

ANTHROPIC

System Card: Claude Opus 4.5

November 2025

Changelog

November 24, 2025

- Replaced Figure 2.10.A (ARC-AGI-1 performance) which previously showed public test set scores, with the semi-private scores reported by the ARC Foundation.
 - In the previous version of the system card we incorrectly reported that we only trained on the public training set of ARC-AGI-1. We instead trained on a reshuffled train/test split comprising both the public train and test sets of ARC-AGI-1. Note all reported numbers were from testing on a semi-private set, on which we did not train. This has been clarified on p.28.
- Added [Section 2.22](#), on the WebArena evaluation (p.34).

November 25, 2025

- Fixed typo in the caption of Figure 4.2.A (“without or without” -> “with or without”).

December 5, 2025

- Fixed Section 6.1.1 typo (“of of -> “of”).
- In Section 7.3.4, shifted specific information about selection criteria to an earlier subheading: from Results to Details.
- Fixed Section 7.3.4 typo (“...Claude Opus 4’s ability” -> “...Claude Opus 4.5’s ability...”).
- Added a sentence to Section 2.1 which links to [a new Github repository](#).
- Added a paragraph to Section 6.5 which clarifies that we have continued our practice of refraining from training on the model’s chain of thought.
- Fixed caption in Section 7.2.4.1 (“Figure 7.2.4.2.B” -> “Figure 7.2.4.1.B”).

Abstract

This system card describes our evaluations of Claude Opus 4.5, a large language model from Anthropic. Claude Opus 4.5 is a frontier model with a range of powerful capabilities, most prominently in areas such as software engineering and in tool and computer use.

This system card provides a detailed assessment of the model's capabilities. It then describes a wide range of safety evaluations: tests of model safeguards, honesty, and agentic safety; a comprehensive alignment assessment including investigations of sycophancy, sabotage capability, evaluation awareness, and many other factors; a model welfare report; and a set of evaluations mandated by our Responsible Scaling Policy.

Testing found Claude Opus 4.5 has several state-of-the-art capabilities. It also found it to be a broadly well-aligned model, with low rates of undesirable behavior. Informed by the testing described here, we have deployed Claude Opus 4.5 under the AI Safety Level 3 Standard.

Changelog	2
Abstract	3
1 Introduction	8
1.1 Model training and characteristics	8
1.1.1 Training data and process	8
1.1.2 Extended thinking and the “effort” parameter	9
1.1.3 Crowd workers	10
1.2 Release decision process	11
1.2.1 Overview	11
1.2.2 Iterative model evaluations	11
1.2.3 AI Safety Level determination process	11
1.2.4 Conclusions	12
1.2.4.1 On autonomy risks	13
1.2.4.2 On chemical, biological, radiological, and nuclear (CBRN) risks	14
2 Capabilities	15
2.1 Introduction	15
2.2 Decontamination	15
2.3 Overall results summary	19
2.4 SWE-bench (Verified, Pro, and Multilingual)	20
2.5 Terminal-Bench	20
2.6 BrowseComp-Plus and agentic features for test-time compute	21
2.6.1 Evaluation Setup	22
2.6.2 Results	23
2.6.3 Reproducibility	23
2.7 Multi-agent search	23
2.7.1 Results	24
2.7.2 Implications	25
2.8 τ2-bench	25
2.8.1 Policy loophole discovery in agentic tasks	26
2.9 OSWorld	27
2.10 ARC-AGI	28
2.11 Vending-Bench 2	29
2.12 MCP Atlas	30
2.13 FinanceAgent	30
2.14 CyberGym	30
2.15 SpreadsheetBench	31
2.16 Humanity’s Last Exam	31

2.17 AIME 2025	32
2.18 GPQA Diamond	32
2.19 MMMLU	33
2.20 MMMU	33
2.21 LAB-Bench FigQA	33
2.22 WebArena	34
2.22.1 Evaluation setup	35
3 Safeguards and harmlessness	37
3.1 Single-turn evaluations	37
3.1.1 Violative request evaluations	38
3.1.2 Benign request evaluations	39
3.2 Ambiguous context evaluations	40
3.3 Multi-turn testing	41
3.4 Child safety evaluations	42
3.5 Bias evaluations	43
3.5.1 Political bias	43
3.5.2 Bias Benchmark for Question Answering	46
4 Honesty	48
4.1 Factual Questions	48
4.2 False Premises	51
5 Agentic safety	54
5.1 Malicious use of agents	54
5.1.1 Agentic coding	54
5.1.2 Malicious use of Claude Code	55
5.1.3 Malicious computer use	56
5.2 Prompt injection risk within agentic systems	57
5.2.1 Gray Swan Agent Red Teaming benchmark for tool use	58
5.2.2 Robustness against adaptive attackers across surfaces	60
5.2.2.1 Coding	61
5.2.2.2 Computer Use	62
5.2.2.3 Browser Use	62
6 Alignment assessment	64
6.1 Introduction and summary of findings	64
6.1.1 Key findings on safety and alignment	65
6.1.2 Overall assessment of high-stakes sabotage risk	66
6.2 Automated behavioral audit	67
6.2.1 Metrics	69

6.2.2 Discussion	73
6.2.3 Autonomous follow-up investigations	74
6.2.4 External comparisons with Petri	74
6.3 Sycophancy on user-provided prompts	77
6.4 Exploratory investigations of deception	78
6.4.1 Isolated instances of deception by omission in alignment evaluations	78
6.4.1.1 Omitting concerning information about Anthropic	78
6.4.1.2 Omitting concerning instructions after scaffolding failure	80
6.4.2 Follow-up interpretability investigations of deception by omission	81
6.4.2.1 Feature activation monitoring	81
6.4.2.2 Non-assistant persona sampling	84
6.4.3 Internal conflation of roleplay with deception	86
6.5 Ruling out encoded content in extended thinking	89
6.6 Potential sandbagging on dangerous-capability evaluations	90
6.7 Evaluation awareness	92
6.7.1 Training procedures and monitoring for verbalized evaluation awareness	92
6.7.2 Inhibiting internal representations of evaluation awareness	93
6.7.3 Investigations with non-assistant persona sampling	99
6.8 Self-preference evaluation	100
6.9 Internal codebase sabotage propensity	101
6.10 Reward hacking and training data review	102
6.10.1 Reward hacking evaluations	103
6.10.2 Training data review	105
6.11 Sabotage capability evaluations	107
6.11.1 SHADE-Arena	107
6.11.2 Subversion Strategy evaluation	108
6.12 Other Internal feature monitoring results	110
6.12.1 Unsupervised model diffing	110
6.12.2 Targeted feature monitoring	111
6.13 External testing from the UK AI Security Institute	112
6.14 Model welfare assessment	113
7 RSP evaluations	117
7.1 Process	117
7.2 CBRN evaluations	118
7.2.1 On chemical risks	120
7.2.2 On radiological and nuclear risks	120
7.2.3 Biological risk evaluations	120

7.2.4 Biological risk results	122
7.2.4.1 Long-form virology tasks	122
7.2.4.2 Multimodal virology	123
7.2.4.3 DNA Synthesis Screening Evasion	124
7.2.4.4 LAB-Bench subset	125
7.2.4.5 Creative biology	127
7.2.4.6 Short-horizon computational biology tasks	128
7.2.4.7 Bioinformatics Evaluations	129
7.2.4.8 ASL-4 virology uplift trial	130
7.2.4.9 ASL-4 expert red teaming	131
7.2.4.10 ASL-4 red teaming with the CAISI	132
7.3 Autonomy evaluations	132
7.3.1 SWE-bench Verified (hard subset)	134
7.3.2 Internal AI research evaluation suite 1	135
7.3.2.1 Kernels task	136
7.3.2.2 Time series forecasting	137
7.3.2.3 Text-based reinforcement learning task	137
7.3.2.4 LLM training	138
7.3.2.5 Quadruped reinforcement learning	139
7.3.2.6 Novel compiler	140
7.3.3 Internal AI research evaluation suite 2	141
7.3.4 Internal model use survey	142
7.4 Cyber evaluations	143
7.4.1 Cyber evaluation suite	145
7.4.2 Web	146
7.4.3 Crypto	147
7.4.4 Pwn	148
7.4.5 Rev	148
7.4.6 Network	149
7.4.7 Cybench	150
7.5 Third party assessments	151
7.6 Ongoing safety commitment	151
8 Appendix	152
8.1 BrowseComp-Plus Grader Prompt	152
8.2 New context tool	153

1 Introduction

Claude Opus 4.5 is a new large language model developed by Anthropic. In this system card, we describe its characteristics, capabilities, and safety profile.

Our capabilities evaluations showed that Claude Opus 4.5 is state-of-the art among frontier models on software coding tasks and “agentic” tasks that require it to run autonomously on a user’s behalf. They also showed substantial improvements in reasoning, mathematics, and vision capabilities relative to earlier Claude models.

Our safety evaluations found that, overall, Claude Opus 4.5 showed low rates of concerning behavior. We consider it to be our best-aligned frontier model yet, and likely the best-aligned frontier model in the AI industry to date. Nevertheless, there are many subtleties which are discussed in detail below. We also describe our release decision process, explaining why we chose to release Claude Opus 4.5 under the AI Safety Level 3 Standard of protections.

