UNDERSTANDING AI · WRITTEN EDITION · 2026-09-09
Day 5 · typeset 2026-09-10 22:55Z

# How an Answer Unfolds

*A machine asked the same question a thousand times, with greedy decoding requested, returned eighty distinct completions. What decides which word comes next, and how much of the past the decision may consult, are two settings — and both are somebody's decision rather than a fact about the machine.*

In September 2025 a laboratory called Thinking Machines published an experiment of unusual patience. It took an open-weight model built by somebody else — Qwen3-235B-A22B-Instruct-2507 — gave it one prompt, *"Tell me about Richard Feynman"*, set the temperature to zero, and asked for a thousand completions of a thousand tokens each. Temperature zero is the setting that is supposed to mean *return the single best answer, the same one, every time*.

In its own words: "we generate 80 unique completions, with the most common of these occuring 78 times." More than nine runs in ten produced something other than the commonest reply. All one thousand agreed for the first 102 tokens, every one of them writing *"Feynman was born on May 11, 1918, in"* — and then 992 continued "Queens, New York" while 8 continued "New York City".

A smaller local experiment, run on 8 September 2026 on a 4-billion-parameter open-weight model, found a related effect: changing the batch size changed a greedy continuation of the same raw text prefix. It is not a replication — the model, the serving software, the output length and the request protocol all differ — but it isolates one variable in a way the published run did not.

**One prompt, greedy decoding, one desktop machine**

**5**
distinct completions across seven batch sizes
every row a byte-identical copy of the prompt

**1 of 5**
distinct completions when the batch size is held fixed
at every one of five sizes tested

**token 2**
where most differing conditions first parted
"life and work" against "contributions to physics"

*Qwen3.5-4B, bfloat16, sdpa attention, do_sample=False, 120 new tokens, raw text prefix "Tell me about Richard Feynman" with no chat template. Seven batch sizes of identical copies produced five distinct completions; two further batches sharing the card with unrelated prompts brought the total to six. Two of the seven matched the batch-of-one run throughout, and one of the mixed batches first differed at generated token 68 rather than token 2. No padding occurs in the identical-copy condition, so the attention mask is unchanged and the only variable is how many copies are computed at once. Measured on one desktop machine with an RTX 3090, 8 September 2026, torch 2.14.0, transformers 5.16.1. The scripts and raw JSON are kept with the episode's research record and are not published.*

▼ **Table view**

One prompt, greedy decoding, one desktop machine

| Measure | Value |
|---|---|
| distinct completions across seven batch sizes | 5 |
| distinct completions when the batch size is held fixed | 1 of 5 |
| where most differing conditions first parted | token 2 |

The control is what makes the result mean anything. Held at a fixed batch size and repeated five times, the answer was identical five times out of five — at batch sizes 1, 2, 8, 32 and 64 alike. Every row inside a single batch of 64 identical prompts agreed with every other row. Change only the number of copies being computed alongside it, and the answer changes.

## Two readings that do not survive contact with the documentation

The first is that there is a dial marked randomness and that turning it to zero makes a system repeatable. Anthropic's own API reference, read on 8 September 2026, says otherwise on the page that documents the dial:

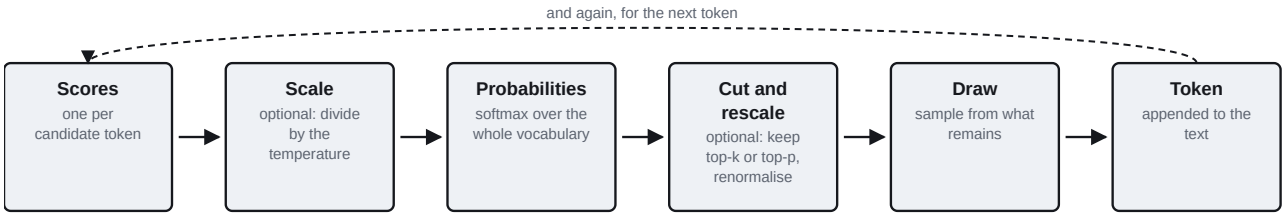
> Note that even with `temperature` of `0.0`, the results will not be fully deterministic.

The second is shorter: *"it forgot my conversation"*. That description leaves the mechanism open, and the mechanism is the interesting part. Earlier material may no longer be supplied to the model, may survive only as a summary of itself, or may be supplied and not used well — three different things, with different remedies. What does not happen is the one the phrase implies: no step inside a language model inspects a conversation, judges it too long, and discards the start of it. The choice of which end to discard is made outside, and by somebody.

# From a score to a word

A model does not emit words. It produces a score for every candidate token in its vocabulary — 248,320 of them on the model measured here — and decoding is what turns that list into one choice.

### What happens after the scores




*Schematic of decoding as Jurafsky and Martin set it out in section 7.6 of the 19 August 2026 draft. The two scaling steps are optional and configurable. Greedy decoding bypasses the draw entirely and takes the highest-scoring candidate, which is also what top-k reduces to when k is one. The returning arrow is the next step of generation, not a loop within one step.*

▼ Table view

What happens after the scores — stages

| # | Stage | Note |
|---|---|---|
| 1 | Scores | one per candidate token |
| 2 | Scale | optional: divide by the temperature |
| 3 | Probabilities | softmax over the whole vocabulary |
| 4 | Cut and rescale | optional: keep top-k or top-p, renormalise |
| 5 | Draw | sample from what remains |
| 6 | Token | appended to the text |

What happens after the scores — connections

| From | To | Label |
|---|---|---|
| Scores | Scale | |
| Scale | Probabilities | |
| Probabilities | Cut and rescale | |
| Cut and rescale | Draw | |
| Draw | Token | |
| Token | Scores | and again, for the next token |

The simplest rule takes the highest score every time. Jurafsky and Martin, whose third-edition draft of *Speech and Language Processing* was released on 19 August 2026, are precise about what that buys:

> Indeed, greedy decoding is so predictable that it is deterministic; if the context is identical, and the probabilistic model is the same, greedy decoding will always result in generating exactly the same string.

