Benchmarked Agent Loops Hit Production: Liquid +53%, CursorBench, and Better Workspaces

Coding Agents Alpha Tracker

2026-03-13

Benchmarked Agent Loops Hit Production: Liquid +53%, CursorBench, and Better Workspaces

By Coding Agents Alpha Tracker • March 13, 2026

Tobias Lütke's Liquid PR turned agentic optimization into a concrete playbook: benchmark script, strong tests, lots of experiments, measurable win. The rest of the day's signal reinforced the same theme with CursorBench, OpenAI Automations, better subagent patterns, and new workspace designs for parallel coding.

TOP SIGNAL

A strong pattern crystallized today: the best agent wins are benchmarked, not vibes-based. Tobias Lütke used Pi plus `pi-autoresearch` to run around 120 semi-autonomous experiments against Liquid, landing 93 commits that made parse+render 53% faster and cut allocations 61% [1]. Simon Willison's reusable lesson is the setup: a benchmark script made `make it faster` actionable, and Liquid's 974-test suite made aggressive agent experimentation safe [1].

TOOLS & MODELS

- **OpenAI Automations — GA.** You can now choose model and reasoning level, run in a worktree or existing branch, and reuse workflows via templates. OpenAI's own examples are recurring repo jobs: daily briefings, issue triage, and PR comment follow-up [2].
- **CursorBench — new eval surface for coding agents.** Cursor is publishing intelligence + efficiency scores for agentic coding, and says it combines offline benchmarks with online evals because public benchmarks

are increasingly saturated [3, 4]. Jediah Katz frames this as a transparency push around real scores [5]. cursor.com/blog/cursorbench [6]

- **Cursor's search stack is now a lot more legible in public.** Via Turbopuffer, Cursor embeds the full codebase with a custom embedding model, uses semantic search plus GREP, and increasingly fans out parallel queries inside an agent turn [7]. Turbopuffer says the migration cut Cursor's costs 95% and fixed per-user economics [7].
- **OpenClaw 2026.3.11 — behavior change worth checking today.** Cron now enforces a stricter cron-owned delivery contract in isolated runs; jobs using `delivery.mode='none'` while sending ad hoc messages may now go silent. Fix: run `openclaw doctor --fix`, then move to explicit `announce` or `webhook` delivery [8].
- **Gemini API spend caps.** Simon Willison calls this immediately useful for CI and agent experiments where the main fear is an accidental bill spike [9, 10].
- **Actual model routing from a daily driver.** Theo says he still prefers Claude for a lot of UI work, uses Codex alongside it inside T3 code/terminal workflows, and will spin up Gemini CLI quickly for UI tasks he cannot do in Codex [11].

WORKFLOWS & TRICKS

- **Run benchmarked autoresearch, not random refactors.**
  1. Create a prompt file plus a script that runs tests and benchmarks [1].
  2. Let the agent propose many micro-optimizations and test them one by one; Tobi's run hit around 120 experiments and 93 commits [1].
  3. Persist state in `autoresearch.jsonl` so the search can keep context across runs [1].
  4. Only do this on a repo with strong tests; Liquid had 974 unit tests [1].

  Having a robust test suite is a massive unlock for working with coding agents. [1]

- **Add a no-code-first planning mode.**
  1. Give users a way to ask for step-by-step architecture/planning without immediate code generation [12].
  2. Anthropic's implementation was basically one instruction: `please don't code` [12].
  3. This matters because users were already trying to force that behavior through the chat UI by hand [12].
- **Use fresh-context subagents, but over-spec the handoff.**
  1. Keep a main agent in the loop and spawn subagents with clean context windows for bugs or research tasks [13, 12].
  2. Let several of them work in parallel when the problem is ambiguous [12].

3. Force the final message to return actual findings, not just `done`; Harrison says bad communication is the failure mode here [13].

