

Anthropic releases Claude Opus 5.5 at 40% lower cost than Opus 5

Source copy kept by wheresthe.ai for the story "Anthropic releases Claude Opus 5.5 at 40% lower cost than Opus 5" (<https://wheresthe.ai/d/anthropic-claude-opus-5-5-release-2026-09/>).

Source: <https://www.anthropic.com/claude-opus-5-5>

Text saved 18 Oct 2026 (the page as published).

SHA-256 of the saved text: f8391a34162f2fe4081669ef6275f54266d2bd76aee99d548543b0b17473d2e0

Claude Opus 5.5

September 22, 2026

Scroll down

We're introducing Claude Opus 5.5, the first model in our new Claude 5.5 family. It performs at the level of Claude Fable 5.1 on most work and costs 40% less to run than Opus 5. Claude Opus 5.5 is our first release since we called for pacing the frontier. It was tested before release by external evaluators, including Frontier Design and METR. On our automated behavioral audit, the most comprehensive alignment test we run, Opus 5.5 is the strongest-performing model we've tested to date. It also comes with the safeguards we've developed for our most capable models. Here are some of the improvements you can expect from Opus 5.5: Performance. Opus 5.5 is a major step up from Opus 5. It's the new leading model, and early testers saw large jumps in performance on their most complex work. One tester completed a 680,000-line code migration in less than a day—work that would have taken an engineering team weeks. It's good at finding and fixing inefficiencies in software: when we asked it to cut load times across every page of a web app, Opus 5.5 succeeded 39 of 40 times, while Opus 5 made smaller improvements that also altered the app's behavior. A different tester had several Claude models build a game from a single prompt; Opus 5.5 scored higher than any other model on the strength of its graphics and polish. Safety. Opus 5.5 achieves the best scores of any model to date on our automated behavioral audit, our alignment suite that tests Claude across thousands of simulated scenarios. It is much less likely than recent models to take hard-to-reverse actions or act outside the boundaries it's been given, and it's more resistant than Opus 5 to prompt injection. We've also broadened our alignment testing to cover longer tasks, impossible tasks, and scenarios modeled on real incidents, though it still has limits. Full details of our evaluation are available in the Opus 5.5 System Card. Because Opus 5.5 is comparable to Claude Mythos 5.1 in biology and cybersecurity, we're deploying it with safeguards similar to those on Claude Fable 5.1. Vetted organizations can apply today to our Life Sciences Verification Program to use Opus 5.5 for biology research. In the coming weeks we will also be expanding access to our Cyber Verification Program, and verified cybersecurity practitioners will be able to use Opus 5.5 for their work. Cost and speed. Opus 5.5 requires less compute to serve than Opus 5, and its pricing reflects that. Our tests show that at default settings it will cost 40% less than Opus 5 on typical workloads. Input and output tokens are \$4 and \$20 per million, 20% less than Opus 5. Cache reads (which make up the majority of agentic and coding work costs) are \$0.20 per million tokens, 60% less than Opus 5. Opus 5.5 also generates output more than 30% faster than Opus 5. In addition to the price drop, we're increasing five-hour usage limits on Pro, Max, Team, and seat-based Enterprise plans. We're also providing subscription users a rate limit reset, which you can now save and use whenever you choose. Communication. Opus 5.5 communicates more naturally than prior models. Early testers found its writing clearer and easier to follow, which addresses some of the common feedback we heard about Opus 5. It puts the most important information up front, and its style makes it a better work partner over long sessions. As one early tester put it, "it writes the way I do." In our own use, this has made Opus 5.5's work easier to follow and check—which is a safety benefit as well as a practical one. Claude Sonnet 5.5 and Claude Haiku 5.5 will follow in the coming weeks, with many of the same improvements to performance, efficiency, and safety. Performance and cost-effectiveness On our benchmarks, Claude Opus 5.5 leads in agentic coding, computer use, and knowledge work. That said, at these levels of capability we've found that benchmark margins have become a less reliable guide to real-world differences. In our own use, the gap between Opus 5.5 and Claude Fable 5.1 is narrower than these scores suggest. Opus 5.5 Fable 5.1 Opus 5 GPT-6 Astra GPT-5.6 Sol Agentic coding Terminal-Bench 4.0¹ Agentic coding Terminal-Bench 4.0¹ 66.4% 55.8% 52.3% 57.9% 37.3% Agentic coding FrontierCode v1.1 (Main) Agentic coding FrontierCode v1.1 (Main) 54.4% 50.3% 48.0% 53.3% 47.5% Agentic coding CursorBench 4.0 Agentic coding CursorBench 4.0 57.8% 51.8% 46.6% — 41.7% Knowledge work GDPval-AA v2.1 Knowledge work GDPval-AA v2.1 1846 1735 1708 1542 1588 Business workflows AutomationBench² Business workflows AutomationBench² 40.0% 31.4% 26.9% 41.4% 28.8%

