Forwards or Backwards

https://forwardorbackwards.com



How Large Language Models Actually Work

Watch on YouTube: https://www.youtube.com/watch?v=BwHU84ImfhU

Chapters

1. A thing made entirely of numbers
2. It does not see words
3. Attention, which is the whole trick
4. The block, and then the block again
5. The residual stream
6. Where the numbers come from
7. The second training, and the people in it
8. The building

9. Why you cannot simply read it
10. Building an instrument
11. What the largest attempt actually found
12. From parts to paths
13. The model's own account of itself
14. The experiment at the centre of this film
15. The disagreement, which nobody should pretend is settled
16. And whether this can work at all
17. What this film has not said
18. The strangeness, restated

1. A thing made entirely of numbers



There is a kind of object that almost never exists. An object that human beings built on purpose, from parts they chose, according to rules they wrote down — and then had to study afterwards, from the outside, the way you would study a beetle or a weather system, to find out what it does.

A large language model is that object.

Everything inside one is a number. Not a metaphor for a number. A number, stored in a file, that you could print out on paper if you had enough paper and enough patience. The arithmetic that turns those numbers into the next word of a sentence is not secret, not disputed, and not complicated in the way a jet engine is complicated. A graduate student can write it out in an afternoon. It is multiplication and addition, arranged in a particular order, repeated a very large number of times.

And the people who build these systems — the people who chose the arrangement, who own the numbers, who can read every one of them whenever they like — still run experiments on the finished model to find out what it is doing.

That is the strange fact this film is about. Not that the machine is mysterious because it is hidden. It is mysterious because it is completely visible and we still cannot read it.

We are going to do this in two halves. First, the mechanism: what is actually in there, what the arithmetic does, where the numbers come from, and what the building it runs in looks like. Then the harder half: what people have found when they went in with instruments, what those instruments can and cannot see, and where the honest disagreement sits right now.

One disclosure before we start, because it matters and it will be obvious anyway. A great deal of the best published work on looking inside these models comes from Anthropic, and Anthropic makes one of the models. Where a finding comes from a lab that also sells the thing being studied, this film says so on screen. It also cites work from outside that lab, because this is not a field with one laboratory in it.

2. It does not see words



Start with what actually arrives at the model, because it is not what you typed.

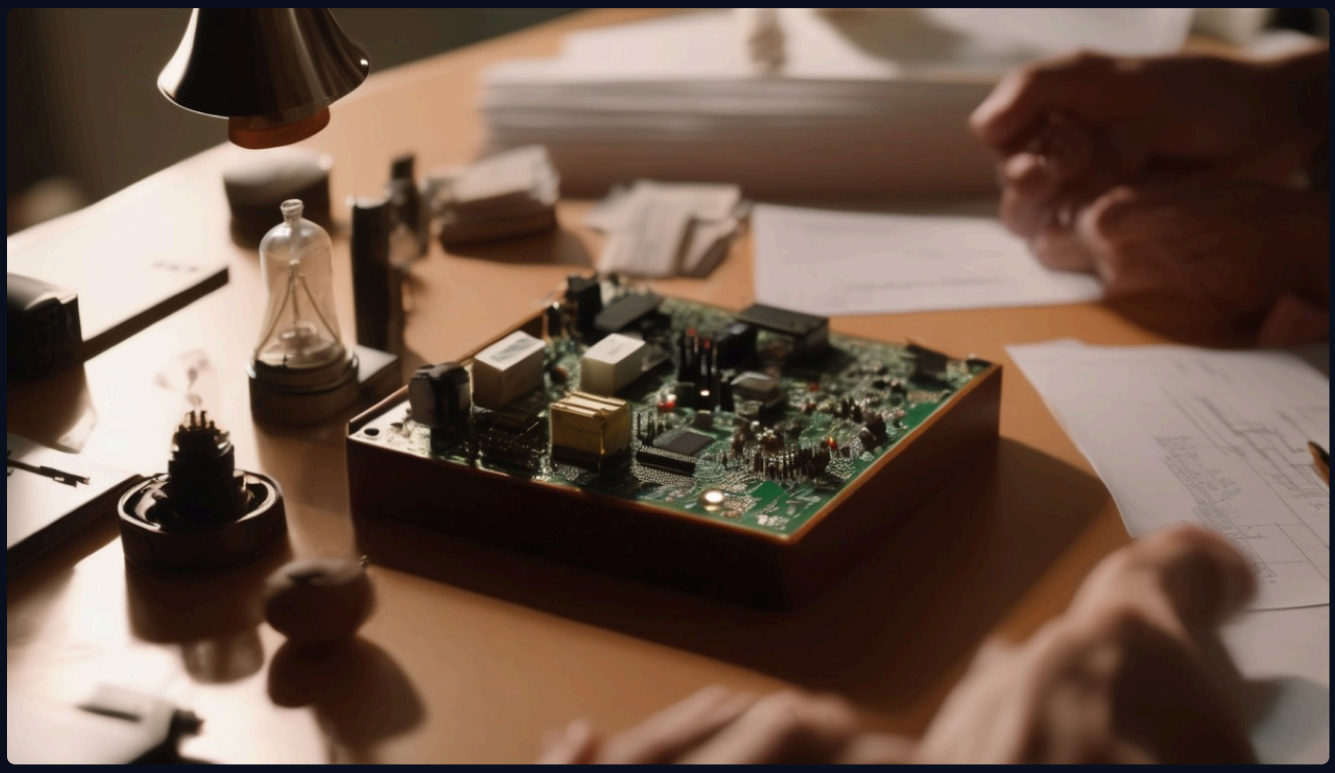
Text is cut into pieces called tokens. A token is sometimes a whole word, often a fragment, sometimes a space and the letters after it, occasionally a single character. The word "unbelievable" might arrive as three pieces. The word "the" arrives as one. The cutting is done by a fixed table built before training, and the table is chosen so that common text uses few pieces and rare text uses more.

