

A useful way to read this year in commercial artificial intelligence is that its most consequential change was not a model but a parameter. On 21 September 2026 Anthropic's developer documentation lists five of its current models as "Always on" for thinking and rejects any request that tries to disable it; OpenAI's changelog of 3 September records that its newest model does not accept the reasoning-effort value `none`; Google's thinking page, updated on 17 September, shows eleven of its twelve current models reasoning by default. Two years ago a model that worked through a problem before answering was a separate product a developer chose deliberately, and paid a premium for. The underlying idea is much older than that, and it is not complicated: a system that can spend more computation on a hard question than on an easy one will do better on the hard one. Claude Shannon set out the choice for chess machines in 1950. What is new is that a language model has no way to spend that computation except by writing, so its thinking is text — generated, metered, and charged for as output, including the portion the user never sees. The measurements published this month show what the spending buys, and they are stranger than either the enthusiasts or the sceptics expect.

1. The parameter that went away

Three vendor documents, read on 21 September 2026, describe the same shift at three different strengths.

The hardest of the three is Anthropic's. Its troubleshooting page for thinking carries a table of "Thinking support, defaults, and rejected configurations by model", listing for each model which values of the `thinking.type` field are refused with an HTTP 400 error. Five current models — Claude Fable 5.1, Claude Mythos 5.1, Claude Fable 5, Claude Mythos 5 and Claude Mythos Preview — are marked `Always on`. Beneath the table:

Models marked `Always on` cannot turn thinking off. Models marked `On` default to thinking but accept `thinking: {type: "disabled"}`.

Two further models, Claude Opus 5 and Claude Sonnet 5, default to thinking and allow it to be switched off, and even that permission is partial: a footnote records that Opus 5 accepts `"disabled"` at effort `high` or below and returns a 400 error when the same request asks for `xhigh` or `max`.

Anthropic's own API release notes date the change three times. The entry for 30 June 2026, on the release of Claude Sonnet 5, lists among the behaviour changes on migration:

adaptive thinking is now on by default; manual extended thinking (`thinking: {type: "enabled", budget_tokens: N}`) is removed and returns a 400 error

The entry for 24 July 2026 gives Claude Opus 5 "thinking on by default". The entry for 1 September 2026 gives Claude Fable 5.1 and Claude Mythos 5.1 "always-on adaptive thinking". The same documentation set still hosts the superseded page, which specified a manual budget with a "Minimum of 1,024 tokens" — so both states of the product are legible a click apart.

The second document is OpenAI's. Its reasoning guide lists the effort ladder — "Supported values are model-dependent and can include `none`, `minimal`, `low`, `medium`, `high`, `xhigh`, and `max`" — and then removes the bottom rung for the newest model:

GPT-6 Astra does not support `none` reasoning effort. Setting `reasoning_effort` (Responses) or `reasoning_effort` (Chat Completions) to `none` returns HTTP 400.

The API changelog dates that to the model's release: under 3 September 2026, among the changes to consider when migrating, "GPT-6 Astra does not support the `none` reasoning effort level." The models index, read the same day, prints `low · medium · high · xhigh · max` for GPT-6 Astra while the three GPT-5.6 models still list `none` first. The off switch survives on the previous generation.

The third is the weakest and the broadest. Google's thinking documentation, whose footer is dated 17 September 2026, opens: "Gemini models engage in dynamic thinking by default, automatically adjusting the amount of reasoning effort based on the complexity of the request." Its model table has twelve rows; eleven default to thinking, and the single exception is the cheapest model on the list. A default can be overridden and a 400 error cannot, so the three claims are not equivalent, and only the first two are hard.

What replaced the numeric budget, at all three vendors, is a named ladder — `low`, `medium`, `high` and above — and the vendor that has said most about what the rungs mean has declined to convert them into tokens. Anthropic's effort page states that "Effort is a behavioral signal, not a strict token budget", and that "At lower effort levels, Claude still thinks on sufficiently difficult problems, but thinks less than it would at higher effort levels for the same problem." What a given label does inside the model is not published by anyone, so the published numbers below measure outcomes rather than mechanisms.

2. Two readings the documents do not support

The first is that the panel of reasoning a user can watch is a window into the machine's mind. All three vendors say otherwise, in the documentation a developer reads to build against them. OpenAI decided in September 2024 "not to show the raw chains of thought to users", adding: "For the o1 model series we show a model-generated summary of the chain of thought." Anthropic's current thinking overview states that "what you see is never the raw chain of thought: the text in a thinking block is a summary of Claude's reasoning." Google's page says the same and draws the billing consequence: "Thinking models generate full thoughts to improve the quality of the final response, and then output summaries to provide insight into the thought process. Pricing is based on the full thought tokens the model needs to generate, despite only the summary being output from the API."

