generic trace-reflection rewriting



(b) Decomposition into description and body

Figure B.1: Method comparison and ablations, all evaluated under GLM-5.2/Claude Code. (a) Comparison of the frozen artifacts with the prompt-optimization and skill-optimization methods; (b) ablation of the presentation components of the frozen GLM-5.2 artifact, where the scatter plot on the right shows the added text and test gain of the description-only and body-only revisions, with tokens counted uniformly by o200k_base; (c) comparison of EBU-Evolve with generic trace-reflection rewriting, run as a separate batch and not pooled with (a) or (b). Error bars are per-bar Wilson 95% intervals.

B.8.1 Decomposition into Description and Body

Let the frozen joint version of GLM-5.2 be $s_{DB} = (d', b')$; we split off the description-only version $s_D = (d', b_0)$ and the body-only version $s_B = (d_0, b')$ and compare them with the original $s_0 = (d_0, b_0)$. s_D and s_B retain, respectively, the description and the body actually revised in the joint version.

On the frozen GLM-5.2 artifact, the joint revision attains the highest skill-use accuracy, 96.55%; the description-only revision reaches 88.51% and the body-only revision 52.87% (Table B.2, Figure B.1b). The description-only revision retains 85.42% of the net gain of the joint revision over the original. It outperforms the body-only revision on all four skills while adding only 28–49 tokens, compared with 128–231 tokens for the body revision (Table B.3), indicating that a small revision of the entry point can already bring a large Natural Skill Use gain. The complementary effect of the body is more visible when combined with the new description: in Enrichment review, revising the body alone raises skill-use accuracy from 25.93% to 29.63%, a gain of 3.70 pp, whereas adding the same body on top of the new description raises it from 74.07% to 88.89%, a gain of 14.81 pp. This suggests that the effect of a body revision should be assessed together with the description: the description may guide the agent to read further or may itself supply the key rule.

Table B.2: Accuracy (%) of the decomposed versions under Natural Skill Use.

Skill	Original	Description only	Body only	Joint
Abstract screening	40.00	95.00	35.00	100.00
Clinical evidence screening	30.77	100.00	53.85	100.00
Enrichment review	25.93	74.07	29.63	88.89
Variant functional evidence	62.96	92.59	88.89	100.00
Overall	41.38	88.51	52.87	96.55

Table B.3: Length of the full `SKILL.md` (o200k_base tokens).

Skill	Original	Description only	Body only	Joint
Abstract screening	89	138	298	347
Clinical evidence screening	84	124	315	355
Enrichment review	91	127	219	255
Variant functional evidence	66	94	253	281

B.8.2 Comparison with Generic Trace-Reflection Rewriting

The setup of this control is described in §4.4; it is run as a separate batch under GLM-5.2/Claude Code, so its original-skill accuracies differ from those in Figures 3 and B.1a. Both groups receive the original skill and the complete reference materials; the comparison concerns how revision evidence is obtained from execution. EBU-Evolve uses the paired executions of the same task under Operationally Supported Skill Use to locate where an existing rule was not acquired or not carried out, and writes the corresponding requirement back into the skill. Overall, EBU-Evolve answers 7 more of the 87 tasks correctly, a skill-use accuracy 8.04 pp higher (96.55% vs. 88.51%); 5 of these come from Abstract screening (85.00% vs. 60.00%), contributing 5.75 pp, while Clinical evidence screening (100.00% vs. 92.31%) and Variant functional evidence (100.00% vs. 96.30%) each contribute one more correct answer, and both groups reach 100.00% on Enrichment review (Table B.4, Figure B.1c). The results support the value of EBU-Evolve's overall EBU-driven design.

B.9 Behavioral Cases

B.9.1 Rule Not Acquired: Variant Functional Evidence

On one Variant functional evidence training task, the agent under Natural Skill Use answers after reading only the main SKILL.md and gets it wrong; under Operationally Supported Skill Use, it goes

Table B.4: Per-skill comparison with generic trace-reflection rewriting (accuracy under Natural Skill Use, %).

