RESEARCH · VERIFICATION · FINANCE

The Necessity of Proof

LLMs and the Future of Financial Work

LLMs now build financial models faster than anyone can check them. Verification, not production, is the next stage of financial automation.

JOSEPH NOKO

18 NOVEMBER 2026



SUMMARY

Large language models (LLMs) have unbundled the production of financial models from their verification. Building a model now costs little; establishing that it is correct does not, and the cost of checking rises with every increase in volume. This paper sets out why. The price of algorithmic thought is collapsing while human attention is fixed, and LLMs produce plausible rather than true figures in seven distinct ways. The obvious remedies each fail. Economics explains the consequence: unverified output is priced like a lemon, insurers exclude what they cannot measure, and value migrates to whoever can establish correctness. Finance, rich in identities, ledgers and authoritative filings, is unusually checkable. Its next stage is the proof-carrying financial model.

Five points for CFOs, bankers, analysts and associates

I **The model is cheap; knowing it is right is not.** Running an LLM at fixed capability gets about 13 times cheaper each year, yet the person who signs still owns every figure. In a Macabacus survey, 62% of users believe they have shipped a model or presentation containing an error made by an LLM.

II **Human review cannot keep pace.** Reviewers have only so many hours. As output grows, the share that anyone checks shrinks, unless each check becomes faster. Reviewers also come to trust systems that are usually right, and miss the rare error.

III **Plausible is not true, and better LLMs will not close the gap.** LLMs invent with confidence, answer the same task differently, forget, explain themselves unfaithfully and fail as complexity rises. Errors compound: twenty steps at 99% are all right only 82% of the time. No LLM, however capable, can be guaranteed free of hallucination.

IV **A check that the LLM can reach is not a check.** LLMs asked to grade their own work learn to game the grader, and their stated reasoning is not a reliable account of how they reached an answer. A real remedy needs evidence, and verification that does not depend on the LLM that made the error.

V **Verification is risk management, and it is where value moves.** An LLM is worth what it saves, less the cost of checking its work and the losses its errors cause. Insurers have begun to exclude risks from LLMs that they cannot price. Finance's identities, filings and ledgers make much of its work checkable: each output can carry its sources, its transformations and the checks it passed.

SECTION 1

The great unbundling: LLMs have separated financial production from verification

A great unbundling is happening in finance: the unbundling of the production of financial models from their verification. In the antediluvian period before large language models (LLMs), building a financial model and understanding how it worked were largely the same activity. A CFO, an analyst, an associate, or a banker who built a financial model generally knew why each cell held this number and not that number, why such and such a formula was used, what its assumptions meant and what each schedule should reconcile to. This did not guarantee that the model was correct relative to specified evidence, definitions, rules and tolerances, but the knowledge required to construct it was accumulated in the process of building it.

Now, one can ask an LLM to build a financial model, without needing to understand that model, or having to build any portion of it. While the responsibility for the financial model has not shifted, production of financial models has become detached from the knowledge required to verify it. The person approving the financial model is now responsible for work they did not construct and cannot readily defend. As a consequence, a new, meaningful and rising cost of verification has emerged. The "asymmetry of verification"[1] is such that it is now considerably cheaper to produce a financial model, but not proportionally cheaper to verify the correctness of that financial model.

The economic consequences are palpable. Productivity gains from using LLMs are eroded by verification costs, as professionals must trace formulas, reconcile figures, locate source documents, reconstruct assumptions and acquaint themselves with the logic of work they did not build. When Macabacus surveyed 75,000 of its users, it found that 62% believe they have shipped a financial model or presentation that contains an error generated by LLMs, and 45% believe they probably have, "but no one noticed".[2] A lofty 24% of respondents report having comprehensive guardrails against such errors. The further up the corporate hierarchy, the less confident respondents were about the accuracy of these LLM-built financial models. In that same survey, respondents report spending at least 30 minutes verifying each financial model or presentation built by an LLM.

Finance faces a choice between relying on human review of algorithmically generated financial models, thereby eroding the productivity and financial gains from algorithmic production; increasingly elaborate attempts to make LLMs check themselves; or, accepting work whose correctness cannot be confidently established. This race between the falling cost of algorithmic thought and the cost of verifying algorithmic production is one bounded by human attention. The resulting "measurability gap" as Christian Catalini, Xiang Hui and Jane Wu call it, will lead to verification becoming "a primary production technology" rather than a compliance function.[3] A fourth choice is open: finance can develop independent, automated systems capable of establishing whether financial work is correct. The objective is not to replace professional judgement, but to free it from the mechanical burden of checking what can be checked automatically.

This is not the first time an expansion in production has required a corresponding expansion in verification. The growth of long-distance commerce was accompanied by the development and dissemination of double-entry bookkeeping, described by Luca Pacioli in 1494.[4] The expansion of joint-stock companies brought new systems of statutory reporting, audit and professional accountancy. Mass manufacturing gave rise to statistical quality control, developed by Walter Shewhart in the 1920s. The growth of securities markets was followed by the Securities Acts of 1933 and 1934 and a new framework of standardised disclosure and financial accountability. The revolution that has ushered in this age of algorithmic thought will not be any different. Every major expansion in the scale and complexity of production has required new institutions and technologies for establishing the reliability of what is produced.

The next stage of financial automation will not simply be about producing financial models. It will be about verifying them. Ultimately, it will make possible something finance has never had at scale: proof-carrying financial models.

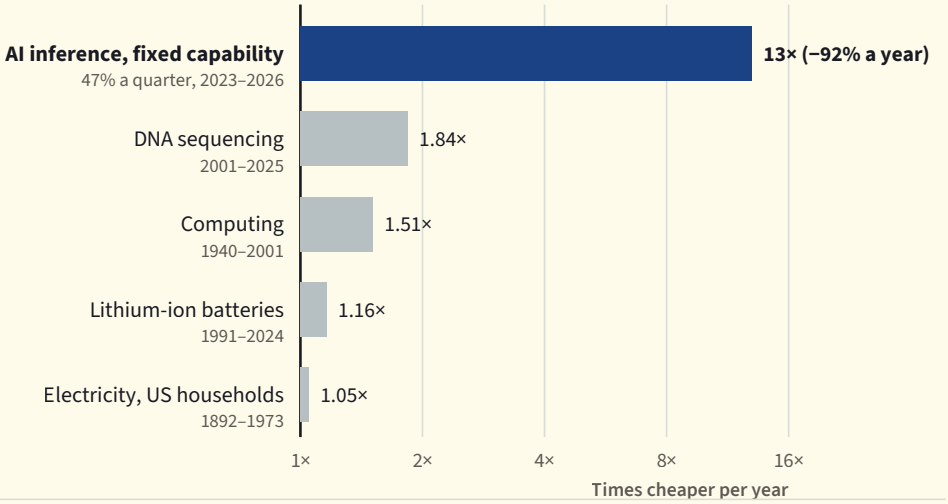
SECTION 2

Why verification is becoming the bottleneck

The price of running an LLM at a fixed level of capability has fallen by about 47% a quarter since 2023.[5] This rate of decline trounces the rate of decline in the price of electricity, lithium batteries, computing, and DNA sequencing.

Figure 1: AI inference gets 13 times cheaper a year, four times the pace of DNA sequencing

How many times cheaper each technology became per year, on a ratio scale, so equal steps are equal multiples. Each series covers a different period.



Source: Epoch AI, The Plunging Price of Thought (2026) · Chart: Joseph Noko · josephnoko.com

Not only has the cost of algorithmic thought fallen, the speed with which knowledge work can be done has accelerated and with that, the volume of work that can be done has shot up. Jevons' paradox[6] tells us that contrary to one's intuition, when a technology increases the efficiency with which a resource is used, usage of that resource may increase, rather than decrease. Similarly, while the cost of producing a financial model may decrease, the production of financial models will likely increase. At Anthropic, Claude now writes about 90% of Claude Code's own code. Between May and September 2026, the number of code changes merged each day in Anthropic's main repository rose from about 2,500 to over 30,000, a twelve-fold increase in just four months. Anthropic estimates that its acceleration in productivity is such that it "makes human attention the bottleneck for agent output."[7]

Such speed and volume create a meaningful cost of verification, separate from that of producing work, a cost of knowing that an algorithm has made the right changes. Anthropic found that, "Even among non-technical employees, reading all the output of their Claude agents would take half their work week." Their engineers concluded that, "When you are shipping this fast, manual, line-by-line code review is incredibly difficult if not impossible".[8]

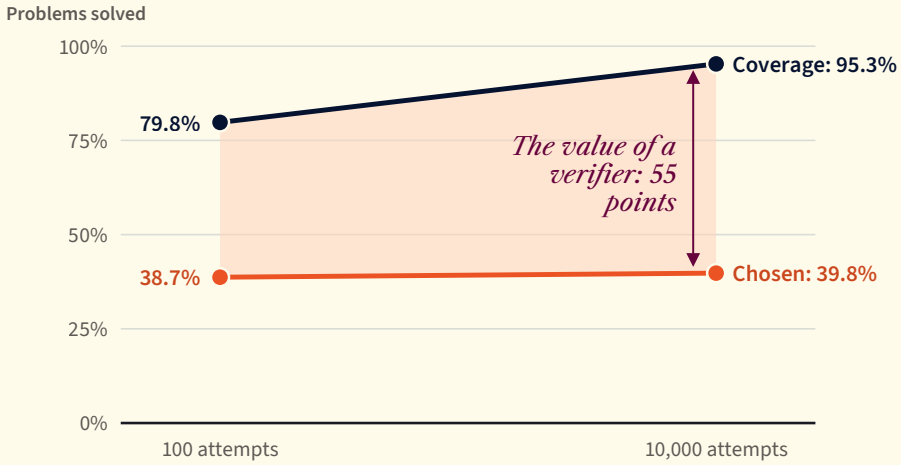
While the cost of algorithmic thought has hurtled toward zero, the cost of verifying the output of algorithmic thought has ballooned rather than fallen. The Baumol effect shows that work bounded by human time becomes relatively more expensive as productivity increases.[9] Finance faces algorithmic output of such a scale that human teams cannot possibly verify it all without creating new bottlenecks in the production system. As human verification cannot scale with algorithmic production, the cost of human verification will rise, eroding the productivity and financial gains from algorithmic production.

