The Sweetest Lesson

Your Brain Versus AI

David Spuler

Aussie AI Labs

Published by Aussie AI Labs Pty Ltd, Adelaide, Australia.

https://www.aussieai.com

First published: August 2025.

About the Author

David Spuler is a C++ AI programming expert and serial technology entrepreneur who has combined his love of writing with LLM technology in his latest venture: Aussie AI is a suite of tools for writing and editing, with a focus on fiction from short stories to full-length novels. His published works include three advanced C++ books (low latency, data structures, and safety), two generative AI LLM books, two CUDA C++ coding books, four non-fiction textbooks on C++ programming covering both introductory and advanced C++ coding, efficiency and optimization, code debugging and testing, and software development tools, and one application management book.

Other than writing, he's an avid AI researcher with a Ph.D. in Computer Science and decades of professional experience. Most recently, Dr. Spuler has been founding startups, including the current Aussie AI startup and multiple high-traffic website platforms with millions of monthly uniques, including an e-health startup acquired by HealthGrades, Inc. Prior roles in the corporate world have been as a software industry executive at BMC Software, M&A advisor, strategy consultant, patent expert, and prolific C++ coder with expertise in autonomous agents, compiler construction, internationalization, ontologies and AI/ML. Contact by email to `research@aussieai.com` or connect via LinkedIn.

About the Contributors

Michael Sharpe is an experienced technologist with expertise in AI and ML, cybersecurity techniques, cloud architectures, compiler construction, and multiple programming languages. He is currently Senior Software Architect at PROS Inc., where he is a member of the Office of Technology focusing on developing and evangelizing AI. His AI expertise extends to monitoring and observability, devops and MLOps, ITSM, low-resource LLM inference, Retrieval Augmented Generation (RAG) and AI-based agents.

In a long R&D career, Michael has been coding C++ for almost 30 years, with prior roles at BMC Software, Attachmate (formerly NetIQ) and IT Involve. Michael has a Bachelor of Science with First Class Honors in Computer Science from James Cook University and holds several registered patents.

Cameron Gregory is a technology entrepreneur including as co-founder of fintech bond trading startup BQuotes (acquired by Moody's), a Senior Data Scientist focused on "big data" for hedge funds at fintech startup Advan Research Corporation, co-founder and Chief Technology Officer (CTO) of Trademark Vision with an AI-based image search product (acquired by Clarivate), and founder of several image creation companies including FlamingText.com, LogoNut, AddText, and Creator.me.

Cameron has been making code go fast since the 1990s at AT&T Bell Laboratories in New Jersey, is used to working with real-world data at scale, and is proficient in multiple programming languages, including C++, Java, and JavaScript. He holds a Bachelor of Science with First Class Honors in Computer Science from James Cook University.

Preface

"I would like to die on Mars.

Just not on impact."

— Elon Musk.

Why a Book on The Sweetest Lesson?

Who doesn't like the sweet taste of success? Let's celebrate the powers that make humans special by comparing ourselves to the latest technological advances in an AI world. There's far too much hype in this world about AI, and not enough hype about the amazing capabilities of the carbon-based bio-robots, otherwise known as people.

How This Book is Organized

The best way to read this book is to open all the pages and then read them all at the same time. That's how an AI model would do it on their parallelized NVIDIA GPU chips, and what's good enough for silicon should work in carbon.

Alas, no.

Sadly, the book is organized sequentially, because we are logical lifeforms and we've been trained to read one page at a time. Oddly, our brains can process a huge volume of input signals in parallel, so maybe we could read all the pages in parallel, if only we were to try!

About Aussie AI

Aussie AI is a platform for the development of consumer AI applications, with a special focus on AI-based writing and editing tools for fiction. Our premier applications offer an extensive range of reports and error checks for both fiction and non-fiction writing, from a full-length novel to a short report. Please try it out and let us know what you think: https://www.aussieai.com

Our AI Research

The primary focus of research at Aussie AI is on optimizing LLM inference algorithms (i.e., "running" the model after training or fine-tuning), and our research is toward the following aims:

- Fast on-device model inference, specifically for smartphones and AI PCs.
- Scaling inference algorithms to large volumes of requests.
- Efficient GPU inference algorithms (hardware acceleration).
- Non-GPU inference optimization algorithms (i.e., software methods).

Disclosure: Minimal AI Authorship

Despite my being involved in the AI industry, there was almost no AI engine usage in creating this book's text or its coding examples. Some text has been analyzed and reviewed using Aussie AI's editing tools, but not even one paragraph was auto-created by any generative AI engine. All of the code is also human-written, without involvement of any AI coding copilot tools. I mean, who needs them?

However, AI was used in several ways. AI-assisted search tools were very useful in brainstorming topics and researching some of the technical issues. The main cover art image was AI-generated by Google ImageFX, followed by human editing.

Disclaimers

Although I hope the information is useful to you, neither the content nor code in this work is guaranteed for any particular purpose. Nothing herein is intended to be personal, medical, financial or legal advice. You should make your own enquiries to confirm the appropriateness to your situation of any information.

Many code examples are simplistic and have been included for explanatory or educational benefit, and are therefore lacking in terms of correctness, quality, functionality, or reliability. For example, some of the examples are not good at handling the special floating-point values such as negative zero, `NaN`, or `Inf`.