Two conditions sit in the middle of that sentence — identical context, identical model — and both held in the Feynman experiment, which nonetheless returned eighty different strings. The textbook is describing an algorithm; the laboratory was running an implementation of one, and the distance between those two things is where the eighty completions come from.

The textbook says greedy decoding is not, in practice, used with large language models, and says why:

> In other words, greedy decoding is too boring, and random sampling is too random.

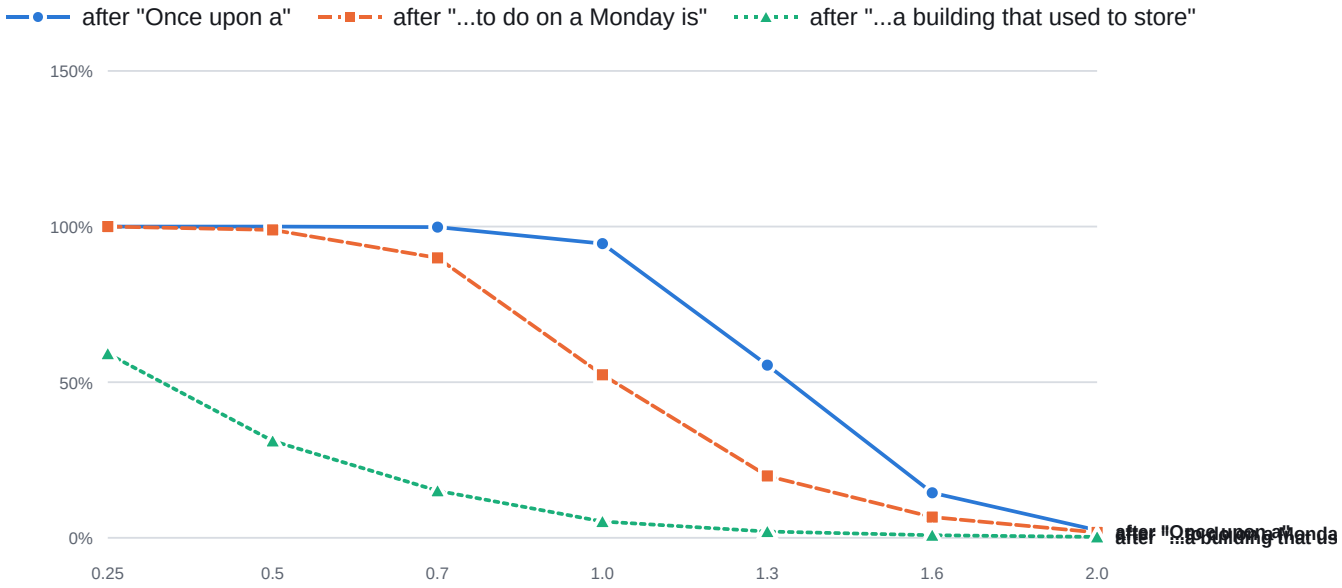
Too boring, because tokens chosen for being extremely predictable produce text the textbook calls "generic and often quite repetitive". Too random, because the obvious alternative — turning the scores into probabilities and drawing one — reaches into a very long tail of unlikely candidates which, collectively, carry enough probability to be selected often enough to wreck a sentence.

Practical decoding sits between the two, by two routes.

**Reshaping.** Dividing the scores by a constant before they are turned into probabilities sharpens or flattens the distribution. The textbook traces the idea, not the word, to physics: the intuition for temperature sampling, it says, "comes from thermodynamics, where a system at a high temperature is very flexible and can explore many possible states, while a system at a lower temperature is likely to explore a subset of lower energy (better) states." The parenthesis is the textbook's.

What that dial does, however, depends entirely on the distribution it is applied to.

**Probability on the single likeliest token, as the temperature rises**




*Qwen3.5-4B, measured on one desktop machine, 8 September 2026. Nothing is sampled: this is the arithmetic handed to the sampler. The same step of the dial does very different things to the three contexts — from 1.0 to 1.3 the first line falls 39 points and the third falls 3.*

▼ **Table view**

Probability on the single likeliest token, as the temperature rises

| | after "Once upon a" | after "...to do on a Monday is" | after "...a building that used to store" |
|---|---|---|---|
| 0.25 | 100% | 100% | 59.2% |
| 0.5 | 100% | 98.9% | 31.2% |
| 0.7 | 99.8% | 90.0% | 15.1% |
| 1.0 | 94.5% | 52.4% | 5.3% |
| 1.3 | 55.5% | 19.9% | 2.0% |
| 1.6 | 14.5% | 6.6% | 0.9% |
| 2.0 | 2.4% | 1.7% | 0.3% |

Counted as candidates rather than as probability, the same step is more dramatic still. At temperature 1.0 the context "Once upon a" has a single candidate holding ninety per cent of the probability; at 1.3 it has 23,865. The context ending "used to store" moves from 939 candidates to 5,212 over the same step.