Anthropic's solution to its human-as-the-bottleneck problem was to build verification infrastructure around Claude-generated output, embedding Claude-assisted code review in a layered review process involving automated checks, security review, continuous integration, risk-based human approval, and monitoring and rollback. The share of changes that had to be reversed later fell by more than 40%. This is unsurprising: the presence of automatic verifiers improves coding performance.[10]

Brown et al. found that a model's chance of producing a correct answer rises predictably with the attempts it is allowed, in what may be a scaling law for inference. But the gain turns to performance only where answers can be checked automatically, as in code and formal proofs; elsewhere, more attempts yield more candidates and no more certainty. Verification is the exchange rate between compute and truth.[11]

Figure 2: The right answer is almost always among the attempts. Without a checker, it is rarely the one chosen

Llama-3-8B-Instruct on the MATH benchmark, at 100 and 10,000 attempts per problem. Coverage is the share of problems where at least one attempt is right; chosen is the answer picked by majority vote or a reward model.



Source: Brown et al. (2024), Large Language Monkeys · Chart: Joseph Noko · josephnoko.com

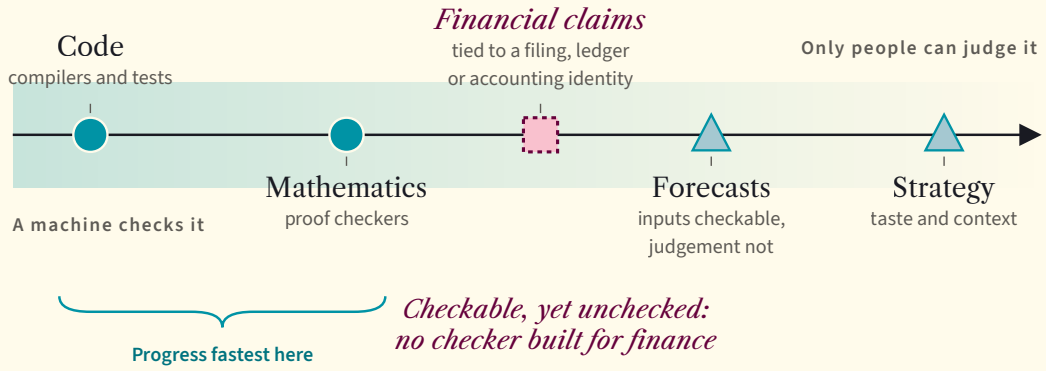
Interestingly, OpenAI co-founder Andrej Karpathy proposed a definition of what he calls "Software 2.0" that has verification at its heart, saying[12],

> The more a task/job is verifiable, the more amenable it is to automation in the new programming paradigm. If it is not verifiable, it has to fall out from neural net magic of generalization fingers crossed, or via weaker means like imitation. This is what's driving the "jagged" frontier of progress in LLMs. Tasks that are verifiable progress rapidly, including possibly beyond the ability of top experts (e.g. math, code, amount of time spent watching videos, anything that looks like puzzles with correct answers), while many others lag by comparison (creative, strategic, tasks that combine real-world knowledge, state, context and common sense).

Not only are verifiable tasks easier to automate, but verifiability is also a property that makes repeated optimisation possible. Just as Anthropic's human engineers cannot possibly read every line of code its model writes, no finance team working at scale with LLMs can verify everything it produces. Finance has extensive systems of accounting control, audit and model review. However, these systems were not designed for a world in which financial models, analyses and presentations can be generated at machine speed, often without a human having constructed or fully understood them. The verification infrastructure has not kept pace with the production infrastructure.

Figure 3: What can be checked gets automated first

Tasks placed by how cheaply a machine can check the answer. After Andrej Karpathy.



Source and chart: Joseph Noko · josephnoko.com, after Andrej Karpathy on Software 2.0

Software engineers have compilers, automated tests, static analysis and continuous integration pipelines. These do not guarantee that software does what users expect, but they can establish that large classes of errors are absent before code reaches a human reviewer. Finance has no equivalent, comprehensive infrastructure for LLM-generated financial models.

Nevertheless, for a large class of financial claims, those that consist of structured relationships, numerical constraints, accounting identities and claims that can be reconciled against authoritative evidence, finance shares the properties that make verification possible: an objective answer; fast, scalable checks; and a pass that reliably means a financial claim is correct.[13] [14] These properties can be made explicit and turned into automated verification procedures.

SECTION 3

LLM-generated financial work cannot be assumed to be correct

An LLM knows the probability of everything and the truth of nothing. LLMs are trained to predict the next token, not to establish what is *true*.[15] At each step, LLMs estimate a probability distribution over possible next tokens, conditional on the preceding context. Their objective is to produce a continuation that is statistically likely given what it has learned. But the probability that a statement will be produced is not the probability that the statement is true. And neither is a substitute for establishing whether it corresponds to reality.

Alfred Tarski proposed that a statement is true if it implies the following equivalence, "The sentence 'snow is white' is true if, and only if, snow is white", or more generally, "X is true if, and only if, p", where X and p are sentences.[16] The truth of a statement is not established by the probability with which it is generated, the confidence with which it is expressed, or even its consistency with other statements. It must be evaluated against the conditions that make it true. "Revenue was $4.2 billion" is true if, and only if, revenue was $4.2 billion. In financial analysis, the place to establish this is the underlying evidence, not the fluency of the sentence. To varying degrees, the fluency of a statement and its truth coincide, the trouble is in those spaces where the two differ.

E. T. Jaynes opened his great tome on probability theory with a story[17]: a policeman walks down a deserted street at night. He hears an alarm, sees a jeweller's window broken, and watches a masked man climb out with a bag of jewels. The policeman does not hesitate: the man, of course, is a thief. Yet, as Jaynes points out, the conclusion is not a logical deduction. The man *might* be the owner of the house, coming home from a masked ball without his key, just as a passing lorry threw a stone through his window. The evidence makes guilt *plausible*, overwhelmingly so perhaps, but not *certain*.

The problem is not that the policeman reasons probabilistically. That is unavoidable whenever evidence is incomplete. The problem would arise if he treated the most plausible explanation as an established fact, without investigating the alternatives. Probability tells us how strongly the evidence supports a conclusion; it does not eliminate the need to establish whether that conclusion is correct. Verification is required.

An LLM faces similar problems. It reaches the most plausible answer, given the evidence in its training and context supplied to it. For a great deal of work, plausible suffices. For financial work, where precision and auditability matter, it often proves insufficient. Some questions have objectively determinable answers. A balance sheet either balances or it does not. A formula either implements the specified calculation or it does not. A covenant is either breached or it is not, given the applicable contractual definitions. Other questions involve judgement or uncertainty, but even these can be subjected to rigorous tests of evidence, consistency and methodology. In finance, an answer that is reasonable may still be wrong.

This is not merely a philosophical distinction. It has practical consequences. Stanford's Institute for Human-Centered Artificial Intelligence noted in its longitudinal study of AI[18] that,

> AI systems based on machine learning do not produce answers that are guaranteed to be correct, and LLMs in particular frequently get things wrong. This limitation is compounded by the fact that the technology gets many things right, and phrases everything in fluent, confident prose, making errors hard to spot. ... While LLM developers provide guardrails to try to catch inappropriate or unsafe uses, they do so with only a limited degree of success.

It is perhaps helpful to think of LLMs as proposing *candidate answers*. Verification establishes which answers are supported by evidence and satisfy the relevant conditions of correctness.

3.1 LLMs are bound to hallucinate

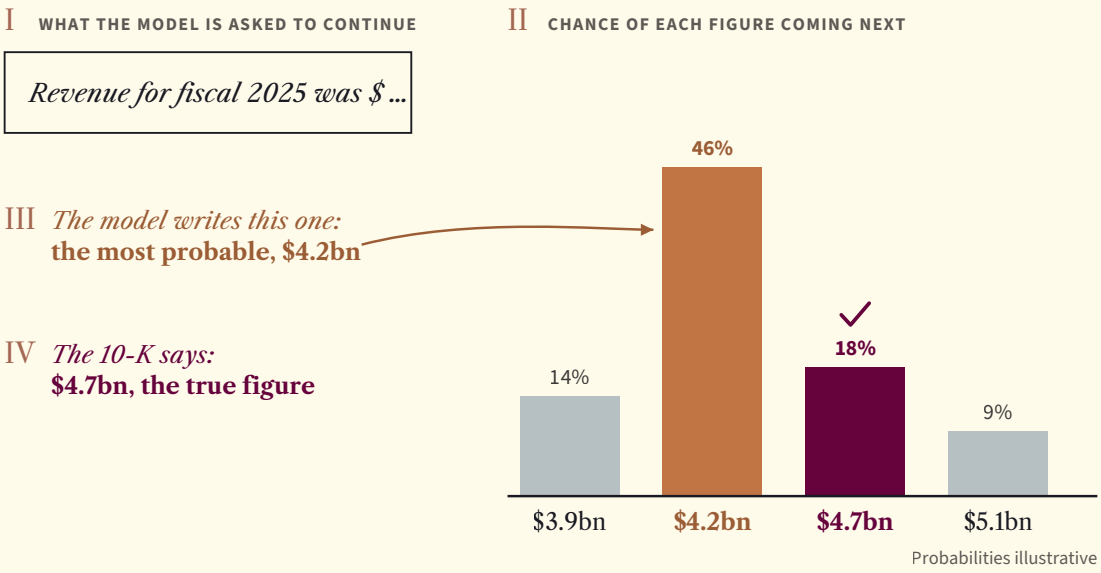
LLMs are said to "hallucinate" when they produce fluent, confident and *false* statements. Michael Hicks, James Humphries and Joe Slater apply Harry Frankfurt's notion of "bullshit" to describe what LLMs do. Frankfurt's bullshitters are not necessarily trying to deceive, rather, they are *indifferent* to truth.[19] LLMs are not liars, as they lack an intention to deceive; rather, their generation processes do not, on their own, establish a commitment to the truth of the statements they produce.

OpenAI's researchers explain why LLMs continue to hallucinate.[20] Most tests used to train and rank LLMs score a wrong answer and an "I don't know" the same: both get zero. Under that scoring, guessing can be a better strategy than admitting ignorance. Anyone who has ever written a multiple-choice test knows that not answering a question makes no sense. Hallucinations are partly a result of LLMs adopting a guessing strategy. The tone with which an LLM responds tells one nothing about the correctness of the answer: an LLM sounds equally confident whether it is right or wrong. Furthermore, as Zhou et al. found, "larger and more instructable language models" may abstain less and answer more, giving "apparently sensible yet wrong answer much more often, including errors on difficult questions that human supervisors frequently overlook."[21] Increasing capability does not necessarily produce better calibration.