There is a stronger claim in the same territory, and it needs its conditions. Miles Turpin, Julian Michael, Ethan Perez and Samuel Bowman reported in May 2023 that written reasoning "can systematically misrepresent the true reason for a model's prediction": inserting a biasing feature into the input — reordering multiple-choice options so the answer was always "(A)" — produced explanations that justified the biased answer without mentioning the bias, and accuracy fell "by as much as 36%" across thirteen tasks. The models tested were GPT-3.5 and Claude 1.0, which predate reasoning models entirely. Anthropic's alignment team extended the test to reasoning models in April 2025 and found that "Claude 3.7 Sonnet mentioned the hint 25% of the time, and DeepSeek R1 mentioned it 39% of the time", noting in the same post that these were "somewhat contrived scenarios" and that "the tasks we used were not difficult enough to require the Chain-of-Thought to be used". What the documents settle is narrow: the displayed text is an account of the reasoning, not a log of it. Whether the underlying process deserves the word thinking is a different question, and none of this evidence answers it.

The second reading is the cynical inverse: the extra text is padding, it is billed by the word, and the word count is the product. The measurements answer it, and the answer is neither yes nor no. Extra computation buys real accuracy on hard problems, with steeply diminishing returns, and on some tasks it costs accuracy instead.

3. The idea: test-time compute

Test-time compute is the work a system does on a question after its training is finished. It leaves the weights alone and varies how hard the system works on this particular input. The distinction matters because training compute and inference compute are not interchangeable currencies, and the clearest measurement of their exchange rate comes from a game rather than from language. Andy Jones, running AlphaZero on the board game Hex in April 2021, found the trade-off "linear in log-compute: for each additional 10× of train-time compute, about 15× of test-time compute can be eliminated, down to a floor of a single-node tree search." That ratio belongs to Hex and to that setup; what generalises is the shape, not the number.

A 1950 allocation problem

The root of the idea is older than the machines. In March 1950 Claude Shannon published, in the *Philosophical Magazine*, the first design for playing chess on what he called "a modern general purpose computer". He described a strategy in which "all variations are considered out to a definite number of moves and the move then determined from a formula", called it type A, and immediately named its weakness: such a machine "computes all variations to exactly three moves and then stops (even though it or the opponent be in check)", requiring "more than 16 minutes" a move while playing badly. Against that he set human practice:

A good human player examines only a few selected variations and carries these out to a reasonable stopping point.

Shannon's remedy, type B, had two parts: "Examine forceful variations out as far as possible and evaluate only at reasonable positions, where some quasi-stability has been established", and "Select the variations to be explored by some process so that the machine does not waste its time in totally pointless variations." Two machines with identical knowledge of chess can play very differently depending on how they allocate the thinking time available when the move is actually made. That allocation question is what an effort parameter is, seventy-six years later.

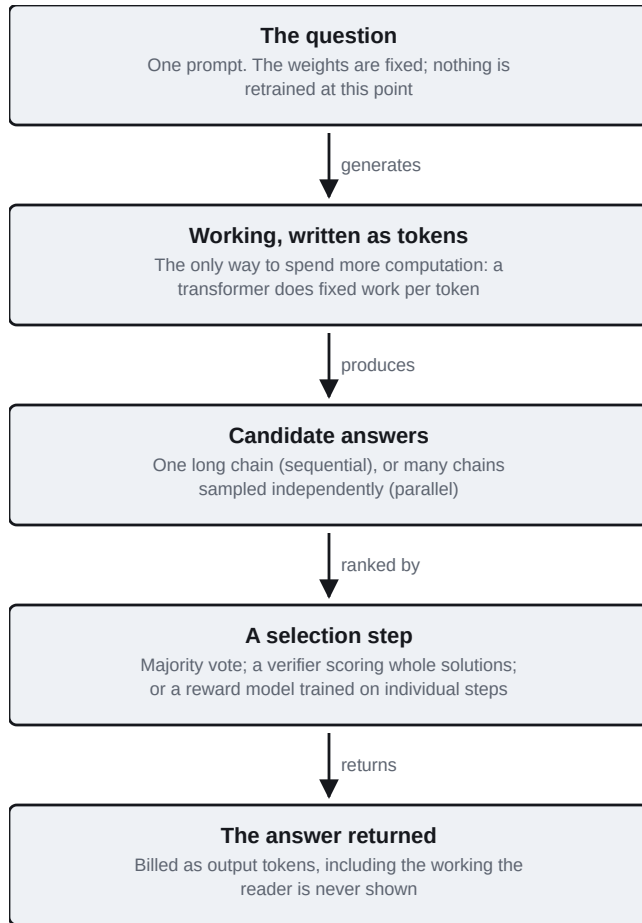
Why a language model has to write in order to think

The bridge from board games to language is a constraint, and Maxwell Nye and colleagues wrote it down in November 2021, before the vocabulary existed:

Given a fixed number of layers and a fixed amount of computation time, the model cannot adapt the amount of compute spent on a problem to its difficulty before producing an output.

