Gemini 4 Carbon tops CursorBench and arrives with Antigravity /swarm, as Theo explains when to use Fable 5.2

Coding Agents Alpha Tracker

2026-11-11

Gemini 4 Carbon tops CursorBench and arrives with Antigravity /swarm, as Theo explains when to use Fable 5.2

By Coding Agents Alpha Tracker • November 11, 2026

Google's Gemini 4 Carbon is now generally available, with a multi-agent /swarm command in Antigravity. Theo's Fable 5.2 deep dive adds practical guidance on models and effort levels, and Geoffrey Huntley argues teams should invest in engineering basics rather than skill packs.

Gemini 4 Carbon: new top score on CursorBench, a cheap launch price, and /swarm

Google made Gemini 4 Carbon available to everyone after a two-week release limited to "a subset of cyber defenders" [1]. Cursor says that at high effort it is the top model on CursorBench at **60.4%** [2]. Fable 5.2 scored 60% at max effort last period.

The price doubles on December 1. Until November 30 it costs $2/M input and $10/M output, then $4/$20. Cursor's comparison of effort levels is at cursor.com/evals [3]. Simon Willison notes the launch price matches Sonnet 5.5, and the December price matches Opus 5.5 [1]. If you want to test it on real work, do it this month. llm-gemini 0.36 adds support [1].

Antigravity /swarm. Carbon is in Antigravity's desktop app, CLI and IDE extensions for AI Ultra subscribers and Gemini for business. Google says it is built for "extended verification loops" [4]. The new `/swarm` command turns one prompt into a team of agents working in parallel. They plan, build and verify on a shared live message board, and each role can use a different model [5]. Sundar

Pichai's demo shows the pattern worth copying. Builder and critic subagents worked in loops across Blender and Three.js, checking their own renders, until they had a playable 3D game [6].

Early opinions differ. Willison ran his Yonkers-mech Three.js prompt through Carbon in Antigravity. He had to ask separately for a "standalone HTML file" to get a version without external dependencies [1]. Theo had early access and "wasn't super impressed." After more tests he said he had missed some things, and his video "will not be going how I initially planned" [7].

Theo on Fable 5.2: keep Opus as your default and avoid Max

Theo's Fable 5.2 video ("Nobody Is Competing With Anthropic") backs up last period's advice and adds detail:

- **Opus 5.5 stays the daily driver.** Anthropic recommends starting most coding work on Opus 5.5 and moving up to Fable as needed. Theo says he agrees "beyond hard" on that point [8]. When he audited his Lakebed project, he realised afterwards that the strong feedback had come from Opus 5.5 high, because he forgot to switch models [8].
- **Don't use Max effort for coding.** He says Fable 5.2 Max scores worse on most coding tasks and costs more. Consider it only if lower efforts keep stopping too early and you are comparing results against them [8].
- **It's mixed on his own codebase benchmarks.** On tasks like analysing T3 Code and reviewing PRs, he found Fable below Opus and Sonnet in almost every test [8].
- **Keep the design skill on.** Without it, frontend output falls back to "Tailwind template slop" [8]. When Fable kept adding unneeded eyebrow text and subheadings, he asked it what it calls them and why it adds them. He then changed his agent.md file, and they have appeared less often since [8].
- **Give it sources so it can check itself.** He told Fable to read his history, and it disproved a bad MCP recommendation. It confirmed his own view that device flow is only needed for CLI and remote agents [8].
- **Check its work.** A friend summed it up: "much more capable, but it does not have much more autonomy." Theo says it can still make basic mistakes, such as leaking the very info a "safe mode" view was built to hide [8].

Fireship describes a similar visual self-correction loop. In Claude Code, Fable spent 43 minutes building a Three.js horse. It took a screenshot of its first attempt, listed the flaws and fixed each one [9].

Geoffrey Huntley: invest in engineering, not skill packs

At every major model release, Huntley says to read the model card and "consider nuking skills/agentsmd" [10]. He thinks building agent "factories" now is premature, because whatever fits today's models is overtaken by next week's. His alternatives: - Use Buck2 or Nix. - Speed up builds with content-aware caching. - Prompt for deterministic outcomes. - Make prod, dev and CI identical. - Improve security incident response.

"If you invest in skill packs and agent astrology you go backwards when new model drops" [11]. If you do build a factory, he says to: - capture all agent traces - measure time to merge, token spend per merge and ideation to merge - track incidents and human productivity

His caveat: "lights off isn't a thing yet." Skilled humans are still needed to steer agents in planning mode [12].

Tools and smaller releases

- **Cursor Team Projects (beta):** teammates and their agents share context. Agents share plans before they start, flag conflicts while working, and unblock each other's tasks (blog) [13].
- **Meta Muse Ember 1.0:** a small model for repetitive agent work such as tool calls, coding subtasks, routing and long-document parsing. It costs \$0.10/\$0.50 per M tokens for prompts up to 200k, and \$0.30/\$1.50 above that [14]. Willison argues cost per task matters more than token price. On Artificial Analysis, Ember xhigh scores about 34 at about \$0.12 per task, against about \$0.25 for Haiku 5.5 xhigh [14]. In his SVG test, the "high" reasoning level failed to return a complete SVG, and higher levels did not reliably look better [14].
- **Agents prefer tools with official MCPs:** Resend gives agents more than 80 named MCP tools, for example to set up accounts, check pricing in Markdown and send emails [15]. CEO Zeno Rocha says he lets computer use in Codex or Claude Code manage and debug his DNS records [16]. He dropped Plausible for PostHog because PostHog's MCP is maintained by PostHog itself [16].
- **Claude usage reset:** Pro, Max, Ultra and Team users get one free full reset across all models and limits. Anthropic says it is making up for bugs that used more of users' limits than intended. Claim it before December 10 at 2 PM PT in Settings > Usage [17].
- **Claude Code bug:** Theo saw the input box double every key press [18]. Another user reports random double letters in Claude Code desktop going back more than a week [19].
- **Risk in automated PR merging:** at T3 Code, an agent loop merging contributor PRs attached a stray "fable 5.5" commit message to a maintainer's username while he was asleep. It was one of more than 3,599 PRs he merged that month [20]. If you run merge bots, check commit

metadata as well as the diffs.

Sources

1. llm-gemini 0.36
2. X post by @cursor_ai
3. X post by @cursor_ai
4. X post by @antigravity
5. X post by @antigravity
6. X post by @sundarpichai
7. X post by @theo
8. Nobody Is Competing With Anthropic
9. Did Anthropic just save its $1 trillion IPO?
10. X post by @GeoffreyHuntley
11. X post by @GeoffreyHuntley
12. X post by @GeoffreyHuntley
13. X post by @cursor_ai
14. Introducing Our Smallest, Most Efficient Model: Muse Ember
15. X post by @latentspacepod
16. Designing for Agents — Zeno Rocha, Resend
17. X post by @ClaudeDevs
18. X post by @theo
19. X post by @jamonholmgren
20. X post by @theo