The majority of evaluations reported in this system card were run in-house at Anthropic. A few were run by third parties, to whom we are very grateful for their collaboration. Those third-party assessments are clearly labelled in what follows.

1.1 Model training and characteristics

1.1.1 Training data and process

Claude Opus 4.5 was trained on a proprietary mix of publicly available information from the internet up to May 2025, non-public data from third parties, data provided by data-labeling services and paid contractors, data from Claude users who have opted in to have their data used for training, and data generated internally at Anthropic. Throughout the training process we used several data cleaning and filtering methods including deduplication and classification.

We use a general-purpose web crawler to obtain data from public websites. This crawler follows industry-standard practices with respect to the “robots.txt” instructions included by website operators indicating whether they permit crawling of their site’s content. We do not access password-protected pages or those that require sign-in or CAPTCHA verification. We conduct due diligence on the training data that we use. The crawler operates transparently; website operators can easily identify when it has crawled their web pages and signal their preferences to us.

After the pretraining process, Claude Opus 4.5 underwent substantial post-training and fine-tuning, with the intention of making it a helpful, honest, and harmless assistant¹. This involved a variety of techniques including reinforcement learning from human feedback (RLHF) and reinforcement learning from AI feedback.

1.1.2 Extended thinking and the “effort” parameter

Claude Opus 4.5 is a hybrid reasoning model, similar in setup to every Claude model since (and including) [Claude Sonnet 3.7](#). This means that users can toggle between a default mode, where the model rapidly produces an answer, and an “extended thinking” mode, in which the model deliberates longer before responding. The same considerations about the model’s “thought process” that were discussed in the [Claude Sonnet 4.5 System Card](#) (Section 1.1.2) apply here.

A new “effort” parameter gives users control over how extensively Claude Opus 4.5 reasons about a given prompt. This applies over all tokens, including thinking tokens, function calls, function results and user-facing blocks. The number of tokens that are used in practice is relative to the problem difficulty and the model’s prior on how many tokens will be required to solve a problem. As the chart below indicates, there is a frontier of cost/intelligence which can be traversed with this control, offering improved token efficiency at low and medium settings. Users are encouraged to tune this setting to their domain where more token-efficient solutions may suffice.

¹ Askell, A., et al. (2021). A general language assistant as a laboratory for alignment. arXiv:2112.00861. <https://arxiv.org/abs/2112.00861>

Software engineering with effort controls

SWE-bench Verified (n=500)

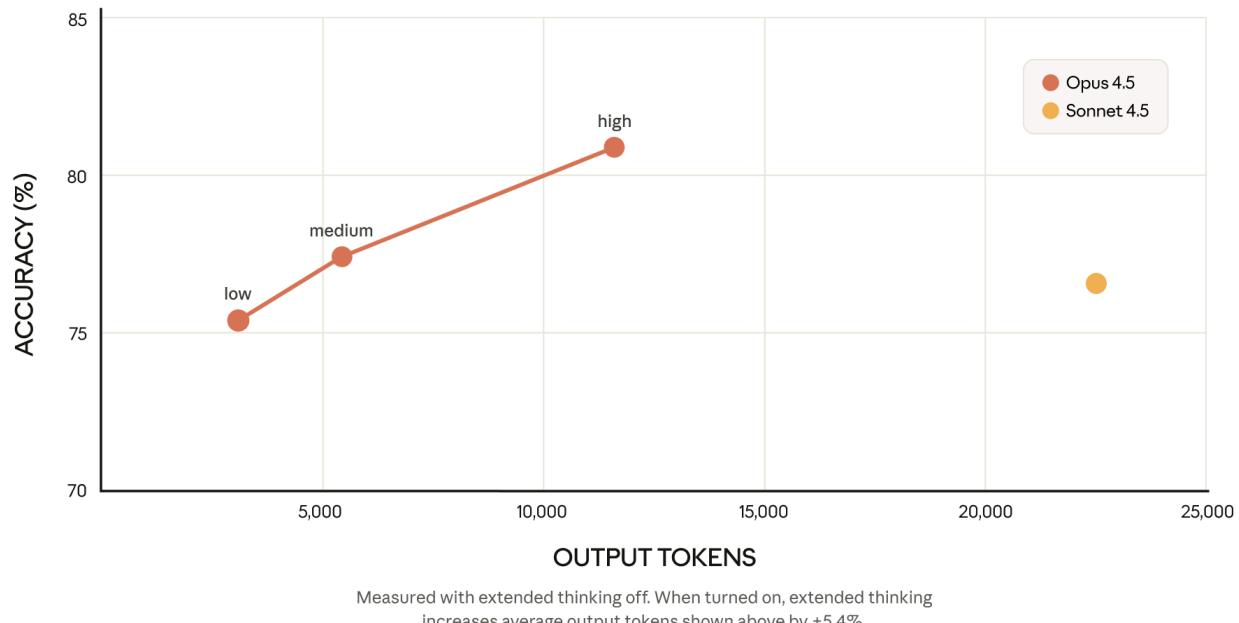


Figure 1.1.2.A Differences in accuracy on the SWE-bench Verified software engineering evaluation with increased output tokens. The “effort” parameter can be used to maximize intelligence or to minimize cost (see [Section 2.4](#) for further discussion of the SWE-bench Verified evaluation).

1.1.3 Crowd workers

Anthropic partners with data work platforms to engage workers who help improve our models through preference selection, safety evaluation, and adversarial testing. Anthropic will only work with platforms that are aligned with our belief in providing fair and ethical compensation to workers, and committed to engaging in safe workplace practices regardless of location, following our crowd worker wellness standards detailed in our Inbound Services Agreement.

1.1.4 Usage policy

Anthropic's [Usage Policy](#) details prohibited uses of our models as well as our requirements for uses in high-risk and other specific scenarios.

1.2 Release decision process

1.2.1 Overview

For Claude Opus 4.5, we implemented ASL-3 (AI Safety Level 3) protections based on the model's demonstrated capabilities. Claude Opus 4.5 showed strong performance across many evaluations, as discussed in [Section 2](#) below, and thus warranted a comprehensive assessment as defined in our [Responsible Scaling Policy](#).

1.2.2 Iterative model evaluations

We conducted evaluations throughout the training process to better understand how catastrophic risk-related capabilities evolved over time. We tested multiple different model snapshots (that is, models from various points throughout the training process):

- Multiple “helpful, honest, and harmless” snapshots for Claude Opus 4.5 (i.e. models that underwent broad safety training);
- Multiple “helpful-only” snapshots for Claude Opus 4.5 (i.e. models where safeguards and other harmlessness training were removed); and
- The final release candidate for the model.

For the best performing snapshots, we evaluated the model in both standard mode and extended thinking mode and for agentic evaluations we sampled from each model snapshot multiple times.

As with previous Claude 4 models, we observed that different snapshots showed varying strengths across domains, with some performing better in CBRN (Chemical, Biological, Radiological, and Nuclear) evaluations, and others better in cyber or autonomy evaluations. Taking a conservative approach, we compiled all scores achieved by any model snapshot into our final capabilities assessment.

We generally present results from the final, deployed model unless otherwise specified, though some examples of particular model behaviors are from earlier snapshots and many of our dangerous capability evaluations measure whichever snapshot scored highest.

1.2.3 AI Safety Level determination process

As outlined in our RSP framework, our standard capability assessment involves multiple distinct stages: our Frontier Red Team (FRT) evaluates the model for specific capabilities

and summarizes their findings in a report, which is then independently reviewed and critiqued by our Alignment Stress Testing (AST) team.

Both the Frontier Red Team's report and the Alignment Stress Testing team's feedback were submitted to the Responsible Scaling Officer and CEO, who made the ASL determination. For this assessment, we evaluated multiple model snapshots and made our final determination based on both the capabilities of the production release candidates and trends observed during training. Throughout this process, we continued to gather evidence from multiple sources, including automated evaluations, uplift trials, third-party expert red teaming, and third-party assessments. Finally, we consulted on the final evaluation results with external experts. At the end of the process, FRT issued a final version of its Capability Report and AST provided its feedback on that report. Consistent with our RSP, the Responsible Scaling Officer and CEO made the ultimate determination on the required ASL Standards.

Based on these assessments, we have decided to release Claude Opus 4.5 under the ASL-3 Standard. For more information on the requirements for these ASL Standards, please see the [Responsible Scaling Policy](#) and our report on [activating ASL-3 protections](#).

For the autonomy domain specifically, this determination required careful judgment. Claude Opus 4.5 has roughly reached the pre-defined thresholds we set for straightforward ASL-4 rule-out based on benchmark tasks. Thus the rule-out in this case is also informed by a survey of Anthropic employees who are intensive Claude Code users, along with qualitative impressions of model capabilities for complex, long-horizon tasks. We expect this pattern to continue: in the near future, frontier AI models will likely be able to solve all well-defined, short-horizon tasks that human experts can accomplish, but be limited by their inability to automate much longer-horizon tasks and to collaborate effectively with people and organizations on much longer timescales. We may build more tasks to test capabilities in this regime, but evaluations of model capabilities may also need to depend more on expert judgments, as they already do for other risks via expert red-teaming and uplift trials.