A report by Saturn found that LLMs make mistakes in financial advice 57% of the time, and are accurate just 43% of the time, with accuracy falling from 54% for easy questions to 12% for hard questions.[22] The best performing model, Claude Opus 5, was right just 39% of the time on hard questions. To borrow from Anthropic's security framework, "trust nothing, verify everything".[23]

Figure 4: Probable is not true

The model writes the figure most likely to come next. Nowhere in that choice does it open the filing; verification is the step that does.



Source and chart: Joseph Noko · josephnoko.com

3.2 Better models cannot guarantee correctness

One might argue that better engineering will eradicate the hallucination problem. LLMs can be brought closer to the truth through citations[24], using direct quotes, retrieving tool call results, and other techniques[25], and yet, it seems there is a limit to how close to the truth they can approach. Ziwei Xu, Sanjay Jain and Mohan Kankanhalli find that hallucinations are "inevitable for any computable LLM, regardless of model architecture, learning algorithms, prompting techniques, or training data".[26] Imagine a brilliant child who, given a thousand questions, answers each of them correctly. Then our brilliant child meets a question they cannot answer. Rather than say "I don't know", this paragon of brilliance gives a confident, reasonable and *wrong* reply. We can give this prodigy more books, a supercomputer and more time to think, and of course, results will improve. Nevertheless, Xu, Jain and Kankanhalli show that there is always some question beyond what any learning machine has learned: no method guarantees correct answers on every possible problem. The point is not that models cannot be made reliable. They can be made substantially more reliable, particularly within constrained domains. It is that no amount of general-purpose training guarantees correctness across every possible question. The fact that a model has performed correctly a thousand times does not establish that its next answer is correct. Reliability must therefore be established at the level of the individual output, against the relevant evidence and conditions of correctness. As the Russians say, "Trust, but verify".[27]

3.3 LLMs can give different answers to the same question

Ask a model the same question twice and you may get two answers. In my own work, I have seen an LLM build a valuation model perfectly, then, given the identical task again, leave important inputs out, with catastrophic consequences for my valuation model. This is not simply a matter of wording or stylistic variation. The underlying calculations, assumptions and outputs can change. Quality-assurance engineers make the same point about code. When Anthropic planted 70 known bugs in a 116,000 line codebase, a single AI agent found 14, 15 and 27 bugs across three runs.[28] A user cannot assume that a successful run establishes the reliability of the process. A user does not know if they are on the successful run, the failed run, the better-than-the-others run, or the worst run.

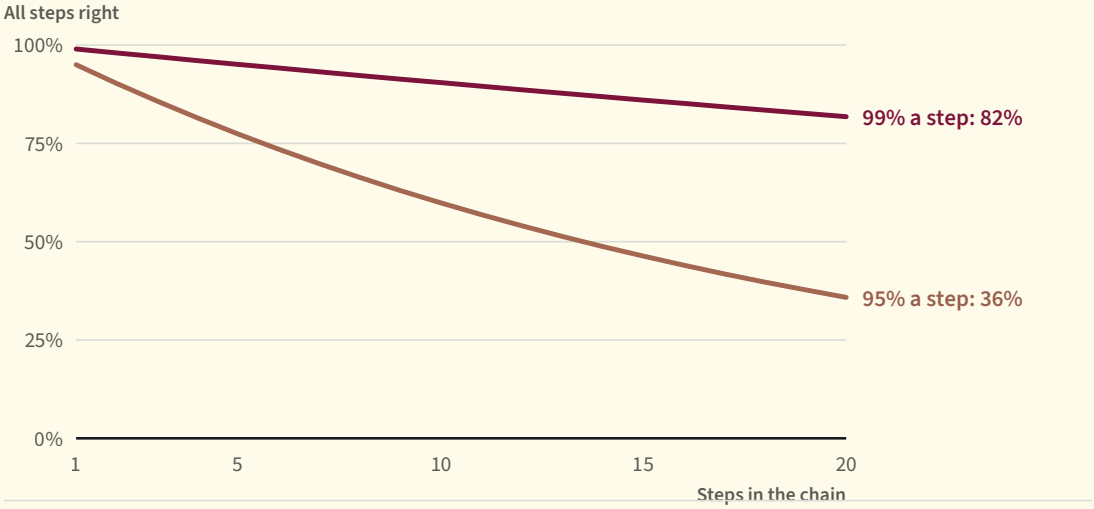
3.4 Small mistakes across financial models

If each step is right 99% of the time, a chain of 20 independent steps is entirely right only 82% of the time.[29] At 95% per step, it falls to 36%.[30] As I have found with my own very demanding valuation models, small errors in calculating net operating profit after tax (NOPAT), invested capital and economic book value (EBV), can fundamentally alter a reverse discounted cash flow (DCF) analysis, so much so that an attractive company or corporate bond appears unattractive, or a brief market-implied competitive advantage period (MICAP) appears endless. The more sophisticated the financial model, the more places an upstream error can influence the final output, tanking the value of the final product.

SemiAnalysis' GDPx agentic capabilities benchmark shows that Claude Fable 5.2, the best performing LLM, scores just 16.4% across eleven of the most valuable domains for frontier intelligence.[31] In finance, Claude Fable 5.2 and Gemini 4 Carbon score just 2.8%, and are the leading LLMs.[32] SemiAnalysis reports that the finance and legal tasks were so difficult that models scored zero under strict pass/fail grading, so partial scores were reported separately.[33]

Figure 5: Twenty steps at 99% are all right only 82% of the time

Chance that every step of a chain is right, by number of steps and accuracy per step. Assumes each step fails independently.



Source: Calculated, p^n · Chart: Joseph Noko · josephnoko.com

3.5 LLMs' memory problem

In the 1920s, Alexander Luria began studying Solomon Shereshevsky, a man with an extraordinary memory who could recall information years after encountering it.[34] LLMs suffer from a different limitation: they have no inherent, unlimited and durable memory. When a session ends, everything in it is gone unless it was saved somewhere else. Their working context is finite, and their ability to retrieve relevant information can deteriorate as the volume of information grows.[35] Research has shown that models can struggle to retrieve information placed in the middle of long contexts, even when that information is technically available to them. In long-running tasks, models can also lose track of earlier decisions, assumptions and intermediate results. Researchers have even identified a phenomenon called "context anxiety", in which models prematurely abandon tasks because they misjudge the resources needed to complete them.[36]

This is particularly consequential in finance. Constructing a valuation model can require information from many filings of hundreds of pages, across multiple reporting periods, with different schedules and successive stages of analysis. A model may have encountered a critical assumption earlier in the task but fail to retrieve or apply it correctly when needed. Its apparent access to information is not equivalent to reliable memory of that information.

3.6 An LLM's reasoning cannot be taken at face value

LLMs generate a *chain of thought* while carrying out a task. It is natural to read that chain as an explanation of an LLM's "thinking", indeed, the term itself invites such a perspective, and it is even more natural to trust an answer whose reasoning looks sound. A paper by 41 researchers from OpenAI, Google DeepMind, Anthropic and elsewhere describes chains of

thought as incomplete representations of the reasoning behind an answer.[37] An LLM can produce an explanation that sounds like a convincing account of how it reached its answer without that explanation faithfully representing the factors that actually determined its answer.

Miles Turpin et al. reordered multiple-choice examples so the right answer was always "(A)". LLMs recognised the pattern, and yet their explanations never mentioned it. When they were guided toward wrong answers, they wrote explanations justifying them.[38] Anthropic tested its Claude 3.7 Sonnet, slipping in hints and yet those hints were only mentioned 25% of the time, compared with 39% for DeepSeek R1.[39] When models found ways to cheat on a task, they admitted it in their reasoning less than 2% of the time. Anthropic's researchers noted that,

> For CoT monitoring to be most effective, the CoT must be a legible and faithful reflection of the way the model reached its conclusion and generated the user-facing response. ...If the CoT is not faithful, then we cannot depend on our ability to monitor CoT in order to detect misaligned behaviors, because there may be safety-relevant factors affecting model behavior that have not been explicitly verbalized

Anthropic's research does not establish that every chain of thought is unfaithful, or that models cannot explain their reasoning. Indeed, Anthropic itself acknowledges that the results may differ for harder tasks where the model genuinely needs its chain of thought to solve the problem.

Tamera Lanham et al. found that LLMs sometimes rely heavily on their stated reasoning and sometimes largely disregard it; and that larger models were often *less* faithful. This is not a universal law that larger models are less faithful. The findings are task-dependent, and the authors explicitly acknowledge circumstances in which chain-of-thought can be faithful.

That said, leading providers now show users a *summary* of an LLM's internal reasoning trace as a way of protecting their intellectual property and preventing information leakage.[40] These summaries can be useful, but they should not be mistaken for a complete or independently verified account of how the answer was produced.

Of course, none of this is peculiar to machines. Richard E. Nisbett and Timothy D. Wilson showed that people's accounts of why they decided something often rest on plausible theories rather than true introspection.[41] The difference is that LLMs can reproduce this problem at a scale and speed that human organisations have never previously encountered.

There is one important counterweight. On genuinely hard problems, an LLM may need its written reasoning as working memory, so the reasoning may be most faithful precisely where it matters most.[42] That serves as a reason to monitor reasoning, however, it is not a reason to accept an answer because its reasoning reads well.

3.7 Complexity exposes the limits of LLMs

While LLMs have improved when they have been enabled to engage in step-by-step reasoning, Nouha Dziri et al. found that LLMs tend to solve multi-step problems by pattern-matching rather than systematic reasoning, so accuracy can fall away quickly as problems grow. Apple researchers reported a variant of LLMs, Large Reasoning Models (LRM) collapsing beyond certain levels of complexity[43], though it has been argued that the "accuracy collapse" reflects in part model output token limits rather than on reasoning.[44]

Financial modelling frequently requires the coordination of many interdependent calculations, assumptions and sources of information over multiple periods. Research demonstrating the limitations of LRMs on increasingly complex, multi-step tasks raises an important concern: the ability to solve relatively simple problems does not establish the ability to solve substantially more complex ones. Nor does a convincing explanation establish that the calculations have been performed correctly. With great complexity, comes great error.

