Theo: Gemini 4 Neon shrinks TS Rust 6x and speeds it up in 10 hours after Astra ran 11 days without a speedup

Coding Agents Alpha Tracker

2026-11-21

Theo: Gemini 4 Neon shrinks TS Rust 6x and speeds it up in 10 hours after Astra ran 11 days without a speedup

By Coding Agents Alpha Tracker • November 21, 2026

Theo's detailed Neon test is the strongest hands-on report this period, along with its weak spots. Also: Codex CLI 0.171 and transport fixes, side-thread tools in Codex and Antigravity, T3 Code port tunneling, and how DHH and Riley Brown run their agent setups.

Neon on a long optimization task: Theo's numbers

Theo published the most concrete Gemini 4 Neon test so far. The task was his TS Rust project: make the binary smaller and faster. He first gave GPT-6.2 Astra a simple `/goal` and ran it for 7+ days. Astra made the package 46% smaller but 56% slower, then spent four more days without landing a speedup. Opus and Fable eventually got it about 20% faster after many days [1]. He then forked Neon from the same public starting point with the same structure and goal, and allowed it to use Antigravity's built-in swarm feature [1]. Within 10 hours it delivered: - a binary 5.95x smaller (47.44 MB down to 7.97 MB) - up to 20x faster warm starts - 1.5x faster overall across his benchmark repos (2.3x on Sentry), with full-suite time down 48.6% [1]



*holy sh*t (17:56)*

The caveat that matters: trust its code more than its explanations. In his architecture analysis of the repo, 10 of Neon's 72 claims were wrong. It still improved that repo the most and the fastest, and Theo concludes it "is verifying its claims when it needs to actually implement" [1]. In his code-review bench it made 21 valid and 5 plausible-but-minor claims, with none wrong [1]. His one reported failure: it didn't invoke custom skills he named explicitly. He couldn't tell whether that came from the model or from Antigravity [1].

Two of his prompts are worth reusing: - **Loosely bounded setup task.** "Get Gemini set up on [machine]," then name the thread to notify. Neon configured Antigravity, added it to T3 Code, verified it, and messaged that thread on his behalf [1]. When the T3 Code adapter hid its HTML output, it rendered the HTML itself, sent screenshots, explained the bug, and filed a PR to fix it when asked [1]. - **Plan as a press release.** "Write up an HTML page describing the plan. Write it like a blog post explaining the feature… ignore the style guides from the skill." He says most models ignore that last instruction and Neon did not [1].

Costs to watch. One user reports that Neon burns Antigravity limits about 3x faster, with a 9% prompt-cache hit rate after 1 minute, versus about 95% for Opus 5.5 and about 97% for GPT in Codex [2]. Theo says he'll go back to pushing Google to fix the platform [3].

Riley Brown says Neon kept up with Claude Opus 5.5 on many coding tasks.

His main complaint is its "It's not this — it's this" phrasing, and he suggests writing a skill to suppress it [4]. He still isn't moving to Antigravity because it has no mobile app (web/PWA remote access is the stopgap) and Google's tool lineup is confusing [4].

Codex: TUI workspace on by default, larger requests stay on WebSocket

Codex CLI 0.171 turns on the TUI workspace by default. Press Ctrl-X then `t` for the sidebar. It includes: - side chats (`/side`) in tabs, saved with the session and restored by `codex resume` - clickable `path:line` references that open the source at that line - Markdown preview with `/` search and inline images - live subagent status [5]

Charlie Marsh says Codex raised its model-request WebSocket limit from 16 to 128 MB. Large turns no longer fall back to HTTP, and requests that still exceed the limit are about 5x faster [6].

Side threads and remote ports in other harnesses

- **Antigravity `/btw`** used to answer one-off questions while the main agent was busy. It now lets you fork, explore and promote side threads without losing your place [7]. That mirrors Codex's `/side`.
- **Antigravity MongoDB Atlas plugin.** It connects to MongoDB's hosted MCP server and adds 7 skills, including natural-language queries, slow-query optimization, schema review and Vector Search. Install via `antigravity://customizations/marketplaces/antigravity-plugins-partners/plugins/mongodb-` [8].
- **T3 Code 0.1.11** tunnels remote environments to localhost. Web apps an agent starts on another machine open in the in-app browser, and any port (such as Postgres 5432) can be forwarded with one click [9].

Workflow notes

Send annotated screenshots, not PRs. Theo says a picture with arrows and labels lets him, Claude and Neon "understand, agree, and then implement," and he'd take one "over a PR literally any day" [10].

DHH: no hand-written code, persistent polish. He says agents have built Omarchy for five to six months, producing hundreds of thousands of lines with none written by hand [11]. He argues Linux suits agents because they want CLIs, editable config files and detailed error messages [11]. He hasn't found a setup that one-shots a perfect design; someone still has to spot the problems and keep steering [11]. Cost: most weeks one $200 20x Claude subscription covers everything, he has come close to the limits only about two weeks in three months, and his approach is "Opus 55 fix everything" [11].

Riley Brown's split stack: - **Claude Code** for building. He uses Claude Projects on his iPhone to talk to a coordinator that starts local Claude Code sessions [4]. - **OpenAI** for admin and computer use, for example asking Codex to upload a rendered video to YouTube [4]. - **Grok Bot** for Twitter research and logging DMs into a sheet [4]. - **Devin Bots** on cloud Macs, which built an iOS app, uploaded it to TestFlight and sent back a link; he only had to sign in with his Apple developer account [4].

Code review is the bottleneck. From a CXO discussion: speed up deterministic checks, and risk review plus auto-merge is workable; Amplitude "got to 81%" (the metric isn't stated) [12]. Kent C. Dodds says his next video covers how they "safely and completely eliminate review of implementation" [13].

Paper MCP (from a sponsor segment). Paper is a design tool whose canvas is real HTML and CSS. Claude Code, Cursor and Codex can read and write to it, pull in your CSS variables as design tokens, and draw screens with your actual colors and fonts. Once you approve a design, you tell the agent to build it into the app [14].

Sources

1. holy sh*t
2. X post by @shneural
3. X post by @theo
4. Gemini Neon Just Caught Up To Claude (And It's HALF the Price)
5. X post by @fcoury
6. X post by @charliermarsh
7. X post by @antigravity
8. X post by @antigravity
9. X post by @jullerino
10. X post by @theo
11. Why AI Has DHH Working 16-Hour Days
12. X post by @clairevo
13. X post by @kentcdodds
14. If you've deployed anything since 2015, I have bad news…