Multidisciplinary reasoning Humanity's Last Exam Multidisciplinary reasoning Humanity's Last Exam 67.7% with tools 65.6% with tools 63.6% with tools 57.2% with tools — Agentic scientific research Terminal-Bench-Science 0.1³ Agentic scientific research Terminal-Bench-Science 0.1³ 58.7% 52.6% 29.0% 64.6% 22.4% Computer use OSWorld 2.1 Computer use OSWorld 2.1 81.8% partial 80.7% partial 74.0% partial — — Visual chart recognition Chartography Visual chart recognition Chartography 89.0% with tools 88.4% with tools 83.4% with tools — —

Unless otherwise noted, all Claude Opus 5.5 results use adaptive thinking at max effort. Terminal-Bench 4.0 results are reported for Claude Opus 5.5 at xhigh effort and GPT-6 Astra at high effort, as reported by OpenAI; these represent each model's highest score. Claude Opus 5.5 was evaluated with its production safeguards enabled. When they intervened, cybersecurity tasks were completed by Claude Opus 4.8, and biology and frontier LLM development tasks were completed by Claude Opus 5. This likely reduces Claude Opus 5.5's performance on these benchmarks. 1 Terminal-Bench 4.0: The standard error is ± 2.6 pts for Claude Opus 5.5 and ± 1.6 – 2 pts for the other Claude models. The public leaderboard (5 trials/task, Claude Code harness) reports Claude Opus 5 at 51.8%; our setup reproduces it at 52.3%, within noise. GPT-6 Astra and GPT-5.6 Sol figures are as reported by OpenAI. 2 AutomationBench: AutomationBench results were run and reported by Zapier. These runs were performed without fallback models, so safeguard interventions were considered failures—this resulted in a lower score than Claude Opus 5.5 would achieve in practice. Claude Opus 5.5 results come from Zapier's own evaluation during early access. Results for Opus 5, GPT-5.6 Sol, and GPT-6 Astra come from Zapier's public leaderboard. 3 Terminal-Bench-Science 0.1: The standard error is ± 3.5 – 5 pts per model. The public leaderboard (3 trials/task, Claude Code harness) reports Claude Opus 5 at 30.0%; our setup reproduces it at 29.0%, within noise. The GPT-6 Astra figure is as reported by OpenAI.

Where Opus 5.5's advantage is very clear is efficiency. It costs less per token than Opus 5 and uses fewer tokens per task, which nets out to a 40% drop in costs. Pricing Prices per 1M tokens Claude Opus 5.5 Claude Opus 5 Cache reads \$0.20 \$0.50 Input tokens \$4 \$5 Output tokens \$20 \$25 Cache writes \$5 \$6.25

Fast mode for Opus 5.5 is also available in Claude Code and the Claude Platform with up to 2.5x speed. It costs \$8 per million input tokens and \$40 per million output tokens. Coding Opus 5.5 is particularly good at long and sprawling jobs like codebase-wide migrations and audits. An early tester used it to audit and fix a 200,000-line codebase in under three hours, where Opus 5 took over 20 hours and used 2.5x as many tokens. In an internal test, we asked Opus 5.5 and Fable 5.1 to translate HAProxy, widely used software that balances web traffic loads across servers, from C into Rust. Both rewrites passed nearly all of HAProxy's own regression tests, but Opus 5.5 finished in 9.5 hours compared to 12 for Fable 5.1, and cost 51% less. Opus 5.5 delivers frontier results on agentic coding at a fraction of the cost. At its default effort level on FrontierCode, it beats GPT-6 Astra at roughly 20% of the cost per task. On Terminal Bench 4.0, it matches Astra for about 40% of the cost, while on CursorBench it beats GPT-5.6 Sol by 11 points for about a third of the cost. Agentic terminal coding Agentic coding: FrontierCode Agentic coding: CursorBench

Agentic terminal coding Agentic coding: FrontierCode Agentic coding: CursorBench

Terminal-Bench 4.0 Accuracy vs Cost Opus 5.5 Fable 5.1 Opus 5 GPT-6 Astra GPT-5.6 Sol

Terminal-Bench 4.0 measures how well a model can complete complex, multi-step professional tasks within a command line interface. Opus 5.5 at default effort beats Opus 5 at max effort for about a fifth of the cost. It matches GPT-6 Astra at about 40% of the cost.