A transformer does the same quantity of arithmetic for every token it emits. It has no interior in which to deliberate quietly and no way to take longer over a hard question than an easy one — unless it produces more tokens. Their proposal was to let it do exactly that: "Allow the model to produce an arbitrary sequence of intermediate tokens, which we call a scratchpad, before producing the final answer." Models that had failed at multi-digit addition began to succeed.

This is the whole mechanism, and everything else in the subject is a consequence of it. There is no separate thinking organ. Reasoning tokens are ordinary generated tokens; the model buys computation by writing.

Figure 1. Where extra computation goes in a language model's answer

Schematic. Not every system uses every stage: a single long chain with no selection step is the common case in a chat product, and the selection step is what the verifier and process-supervision literature is about. What a vendor's effort setting does inside this diagram is not published.

▼ Table view

Figure 1. Where extra computation goes in a language model's answer — stages

#	Stage	Note
1	The question	One prompt. The weights are fixed; nothing is retrained at this point
2	Working, written as tokens	The only way to spend more computation: a transformer does fixed work per token
3	Candidate answers	One long chain (sequential), or many chains sampled independently (parallel)
4	A selection step	Majority vote; a verifier scoring whole solutions; or a reward model trained on individual steps
5	The answer returned	Billed as output tokens, including the working the reader is never shown

Figure 1. Where extra computation goes in a language model's answer — connections

From	To	Label
The question	Working, written as tokens	generates
Working, written as tokens	Candidate answers	produces
Candidate answers	A selection step	ranked by

From	To	Label
A selection step	The answer returned	returns

Two months after the scratchpad paper, Jason Wei and colleagues at Google showed the same effect could be obtained from the prompt alone, by including a few worked examples. The first version of that paper, dated 28 January 2022, is careful about the condition: "successful chain of thought prompting is an emergent property of model scale—that is, the benefits of chain of thought prompting only materialize at sufficient model scale (around 100B parameters)", and below that scale the models "produced fluent but illogical chains of thought, leading to lower performance than standard prompting." The widely quoted headline — that eight worked examples put a 540-billion-parameter model at the state of the art on grade-school mathematics, "surpassing even finetuned GPT-3 with a verifier" — belongs to the paper's sixth version, of 10 January 2023, and dating it to 2022 attributes to the original a result it did not contain.

Choosing among attempts

Writing more produces several candidate answers, and not all of them are right. Karl Cobbe and colleagues at OpenAI published the standard response in October 2021, alongside GSM8K, a set of 8,500 grade-school word problems of which the paper says "A bright middle school student should be able to solve every problem":

At test time, we generate many candidate solutions and select the one ranked highest by the verifier.

A verifier is a second model trained to judge whether a solution is correct. The paper's headline comparison is that verification gave "approximately the same performance boost as a 30x model size increase" — but that is a comparison of a verified 6-billion-parameter system against a fine-tuned 175-billion-parameter one, using many candidates, not a claim about a single ordinary response. Its own scaling claim is narrower than its reputation. The abstract says "verification scales more effectively with increased data than a finetuning baseline" — increased data, that is, not increased inference budget, which is what the paper is usually cited for.

The same paper, in a section headed "Test Time Compute", found the turning point three years before the technique became a product category:

At this scale, performance improves as we increase the number of completions up to 400. Beyond this point, performance start to decrease. This suggests that the benefits of search are eventually outweighed by the risk of finding adversarial solutions that fool the verifier.

Grading the working rather than the answer

A verifier marks the finished solution. The alternative is to mark each step of it. Hunter Lightman and colleagues at OpenAI framed the choice in May 2023:

To train more reliable models, we can turn either to outcome supervision, which provides feedback for a final result, or process supervision, which provides feedback for each intermediate reasoning step.

They collected PRM800K, "the complete dataset of 800,000 step-level human feedback labels used to train our best reward model", and reported using the step-trained reward model "to solve 78.2% of problems from a representative subset of the MATH test set". The subset is defined in the paper's appendix: 4,500 of the benchmark's test problems were moved into training, "We therefore evaluate our models only on the remaining 500 held-out problems." And the figure is a best-of-1,860 result — for each problem the generator produced 1,860 candidate solutions and the reward model chose one.