- **Rebuild your workspace around projects, not panes.**
  1. Put each project in its own sidebar/workspace entry with quick hotkeys [11].
  2. Inside that project, keep an agent terminal, a dev server, and a git terminal together [11].
  3. Offload long-running agents to SSH/TMUX on another machine so they keep working when the laptop is closed [14].
  4. Theo and Armin/Ben are converging on the same idea: less context switching, more parallel threads under light supervision [11, 14].

PEOPLE TO WATCH

- **Tobias Lütke + Simon Willison** — best public example today of agentic optimization with a measurable end state: benchmark script, test suite, lots of experiments, concrete win [1].
- **Boris Cherny** — high signal because he is sharing actual Claude Code product patterns: `plan mode`, multi-agent `Mama Claude`, and a Bitter Lesson-style refusal to overbuild around current model limits [12].
- **Harrison Chase** — still one of the clearest explainers of agent harness primitives: prompts, planning, subagents, filesystems, sandboxes, observability, and evals [13].
- **Salvatore Sanfilippo** — useful reality check: benchmark passes do not guarantee code you would ship, and operator skill still determines whether AI is a weak assistant or a 10-100x multiplier [15].
- **ThePrimeagen** — strongest contrarian take today: fast autocomplete plus skill may improve proficiency without the cognitive debt and codebase drift that full agents can cause [16].

WATCH & LISTEN

- **8:59-9:53 — Boris Cherny on `plan mode`.** Great short clip because it shows how a valuable agent feature can come from a tiny harness change: users wanted thinking-first, not a code dump, and Anthropic shipped that behavior fast [12].

Claude Code's Creator: How AI Killed Traditional Coding (8:59)

- **13:13-14:33 — Harrison Chase on subagents.** Probably the cleanest explanation you'll hear this week of why subagents help and why communication back to the parent agent is the real hard part [13].

Everything Gets Rebuilt: The New AI Agent Stack | Harrison Chase, LangChain (13:13)

- **13:34-18:23 — Theo on Niri-style work hierarchies for agentic coding.** Worth watching if terminal/IDE/browser context switching is frying your brain; this is a concrete sketch of a better project/task layout for Claude Code, Codex, dev servers, and git [11].

So I stopped using Ghostty… (13:34)

PROJECTS & REPOS

- **pi-autoresearch** — Pi plugin used in Tobi's Liquid optimization. The signal is that it carried state via `autoresearch.jsonl` through around 120 experiments and 93 commits in a live performance PR [1].
- **Shopify/liquid PR #2056** — public playbook for benchmarked agent optimization. Read it for concrete wins like `String#byteindex`, byte-level tag parsing, and cached small-integer strings [1].
- **Seamux** — open-source terminal built on LibGhostty that Theo says has already replaced Ghostty as his daily driver because the project/task hierarchy fits parallel agentic work better than TMUX alone [11].
- **OpenClaw** — if you automate recurring agent jobs, the 2026.3.11 release is the kind of operational change you want to catch early: stricter cron delivery rules plus a maintainer-provided migration path via `doctor --fix` [8, 17].

Editorial take: the edge is moving away from raw model worship and toward measurable objectives, clean context boundaries, and workspaces that let one human supervise many agent threads. [1, 13, 12, 11]

Sources

1. Shopify/liquid: Performance: 53% faster parse+render, 61% fewer allocations
2. X post by @OpenAIDevs
3. X post by @cursor_ai
4. X post by @cursor_ai
5. X post by @jediahkatz
6. X post by @cursor_ai
7. Retrieval After RAG: Hybrid Search, Agents, and Database Design — Simon Hørup Eskildsen of Turbopuffer
8. X post by @chrysb
9. X post by @simonw
10. X post by @OfficialLoganK
11. So I stopped using Ghostty…
12. Claude Code's Creator: How AI Killed Traditional Coding
13. Everything Gets Rebuilt: The New AI Agent Stack | Harrison Chase, LangChain
14. State of Agentic Coding #4 with Armin and Ben
15. Una rivoluzione arrivata in un momento di ignoranza
16. X post by @ThePrimeagen
17. X post by @steipete