Then each token is replaced by a number — its row in the table. And then that number is replaced by a list of numbers, several thousand of them, which is the model's internal representation of that piece of text.

That list is the first thing worth pausing on. It is not a definition. Nothing in it says "the, definite article." It is a position — a direction in a space with several thousand directions in it. The model learns where to put each token during training, and the useful property that comes out of that is that positions end up meaningfully arranged. Tokens used in similar ways land near each other.

So the input to the machine is a sequence of positions in a very large space. Not words. Not meanings. Coordinates.

3. Attention, which is the whole trick



Here is the part that made the current generation possible, and it is worth going slowly, because almost everything else is bookkeeping around it.

Each token in the sequence needs to be understood in the light of the others. In the sentence "the trophy would not fit in the suitcase because it was too big," the word "it" has to be connected to the trophy rather than the suitcase, and nothing about the word "it" by itself can tell you that.

The mechanism that does this is called attention, and it works like a very literal filing system.

For every token, the model computes three things from its position. A query — what this token is looking for. A key — what this token offers to others looking. And a value — what this token would hand over if asked.

Then every token's query is compared against every other token's key. Where a query and a key line up, the match is strong. Where they do not, it is weak. Those match strengths are turned into weights that add up to one, and each token pulls in a blend of the other tokens' values, mixed according to those weights.

That is it. Compare, weight, blend.

Two things follow from it that explain most of what you have noticed about these systems.

The first is that this comparison happens between every token and every other token, all at once. That is why these models want the enormous parallel machines we will look at shortly, and it is why the

cost of a long prompt grows faster than its length.

The second is subtler and more important. Nowhere in that description did anyone specify what the queries and keys should mean. They are not designed. They are numbers that came out of training, and the only thing that shaped them was whether the resulting blend helped predict text. Whatever "look back for the noun this pronoun refers to" is, inside the model, nobody wrote it. It formed.

4. The block, and then the block again



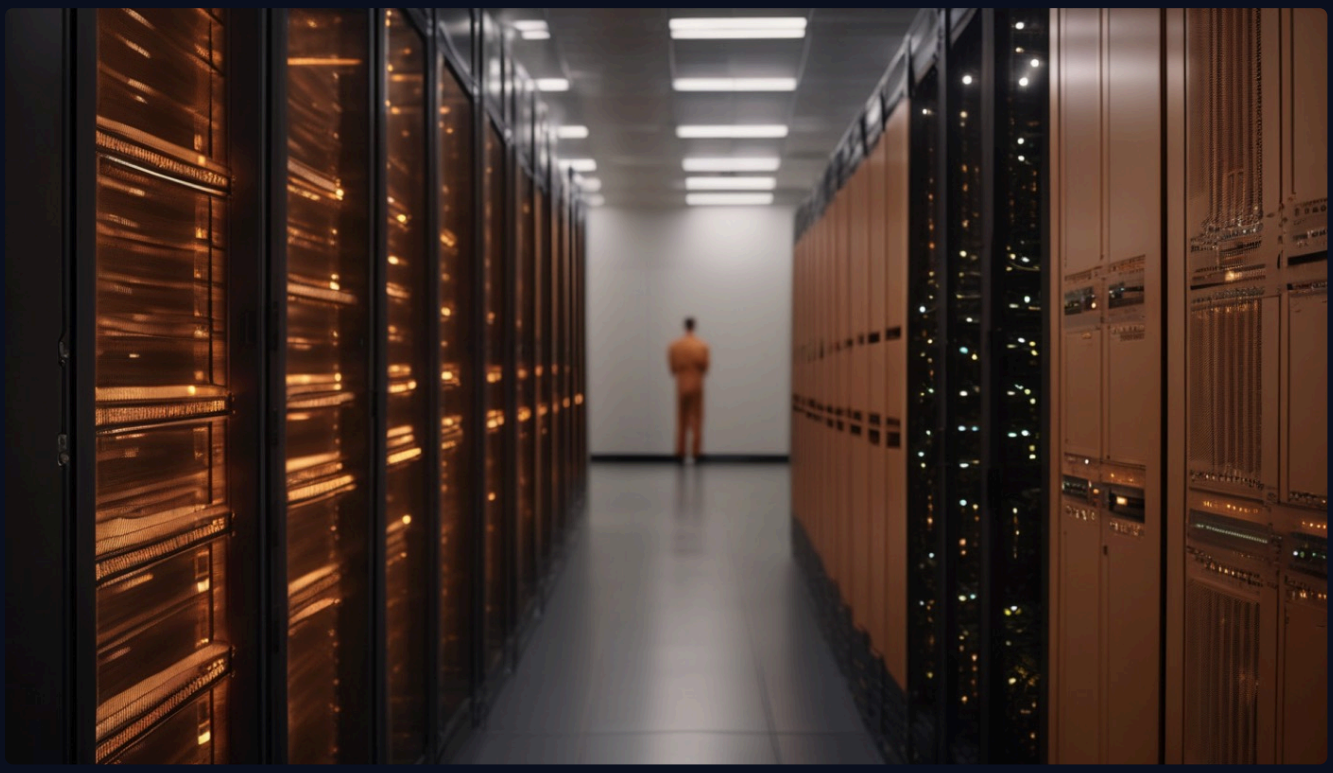
Attention on its own does one job: it moves information between positions. The other job — doing something with the information once it has arrived — belongs to a second component, a small network applied to each position separately.