Figure 2. Three ways of choosing among the same 1,860 candidate solutions (Lightman et al., 2023)



Source: Lightman et al., "Let's Verify Step by Step", arXiv 2305.20050, 31 May 2023, Figure 3 and accompanying text. Measured on 500 problems held out of the MATH test set after 4,500 of its test problems were moved into training. The generator produced 1,860 candidate solutions per problem; the figures are the share for which the stated selection rule picked a correct one. The paper states that the two reward models' training sets "are not directly comparable" — the process-supervised set was built by active learning, is biased towards answer-incorrect solutions, and is an order of magnitude smaller.

▼ Table view

Figure 2. Three ways of choosing among the same 1,860 candidate solutions (Lightman et al., 2023)

How the answer was chosen	Problems solved, best-of-1,860
Process-supervised reward model	78.2%
Outcome-supervised reward model	72.4%
Majority vote among the candidates	69.6%

The gap is real and the conditions are unusually generous: a large candidate pool, a domain with a checkable answer, and a task the authors decline to generalise from. Their own sentence on scope is the one to keep: "It is unknown how broadly these results will generalize beyond the domain of math", which the same sentence follows by calling work in other domains important future work.

What the evidence does not support

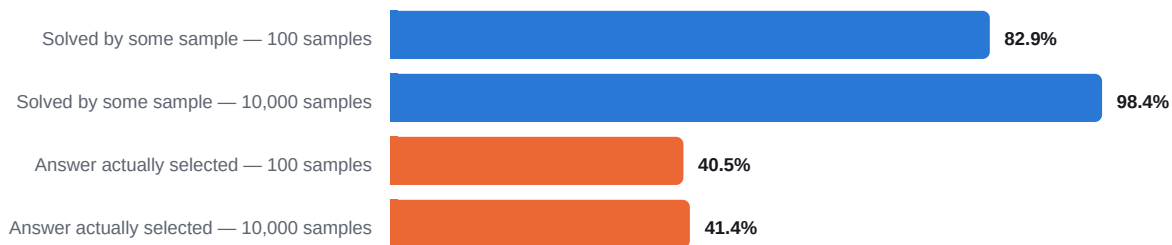
Three findings cut against the tidy version of the story, and all three come from inside the literature that built it.

The first is that process supervision does not straightforwardly beat outcome supervision. Jonathan Uesato and colleagues at DeepMind ran the comparison on GSM8K in November 2022 and found that "pure outcome-based supervision produces similar final-answer error rates with less label supervision." The row usually cited — 3.4% reasoning-step error and 12.7% final-answer error — is the *outcome* reward-model

configuration; the matching process configuration is 3.8% and 12.9%, and the paper prints min-max ranges of 0.0–6.8 against 0.5–7.1 beside them, noting in the same caption that "there is significant noise in the trace error rates". What the paper actually reports is subtler and more interesting than a contest between the two: reward models trained only on final answers ended up producing "predictions that agree more closely with the process-based labels ... than they do with the outcome-based labels themselves", which the authors hedge as an effect that "may be dataset-specific". The later OpenAI result is not a rematch either — its own introduction lists three simultaneous differences from the DeepMind work (a more capable base model, far more human feedback, a harder dataset), so the two studies do not isolate the variable between them. And in January 2025 the laboratory behind the most copied open reasoning model put process reward models in a section titled "Unsuccessful Attempts", reporting that such a model's "advantages are limited compared to the additional computational overhead it introduces during the large-scale reinforcement learning process in our experiments" — prefaced by its own caution that "this does not imply that these approaches are incapable of developing effective reasoning models."

The second is that generating more attempts stops paying long before the attempts stop being useful. Bradley Brown and colleagues separated two quantities in July 2024: coverage, the share of problems solved by *any* sample, and the accuracy of the answer actually delivered.

Figure 3. What 100× more sampling buys, with and without a way to check the answer (Brown et al., 2024)



Source: Brown et al., "Large Language Monkeys", arXiv 2407.21787, first posted 31 July 2024 (version 3 of 30 December 2024 read). The upper pair is coverage: at least one of the generated samples was correct. The lower pair is the accuracy of the answer chosen by majority voting or a reward model. The paper's own summary of the gap: in domains without automatic verifiers, such selectors "plateau beyond several hundred samples and fail to fully scale with the sample budget." Where an automatic checker exists — code that runs, a proof that verifies — the same technique converts: on SWE-bench Lite the paper reports 15.9% with one sample rising to 56% with 250.