Skill	Original	Generic trace-reflection rewriting	EBU-Evolve
Abstract screening	20.00	60.00	85.00
Clinical evidence screening	38.46	92.31	100.00
Enrichment review	22.22	100.00	100.00
Variant functional evidence	70.37	96.30	100.00
Overall	39.08	88.51	96.55

on to read the secondary reference files index, functional, records, frequency, and segregation, and answers correctly. Reflection accordingly proposes a rule that points explicitly to the reference files: the original skill only asked, in general terms, to "locate the applicable material via the reference index", whereas the evolved skill requires opening the applicable reference files first, extracting the decision rules, and "not answering from this SKILL.md alone".

B.9.2 Condition Recognized but Not Applied: Enrichment Review



Figure B.2: Intervention-to-instruction distillation in Enrichment review. The figure shows the executions of one training task under Natural Skill Use and Operationally Supported Skill Use, together with the actual skill revision; the test-task case discussed in the text is a different task, not a re-run of the training task shown here after evolution.

Figure B.2 shows the paired executions of GLM-5.2 on one Enrichment review training task and the corresponding skill revision. Under Natural Skill Use, the agent recognizes 12 assay-eligible entities, yet excludes 2 of them at computation time because they lack term annotations, shrinking the background set to 10; this run reads neither the skill nor the reference materials. Under Operationally Supported Skill Use, these entities are kept and the agent answers correctly. The original skill refers to the software draft as material pending review, and the construction rule for the background set appears only in a reference file; the evolved skill requires rebuilding, before any statistics, the background set of assay-eligible entities under the currently signed-off conditions, rather than inheriting the annotation universe of the draft.

On another Enrichment review test task that took part in neither revision nor version selection, GLM-5.2 reads the same files under the original and the evolved skill: both runs read the main SKILL.md and neither reads the secondary references. The original computes with only the 11 annotated entities (5 candidates) and repeatedly re-checks the statistics without noticing that the object of computation is itself misdefined; the evolved skill keeps all 12 assay-eligible entities (6 candidates) according to the new instruction and answers the task correctly.

C Evaluation Configuration and Execution Protocol

C.1 Model and Harness Configuration

All evaluations are executed through BenchFlow 0.7.7; each run uses an independent container, with external retrieval and data downloads disabled. The two harness versions are fixed across all runs: Claude Code 2.1.257, and Codex 0.148.0 invoked via codex-acp 1.6.0. Table C.1 lists the request identifiers of the nine models.

Table C.1: Model request identifiers.

Model	Request identifier
GPT-5.5	`gpt-5.5`
Claude Opus 4.8	`claude-opus-4-8`
Gemini 3.8 Flash	`gemini-3.8-flash`
DeepSeek V4 Pro	`deepseek-v4-pro-0813`
GLM-5.2	`glm-5.2` (some requests use the alias `glm-5.2-1m` of the same model)
Atria	`Atria-Dawn-Preview`
Kimi K2.7 Code	`kimi-k2.7-code`
Qwen3.8 Max	`qwen3.8-max`
Qwen3.8 27B FP8	`qwen3.8-27b-fp8`

C.2 Scope of Reruns

The initial time limit is 900 or 3,600 seconds depending on the batch. The 244 evaluation cells that timed out under the 900-second limit (2.2% of 11,088) were rerun with a budget of up to 7,200 seconds, and the rerun replaced the original record; 55 of them completed after the rerun. If these 55 cells are all counted as failures instead, the mean EBU gap is 43.72 pp, 0.22 pp below the 43.94 pp reported in the main text.

C.3 Distribution of Invalid Runs

In the final evaluation, 240 runs (2.2%) did not receive a valid score and were scored 0: 234 did not submit a valid answer and 6 timed out. The four conditions account for 63, 55, 58, and 64 of these runs, so they are not concentrated in any one condition.