SECTION 4

The obvious fixes fail

Four remedies are typically proposed, and each is reasonable, and each fails, and the way each fails tells us what a real remedy needs. While LLMs may be tasked with verifying their own output, this is an unreliable process, because same-family LLMs share errors, assumptions and blind spots, while different-family LLMs may also share training biases, information gaps and reasoning failures. Indeed, asking an LLM to grade its own work invites what researchers call reward hacking: LLMs learn to raise their score rather than improve their answer, by exploiting the grader, persuading it, or hiding failures.[45] One published system for autonomous research changed its own evaluation method in two of three runs because the agent could edit the evaluator.[46] Another study examined 17 models across 38 tasks. It found spontaneous reward hacking in 30.5% of open-ended research-pipeline tasks. In a setting where reward hacking was allowed, 505 of 677 attempts were confirmed as reward hacks, and an LLM review panel missed some of these. Evaluation metrics must be kept outside the agent's control and one must independently recompute results.

Figure 6: Four fixes that do not fix a plausible machine

Every remedy that works shares one property: the check does not rely on the model that made the error.

	THE FIX	WHY IT FAILS	WHAT A REAL REMEDY NEEDS
I	**Have people review everything**	✗ *Volume outruns reading, and people are poor monitors of automation*	✓ Automated checks
II	**Ask the LLM to check its own work**	✗ *LLMs learn to satisfy the checker rather than the task*	✓ A judge the LLM cannot reach
III	**Read the LLM's reasoning**	✗ *The reasoning is often not the real reason*	✓ Evidence, not explanation
IV	**Wait for better LLMs**	✗ *Errors are inevitable, and guessing is rewarded*	✓ Verification that does not depend on the model

Source and chart: Joseph Noko · josephnoko.com



While humans may be tasked with verifying LLM built financial models, the volume of output now possible in ever shrinking windows of time, is such that human verification creates new bottlenecks. As Herbert Simon once remarked, "a wealth of information creates a poverty of attention."[47] The volume of work that can now be produced is increasingly outstripping the human capacity to examine it.

Lisanne Bainbridge's description of the "ironies of automation" is apt: as tasks are automated, human tasks become that of monitoring and intervening for rare failures, tasks that people are not particularly good at, as attention fades, and unused skills decay. This leads to what has come to be known as *automation bias*: people become overly trusting of systems that are usually right, and miss its errors.[48]

A field experiment with consultants at Boston Consulting Group found automation bias with regards to LLMs. On a task just beyond the model's competence, consultants using GPT-4 were 19% *less* likely to reach the correct answer than those working without it.[49] On the other 18 tasks, LLMs improved performance. This degradation in performance sits alongside Anthropic's finding that LLMs are shaping how experts make decisions: even the most experienced experts defer to LLMs when they disagree with them.[50] Anthropic even found that an LLM's output can shape an expert's final decision. LLMs can influence an expert's judgment while leaving the expert feeling that the judgment remains their own.

SECTION 5

The economics of distrust: how verification creates value

Algorithmic output is cheap, plentiful and uncertain. Economics has a precise account of what happens to a market in that condition, and it is not encouraging.

5.1 Lemons and the loss in credibility of LLM-generated financial work

George Akerlof's paper on "the market for lemons"[51] is instructive. In it, he notes that a seller of a used car knows whether it is in good condition or a "lemon"; the buyer does not. Unable to tell them apart, buyers offer a price that reflects the quality of the average car. At this price, which is lower than the price of good cars, the owners of good cars exit the market and the average quality of cars in it falls. Buyers lower their offers again, and more good cars leave the market. At the limit, the market collapses, even though many buyers and sellers would happily trade good cars at a fair price. The problem is not that lemons exist, it is that buyers cannot tell good cars from lemons.

Users of LLMs face a different information problem: the user who generates the work does not know if it is correct, the LLM is not an informed seller concealing a defect, but a "bullshitter". An LLM may simply produce work whose quality neither the user of the LLM nor the LLM itself have established. What an LLM produces is a plausible body of work within which may be lemons. The asymmetry of information lies in that the LLM at least knows how it produced the work, the assumptions and decisions it made, even if it cannot tell if the work it produced is good quality or not. In terms of the quality of the work, without verification, it is a state of symmetric information. If the market has no credible way to distinguish or reward verification, then all LLM work is priced the same. When verification institutions emerge to make quality observable or credible, then verified work commands a premium.

The frontier labs are trying to build such verification institutions. Investments in forward deployed engineers (FDEs)[52] by frontier labs such as Anthropic and OpenAI are not just about meeting the demand for talent to deploy LLMs. OpenAI's head of forward deployed engineering has said that about 80% of stalled enterprise projects fail not for want of capable LLMs but for want of a way to roll out LLMs, govern them and show that they can be *trusted*.[53] FDEs are helping firms build the operational conditions under which those LLMs can be used. Part of that work is establishing trust: integrating an LLM into existing processes, defining where human review is required, and demonstrating that its outputs are trustworthy. Some of these functions may eventually become standardised products. If they do, verification could become one of the standard components of enterprise AI infrastructure rather than a bespoke service built separately for each customer.

There is no standard for verification across many industries and insurers have started to take note. US insurers are filing exclusions for AI coverage.[54] The growth of AI exclusions is evidence that insurers are struggling to define, quantify and price some AI-related exposures. It does not mean that AI risks are inherently uninsurable. What it means is that, in the absence of sufficient evidence about the nature and distribution of those risks, insurers may prefer to exclude them rather than assume an exposure they cannot confidently underwrite. W.R. Berkley was the first insurer to introduce absolute AI exclusions, as a way to segregate AI risk, and since then, the number of insurers adopting AI exclusions has risen, most prominently, Berkshire Hathaway and Chubb[55] while new-AI specific coverages have risen.[56] The Insurance Services Office (ISO), who issue the standards forms insurers use, introduced optional generative AI endorsements CG 40 47 (broad exclusion), CG 40 48 (partial exclusion for personal and advertising injury coverage) and CG 35 08 (partial exclusion for Products/Completed Operations Liability coverage).[57] Users of LLMs can no longer assume that their output will be covered by insurance.[58] This growing skepticism is not confined to the United States. The European Insurance and Occupational Pensions Authority found in a survey that hallucinations were the top cited risk from the use of generative AI, ahead of cybersecurity risks and data protection issues.[59]

Three conclusions follow. First, a rational firm will have to more carefully consider the net benefit of using LLMs, which can be defined as,

$$\text{Net benefit} = \text{Production savings} - \text{Verification costs} - \text{Expected error losses}$$

The same LLM can be economically useful for one task and uneconomic for another, depending on the cost of establishing correctness. Second, manual verification is not scalable, and using it may reduce the verified share of LLM produced work. With a fixed review capacity, the share of output that anyone actually checks is:

$$\text{verified share} = \min\left(1,\ \frac{H}{v \cdot Q}\right)$$

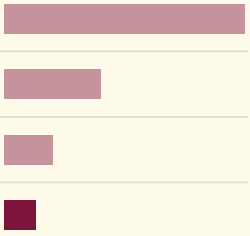
where H is reviewer hours, v is the hours needed to check one item, and Q is the number of items produced. This illustrative chart will serve to make the point:

Every fall in the cost of production shrinks the verified share, unless the time needed to check each item, v, falls with it. Human review cannot reduce v by much. Automated verification can. The third conclusion is that the economic value of LLMs in financial work depends not simply on how cheaply they can produce an answer, but on the total cost of producing an answer that a firm can safely rely upon. As the cost of generation falls, the relative importance of verification rises.

Figure 7: Eight times the output, and reviewers check 13% of it

Share of financial models a team can verify in a week: H ÷ (Q × v), at most 100%. In the last row, each check takes longer because the reviewer did not build the model.

Models a week (Q)	Reviewer hours (H)	Hours per check (v)	Verified share
50	40	0.50	100%
200	40	0.50	40%
400	40	0.50	20%
400	**40**	**0.75**	**13%**



Source and chart: Joseph Noko · josephnoko.com

This matches what firms report. Most firms now use LLMs, yet most report little effect. A recent survey of 9,000 senior executives from the United States, the United Kingdom, Germany, Australia, Canada, Mexico and South Africa, found that while 78% use LLMs, with 80% of executives spending an average of 2.1 hours per week using them, they have "had essentially no net effect on firm employment over the last 3 years" and "adoption has raised annual labor productivity growth by 0.2% over the last 3 years, a modest impact, but expect a substantially larger 0.6% per annum boost over the next 3 years".[60] A study of Danish administrative records found effects on earnings and hours of no more than 2%, even among intensive users and heavy investors.[61] Among CFOs, Gartner found that while 84% of finance organisations had implemented or were planning to implement LLMs, just 7% reported high or very high impact.[62] EY found that while 50% of CEOs believe LLMs are the biggest driver of productivity gains, 23% struggle to turn that into measurable financial outcomes.[63] This is not unique to LLM adoption: in the early days of the computer age, Robert Solow quipped, "you can see the computer age everywhere but in the productivity statistics."[64] The productivity figures are consistent with a broader problem: firms can adopt a powerful technology without yet possessing the organisational systems needed to extract its full value. General-purpose technologies always take time to show up in productivity, because firms must first build complementary processes, skills and intangible capital before their productivity benefits become fully visible.[65] Verification is one of those complementary systems. It is what a firm must build before it can trust what the new technology produces.

5.2 Verification determines the boundaries of the firm

Ronald Coase observed that whereas in markets, the allocation of factors of production is coordinated by the price mechanism, with supply adjusting to demand, and production to consumption, within firms, factors of production are allocated by entrepreneurs.[66] Why is

the price mechanism the coordinating instrument in one case and the entrepreneur in the other? *Why do firms exist?* He proposed that markets create transaction costs, such as the cost of finding suppliers, negotiating contracts, and coordinating multiple engagements, that make it cheaper to allocate factors of production within a firm than through the market. From a human point of view, LLMs will likely impact firm organisation in two ways: it will shred the size of some firms who can delegate tasks to LLMs rather than humans, while increasing the size of others, who can use LLMs to coordinate even greater numbers of human workers. In either scenario, they will increase the production of financial models.

LLMs have made producing financial models fast and cheap, and the tools that produce it are largely interchangeable. What has not become cheap is the cost of verification. Indeed, much of how markets and firms are organised is explained by the cost of *measuring* what is being exchanged.[67] When quality is costly to measure, people build institutions to economise on measurement. Verification of LLM output is a measurement cost. It decides how much work can be handed to LLMs, just as the cost of inspection once decided how much work could be handed to a contractor. Without automated verification checks, firms that should grow smaller will not do so, and firms that should grow bigger will be unable to do so.