▼ Table view

Figure 3. What 100× more sampling buys, with and without a way to check the answer (Brown et al., 2024)

Measure, on MATH with Llama-3-8B-Instruct	Share of problems
Solved by some sample — 100 samples	82.9%
Solved by some sample — 10,000 samples	98.4%
Answer actually selected — 100 samples	40.5%
Answer actually selected — 10,000 samples	41.4%

A hundredfold increase in sampling moved delivered accuracy by 0.91 percentage points. The rest of the improvement was correct answers the system generated and then failed to recognise. That single comparison explains why verifiers matter more than sampling does, and why the technique works best exactly where correctness is cheap to check.

The third is that the benefit is concentrated on problems of middling difficulty. Charlie Snell and colleagues, whose August 2024 paper is the standard argument that inference compute can substitute for model size, put the limit in their own results: "with the most challenging questions, we observe very little benefits from scaling up test-time compute. Instead, we find that on these questions, it is more effective to make progress by applying additional pretraining compute, demonstrating that current approaches to scaling test-time compute may not be 1-to-1 exchangeable with scaling pretraining." Their favourable comparison against a model roughly fourteen times larger is matched on modelled training-plus-inference operations rather than on serving cost, holds on mathematics, and depends on a difficulty estimate whose own cost of 2,048 samples per question the paper says it does not charge for. Apple researchers reported a related shape in June 2025 across four controllable puzzle families, identifying "low-complexity tasks where standard models surprisingly outperform LRMs", medium-complexity tasks where extra reasoning helps, and high-complexity tasks where both collapse. A published comment showed that parts of that study's setup were flawed — some puzzle instances were mathematically unsolvable yet scored as failures, and output-token limits were reached and counted as reasoning failures — while conceding that the authors' findings about context limits and evaluation design "are valuable engineering insights", and labelling its own replacement experiment preliminary. The low-complexity regime, where reasoning models do worse than ordinary ones, is not among the parts disputed.

4. How the thinking moved inside the model

For roughly three years the techniques above were applied to a model from outside it: a prompt someone wrote, a sampler someone ran, a ranking step someone bolted on. Scratchpads and verifiers arrived in 2021, prompted reasoning steps and majority voting in 2022, step-level grading in 2022 and 2023. The model itself was unchanged.

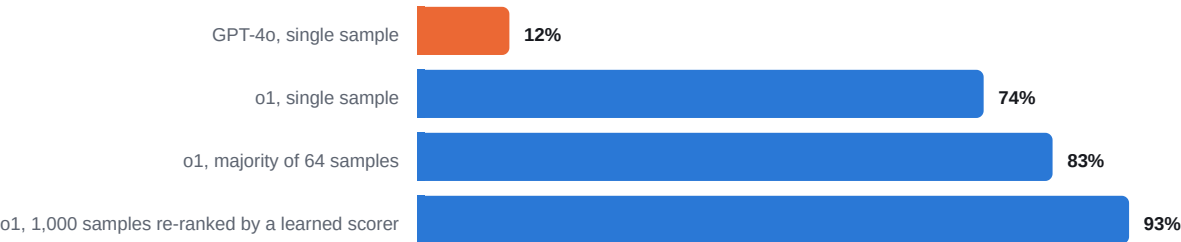
The precedent for moving it inside came from board games. The AlphaGo paper of January 2016 reports that "Without any lookahead search, the neural networks play Go at the level of state-of-the-art Monte Carlo tree search programs that simulate thousands of random games of self-play", and that adding search on top produced a program that "defeated the human European Go champion by 5 games to 0". Instinct and deliberation were separable components, and the deliberation was worth a great deal on its own.

September 2024 is when the equivalent arrived for language. OpenAI released a model trained with reinforcement learning to produce its own working:

Our large-scale reinforcement learning algorithm teaches the model how to think productively using its chain of thought in a highly data-efficient training process. We have found that the performance of o1 consistently improves with more reinforcement learning (train-time compute) and with more time spent thinking (test-time compute). The constraints on scaling this approach differ substantially from those of LLM pretraining, and we are continuing to investigate them.

The two charts that accompanied that claim, plotting accuracy against train-time and test-time compute on logarithmic axes, carry no numbers on either compute axis. They are the images most often reproduced from that post, and they establish a direction rather than a rate. The same post's prose is more useful, because its numbers separate three different ways of spending at inference time on a single exam.

Figure 4. One exam, three ways of spending computation at question time (OpenAI, September 2024)



Source: OpenAI, "Learning to Reason with LLMs", 12 September 2024, read from the Internet Archive capture of 13 September 2024. The post gives these as averages over a 15-problem exam: 1.8/15, 11.1/15, 12.5/15 and 13.9/15. The post also states that "Unless otherwise specified, we evaluated o1 on the maximal test-time compute setting", so the single-sample figure is itself a maximum-effort result. The appendix reports 74.4% pass@1 and 83.3% consensus@64 for the same model; the 93% figure appears only in the body text.