C.4 Operational-Support Text by Task Source

Operationally Supported Skill Use reuses the text fixed for each task group. For tasks derived from public benchmarks, the prompt requires the agent to read the associated skill in full, identify the applicable procedure, and apply its steps to the current inputs, for example by extracting and converting calculation parameters or by running code to obtain the required statistics. For tasks authored around research questions, the prompt requires reading the entry file and consulting the necessary reference materials it points to. Neither form of support supplies the answer to the current task, but their requirements differ in specificity, which is why the main text also reports results by task source.

D Per-Cell Results of the Main Evaluation

Table D.1 lists, for nine models under two harnesses and four conditions, the skill-use accuracy, the number of successes, and the EBU gap, corresponding to Figure 2 in the main text.

Table D.1: Skill-use accuracy and EBU gap of each model–harness combination. Each cell gives "skill-use accuracy (successes/tasks)"; every combination is evaluated on the same 154 tasks, and the EBU gap is in pp. The last two rows are the equally weighted means of the nine models under each harness, which, since the task counts are equal, also equal the pooled successes divided by 1,386. Each combination keeps the score of one run, with invalid runs scored 0 (Appendix C). †: GLM-5.1, used for task screening, belongs to the same model family as GLM-5.2, which may inflate the EBU gap of that family.

Model	Harness	No Skill	Natural Skill Use	Prompted Skill Use	Operationally Supported Skill Use	EBU gap
GPT-5.5	Claude Code	20.13 (31/154)	18.18 (28/154)	21.43 (33/154)	80.52 (124/154)	62.34
GPT-5.5	Codex	19.48 (30/154)	42.86 (66/154)	39.61 (61/154)	71.43 (110/154)	28.57
Claude Opus 4.8	Claude Code	18.83 (29/154)	35.71 (55/154)	52.60 (81/154)	87.01 (134/154)	51.30
Claude Opus 4.8	Codex	19.48 (30/154)	37.66 (58/154)	46.75 (72/154)	89.61 (138/154)	51.95
Gemini 3.8 Flash	Claude Code	20.78 (32/154)	35.71 (55/154)	70.78 (109/154)	82.47 (127/154)	46.75
Gemini 3.8 Flash	Codex	24.03 (37/154)	68.18 (105/154)	68.18 (105/154)	76.62 (118/154)	8.44
DeepSeek V4 Pro	Claude Code	23.38 (36/154)	25.97 (40/154)	22.73 (35/154)	72.08 (111/154)	46.10
DeepSeek V4 Pro	Codex	22.08 (34/154)	64.94 (100/154)	71.43 (110/154)	79.87 (123/154)	14.94
GLM-5.2†	Claude Code	14.94 (23/154)	15.58 (24/154)	16.23 (25/154)	88.96 (137/154)	73.38
GLM-5.2†	Codex	16.88 (26/154)	30.52 (47/154)	43.51 (67/154)	92.21 (142/154)	61.69
Atria	Claude Code	17.53 (27/154)	18.83 (29/154)	18.83 (29/154)	84.42 (130/154)	65.58
Atria	Codex	19.48 (30/154)	55.19 (85/154)	60.39 (93/154)	87.66 (135/154)	32.47
Kimi K2.7 Code	Claude Code	22.08 (34/154)	51.30 (79/154)	57.79 (89/154)	77.27 (119/154)	25.97
Kimi K2.7 Code	Codex	21.43 (33/154)	55.19 (85/154)	58.44 (90/154)	72.08 (111/154)	16.88
Qwen3.8 Max	Claude Code	19.48 (30/154)	21.43 (33/154)	25.32 (39/154)	88.31 (136/154)	66.88
Qwen3.8 Max	Codex	20.78 (32/154)	59.09 (91/154)	66.88 (103/154)	81.17 (125/154)	22.08
Qwen3.8 27B FP8	Claude Code	12.99 (20/154)	15.58 (24/154)	48.05 (74/154)	87.01 (134/154)	71.43
Qwen3.8 27B FP8	Codex	14.94 (23/154)	38.96 (60/154)	50.65 (78/154)	83.12 (128/154)	44.16
Mean over models	Claude Code	18.90 (262/1,386)	26.48 (367/1,386)	37.09 (514/1,386)	83.12 (1152/1,386)	56.64
Mean over models	Codex	19.84 (275/1,386)	50.29 (697/1,386)	56.20 (779/1,386)	81.53 (1130/1,386)	31.24