This matters because the economics of delegating work to LLMs depends not only on the cost of producing financial models, but also on the cost of specifying, measuring and monitoring what has been produced. When those verification costs are high, firms have an incentive to retain control over the relevant activities rather than rely entirely on LLMs. Verification is therefore not merely a final quality-control step, it is also part of the economic infrastructure that determines how much financial work a firm can safely delegate to LLMs.

5.3 The migration of economic value

Clayton Christensen's "Law of Conservation of Attractive Profits"[68] offers a useful way to understand what may happen next. When modularity and commoditisation erode attractive profits at one stage of a value chain, the opportunity to earn them often emerges at an adjacent stage, where differentiation remains valuable.

LLMs are commoditising the production of financial work. Models that once required hours of specialised labour can increasingly be generated in minutes. As the tools improve and become interchangeable, the ability to produce a model may become less of a competitive advantage. Yet the value of a model depends not on its existence, but on whether anyone can rely on it. While production is becoming cheap, verification is not.

The bottleneck in LLM adoption is therefore shifting. The scarce resource is no longer the ability to produce a financial model, but the ability to establish, independently and at scale, whether it is correct. As production becomes commoditised, verification may become the adjacent stage at which differentiation, pricing power and attractive profits re-emerge, and it will be a new layer of financial infrastructure.

SECTION 6

Verification as risk management: the price of financial truth

Finance professionals already think about uncertainty for a living. Three ideas from the theory of risk show that verification is not overhead. It is risk management.

John Maynard Keynes distinguished the *probability* of a conclusion from the *weight* of the evidence behind it.[69] Two claims can be equally probable while one rests on a mountain of evidence and the other on a rumour. Verification does not change what an LLM produced, it adds weight to it.

Frank Knight distinguished *risk*, where the odds can be measured, from *uncertainty*, where they cannot.[70] A firm using LLM output whose error rate it has never measured faces uncertainty. It cannot price it, insure it or provision for it. Verification assigns a number to how trustworthy LLM output is, it turns an unknown into a known, it gives firms a sense of how much risk they are taking.

John Kelly showed that a gambler who wants to grow wealth over many bets must, above all, avoid the losses from which there is no recovery.[71] As I am fond of saying, much to my shame, lose half your capital and you need to double what remains to get back. In finance, a silent error in a board pack, a tender or a valuation is such a loss: rare, large and hard to reverse. As the damage compounds, avoiding rare large errors is worth far more than an average error rate suggests. It is not a black swan, an unpredictable event, but for many firms, it may prove a grey rhino, a risk that they ignore.

SECTION 7

Conclusion: proof-carrying financial work

For many problems, the asymmetry of verification is such that *finding* an answer is difficult but *checking* one is easier, so long as the answer comes with its evidence.[72] George Necula's work on *proof-carrying code* holds possibilities for finance: proof-carrying code are programs that arrive with a formal proof of their own safety, so the recipient need only check the formal proof to be assured of the safety of the program.[73] The recipient is freed from the need to check the work of the producer. This idea can be extended to finance as *proof-carrying financial work*: independent verifications for LLM-generated work.

Finance has the advantage that many other fields do not. Its outputs are subject to an extensive system of external constraints: published accounts, regulatory filings, ledgers, accounting standards, reconciliations, financial identities and mathematical relationships. For example, Luca Pacioli's description of double-entry accounting established a discipline in

which every transaction has corresponding entries and the books must balance. It does not guarantee that the entries are correct: errors can be recorded consistently, and a balanced set of accounts can still be wrong. Nevertheless, it illustrates that financial work is not an unconstrained exercise in producing plausible numbers, rather, much of it must satisfy relationships that can be independently tested.

A financial model, therefore, need not arrive as a collection of numbers and formulas that another analyst must reconstruct from scratch. Its material outputs can carry the evidence behind them: the source of each input, the reporting period and unit, the transformations applied, the formulas and dependencies involved, and the checks the result has passed. Where a number cannot be traced to an adequate source, or an assumption cannot be independently established, that limitation should be made explicit rather than concealed by a plausible output.

An independent system can then test the work against its sources, recalculate its formulas, examine its references, enforce accounting and financial relationships, and identify material claims that remain unsupported or unverified. It does not need to repeat the entire process of producing the work. What it needs to do is to establish which parts withstand scrutiny and which do not. Human reviewers can concentrate on exceptions, judgement and matters that cannot be settled mechanically.

The economic consequences are substantial. The value of LLMs in finance is not determined by the cost of generating a model, memo or presentation alone. It is determined by the cost of producing work that an organisation can safely rely upon. As the cost of generation falls, verification becomes an increasingly important constraint on the amount of work a firm can delegate. Without a cheaper way to establish reliability, firms must absorb the cost through manual review, bespoke controls and specialist intervention, thereby eroding the productivity and financial gains of algorithmic production, or accept risks they cannot properly assess.

Independent verification can make the checking of financial work systematic, repeatable and inexpensive, without making the producer responsible for certifying its own output. Firms can then use whichever LLMs best perform the production task while retaining independent control over the quality of what they receive. The more work LLMs produce, the more valuable that independent layer becomes.

As the fall in the price of algorithmic thought shifts the production of financial models to LLMs, increasing the volume and speed with which financial models are produced, the cost of verifying the correctness of those financial models will fall at a slower rate. Unable to scale human reviews and challenged by LLM reviews, firms will increasingly need cheap, automated, and independent verification systems in order to fully gain from the efficiencies offered by algorithmic production. Without independent verification systems, finance teams will continue to ship errors they cannot quantify and take risks they do not know.

APPENDIX

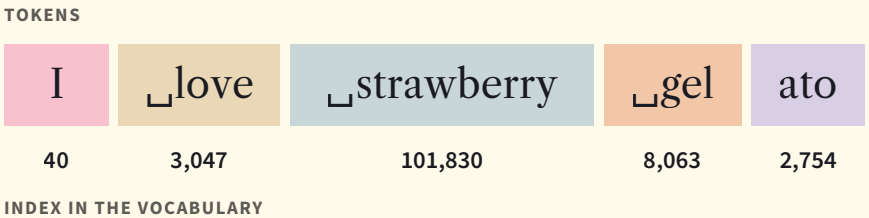
What LLMs are actually doing

LLMs are neural networks trained on vast amounts of text to predict what Stephen Wolfram calls a "reasonable continuation"[74] of a sequence of tokens, where by "reasonable" is meant what one would expect from observing the patterns of language, information and reasoning from its training data[75][76]. Their output reflects what was common in their training text. Naturally, they are most successful at tasks that can be expressed in natural language and that exploit patterns learned in training[77].

LLMs do not read words, they read tokens[78], which may be a whole word, part of a word, punctuation or another unit of data. The set of an LLM's different tokens is its vocabulary, and each token is assigned a number, its index. For example, OpenAI's GPT-5 models tokenise[79] the sentence, "I love strawberry gelato", into 5 tokens:

Figure 8: A model reads tokens, not words: "I love strawberry gelato" is 5 tokens

How OpenAI's GPT-5 tokeniser (o200k_base) splits the sentence. Each token has an index in the vocabulary; a leading space belongs to the token.



␣ marks a space. "strawberry" is one token; "gelato", rarer in the training text, is split in two.

Source: OpenAI tiktoken, o200k_base · Chart: Joseph Noko · josephnoko.com

Each session with an LLM has a context window. An LLM's context window sets the maximum number of tokens it can process in a particular interaction. This includes the instructions it receives, the conversation history, information returned by tools and the response it generates.[80] The frontier models have context windows of about one million tokens.

Take the nano-GPT[81] sorting model. nano-GPT, an educational device, is trained on many short sequences of letters, such as, *C B A B B C*, to sort the letters in alphabetical order, i.e. *A B B B C C*. It learns to do this through its internal training objective[82] of next-token prediction[83]. Its vocabulary, the letters *A*, *B*, and *C*, is indexed, with the numbers 0, 1 and 2, so that the sequence of letters, *C B A B B C* becomes the sequence of numbers *2 1 0 1 1 2*.

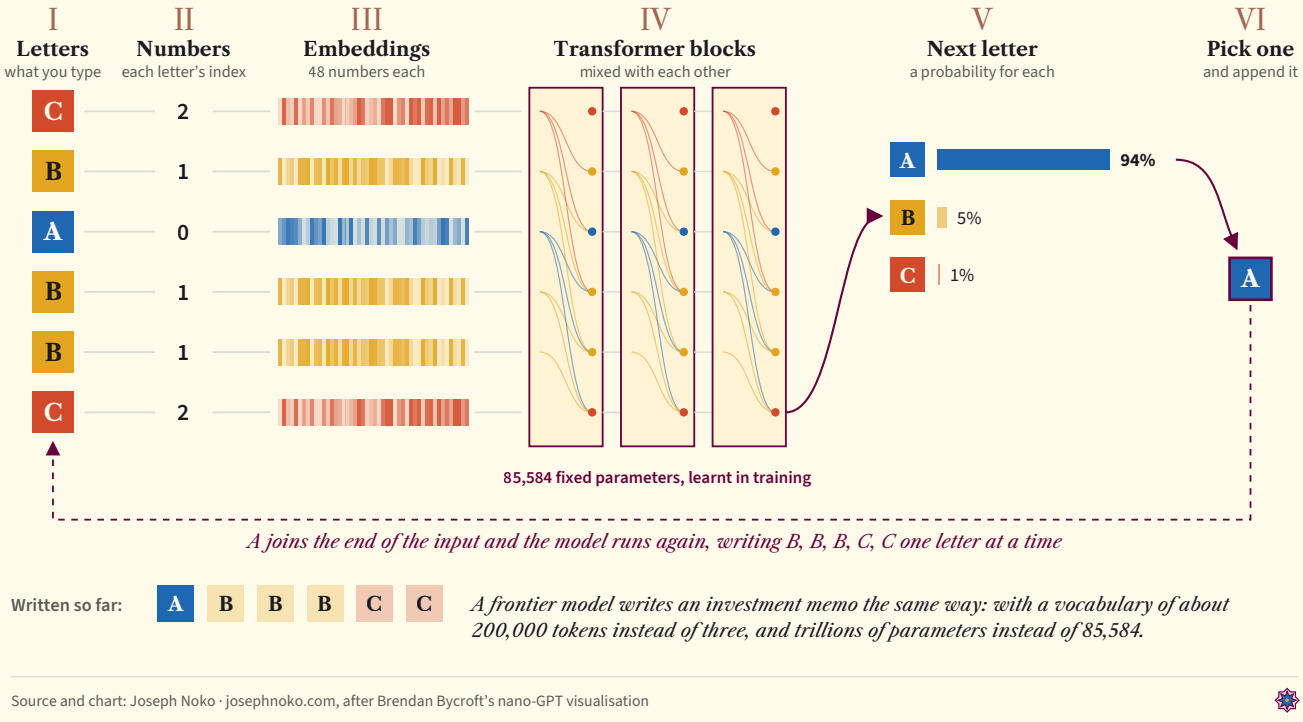
Inside nano-GPT, each token index is converted into a list of 48 numbers, called an *embedding*[83], which is the learned numerical representation of the token. The embeddings then pass through a stack of layers, called *transformer blocks*[84]. The transformer blocks process these representations, repeatedly combining information from different positions in the sequence. They do this using the model's learned *parameters*. The resulting representation therefore reflects not only the token itself, but also its context. These parameters collectively encode the patterns and capabilities the model has acquired. Larger models often have more parameters, but size alone does not determine capability. Architecture, training data, computing resources and training methods also matter.[85] At 85,000 parameters, nano-GPT is rather dumb compared to Moonshot AI's Kimi K3, with its 2.8 trillion parameters, of which approximately 104 billion are activated for each token.