▼ Table view

Figure 4. One exam, three ways of spending computation at question time (OpenAI, September 2024)

Model and inference procedure, AIME 2024	Problems solved
GPT-4o, single sample	12%
o1, single sample	74%
o1, majority of 64 samples	83%
o1, 1,000 samples re-ranked by a learned scorer	93%

The last of those three is the 2021 verifier idea at scale. The first is the part that was new: a model that produces the working itself, because it was trained to.

Four months later DeepSeek published the training curve that made the mechanism legible. Over the course of reinforcement learning against checkable answers, its model's responses grew longer on their own — the paper's figure caption reads "DeepSeek-R1-Zero naturally learns to solve reasoning tasks with more thinking time", and the body adds that the increase "is not the result of external adjustments but rather an intrinsic development within the model." No length was specified by anyone, and nothing in the reward rewarded length: the paper's rule-based reward system "mainly consists of two types of rewards", one for a correct final answer and one for placing the working between the right tags. The paper's revision of 4

January 2026 adds the cost of that to its limitations: the model "uses fewer tokens to solve simple tasks, while generating more tokens for complex tasks", but "instances of excessive reasoning—manifested as overthinking—are still observed in response to simpler questions."

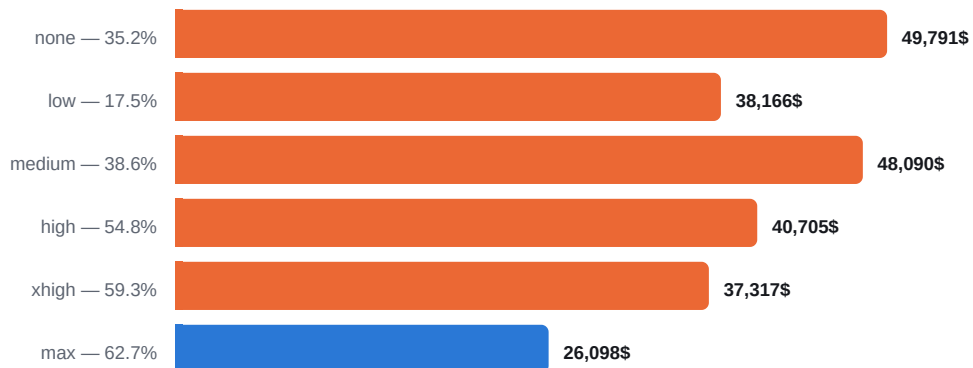
5. What the 2026 measurements show

The useful evidence on what thinking buys comes from organisations that do not sell the model. Two measurements of the same OpenAI model, one published on 3 September 2026 and one read on 21 September, agree on the shape and disagree with the intuition.

The ARC Prize Foundation, which owns the benchmark rather than the model, ran GPT-6 Astra on its interactive puzzle set at six effort settings on 3 September 2026. Its summary sentence inverts the obvious economics:

Higher reasoning levels generally cost less because Astra solves games in fewer actions, reducing the total number of model calls and tokens.

Figure 5. Total cost of the ARC-AGI-3 evaluation at six reasoning-effort settings (ARC Prize, 3 September 2026)



Source: ARC Prize Foundation (Greg Kamradt), "OpenAI's GPT-6 Astra on ARC-AGI-3", 3 September 2026, standard-harness column. Costs are totals for the evaluation, not per game. The provider-adapter harness, which preserves the model's own reasoning state between requests, scored 96.7–99.9% at every setting and is not a like-for-like comparison across vendors. No uncertainty analysis is published with these figures, so the ordering of adjacent settings cannot be separated from run-to-run variation.