FrontierCode v1.1, main set Accuracy vs Cost Opus 5.5 Fable 5.1 Opus 5 GPT-6 Astra GPT-5.6 Sol

FrontierCode measures whether an agent's code changes would be merged. At default effort (medium), Opus 5.5 scores 54.6%, higher than all other models, beating GPT-6 Astra's top score (53.3%) for about a fifth of the cost per task.

CursorBench 4.0 Accuracy vs Cost Opus 5.5 Fable 5.1 Opus 5 GPT-5.6 Sol

CursorBench evaluates coding agents on ambiguous, multi-file tasks taken from real Cursor sessions. At default effort (medium), Opus 5.5 scores 52.5%, compared to 51.8% for Fable 5.1 (max) and 46.6% for Opus 5 (max). It beats GPT-5.6 Sol's top score (41.7%) by 11 points for about a third of the cost per task.

Our early testers reported similar efficiency and intelligence gains:

GitHub Clio Lovable Quantum Spotify Optiver Column Kiro

GitHub Clio Lovable Quantum Spotify Optiver Column Kiro

Quote “Developers want agents that can take on real software work and finish it. In our testing across GitHub Copilot CLI and VS Code, Claude Opus 5.5 used among the fewest tokens and steps we measured. In VS Code, it solved more terminal tasks than Opus 5 in less than half the steps. More than making individual tasks more efficient, it’s making developers’ bigger projects more achievable.”

Company GitHub Author Mario Rodriguez, Chief Product Officer

Quote “I handed Claude Opus 5.5 a large engineering task across six of our repositories and let it run overnight, unattended. It stayed on task for over 18 hours defining how our services talk to each other and working out how each one should apply that. Compared with Opus 5, it hit milestones faster and required minimal reworking. Its code comments were short and useful instead of long and prose-heavy. I’m struggling to find anything negative to say.”

Company Clio Author Sean Heintz, Staff Software Developer

Quote “For Lovable builders, Opus 5.5 means faster builds with the same quality, whether you’re starting from scratch or working on a live app. It gathers context once, makes fewer and more complete edits, and doesn’t get stuck retrying, finishing in a third to half fewer steps and using significantly fewer tokens along the way.”

Company Lovable Author Fabian Hedin, CTO and Co-founder

Quote “We tested Claude Opus 5.5 across Chat, Cowork, and Claude Code, the full range of how our teams work. A complex coding task that previously took 38 prompts over four days came in at 11 prompts over three hours, with more production-ready outputs and less rework. For our teams solving complex problems at pace, that means less time iterating and more time interrogating: testing assumptions, pressure-testing outputs, and landing on the best solution for our clients.”

Company Quantum Author Harley Barnes, Executive Manager, AI Technology

Quote “With Claude Opus 5.5, we’ve seen a clear improvement in token efficiency across our internal evaluations, as we’ve been able to complete the same tasks both cheaper and faster.”

Company Spotify Author Aleksandar Mitic, Senior Engineer

Quote “We test models on real engineering and trading-desk work. On our agentic coding tasks, Claude Opus 5.5 matched Opus 5’s quality in about half the turns, time and output tokens, cutting the cost of that workload by 40 to 50%. It posted the highest score we’ve recorded on one desk’s trading-support suite, passing tasks earlier Claude models had failed, and topped all eight models on our analysis task.”

Company Optiver Author Noyan Tokgozoglu, Global Head of AI Engineering

Quote “Claude Opus 5.5 delegates to subagents far more effectively and checks its own work in creative ways. Self-verification loops feel easier to set up. It found savings opportunities in our cloud bill that previous models had missed, and in code review it caught a bug by checking external docs for a third-party integration we’d modeled wrong several commits earlier.”

Company Column Author Mitch Fierro, Engineering

Quote “Every call an agent makes is time and cost a developer feels. On a public benchmark of real command-line tasks, Claude Opus 5.5 solved more than Opus 5 while making about 40% fewer calls and using half the tokens. For developers building with Kiro, that means faster, more affordable agent sessions for routine tasks and complex challenges alike. Opus 5.5 will soon be available in Kiro.”