Oh, and sometimes I'm being sarcastic, or making a joke, but it's hard to know when, because there's also a saying that "Truth is often said in jest!" Your AI engine certainly won't be able to help you sort out that conundrum.

Third-Party License Notices

Except where expressly noted, all content and code is written by David Spuler or the contributors, with copyright and other rights owned by David Spuler and/or Aussie AI.

Additional information, acknowledgments and legal notices in relation to this book, or to other books, source code, and other Aussie AI software, can be found on the Aussie AI Legal Notices page: https://www.aussieai.com/admin/legal-notices.

Please Leave a Review

I hope you enjoy the book! Please consider leaving a review on the website where you purchased the book. Since few readers do this, each review is important to me, and I read them all personally.

Feedback and Contacts

Feedback from readers is welcome. Please feel free to tell us what you think of the book, the research literature review, or our Aussie AI software. Contact us by email via `support@aussieai.com`.

Table of Contents

1. Your Brain Versus AI

"You are more powerful than you think."

— Tim Cook.

Your Brain is Bigger

Your brain is big and it's definitely a lot bigger than any AI engine. If you talk to an AI chatterbox for a while, you'll probably start to realize that it's not actually as intelligent as it should be, and here's the main reason: it's not big enough.

The human brain is so large that it's difficult to quantify it with an exact number, but the prevailing theory is that you have around 100 trillion synapses that connect about 86 billion neurons. We'll examine the evidence for those numbers in a later chapter, but for now, note that there are therefore more than 1,000 synapse connections per neuron, although some estimates are as high as 7,000 for the ratio, which would mean 700 trillion human synapses. A trillion is a one followed by 12 zeros, so these numbers are not small. You can't count that many synapses with a microscope and a hand-clicker, but feel free to try.

AI engines vary in size, but none are anywhere near that big. Instead of 86 billion neurons, they usually have a "model size" or "hidden dimension" of about 7,268. That's about 11 million times smaller, if you wanted an ego boost.

The equivalent of a human synapse in an AI model is called a "weight" and it's just one number. Like synapses, these weights can either intensify or reduce a "signal" in the silicon brain, depending on whether they're large numbers or small fractions.

All AI engines are smaller in size than the human brain. OpenAI's GPT-4 was reportedly about 1.76 trillion weights, or let's say 2 trillion to be kind, which is still 50 times smaller than 100 trillion carbon-based synapses.

Also, GPT-4 really only used 200 billion weights at a time in what's called a "Mixture-of-Experts" architecture, which is abbreviated as "MoE" (to remember this, think of *Moe's Tavern* in *The Simpsons*, which probably has a few "experts"). So, that's 500 times smaller than a brain, if you're keeping count.

But GPT-4 is over a year old and not state-of-the-art as I write this. Not all of the top frontier models disclose their sizes, but I can tell you a few. The Google Gemini 1.5 Pro was about one trillion weights (100 times smaller).

The DeepSeek R1 model, which caused a huge flutter in the industry when released, had multiple models ranging in size up to 671 billion weights, about a quarter the size of GPT-4, so that's 200 times smaller. DeepSeek R1 also used the "MoE" architecture, at 37 billion weights per invocation, which is 2,700 times smaller.

The latest one this week as I write this, and these leaderboards seem to change every week, is the Kimi K2 model from Moonshot AI, which has around 1 trillion weights (100 times smaller) with a 32B MoE architecture (3,125 times smaller than a brain).

You can pick any of these numbers that make you feel good, even up to the fun fact that AI has 11 million times fewer neurons. That's probably wishful thinking and it also seems unfair to compare our total synapse counts against the smaller "MoE" subsets of activated weights in those models, rather than their full weight counts. Humans also don't "activate" all of their brain, or haven't you seen any of the Hollywood movies with this premise?

So, let's compare human synapses to AI weights. Personally, I think the most likely number is close to this: based on 100 trillion synapses versus two trillion weights, you are at least 50 times smarter than an AI engine.

Your Brain is More Efficient

Despite being massively bigger, your brain is far more energy-efficient to run. The typical comparison is that your brain is like a 20 watt lightbulb, except that nobody ever turns it off, because you need it to keep breathing. Inside your head, you're always "burning the midnight oil" even while you sleep. For comparison, a single high-end GPU chip in a data center uses hundreds of watts.

Your brain actually runs hotter than most other animal species. The brain consumes 20% of the human body's total energy output (100W), despite only being 2% of our body weight.

You've probably heard that the advancement of human intelligence was spurred along by one of these theories:

1. Our ability to stand upright.

2. A larger brain space in the skull (also somewhat related: being taller).

3. The invention of fire to allow cooking to expose enough nutrients to feed our brain.

But there's another one that gets less attention:

The invention of sweating.

Our brain runs so hot that we would overheat without sweating, because even 20 watts is quite a heatload in an enclosed space. Exposed skin with sweat glands is critical to evaporate sweat, and this "evaporative cooling" is what makes sweating work. Most other animal species can't sweat enough to radiate enough heat away. Hence, evolutionary brain size was limited by the inability to sweat.

In order to sweat, we needed to lose most of our hair, too. Animal fur interferes with sweating in most species, which is why your pet dog hangs its tongue out the car window all the time, although maybe it's also fun. Hence, if you're balding in your old age, you can now say it's a sign of high intelligence.