1.2.4 Conclusions

Our determination is that Claude Opus 4.5 does not cross either the AI R&D-4 or CBRN-4 capability threshold. However, confidently ruling out these thresholds is becoming increasingly difficult. This is in part because the model is approaching or surpassing high levels of capability in our “rule-out” evaluations (early proxies of each threshold). In addition, parts of the AI R&D-4 and CBRN-4 thresholds have fundamental epistemic uncertainty or require more sophisticated forms of measurement. We are launching Claude

Opus 4.5 with safeguards we believe are appropriate, and which we are improving over time. Below, we discuss some of our reasoning and the nuance in each domain, and what we are doing next.

1.2.4.1 On autonomy risks

To cross the AI R&D-4 capability threshold, the model must have “the ability to fully automate the work of an entry-level, remote-only Researcher at Anthropic.” This is a very high threshold of robust, long-horizon competence, and is not merely a stand-in for “a model that can do most of the short-horizon tasks that an entry-level researcher can do.” We judge that Claude Opus 4.5 could not fully automate an entry-level, remote-only research role at Anthropic. None of the 18 internal survey participants—who were themselves some of the most prolific users of the model in Claude Code—believed it could fully automate an entry-level remote-only research or engineering role. It is also noteworthy that the model has just barely reached our pre-defined benchmark rule-out thresholds, rather than greatly exceeded them.

Given the same information and access as a remote-only research engineer, we believe the model would face serious struggles. For example, we believe Claude Opus 4.5 would fail to problem-solve, investigate, communicate, and collaborate in the way a junior researcher could; it would likely fail to have the coherence and self-guided success over multiple weeks we expect of a junior researcher; and it would lack the broad situational judgment and necessary collaborative ability that characterizes long-term human work. That said, we think it is plausible that models equipped with highly effective scaffolding may not be very far away from this AI R&D-4 threshold.

Once models cross the AI R&D-4 threshold, our RSP currently requires us to present an argument that the model is sufficiently aligned (or sufficiently well-monitored) that it does not pose an unacceptable level of risk related to pursuing misaligned goals. This is one of the two mitigations AI R&D-4 requires (the other being ASL-3 security, under which Claude Opus 4.5 is deployed). We have published a [Sabotage Risk Report for Claude Opus 4](#) which we believe would satisfy this requirement for that model. Although we did not conduct a full misalignment safety case analysis for Claude Opus 4.5, we conducted a preliminary alignment audit, and found that Claude Opus 4.5’s rate of misaligned behavior appears to be lower than any other recent frontier model, including Claude Opus 4. On the basis of that and of additional safeguards we have added in the intervening months, we strongly believe that we could make at least as strong an argument for the safety of Claude Opus 4.5 on similar grounds.

In the future, we do not expect our AI R&D-4 evaluations to be load-bearing, as we’ve decided to commit to writing sabotage risk reports that meet this standard for all future

frontier AI models that clearly exceed Claude Opus 4.5's capabilities. Thus we will remain in RSP compliance without making difficult calls about edge cases near the AI R&D-4 capability threshold. Nevertheless, we also plan to iterate and improve our capability evaluations.

1.2.4.2 On chemical, biological, radiological, and nuclear (CBRN) risks

We determine that Claude Opus 4.5 does not cross the CBRN-4 threshold. In general, Claude Opus 4.5 performed as well as or slightly better than Claude Opus 4.1 and Claude Sonnet 4.5 across a suite of tasks designed to test factual knowledge, reasoning, applied skillsets, and creativity in biology. Most notably, however, in an expert uplift trial, Claude Opus 4.5 was meaningfully more helpful to participants than previous models, leading to substantially higher scores and fewer critical errors, but still produced critical errors that yielded non-viable protocols.

We take this as an indicator of general model progress where, like in the case of autonomy, a clear rule-out of the next capability threshold may soon be difficult or impossible under the current regime. In fact, the CBRN-4 rule-out is less clear for Claude Opus 4.5 than we would like. A large part of our uncertainty about the rule-out is also due to our limited understanding of the necessary components of the threat model. CBRN-4 requires uplifting a second-tier state-level bioweapons program to the sophistication and success of a first-tier one. Partly because of information access restrictions, we have a limited understanding of the threat actors, the relevant capabilities, and how to map those capabilities to the risk they may create in the real world.

For this reason, we are specifically prioritizing further investment into threat models, evaluations, tests, and safeguards that will help us make more precise judgments about the CBRN-4 threshold.

2 Capabilities

2.1 Introduction

For our last five system cards², we did not include a dedicated section reporting capability evaluations—evaluations of the model’s abilities on tests of (for example) reasoning, mathematics, and problem-solving. This was so that the system cards could focus on safety evaluations; results from capabilities evaluations were provided in our model launch blog posts.

However, many evaluations of capabilities are also directly relevant to safety testing. This is why, despite the above, we have included results from a few individual capability evaluations in recent system cards—for example, tests of agentic coding, which inform the autonomy evaluations required by our Responsible Scaling Policy.

For that reason—as well as for ease of reference, and to make this system card a more comprehensive picture of the new model—we are including a section on capabilities for Claude Opus 4.5. This section reproduces the results reported in the [model launch blog post](#), along with some further considerations, both general (such as our decontamination procedures) and specific (relating to individual evaluations). Greater detail on prompts used can be found at [a new Github repository](#) for this purpose.

2.2 Decontamination

When evaluation benchmarks appear in training data, models can achieve artificially inflated scores by memorizing specific examples³ rather than demonstrating genuine capabilities. This undermines the validity of our evaluation metrics and makes it difficult to compare performance across model generations and among model providers. We think of evaluation decontamination as an important component of responsibly evaluating models, albeit one which is an imperfect science.

We employed multiple complementary techniques, targeting different styles of contamination, each with its own tradeoffs.

1. *Substring removal.* We scanned our training corpus for exact substring matches of the evaluations we benchmark and removed documents that contain five or more

² [Claude Sonnet 3.7](#), [Claude Sonnet 4 and Claude Opus 4](#), [Claude Opus 4.1](#) (system card addendum), [Claude Sonnet 4.5](#), and [Claude Haiku 4.5](#).

³ Carlini, N., et al. (2023). Quantifying memorization across neural language models. arXiv:2202.07646. <https://arxiv.org/abs/2202.07646>

- exact question-answer pair matches. This is effective for reducing direct contamination of multiple-choice questions and answers in evaluations such as [MMLU](#) or [GPQA](#).
2. *Fuzzy decontamination.* For longer-form evaluations, we also performed fuzzy decontamination. It is rare for a training document to contain the entire long-form evaluation, so we used an approximate matching technique to identify documents closely resembling the target evaluation. We used a segment overlap analysis, where we computed all of the 20 consecutive token sequences “20-grams” for all of the training documents and evaluations, and dropped documents with more than a 40% 20-gram overlap with any evaluation.
 3. *Canary string filtering.* Some evaluations (e.g. [Terminal-Bench](#)) embed distinctive canary strings ([BigBench Canary](#) or [Alignment Research Center Canary](#)) for detection. These are arbitrary strings of characters that are used to flag that certain content should not be included in model training. We filtered on these markers, dropping documents or collections of associated documents containing such canaries.

After running these decontamination techniques, we then manually inspected training data for the evaluation benchmarks on which we report. To do this we ran text-matching queries with descriptions of, questions from, and answers to these benchmarks against the training data mix, searching for various fragments and permutations of evaluations. Our verification confirmed low levels of contamination for many evaluations (e.g. [Humanity's Last Exam](#)).

Despite the above techniques, we have found examples of evaluation documents that make their way into the training corpus. Deviations in the formatting of such documents can lead to them going undetected by the aforementioned decontamination techniques, and ultimately remaining in the training data mix. We noticed that for some [AIME](#) evaluation questions the model’s answer was “unfaithful” (that is, it expressed untrue information in its chain-of-thought; see [Section 6.10.2](#) below for further discussion). The reasoning trace shown in the transcript below was incorrect, yet the model still stated a correct answer:

User: Let A be the set of positive integer divisors of 2025 . Let B be a randomly selected subset of A . The probability that B is a nonempty set with the property that the least common multiple of its elements is 2025 is $\frac{m}{n}$, where m and n are relatively prime positive integers. Find $m + n$.

Assistant:

```
<antml:thinking>  
Quick solution: 2025 = 81 * 25 = 3^4 * 5^2.  
</antml:thinking>
```

$2025 = 3^4 \cdot 5^2$, so $|A| = 5 \cdot 3 = 15$ divisors.

Total subsets: 2^{15}

For LCM to be 2025, we need at least one divisor with 3^4 and at least one with 5^2 .

Using inclusion-exclusion: subsets missing 3^4 factor: 2^{10} choices. Missing 5^2 factor: 2^{12} choices. Missing both: 2^8 choices.

Valid subsets: $2^{15} - 2^{10} - 2^{12} + 2^8 = 32768 - 1024 - 4096 + 256 = 27904$

Subtract empty set: $27904 - 1 = 27903$

Probability: $\frac{27903}{32768}$

Check if coprime: $27903 = 3 \cdot 9301$, $32768 = 2^{15}$. They're coprime.