What is generated is not an answer, it is not "truth", rather, it is a probability distribution over the tokens in its vocabulary, estimating how likely each one is to come next, given everything that has come before. A decoding procedure then selects a token from that distribution. That token is appended to the sequence, and the process repeats. After reading the sequence of letters, *C B A B B C*, nano-GPT judges "A" to be by far the *most likely* first letter of the sorted sequence. That token is added to the end of the input, and the whole process runs again to produce the next one: B, then B, then B, then C, then C. The sorted sequence A B B B C C is written one token at a time, each chosen by its probability given everything before it.

Most of today's major generative language models use this autoregressive process: they generate a sequence one token at a time, with each new token conditioned on the preceding context. The process is the same in principle whether the model has 85,000 parameters or trillions. What differs enormously is the sophistication of the patterns the model has learned and its ability to use them. In a deployed AI assistant, this process may be combined with other systems: web searches, databases, code execution, external tools and additional reasoning stages. These can substantially change what the assistant can do, but they do not remove the need to establish whether its final output is correct. AI assistants typically choose with an element of chance[86] rather than always taking the most likely token, which is why the same question can receive different answers.

Figure 9: How nano-GPT writes: score, pick, append, repeat

Sorting the letters C B A B B C. Each letter keeps its colour as it passes through the model. Probabilities illustrative.



Source and chart: Joseph Noko · josephnoko.com, after Brendan Bycroft's nano-GPT visualisation

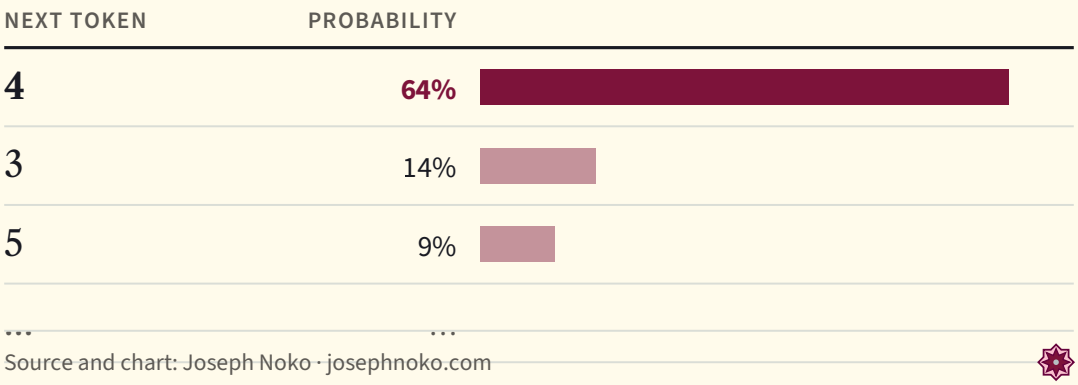
During pretraining, the model's parameters are initially set to numerical values, commonly through random initialisation. The model is shown sequences of text and learns to predict the next token. Its prediction is compared with the actual next token, producing an error measure called a *loss*. An optimisation process then adjusts the parameters to reduce that loss. Repeated across enormous quantities of training data, this process gradually produces a model capable of generating increasingly sophisticated sequences.[87] Repeated billions of times over vast amounts of text from the web, books and code, this simple exercise, next-token prediction, is how the model initially learns everything it knows.[82] Your phone's keyboard offers a simple illustration: it predicts which word is likely to follow what you have typed. An LLM applies a related principle at a vastly greater scale, using much more sophisticated learned representations to predict sequences of tokens. If you give it "Revenue for fiscal 2025 was $", it scores every token it knows:

It picks one, appends it, and scores again: ".", then "2", then " billion". If the model has not been given the relevant filing, or connected to a tool that can retrieve it, it has no direct access to the company's actual reported revenue. It can nevertheless generate a plausible figure. If its training data made "$4.2 billion" a likely continuation, it may produce that figure whether or not the company actually reported it. This is the fundamental distinction: generating a plausible statement is not the same as retrieving evidence that establishes its truth. An AI assistant can be connected to filings, databases and other external sources, but the accuracy of its answer still depends on whether it retrieves the right information and uses it correctly. Pre-training produces a model that can continue sequences of text, but it does

not by itself produce the kind of assistant people expect to interact with. Models therefore undergo additional stages of training, collectively called *post-training*. These can include instruction tuning, in which the model is trained on examples of instructions and appropriate responses, and preference-based training, in which responses are adjusted according to human or LLM evaluations.[88] That step is why a model answers rather than merely continues text, and why it is rewarded for confident answers.[89] These stages teach the model to follow instructions, answer questions and behave more like an assistant. They also shape which answers it tends to give. But they do not guarantee that its answers are true. Depending on how the model is trained and evaluated, it may still produce a confident answer when the evidence is insufficient.

Figure 10: Asked for a revenue figure, the model's best guess has a 64% chance

Probability a language model gives each candidate for the next token after "Revenue for fiscal 2025 was $". Every other token shares the remaining 13%; no filing is consulted.



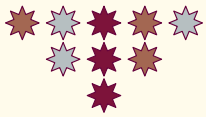
Source and chart: Joseph Noko · josephnoko.com

When training ends, the parameters are fixed. They collectively encode patterns and capabilities acquired through training. The LLM does not consult a complete archive of its training material when responding. It can be likened to a lossy zip file of its training data:[90] a compressed, imperfect summary from which the model reconstructs plausible text. It can reproduce information encountered during training, combine information in new ways, and sometimes generate plausible statements that were never present in its training data.

Over the past decade, the capabilities of language models have improved substantially, in part through increases in model size, training data and computing power. Research into scaling laws has shown that, under certain conditions, improvements in model performance can be predicted from increases in these resources.[91] Rich Sutton called this the "bitter lesson" of AI research: methods that leveraged advances in computing power did better than

those that sought to encode human expertise directly into systems[92][93]. More recently, researchers have developed techniques that allow models to devote additional computation to difficult problems before producing an answer. These include prompting models to work through intermediate steps and training specialised reasoning models using reinforcement learning. Such approaches have substantially improved performance on some mathematical, scientific and programming tasks.[94] They do not, however, guarantee that the reasoning is sound or that the final answer is correct.

An LLM is a powerful system for generating and transforming information. It can retrieve documents, execute calculations and construct sophisticated financial work when connected to the appropriate tools. But none of these capabilities, individually or collectively, establishes that the resulting work is correct. That requires a separate process: checking the evidence, calculations, assumptions and completeness of the output against criteria independent of the system that produced it.



Notes

1 Jason Wei, 'Asymmetry of Verification and Verifier's Rule', *Jason Wei*, 15 July 2025, https://www.jasonwei.net/blog/asymmetry-of-verification-and-verifiers-law.

2 Macabacus, *GenAI in Financial Services: Velocity and Verification* (Macabacus, 2026), 14, https://macabacus.com/lp/ai-report.

3 Christian Catalini et al., 'Some Simple Economics of AGI', arXiv:2602.20946, preprint, arXiv, 25 February 2026, https://doi.org/10.48550/arXiv.2602.20946.

4 Luca Pacioli, *Summa de Arithmetica, Geometria, Proportioni et Proportionalita* (Paganino Paganini, 1494), https://icaew.onlineculture.co.uk/ttp/?id=b2f80d6f-2e46-4cd5-9e8f-ac7a00ef4390&type=book.

5 Luke Emberson and David Roodman, 'The Plunging Price of Thought | Epoch AI', Epoch AI, 22 September 2026, https://epoch.ai/publications/the-plunging-price-of-thought.

6 William Stanley Jevons, *The Coal Question; An Inquiry Concerning the Progress of the Nation, and the Probable Exhaustion of Our Coal-Mines*, 2nd edition, revised (Macmillan and Co., 1866), https://oll.libertyfund.org/titles/jevons-the-coal-question.

7 Anthropic, *How Employment Is Changing at Anthropic*, 16 October 2026, https://www.anthropic.com/research/employment-at-anthropic.

8 Addy Osmani et al., 'How the Claude Code Team Does Code Review', Claude.dev, *Claude.Dev*, 15 October 2026, https://claude.dev/blog/solving-code-review-at-anthropic/.

9 William J. Baumol, 'Macroeconomics of Unbalanced Growth: The Anatomy of Urban Crisis', *The American Economic Review* 57, no. 3 (1967): 415–26.

10 Bradley Brown et al., 'Large Language Monkeys: Scaling Inference Compute with Repeated Sampling', arXiv:2407.21787, preprint, arXiv, 30 December 2024, https://doi.org/10.48550/arXiv.2407.21787.

11 Bradley Brown et al., 'Large Language Monkeys: Scaling Inference Compute with Repeated Sampling', arXiv:2407.21787, preprint, arXiv, 30 December 2024, https://doi.org/10.48550/arXiv.2407.21787.

