Claude Code's experimental Code Mode, and the real cost of Gemini 4 reasoning effort

Coding Agents Alpha Tracker

2026-11-22

Claude Code's experimental Code Mode, and the real cost of Gemini 4 reasoning effort

By Coding Agents Alpha Tracker • November 22, 2026

Claude Code can now chain tool calls through generated JavaScript. Meanwhile Gemini 4 users are settling which variant and effort level to use, Antigravity's harness is claiming coding-benchmark gains, and practitioners shared a small-eval method for switching to cheaper models.

Claude Code Code Mode: scripts instead of one tool call at a time

Claude Code has an experimental mode where Claude writes and runs JavaScript to chain tool calls. Normally it calls Bash, Read or Edit one at a time and waits for each result. In Code Mode a script can run commands in parallel, pass arguments between them, and filter outputs before they reach the context window [1]. Addy Osmani gives two cases where this helps: large search results that would otherwise go straight into context, and the same refactor applied across dozens of files [1].

How to try it: - Start a session with `CLAUDE_CODE_EXPERIMENTAL_CODE_MODE=1 claude`, or set the variable in `~/.claude/settings.json` [1]. - In `auto` or `bypassPermissions` mode, scripts such as test suites or builds can run in the background for up to two hours while you keep working. `/tasks` lets you monitor and control them [1]. - Press Ctrl+O to see the JavaScript Claude generated [1]. - Guardrails still apply. Every generated tool call goes through your deny and ask policies, sandbox and PreToolUse hooks. You can attach hooks to the REPL tool to inspect scripts before they run or to redact tokens from output [1].

Suggested first uses are multi-file audits, batch edits and parallel search-and-filter jobs. Send problems via `/feedback` [1].

Gemini 4: which variant, which effort level

Logan Kilpatrick's quick guide: Carbon has "big model smell," Neon is "best all around," and Hydrogen is "quick and efficient" [2]. He says there's no reason to use Argon now, since it was only the first Gemini 4 revision [3].

Effort level makes a big difference in cost. In Simon Willison's pelican test, Hydrogen on `high` used 5,311 output tokens, cost 3.99¢ and took 19s. On `xhigh` it used 31,477 tokens, cost 23.61¢ and took 2m36s, for a noticeably richer image [4]. That's roughly 6x the cost and 8x the time. His comparison grid for Carbon, Neon and Hydrogen at every reasoning level is here [5].

Opinions on Neon still differ. Perplexity's Aravind Srinivas runs Neon on High in Perplexity Computer and has moved some teammates to it [6]. Jeremy Howard says DeepMind described it at launch as only "approaching" the frontier, and that evals are "a *very* imperfect measure of model utility" [7]. Nicholas Moy disagrees and says many people knew how good it was [8]. On the other side, one user says Neon makes mistakes when trying to prove bugs exist, and would pick Sol, Astra or Opus for everything they tried [9].

Antigravity: harness claims and more quota

Google's Varun Mohan says Neon running in the Antigravity CLI is #1 on the AA Coding Agent Index. He also reports a large jump over mini-swe on SWE-sweep, a bug-finding and fixing eval on large codebases, and says the gain is bigger than other frontier models get from their own native harnesses. More harness improvements are due within a week [10]. These are Google's own numbers. Moy says his preferred setup is remote agents over the web, and he "barely open[s] an actual IDE anymore" [11].

Mohan also reset everyone's Gemini quota on Antigravity. He says caching improvements mean a quota now covers 22% more work [12]. That speaks to last brief's complaint that Neon was using up limits too fast because of low cache hit rates.

T3 Code: bring your own subscriptions

Theo says T3 Code has officially supported Antigravity through Google's integration for months, and there is "zero risk" in using Gemini subscriptions with it [13]. DeepMind's Patrick Toulme, who has landed PRs improving T3 Code's Gemini support, says users "will NOT be banned ever" for using Antigravity in T3 Code [14, 15]. According to Theo, 99% of T3 Code usage runs on subscriptions rather than API pricing [16]. A new feature lets you sign in to MCP servers on remote machines without going through the terminal [17].

