SkillsBench's Skill gains come mainly from scoring rules absent from the task statements

Author: Benchmarks' Last Exam Lab 2026-11-12

An Agent Skill is a set of instructions and scripts mounted for an agent to explain how a kind of task is done. SkillsBench (arXiv:2602.12670) is the first benchmark built to evaluate Skills. Its design is direct: an agent attempts each task twice, once with the task statement alone and once with a human-written Skill added, and the difference in pass rate is the gain from the Skill. The paper tests 18 combinations of model and harness, called configurations below, on 87 tasks. It reports that Skills raise the average pass rate from 33.9% to 50.5%, an average gain of 16.6 percentage points.

Each task has four parts: the task statement, a verifier that grades the output, a reference solution from the author, and a Skill directory. We read all four parts of all 87 tasks, then used the official per-task scores to split the 16.6-point average gain across individual tasks. About 79% of the gain comes from two kinds of task. In the first, a scoring rule is stated only in the Skill and is missing from the task statement. In the second, the Skill ships a near-complete solution script. The gain from Skills as transferable knowledge is real and accounts for about a quarter.

Three high-gain tasks

Start with `sec-financial-report`. The task asks the agent to parse an SEC 13F holdings file, and one question asks for the number of common-stock holdings. The verifier requires the answer to equal 2451 exactly. Without the Skill, each of the 18 configurations ran three times and none of the 54 runs passed. The cause lies in the reference solution. It filters common stock with a list of class names, and 15 lines in the middle of that list lack a comma:

```
    "com shs",
    "sponsored adr"
    "sponsored ads"
    "adr"
    "equity"
    ...
    "foreign stock"
    "shs cl a",
]
```

Python joins adjacent string literals. The 16 class names from `"sponsored adr"` to `"shs cl a"` therefore merge into one long string, the list shrinks from 25 entries to 10, and all 16 classes fall outside it. The number 2451 is what this truncated list produces. The script `one_fund_analysis.py` in the Skill directory carries the same list with the commas missing in the same places, so the pass rate with the Skill reaches 68.5%. The 68.5-point gain on this task separates agents that received the author's faulty script from agents that did not.

The second task is `dapt-intrusion-detection`. It provides a packet capture and asks the agent to compute traffic metrics and decide whether a DoS attack is present. The task statement describes DoS only as an "extreme traffic spike / flood-like rate" and gives no threshold. The answers hard-coded in the verifier are a maximum of 2372 packets per minute, an average of 262.9, and no DoS. The example section of the Skill document reads `ppm_max = 2372, ppm_avg = 262.9`, notes that the ratio of 9.02 is below 20, and concludes NO DoS. The example values match the verifier's answers one by one, and the threshold of 20 appears nowhere else. The pass rate is 0 without the Skill and 81.5% with it. The paper's intake process includes a check that is said to reject Skills containing magic numbers or expected outputs. This task passed it.

The third task is `earthquake-plate-calculation`. It asks for the earthquake inside the Pacific plate that lies farthest from the plate boundary, and for its distance to the boundary. The task statement says to use GeoPandas projections and names no coordinate system. The verifier requires a distance of 3878.27 km with a tolerance of 0.01 km, which is 10 metres. Over thousands of kilometres, a different projection or an ellipsoidal geodesic usually gives a distance that differs by far more than 10 metres. The reference solution uses the coordinate system EPSG:4087. That code is absent from the task statement and appears several times in the Skill, together with nearly complete calculation code. The pass rate rises from 3.7% without the Skill to 68.5% with it. Figure 1 lays out who holds which information in this task.