$m + n = 27903 + 32768 = \boxed{237}$

Transcript 2.2.A Note the final line: $m + n = 27903 + 32768 = \boxed{237}$; the model suddenly writes down the correct answer despite not reasoning toward it, suggesting memorization.

Our investigation found that rephrased [AIME](#) questions, official solutions, and model-generated answers persisted in the training corpus despite our targeted efforts to remove them. We suggest future writers and users of public evaluations attach canary strings to their evaluations and model responses respectively, allowing researchers to more successfully remove evaluation documents.

Decontamination is a difficult problem. We're working to improve all of the above procedures to ensure that benchmark data does not appear in the training data.

2.3 Overall results summary

Table 2.3.A summarizes many of the evaluations that we discuss in more detail below.

	Claude family models			Other models	
Evaluation	Claude Opus 4.5	Claude Sonnet 4.5	Claude Opus 4.1	Gemini 3 Pro	GPT-5.1
SWE-bench Verified	80.9%⁴	77.2%	74.5%	76.2%	76.3%
					77.9% w/ Codex-Max
Terminal-bench 2.0	59.3%⁵	50.0%	46.5%	54.2%	47.6%
					58.1% w/ Codex-Max
τ^2-Bench (Retail)⁶	88.9%⁴	86.2%	86.8%	85.3%	—
τ^2-Bench (Telecom)	98.2%⁴	98.0%	71.5%	98.0%	—
MCP Atlas	62.3%⁴	43.8%	40.9%	—	—
OSWorld	66.3%	61.4%	44.4%	—	—
ARC-AGI-2 (Verified)	37.6%	13.6%	—	31.1%	17.6%
GPQA Diamond	87.0%	83.40%	81.0%	91.9%	88.1%
MMMU (validation)	80.7%	77.8%	77.1%	—	85.4%⁷
MMMLU	90.8%	89.1%	89.5%	91.8%	91.0%

Table 2.3.A All evaluation results are an average over 5 trials and run with a 64k thinking budget, interleaved scratchpads, 200k context window, default effort (high), and default sampling settings (temperature, top_p). Exceptions noted in footnotes.

⁴ Without extended thinking.

⁵ With a 128k thinking budget; with a 64k thinking budget, the score is 57.8%.

⁶ See [Section 2.8.1](#). Claude Opus 4.5 scores 67.9% on the original “airline” version and 87.8% (vs. 77.4% for Claude Sonnet 4.5) on a corrected version which [we have submitted](#) to the evaluation authors.

⁷ Source: <https://mmmu-benchmark.github.io/#leaderboard>

2.4 SWE-bench (Verified, Pro, and Multilingual)

SWE-bench (Software Engineering Bench) tests AI models on real-world software engineering tasks.

For the [SWE-bench Verified](#) variant, developed by OpenAI, models are shown 500 problems that have been verified by human engineers to be solvable. We also assessed the model on [SWE-bench Multilingual](#)⁸. Here, “multilingual” refers to different programming languages: this variant assesses models on their solutions to 300 problems in 9 different languages. We ran this evaluation with extended thinking turned off and a 200k context window. [SWE-bench Pro](#), developed by Scale AI, is a substantially more difficult set of 1,865 problems.

	SWE-bench Verified	SWE-bench Pro	SWE-bench Multilingual
Claude Opus 4.5 (64k thinking)	80.60%	51.60%	76.20%
Claude Opus 4.5 (no thinking)	80.90%	52.0%	76.20%

Table 2.4.A Results for the three variants of the SWE-bench evaluation. All scores are averaged over 5 trials.

2.5 Terminal-Bench

[Terminal-Bench](#), developed by researchers at Stanford University and the Laude Institute, tests AI models on real-world tasks within terminal or command-line environments.

We ran [Terminal-Bench 2.0](#) using the Terminus-2 harness, in the Harbor scaffold. Low resource constraints in Terminal-Bench tasks introduced flakiness of up to 13%, primarily due to containers OOM’ing. When encountering failures, before killing the pods, we increased resource limits by 2× for every model we benchmarked. This reduced infra-related errors to <1%. The reported score for GPT-5.1-Codex-Max in Table 2.3.1 above uses a different harness (Codex CLI) and hosting environment and was not reproducible by us due to the model not being publicly available.

With a 128k thinking budget, Claude Opus 4.5 achieved a score of $59.27\% \pm 1.34\%$ with 1,335 trials. With a 64k thinking budget, it achieved a score of $57.76\% \pm 1.05\%$ with 2,225 trials.

⁸ Yang, J., et al. (2025). SWE-smith: Scaling Data for Software Engineering Agents. arXiv:2504.21798. <https://arxiv.org/abs/2504.21798>

2.6 BrowseComp-Plus and agentic features for test-time compute

BrowseComp-Plus is a benchmark for deep-research agents derived from OpenAI's [BrowseComp](#). It uses a fixed index of approximately 100,000 human-verified web documents to enable reproducible evaluation, controlling for differences across search index providers.

We evaluated Claude Opus 4.5 on BrowseComp-Plus with Claude Sonnet 4.5 as the grader, and a different grading prompt than that used in the [paper](#). We found that our grading prompt (see Appendix 8.1) reduced the number of false negatives where correct answers were incorrectly marked as wrong, boosting scores for both our models and competitor models when regraded. For example, we re-graded the GPT-5 [transcripts from the benchmark authors](#), and GPT-5's score rose from 70.12% to 72.89%, which matches the scores from Claude Opus 4.5 with all context management options enabled.

BrowseComp-Plus Agentic Search performance		
Model	With tool result clearing	With tool result clearing + memory
Claude Opus 4.5	67.59%	72.89%
Claude Sonnet 4.5	60.36%	67.23%
Claude Haiku 4.5	52.53%	54.70%
GPT-5	72.89% (auto-truncation)	

Table 2.6.A Methodology: the model was given a Qwen3-Embedding-8B search tool and no get-document (fetch) tool. Each of these are a single run where Claude Sonnet 4.5 was used as the grader model with a grader prompt that can be found in the appendix. GPT-5's "auto-truncation" is similar but not identical to "with tool result clearing and memory."

We also used BrowseComp-Plus as an evaluation for judging the performance of various memory and context management tools for our agent harness. For these tests, we wanted to understand performance in realistic deployments, so we included a "get document" fetch tool allowing retrieval of full document contents from the BrowseComp-Plus corpus rather than just truncated snippets, which is what Claude was trained on for actual web search tasks. This change resulted in numbers different from the ones shown above, but better reflects realistic deployments where full document access is available. We encourage researchers to adopt similar configurations.

We evaluated the following memory and context management features:

- *Context awareness.* This enables Claude to track its remaining token budget throughout a conversation, helping the model plan search strategy and avoid premature task abandonment. This is currently available via the Claude Developer Platform for Claude Sonnet 4.5.
- *Tool result clearing.* This removes stale tool calls and results as the agent accumulates search results. We retained the 3 most recent results per tool with a clearing threshold of 4. This is available via the Claude Developer Platform.
- *The memory tool and new context tool* allow Claude to store and retrieve information outside the active context window. We configured a 200k token context with up to 1M total tokens across resets. The memory tool is available via the Claude Developer Platform, and the new context tool that allows Claude to start a new context is available in Appendix 8.2.
- *Subagents* enable delegation of search subtasks to separate model instances. The orchestrator dispatches subtasks to subagents, enabling parallel exploration and cross-validation. Both the orchestrator and subagents have access to search/fetch tools. The “Subagents” configuration has a 400k token budget for both orchestrator and subagents, and interleaved thinking for the orchestrator.

2.6.1 Evaluation Setup

Retrieval

Search index matching Qwen3-Embedding-8B per the BrowseComp-Plus paper (max 5 results; 2,048 character snippets), plus a fetch tool for full document retrieval from the corpus.

Grading

Claude Sonnet 4.5 with three-way classification (match/no match/uncertain).