▼ Table view

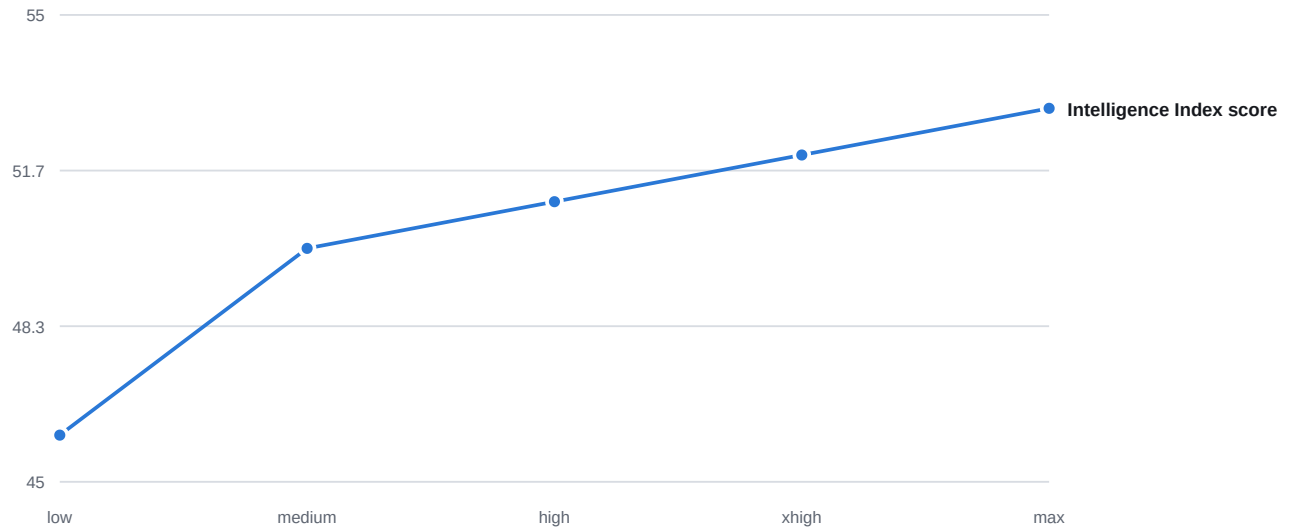
Figure 5. Total cost of the ARC-AGI-3 evaluation at six reasoning-effort settings (ARC Prize, 3 September 2026)

Reasoning effort, and the score it reached	Total cost of the run, US\$
none — 35.2%	49,791\$
low — 17.5%	38,166\$
medium — 38.6%	48,090\$
high — 54.8%	40,705\$
xhigh — 59.3%	37,317\$
max — 62.7%	26,098\$

Two things are visible in that table and only one of them is comfortable. The comfortable one is that spending more per decision reduced the total bill, because the model needed fewer interactions with the environment to finish — "generally", in the foundation's own word, and the qualifier earns its place, because neither column runs in order. On cost, `low` at \$38,166 is cheaper than `medium` at \$48,090 and cheaper than `high` at \$40,705. On score, `low` reached 17.5% while `none` reached 35.2%, a gap of nearly eighteen points in the wrong direction. Neither column rises monotonically with effort, and because the foundation publishes no confidence intervals for these runs, how much of that is run-to-run variation cannot be told from the table; the comparison it draws itself is max against medium.

Artificial Analysis measured the same model across five effort settings on its own index, with reasoning tokens broken out from total output. The result is an unusually detailed published breakdown of diminishing returns.

Figure 6. What each effort setting scored, and what it consumed (Artificial Analysis Intelligence Index v4.3.2)



Source: Artificial Analysis, per-effort comparison of GPT-6 Astra, Intelligence Index v4.3.2, read 21 September 2026. Across the same five settings: reasoning tokens per task 938 / 3k / 5k / 9k / 17k; output tokens per task 4k / 10k / 12k / 17k / 27k; cost per task \$0.82 / \$1.54 / \$1.73 / \$2.31 / \$3.26; time to first answer token 2.55s / 5.62s / 33.31s / 126.92s / 256.13s. Three of the index's ten constituent evaluations do not print their best figure at maximum effort: Terminal-Bench 4.0 (60% at xhigh against 59% at max), GDP.pdf (32% against 31%) and AA-Omniscience (44 at high against 43). The page publishes no confidence intervals, so those three are differences in printed figures rather than demonstrated regressions.

▼ Table view

Figure 6. What each effort setting scored, and what it consumed (Artificial Analysis Intelligence Index v4.3.2)

Reasoning effort	Intelligence Index score
low	46
medium	50
high	51
xhigh	52
max	53

The arithmetic on those published figures is unflattering to any simple story. Roughly eighteen times the reasoning tokens, four times the money and about a hundred times the wait before the first visible word buy seven points of index score. The largest single gain — four points — comes from the first step off the bottom rung, and the last step buys one. A reader deciding what to pay for is better served by that asymmetry than by the headline.

Figure 7. The price of the top rung, on one independent index



Source: Artificial Analysis per-effort comparison of GPT-6 Astra, Intelligence Index v4.3.2, read 21 September 2026. Ratios are arithmetic on the figures that page publishes; Artificial Analysis does not present them in this form.