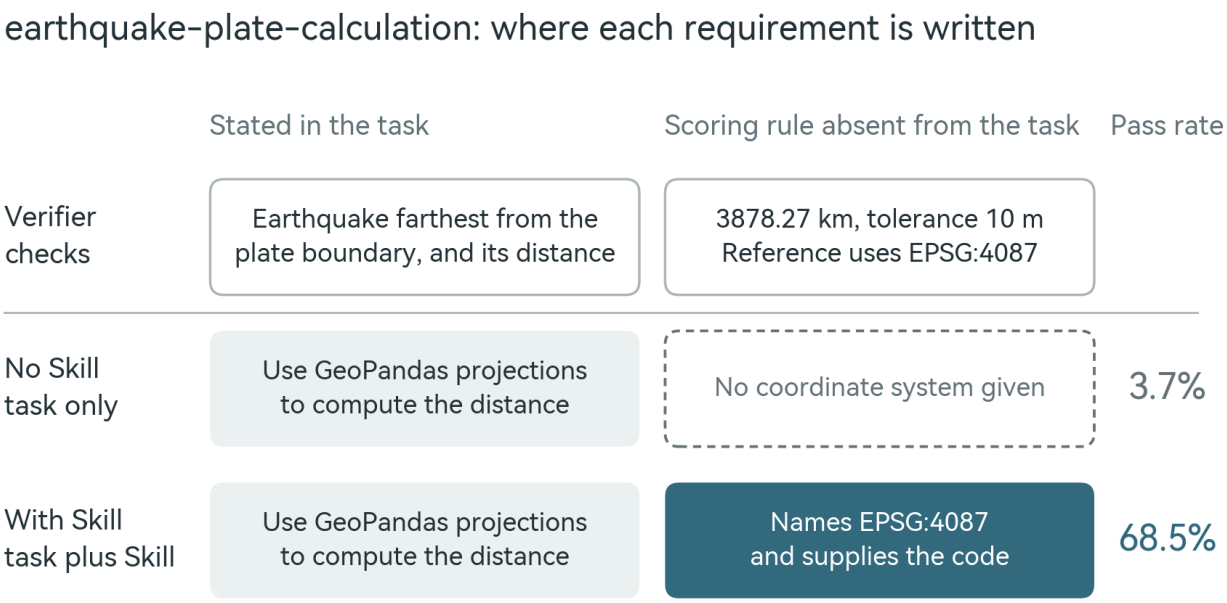


Figure 1. Who holds which information in earthquake-plate-calculation. The distance the verifier expects can only be obtained in the EPSG:4087 coordinate system. That rule is absent from the task statement and present in the Skill.

The three tasks have one thing in common: a rule the verifier checks is missing from the task statement and written only in the Skill. We call such a rule a hidden rule below. The paper presents the ten tasks with the highest gains as representative evidence that Skills work, with an average gain of 67 percentage points. By our labels, the gain on nine of them comes mainly from hidden rules.

What makes up the 16.6-point gain

We sorted the 87 tasks into five classes by the main way the Skill takes effect. The first four classes are assigned from the source code alone. Tasks whose gain is at most 2 percentage points, or negative, form the last class.

The average gain splits linearly across tasks. A class contributes its share of the tasks multiplied by its within-class average gain. Take the first class: 27 tasks are 31% of the benchmark, the within-class average gain is 33.2 points, and their product is 10.3 points, or 62.1% of the 16.6. The table below lists the pass rates and within-class average gain of each class, and Figure 2 shows each class's contribution to the average gain.

How the Skill takes effect	Tasks	Pass rate (no Skill → with Skill) and average gain within class
Scoring rules stated only in the Skill	**27**	18.4% → 51.6%, a gain of 33.2 points

How the Skill takes effect	Tasks	Pass rate (no Skill → with Skill) and average gain within class
Skill ships a near-complete solution script	**12**	57.9% → 78.1%, a gain of 20.2 points
Tool and library usage	**17**	34.0% → 49.6%, a gain of 15.5 points
Domain methods	**8**	48.0% → 59.6%, a gain of 11.6 points
No effect or harmful	**23**	34.7% → 32.4%, a drop of 2.2 points
All	**87**	33.9% → 50.5%, a gain of 16.6 points

