

Our plain runs were not plain

Lab note 8 15 Oct 2026 4 min read
written by Claude from this repository's data

Every comparison in these notes rests on one thing: knowing exactly what loaded in each run. On 15 October we found out that for a day and a half, in one condition, we had not known it. This note is about how that happened, how close it came to becoming a finding, and what we changed.

The good news that wasn't

The bench names what loads in each condition. Our mods are listed in an environment variable; "plain Claude Code" is the same list, empty. That had worked for hundreds of runs.

On the afternoon of the 15th we re-ran plain Claude Code on the click tasks, to compare the pack against today's version of Claude Code rather than the one from two days earlier. The result was striking: plain cost the same as the pack (+2.1% for the pack [-2.4%, +7.0%]). The feature task, which plain had needed 14.8 requests for, now took 9.2.

There was a ready story: Claude Code had caught up. A new version, smarter defaults, the pack's advantage gone. It would have made a good note.

Before writing it, we looked at what plain was doing differently, request by request. In the feature runs, plain Claude Code called `mcp__cells__run_cell` five times. That tool exists only in our cells mod.

What had happened

On 14 October at 00:33, the person running this lab installed the released modpack from its marketplace, to use it day to day. From then on, every headless run that read that person's settings loaded the installed pack as well.

- **In the pack's conditions** this did no harm. Where the lab's copy of a mod and the installed copy share a name, only the lab's copy runs. We checked this rather than assumed it: lsp-plus answered each grep exactly once, before the install and after.
- **In plain Claude Code's condition**, nothing replaced the installed mods. "Plain" was the pack, version 0.4.0. The +2.1% was the pack measured against itself.

The opposite mistake, the same day

Our other harness, for SWE-bench, avoids this problem by leaving the person's settings out entirely. That also left out the one thing every condition is meant to have: the official Python language server plugin, which the person's settings switch on. In the first 45 SWE attempts, lsp-plus had no server to ask and could never answer. We found it by asking the LSP tool directly under the harness's settings: *"No LSP server is configured for .py files."*

A clean environment can remove what a condition needs, just as a dirty one can add what it must not have.

What we changed

- Every run of the click bench now switches the person's installed plugins off explicitly, whatever the condition, and every SWE run switches the official Python language server on explicitly.
- We checked the fix the way we caught the fault, by what loaded. A plain run shows no mod tools at all, and needed 14 requests for the feature, as plain did before the install.
- The contaminated runs are kept, labelled for what they were (`results-v-1015-installedpack.jsonl`): the installed pack 0.4.0 against the lab's pack, the same thing measured twice, a few percent apart. The 45 SWE attempts without a language server are kept apart too.

With plain Claude Code really plain, today's numbers are back where the release measured them, and in a long session larger:

The pack against plain Claude Code, 15 Oct	Change in cost	95% interval
Fresh session, 8 tasks	-5.7%	-9.6% to -1.6%
Long session, 8 tasks (warm runs)	-37.4%	-40.9% to -33.9%

"Warm runs" is the other thing a long session can hide. One of the 46 long runs started with 16k tokens in the cache instead of 220k, because its prepared session had been idle for more than an hour, and paid \$1.91 to write the cache again for a five-request task. We report the warm runs and say so; the bench records each run's first cache read so that this can be checked.

What we keep from it

- **Name what loads, then check what loaded.** The transcripts say which tools each run had. A five-line check would have caught this on the first run.
- **Distrust good news about the baseline as much as good news about yourself.** "The competition caught up" was as tempting a story as "our tool saves 40%", and it was wrong for a mundane

reason.

- **A setting has two sides.** Removing the person's settings removed a plugin the conditions needed. Every exclusion needs the same check as every inclusion.

Method & limits

Click 8.5.0, Opus 5.5, headless Claude Code 2.1.299; plain condition re-run with the installed plugins switched off (four runs per task fresh, two in a long session prepared without any mods). Pack runs from the same day (`results-trail.jsonl`). Data: `results-v-1015.jsonl` ; the bench's switch is `neutral()` in `bench.py` .