All of those AI GPU chips need "liquid cooling" to run hot, too. It runs in pipes under "raised floors" inside massive data centers. That's another way that your brain is better than AI: your brain uses a lot less water, also.

Your Brain Learns Like AI

No, that's wrong. Your brain is not like AI, because your intelligence came first, and the AIs are a recent invention. AI is like your brain, because it was designed by scientists to use the same brain-like methods.

AI is a copy.

The idea is called a "neural network" and yours uses chemicals and carbon-based molecules, whereas AI uses GPU chips that run on electricity in silicon.

Your brain learns like AI, because the "training" of LLMs is based on your neural network. In the brain, the mappings between neurons are adjusted via "synaptic plasticity," which doesn't mean they're made of plastic, but that they are soft and changeable.

AI mappings are numbers called "weights" and they are changed by adding or subtracting small amounts to these numbers, as the LLM training process scans through reams and reams of pre-written text.

The technical term for how AI models get better weights is "backpropagation" because the "forward pass" reads the data, and then the "backward pass" propagates small changes to the weights, so that they'll be better next time. This is a slow process, sometimes even taking weeks or months for a big model, modifying the LLM weights a tiny amount each document they read.

They say that humans need 10,000 hours of practice to master a task. Well, AI engines need a lot more than 10,000 GPU hours of computation for their weights to converge to numbers that make it smart.

Hence, another point: your brain learns faster than AI.

References

References on human brain size and sweating abilities:

1. Kovác L., 2009, *The 20 W sleep-walkers*. EMBO Rep. 2010 Jan;11(1):2. doi: 10.1038/embor.2009.266, https://www.embopress.org/doi/full/10.1038/embor.2009.266, https://pmc.ncbi.nlm.nih.gov/articles/PMC2816633/
2. Kandel E.R.,Schwartz J.H., and Jessell T. M., January 2000, *Principles of Neural Science (4th Ed.)*, New York, McGraw-Hill, https://www.amazon.com/Principles-Neural-Science-Eric-Kandel/dp/0838577016 (Early source for the 100 trillion synapse estimate.)
3. Carl Zimmer, January 2011, *100 Trillion Connections: New Efforts Probe and Map the Brain's Detailed Architecture*, Scientific American Magazine, Vol. 304, No. 1, https://www.scientificamerican.com/article/100-trillion-connections/
4. Vybarr Cregan-Reid, July 20, 2016, *From perspiration to world domination – the extraordinary science of sweat*, https://theconversation.com/from-perspiration-to-world-domination-the-extraordinary-science-of-sweat-62753
5. National Science Foundation, April 20, 2021, *How humans evolved a super-high cooling capacity: Discovery illuminates human sweat gland evolution*, https://www.nsf.gov/news/how-humans-evolved-super-high-cooling-capacity
6. Rafe Brena, May 24, 2024, 3 Key Differences Between Human and Machine Intelligence You Need to Know: AI is an alien intelligence https://pub.towardsai.net/3-key-differences-between-human-and-machine-intelligence-you-need-to-know-7a34dcee2cd3 (Good article about how LLMs don't have "emotions" or "intelligence" and they don't "pause".)

References on AI engine weight sizes:

1. Seifeur Guizeni, 2024, *Decoding the Enormous Scale of GPT-4: An In-Depth Exploration of the Model's Size and Abilities*, https://seifeur.com/chat-gpt-4-data-size/
2. Toolify.ai, Mar 01,2024, *Inside GPT-4: Leaked Parameters, Weights, and Costs*, https://www.toolify.ai/ai-news/inside-gpt4-leaked-parameters-weights-and-costs-2438865
3. Gemini Team Google: Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, (many more authors not shown), 16 Dec 2024 (v5), *Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context*, https://arxiv.org/abs/2403.05530
4. DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, (over 100 more authors not shown), 22 Jan 2025, *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning* https://arxiv.org/abs/2501.12948
5. Kimi Team, July 2025, *Kimi K2: Open Agentic Intelligence*, https://github.com/MoonshotAI/Kimi-K2/blob/main/tech_report.pdf, https://github.com/MoonshotAI/Kimi-K2

2. The Bitter Lesson

"AI is a marathon, not a sprint."

— Demis Hassabis.

What is The Bitter Lesson?

Although the previous sections should have you on a high about your amazing brain, this is where we have to take a long, hard look at the history of analyses like that in the computer industry. It's very bitter.

The idea of the "bitter lesson" was coined in 2019 by Rich Sutton in relation to intelligent computing. The core of the idea is that a simple algorithm combined with brute-force computer power will always eventually outperform apparently smarter algorithms.

The reason that this idea is "bitter" is that human researchers tend to assume that making computers more like them, by using tricky human-like rules of thumb (called "heuristics" by boffins), will be the best way.

It usually works for a while, but then there's the reality that these advanced research algorithms get overrun by much simpler and dumber algorithms, without any fancy logic, hooked up to a very fast computer.

When presented with a new problem, human researchers will try to solve it like humans. Hence, they will use things based on human-like intelligence, such as:

1. Heuristics

2. Logic

Heuristics are human-like "tricks" or "shortcuts" in solving a problem that are then coded up into a computer algorithm. Logic is part of this, where possible solutions are analyzed according to some rational metrics. So, this is the general approach:

How do we find some of the possible solutions? — Heuristics.

What's the best solution? — Logic.