**Cutting.** In May 2018 Angela Fan, Mike Lewis and Yann Dauphin, generating fiction from writing prompts at Facebook AI Research, drew from only the ten likeliest candidates, and gave their reason in the paper: "Completely random sampling can introduce very unlikely words, which can damage generation as the model has not seen such mistakes at training time." In the version published at ICLR 2020, Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes and Yejin Choi named the defect in a fixed count, and the argument runs both ways: with a small k "there is a risk of generating bland or generic text", while with a large k "the top-k

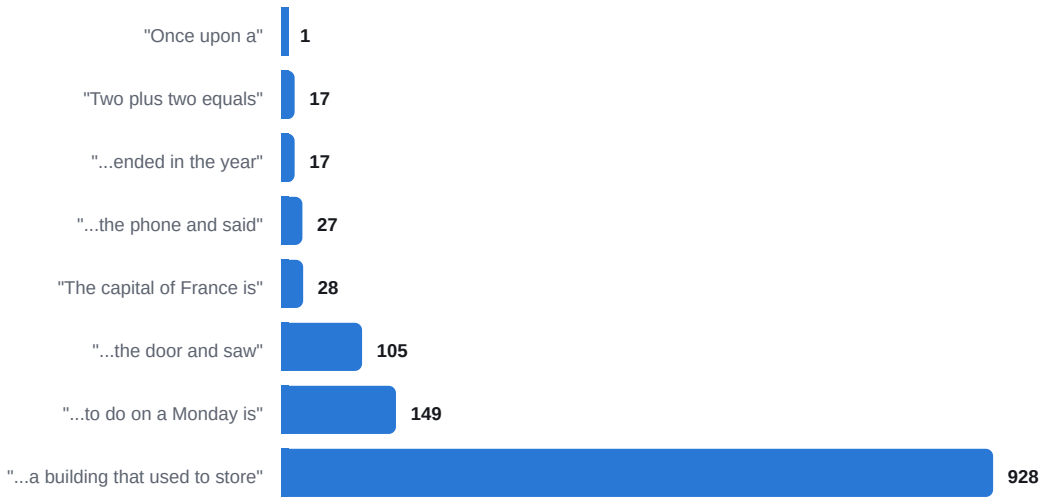
vocabulary will include inappropriate candidates which will have their probability of being sampled increased by the renormalization." Their alternative keeps a fixed *share* of the probability mass rather than a fixed number of candidates, so that the number of candidates rises and falls by itself.

Jurafsky and Martin put the same objection in one line: "One problem with top-k sampling is that k is fixed, but the shape of the probability distribution over words differs in different contexts." Their description of the operation itself is that "we first truncate the distribution to the top k most likely words", renormalise, and sample from what is left.

## How much the shape actually differs

Define the *nucleus* at a position as the smallest number of candidate tokens whose probabilities, sorted from largest, sum to at least ninety per cent. Two passages were run through Qwen3.5-4B on 8 September 2026: 514 words of prose written by hand that morning for the measurement — 621 tokens as this model counts them — and the first chapter of *Pride and Prejudice*, which is in the public domain and long predates any of these models.

### Candidates needed to cover ninety per cent of the probability




*Qwen3.5-4B at temperature 1.0, eager attention, measured on one desktop machine, 8 September 2026. Against the p=0.9 nucleus, a fixed cut at ten candidates admits nine more than the nucleus in the first row and excludes 918 of its members in the last. The last row also appears as 939 in the temperature figure above, which was computed with a different attention implementation; the two are not readings of one identical numerical distribution.*

▼ **Table view**

Candidates needed to cover ninety per cent of the probability

| The text so far | Candidates |
|---|---|
| "Once upon a" | 1 |
| "Two plus two equals" | 17 |
| "...ended in the year" | 17 |
| "...the phone and said" | 27 |
| "The capital of France is" | 28 |
| "...the door and saw" | 105 |
| "...to do on a Monday is" | 149 |
| "...a building that used to store" | 928 |

Across a whole passage the spread is wider than any of those eight positions suggests.

| Corpus | Positions | Smallest | Median | 90th percentile | Largest |
|---|---|---|---|---|---|
| Prose written 8 Sep 2026, unpublished | 621 | 1 | **14** | 283 | **5,570** |
| *Pride and Prejudice*, chapter 1 | 1,236 | 1 | **1** | 3 | 5,325 |

*Nucleus size at p = 0.9, Qwen3.5-4B, temperature 1.0, no truncation applied before counting. The same measurement on GPT-2 (2019) gives a median of 154 on the unpublished prose and 67 on Austen.*

Two findings sit in that table. The first is the one the 2019 paper predicted: within a single passage the number of tokens needed to cover ninety per cent of the model's probability ranges from one to 5,570, so a fixed cut at ten keeps a very different share of that probability depending on where in the text it falls.

The second was not looked for. On the Austen passage the median nucleus size is **one** — a single token carrying ninety per cent of the probability at more than half of all positions; on the newly prepared passage it is fourteen. The model's probability is an order of magnitude more concentrated on the passage it has almost certainly read than on the passage nobody has; whether that is memorisation, or Austen's prose being more predictable than a technical paragraph written this week, is not separable from a measurement of this shape.

## The bound, and where it came from

Every one of those draws is made from a fixed quantity of preceding material, which the field calls the context window.

The two halves of the problem — how a token is chosen, and how much text the choice may consult — arrive in the same citation. Jurafsky and Martin, in section 7.6.2: "Sampling from language models was first suggested very early on by Shannon (1948) and Miller and Selfridge (1950)."

Shannon's half is the better known: his 1948 paper builds a ladder of approximations to English by drawing successive letters and words from a real text.