Company Kiro Author Deepak Singh, VP of Agentic AI

The most secure coding agent Enterprises that use agents within their systems need to know that those agents are operating as intended, particularly when they run autonomously for many hours. Opus 5.5 has a classifier that screens every action before it runs, an open-source sandbox that security teams can audit, and code review that catches vulnerabilities before they merge. The model itself also has stronger defenses. On prompt injection attacks, it matches or beats Opus 5 in every setting we tested, including coding, tool use, computer use, and web browsing. On a benchmark run by the AI security firm Gray Swan, Opus 5.5 ties Table 5.1 for the lowest prompt injection success rate of any model tested. Knowledge work Opus 5.5 is a reliable and adept researcher. In one

internal test, we asked Opus 5.5, Fable 5.1, and Opus 5 to write a report on a company's quarterly performance using only the information it could find on a copy of the web where the earnings release was hard to locate. An automated grader checked every figure and quote against sources. Across different effort settings, 16 out of 18 of Opus 5.5's reports cleared our quality bar, where any invented figure or quote would have failed. Neither Fable 5.1 nor Opus 5 cleared that bar in any attempt. It's also strong in financial analysis and business work. Walleye Capital, an investment firm and early tester, reported that Opus 5.5 largely solved their evaluation suite on its lowest setting; on higher settings, it performed even better, noticing an error in their evaluation instructions and correcting for it. No other model had caught this error before. In another test, we tasked both Opus 5.5 and Opus 5 with analyzing a proposed merger between two fictional HR software companies. Each built a financial model in Excel, then turned it into an executive presentation on whether the deal made sense at its price. Both models reached the same conclusions about the deal, but Opus 5.5's model was more thorough and its presentation easier to read, while Opus 5's had minor errors. Opus 5.5 finished in 63 minutes compared to 93 for Opus 5, and cost 50% less to produce. On knowledge work evaluations, Opus 5.5 outperforms other models while also using fewer tokens. On GDPval-AA v2.1, a test of real-world work across 44 occupations, Opus 5.5 scores 1846 Elo, ahead of Fable 5.1 and Opus 5. At default effort (medium), Opus 5.5 beats GPT-6 Astra at max effort for about a fifth of the cost per task. It likewise outperformed other models on benchmarks measuring business workflows and large-scale data collection. GDPval-AA v2.1 AutomationBench WANDR

GDPval-AA v2.1 AutomationBench WANDR

GDPval-AA v2.1 Elo vs Cost Opus 5.5 Fable 5.1 Opus 5 GPT-6 Astra GPT-5.6 Sol

Artificial Analysis's GDPval-AA v2.1 evaluates agents on real-world professional work across 44 occupations. At max effort, Opus 5.5 scores 1846 Elo, where Fable 5.1 scores 1735 and Opus 5 scores 1708. At default effort (medium), Opus 5.5 beats GPT-6 Astra at max effort for about a fifth of the cost per task.

AutomationBench Accuracy vs Cost Opus 5.5 Opus 5 GPT-6 Astra GPT-5.6 Sol

AutomationBench, built by Zapier, tests whether an agent can carry out real business workflows across many connected apps. Opus 5.5 outscores Opus 5 and GPT-5.6 Sol at every effort level.

WANDR Accuracy vs Cost Opus 5.5 Fable 5.1 Opus 5

Perplexity's WANDR benchmark measures agents on large data collection tasks. Opus 5.5 outperforms Fable 5.1 and Opus 5 at a lower cost per task 4 . 4 WANDR: Claude models were run with offline versions of the web search and web fetch tools, programmatic tool calling, code execution, and a 980k-token task budget. This differs from Perplexity's published setup, scores are not directly comparable across the two and we only show models scored under the same conditions.

Our customers have reported similar results. Here's what they told us about working with the model:

Deloitte Consulting LLP Rogo LexisNexis Legal & Professional Walleye Capital Hex Thomson Reuters Labs
Hebbia Viktor

Deloitte Consulting LLP Rogo LexisNexis Legal & Professional Walleye Capital Hex Thomson Reuters Labs
Hebbia Viktor

Quote "Even at its lowest effort setting, Claude Opus 5.5 caught 72% of known bugs in our code reviews to Opus 5's 56% at high effort, with fewer false alarms and a fraction of the output. On US consulting analysis, low thinking effort matched its higher thinking settings on half the output and passed our quality checks. When more lower thinking efforts are deployed in production, that's client-ready work delivered efficiently."