E Runs That Call Unregistered Skill Names

We count the cases in which the skill name or path requested by the agent is not registered in the current run. Such requests return an "unknown skill" error, after which the agent must search again or give up using the skill. The count covers the three conditions that provide the associated skill, 8,316 runs in total (9 models × 2 harnesses × 154 tasks × 3 conditions).

These runs are concentrated in Qwen3.8 Max/Claude Code under Prompted Skill Use (Table E.1). In that cell, 60/154 runs call an unregistered skill name: 56 never open the target skill during the whole run yet all submit a valid answer, and the other 4 subsequently open the target skill. The same model and harness show no such calls under Natural Skill Use and 23/154 under Operationally Supported Skill Use, all of which subsequently open the target skill. The Prompted Skill Use row of Figure 1a is one of these cases: the agent calls the unregistered `acute_medical_calculator`, receives "Unknown skill", and answers directly without searching for the target skill.

Under Prompted Skill Use, the 17 combinations other than Qwen3.8 Max/Claude Code call unregistered skill names in 6/2,618 runs in total, and all 6 subsequently open the target skill.

Across the three conditions there are 110 such requests, involving 88 distinct names or paths. Six are close to the target skill's name (e.g., spelling variants), six are abbreviations of the target skill's category (e.g., `medcalc`), and the remaining 98 are other unlisted names or paths. The most frequent are `medcalc` (6 times), `medical-calculator` and `multitable-derived-variables-and-association-test` (4 each); the other 85 names each occur 1–2 times.

Table E.1: Number of runs that call an unregistered skill name.

Model	Harness	Natural Skill Use	Prompted Skill Use	Operationally Supported Skill Use
Qwen3.8 Max	Claude Code	0/154	60/154 (56 never open the target skill)	23/154 (all open the target skill)
Qwen3.8 27B FP8	Claude Code	0/154	3/154 (all open the target skill)	0/154
Kimi K2.7 Code	Claude Code	3/154 (all open the target skill)	2/154 (all open the target skill)	0/154
Atria	Codex	1/154 (opens the target skill)	1/154 (opens the target skill)	0/154
Other 14 model–harness combinations		0	0	0
Total		**4/2,772**	**66/2,772**	**23/2,772**

F THE FULL EBU-BENCH

GLM-5.2, Atria, and DeepSeek V4 Pro completed the four-condition evaluation on all 331 tasks of EBU-Bench under Claude Code and Codex (Figure F.1, Table F.1). The setup is the same as in Section 4, and the skill-use accuracy of each model–harness combination uses 331 tasks as the denominator; the only difference is that the 177 tasks outside EBU-Bench Lite received no time-out reruns, so timed-out runs are scored 0 directly (86, 25, and 16 runs for GLM-5.2, Atria, and DeepSeek V4 Pro). Even under the least favorable assumption about these timed-out runs, all six combinations retain a positive EBU gap.

The results on the full benchmark agree with EBU-Bench Lite. All six combinations show a positive EBU gap, ranging from 11.48 to 77.04 pp (14.94–73.38 on Lite). Under both harnesses the gaps of the three models are ordered DeepSeek V4 Pro < Atria < GLM-5.2, as on Lite, and each model's gap under Claude Code exceeds its gap under Codex. Averaging the three models with equal weight, the EBU gaps of Claude Code and Codex are 64.35 and 37.87 pp.