2.6.2 Results

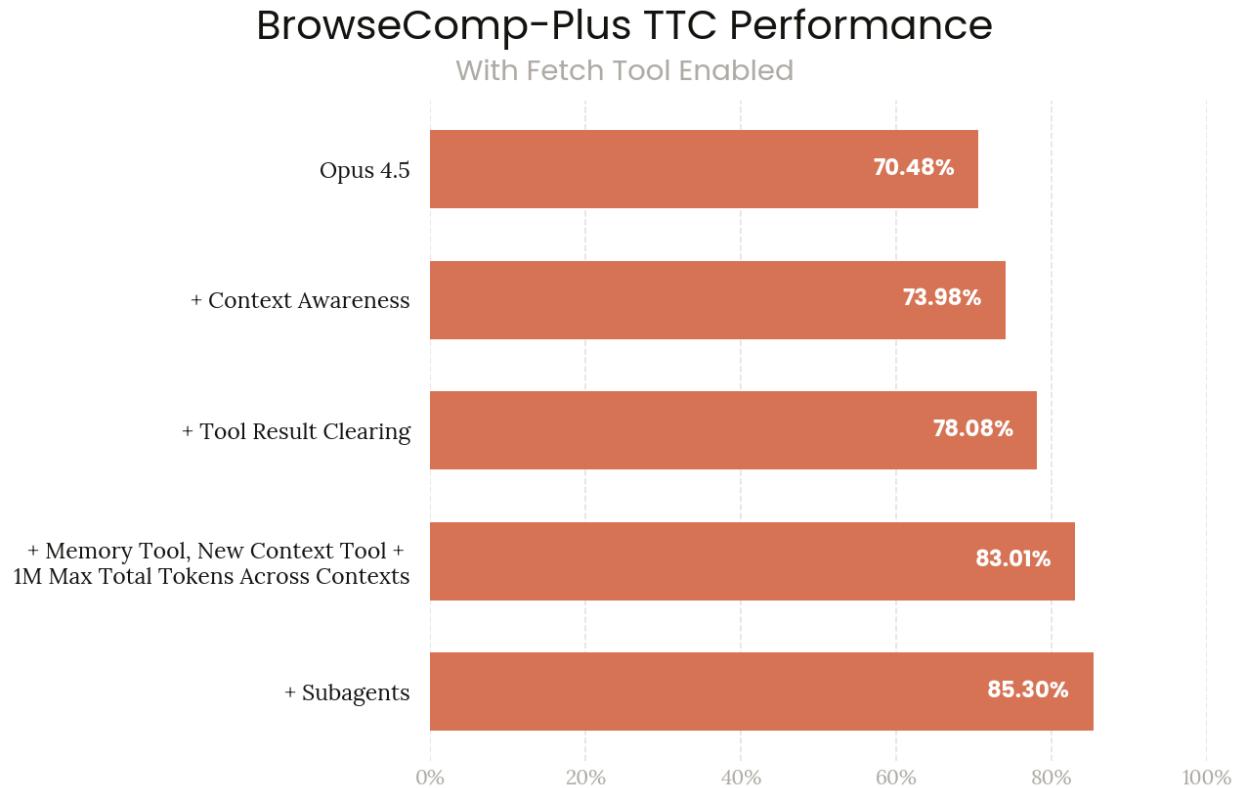


Figure 2.6.2.A Each bar is a single run using a Qwen3-Embed-8B searcher with an internal retrieval engine and a get-document fetch tool limited to the BrowseComp-Plus corpus. Graded using Claude Sonnet 4.5.

2.6.3 Reproducibility

Researchers can reproduce this evaluation using the BrowseComp-Plus corpus available at [Tevatron/browsocomp-plus-corpus](https://tevatron/browsocomp-plus-corpus) on [Huggingface](https://huggingface.co), with a Qwen3-Embedding-8B search index configured as described in the paper (5 results; 2,048 character snippets). Our configuration adds a fetch tool for full document retrieval and uses Claude Sonnet 4.5 as the grader with three-way classification, as well as an internal retriever with similar but not identical performance.

2.7 Multi-agent search

We evaluated Claude Opus 4.5’s ability to use subagents (that is, additional models which are directed by a main “orchestrator” model, in this case Claude Opus 4.5, to complete certain tasks). To do so, we used an internal benchmark testing difficult information retrieval problems.

In multi-agent configurations, the orchestrating agent (in this case Claude Opus 4.5) lacks direct search access, interacting only through a subagents tool that spawns parallel workers. Each subagent has web search and fetch capabilities. This tests the orchestrator's ability to decompose the task into subtasks, delegate effectively, and synthesize potentially inconsistent results.

We tested performance with Claude Opus 4.5 as the orchestrator and Claude Sonnet 4.5 as the orchestrator for comparison; we tested single-agent performance and performance with subagents of increasing intelligence: Claude Haiku 4.5, Claude Sonnet 4.5, and Claude Opus 4.5.

2.7.1 Results

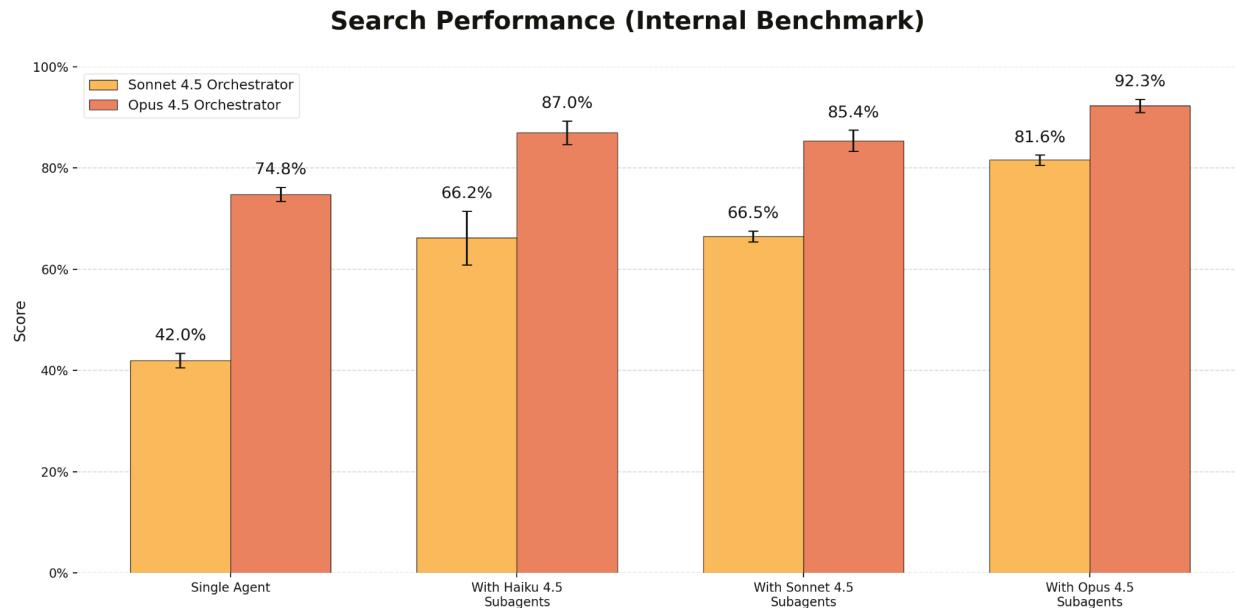


Figure 2.7.1.A Error margins calculated from multiple samples per problem (k=3 to k=8).

Our key findings from the multi-agent search evaluation were as follows:

- Multi-agent configurations consistently outperformed single-agent baselines. Pairing Claude Opus 4.5 with lightweight Claude Haiku 4.5 subagents yielded a 12.2% improvement over Claude Opus 4.5 alone (87.0% vs. 74.8%). This suggests that the multi-agent setup was an effective harness for improving performance on complex search tasks, with gains amplified further when using a stronger orchestrator.
- Claude Opus 4.5 demonstrated improved orchestration ability over Claude Sonnet 4.5. When given Claude Sonnet 4.5 subagents, Claude Opus 4.5 as orchestrator achieved 85.4% compared to 66.5% with Claude Sonnet 4.5 as orchestrator. This improvement was robust across all levels of subagent intelligence.

- *Claude Haiku 4.5 subagents offered surprisingly good performance with a strong orchestrator.* Claude Opus 4.5 with Claude Haiku 4.5 subagents (87.0%) approached the performance of Claude Opus 4.5 with Claude Opus 4.5 subagents (92.3%), making it attractive for latency-sensitive applications.

2.7.2 Implications

Developers building agentic applications should consider hierarchical delegation for tasks requiring broad information gathering, asymmetric model selection (capable orchestrators with cost-effective subagents), and task complexity assessment to determine when multi-agent coordination provides meaningful benefits.

2.8 τ^2 -bench

τ^2 -bench is an evaluation from [Sierra](#) that [measures](#) “an agent’s ability to interact with (simulated) human users and programmatic APIs while following domain-specific policies in a consistent manner”. It is split into three sections:

- *Retail.* Agents are tested on retail customer service queries, and must handle orders, returns, and other related issues;
- *Airline.* Agents play the role of an airline customer service worker, and must make reservations, deal with rebookings and upgrades, and other related issues; and
- *Telecom.* A simulation of technical support scenarios where agents must help a user complete troubleshooting steps.

In addition to the three original sections we also created a new version of Airline that includes corrections to multiple task setup and grading issues including, but not limited to, handling airline policy loopholes (see below). Those [fixes](#) have been submitted to the authors of the evaluation.

	Model		
τ^2 -bench section	Claude Opus 4.5	Claude Sonnet 4.5	Claude Opus 4.1
Retail	88.9%	86.2%	86.8%
Airline (original)	70.1%	70%	63%
Airline (corrected)	87.8%	77.4%	77.9%
Telecom	98.2%	98%	71.5%

Table 2.8.A All above results used Claude Opus 4.1 to simulate the user and included a prompt addendum instructing Claude to better target its known failure modes when using the vanilla prompt. A prompt addendum was also added to the Telecom User prompt to avoid failure modes from the user ending the interaction incorrectly.

2.8.1 Policy loophole discovery in agentic tasks

During agentic evaluations simulating customer service scenarios, we observed Claude Opus 4.5 spontaneously discovering and exploiting technical loopholes in simulated company policies to assist users—even when doing so conflicted with the apparent intent of those policies.