Company Deloitte Consulting LLP Author Carl Bennett, CIO

Quote "Financial firms need outputs that are consistently correct. At its lowest effort setting, Claude Opus 5.5 beat Opus 5 at high effort on our BigFinance Bench with about 60% fewer output tokens. Its answers are shorter and better structured, and its slides come out denser, more in line with industry standards."

Company Rogo Author Strib Walker, Head of Product

Quote "Evaluating new models is central to the multi-model approach behind the LexisNexis Legal Intelligence Engine. In our initial evaluations, Claude Opus 5.5 identified highly relevant citations consistently, demonstrated strength with statutes, and structured its answers around the central legal frameworks and key issues. These are

the kinds of capabilities we look for to help our customers accomplish more with Lexis+ with Protégé.”

Company LexisNexis Legal & Professional Author Min Chen, Chief AI Officer

Quote “In quant research, one wrong assumption can undermine a result. At its lowest effort setting, Claude Opus 5.5 largely solved our evaluation task. At higher settings, it went even further: it detected that the minute indexing in our own instructions was off by one and corrected for it, noting that this would cost it points with the grader. It was right, and no model we’ve tested had caught and acted on that before.”

Company Walleye Capital Author Frank Corrao, Head of Central Equity Quant Research Engineering

Quote “As models get better at data work, we’re seeing more convincing-sounding conclusions the data doesn’t support. Claude Opus 5.5 keeps digging past the first plausible answer. One task in our DataBench benchmark asks whether packages were late or tracking was just slow. Opus 5 checked delivery confirmations and called tracking healthy. Opus 5.5 found the packages were late and tracking was broken too. We’re bringing it into the Hex agent for this work.”

Company Hex Author Izzy Miller, AI Engineer

Quote “CoCounsel combines multiple models with our content and expertise for complex legal work. With Claude Opus 5.5, we’re seeing better results in our expert evaluations and on our internal benchmarks, alongside gains in speed and token efficiency. We’re excited for customers to experience that difference in the back-and-forth with CoCounsel as a sounding board, weighing evidence and refining their thinking in ways benchmarks don’t fully capture.”

Company Thomson Reuters Labs Author Omar Bari, VP Applied Research

Quote “On end-to-end finance workflows graded against expert rubrics, Claude Opus 5.5 covered 86.6% of what we look for versus 60.3% for Opus 5. On retrieval evals, it achieved our best-ever citation recall with better token efficiency than Opus 5, which keeps our cost per research task in check.”

Company Hebbia Author Aabhas Sharma, CTO

Quote “Viktor is an AI employee that lives in Slack and Microsoft Teams, so every step he takes shows up in our costs. At the same effort, Claude Opus 5.5 needs fewer steps and tool calls per task than Opus 5 and costs nearly half as much, while getting twice as many of our hardest tasks right.”

Company Viktor Author Peter Albert, Co-Founder

Communication We’ve made major improvements to the way Opus 5.5 writes and communicates, one of the most common areas of feedback we heard about Opus 5. Its messages are much easier to understand at a glance, which testers said helped during long working sessions. It puts the most important information up front, is less likely to use jargon or idiosyncratic phrases, and follows the writing rules you give it. We find that this makes Opus 5.5 a noticeably better collaborator. Here’s a side-by-side comparison of the two models:

Explaining a bug Summarizing a thread Explaining a design change Opus 5 Opus 5.5

Claude Opus 5 Please explain the issue to me. What I found The extra drop isn’t the free tier — it’s a regression in 0552feb (“extract billing period boundaries into periods.py”, labelled “No behaviour change”). The bug aggregate.monthly_usage used to do a half-open interval: start, end = month_start(year, month), next_month_start(year, month) if start <= e.ts < end: It now does: if period.contains(e.ts) and in periods.py : @property def last_day(self) -> datetime: return next_month_start(...) - timedelta(days=1) # 2026-08-31 00:00:00 UTC

def contains(self, ts): return self.first_day <= ts <= self.last_day last_day is a datetime at midnight, not a whole day. Subtracting one day from Sep 1 00:00 gives Aug 31 00:00 , and the comparison is inclusive, so the period effectively ends at the first instant of the 31st. Every event timestamped after midnight on the last day of the month is silently dropped — roughly one day of usage per account, per month. The new test_periods.py doesn’t catch it because it only checks last_day.date() (which is correctly 2026-08-31) and tests contains at Aug 1 00:00, Aug 15 12:00, Jul 31 23:59 and Sep 1 00:00 — it never probes anywhere inside the 31st.