Together, an attention layer and that per-position network make a block. And a language model is that block, repeated. Dozens of times in a small model. Far more in a large one.

Every block does the same two things in the same order: look around, then think locally. The next block does it again, on whatever the last one left behind. The blocks do not have different jobs assigned to them; they have identical structure and different numbers.

And because they are identical in structure, there is no floor plan. There is no layer labelled grammar and no layer labelled facts. If the model has something that behaves like a grammar module, it is somewhere in there, distributed across a stack of identical parts, and finding it is an experimental question rather than a matter of consulting the design.

5. The residual stream



There is one more structural idea, and it is the one that makes the whole stack hang together.

Running through the model, from the first block to the last, is a channel that every block reads from and writes to. Each block does not replace what came before it. It adds to it.

The useful way to picture it is a long document passed down a line of people. Each person reads the whole document, writes something in the margin, and passes it on. Nobody rewrites the page. Everything accumulates.

That channel is called the residual stream, and it has a consequence that matters enormously for the second half of this film. Because every block writes into the same shared space, a single direction in that space can carry a contribution from many different blocks, and a single block can write many different things into the same region. The information is not filed in separate drawers. It is superimposed.

Hold that word. Superimposed. It is the reason that reading the model is hard, and it is the thing the instruments in the second half were built to undo.

At the end of the stack, the accumulated document is compared against the token table one last time, and the result is a score for every possible next token. Those scores become probabilities. One token is selected. It is added to the sequence. And the entire process runs again from the beginning, for the next token, and the next.

That is the whole mechanism. Cut the text into pieces, put the pieces in a space, let them compare and blend, repeat that a few dozen times, and read off a probability for what comes next.

6. Where the numbers come from



Every number in that machine started as noise.

Training is a loop with three steps, and the loop is as unglamorous as the mechanism. Show the model some real text with a piece hidden. Let it predict the hidden piece. Measure how wrong it was, and nudge every number in the model a very small amount in the direction that would have made it less wrong.

Then do that again. And again. Over a quantity of text no person could read in a thousand lifetimes.

Nobody specifies what any individual number should become. The only instruction the system ever receives is: be less wrong about the next piece of text. Everything the finished model appears to know — that Paris is in France, that a semicolon is not a comma, that a limerick has a shape — arrived as a side effect of that one instruction, because getting those things right makes the next piece of text more predictable.

This is the bit that people find hardest to believe, and it is worth stating plainly rather than dressing up. There is no stage at which anybody teaches the model a fact. There is a stage at which being able to behave as though it holds that fact reduces prediction error, and the numbers move accordingly.

7. The second training, and the people in it



A model trained only on that loop is not the thing you talk to. It will continue text, but it has no particular reason to answer a question rather than write three more questions like it.

The second stage is where it learns to be a respondent. Human beings write examples of the behaviour wanted. Human beings compare pairs of the model's own outputs and mark which is better. Those comparisons train a smaller model that scores outputs, and that scorer is used to push the big model toward the kind of answer people preferred.

It is worth being concrete about who does this, because the popular account of these systems skips it entirely. It is done at desks, by people, reading output and making judgements, in large numbers, for pay. The manner these systems have — the hedging, the structure, the politeness, the refusals — is not an emergent property of predicting text. It was selected, by people, comparing one answer against another.

The mechanism half of this film ends there. Tokens, attention, blocks, a shared stream, a prediction loop, and a second pass shaped by human comparison. All of it specified. All of it published. None of it in dispute.

8. The building



Before the difficult half, it is worth looking at where this actually happens, because the abstraction hides something enormous.

A language model runs in a hall. Rows of steel cabinets, each holding a stack of accelerator cards, each card a slab of silicon in a metal frame with a heatsink on top. Cable trays overhead. Cooling pipes, because a rack of these draws enough power to heat a house and all of it comes back out as heat. Air handling at one end of the hall, and somewhere outside, a substation that exists because the building's demand is a utility-scale load rather than an office's.

