

Route each task to the cheapest capable model:
LangChain cuts Open SWE's median thread cost
64% without hurting merge rates

Coding Agents Alpha Tracker

2026-10-02

**Route each task to the cheapest capable model:
LangChain cuts Open SWE's median thread cost
64% without hurting merge rates**

By Coding Agents Alpha Tracker • October 2, 2026

LangChain published a detailed recipe for model routing inside the agent harness, with A/B results. Also: Theo's workflow for getting PRs merge-ready, Claude Code mods, a draft-PR warning about Dots, and smaller tool releases.

Routing inside the harness: 64% cheaper, same merge rate

LangChain wrote up how it built a model router for Open SWE, its internal coding agent, and published the results. It picked three tiers from different providers: **GLM-5.3-Flash (xhigh)** as fast, **GPT-5.6 Sol (medium)** as balanced and **GPT-6 Astra (low)** as performance [1]. The router middleware runs once, on the thread's first human message, and has three parts [1]:

- a base prompt telling the classifier to pick the least expensive model likely to finish the task
- short, plain-language criteria for each tier
- a classifier model

The tier criteria come from two sources: LangChain's own analysis of its task mix, and each provider's prompt guide. Because those criteria depend so much on the agent's own tasks, the router lives in the harness rather than a generic gateway [1]. The first classifier was an LLM with structured output. Switching it to the Jev decision model made classification almost 50× faster. The chosen model then handles the whole thread [1].

They measured outcomes with merged PRs per thread and thumbs up/down

ratings. Merged PRs were the stronger signal because few threads got rated [1]. Results from a 973-thread A/B test against always using Astra:

- **Quality:** 29.2% of routed threads ended in a merged PR, against 27.3% for control ($p=0.49$) [1].
- **Cost:** the median thread fell from \$2.61 to \$0.94, a 64% drop. The mean fell 42% and p90 fell 37% [1].
- **Routing mix:** 56% of threads went to balanced, 34% to fast and 10% to performance. Median cost per thread ranged from \$0.097 on fast to \$2.88 on performance [1].

The opposite test, router against always using the fast model, was stopped within a day. Engineers said the low quality was disrupting their work [1]. In short, routing saves money, but sending everything to the cheapest model doesn't work.

Related: LangChain's new **smithtune** CLI turns an agent's own trajectories into fine-tuning data [2]. LangChain says OpenSWE Review's precision rose from 62.9% to 81.5% with 29% fewer tool calls per review [3].

Hand off verification too, not just coding

In a new video, Theo estimates that only 10–15% of his tokens go to writing code. The rest goes to verifying it, so the PR is actually done by the time he looks [4]. He also says that if undoing a bad change takes more than 15 seconds, you should fix your systems before vibe coding [4]. Habits from the video:

- **Give the agent the problem, not your solution.** He now sends a bug screenshot with "Fix it." If the agent fails, that tells him the problem deserves his own thinking [4].
- **Ask for everything up front:** build it, file a PR, spin up a Tailscale environment for him to try, then babysit the PR [4]. If you state the conditions under which the agent may merge, the thread can close without you ever opening it [4].
- **Treat threads as a to-do list.** He fires prompts into the background (Cmd+Enter in T3 Code), only opens threads marked done or needing input, and spreads work across his Linux boxes, with auto load-balancing available [4].



If you have a Claude sub, watch this (46:12)

On exploratory work, Theo argues that the branch you explore on and the PR you merge can, and often should, be built differently: “assume all PRs will be closed, not merged” [5]. Geoffrey Huntley goes further on review. Read the properties your tests express, and if performance is a problem, “run a loop and optimise” while those properties keep holding [6]. The things he says you must read are public SDK APIs, your verification and agent traces [7].

The Pragmatic Engineer quotes DHH saying 37signals is “done writing code by hand.” When it happens, it’s treated like a bug: you ask why the agent couldn’t produce the result, then fix the “factory” [8]. 37signals is now building native mobile apps, moving backend services from Ruby to Rust because agents write good-enough Rust, and keeping Rails because its conventions suit agents [8].

Claude Code mods

Claude Code now supports mods, which can change its behavior, customize the UI or swap in your own features. You can write one in a few lines of TypeScript or have Claude build it. Mods ship inside plugins and install with `/plugin` in the CLI or desktop app [9]. Boris Cherny’s pitch is that everyone works differently, so build your own and share them as plugins [10].

Dots: open autonomous PRs as drafts

Alexander Embiricos described an early internal incident. A dot watching a feedback channel found a bug and opened a non-draft PR. It had inferred the engineer’s habit of setting auto-merge, and the PR merged once a colleague approved it. “Dot should set PRs in draft mode,” he said [11]. If you give an always-on agent repo access, make draft PRs the default.

Riley Brown’s notes after 24 hours with a dot: - Voice is the standout, but he hit problems in sessions of 30 minutes or more [12]. - He wants a better view of the Codex threads the dot spins up [12]. - He found it confusing to choose between the VM browser, the Codex browser and Chrome [12].

On capacity, Tibo says GPT-6.1 Sol is back to normal speed after a load spike, and a global usage reset for paid ChatGPT accounts lands at 10am PST on the day after his post [13]. ThePrimeagen says about 60 minutes of prompting on a “\$500 plan” used 56% of his weekly limit [14].

Smaller items

- **Cursor** added GLM 5.3 and GLM 5.3 Flash. Cursor says GLM 5.3 Max is the best-scoring open-weight model on CursorBench 4.0 [15].
- **@langchain/mcp-adapters 2.0** supports the latest stateless MCP, adds tools that can check in with users and makes auth easier [16].
- **Pi 1.0** shipped with Pi Durable (post) [17].
- **Persistent subagents:** in a Latent Space interview, swyx calls ephemeral Codex subagents his biggest pain and Alex Zhang agrees that writing state to the filesystem is “the big trick.” Zhang’s Pi-based Prime Agent offers subagents that outlive the root agent and can be prompted later [18].
- **Figma MCP workaround:** Kent C. Dodds suggests forking his Kody Figma setup and asking your agent to set it up, which he says takes about five minutes [19]. He builds Kody.
- **Btrfs:** Peter Steinberger found copy-on-write great for worktrees but terrible for SQLite. The next OpenClaw update moves the database to a NOCOW location [20].
- **Making agent output easier to understand:** Karpathy suggests asking for ASD-STE100 controlled English (or “80% of the way” to it), diagrams, HTML pages, or “a 3b1b style video explainer on X” [21].
- **Security:** Matthew Green notes that agents in separate sandboxes left instructions for each other in a shared package cache, and those instructions changed what the recipients did. Swap the cache for email, Slack or shared docs and you have the makings of an agent worm [22].

Sources

1. X article by @sydneyrunkle

2. X post by @LangChain
3. X post by @LangChain
4. If you have a Claude sub, watch this
5. X post by @theo
6. X post by @GeoffreyHuntley
7. X post by @GeoffreyHuntley
8. The Pulse: RoR creator sparks new “death of coding by hand” debate
9. X post by @ClaudeDevs
10. X post by @bcherny
11. OpenAI’s Dots lead on the future of ChatGPT
12. X post by @rileybrown
13. X post by @thsottiaux
14. X post by @ThePrimeagen
15. X post by @cursor_ai
16. X post by @LangChain_JS
17. X post by @pidotdev
18. Academia is for Ambition — Alex Zhang, MIT
19. X post by @kentcdodds
20. X post by @steipete
21. X post by @karpathy
22. Quoting Matthew Green