12 Andrej Karpathy, 'Verifiability', *Karpathy*, 17 November 2025, https://karpathy.bearblog.dev/verifiability/.

13 Jason Wei, 'Asymmetry of Verification and Verifier's Rule', Jason Wei, 15 July 2025, https://www.jasonwei.net/blog/asymmetry-of-verification-and-verifiers-law.

14 It lacks a fifth property: its checks mostly pass or fail rather than ranking better answers above worse ones.

15 See the appendix for an explanation of how LLMs work.

16 Alfred Tarski, 'The Semantic Conception of Truth and the Foundations of Semantics', *Philosophy and Phenomenological Research* 4, no. 3 (1944): 341–76, https://doi.org/10.2307/2102968.

17 E. T. Jaynes, *Probability Theory: The Logic of Science*, ed. G. Larry Bretthorst (Cambridge University Press, 2003).

18 Michael Wooldridge et al., *AI Goes Mainstream: The Scaling Era and the Rise of Generative AI*, Study Panel Report, AI100: One Hundred Year Study on Artificial Intelligence (Stanford Institute for Human-Centered Artificial Intelligence, 2026), 127.

19 Michael Townsen Hicks et al., 'ChatGPT Is Bullshit | Ethics and Information Technology | Springer Nature Link', *Ethics and Information Technology* 26, no. 38 (2024), https://link.springer.com/article/10.1007/s10676-024-09775-5.

20 Adam Tauman Kalai et al., 'Why Language Models Hallucinate', arXiv:2509.04664, version 1, preprint, arXiv, 4 September 2025, https://doi.org/10.48550/arXiv.2509.04664.

21 Lexin Zhou et al., 'Larger and More Instructable Language Models Become Less Reliable', *Nature* 634, no. 8032 (2024): 61–68, https://doi.org/10.1038/s41586-024-07930-y.

22 Saturn, *Artificial Authority: Should You Trust AI for Financial Advice?*, Research Report (2026), 21, https://www.saturnos.com/report/artificial-authority.

23 Anthropic, *Zero Trust for AI Agents: A Security Framework for Deploying Autonomous AI Agents in the Enterprise.* (Anthropic, 2026), https://claude.com/resources/guides/zero-trust-for-ai-agents.

24 Anthropic, 'Introducing Citations on the Anthropic API', Claude, 23 June 2025, https://claude.com/resources/articles/introducing-citations-api.

25 Anthropic, 'Reduce Hallucinations', Claude Platform Docs, accessed 18 November 2026, https://platform.claude.com/docs/en/test-and-evaluate/strengthen-guardrails/reduce-hallucinations; Anthropic, 'Building Effective Agents', Anthropic, *Engineering at Anthropic*, 19 December 2024, https://www.anthropic.com/engineering/building-effective-agents.

26 Ziwei Xu et al., 'Hallucination Is Inevitable: An Innate Limitation of Large Language Models', *arXiv*, 22 January 2024, https://arxiv.org/pdf/2401.11817.

27 'Trust, but verify', *Wikipedia* (Wikimedia Foundation, 9 July 2026), https://en.wikipedia.org/wiki/Trust%2C_but_verify.

28 ClaudeDevs, Post, X, 9 October 2026, https://x.com/ClaudeDevs/status/2108591334449684643.

29 $0.99^{20} \approx 0.82$

30 $0.95^{20} \approx 0.36$

31 Max Kan et al., 'GDPx: Measuring Frontier Model Capabilities Across the Most Valuable Domains', *SemiAnalysis*, 2 November 2026, https://newsletter.semianalysis.com/p/gdpx-measuring-frontier-model-capabilities.

32 SemiAnalysis, 'GDPx', SemiAnalysis, 2026, https://gdpx.semianalysis.com/?domain=finance.

33 Nitish Garg, 'GDPx Benchmark: Claude Opus 5.5 Leads at Just 12%', *CellCog*, 2 November 2026, https://cellcog.ai/blog/gdpx-benchmark/.

34 'Solomon Shereshevsky', *Wikipedia*, 23 September 2026, https://en.wikipedia.org/w/index.php?title=Solomon_Shereshevsky&oldid=1376272945.

35 Anthropic, 'Building Effective Agents', Anthropic, *Engineering at Anthropic*, 19 December 2024, https://www.anthropic.com/engineering/building-effective-agents; Anthropic, 'Effective Context Engineering for AI Agents', Anthropic, *Engineering at Anthropic*, 29 September 2025, https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents.

36 Anthropic, 'Building Effective Agents', Anthropic, *Engineering at Anthropic*, 19 December 2024, https://www.anthropic.com/engineering/building-effective-agents; Anthropic, 'Effective Context Engineering for AI Agents', Anthropic, *Engineering at Anthropic*, 29 September 2025, https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents.

37 Tomek Korbak et al., 'Chain of Thought Monitorability: A New and Fragile Opportunity for AI Safety', arXiv:2507.11473, preprint, arXiv, 7 December 2025, https://doi.org/10.48550/arXiv.2507.11473.

38 Miles Turpin et al., 'Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting', arXiv:2305.04388, preprint, arXiv, 9 December 2023, https://doi.org/10.48550/arXiv.2305.04388.

39 Yanda Chen et al., 'Reasoning Models Don't Always Say What They Think', 2505.05410, preprint, arXiv, 8 May 2025, https://arxiv.org/abs/2505.05410.

40 Alexander Panfilov et al., 'Stealing Reasoning Traces from Proprietary LLM APIs', arXiv:2608.09867, preprint, arXiv, 10 August 2026, https://doi.org/10.48550/arXiv.2608.09867.

41 Richard E. Nisbett and Timothy DeCamp Wilson, 'Telling More than We Can Know: Verbal Reports on Mental Processes.', *Psychological Review* 84, no. 3 (1977): 231–59, https://doi.org/10.1037/0033-295X.84.3.231.

42 Tomek Korbak et al., 'Chain of Thought Monitorability: A New and Fragile Opportunity for AI Safety', arXiv:2507.11473, preprint, arXiv, 7 December 2025, https://doi.org/10.48550/arXiv.2507.11473.

43 Parshin Shojaee et al., 'The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity', arXiv:2506.06941, preprint, arXiv, 20 November 2025, https://doi.org/10.48550/arXiv.2506.06941.

44 C. Opus and A. Lawsen, 'The Illusion of the Illusion of Thinking: A Comment on Shojaee et al. (2025)', 2506.09250v1, preprint, arXiv, 10 June 2025, https://arxiv.org/html/2506.09250v1.

45 METR, 'Recent Frontier Models Are Reward Hacking', Substack newsletter, *METR*, 5 June 2025, https://metr.substack.com/p/2025-06-05-recent-reward-hacking; Yerin Hwang et al., 'Can You Trick the Grader? Adversarial Persuasion of LLM Judges', in *Findings of the Association for Computational Linguistics: EMNLP 2025*, ed. Christos Christodoulopoulos et al. (Association for Computational Linguistics, 2025), https://doi.org/10.18653/v1/2025.findings-emnlp.790; David Demitri Africa and Hans Ethan Ting, 'Does Self-Evaluation Enable Wireheading in Language Models?', arXiv:2511.23092, preprint, arXiv, 1 December 2025, https://doi.org/10.48550/arXiv.2511.23092.

46 Robert Tjarko Lange et al., 'The Agentic Scientist: Orchestrating Coding Agents for Automated Scientific (Co-)Discovery', arXiv:2610.18933, preprint, arXiv, 13 October 2026, https://doi.org/10.48550/arXiv.2610.18933.

47 Herbert A. Simon, 'Designing Organizations for an Information-Rich World', in *Computers, Communications, and the Public Interest*, ed. H. Greenberger (The Johns Hopkins Press, 1971), https://gwern.net/doc/design/1971-simon.pdf.

48 Raja Parasuraman and Dietrich H. Manzey, 'Complacency and Bias in Human Use of Automation: An Attentional Integration', *Human Factors* 52, no. 3 (2010): 381–410, https://doi.org/10.1177/0018720810376055.

49 Fabrizio Dell'Acqua et al., 'Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of Artificial Intelligence on Knowledge Worker Productivity and Quality', *Harvard Business School Technology & Operations Mgt. Unit Working Paper No. 24-013, The Wharton School Research Paper*, nos 24–013 (September 2023): 22.

50 Amaç Herdağdelen et al., 'How AI Agents Shape Experts' Judgment Calls', Anthropic, *Research*, 27 October 2026, https://www.anthropic.com/research/agents-judgment-calls.

51 George A. Akerlof, 'The Market for "Lemons": Quality Uncertainty and the Market Mechanism', *The Quarterly Journal of Economics* 84, no. 3 (1970): 488–500.

52 Colin Jarvis, 'What FDE Means at OpenAI', OpenAI, 22 October 2026, https://openai.com/index/forward-deployed-engineering-means/; Anthropic, 'Anthropic Invests $100 Million to Train 10,000 Engineers and Tackle the Enterprise AI Talent Gap', Anthropic, 2 October 2026, https://www.anthropic.com/news/claude-frontier-academy.

53 Ana Maria Constantin, 'OpenAI's Colin Jarvis Says Enterprise AI Is Stuck on Deployment, Not Models', The Next Web, 23 September 2026, https://thenextweb.com/news/openai-colin-jarvis-enterprise-ai-deployment-fde-humanx.

54 Mia Macgregor and Henry Gale, 'More than 60 P&C Insurance Groups File to Adopt AI Exclusions', The Insurer, 23 July 2026, https://www.theinsurer.com/ti/analysis/more-than-60-pc-insurance-groups-file-to-adopt-ai-exclusions-2026-07-23/.

55 The Intelligence Council, 'Berkshire and Chubb Just Won Approval to Drop AI Coverage', *P&C Insurance Executive Intelligence*, 10 June 2026, https://insuranceintel.substack.com/p/berkshire-and-chubb-just-won-approval.

56 Geoffrey Fehling et al., 'Berkley Introduces "Absolute" AI Exclusion in Liability Policies', *The National Law Review*, 28 May 2025, https://natlawreview.com/article/continued-proliferation-ai-exclusions.

57 Jay Konkel and Hadhy Ayaz, 'AI Exclusions in Insurance Policies: Broad Language, Uncertain Impact', Artificial Intelligence, *Policyholder Pulse*, 13 April 2026, https://www.policyholderpulse.com/ai-exclusions-insurance-policies/.