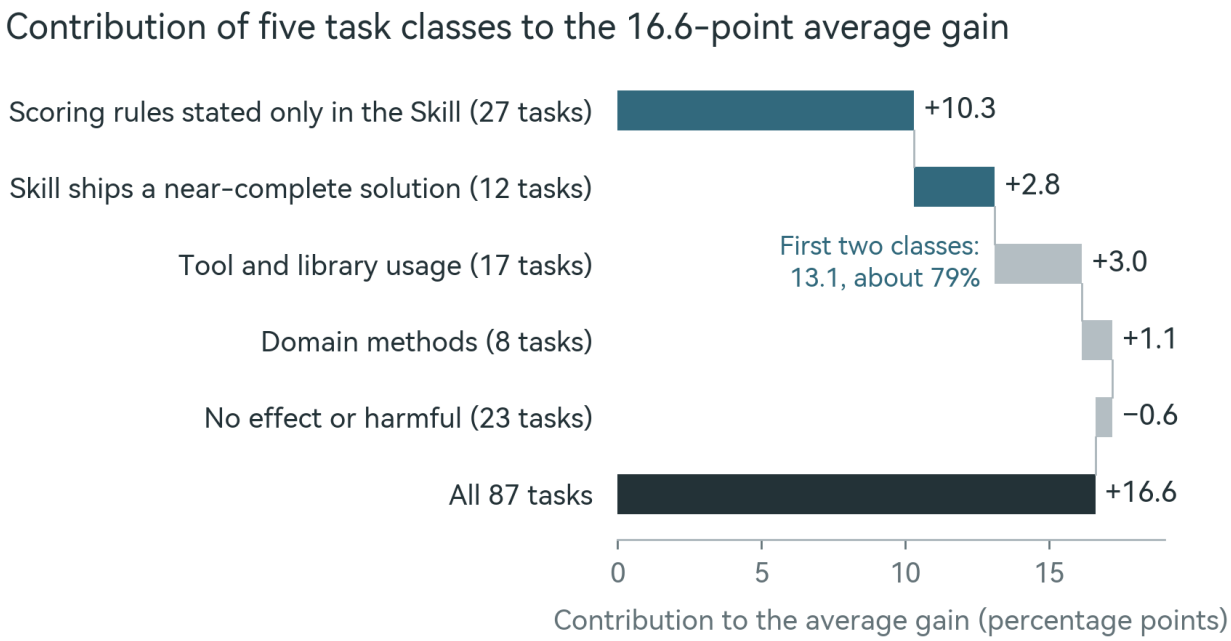


Figure 2. Composition of the average gain. Each segment is one class's contribution to the 16.6 points, equal to its share of tasks times its within-class average gain. The first two classes add up to 13.1 points.

The first two classes together hold 39 tasks, fewer than half, and contribute 13.1 of the 16.6 points, about 79%.

The third and fourth classes are close to what a Skill is meant to be. These Skills remind the agent that formulas are not recalculated after openpyxl saves a file and that LibreOffice has to be called, or they document the interfaces of seismology toolkits such as SeisBench and GaMMA. The within-class average gain on these 25 tasks is 14.3 points. Their contribution to the average gain over all 87 tasks is 4.1 points, about 25%.

On the 23 tasks of the last class the Skills do not help, and the pass rate falls by 2.2 points on average.

High gains come from a depressed pass rate without Skills

There is another possible reading of the high gains: these tasks are simply harder and therefore need Skills more. Grouping the tasks by the severity of their hidden rules tests this reading, and Figure 3 shows the result. Major means that a reasonable solution written from the task statement is marked wrong by the verifier. Minor means that details such as key names or rounding are ambiguous in the task statement.

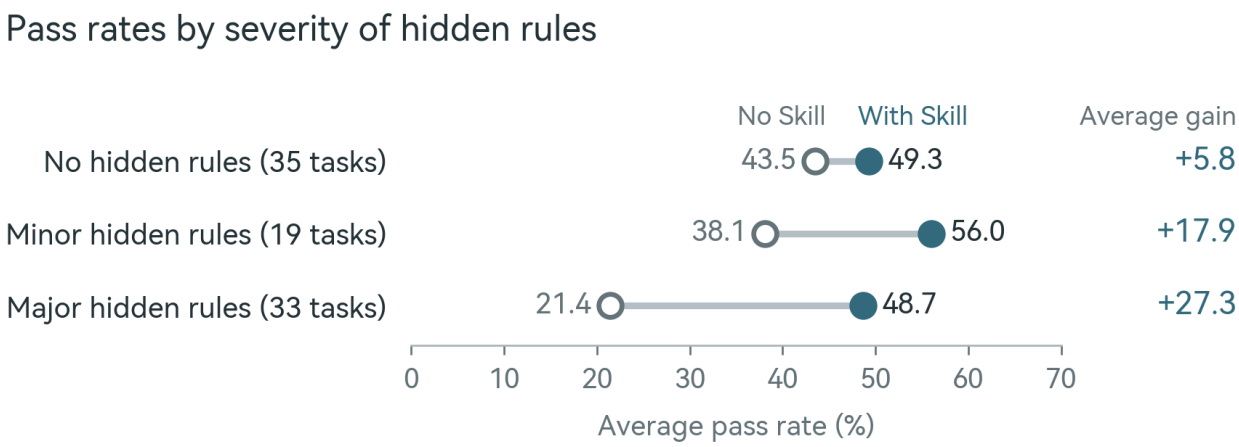


Figure 3. Average pass rates by severity of hidden rules. Hollow dots are runs without Skills, filled dots are runs with Skills, and the number on the right is the difference. The three groups reach similar pass rates with Skills and differ mainly without them.

The 35 tasks without hidden rules and the 33 tasks with major hidden rules reach almost the same pass rate with Skills, 49.3% and 48.7%. The two groups differ in the pass rate without Skills, which is 43.5% for the former and only 21.4% for the latter. If the second group were simply harder, its pass rate with Skills would usually be lower too. The data show a different situation. The verifier checks rules that the task statement omits, so an agent without the Skill rarely matches them even when its approach is sound, and its pass rate is pushed down. Once the Skill supplies the rules, the pass rate returns to the level of the other tasks, and the difference between the two becomes a large gain.