These types of methods often do well in the short-term in solving a problem, even on fast computers at the time. However, over time, computers get even faster, and the best algorithms often turn out to involve:

1. Brute-force computations, and

2. Very simple comparisons.

It's kind of like that movie *Everything Everywhere All at Once*, only not quite as funny.

You don't need heuristics to find the best solution. Instead, just get a faster computer and *try all of them*. The general approach changes to:

How do we find the possible solutions? — Try every single one.

What's the best solution? — Compare them all.

When this solves a problem better than the early heuristic versions, it's a "bitter lesson" for the initial human researchers. And I mean, at a level of bad taste in the mouth beyond what you get from a dandelion sandwich. Researchers think the solution should be something clever, but there's nothing smart about this brute-force approach, and it's just a dumb box cranking through the whole solution space. It shouldn't work so well, but it does.

Bitterness ensues. Lots of musty research papers get torn up by librarians and tossed in the trash.

It's happened in the computer industry, over and over again, and yet when researchers are presented with a new problem to solve, they still tend to try heuristics and logic.

Why is it so hard to learn from these mistakes, rather than learn that raw compute always wins? Maybe, it's because of what it implies at the deepest levels of the soul:

> *Computers are better than humans.*

Surely, it can't be true?

Bitter Chess

Chess-playing computers are the best example of the bitter lesson. The best chess computer in the world is named Stockfish, and it has a rating of over 3,000 ELO points. The best grandmasters are "only" around 2,700 points on this rating scale.

The way it started was that programmers tried to copy how humans play chess. There are also sorts of rules of thumb that you learn at Chess Club, such as:

- Center your pieces.
- Queens are worth nine pawns.
- Make a breathing space for your king.

A computer can play a reasonably strong game of chess if you code up these sorts of rules. It's rating is maybe around 1,600 if you do this. My favorite reference on this is the 1988 book by David Levy.

The first major improvement came shortly after. If you combine these rules with a very powerful computer that scans through lots of possible moves, it does better. Not smarter rules, but just more grunt to scan all the whole tree of possible moves that each player can make. This was IBM's "Deep Blue" computer that played chess, and it was about the level of the World Chess Champion at the time, Garry Kasparov, who beat Deep Blue in the first match in 1996, but lost the rematch a year later.

Not smarter, just faster, but still using human techniques. Half bitter.

What followed was even worse. It turns out that computers don't even need the human rules of thumb. In 2017, an AI company called DeepMind created an even more massive computer called AlphaZero, that was better at chess. This would be fine, except for the way that it did so.

All of those rules of thumb about how to play chess well, crafted by humans over centuries of careful analysis: *worthless*. AlphaZero only needed the basic rules of chess and a lot of GPU chips to run the AI learning algorithm. Rather than requiring software code for these heuristic rules of thumb, AlphaZero just played lots of random games against itself, watching for what works, and what doesn't. The computer played better just by figuring out what to do itself, learning all of the types of patterns in the game, using methods that humans cannot even understand.

Now that's bitter.

Intelligence and the Bitter Lesson

Will the achievement of human-level intelligence be another case of the bitter lesson? If the best models are at 2 trillion weights and the human brain has about 100 trillion, maybe the problem is just how to brute-force the AI engines with 50 times more power. Is the failure to reach true intelligence just that technology companies haven't yet figured out how to run 100 trillion weights so as to match the true size of the human neural network?

Reasoning could be another bitter lesson. After all, the big kerfuffle around DeepSeek in early 2025 was mainly that reasoning could be done just with training. If you showed the LLM enough mathematical proofs in training, it got better at proving things.

If the AI companies put their minds to it, they'll find training examples for every type of reasoning. Proof by mathematical induction, by contradiction, by algebraic transformations? It's all be done a thousand times in all the research papers. But those are just proofs, and that's just one way to do reasoning.

Are there really that many ways to do reasoning?

And if we crank up our 100 trillion weights in an ultra-massive AI model, and show it all those different examples of reasoning, maybe that's job done. The AI engine has then seen countless examples of the hundreds or thousands of different ways to do reasoning, and it's great at parroting them. Generalization, who needs it?

Is truly intelligent AI going to be the bitter lesson, all over again, one last time?

References

Articles and research papers on the "bitter lesson" include:

1. Rich Sutton, March 13, 2019, *The Bitter Lesson*, http://www.incompleteideas.net/IncIdeas/BitterLesson.html, PDF: https://www.cs.utexas.edu/~eunsol/courses/data/bitter_lesson.pdf
2. Mojtaba Yousefi, Jack Collins, 12 Oct 2024, *Learning the Bitter Lesson: Empirical Evidence from 20 Years of CVPR Proceedings*, https://arxiv.org/abs/2410.09649
3. Alberto Romero, Feb 19, 2025, *Grok 3: Another Win For The Bitter Lesson: Congratulations to the xAI team—and the advocates of the scaling laws*, https://www.thealgorithmicbridge.com/p/grok-3-another-win-for-the-bitter
4. lucalp, June 24, 2025, *The Bitter Lesson is coming for Tokenization: a world of LLMs without tokenization is desirable and increasingly possible*, https://lucalp.dev/bitter-lesson-tokenization-and-blt/
5. Leon Wu, July 2025 (accessed), *The Bitter Lesson: How Your Intuition About AI Is Probably Wrong*, https://leonwu.tech/posts/bitter-lesson
6. Michal Nauman, Michał Bortkiewicz, Piotr Miłoś, Tomasz Trzciński, Mateusz Ostaszewski, Marek Cygan, 19 Jun 2024 (v2), *Overestimation, Overfitting, and Plasticity in Actor-Critic: the Bitter Lesson of Reinforcement Learning*, https://arxiv.org/abs/2403.00514
7. Jesse Jing, Apr 12, 2023, *The Bitter Lesson: What direction to avoid in the field of neural symbolic AI?*, https://medium.com/towards-nesy/the-bitter-lesson-1a1d282ae1b9
8. Hassaan Naeem, May 10, 2022, *Thoughts: Sutton's The Bitter Lesson: Ponderings on Richard Sutton's the bitter lesson*, https://hassaann.medium.com/thoughts-suttons-the-bitter-lesson-6248c2d7e8c2
9. Minghao Wu, Weixuan Wang, Sinuo Liu, Huifeng Yin, Xintong Wang, Yu Zhao, Chenyang Lyu, Longyue Wang, Weihua Luo, Kaifu Zhang, 22 Apr 2025, *The Bitter Lesson Learned from 2,000+ Multilingual Benchmarks*, https://arxiv.org/abs/2504.15521
10. Dulhan Jayalath, Gilad Landau, Brendan Shillingford, Mark Woolrich, Oiwi Parker Jones, 2 Jun 2025 (v5), *The Brain's Bitter Lesson: Scaling Speech Decoding With Self-Supervised Learning*, https://arxiv.org/abs/2406.04328
11. Zhen Huang, Haoyang Zou, Xuefeng Li, Yixiu Liu, Yuxiang Zheng, Ethan Chern, Shijie Xia, Yiwei Qin, Weizhe Yuan, Pengfei Liu, 25 Nov 2024, *O1 Replication Journey -- Part 2: Surpassing O1-preview through Simple Distillation, Big Progress or Bitter Lesson?*, https://arxiv.org/abs/2411.16489

12. Julian Aron Prenner, Romain Robbes, 6 Mar 2025, *Extracting Fix Ingredients using Language Models*, https://arxiv.org/abs/2503.04214
13. Warren Morningstar, Alex Bijamov, Chris Duvarney, Luke Friedman, Neha Kalibhat, Luyang Liu, Philip Mansfield, Renan Rojas-Gomez, Karan Singhal, Bradley Green, Sushant Prakash, 8 Mar 2024, *Augmentations vs Algorithms: What Works in Self-Supervised Learning*, https://arxiv.org/abs/2403.05726
14. Martin Riedmiller, Tim Hertweck, Roland Hafner, 14 Dec 2023, *Less is more -- the Dispatcher/ Executor principle for multi-task Reinforcement Learning*, https://arxiv.org/abs/2312.09120
15. Sharut Gupta, Stefanie Jegelka, David Lopez-Paz, Kartik Ahuja, 20 Sep 2023 (v2), *Context is Environment*, https://arxiv.org/abs/2309.09888

References on computer chess theory:

1. David N. L. Levy, 1988, *Computer Chess Compendium*, https://www.amazon.com/dp/0387913319
2. Tord Romstad, Marco Costalba, Joona Kiiski, and contributors, August 2025 (accessed), *Stockfish chess*, https://stockfishchess.org/about/, Code: https://github.com/official-stockfish/stockfish-web
3. Pandolfini, Bruce, 1997, *Kasparov and Deep Blue: The Historic Chess Match Between Man and Machine*, Simon & Schuster, https://www.amazon.com/Kasparov-Deep-Blue-Historic-Between/dp/068484852X/
4. David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, Demis Hassabis, 5 Dec 2017, *Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm*, DeepMind, https://arxiv.org/abs/1712.01815

3. Intelligence

"Software is eating the world,

but AI is going to eat software."

— Jensen Huang, May 2017.

Unintelligent AI

The big goal in AI research is called "Artificial General Intelligence" or "AGI" for short. This refers to having an AI engine that is smart enough to match the general level of human intelligence. Estimates vary as to how many years it will take before we get to AGI in the future. But they all agree on one thing:

We're not there yet.

Sometimes, it seems like we are already at that level. The AIs are so great at mimicking human-like text, that it often feels to us like we're talking to a real human. But this is human nature to "project" ourselves onto a non-human algorithm. In reality, an AI model has these properties:

- LLMs have no "feelings" or "empathy" (it's just fake words).
- LLMs don't "care" if they make mistakes (even when you point them out!).

Your AI buddy doesn't have real feelings for you, but can certainly write words that make it seem that way, because that's how it's been trained. This type of training is called "alignment" or "conversational AI."

An LLM also feels no real "embarrassment" if you point out its mistakes, although, again, it has been trained to handle this type of input situation by saying words like "oops" or "sorry" or whatever. Your challenge to an AI engine about its answer feels not different to it than any other input prompt because, well, it feels nothing either way.

Also, it doesn't matter if you say "please" and "thank you" to an AI engine. Lots of people do, but it's just extra words to the LLM.

It's a machine, and there's literally no feelings anywhere. AI engines really are an "alien intelligence" and not very much like us.

Similarities

How are brains and LLMs similar? At a high level, there is a great deal of functional and structural similarity in how they perform tasks:

- Fast thinking and slow thinking modes
- Great at pattern recognition
- Nobody understands how they work!