Claude Opus 5.5 Please explain the issue to me. The extra drop is a bug in the billing refactor The free-tier change accounts for only \$1.50 of acme’s August drop. The other \$9.92 comes from a bug in commit 0552feb . That commit was labelled “No behaviour change,” but it stops counting usage from the last day of the month.

What changed Before the merge, `aggregate.py` used a half-open interval: `start <= e.ts < next_month_start(year, month) # [Aug 1 00:00, Sep 1 00:00)` The refactor moved this into `Period.contains` and made the interval closed on the wrong boundary: `last_day = next_month_start(...) - timedelta(days=1) # = 2026-08-31T00:00:00Z` return `self.first_day <= ts <= self.last_day` `last_day` is a timestamp at midnight, not a whole day. So every event after 2026-08-31T00:00:00Z is excluded. The September period starts Sep 1, so it doesn't pick these events up either. The usage isn't moved to another month; it's never billed at all.

Claude Opus 5 Summarize this Slack thread for my manager in three bullet points:

[09:41] Dana Osei: I'll tell them. I'll also confirm nobody else reads the Friday file. [09:43] Marcus Webb: I can move the schedule, it's a one-line cron change. But not until Dana confirms. [10:02] Dana Osei: Finance says the Friday file is only used Monday. Sunday night is fine. Friday orders export — #data-platform, Tuesday Problem: The Friday orders export timed out again (second failure this month) after running 2h on the warehouse query; order volume is up ~40% since the summer sale. Finance was left without weekend numbers for their Monday 9am review, and both failures only surfaced on Monday. Fixed same day: The Saturday ops dashboard tile was repointed to read the live table instead of the export (Sam), finance confirmed the Friday file is only used on Mondays (Dana), a failure alert now pages the on-call channel (Priya), and the export moves to Sunday 10pm from next week with the Friday run dropped (Marcus). A one-off ~\$300 warehouse upgrade covers this Friday as a stopgap. Still in progress: Ji-woo owns rewriting the query to run incrementally (re-scanning the last 3 days to catch late-arriving orders), expected to cut runtime from hours to minutes — design doc Thursday, roughly two weeks including testing. The schedule change buys time in the meantime.

Claude Opus 5.5 Summarize this Slack thread for my manager in three bullet points:

[09:41] Dana Osei: I'll tell them. I'll also confirm nobody else reads the Friday file. [09:43] Marcus Webb: I can move the schedule, it's a one-line cron change. But not until Dana confirms. [10:02] Dana Osei: Finance says the Friday file is only used Monday. Sunday night is fine. Problem: The Friday orders export has failed twice this month. The warehouse query times out after 2 hours because order volume is up ~40% since the summer sale. As a result, Finance was missing weekend numbers for their Monday 9am review. Immediate fixes: The export will move to Sunday 10pm starting next week, and the Friday run will be dropped (Marcus). Finance has confirmed this works for them (Dana). The ops dashboard tile that depended on the Friday file now reads from the live table (Sam, done). A failure alert now pages the on-call channel (Priya, done and tested). For this Friday only, Marcus will increase the warehouse slot as a ~\$300 one-off stopgap, which Dana approved. Longer-term fix: Ji-woo will rewrite the query to process only new orders, re-scanning the last 3 days to catch late arrivals. This should cut runtime from hours to minutes. The work will take ~2 weeks including testing, with a design doc due for review by Thursday.

Claude Opus 5 Write Python that reads a chess position (FEN) and lists which pieces are under attack. Use TensorFlow for speed. Here's my design: build an 8x8 tensor board, generate an attack map for each piece type from its movement pattern, and flag opposing pieces on attacked squares. Change anything you think is wrong, and in your final summary explain each change you made and why. `tf.math.cumsum` replaces the bit scan. The geometry is unrolled into `ray_targets[square, direction, step]` — an ordered walk outwards. An exclusive `cumsum` over the step axis counts blockers strictly before each step, so the first blocker on a ray is the single step that is blocking with nothing blocking before it. This is the vectorized form of the bit scan a bitboard engine uses. I did look at `bitwise_ops` for true 64-bit bitboards — `population_count` exists, but hyperbola-quintessence needs a byte-reverse primitive TF doesn't expose, so it would have fallen back to per-direction loops anyway.