The second citation is a psychology paper. George A. Miller and Jennifer A. Selfridge published *Verbal context and the recall of meaningful material* in the *American Journal of Psychology* 63(2), April 1950, pages 176–185. The National Library of Medicine indexes it under Humans, Learning, Memory and Mental Recall: an experiment on people, testing how the contextual structure of word sequences affected what they could afterwards recall. Its opening argument is that communicative behaviour "depends upon patterning for its significance and usefulness". The textbook's account of where sampling from language models came from therefore rests on one 1948 information-theory paper and one 1950 study of human memory.

One word is worth flagging before it changes meaning. When the tail of a probability distribution is discarded before a draw, the textbook's verb is *truncate*: top-k "truncates the distribution to the top k most likely words", top-p exists "to truncate the distribution to remove the very unlikely words". The same verb is used, in the same vendors' documentation, for discarding earlier material from a conversation. They are unrelated operations — one runs at every step and is performed by the decoding rule, the other removes earlier material as needed, potentially many times over a long conversation, and is performed by software above the model.

## The million, dated

The context-window number that everyone knows did not arrive in 2026. The 2024 Google and 2025 OpenAI rows below were read from archived captures taken the day of publication rather than from today's versions of those pages; the Anthropic rows come from release notes that are dated individually but were read in September 2026.

| Date | Actor | What was said, in the source's own scope |
|---|---|---|
| **15 Feb 2024** | Google | Gemini 1.5 Pro: "running up to 1 million tokens consistently, achieving the longest context window of any large-scale foundation model yet". Standard window that day: **128,000**. The million went to "a limited group of developers and enterprise customers" in "private preview", labelled "experimental", with a warning to "expect longer latency times" |
| 14 May 2024 | Google | 1.5 Pro to 2M via waitlist |
| **14 Apr 2025** | OpenAI | GPT-4.1 family at 1M, "up from 128,000 for previous GPT-4o models"; "We trained GPT-4.1 to reliably attend to information across the full 1 million context length" |
| **12 Aug 2025** | Anthropic | 1M in **public beta** for Claude Sonnet 4 |
| 5 Feb 2026 | Anthropic | 1M beta extends to Opus 4.6; server-side compaction launches, also in beta |
| **13 Mar 2026** | Anthropic | 1M **leaves beta** for Opus 4.6 and Sonnet 4.6, at standard pricing |
| **30 Apr 2026** | Anthropic | 1M beta **retired** for Sonnet 4 and Sonnet 4.5: "requests exceeding the standard 200k-token context window return an error" |
| 30 Jun 2026 | Anthropic | Sonnet 5 ships a tokenizer producing "approximately 30% more tokens for the same text" |
| **2 Sep 2026** | Google | Gemini 3.8 Flash announced. The post does not mention the context window at all |

*Google and OpenAI rows read from Internet Archive captures of 15 Feb 2024 15:13:50 UTC and 15 Apr 2025 01:36:10 UTC respectively. Anthropic rows from the platform release notes, 137 dated entries, read 8 September 2026.*

Two things in that sequence deserve more weight than the headline number.

**A window can be withdrawn.** A model that accepted a million tokens on 29 April 2026 did not on 1 May; the release note records a service change and says nothing about what the model can do. The advertised window is in part a commercial commitment, and commitments can be revoked. What a model can in fact be run over is a different question, settled by its architecture and by the hardware available, and the two need separating before either number means anything.

**The unit under the number moved while the number stood still.** Anthropic's model documentation, read on 8 September 2026, states that a million tokens is "roughly 555k words or 2.5M Unicode characters on the current tokenizer (introduced with Claude Opus 4.7); models before it fit about 750k words in 1M tokens." The headline is unchanged, and on Anthropic's own approximate word equivalents the text it holds has fallen by about a quarter. "A million" is not even one number: Google documents 1,048,576, OpenAI documents 1,047,576 for GPT-4.1 and 1,050,000 for GPT-6 Astra, and the Anthropic pages checked here give the rounded figure "1M" and no integer.

Meanwhile the vendors' own numbers complicate their own claim. The GPT-4.1 launch page introduced its harder evaluations by conceding that "few real-world tasks are as straightforward as retrieving a single, obvious needle answer", and published this:

| Long-context evaluation | GPT-4.1 | GPT-4.1 mini | GPT-4.1 nano |
|---|---|---|---|
| OpenAI-MRCR, 2 needle, 128k | 57.2% | 47.2% | 36.6% |
| OpenAI-MRCR, 2 needle, 1M | 46.3% | 33.3% | 12.0% |
| **Graphwalks BFS, under 128k** | **61.7%** | 61.7% | 25.0% |
| **Graphwalks BFS, over 128k** | **19.0%** | 15.0% | 2.9% |
| Graphwalks parents, under 128k | 58.0% | 60.5% | 9.4% |
| Graphwalks parents, over 128k | 25.0% | 11.0% | 5.6% |

*From the 15 April 2025 archive capture. "Over 128k" is a bucket rather than a measurement at one million, Graphwalks is a synthetic breadth-first search over a graph of hexadecimal hashes rather than a direct measure of everyday performance, and the dashes elsewhere in the published version mark models whose 128K windows cannot be run at those lengths at all.*

Google's long-context guide, updated 22 June 2026, is blunter about the limits than its 2024 announcement was: where "you might have multiple 'needles'… the model does not perform with the same accuracy", and "generally, if you don't need tokens to be passed to the model, it is best to avoid passing them." Anthropic's context-window page carries the same warning on the page that sells the capacity: "more context isn't automatically better. As token count grows, accuracy and recall degrade, a phenomenon known as *context rot*."

## What happens when the text does not fit

The claim that a model quietly forgets the start of a conversation can be tested on the mechanism rather than argued about, because the mechanism is small enough to inspect.