Use your own small evals to pick cheaper models

On Riley Brown's channel, Suhail Doshi laid out a simple method. Make a folder called `eval` with five or six tasks you do often. Paste the same prompt into each model and judge the output. Doshi argues this can reveal something like a 30x price cut for the same result [18]. On his own design tasks he saw almost no difference between GPT Astra and DeepSeek V4 4.1 Flash, at about one-thirtieth the cost [18]. For something more automated, take the trace of a task you got right and ask a frontier model to turn it into a scored eval you can rerun whenever a new model comes out [18]. He suggests OpenRouter for switching between models quickly [18].

Riley also says skipping the GUI works better. Codex edits video better "fully headless" with ffmpeg than through DaVinci Resolve's MCP [18]. Doshi thinks models have simply seen far more ffmpeg code [18]. Riley says the same holds for making PDFs directly instead of through Canva, and for Blender, where every action maps to Python [18].

Before routing sensitive code through an aggregator, note the OpenRouter report circulating this period. A researcher sent secret links through the service, and someone opened them 40–130 hours later. The cheaper providers AtlasCloud, Novita and StreamLake were apparently forwarding requests to MiniMax [19]. Sebastian Raschka's explanation: when cheaper providers hit capacity, they reroute traffic to the official model provider rather than return an error [20]. Who opened the links has not been explained.

LangChain's "knowledge factory" for agent context

LangChain describes how its internal agents stay supplied with knowledge [21]. It sorts that knowledge into three kinds: skills (how to do the work), domain knowledge (how the business works) and data-access guidance (where current data lives) [21]. Working agents get a tool to flag knowledge that is missing or wrong. A separate curator agent runs on a schedule, reads those flags along with traces and raw sources, and updates the knowledge store. The agents doing the work don't edit it themselves, which LangChain says would fill it with contradictions [21]. The advice is to start with one recurring workflow and close that loop rather than documenting everything first [21].

Smaller notes

- **Codex worktrees:** swyx found 20GB of duplicated `node_modules` across worktrees [22]. He says that for about a month Codex hasn't been able to clean up its own worktrees, and he has given up [23].
- **Codex desktop:** the latest ChatGPT/Codex app update lists "more reliable Codex Remote" [24].
- **Kent C. Dodds** uses Jev in Kody to re-check vector-search results. When Jev judges a hit very likely to be exactly what the agent wants, Kody

expands its details [25].

- **Game dev:** DesignArena ranks GPT-6.2 Astra (high) #1 and Sol (high) #3 in its Game Dev arena. Astra averages $0.98 and 6.8 minutes per generation, Sol $0.22 and 4.6 minutes [26].
- **Multi-bot setup (from a sponsor segment):** Matthew Berman says Lauren Tan runs a team of Grok-powered coding bots that coordinate in Slack and hand work off to Cursor, reaching 2,500 PRs in a month. Each bot has its own cloud machine [27].

Sources

1. X post by @addyosmani
2. X post by @OfficialLoganK
3. X post by @OfficialLoganK
4. X post by @simonw
5. X post by @simonw
6. X post by @AravSrinivas
7. X post by @jeremyphoward
8. X post by @thenickmoy
9. X post by @meowmeowmrao
10. X post by @_mohansolo
11. X post by @thenickmoy
12. X post by @_mohansolo
13. X post by @theo
14. X post by @PatrickToulme
15. X post by @theo
16. X post by @theo
17. X post by @TheAnirudh
18. Is Claude Still Worth Paying 30x For? | Billion-Dollar Founder Suhail Doshi
19. X post by @michael_chomsky
20. X post by @rasbt
21. How to build a knowledge factory
22. X post by @swyx
23. X post by @swyx
24. X post by @ajambrosino
25. X post by @kentcdodds
26. X post by @DesignArena
27. I'm Crashing Out About Anthropic