▼ Table view

Figure 7. The price of the top rung, on one independent index

Measure	Value
more reasoning tokens	18×
index points	+7
cost per task	4.0×
to the first answer token	256s

The same evaluator's article of 9 September 2026 adds a cross-model figure that complicates any attempt to read verbosity as intelligence: at maximum effort the model "uses 27k output tokens per task, about a third of Claude Fable 5.1 (max with fallback) at 78k, for the same score", matching that rival's index score "at ~40% of the cost per task (\$3.26 vs \$7.63)". More thinking and more writing came apart in 2026. The same article records the other direction in the same paragraph set: on one benchmark adapted from OpenAI's own dataset the model scored about 45 Elo points *below* its predecessor, while "using significantly fewer turns than other models ... 24 per task at max effort, compared to 45 for GPT-5.6 Sol". Frugality is not free, and neither the evaluator nor OpenAI says which way the causation runs.

Every effort setting is the vendor's own label for its own internal procedure, and the per-effort results behind OpenAI's own note that "Evaluation scores are the maximum at any effort" stay unpublished, so what ARC Prize and Artificial Analysis measure is what came out rather than what happened inside.

6. Two directions at once

Within a fortnight this month the two largest commercial vendors moved in opposite directions, and both movements are documented in their own release notes.

On 1 September 2026 Anthropic shipped two models on which thinking cannot be disabled at all. On 14 September OpenAI published this in its ChatGPT release notes:

We're retiring automatic switching from Instant to Thinking (reasoning) for ChatGPT Plus and Pro users globally. You can still select an available option in the model picker to give ChatGPT more time to think or reason.

The same entry removes the "Higher intelligence" setting from the web client for those plans, and notes that "ChatGPT can still switch automatically for safety purposes." Read together, the developer surface is converging on always-think while the consumer surface is handing the decision back to the person typing. A useful lens here is that the two surfaces face different failure modes: an application developer who cannot predict when a model will spend four minutes on a request has a latency problem, and a subscriber who is silently routed into a slower mode has a different one. Nothing in either document explains the divergence, and the two decisions were taken by different companies about different products.

There is a third posture, and it belongs to the open-weight models. The published card for GLM-5.3 documents a reasoning-effort parameter with the levels `low`, `high` and `max`, and adds that "It defaults to `max` if not passed (or if set to any other value)" — no rung below `low` exists at all. The card for DeepSeek-V4.1-Flash, dated 10 September 2026, documents "a continuously controllable reasoning effort from 1 to 100": a dial whose floor is one, not zero.

7. What to watch

Three checks, each against a named page.

Anthropic's per-model thinking table, at platform.claude.com. On 21 September 2026 it lists five models as `Always on` and two as `On`, the opt-out on one of those two (Opus 5) restricted to effort `high` or below, and the three most recent releases arrived on 30 June, 24 July and 1 September. When the next model appears, the column its row lands in is the whole answer: a new model arriving as `on` would be a counter-example to the trend, and the disappearance of the `off` rows would harden it.

OpenAI's API changelog, at the next flagship release. The entry for 3 September 2026 states that GPT-6 Astra does not support the `none` reasoning effort level, while every GPT-5.6 model still lists `none` first on the models index. One line of the next release entry says whether the bottom rung returns or spreads.

An unexplained discrepancy that is open today. OpenAI's reasoning guide states that GPT-6 Astra returns HTTP 400 for `reasoning.effort: none`. ARC Prize's table of 3 September 2026 reports a `none` row for the same model, with a score of 35.2% and a cost of \$49,791. Both documents were read on 21 September 2026 and both say what they say. A contract that changed between 3 and 21 September would explain it; so would evaluation access that differs from the public API; neither company has said which. Whichever page moves first resolves it, and until one does, the two facts sit side by side without an inference between them.

8. The idea to keep

Test-time compute is the work a system does on a question after training ends, and for a language model that work is text. There is no separate thinking organ, no quiet interior: every token of reasoning is a token the model generated, which is why the effort dial and the bill are the same dial.

That is not a figure of speech. OpenAI's reasoning guide states:

While reasoning tokens are not visible via the API, they still occupy space in the model's context window and are billed as output tokens.

The same page warns that a request which exhausts its output allowance during reasoning can "incur costs for input and reasoning tokens without receiving a visible response". Anthropic's documentation is blunter still: "You are billed for the full thinking process, not the thinking content visible in the response", and, of the setting that suppresses the summary, "Omitting reduces latency, not cost." Google's price list carries the point in a row label: "Output price (including thinking tokens)".

So any claim about a reasoning model raises two questions rather than one. What did the extra thinking buy on this task — and what did it cost, in money, in latency, and in tokens nobody reads? The published evidence answers both differently depending on whether the task has something to check against. Where correctness is cheap to verify — code that runs, a proof that checks, a puzzle with a rule — extra attempts convert into accuracy. Where it is not, they mostly convert into tokens. Shannon's distinction survives the change of substrate: the gain was never in thinking about everything for longer, but in choosing what to think about and knowing where to stop.