Opus 5.5's 'xhigh' Setting Looks Better Than 'max'—So Far

Coding Agents Alpha Tracker

2026-09-25

Opus 5.5's `xhigh` Setting Looks Better Than `max`—So Far

By Coding Agents Alpha Tracker • September 25, 2026

Theo's informal comparison and a separate scientific-research benchmark both favor `xhigh` over `max`; the brief adds practical guardrails for unattended runs, verification, test pruning, and LangSmith's new trace-to-SFT workflow.

TOP SIGNAL

Use Opus 5.5 `xhigh` as the baseline, not `max`. [1] Theo's self-described "silly" Skatebench moved from 78% to 79% accuracy on `max`, while token use grew 15×, cost 13×, and average latency rose from 6s to 50s; the slowest `max` run was 20× slower. [2] A separate 70-task scientific-research benchmark reported 62% on `xhigh` versus 59% on `max`—useful direction, not a coding benchmark. [3]

TRY THIS

- **Make unattended runs bounded.** Put this in `CLAUDE.md`: "When a step doesn't need my input, keep going. Put status notes in the same message as your next action. Stop and ask only when you can't continue without me, or before anything destructive: deleting data, force-pushing, or changing anything outside this repository." Keep destructive-command permission prompts on. [4] For remote work, Theo explicitly authorized environment-variable copying, required checks for the clone, environment, and Claude Code/T3 Code setup, told the agent not to control his active machine, and said to ask if setup failed; he reports the handoff took about 17 minutes. [2]
- **Build executable backpressure.** Huntley recommends tests for architectural rules (e.g. fail when SQL-domain code couples to REST) and

properties checked against generated inputs, rather than only hand-picked cases; his string-reversal example shows how Unicode can expose gaps. Feed reproducible failure reports back to the agent. Huntley says he works at Antithesis, so the deterministic fault-injection offering is a company-affiliated pitch. [5, 6, 7, 8, 9]

- **Separate builders from reviewers.** In his T3-codebase experience, Theo says a different model family can catch issues Claude misses, though it may return more low-value findings. Give the reviewer the diff; ask for only merge blockers, with file/line, why it is wrong, and a reproduction path. Require it to flag what it could not verify, and provide tests or browser access to check claims. [2, 4]
- **Prune tests by proof, not quota.** Steipete reports OpenClaw removed around 400k LOC of tests with little change in coverage; his suggested prompt is to "remove 20% of the least useful tests while maintaining code coverage within 2%." The linked skill is more conservative: do read-only discovery, document what each candidate catches and what stronger proof remains, retain independent contracts and regressions, then validate a coherent owner-boundary batch with owner and sibling tests. Coverage parity alone is not a deletion case. [10, 11, 12]

WHAT SHIPPED

- **LangSmith Trajectories + Fine-Tuning (`smithtune`, public beta).** Trajectories orders human, AI, tool, and system-prompt messages across the main agent and subagents. `smithtune` turns LangSmith traces into supervised fine-tuning data (SFT only), trains via Fireworks or Baseten, and replay-evaluates the tuned model against the base before deployment; preserve each turn's context and tool availability, which can change during long runs. [13, 14, 15, 16] The demo's 50 accepted examples split 41/4/5 across train/validation/test and showed only a slight agreement lift; the presenter said the sample was too small to expect much benefit. It also knowingly uploaded PII for later removal—redact before reproducing. [16]
- **LangSmith Engine v2:** LangChain says it red-teams agents before production, tests proposed fixes before presenting them, and surfaces inefficient agent work plus cost/latency trends. [17]

GO DEEPER

- **Video — Theo, "Getting the most out of Opus 5.5":** The `xhigh/max` segment shows the informal comparison behind the top signal; treat it as a practitioner test, not a broad benchmark. [2]

Getting the most out of Opus 5.5 (12:41)

- **Video — LangChain's SmithTune walkthrough:** Watch how the demo preserves per-turn context, drafts a task rubric, and uses two judges to curate traces; it demonstrates the data workflow, not robust model gains. [16]

How to go from your agent's traces to a fine-tuned model in one workflow

(1:23)

- **Repo — OpenClaw's `test-audit` skill:** A practical counterweight to percentage-based pruning: read-only discovery, evidence for each deletion, and validation at the owner boundary. [10, 12]

Editorial take: Keep reasoning effort adaptive; spend the saved budget on checks that can disprove a patch and give the agent a reproducible fix target. [2, 8]

Sources

1. X post by @theo
2. Getting the most out of Opus 5.5
3. X post by @ArtificialAnlys
4. Getting the most out of Opus 5.5 in Claude and Claude Code
5. X post by @GeoffreyHuntley
6. X post by @GeoffreyHuntley
7. X post by @GeoffreyHuntley
8. X post by @GeoffreyHuntley
9. X post by @GeoffreyHuntley
10. X post by @steipete
11. X post by @steipete
12. openclaw/.agents/skills/test-audit/SKILL.md at main
13. X post by @LangChain
14. X post by @LangChain
15. Introducing LangSmith Fine-Tuning
16. How to go from your agent's traces to a fine-tuned model in one workflow
17. X post by @LangChain