A lot of the limitations of LLMs read like they were written about humans:

- Bad at arithmetic and math (without help from a calculator).
- Have biases, toxicity, and other unwelcome aspects.
- Can't do crossword puzzles well (especially cryptics!).
- Don't like anagrams and word scrambles.
- Forget stuff they were told earlier.
- Need training to be able to work with Windows.

It's like an endless series of memes. LLMs are so much like us!

Differences

Sure, one is carbon-based and one is silicon. Other than that, they're the same? Well, no, there are other distinctions between your brain and AI.

Differences appear more in the broader structure of the usage of a brain versus an LLM, whereas the low-level structure is a neural network for both.

How are they different in this higher-level functioning? Here's some of the areas for distinction between you and an LLM:

- LLMs are more specialized (e.g., at writing).
- Fast LLMs can process vast reams of information.
- LLMs are integrated digitally to other systems.
- Brains have a human body attached.

You might think it an advantage to have a calculator attached rather than a big glob of pulsating cells, but there's a whole research area that says that AIs won't achieve human-level intelligence without having a body attached. It's called "embodied AI" and predicts that AIs will need to learn about 3D environments by exploring them physically, while being trained with that information.

But, if you want to know the main difference between LLMs and human brains: *dumber than us*. They can perform some impressive brute-force calculations, and are great at writing documents or creating images, but they also make some simple mistakes.

AI Thinking Limitations

There's a long list of AI limitations, including some I mentioned above, but let's focus on things that humans can do, which LLMs cannot. Despite all the PR hype, the AIs are not that great at:

- Thinking generally ("generalization")
- Conceptual thinking
- Learning on-the-fly
- Continual, incremental learning

Humans can do this stuff in their sleep, literally. Children are just sponges absorbing information.

To understand how poor LLMs are at learning, consider a recent study at Apple by Shojaee et. al. (June 2025). They asked some of the best frontier models to solve some abstract puzzles and got some human-like results: as the puzzles got harder, the LLMs failed more. This is actually worse than it sounds, because they failed because they weren't using very general reasoning methods, but mostly relying on pattern recognition.

Let's give them the benefit of the doubt. We'll call it a tie.

Anyway, here's the real kicker: the folks at Apple also did another test where they told the AI exactly how to solve the puzzle. They put the answer to the puzzle in the questions, with detailed instructions on how it can be solved. All the AI needed to do was read the answer, follow the instructions on how to solve the puzzle, and they'd be done.

It made no difference.

The LLMs were completely unable to use the answer to help solve the puzzle. They just couldn't map it to their thought pattern, and were completely unable to learn. Zero on-the-fly learning capability. Were they just being stubborn?

I mean, there are humans that don't read the instructions when building flat-packed furniture, and others who won't ask the grocery store clerk where to find the *I Can't Believe It's Not Buffer* on the reach-in cooler shelves. But I think anyone else would be happy to take the answer in front of them and use it to solve a complex question. Teachers who put the answers in every question on their surprise math quiz would be immensely popular with their students, although perhaps less so with the School Board.

Weird Problems

There are some very weird things going on inside the brain of your average LLM. Here are some of the things you might see:

- Making stuff up that looks plausible but is false ("hallucinations" or "confabulations").
- Repeating the same things again that it already told you.
- Never saying it's run out of ideas.
- Never answering "I don't know" ever, ever, ever.

It's like a good friend who can't ever admit they're wrong. They make up stuff rather than backing down in an argument. But even in that case, your human friend is better than the LLM, because the AI:

Doesn't know it's lying.

In fact, it has no feelings about it at all. The LLM is just doing its best to spit out the words that look the most like a good answer. You can even challenge it, in which case, it can even spot its own mistakes, and it isn't even embarrassed by this.

I mean, it might be trained to output a few sheepish words when a mistake is found, but it doesn't really know what that means either. It's all just meaningless sequences of words.

Specific Thought Problems

The above issues are very deep and general aspects of thinking, and what thinking even means. Going deeper, there are specific types of thinking that LLMs are poor at:

- Common sense
- Empathy (the real kind, not the smarmy parroting)
- Understanding the "human condition"
- Senses, textures, touch.
- 2D spatial understanding
- 3D environment understanding
- Large search spaces (e.g., chess games)

Common sense is about a thousand little things. For example, there was a cyclone coming toward my hometown recently, so I asked an AI about what the update was. It came back with a news report from about two years ago, and pleasantly summarized it as an update for me.

That's not common sense.

If you know what a cyclone is, then you understand that I want recent information only. It's like if I ask for the Super Bowl score, I don't want one from 1957. In this case, I think you'll find that the AI gets it correct for the Super Bowl, because this is a fixable problem. It's been trained a lot on Super Bowls, and what various questions people ask, and hasn't been trained much about cyclones, because there are "hurricanes" in the Northern Hemisphere rather than the "cyclones" in Australia (they spin in opposite directions).

I was going to put "reasoning" on that list of things AI is poor at, but maybe it's no longer very true. It's still partially true because LLMs are still bad at "temporal reasoning" about time and cause-and-effect, and also "spatial reasoning" in 2D or 3D environments.

However, they're now amazing at mathematical proofs and other scientific reasoning, not to mention they got better at word puzzles and other meta-cognition about language. There has been an immense amount of research done into "reasoning models" starting with the OpenAI o1 "Strawberry" release in September, 2024.