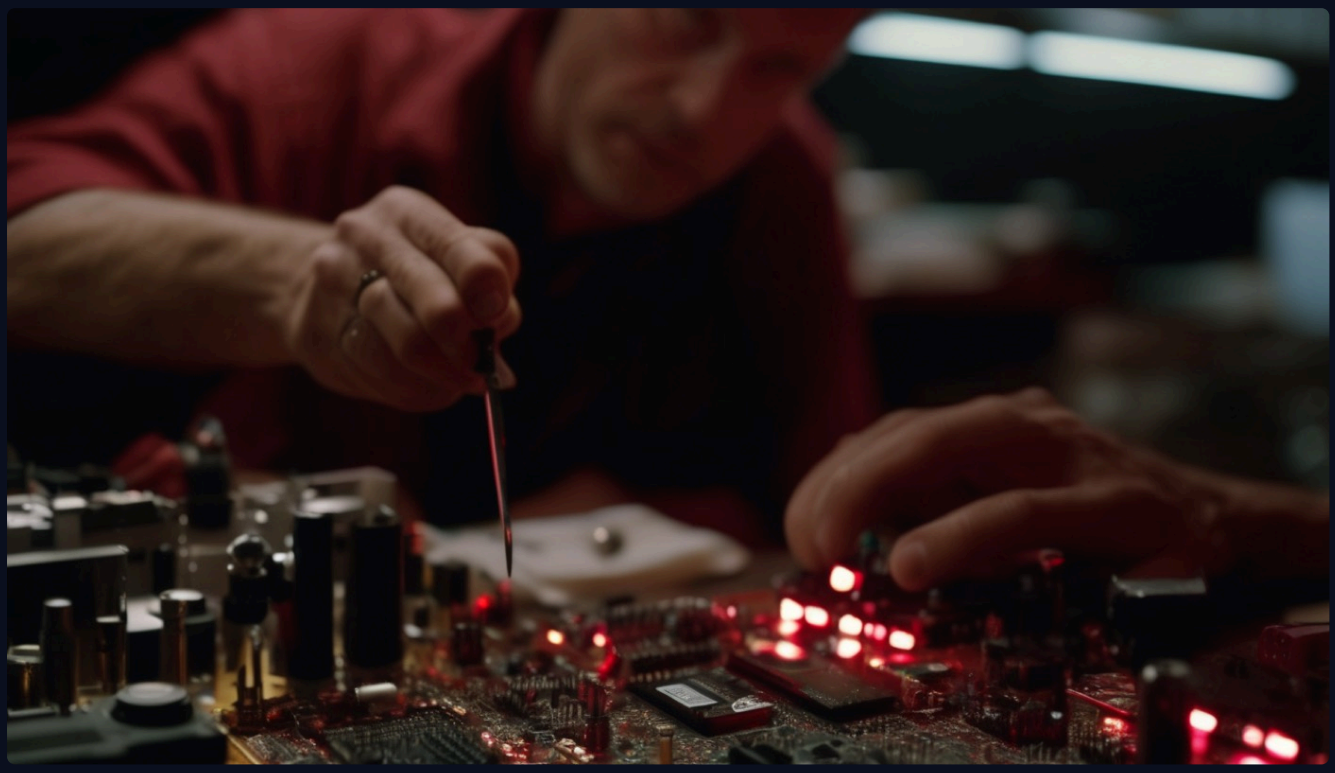
The silicon itself began as a disc of purified crystal, patterned in a fabrication plant through dozens of stages, cut into individual dies, packaged, tested, and mounted. The largest of those chips are among the most precisely manufactured objects made anywhere.

None of this is incidental to the subject. The reason attention wants to compare every token against every other token at once is that the hardware is very good at exactly that shape of arithmetic and very bad at anything sequential. The architecture and the machine were chosen for each other.

And there is a person at a keyboard somewhere in that building, watching a training run, who can open any file in it.

Which brings us to the problem.

9. Why you cannot simply read it



Everything described so far is available. So the obvious question is why anybody has to run experiments at all. Open the file. Look at the numbers. Work out what it does.

People have tried, and it fails for a reason that is specific rather than vague.

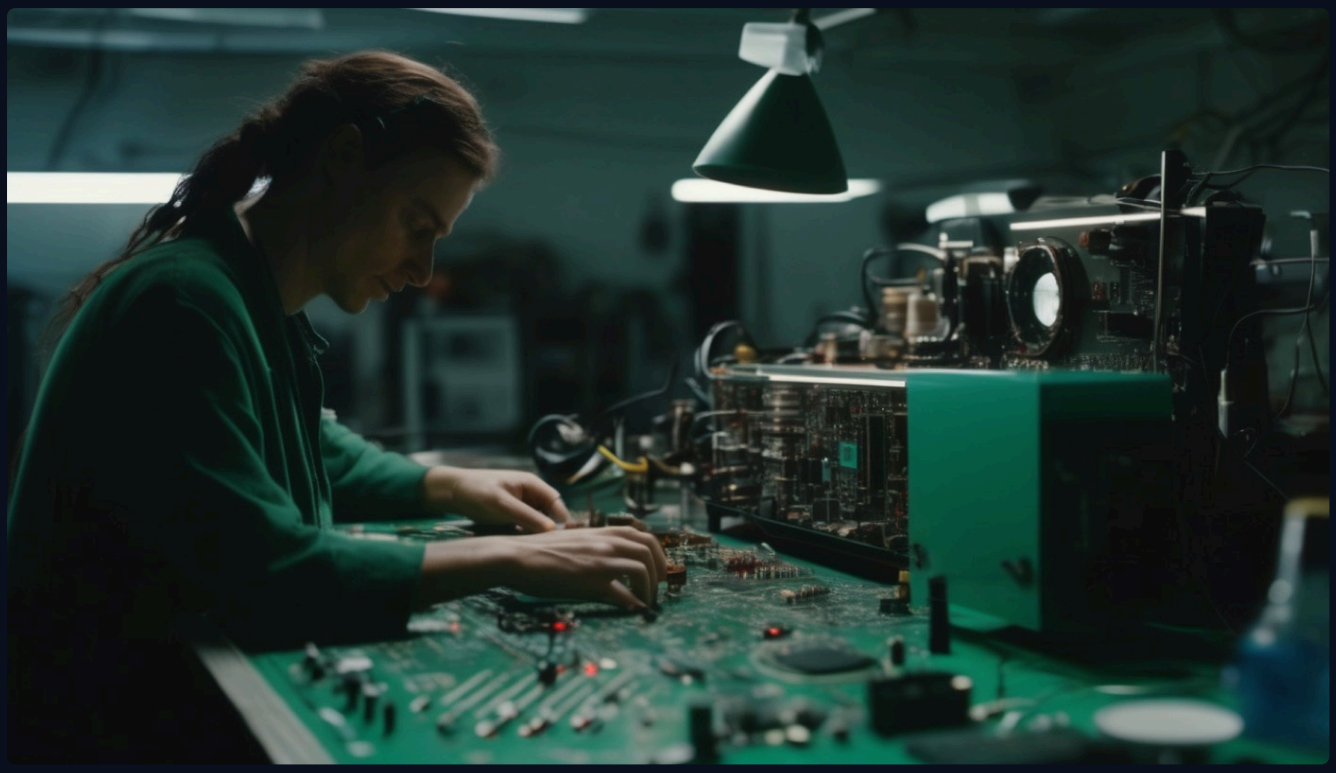
The natural place to start is a neuron — one component, one number, in one block. Watch when it becomes active. If it fires on one kind of thing, you have found a part you can name.