Table F.1: Skill-use accuracy on the full benchmark (%, with successes/tasks in parentheses) and EBU gap (pp). The last column is the EBU gap of the same combination on EBU-Bench Lite (154 tasks), corresponding to the main results.

Model	Harness	No Skill	Natural Skill Use	Prompted Skill Use	Operationally Supported Skill Use	EBU gap	Lite EBU gap
DeepSeek V4 Pro	Claude Code	36.86 (122/331)	39.58 (131/331)	37.16 (123/331)	84.89 (281/331)	45.32	46.10
DeepSeek V4 Pro	Codex	34.44 (114/331)	77.64 (257/331)	81.57 (270/331)	89.12 (295/331)	11.48	14.94
Atria	Claude Code	18.13 (60/331)	18.43 (61/331)	26.89 (89/331)	89.12 (295/331)	70.69	65.58
Atria	Codex	16.01 (53/331)	51.36 (170/331)	63.75 (211/331)	88.82 (294/331)	37.46	32.47
GLM-5.2†	Claude Code	16.01 (53/331)	12.99 (43/331)	18.43 (61/331)	90.03 (298/331)	77.04	73.38
GLM-5.2†	Codex	11.48 (38/331)	29.91 (99/331)	47.13 (156/331)	94.56 (313/331)	64.65	61.69