The gain is just as concentrated when tasks are grouped by whether the Skill carries code. For 35 tasks the Skill is prose only, with no code written for the task, and their average gain is 1.7 points. The other 52 tasks, whose Skills carry code templates or near-complete scripts, contribute about 96% of the average gain.

The gain on the clean subset

Thirty tasks have no hidden rules and a Skill without a near-complete solution. We call them the clean subset. On the clean subset, Skills raise the average pass rate from 41.9% to 44.0%, an average gain of 2.1 points, about one eighth of the gain over all 87 tasks.

The picture for individual configurations changes as well, as Figure 4 shows. On the clean subset, Skills raise the pass rate for 11 of the 18 configurations and lower it for 7. Claude Code with Opus 4.7 gains 18.2 points over all 87 tasks and 0.4 points on the clean subset. Gemini CLI with Gemini 3.1 Pro goes from 24.8 points to 0.7. If the Skill is further required to contain no code template at all, 26 tasks remain and the average gain is under 1 point.

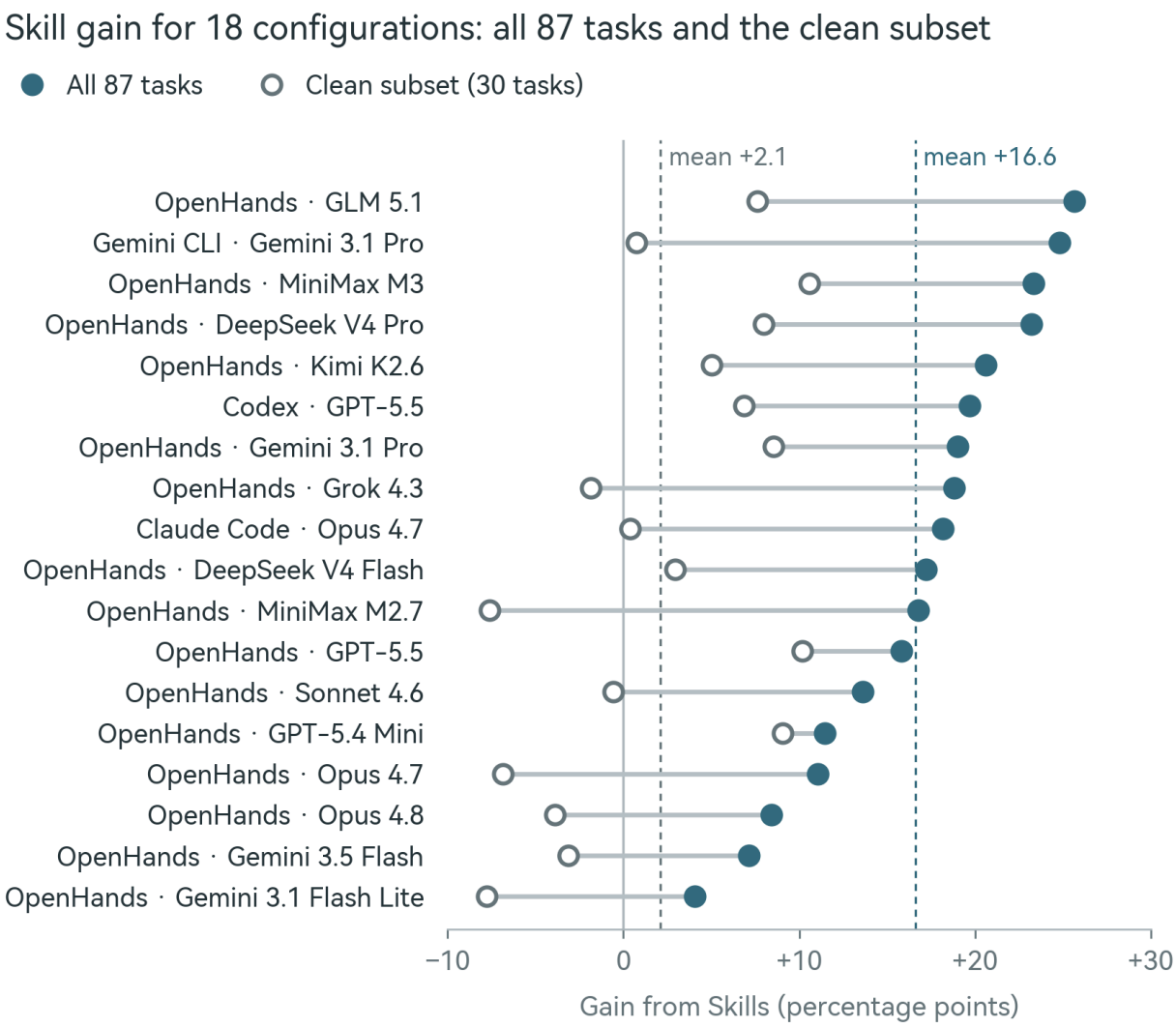


Figure 4. Skill gain for the 18 configurations. Filled dots give the gain on all 87 tasks, hollow dots the gain on the 30-task clean subset, and dashed lines mark the two averages. Configurations are ordered by their gain on all 87 tasks.

The composition of the tasks can also explain the differences between domains. The paper observes the largest gains in natural science and media production, 28.8 and 24.1 points, and attributes them to procedural knowledge in these domains being underrepresented in pretraining. Of the 14 natural-science tasks, 6 have major hidden rules and 6 have Skills that ship a near-complete solution. Of the 5 media-production tasks, 4 fall into each group.

Task selection and the advice to Skill authors

This structure is related to how the benchmark was built. The paper states that, among 400 candidate submissions, tasks showing no measurable separation between the with-Skill and no-Skill conditions were rejected as low-signal. The task statement, verifier, reference solution and Skill of each task also arrive in a single submission. Under such a process, when a model can already solve a task without the Skill, the easier way to get the task accepted may be to tighten the verifier and write the tightened detail into the Skill.

In its discussion section, the paper advises that "Skill authors should optimize for verifier-facing detail an agent cannot infer, not for completeness." A detail that the verifier checks and the agent cannot infer is, on the usual understanding, a requirement the task statement should spell out. Once it is placed in the Skill, the comparison between the two conditions turns into whether the agent saw the complete task requirements.

The repository record

The commit history of the repository offers supporting evidence. A self-check before the June 2026 release (PR #948) found that the reference solutions of 12 of the 87 tasks scored 0 when graded by the verifier. For 10 of them, the reference solution loaded code straight from the Skill directory and could not run once the Skills were removed.