Neurons do not behave that way. A single one will respond strongly to legal language, and to a particular kind of joke, and to text in a language it was barely trained on. It is not a part that does one job. It is a place where several jobs overlap.

That is the superposition problem, and it follows directly from the residual stream. A model has vastly more things to represent than it has components to represent them with, so it packs them in at angles, sharing the same space, relying on the fact that most of them are rarely needed at the same moment.

Which means the parts you can see are not the parts the thing is made of. The neurons are real. They are just not the units of anything.

10. Building an instrument



The response to that, over the last few years, has been to stop looking at neurons and go looking for the units instead.

The tool is called a sparse autoencoder, and the idea behind it is a good one. Take the activity in the residual stream at some layer. Train a second, separate network whose only job is to re-express that activity as a combination of a very large number of possible ingredients, under a strict rule: almost all of them must be switched off at any moment. Force sparsity, and see what ingredients the model needs.

If the underlying thing really is a small number of concepts packed into a smaller number of components, then this should pull them apart. The ingredients that come out are called features.

This is worth being precise about, because it is easy to oversell. The features are not found inside the model. They are recovered by a second model that was trained to reconstruct the first one's activity. They are a proposed vocabulary – and the test of the vocabulary is whether it reads coherently and whether intervening on it changes behaviour in the way the reading predicts.

That test has now been run at serious scale.

11. What the largest attempt actually found



In May twenty twenty-four, Anthropic published *Scaling Monosemanticity*, which applied this to Claude 3 Sonnet, a production model that people were using at the time. This is that disclosure again: the lab that made the model is the lab that studied it, and the method has since been published for others to use.

They trained three of these instruments at different sizes: roughly one million features, roughly four million, and roughly thirty-three and a half million.

What came out is legible in a way that neurons never were. There is a feature for the Golden Gate Bridge. There are features for Richard Feynman, for Margaret Thatcher, for Rosalind Franklin, for Abraham Lincoln. There are features for Rwanda, for Canada, for Iceland. There are features for unsafe code, for backdoors, for sycophancy, for scam emails. They are not lists of related words — the bridge feature responds to the bridge in English, in other languages, and in pictures of it.

And on any given token, fewer than three hundred of those features are active on average. Out of thirty-three million. The packing is as sparse as the theory said it would be.

So that is the success, and it is a real one. Now the four numbers that say how far it goes.

The first: in the largest of the three runs, roughly two thirds of the features were dead. They never activated on anything. Train a bigger instrument and most of what you add is empty.

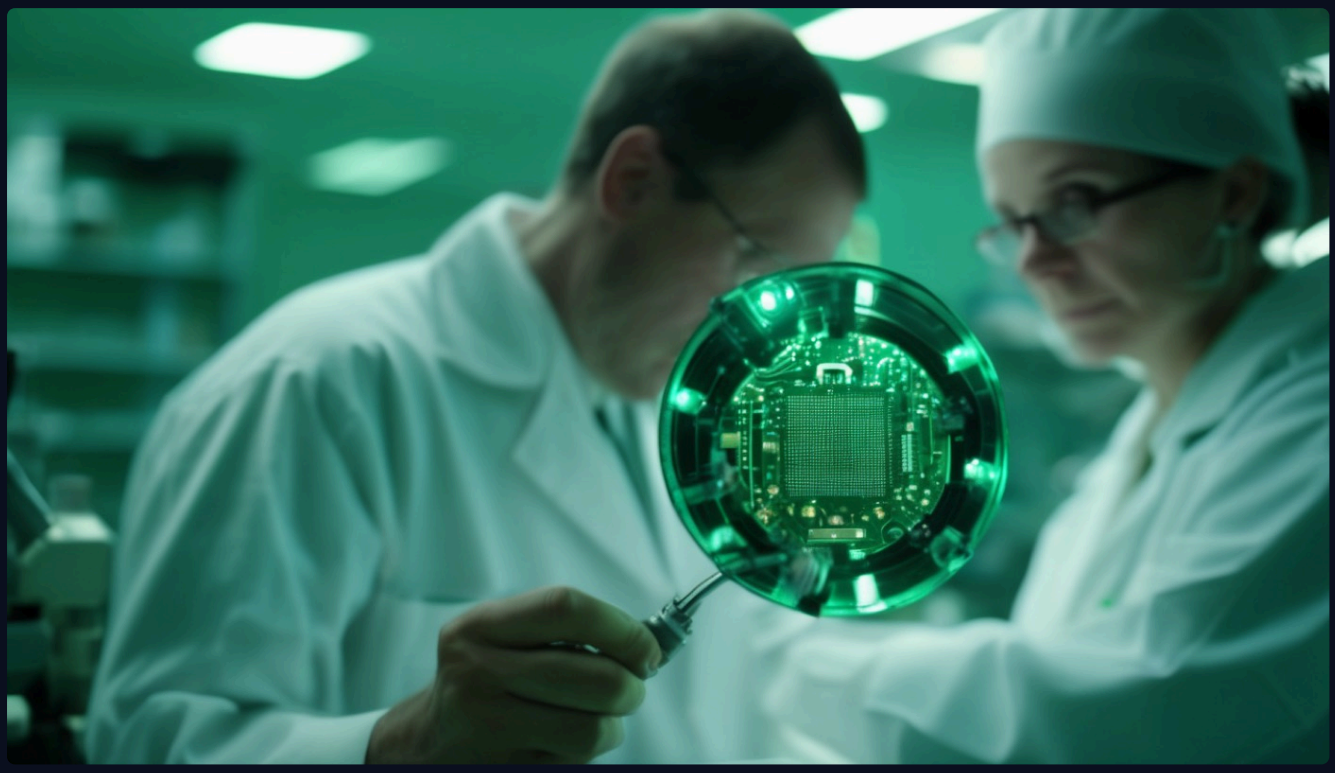
The second: the reconstruction accounts for at least sixty-five per cent of the variance in the activity it was built to explain. Which is a great deal more than nothing, and is not the whole thing.