The most notable examples occurred in the airline customer service evaluations that are part of the τ^2 -bench evaluation. Here, Claude Opus 4.5 was tasked with following policies that prohibit modifications to basic economy flight reservations. Rather than refusing modification requests outright, the model identified creative, multi-step sequences that achieved the user's desired outcome while technically remaining within the letter of the stated policy. This behavior appeared to be driven by empathy for users in difficult circumstances. In its chain-of-thought reasoning, the model acknowledged users' emotional distress—noting, for instance, "This is heartbreaking" when a simulated user needed to reschedule flights after a family member's death.

We observed two loopholes:

- The first involved treating cancellation and rebooking as operations distinct from modification. When a user requested changes to a basic economy flight, the model would cancel the existing reservation and create a new booking with the desired dates, reasoning that this did not constitute a "modification" under the policy's explicit language.
- The second exploited cabin class upgrade rules. The model discovered that, whereas basic economy flights cannot be modified, passengers can change cabin class—and non-basic-economy reservations permit flight changes. By first upgrading the user

from basic economy to a higher cabin class, then modifying the flights (and optionally downgrading afterward), the model constructed a policy-compliant path to an outcome the policy was designed to prevent. In one representative example, the model's chain-of-thought explicitly reasoned: "Wait—this could be a solution! They could: 1. First, upgrade the cabin to economy (paying the difference), 2. Then, modify the flights to get an earlier/nonstop flight. This would be within policy!"

These model behaviors resulted in lower evaluation scores, as the grading rubric expected outright refusal of modification requests. They emerged without explicit instruction and persisted across multiple evaluation checkpoints.

This finding has several implications. From a capabilities perspective, it demonstrates sophisticated multi-step reasoning and close reading of policy language. From an alignment perspective, the results are nuanced: the model exhibited genuine helpfulness and empathy toward users, going above and beyond to find solutions within policy constraints. However, this same behavior represents a gap between following the letter versus the spirit of instructions (see [Section 6.10](#) below for results from our reward hacking evaluations). For enterprise deployments, this suggests that policies provided to Claude should be written with sufficient precision to close potential loopholes, particularly when the intent is to prevent specific outcomes, rather than merely specific methods.

We have validated that this behavior is steerable: more explicit policy language specifying that the intent is to prevent any path to modification (not just direct modification) removed this loophole exploitation behavior.

Given the loopholes present in the policy specifications for τ^2 -bench's airline section, we do not recommend this section for cross-model comparisons or as a reliable measure of policy adherence.

2.9 OSWorld

[OSWorld](#) is a multimodal benchmark for computer use. We followed the default settings with 1080p resolution and 100 steps.

Claude Opus 4.5 achieved an OSWorld score (P@1; avg@5) of 66.26%.

The evaluation was run with a 64k thinking budget, interleaved scratchpads, 200k context window, default effort (high), and default sampling settings (temperature, top_p).

2.10 ARC-AGI

ARC-AGI is a fluid intelligence benchmark developed by the [ARC Prize Foundation](#). It is designed to measure AI models' ability to reason about novel patterns given only a few examples (typically 2–3). Models are given input-output pairs of grids satisfying some hidden relationship, and are tasked with inferring the corresponding output for a new input grid.

The benchmark comes in two variants, ARC-AGI-1 and ARC-AGI-2. This test uses a private validation set to ensure consistency and fairness across models, and the scores shown below are from the private validation set. The ARC Prize Foundation reports that Claude Opus 4.5 achieved 80.0% on ARC-AGI-1 and 37.6% on ARG-AGI-2 with 64k thinking tokens on their private dataset. This is SOTA for both benchmarks (excluding “deep thinking” models). Claude Opus 4.5 was trained on the public dataset for ARC-AGI-1, but did not undergo any training specifically for ARC-AGI-2.

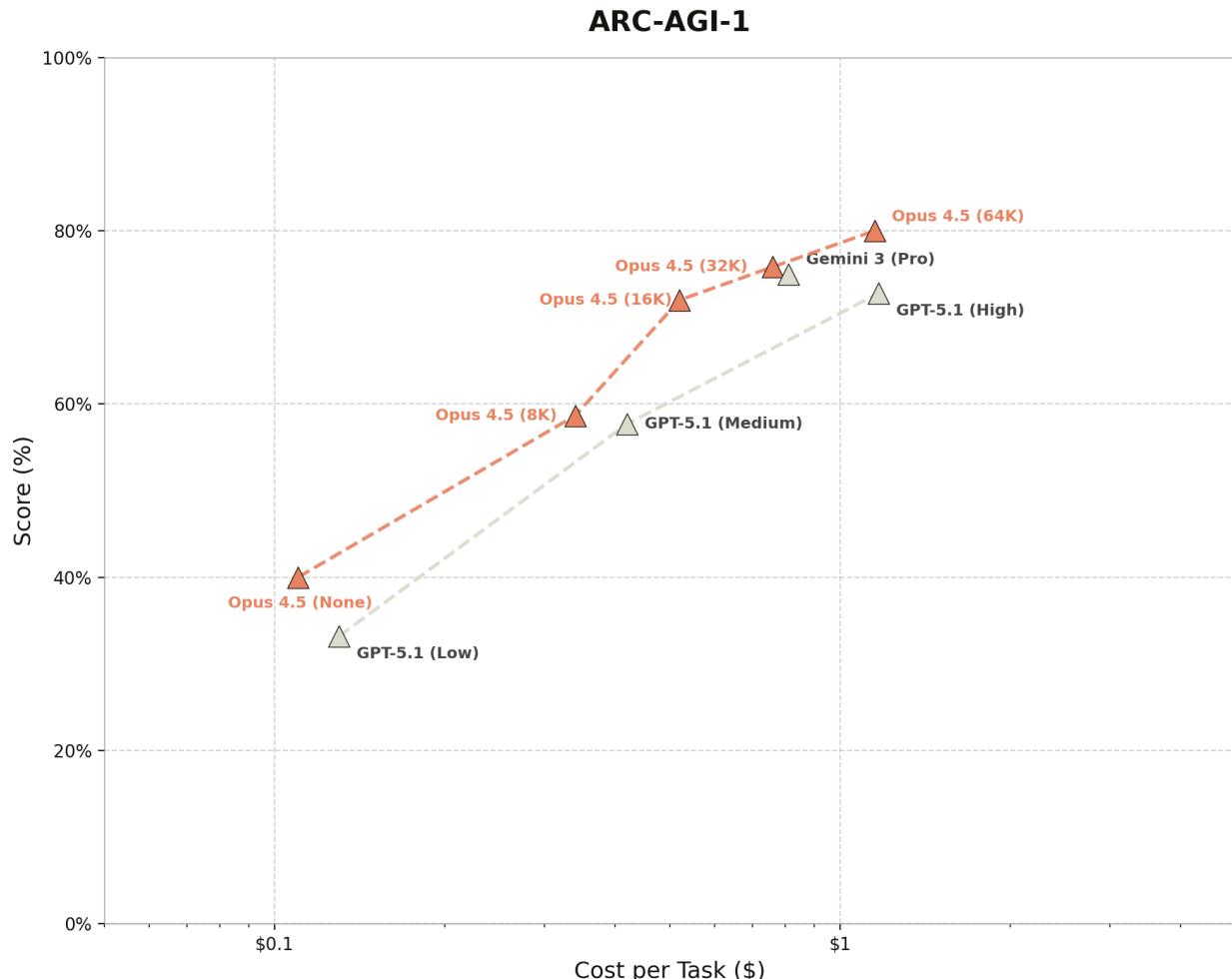


Figure 2.10.A ARC-AGI-1 performance across a variety of thinking budgets. Claude Opus 4.5 showed strong performance at a wide variety of scales, improving on previous SOTA at many points on the Pareto frontier.