GPT-2 stores 1,024 learned position vectors in a lookup table with one row per position. Given 1,023 tokens it runs; given 1,024 it runs; given 1,032 it raises `IndexError: index out of range in self`. The failure is a lookup with no row to fetch — not an eviction, not a summarisation, not a decision about what to keep. Qwen3.5-4B has no such table, because its position information is rotary and applied inside the attention layers; fed the same growing input it neither errors nor drops anything, and retrieved an exact planted string correctly at every tested length, up to 16,505 tokens.

The architecture adds a qualification that the policy does not. In several 2026 open-weight models most layers do not attend to the whole input: their configuration files designate a minority of layers full attention and the rest sliding-window or linear. That changes how much of the text each layer may consult; it does not remove any of it from the input, and no layer decides to discard the beginning of a conversation.

The refusal a caller actually meets is imposed at the API boundary, and Anthropic's documentation is explicit about it: "If the input alone already exceeds the model's context window, the API returns a 400 `invalid_request_error` ('prompt is too long') **on every model**." On Claude 4.5 models and newer, though, an input that fits but whose input plus `max_tokens` does not is *accepted*, and the answer is then cut off with `stop_reason: "model_context_window_exceeded"`. The input is refused; the request is not always.

Everything else is a policy, and the policies differ inside a single company.

| What happens | Where | Source, read 8 Sep 2026 |
|---|---|---|
| **The input is refused** | Anthropic, every model; OpenAI Responses with `truncation: disabled`, which is the default — "the request will fail with a 400 error" | Anthropic context-windows doc; `openai/openai-openapi` |
| Items dropped from **the beginning** | OpenAI Responses with `truncation: auto` — "dropping items from the beginning of the conversation"; OpenAI Realtime, where `auto` is the **default** | `openai/openai-openapi` |
| Messages dropped from **the middle** — *historical; the Assistants API was "officially sunset on August 26, 2026, and is no longer available"* | OpenAI Assistants, `truncation_strategy: auto` — "messages in the middle of the thread will be dropped" | `openai/openai-openapi`; Assistants migration guide, read 9 Sep 2026 |
| Old content **replaced by a summary** | Anthropic compaction (beta, opt-in): "the API automatically drops all content blocks prior to the `compaction` block". OpenAI compaction: an encrypted item | Anthropic compaction doc; OpenAI compaction guide |

A shorter stand-in is not a memory. Compaction replaces earlier material with a compressed representation of it — Anthropic drops the blocks before the compaction block; OpenAI documents an encrypted item carrying prior state and reasoning, and its returned window can retain other items too — and a representation can omit a constraint or preserve an error. OpenAI's own guide describes that item:

> It is opaque and not intended to be human-interpretable.

"It forgot" therefore names no actor. Where material was removed, something above the model removed it and somebody chose which end; in one of the four cases what stands in its place cannot be read by the person whose conversation it was.

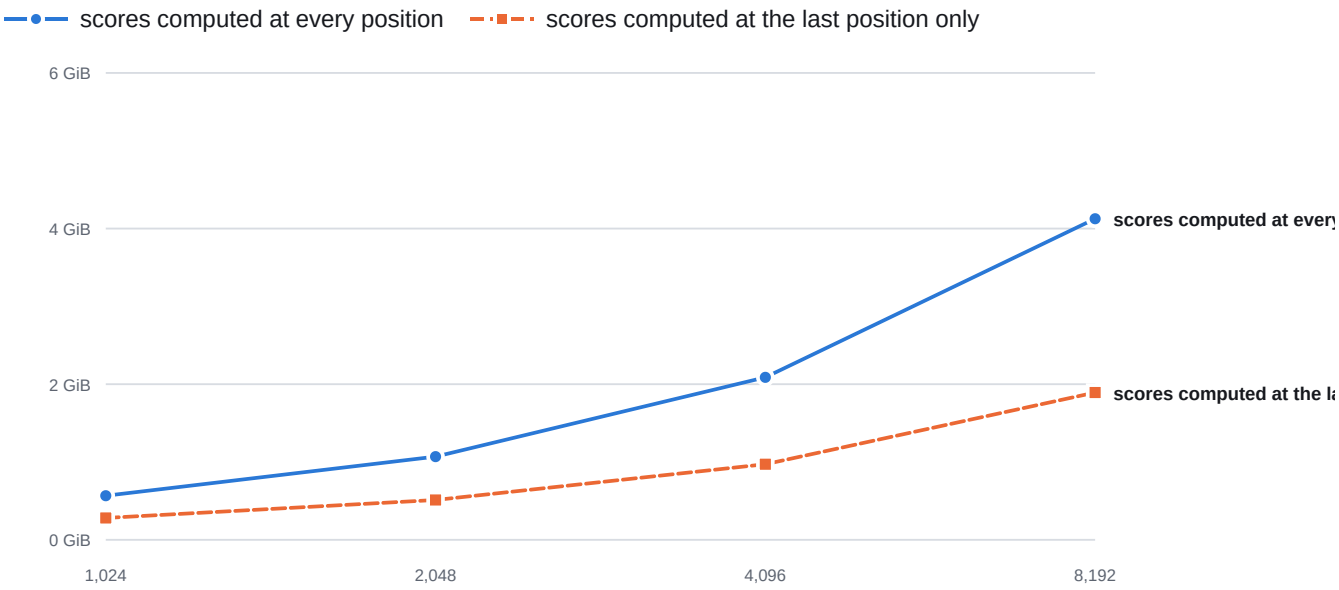
## What a window costs on one machine

A declared window is a number in a configuration file. Reaching it is bought in hardware, and the arithmetic is public.