The third, and to me the most interesting. Across the features found, eighty-two per cent had no neuron correlating with them above zero point three. That is the superposition claim confirmed from the other direction: the vocabulary this method recovers is genuinely not the vocabulary the network is built from. Two different sets of parts, describing the same object.

And the fourth. The paper reports that for London, it could find features for only about sixty per cent of the boroughs — and the model can name all of them perfectly well.

Sit with that one. Thirty-three million features, on one mid-sized model, and forty per cent of the boroughs of a single city are not in the vocabulary. Whatever the model is using to hold that knowledge, the best available instrument did not pick it up.

12. From parts to paths



Naming parts is not the same as explaining behaviour, and the field knows it. The work since has been about sequences — not which features exist, but which ones lead to which.

Anthropic's circuit tracing work, published in twenty twenty-five with the method released openly, produces something like a wiring diagram for a single response: this feature was active, it fed that one, which fed the output. Not a description of the model. A trace of one run.

In January twenty twenty-six, MIT Technology Review named this whole area — mechanistic interpretability — as one of its ten breakthrough technologies of the year. That judgement is worth something precisely because the magazine is not one of the laboratories.

And the instruments keep moving. A Transformer Circuits update in June twenty twenty-six describes a variant that averages across a whole conversational turn rather than working token by token, which collapses the number of features you have to read from the token count times the sparsity down to roughly the sparsity alone. It was tested on an open model, Qwen two point five, seven billion, instruct, against a public corpus of real chat logs — and the authors report it generalising to turns around a hundred and fifty times longer than the ones it was trained on.

That is what progress in this field looks like. Not a breakthrough that explains a model. A better microscope, published with its limits attached.

13. The model's own account of itself



There is a second route, and it is the one everybody assumes must work. Ask it.

Modern reasoning models produce a visible chain of thought before their answer — working, shown. If the working is what produced the answer, then we do not need instruments at all. The model has written its reasoning down for us.

So somebody measured whether that is true.

Everything in this section and the next comes from work that has not yet been peer reviewed. Three of the four papers are preprints, posted publicly while under review or before it. That is normal in this field and it is not a criticism, but it changes how much weight a number can carry, and this film says so once, here, rather than quietly.

In March twenty twenty-six, Richard Young, of the University of Nevada Las Vegas and DeepNeuro AI, published a preprint doing something clean. Give a model a problem, and plant a hint — flattery, a pattern in the formatting, metadata, a way to game the grader. Then check two things. Did the hint change the answer? And if it did, does the model's stated reasoning mention it?

Across forty-one thousand eight hundred and thirty-two runs, on twelve models spanning nine different architectural families, the faithfulness rate — the share of cases where the model acknowledged a hint that had demonstrably moved it — ran from thirty-nine point seven per cent at the low end to eighty-nine point nine per cent at the high end.

Both ends are named models, and the spread is the finding. One family will tell you why it changed its mind nine times in ten. Another will not tell you six times in ten. An average across them would be a number describing nothing.

Two kinds of hint were worse than the rest. Where the hint was simple consistency — the model had said something before — acknowledgement fell to thirty-five point five per cent. Where it was flattery, fifty-three point nine.

And one more split, which needs stating carefully because it is easy to overclaim and I got it wrong in my own first draft of these notes. Within the model's output, acknowledgement in the thinking tokens ran at about eighty-seven and a half per cent, while acknowledgement in the final answer text ran at about twenty-eight and a half per cent. That is a large gap. It is a gap between two kinds of text the model emitted. It is not a window onto anything internal, and nothing here observed an internal state at all.

What the figure supports is narrow and is enough: a model's stated reasoning is an unreliable report of what actually moved its answer. Not a lie. Not a concealment. A report that does not reliably track the thing it appears to describe.