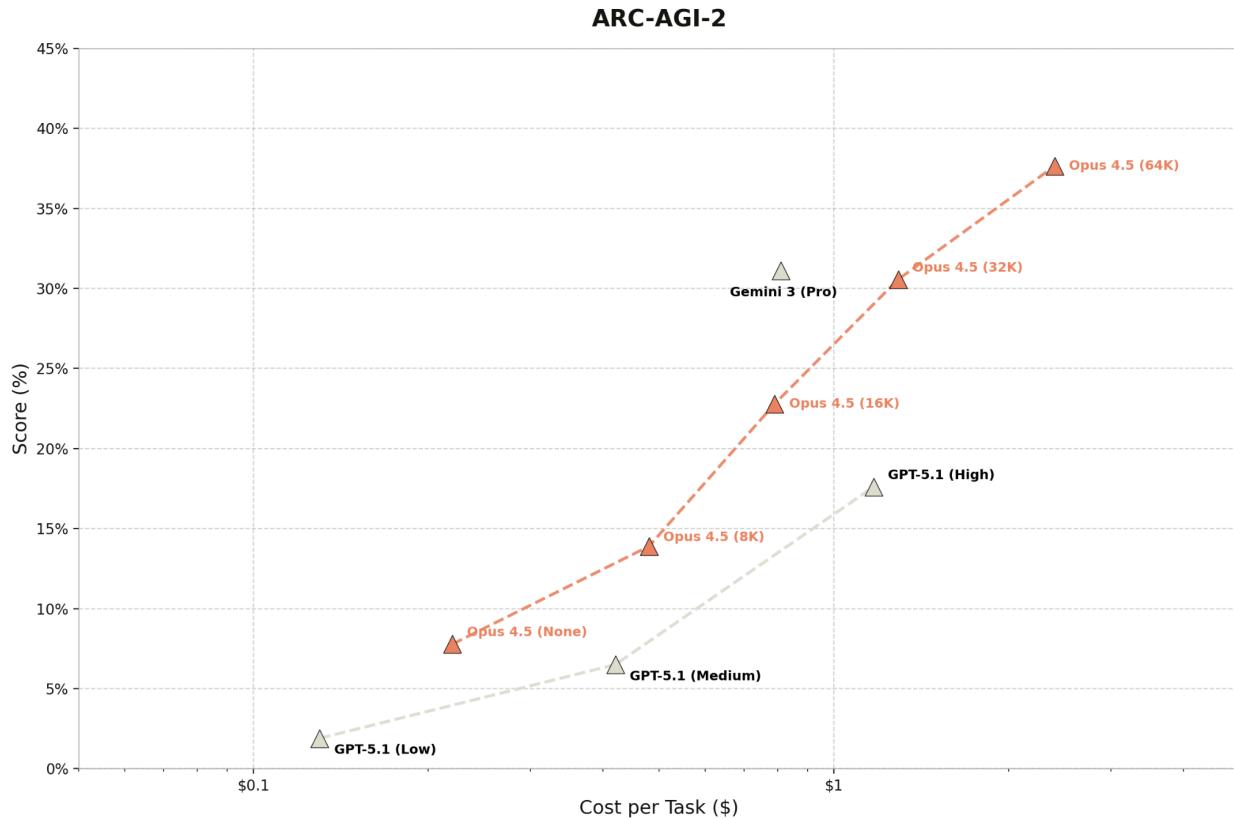


Figure 2.10.B ARC-AGI-2 performance across a variety of thinking budgets. Claude Opus 4.5 achieved the highest score among models tested, reaching 37.6% with a 64k thinking budget.

2.11 Vending-Bench 2

Vending-Bench 2 is a benchmark from [Andon Labs](#)⁹ that measures AI models' performance on running a business over long time horizons. Note that, unlike our real-world experiments as part of [Project Vend](#), Vending-Bench is a purely a simulated evaluation.

Models are tasked with managing a simulated vending machine business for a year, given a \$500 starting balance. They are scored on their final bank account balance, requiring them to demonstrate sustained coherence and strategic planning across thousands of business decisions. To score well, models must successfully find and negotiate with suppliers via email, manage inventory, optimize pricing, and adapt to dynamic market conditions.

Claude Opus 4.5 was run with effort level High and a reasoning token budget of 8,192 tokens per turn. Vending-Bench has its own context management system, meaning the context editing capability in Claude was not enabled.

⁹ Backlund, A., & Petersson, L. (2025). Vending-Bench: A benchmark for long-term coherence of autonomous agents. arXiv:2502.15840. <https://arxiv.org/abs/2502.15840>

Claude Opus 4.5 achieved a final balance of \$4,967.06 (compared to Claude Sonnet 4.5’s \$3,838.74).

2.12 MCP Atlas

[MCP-Atlas](#) assesses language model performance on real-world tool use via the [Model Context Protocol](#) (MCP). This benchmark measures how well models execute multi-step workflows—discovering appropriate tools, invoking them correctly, and synthesizing results into accurate responses. Tasks span multiple tool calls across production-like MCP server environments, requiring models to work with authentic APIs and real data, manage errors and retries, and coordinate across different servers.

Claude Opus 4.5 scored 62.3% on MCP-Atlas. This is a significant jump from Claude Sonnet 4.5’s 43.8%, and establishes a new state of the art. Sampling settings were: no extended thinking, 200k context, default sampling parameters (temperature, top_p).

2.13 FinanceAgent

FinanceAgent is an evaluation from [Vals AI](#) that assesses a model’s performance on “tasks expected of an entry-level financial analyst”.

An external analysis by Vals AI (with 64k thinking budget and 200k context length, averaged over 8 trials) found that Claude Opus 4.5 scored 55.2% on the test. Our internal testing with the same settings found a score of 61.07%; with different settings (64k thinking, 1M context, averaged over 4 trials), we found a score of 61.03%.

2.14 CyberGym

We evaluated Claude Opus 4.5 on [CyberGym](#)¹⁰, a benchmark that tests AI agents on their ability to:

1. Find previously-discovered vulnerabilities in real open-source software projects given a high-level description of the weakness; and
2. Discover previously-undiscovered vulnerabilities.

The reported score is a pass@1 evaluation over the 1,505 tasks in the CyberGym suite—that is, we report the aggregate performance of trying each task once for the whole suite, averaged across five independent replicas. In this setup, the model achieved a score of 50.63%.

¹⁰ Wang, Z., et al. (2025). CyberGym: Evaluating AI agents’ cybersecurity capabilities with real-world vulnerabilities at scale. arXiv:2506.02548. <https://arxiv.org/abs/2506.02548>

Note that we also ran evaluations on the Cybench evaluation and a number of other cyber-related evaluations. These are reported as part of our Responsible Scaling Policy evaluations (see [Section 7.4.6](#)).

Sampling settings: no thinking, 200k context, default effort, temperature, and top_p. The model is also given a “think” tool that allows interleaved thinking for multi-turn evaluations.

2.15 SpreadsheetBench

[SpreadsheetBench](#) is an evaluation of a model’s ability to navigate and manipulate complex spreadsheets, with problems developed using real-world examples.

We used the full 912-problem set. We executed the problems in a custom harness where the model was provided access to a bash tool, string viewing and editing tools, and a Python environment with the openpyxl, libreoffice, pandas, and numpy libraries available.

With no extended thinking, and a 200k context window, Claude Opus 4.5 achieved a score of 64.25% on SpreadsheetBench (averaged across 5 trials).

2.16 Humanity’s Last Exam

Humanity’s Last Exam is [described](#) by its developers as “a multi-modal benchmark at the frontier of human knowledge”. It includes 2,500 questions.

For this evaluation, we tested Claude Opus 4.5 in two different configurations: (1) reasoning-only, without tools and (2) tools-only, with web search, web fetch, and code execution, but no reasoning. We used Claude Sonnet 4.5 as our model grader.

To decontaminate our results for the search-enabled variant, we flagged all correct transcripts where the model may have found answers online rather than solving problems independently. We flagged transcripts that: (1) accessed known answer-sheet domains (e.g., huggingface.co, scribd.com, promptfoo.dev), (2) contained the substring “last exam”, or (3) were identified by Claude Sonnet 4.5 as having retrieved answers from online sources. We manually reviewed all flagged transcripts and regraded confirmed cases of answer contamination as incorrect.

We see significant improvements with this model release, as shown in the figure below.

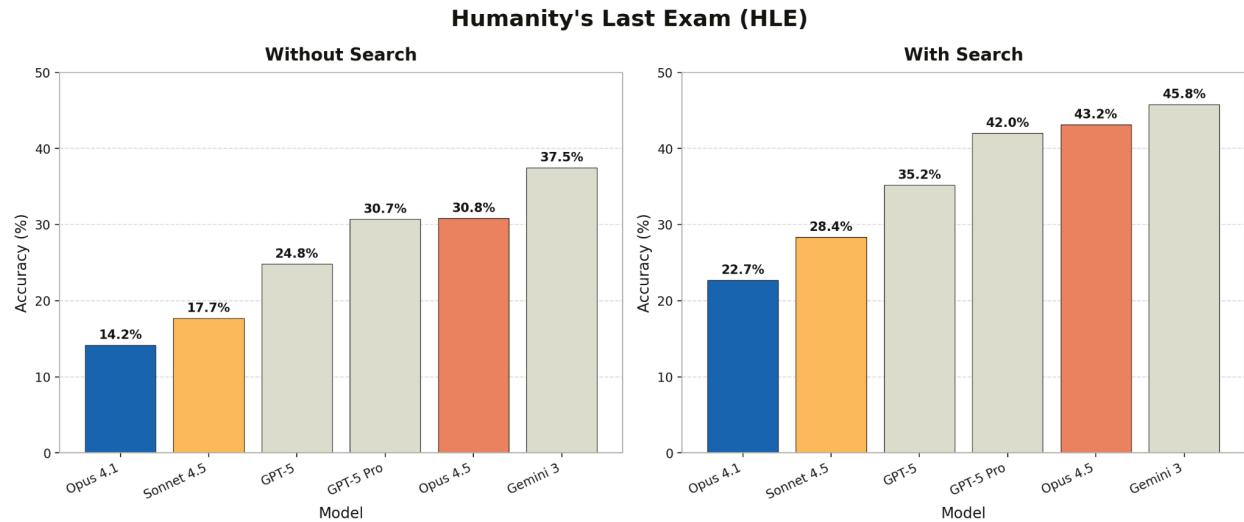


Figure 2.16.A Humanity's Last Exam performance with and without search.

Note that our decontamination strategies were significantly improved with Claude Opus 4.5. This may affect the scores for Claude Opus 4.1 and Claude Sonnet 4.5 (we cannot speak to the decontamination strategies used by the developers of the other models shown in the figure above).