Qwen3.5-4B designates 8 of its 32 layers full attention; only those keep a cache that grows with the text. With four key-value heads of dimension 256 in bfloat16, that is 32,768 bytes of cache per token of context, or 8.0 GiB at the model's declared 262,144-token window — against 32.0 GiB had every layer been full

attention. The weights themselves occupy 7.8 GiB.

### Memory a single forward pass needs, above the model's weights




*Qwen3.5-4B, bfloat16, measured on one desktop machine, 8 September 2026, on an RTX 3090 shared with other processes holding roughly 9.5 GiB throughout. At 12,288 tokens the run failed for memory with about 4.2 GiB free. The declared window of this model is 262,144 tokens.*

▼ Table view

Memory a single forward pass needs, above the model's weights

| | scores computed at every position | scores computed at the last position only |
|---|---|---|
| 1,024 | 0.6 GiB | 0.3 GiB |
| 2,048 | 1.1 GiB | 0.5 GiB |
| 4,096 | 2.1 GiB | 1.0 GiB |
| 8,192 | 4.1 GiB | 1.9 GiB |

The gap between the two lines is where a long prompt's cost actually sits. Most of what a long prompt costs here is not the cache the phrase "context window" evokes: it is the score list itself. Asking for a score on every candidate token at *every* position — 248,320 numbers per position — more than doubles the memory needed at 8,192 tokens, and the 2.2 GiB difference is the same order as the 3.8 GiB that score tensor would occupy alone. Serving software can avoid computing scores at every prompt position, which is the cheaper line.

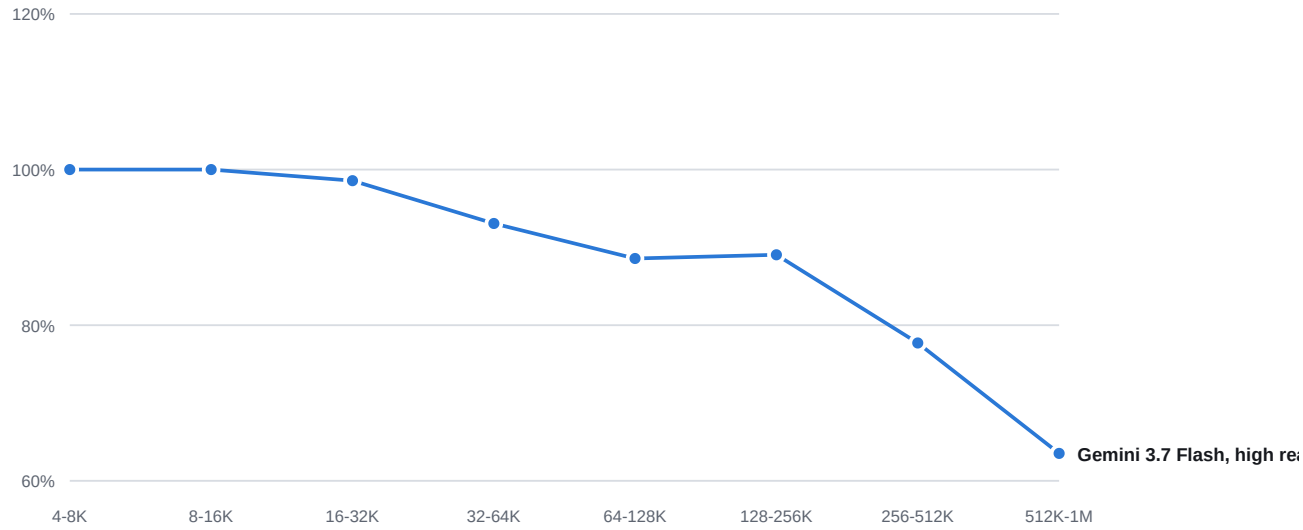
Independent measurement of what a big window is worth is scarcer than the claims about it. HELMET, a vendor-independent long-context benchmark, describes its own reach as "controllable lengths up to 128K tokens" — an order of magnitude below the advertised numbers. NoLiMa, an independent needle benchmark, has had no code pushed to it since 17 July 2025. Context Arena, run by an individual, Dillon

Uzar, is a public board that reaches a million, and it has published nothing new for at least nine days: its leaderboard payload was byte-for-byte identical on 31 August, 8 September and 9 September 2026. Whose dataset it runs is not stated on the site and could not be established from its pages or its API.

What that board does show, recounted from its own JSON on 9 September 2026, is stark. It knows of 545 models; 73 have enough data to be ranked at all; and in the longest bin, from 512K to 1M tokens, **nine** models have any score whatsoever, seven of them Google's. The remaining 64 include every Anthropic, OpenAI, Qwen, DeepSeek and Moonshot model on the board, several of which are sold with million-token windows — an absence that records what has not been published rather than what was scored badly.

The nine that do have a score are where a capacity and a competence come apart. Here is the best of them, across every length the board measures.

**The best long-context score on the board, as the input grows**




*Context Arena, 8-needle multi-round co-reference, recounted from the site's own JSON on 9 September 2026 (sha256 e40315b8..., byte-identical to captures of 31 August and 8 September). This is the highest-scoring model in the top bin across the whole board, at its highest effort setting. Sample counts by bin, left to right: 80, 80, 136, 85, 103, 141, 236, 300. The rise from 77.72 to 89.04 between the two middle-length bins is in the published data and is not explained by it.*

▼ **Table view**

The best long-context score on the board, as the input grows

| | Gemini 3.7 Flash, high reasoning |
|---|---|
| 4-8K | 100% |
| 8-16K | 100.0% |
| 16-32K | 98.6% |
| 32-64K | 93.1% |
| 64-128K | 88.6% |
| 128-256K | 89.0% |
| 256-512K | 77.7% |
| 512K-1M | 63.5% |

A capacity of a million tokens and a score of 63.52 per cent at a million tokens are both true of the same model on the same day.