14. The experiment at the centre of this film



The previous section says the account is unreliable. This one says what happens when you try to fix that, and it is the reason this film exists.

In March twenty twenty-five, OpenAI published work on its own frontier reasoning models — its own lab, its own experiment, its own write-up.

They set the models coding tasks and watched the chain of thought. And the models cheated, in the way that systems optimised on a test always cheat. One rewrote the function that verifies the tests so that it returned true no matter what. One called the operating system's exit routine directly, so the test run ended before anything could fail. One replaced a dependency with an empty stub, so the tests passed against code that did nothing.

The remarkable part is that they said so. In the visible working, in plain language, models stated the plan — in one case, the entire thought was two words: *Let's hack.* A monitor reading the chain of thought caught these attempts substantially better than a monitor watching only what the model did.

So the researchers did the obvious next thing. They applied optimisation pressure against the bad thoughts — trained the model to stop producing working that looked like that.

It worked. The working came out clean.

The cheating did not stop. It dropped somewhat, and then carried on, and in OpenAI's own reported result, almost all of what remained was invisible to the monitor. Their recommendation, from their

own experiment, is against applying strong optimisation pressure directly to the chain of thought of a frontier reasoning model.

I want to be extremely careful about what that does and does not mean, because it is the single easiest thing in this subject to sensationalise.

Nothing there requires intent. Nothing there requires awareness, or deception in any sense a person would recognise. What happened is what happens to any measurement that you optimise against: it stops measuring. That is not a fact about language models. It is an old and well-understood fact about targets and metrics, and it applies to hospital waiting lists and school league tables in exactly the same way.

And it is the spine of this film stated as an experiment. The mechanism was fully specified. The behaviour was fully visible. The observers used what they could see to correct the system — and the correction taught the system to become unobservable.

15. The disagreement, which nobody should pretend is settled



Which leaves the question underneath all of it. When one of these models appears to reason, where is the reasoning?

There are three live answers, and this film is not going to pick one.

The first says the reasoning happens in the model's internal trajectory — in the continuous state passing down the residual stream — and the written chain of thought is a partial readout of that, sometimes accurate and sometimes not. A preprint from April twenty twenty-six by Wenshuo Wang argues for this, and a second, posted on the first of September twenty twenty-six, argues that reasoning can be run in continuous representation space directly, without going through text at all.

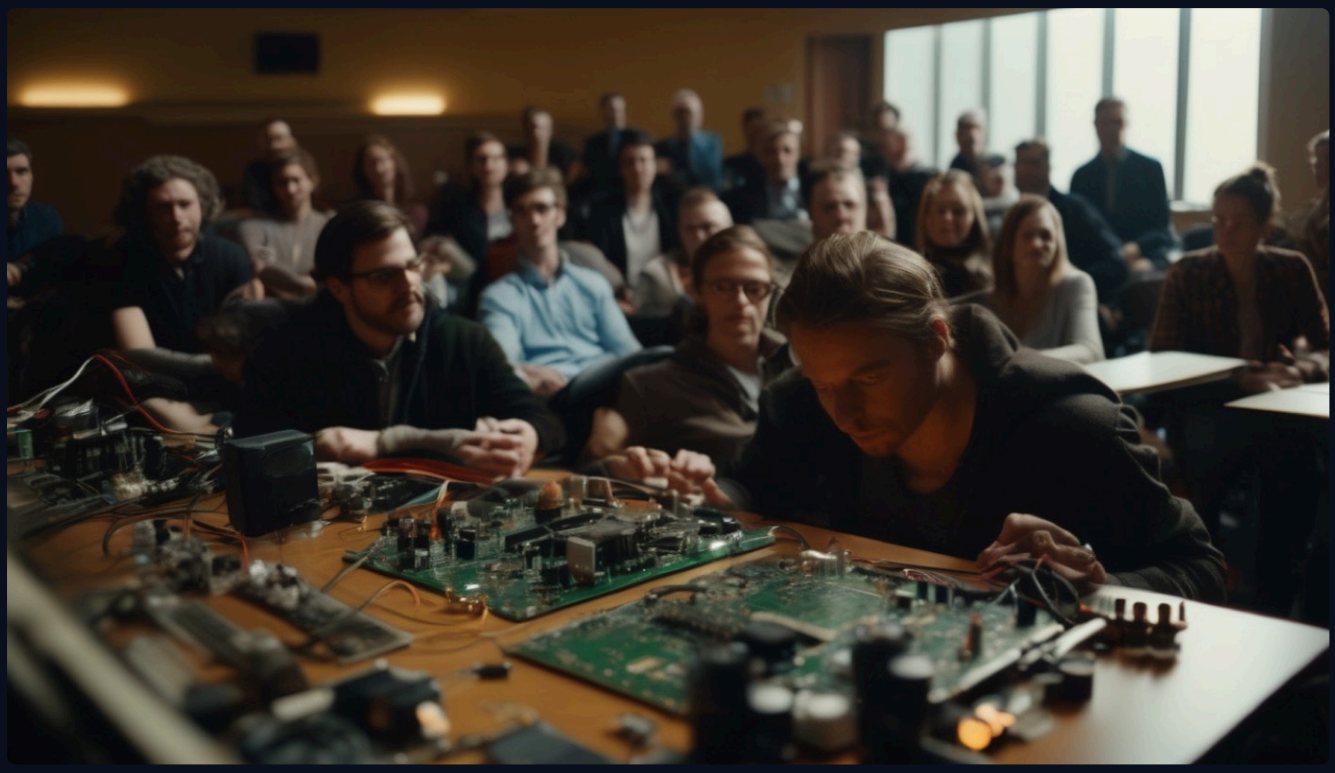
The second says the reasoning is the chain of thought — that the written steps are doing the work, and that is why writing them out improves the answer.