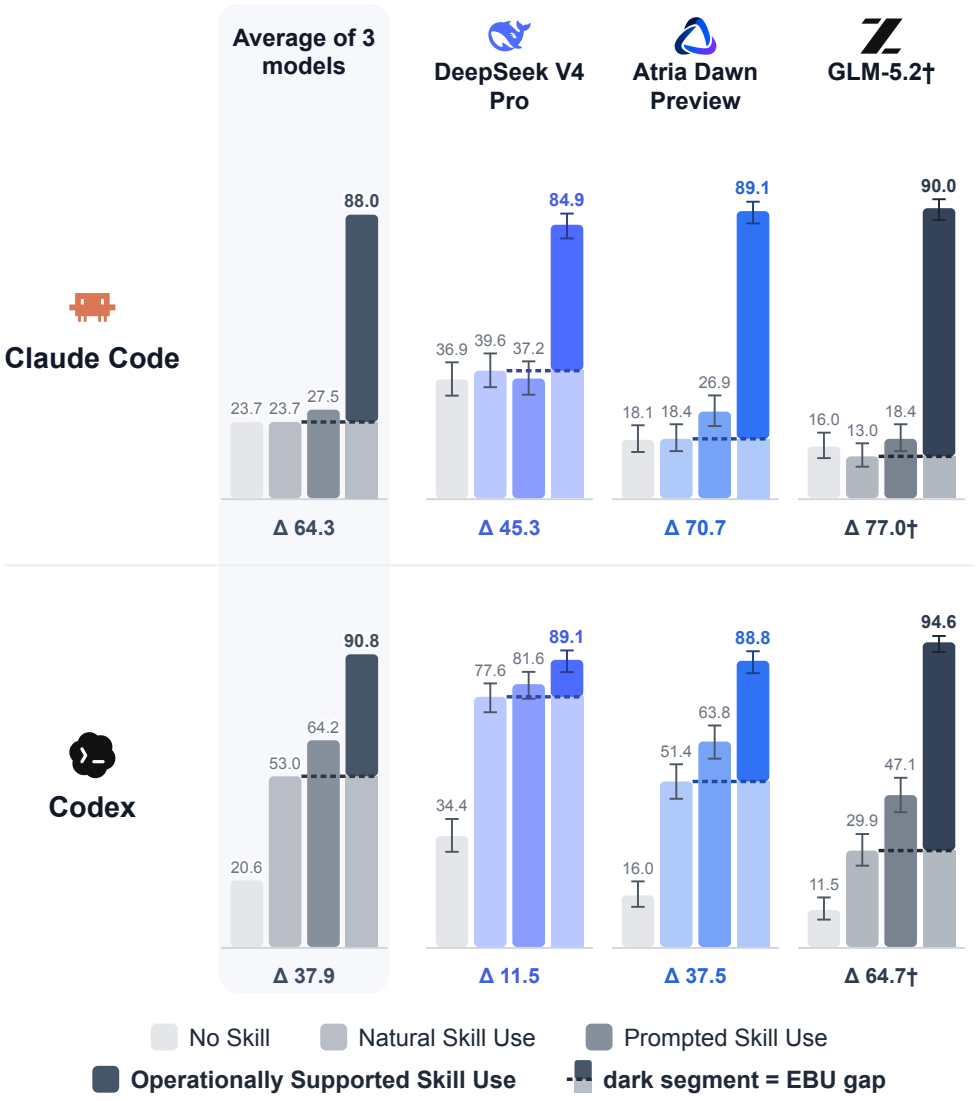


Figure F.1: Skill-use accuracy (%) of GLM-5.2, Atria (labeled Atria Dawn Preview in the figure), and DeepSeek V4 Pro under four conditions on the full EBU-Bench (331 tasks). The layout follows the main-results figure: the top row is Claude Code, the bottom row Codex, and the leftmost column the equally weighted mean of the three models; the Operationally Supported Skill Use bar is split at the Natural Skill Use accuracy, the dark upper segment is the EBU gap, and Δ below each bar is its value in pp. Error bars are per-bar Wilson 95% intervals (n = 331). †: GLM-5.1, used for task screening, belongs to the same model family as GLM-5.2.

G Counts for Skill-Opening Rates and Paired Analysis

Table G.1 gives the counts behind the opening rates and the paired analysis in §5.1. A skill counts as opened once the target skill is successfully read or loaded. The denominator of an opening rate is the number of runs whose opening status can be determined from the execution records.

Over both harnesses, the net-success increase of Prompted Skill Use relative to Natural Skill Use is 229, which is 18.8% of the net successes corresponding to the EBU gap (1,218).

Table G.1: Counts for the opening rates and the same-task paired analysis. A pair consists of the Natural Skill Use and Operationally Supported Skill Use runs of the same model, harness, and task; net successes are the difference between the success counts of the two conditions.

Statistic	Claude Code	Codex
Opening rate under Natural Skill Use	12.45% (171/1,373)	65.42% (906/1,385)
Opening rate under Operationally Supported Skill Use	96.90% (1,343/1,386)	99.64% (1,381/1,386)
Net successes corresponding to the EBU gap	785	433
Of which from pairs opened under both conditions	27	182 (42.0%)
Number of pairs opened under both conditions	169	902
Correct answers of these pairs under Natural Skill Use	117 (69.23%)	577 (63.97%)
Correct answers of these pairs under Operationally Supported Skill Use	144 (85.21%)	759 (84.15%)

In the decomposition of the net EBU gap by opening status (§5.1), a small number of pairs leave the skill unopened under Operationally Supported Skill Use although it was opened under Natural Skill Use; these pairs contribute -0.22 pp in total.

H A TRACE IN WHICH A MEMORIZED DEFINITION OVERRIDES THE SPECIFIED RULE

This appendix gives the Atria/Codex trace summarized in §5.1. In the Delta gap task (Na^+ = 130, Cl^- = 96, HCO_3^- = 12 mEq/L), the model under Natural Skill Use has read the skill's rule AG − 12 and states that it is "unambiguous", yet submits −2, computed from the memorized definition $\Delta AG - \Delta HCO_3$. Under Operationally Supported Skill Use, the same model follows the resource, obtaining AG = 130 − (96 + 12) = 22 and hence Delta gap = 10. Against the frozen specification of this task, the two runs are scored as a failure and a success, respectively. This single case cannot establish the overall frequency of such errors; it illustrates the risk that, in medical tasks governed by a specification, a model may read the given rule and still substitute a familiar definition, producing a plausible result that does not comply with the provided specification.