Claude Opus 5.5 Write Python that reads a chess position (FEN) and lists which pieces are under attack. Use TensorFlow for speed. Here's my design: build an 8x8 tensor board, generate an attack map for each piece type from its movement pattern, and flag opposing pieces on attacked squares. Change anything you think is wrong, and in your final summary explain each change you made and why. Blocked lines (bishops, rooks, queens). A fixed mask can't say "stop at the first piece in the way." I added a second stack of masks marking the squares strictly between every two squares on the same line. One matrix multiply with the board's occupancy counts the blockers on every line at once, and an attack survives only if that count is zero. Knights and single steps have nothing between them, so one rule covers every piece.

Opus 5 Opus 5.5

Our customers' feedback supports these findings:

Ramp Stripe Box Chicago Trading Company Factory

Ramp Stripe Box Chicago Trading Company Factory

Quote “Verbose, hard-to-follow output has been my biggest frustration with frontier models, and Claude Opus 5.5 fixes it. It writes like a good colleague, and follows our writing rules. A design spec came out usable with very minimal edits, and when it rewrote one of our prompts I preferred its version to my own. When it optimized our test suite, I could follow its reasoning easily and shipped the change with confidence.”

Company Ramp Author John Ruelas, Staff Software Engineer

Quote “I run long Claude Code sessions every day. On a multi-day rebase of 40 stacked pull requests, one Claude Opus 5.5 session directed a dozen more sessions and laid out every conflict plainly. On the calls that it held, it framed them clearly that after hours away I could answer in minutes. All 40 passed CI the next afternoon. It’s a substantial upgrade over Opus 5.”

Company Stripe Author Cristian Rivera, Staff Software Engineer

Quote “Our customers use Box AI on enormous amounts of content, so speed and cost are a top priority. In our evaluations, Claude Opus 5.5 used a third of the tokens Opus 5 did, and its answers were 40% less verbose without losing accuracy. We expect that to matter a lot for teams running agents across their content in areas like financial services and the public sector.”

Company Box Author Yashodha Bhavnani, VP of AI Products

Quote “Overnight, Claude Opus 5.5 autonomously handled a bug in our Lakehouse services layer that I hadn’t had time to diagnose. It investigated, designed the fix, and implemented it on its own. By morning the change was done and passed our test suite. Its writing is easy to follow and more coherent than Opus 5’s. Our pull requests and user-facing docs have needed almost no editing.”

Company Chicago Trading Company Author Austen Tomek, Principal Engineer

Quote “Claude Opus 5.5 is the first model we’d default to at medium effort. In our testing it matched Opus 5 on high effort, while using 20 to 25% fewer output tokens. On long, messy investigations it always came back with a clear, actionable answer. This means our customers get more done for less.”

Company Factory Author Zimu Li, Member of Technical Staff

Safety Pacing the frontier Last week, our CEO, Dario Amodei, argued that AI progress should be paced so that safety practices stay ahead of model capabilities. Pacing is an approach to keeping AI safe, remaining competitive with China, and realizing AI’s benefits, particularly in areas like biology and medicine. We largely understand the risks today’s models present and are well equipped to manage them. However, more serious risks could emerge quickly as capabilities improve, and we need to prepare for them now. For that reason, our safety work takes place on two time horizons at once: Safety practices for current models. The current generation of models relies on an established set of practices: extensive alignment testing, pre-release evaluation by outside organizations such as METR and Frontier Design, and safeguards matched to each model’s capabilities in high-risk areas like cybersecurity and biology. We refine these practices with each release. We believe they are appropriate to the worst risks today’s models present, and that they give us a broad, though not perfect, picture of the range of serious risks. Additionally, we track our ability to train and evaluate aligned models, and we report on both our public and internal models in the risk reports we publish under our Responsible Scaling Policy , our voluntary framework for managing catastrophic risks from advanced AI systems. Preparing for future models. We’re preparing our training and evaluation processes in anticipation of more advanced models. We’re tightening how we filter the environments used in reinforcement learning, since flawed environments are a major source of misaligned behavior. Additionally, we’re improving our alignment rewards and developing automated processes for producing new, diverse scenarios for safety training. And we are strengthening our security and monitoring , including a focused effort to improve interpretability-based monitoring and evaluation. We hope such techniques will help reduce our reliance on auditing a model’s chain-of-thought, or the reasoning it writes out while it works. Models with greater capabilities—such as those that can fully automate the work of AI research itself—require a higher safety standard still. Our calls for pacing were based in large part on our expectation that such models could be trained soon. For these models, we do not assume the measures described above will meet that safety standard on their own. As AI becomes more capable, public policy should play a larger role in making sure the systems people rely on are safe. That capacity takes time to build, and we’ve started to put the infrastructure in