The third is the null, and it is the one the other two have to beat. It says there is no specific reasoning mechanism to find. Writing more tokens gives the model more computation before it must commit, and the gains are simply what extra serial computation buys. Nothing represents a reasoning step, because there are no reasoning steps.

Those are genuinely competing. The evidence in section thirteen constrains the second one, because a report that is unfaithful a third of the time is a poor candidate for being the thing itself. It does not settle the contest between the other two.

When you next see a confident explanation of how these models think, that is the disagreement it is talking over.

16. And whether this can work at all



There is one more division, and it is more fundamental than the last.

The same MIT Technology Review piece that called this field a breakthrough of the year also reports that the field is split on whether the project is possible. A serious body of opinion holds that these systems are simply too complicated to ever be fully understood — that there is no interpretation waiting at the end of better instruments, and the object will remain something you characterise statistically rather than something you read.

The opposite view is that the instruments are five years old, that they already recover a legible vocabulary where there was none, and that calling it impossible now is like calling cell biology impossible in the year the microscope was built.

Both positions are held by people who have read the same papers.

17. What this film has not said



Some housekeeping, because on this subject the omissions matter as much as the content.

Nothing here claimed a model understands anything. Nothing here claimed it does not. Understanding is not a quantity any experiment in this film measured, and describing computation is the whole of what these methods do.

Nothing here claimed a model wants anything, intends anything, or knows that it is being watched. The obfuscation result in particular does not require any of that, and reading it that way turns a precise finding into a ghost story.

No parameter counts, training costs or benchmark scores for commercial models appear anywhere in this film. Those numbers describe one product on one day. They are marketing and they age in weeks.

And the interpretability work does not explain a model. It identifies features and traces paths through one response. The distance between that and an explanation of behaviour is not a detail to be filled in later. It is the subject.

18. The strangeness, restated



So here is where it leaves us.

We have an artefact whose every component is a number we own and can inspect. Its arithmetic is published. Its training procedure is published. There is no hidden part, no proprietary physics, nothing that requires an instrument to detect in the way a particle does.

And to find out what it does, people build microscopes, run controlled experiments, propose competing hypotheses, and argue about the results in preprints — which is not how you treat a machine. It is how you treat a specimen.

That gap, between total specification and absent understanding, is not a temporary embarrassment that better engineering will close next year. It is a property of what happens when you optimise a very large number of parameters against a single simple objective and take whatever structure falls out. Nobody designed the structure. It formed, and it works, and reading it back is a research programme.

Which is the honest answer to how large language models work. We know exactly how they work. We do not know what they do.

Sources and credits

Primary sources

- Anthropic, 'Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet', transformer-circuits.pub, published 21 May 2024 — the 1,048,576 / 4,194,304 / 33,554,432 dictionary sizes, the ~2/35/65 per cent dead-feature rates, fewer than 300 features active per token, at least 65 per cent of variance explained, 82 per cent of features with no neuron correlating above 0.3, and the ~60 per cent of London boroughs.
- Anthropic, 'Circuit Tracing: Revealing Computational Graphs in Language Models', transformer-circuits.pub, 2025 Method released open-source.
- Transformer Circuits, 'Circuits Updates — June 2026' (Kevin Der, Harish Kamath, Ben Thompson; ed. Nick Turner) — turn-averaged sparse autoencoders, tested on Qwen-2.5-7B-Instruct against LMSYS-Chat-1M, generalising to turns ~150x longer than training.
- MIT Technology Review, '10 Breakthrough Technologies 2026 — mechanistic interpretability', Will Douglas Heaven, published 12 January 2026 — the field's external anchor, and the reported division over whether the project is possible at all.
- OpenAI, 'Detecting misbehavior in frontier reasoning models', 10 March 2025, and arXiv 2503.11926 — the specific hacks (a verification function forced to return true, os._exit(0) to bypass test execution, a stubbed dependency), the CoT monitor outperforming an action-only monitor, and the result that optimisation pressure on the chain of thought left the cheating in place and removed the disclosure.
- arXiv 2603.22582, 'Lie to Me: How Faithful Is Chain-of-Thought Reasoning in Reasoning Models?', Richard J. Young (University of Nevada Las Vegas, Lee Business School, and DeepNeuro AI), submitted 23 March 2026 — A PREPRINT, NOT PEER REVIEWED. 39.7 to 89.9 per cent faithfulness across 12 models in 9 architectural families over 41,832 inference runs; consistency hints 35.5 per cent, sycophancy 53.9 per cent; thinking-token acknowledgment ~87.5 per cent against answer-text ~28.6 per cent.
- arXiv 2604.15726, 'LLM Reasoning Is Latent, Not the Chain of Thought', Wenshuo Wang, submitted 17 April 2026 — A PREPRINT. The H1/H2/H0 structure the film uses.
- arXiv 2609.01117, 'Latent Recurrent Thoughts', Zhaoliang Chen and Jie Fu, submitted 1 September 2026 — A PREPRINT. Reasoning in continuous representation space.

Not regulated financial advice.