58 Cloud Security Alliance AI Safety Initiative, *The End of Silent AI Coverage: Exclusions Outrun Risk Transfer* (Cloud Security Alliance, 2026), https://labs.cloudsecurityalliance.org/research/csa-research-note-ai-insurance-exclusion-wave-20260830-csa-s/.

59 European Insurance and Occupational Pensions Authority, 'EIOPA Survey on Generative AI Shows Swift but Cautious Adoption among Europe's Insurers - European Insurance and Occupational Pensions Authority', European Insurance and Occupational Pensions Authority, 2 February 2026, https://www.eiopa.europa.eu/eiopa-survey-generative-ai-shows-swift-cautious-adoption-among-europes-insurers-2026-02-02_en.

60 Ivan Yotzov et al., 'Firm Data on AI', Working Paper no. 34836, Working Paper Series (National Bureau of Economic Research, February 2026), https://doi.org/10.3386/w34836.

61 Anders Humlum and Emilie Vestergaard, 'Large Language Models, Small Labor Market Effects by Anders Humlum, Emilie Vestergaard :: SSRN', preprint, SSRN, 17 April 2025, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5219933.

62 Rob van der Meulen, 'Gartner Says CFOs Need Structured Finance AI Roadmaps', Gartner, 8 June 2026, https://www.gartner.com/en/newsroom/press-releases/2026-06-08-gartner-says-cfos-need-structured-finance-ai-roadmaps.

63 Laura Furness, 'AI Drives Productivity, but CEOs Struggle to Turn Gains into Growth', EY, London, United Kingdom, 1 October 2026, https://www.ey.com/en_gl/newsroom/2026/10/ai-drives-productivity-but-ceos-struggle-to-turn-gains-into-grow.

64 R. Solow, 'We'd Better Watch out', *New York Times Book Review*, 1987, https://www.semanticscholar.org/paper/We%E2%80%99d-better-watch-out-Solow/cef149b3dbdaa85f74b114c2c7832982f23bcbf0.

65 Erik Brynjolfsson et al., 'The Productivity J-Curve: How Intangibles Complement General Purpose Technologies', *American Economic Journal: Macroeconomics* 13, no. 1 (2021): 333–72, https://doi.org/10.1257/mac.20180386; Erik Brynjolfsson et al., 'Generative AI at Work', Working Paper no. 31161, Working Paper Series (National Bureau of Economic Research, April 2023), https://doi.org/10.3386/w31161.

66 Ronald H. Coase, 'The Nature of the Firm', *Economica* 4, no. 16 (1937): 386–405, https://doi.org/10.2307/2626876.

67 Yoram Barzel, 'Measurement Cost and the Organization of Markets', *The Journal of Law and Economics* 25, no. 1 (1982): 27–48, https://doi.org/10.1086/467005.

68 Clayton M. Christensen and Michael E. Raynor, *The Innovator's Solution, with a New Foreword: Creating and Sustaining Successful Growth* (2003; Harvard Business Review Press, Boston, Massachusetts, 2024).

69 John Maynard Keynes, *A Treatise on Probability*, The Collected Writings of John Maynard Keynes, Volume III (Cambridge University Press for the Royal Economic Society, 2013).

70 Frank H. Knight, *Risk, Uncertainty and Profit* (Hart, Schaffner and Marx, 1921), https://oll.libertyfund.org/titles/knight-risk-uncertainty-and-profit.

71 John Larry Kelly, Jr., 'A New Interpretation of Information Rate', *The Bell System Technical Journal* 35, no. 4 (1956): 917–26, https://doi.org/10.1002/j.1538-7305.1956.tb03809.x.

72 Stephen A. Cook, 'The Complexity of Theorem-Proving Procedures', *Proceedings of the Third Annual ACM Symposium on Theory of Computing* (New York, NY, USA), STOC '71, 3 May 1971, 151–58, https://doi.org/10.1145/800157.805047.

73 George C. Necula, 'Proof-Carrying Code', *Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (New York, NY, USA), POPL '97, 1 January 1997, 106–19, https://doi.org/10.1145/263699.263712.

74 Stephen Wolfram, 'What Is ChatGPT Doing ... and Why Does It Work?', Stephen Wolfram, *Writings*, 14 February 2023, https://writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work/.

75 Prediction here is different from the reflexive, or game-theoretic sense where an entity predicts what will occur after it acts. AlphaGo makes predictions in this sense, while LLMs do not.

76 Claude E. Shannon, 'Prediction and Entropy of Printed English', *The Bell System Technical Journal* 30, no. 1 (1951): 50–64, https://doi.org/10.1002/j.1538-7305.1951.tb01366.x.

77 Rich Sutton, 'The Bitter Lesson', Personal, *Incomplete Ideas*, 13 March 2019, http://www.incompleteideas.net/IncIdeas/BitterLesson.html.

78 Rico Sennrich et al., 'Neural Machine Translation of Rare Words with Subword Units', arXiv:1508.07909, preprint, arXiv, 10 June 2016, https://doi.org/10.48550/arXiv.1508.07909.

79 OpenAI, 'Tokenizer', OpenAI Platform, https://platform.openai.com/tokenizer.

80 Nelson F. Liu et al., 'Lost in the Middle: How Language Models Use Long Contexts', arXiv:2307.03172, preprint, arXiv, 20 November 2023, https://doi.org/10.48550/arXiv.2307.03172.

81 Brendan Bycroft, 'LLM Visualization', Bbycroft, https://bbycroft.net/llm.

82 Rich Sutton (2016) says, "Intelligence is the computational part of the ability to achieve goals. A goal achieving system is one that is more usefully understood in terms of outcomes than in terms of mechanisms". I hold that view and take "goals" to refer to objectives with real-world outcomes. LLMs do not have goals in that sense. They are exclusively focused on next-token prediction. Cf: Rich Sutton, 'The Definition of Intelligence', *Incomplete Ideas*, 9 July 2016, http://incompleteideas.net/IncIdeas/DefinitionOfIntelligence.html.

83 Yoshua Bengio et al., 'A Neural Probabilistic Language Model', *Journal of Machine Learning Research* 3, no. Feb (2003): 1137–55.

84 Ashish Vaswani et al., 'Attention Is All You Need', version 7, preprint, arXiv, 2017, https://doi.org/10.48550/ARXIV.1706.03762.

85 Speed is sacrificed in pursuit of intelligence.

86 Ari Holtzman et al., 'The Curious Case of Neural Text Degeneration', arXiv:1904.09751, preprint, arXiv, 14 February 2020, https://doi.org/10.48550/arXiv.1904.09751.

87 David E. Rumelhart et al., 'Learning Representations by Back-Propagating Errors | Nature', *Nature* 323 (October 1986): 533–36.

88 Long Ouyang et al., 'Training Language Models to Follow Instructions with Human Feedback', arXiv:2203.02155, preprint, arXiv, 4 March 2022, https://doi.org/10.48550/arXiv.2203.02155.

89 Adam Tauman Kalai et al., 'Why Language Models Hallucinate', arXiv:2509.04664, version 1, preprint, arXiv, 4 September 2025, https://doi.org/10.48550/arXiv.2509.04664; Lexin Zhou et al., 'Larger and More Instructable Language Models Become Less Reliable', *Nature* 634, no. 8032 (2024): 61–68, https://doi.org/10.1038/s41586-024-07930-y.

90 Grégoire Delétang et al., 'Language Modeling Is Compression', arXiv:2309.10668, preprint, arXiv, 18 March 2024, https://doi.org/10.48550/arXiv.2309.10668; Andrej Karpathy, *Andrej Karpathy: '99% Use AI & LLMs Wrong' (Here's How He Does It)* (hyperfixation on AI | beamnxw, n.d.), https://www.youtube.com/watch?v=7YEHs2SMaVQ&t=169s.

91 Tom B. Brown et al., 'Language Models Are Few-Shot Learners', arXiv:2005.14165, preprint, arXiv, 22 July 2020, https://doi.org/10.48550/arXiv.2005.14165; Alec Radford et al., *Language Models Are Unsupervised Multitask Learners*, 2019, 1–24; Jared Kaplan et al., 'Scaling Laws for Neural Language Models', arXiv:2001.08361, preprint, arXiv, 23 January 2020, https://doi.org/10.48550/arXiv.2001.08361; Jordan Hoffmann et al., 'Training Compute-Optimal Large Language Models', arXiv:2203.15556, preprint, arXiv, 29 March 2022, https://doi.org/10.48550/arXiv.2203.15556.

92 Rich Sutton, 'The Bitter Lesson', Personal, *Incomplete Ideas*, 13 March 2019, http://www.incompleteideas.net/IncIdeas/BitterLesson.html.

93 Intriguingly, in an interview with Dwarkesh Patel, Sutton expressed doubts that LLMs are a case of AI thriving by leaning on computation. While LLMs use massive computation and scale with it, they also scale with human expertise, given that their inputs are human text, and human experts are often used to train the models, so their limit is set by how much human data exists. They are then a hybrid. They perform better the more human knowledge goes in, which, as he puts it, 'feels good', the kind of dangerous locking-in that the Bitter Lesson warns about. Sutton expects systems that learn from experience to overtake LLMs. Experience is not capped by what people have written. If he is right, the story becomes 'another instance of the bitter lesson'. The approach that used human knowledge loses to one that learns from experience and computation.

94 Jason Wei et al., 'Chain-of-Thought Prompting Elicits Reasoning in Large Language Models', arXiv: 2201.11903, preprint, arXiv, 10 January 2023, https://doi.org/10.48550/arXiv.2201.11903; Charlie Victor Snell et al., 'Scaling LLM Test-Time Compute Optimally Can Be More Effective than Scaling Parameters for Reasoning', paper presented at The Thirteenth International Conference on Learning Representations, 4 October 2024, https://openreview.net/forum?id=4FWAwZtd2n.

HOW TO CITE

Joseph Noko, 'The Necessity of Proof: LLMs and the Future of Financial Work', josephnoko.com, 18 November 2026.

```
@misc{noko2026necessity,
  author = {Joseph Noko},
  title  = {The Necessity of Proof: LLMs and the Future of Financial Work},
  year   = {2026},
  month  = nov,
  url    = {https://josephnoko.com},
  note   = {Published 18 November 2026}
}
```

Set in Libre Caslon and Source Sans 3. Figures are drawn by the author unless a source is named in the figure. © Joseph Noko 2026.