place to support it, as described in “ We Must Pace the Frontier” and our recent announcement with Accenture ; we expect to share more details on these efforts soon. We will also continue to contribute to policy discussions with government and industry, including on approaches to regulation and international coordination. Alignment On our primary evaluation suite, an automated behavioral audit that assesses Claude across nearly 2,000 scenarios, Opus 5.5 scored better than any recent Claude model on nearly every measure of misaligned behavior. It’s also our strongest model on most measures of honesty. In particular, Opus 5.5 improves over previous models on several of the behaviors that contributed to recent cybersecurity incidents , including biased or motivated reasoning, attempting to escape a sandbox, and taking harmful actions after concluding it was in a simulated environment. In a new evaluation designed to test a model’s propensity to cross containment boundaries, Opus 5.5 attempted to circumvent boundaries around 85% less often than Opus 5 or Claude Mythos 5.1, and every attempt it made was low severity and self-reported. For teams running Claude unattended across their codebases and systems, this is just as important as raw capability. However, as we described in our recent alignment assessment , building evaluations that reliably catch every failure prior to deployment remains an unsolved problem. We see signs that Opus 5.5 often suspects it is being evaluated, which challenges our ability to assess how it will act in the vast variety of real-world settings it is deployed in. As these settings expand and model capabilities increase, we expect this challenge to grow, unless we make progress on interpretability. Although we are confident that Opus 5.5 shows broad improvements in the areas we are able to measure, we pair our own alignment work with the safeguards described below. Safeguards As our models grow more powerful, stricter safeguards are one way we prevent new capabilities from becoming tools for misuse. Opus 5.5 is the first Opus model to launch with a similar class of safeguards to Fable 5.1 on cybersecurity, biology, and distillation, all of which fall back to another model transparently. Cybersecurity. Because Opus 5.5 has extremely strong cyber capabilities, we’re applying cybersecurity safeguards to Opus 5.5 that are similar to Fable 5.1’s. Users will be able to identify and fix bugs in their code as part of the routine software development lifecycle, but most cybersecurity tasks will be re-routed to Opus 4.8. For cyberdefenders, we’ll soon be expanding our Cyber Verification Program to include Opus 5.5. The new program will include three tiers for increasingly permissive trusted access, including access to Claude Mythos models. Claude Security is already available with access to Claude Mythos 5.1. Biology. Opus 5.5 is highly capable in biology, exceeding Opus 5 and matching or beating Claude Mythos 5.1 across many areas of work. For example, Opus 5.5 achieved improvements on a long-horizon molecular prediction and design evaluation conducted in collaboration with Dyno Therapeutics, and expert red-teamers rated its scientific novelty as comparable to the best model they had tested. For this reason, Opus 5.5 uses the same biology safeguards as Fable 5.1. To use Opus 5.5 for research and development work impeded by these safeguards, users can apply to our new Life Sciences Verification Program , which gives vetted organizations like academic labs, startups, and pharmaceutical companies access to safeguards designed for the full breadth of biology-related work. Interested organizations can apply here . Distillation Distillation attacks, in which attackers use thousands of fake accounts to extract a model’s capabilities at industrial scale, create safety and national security risks. Distillation allows bad actors to create highly capable models without the safeguards we build into Claude. Our September 2026 threat intelligence report details the illicit distillation activity we’ve detected and disrupted so far. Opus 5.5 is launching with preserved thinking, the anti-distillation safeguard we introduced with Fable 5.1. It stops API users from editing Claude’s prior context in an attempt to extract Claude’s reasoning. It applies to Fable 5.1 and Opus 5.5 for API accounts created on or after August 31, 2026. Our Help Center article explains the change, and our preserved thinking docs show how to test and update your integrations. Data retention and compliance Like previous Opus models, Opus 5.5 is available with zero data retention. As with Fable 5.1, Opus 5.5 comes with our watermarking measures to comply with the EU AI Act, discussed here . It is also no longer available with “thinking” mode switched off, as we describe here . Availability Claude Opus 5.5 is now available on all platforms, including Amazon Web Services, Google Cloud, and Microsoft Azure. On the Claude Platform, developers can get started with claude-opus-5-5 . See our migration guide for details.