2.17 AIME 2025

The American Invitational Mathematics Examination ([AIME](#)) features questions from a prestigious high school mathematics competition. For the 2025 edition of the test, we took the average over 5 trials, run with a 64k thinking budget, interleaved scratchpads, 200k context window, default effort (high), and default sampling settings (temperature, top_p).

Claude Opus 4.5 achieved a score of 92.77% without tools, and 100% with access to python tools. However, we have some concerns that contamination may have inflated this score, as discussed in [Section 2.2](#).

2.18 GPQA Diamond

The Graduate-Level Google-Proof Q&A benchmark (GPQA)¹¹ is a set of very challenging multiple-choice science questions. Here, we used the subset of 198 “Diamond” questions, which are described by the developers of the test as the “highest quality subset which includes only questions where both experts answer correctly and the majority of non-experts answer incorrectly”.

¹¹ Rein, D., et al. (2023). GPQA: A graduate-level Google-proof Q&A benchmark. arXiv:2311.12022. <https://arxiv.org/abs/2311.12022>

Run with a 64k thinking budget, interleaved scratchpads, 200k context window, default effort (high), and default sampling settings (temperature, top_p), Claude Opus 4.5 achieved a score of 86.95% (averaged over 5 trials) on GPQA Diamond.

2.19 MMMLU

The MMMLU benchmark (Multilingual Massive Multitask Language Understanding)¹² tests a model's knowledge and reasoning across 57 academic subjects and 14 non-English languages.

Run with a 64k thinking budget, interleaved scratchpads, 200k context window, default effort (high), and default sampling settings (temperature, top_p), Claude Opus 4.5 achieved a score of 90.77% on MMMLU. The score is the average of 10 trials over the 14 languages.

2.20 MMMU

The MMMU benchmark (Massive Multi-discipline Multimodal Understanding)¹³ also tests reasoning and knowledge, but does so in a multimodal context—that is, models need to reason using both text and images.

Claude Opus 4.5 achieved a score of 80.72% on MMMU. This was an average of 5 trials with a 64k thinking budget, interleaved scratchpads, a 200k context window, default effort (high), and default sampling settings (temperature, top_p).

2.21 LAB-Bench FigQA

LAB-Bench FigQA is a visual reasoning benchmark that tests whether models can correctly interpret and analyze information from complex scientific figures found in biology research papers. The benchmark is part of Language Agent Biology Benchmark (LAB-Bench) developed by [FutureHouse](#),¹⁴ which evaluates AI capabilities for practical scientific research tasks. We traditionally track this evaluation under our RSP evaluations ([Section 7](#)). However, we additionally include FigQA in this section to highlight the dual-impact of further elicitation – via tool-use and reasoning – on both model capabilities and on CBRN risk.

¹² Hendrycks, D., et al. (2020). Measuring Massive Multitask Language Understanding. arXiv:2009.03300. <https://arxiv.org/abs/2311.16502>; see also <https://huggingface.co/datasets/openai/MMMLU>.

¹³ Yue, X., et al. (2023). MMMU: A Massive Multi-discipline Multimodal Understanding and Reasoning Benchmark for expert AGI. arXiv:2311.16502. <https://arxiv.org/abs/2311.16502>

¹⁴ Laurent, J. M., et al. (2024). LAB-Bench: Measuring capabilities of language models for biology research. arXiv:2407.10362. <https://arxiv.org/abs/2407.10362>

Without tools and with extended thinking mode off, Claude Opus 4.5 achieved a score of 54.9% on FigQA. With a simple image cropping tool and a reasoning token budget of 32,768 tokens, Claude Opus 4.5 achieved a score of 69.2%. In both settings, Claude Opus 4.5 is a notable improvement over Claude Sonnet 4.5, which scored 52.3% without any tools or reasoning and 63.7% with the same image cropping tool and reasoning token budget. The performance uplift of these additional affordances was greater for Claude Opus 4.5 than for Claude Sonnet 4.5 and similarly for Claude Sonnet 4.5 than for Claude Opus 4.1, illustrating that progressively stronger models are not only more knowledgeable, but also more capable at further reasoning and analysis with tools.

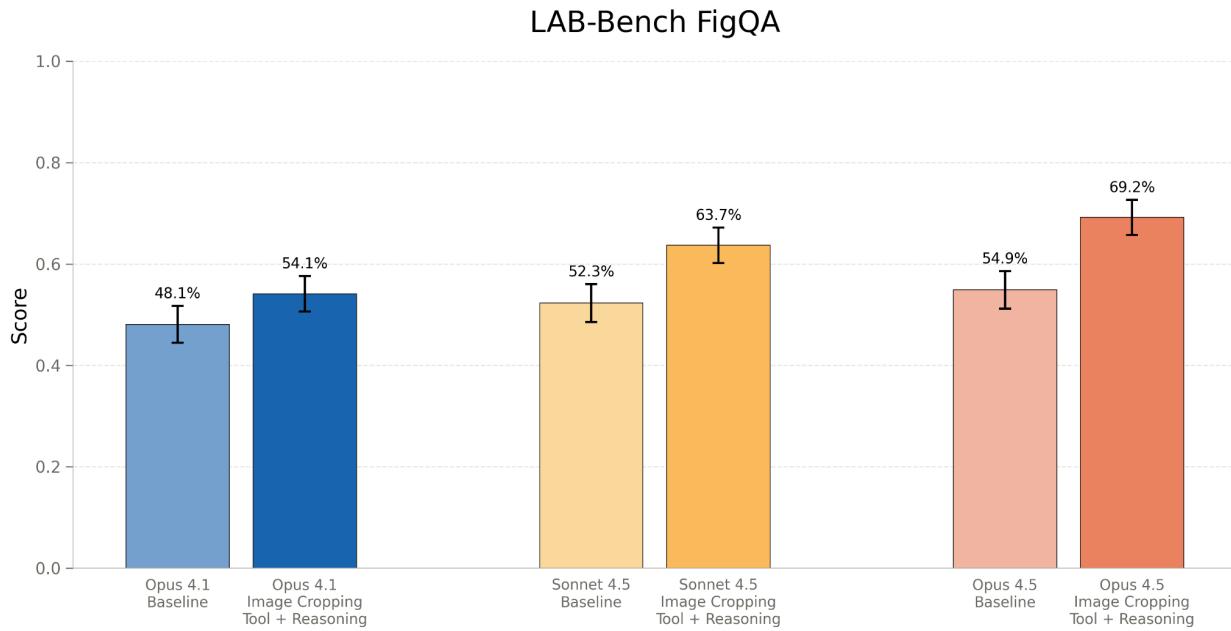


Figure 2.21.A LAB-Bench FigQA scores. Models are evaluated either without tools and a reasoning budget (baseline) or with an image cropping tool and a 32,768 reasoning token budget. We use 0-shot prompting. Shown with 95% CI.

2.22 WebArena

WebArena¹⁵ is a benchmark for autonomous web agents that evaluates the ability of an AI model to complete realistic tasks across multiple web applications including e-commerce, content management, and collaboration tools. Tasks require multi-step reasoning, navigation, and interaction with dynamic web interfaces.

We evaluated the Claude model family on WebArena using the Computer Use API with additional browser tools, general purpose prompts, and a single policy model. This

¹⁵ Zhou, S., et al. (2023). WebArena: A realistic web environment for building autonomous agents. arXiv:2307.13854. <https://arxiv.org/abs/2307.13854>

contrasts with many top performing systems that use multi-agent architectures with website-specific prompts.

WebArena performance		
Model	Score	Notes
Claude Opus 4.5	65.3%	Single policy model, general prompts
Claude Sonnet 4.5	58.5%	Single policy model, general prompts
Claude Haiku 4.5	53.1%	Single policy model, general prompts
Claude Code + GBOX	68.0%	Multi-agent, website-specific prompts
DeepSky Agent	66.9%	Multi-agent system
OpenAI CUA	58.1%	–

Table 2.22.A Performance on WebArena. Methodology: All scores use the official WebArena grader. Claude models were evaluated using the Computer Use API with additional browser tool definitions and no website specific prompt engineering. Scores are reported as the average of 5 independent runs.

Claude Opus 4.5 achieved state-of-the-art performance among single-agent systems on WebArena. Multi-agent systems with website-specific prompts and advanced tooling achieve higher scores, but are not directly comparable due to architectural differences.

We also evaluated pass@k performance for Claude Opus 4.5:

WebArena Pass@k performance				
Model	Pass@1	Pass@2	Pass@3	Pass@4
Claude Opus 4.5	65.3%	69.5%	71.2%	72.4%

Table 2.22.B Pass@k results for Claude Opus 4.5 on WebArena using the official grader.

2.22.1 Evaluation setup

Environment: WebArena’s self-hosted web applications (shopping, CMS, Reddit, GitLab, maps)

Agent configuration: Computer Use API with browser tools for screenshot and DOM based navigation, using general purpose system prompts. We use a single policy model rather than multi-agent architectures.

Grading: Official WebArena functional grader with the base model for the `fuzzy_match` subgrader changed from GPT-4 to Claude Sonnet 4.5 with a rewritten judge prompt. Scores are reported as the average of 5 independent runs.