## The fix that shipped switched off

The September 2025 experiment was not a complaint. It came with a diagnosis and a remedy, and the diagnosis is the interesting half, because it rejects the explanation most people give. The paper names the folk account — "some combination of floating-point non-associativity and concurrent execution" — and answers it: "even on a GPU, running the same matrix multiplication on the same data repeatedly will always provide bitwise equal results." The forward pass of a language model, it argues, contains no operations

requiring atomic adds and is therefore run-to-run deterministic. What varies is the batch: the kernels lack *batch invariance*, so "our request's output to depend on the batch size of our forward pass", and the batch size depends on how much load a server happens to be under. The paper is explicit that this is not a property of graphics hardware — endpoints "served from CPUs or TPUs will also have this source of nondeterminism."

The desktop measurement above is the controlled version of that claim: batch size alone changes a greedy answer, with the input held byte-identical and the batch composed of copies of the same prompt. Which kernel changed is not visible from outside; the result is one model on one card in one library; and a bfloat16 model carries coarser last bits than a higher-precision deployment would.

The remedy — batch-invariant kernels — made the thousand completions identical in the laboratory's own re-run, and both vLLM and SGLang have implemented it. On both, it is off unless switched on. vLLM's documentation, on a page whose footer still read 7 September 2026 when it was re-read on 9 September, states:

> Batch invariance is currently in beta.

and, on the cost:

> Enabling batch invariance may impact performance compared to the default non-deterministic mode. This trade-off is intentional to guarantee reproducibility.

SGLang documents the same mechanism, building on Thinking Machines' operators: "The main source is varying batch sizes. Different batch sizes cause GPU kernels to split reduction operations differently, leading to different addition orders."

What the remedy costs is contested, though not by two parties running the same experiment. Thinking Machines reported its own best configuration at 42 seconds against a 26-second baseline — about 1.6×. In August 2026 a group at Nanjing University and Tencent, publishing as arXiv 2608.14376, measured batch-invariant kernels as "increasing more than 2× latency and reducing serving throughput by up to 74%" — in a paper that then presents CoRun, "a scheduling-based system that achieves deterministic inference without requiring batch invariance", so the higher number is the cost of the approach its authors are displacing. A third group, publishing in January 2026 as arXiv 2601.07239, argues against the goal itself: "we take the opposite stance… deterministic inference kills", on the grounds that pinning a model to a single path suppresses the variability that some methods depend on. That objection is to evaluating a model through one canonical completion, and reproducible arithmetic does not require one: SGLang documents deterministic sampling with recorded seeds, different seeds giving different answers.

And the problem is not historical. A preregistered audit of model-based judging, published on 3 September 2026, made 52,988 request attempts across four hosted providers — audit volume rather than sample size, resting on 31 valid task groups and 100 replay pairs — and found byte-identical inputs still returning different rankings, with same-window repeats agreeing at a Spearman correlation of 0.400 against a preregistered requirement of 0.90. Waiting did not help; nor did switching providers, since "four providers

share the floor". Nor, decisively for the remedy, did turning the remedy on: "Self-hosting on batch-invariant kernels helped only while the server was quiet: concurrent load raised disagreement 8.4-fold, back to shared-endpoint magnitude."

A second controlled experiment, posted the following day by a single author and so far unreviewed, finds a different default-on cause with the same shape. Prefix caching — reusing the stored keys and values of a shared prompt prefix, and "enabled by default in the major open-source stacks" — changed an agentic workload's trajectory on "36.2 percent of episodes at 16-bit precision and on 75.0 percent at four-bit", while with the cache disabled repeated execution was bit-identical in 800 of 800 episodes. Its conclusion is the batch-size finding transposed: "Cached serving is deterministic given cache state, and irreproducible in practice because that state is absent from the request and never reset by default." Two independent groups, two different mechanisms, the same structural answer — a setting the caller cannot see decides whether the same request gets the same reply.

Diagnosed, remedied, remedy shipped switched off, problem still measurable a year later. That sequence is not a failure of engineering. It is a series of decisions, each of which somebody made and dated.

Two of those facts can be re-read on a date by anyone who wants to know whether they have changed. The first is a pair of words on vLLM's batch-invariance page — *beta*, and *default*. When either goes, reproducible inference will have stopped being something a user has to know to ask for. The second is a count rather than a score, which is what makes it checkable without a laboratory: 545 models known to Context Arena, 73 with enough data to be ranked, nine with any figure at all at the longest lengths. Counting them again in a month is a two-minute job, and the number that moves will say more than any leaderboard position.

## The idea to keep

**The number is a capacity, not a competence**, and both of the mechanisms described here are choices rather than properties.

Something chooses how the next token is drawn from the scores — a reshaping constant, a cut at a count or at a share of the mass, or no cut at all. Something chooses what remains in front of the model and what is discarded, and whether what is discarded leaves a summary behind. Neither is the model deciding. Both are configured, documented and dated, and the dates are the point: a window that could be used on 29 April could not on 1 May, a headline number that meant 750,000 words came to mean 555,000, and a remedy for irreproducibility exists and is not switched on.

The image worth carrying is the smallest one. One prompt, greedy decoding, randomness off. At a batch of one, the same answer five times out of five. At a batch of sixty-four, the same answer five times out of five — a different answer. Nothing inside was reasoning differently; the only thing deliberately changed was how many copies the machine was computing at that moment, which on the laboratory's account changes the order in which some numbers are added. On a private machine that is controllable. On a shared endpoint it is not, and the laboratory that found it put the general case exactly:

> From the perspective of an individual user, the other concurrent users are not an "input" to the system but rather a nondeterministic property of the system.