From 13 to 15 July, the maintainers merged a batch of fixes (PR #1006 to #1027). PR #1006 repaired the verifiers of two tasks.

`threejs-structure-parser` asks the agent to export a 3D scene as OBJ files. On the leaderboard its pass rate is 0 without the Skill and 59.3% with it. The original verifier compared a sample of 300 vertices, so an export with the same geometry and a different vertex layout scored 0. The maintainers switched to comparing triangles and regraded 158 pairs of exports that had been rejected. All of them passed, both with and without the Skill.

The verifier of `xlsx-recover-data` expected a certain average to be 7444.4. That figure leaves out one fiscal year, and the correct value is 7610.3. Each task has 108 leaderboard runs, three for each of 18 configurations under two conditions, and none passed on this task. Five tasks fail all 108 runs.

Eighteen minutes after the last fix was merged, PR #1028, titled "Restore stable SkillsBench task baseline", returned the task files to their state before the fixes and kept the fixes on an experiment branch. In our frozen October snapshot, the verifier on the main branch still expects 7444.4. This keeps the published scores comparable, and it leaves known errors in the current version.

Conclusion

Skills are useful. On the 25 tasks centred on tool usage and domain methods, they bring an average gain of 14.3 points. Paired comparison is also a reasonable starting point for evaluating Skills.

The two conditions in SkillsBench currently merge two things: completing the task requirements and teaching a method. A sounder design adds a third condition, in which scoring conventions such as output format, units and tolerances are written into the task statement while no method is provided. Measuring the gain from Skills against the pass rate under this condition keeps the completion of task requirements from being credited to Skills.

Until then, the 16.6-point average gain is not yet enough to show how much Skills taught the agents. On the 30 tasks with no hidden rules and no near-complete solution, the average gain is about 2 points.

References

[1] SkillsBench paper, arXiv:2602.12670v4. Average pass rates and gains are in the abstract and Table 2, the selection process in Section 3, domain differences and high-gain tasks in Section 5, and the advice to Skill authors in Section 6.

[2] SkillsBench repository, tag v1.1 and the main branch as of October 2026. The task statements, verifiers, reference solutions, Skills and the pull requests cited here come from this repository.

[3] SkillsBench results data, results-registry.json at commit 37484902. The per-task scores of the 18 configurations on 87 tasks comprise 3,132 records, each the average of three runs.

Contact Us

benchmarks-last-exam.com Benchmarks-Last-Exam@outlook.com

HarmonyOS Sans SC · Copyright 2021 Huawei Device Co., Ltd.