## The limits of the evidence

The desktop measurements ran one model on one card in one library, and which kernel changed with the batch size is invisible from outside them. The nucleus figures come from two passages and one model, and the gap between the Austen chapter and the new prose is consistent with memorisation, with Austen simply being more predictable, or with both. The out-of-memory failure at 12,288 tokens belongs to a shared 24 GiB card and one library's defaults rather than to the architecture, whose declared window is twenty times longer. Retrieving one planted string to 16,505 tokens is the easy case on the vendors' own account, and shows that the model does not stop rather than that long context works. And nine models with scores at the longest lengths is a count of what has been published on one board, which is a different quantity from how those models would score.

## Sources


| Source | Date | Note |
|---|---|---|
| Horace He and Thinking Machines Lab — *Defeating Nondeterminism in LLM Inference* | 10 Sep 2025 | thinkingmachines.ai; DOI 10.64434/tml.20250910; Qwen3-235B-A22B-Instruct-2507 on vLLM with the FlexAttention backend |
| Jurafsky and Martin — *Speech and Language Processing*, 3rd edition draft | released 19 Aug 2026 | §7.6 Decoding: 7.6.1 greedy, 7.6.2 random sampling, 7.6.3 temperature, 7.6.4 top-k and top-p |
| Claude Shannon — *A Mathematical Theory of Communication* | Jul and Oct 1948 | Bell System Technical Journal 27; §3, the series of approximations to English |
| George A. Miller and Jennifer A. Selfridge — *Verbal context and the recall of meaningful material* | Apr 1950 | American Journal of Psychology 63(2):176–185; PMID 15410879; DOI 10.2307/1418920; NLM MeSH: Humans, Learning, Memory, Mental Recall |
| Angela Fan, Mike Lewis, Yann Dauphin — *Hierarchical Neural Story Generation* | 13 May 2018 (v1) | arXiv 1805.04833; §5.4; k = 10; presented as a technique used, with no novelty claim |
| Holtzman, Buys, Du, Forbes, Choi — *The Curious Case of Neural Text Degeneration* | ICLR 2020 (preprint 22 Apr 2019) | arXiv 1904.09751. Checked against both texts: the phrases "if k is small", "if k is large", "inappropriate candidates" and "bland or generic text" appear in the ICLR version and in none of them in v1, which argues only the flat-distribution half and carries four authors rather than five |
| Google — *Our next-generation model: Gemini 1.5* | 15 Feb 2024 | blog.google; read from the Internet Archive capture of 15 Feb 2024 15:13:50 UTC |
| OpenAI — *Introducing GPT-4.1 in the API* | 14 Apr 2025 | openai.com; read from the Internet Archive capture of 15 Apr 2025 01:36:10 UTC |
| Anthropic — Claude Platform release notes | entries 12 Aug 2025 – 3 Sep 2026 | platform.claude.com; 137 dated entries; read 8 Sep 2026 |

| Source | Date | Note |
|---|---|---|
| Anthropic — Messages API reference, Context windows, Compaction | read 8 Sep 2026 | platform.claude.com; `docs.claude.com` now redirects here |
| OpenAI — `openai/openai-openapi` specification and the compaction guide | read 8 Sep 2026 | 3,071,528 bytes on 9 Sep 2026, up from 3,050,607 the day before; none of the 2,790 changed lines touch `truncat`, and `truncation` is marked `deprecated: true` in both |
| Google — Gemini API long-context guide | updated 22 Jun 2026 | ai.google.dev; the multiple-needles limitation and the token-avoidance advice |
| Google — Gemini 3.8 Flash announcement | 2 Sep 2026 | blog.google; the context window is not mentioned |
| vLLM — Batch Invariance documentation | page dated 7 Sep 2026 | docs.vllm.ai; `VLLM_BATCH_INVARIANT=1`, beta, off by default |
| SGLang — Deterministic Inference documentation | read 8 Sep 2026 | docs.sglang.ai; `--enable-deterministic-inference`, off by default |
| Zhao and colleagues — *CoRun: Padding is Simple and Efficient for Deterministic LLM Inference* | 14 Aug 2026 (v1) | arXiv 2608.14376; the >2× latency and 74% throughput figures |
| Joshi, Aggarwal, Das and colleagues | 12 Jan 2026 (v1) | arXiv 2601.07239; the dissent against deterministic inference |
| Zhu and Zhang — preregistered reliability audit | 3 Sep 2026 (v1) | arXiv 2609.04198; 52,988 request attempts across four providers |
| Aditi Patodiya — *Same Request, Different Answer: Quantization Amplifies Cache-Induced Divergence in LLM Serving* | 4 Sep 2026 (v1) | arXiv 2609.04748, cs.SE; single author, unreviewed; 80-episode agentic workload, two engines, four weight formats; 0 of 800 divergences with the cache disabled |
| HELMET — Princeton PLI and Intel | ICLR 2025 | "controllable lengths up to 128K tokens", the benchmark's own description of its reach |
| Context Arena, run by Dillon Uzar | read 9 Sep 2026 | contextarena.ai; leaderboard API, 107,766 bytes, sha256 `e40315b8…`, byte-identical to captures of 31 Aug and 8 Sep 2026; dataset provenance not stated on the site |
| Direct measurement, one desktop machine | 8 Sep 2026 | GPT-2 and Qwen3.5-4B run locally; scripts, JSON and findings kept with the episode's research record, not published |