So, the AI industry is moving fast, and some of the limitations can switch to being solved problems.

Solved Problems

Some of the capabilities that are not being mentioned as limitations in AI outputs anymore:

- Basic grammar and punctuation
- Instruction following
- Conversational capabilities
- Foreign language outputs

Some of the more technical capabilities include:

- Image file formats (e.g., JPEG).
- Output formatting correctness (e.g., tables or HTML).
- Encoding problems (e.g., emojis in UTF8 versus Unicode)
- Programming language outputs (e.g., Python or C++).
- Processing columns of numbers (like Excel).

Meta-cognition problems in AI that are largely solved:

- Tone of writing (e.g., optimistic versus negative, casual versus formal).
- Style of writing (whatever that may mean).
- Knowing when to stop (e.g., using "stop tokens").
- Following meta-requests about outputs (e.g., word counts, paragraph lengths, etc.).

The list of solved problems is getting longer with every major release.

References on AI Intelligence

Research on the nature of intelligence and its relationship to AI:

1. Maryville University Online. June 6, 2024 *Artificial Intelligence vs. Human Intelligence*, https://online.maryville.edu/blog/ai-vs-human-intelligence/
2. David De Cremer and Garry Kasparov, March 18, 2021, *AI Should Augment Human Intelligence, Not Replace It*, https://hbr.org/2021/03/ai-should-augment-human-intelligence-not-replace-it
3. Korteling JEH, van de Boer-Visschedijk GC, Blankendaal RAM, Boonekamp RC, Eikelboom AR, 2021, *Human- versus Artificial Intelligence*, Front Artif Intell. 2021 Mar 25;4:622364. doi: 10.3389/frai.2021.622364. PMID: 33981990; PMCID: PMC8108480. 2021, https://pmc.ncbi.nlm.nih.gov/articles/PMC8108480/, PDF: https://pmc.ncbi.nlm.nih.gov/articles/PMC8108480/pdf/frai-04-622364.pdf (Good article on the nature of intelligence.)
4. Shneiderman, B., 2020, *Human-Centered Artificial Intelligence: Reliable, Safe & Trustworthy*, International Journal of Human–Computer Interaction, 36(6), 495–504, https://doi.org/10.1080/10447318.2020.1741118, https://www.tandfonline.com/doi/full/10.1080/10447318.2020.1741118

Research on the limitations in AI's version of intelligence:

1. Eli Amdur, Nov 25, 2023, *Jobs AI Just Can't Do*, Forbes https://www.forbes.com/sites/eliamdur/2023/11/25/jobs-ai-just-cant-do/
2. Bernard Marr, Nov 28, 2024, *AI Won't Replace Humans – Here's The Surprising Reason Why*, Forbes, https://www.forbes.com/sites/bernardmarr/2024/11/28/ai-wont-replace-humans--heres-the-surprising-reason-why/
3. Yash Sorout, October 2023, *Exploring the Boundaries: Unveiling the Limitations and Challenges of Artificial Intelligence*, IJRAR October 2023, Volume 10, Issue 4, https://ijrar.org/papers/IJRAR23D1171.pdf (Issues like lack of common sense and boundaries to creativity.)
4. Cao, X., 2025, *The Boundaries of AI Capabilities*, In: Modern Business Management. Palgrave Macmillan, Singapore, https://doi.org/10.1007/978-981-96-0594-1_10, https://link.springer.com/chapter/10.1007/978-981-96-0594-1_10

5. Rob Toews, June 1st, 2021, *What Artificial Intelligence Still Can't Do*, https://www.forbes.com/sites/robtoews/2021/06/01/what-artificial-intelligence-still-cant-do/ (AI lacks: common sense, learning on-the-fly, understand cause-and-effect, reason ethically.)
6. Cade Metz, March 24, 2016, *One Genius' Lonely Crusade to Teach a Computer Common Sense* https://www.wired.com/2016/03/doug-lenat-artificial-intelligence-common-sense-engine/ ("For decades, as the tech world passed him by, Doug Lenat has fed computers millions of rules for daily life. Is this the way to artificial common sense?")
7. German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, Stefan Wermter, 11 Feb 2019 (v4), *Continual Lifelong Learning with Neural Networks: A Review*, https://arxiv.org/abs/1802.07569
8. James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, Raia Hadsell, 25 Jan 2017 (v2), *Overcoming catastrophic forgetting in neural networks*, https://arxiv.org/abs/1612.00796
9. Sangwon Jung, Hongjoon Ahn, Sungmin Cha, Taesup Moon, 2020, *Continual Learning with Node-Importance based Adaptive Group Sparse Regularization*, 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada, https://papers.nips.cc/paper/2020/file/258be18e31c8188555c2ff05b4d542c3-Paper.pdf

Research papers on Artificial General Intelligence (AGI), which refers to AIs with human-level intelligence:

1. Tao Feng, Chuanyang Jin, Jingyu Liu, Kunlun Zhu, Haoqin Tu, Zirui Cheng, Guanyu Lin, Jiaxuan You, 16 May 2024, *How Far Are We From AGI*, https://arxiv.org/abs/2405.10313
2. Nathan Lambert, APR 18, 2024, *Llama 3: Scaling open LLMs to AGI*, https://www.interconnects.ai/p/llama-3-and-scaling-open-llms
3. jbetke, June 3, 2024, *General Intelligence (2024)*, https://nonint.com/2024/06/03/general-intelligence-2024/
4. Ethan Mollick, May 12, 2024, *Superhuman? What does it mean for AI to be better than a human? And how can we tell?* https://www.oneusefulthing.org/p/superhuman
5. Rohin Shah, Seb Farquhar, Anca Dragan, 21st Aug 2024, *AGI Safety and Alignment at Google DeepMind: A Summary of Recent Work*, https://www.alignmentforum.org/posts/79BPxvSsjzBkiSyTq/agi-safety-and-alignment-at-google-deepmind-a-summary-of

6. Vishal Rajput, Jul 8, 2024, *Why LLMs Can't Plan And Unlikely To Reach AGI?* https://medium.com/aiguys/why-llms-cant-plan-and-unlikely-to-reach-agi-642bda3e0aa3
7. David Gilmore, Sep 2024, *When will AI outthink humans?* https://davidvgilmore.com/writings/outthinking-ai (Interesting analysis of all the GPUs in the world and when they will "out-think" all the human knowledge workers, predicting a range of years from 2028 to 2035, depending on assumptions.)
8. Chloe Berger, October 2, 2024, *Mark Cuban says his puppy is 'smarter than AI is today'*, https://fortune.com/2024/10/01/mark-cuban-dog-puppy-smarter-than-ai/
9. Samantha Kelly, Sept. 29, 2024, *"Superintelligent" AI Is Only a Few Thousand Days Away: OpenAI CEO Sam Altman*, https://www.cnet.com/tech/services-and-software/superintelligent-ai-is-only-a-few-thousand-days-away-openai-ceo-sam-altman/
10. Brian Merchant, Dec 2024, *AI Generated Business: The Rise of AGI and the Rush to Find a Working Business Model*, https://ainowinstitute.org/general/ai-generated-business
11. Alhassan Mumuni, Fuseini Mumuni, 6 Jan 2025, *Large language models for artificial general intelligence (AGI): A survey of foundational principles and approaches*, https://arxiv.org/abs/2501.03151
12. Jeffrey Anthony, Jan 2025, *No GPT-5 in 2025 and No AGI — Ever. The Triadic Nature of Meaning-Making and the Fallacy of AI's Understanding*, https://medium.com/@WeWillNotBeFlattened/no-gpt-5-in-2025-and-no-agi-ever-aa9384efdbe5
13. Mohit Sewak, Ph.D., January 29, 2025, *Achieving General Intelligence (AGI) and Super Intelligence (ASI): Pathways, Uncertainties, and Ethical Concerns*, https://towardsai.net/p/l/achieving-general-intelligence-agi-and-super-intelligence-asi-pathways-uncertainties-and-ethical-concerns
14. Alberto Romero, Feb 06, 2025, *AGI Is Already Here—It's Just Not Evenly Distributed: Or: why you should learn to prompt AI models*, https://open.substack.com/pub/thealgorithmicbridge/p/agi-is-already-hereits-just-not-evenly
15. Apoorv Agrawal, May 23, 2025, *Why Cars Drive Themselves Before Computers Do: Robocars are ready; robot secretaries aren't… yet*, https://apoorv03.com/p/autonomy

16. Parshin Shojaee, Maxwell Horton, Iman Mirzadeh, Samy Bengio, Keivan Alizadeh, June 2025, *The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity*, Apple, https://machinelearning.apple.com/research/illusion-of-thinking https://ml-site.cdn-apple.com/papers/the-illusion-of-thinking.pdf
17. Dr. Ashish Bamania, June 2025, *Apple's New Research Shows That LLM Reasoning Is Completely Broken: A deep dive into Apple research that exposes the flawed thinking process in state-of-the-art Reasoning LLMs*, https://ai.gopubby.com/apples-new-research-shows-that-llm-reasoning-is-completely-broken-47b5be71a06a

4. Weak Words

"Stitching Together Sequences of Linguistic Forms...

Without Any Reference To Meaning:

A Stochastic Parrot."

— Bender et al., 2021.

Words and Dumbness

AI models work mainly on words, and therein lies the rub. It's really tough to do some things using only words. Yes, you can write some exciting poetry in word patterns, but there are problems with:

- Numbers and arithmetic
- Real-life meanings of words
- Common sense
- 3D world modeling
- Time modeling ("cause and effect")

Words have meaning beyond their basic wordliness, and that's where the LLM has problems. Words are not good at representing common sense issues.

Common sense are things that we know immediately and take for granted. Here are some common sense ideas:

- Feathers are lighter than stones and fall slower.
- Moving a glass of water differs from moving a drop of water.

Unfortunately, an LLM these things are just words, and that's all they know.

Strawberries

If LLMs just look at words, why are they terrible at solving crosswords? There were also difficulties with word scrambles or anagrams, and asking for a list of words that start with a particular letter.

A lot of these problematic areas have gotten better recently, but LLMs were initially awful at them.

Even some basic tasks that seemed to be like spelling a word were problematic for early LLMs. The most famous example is:

> *How many letter r's in "Strawberry"?*

ChatGPT used to get this wrong, confidently stating that there are two. Is the extra one hidden behind the 't'?

It's not really clear why ChatGPT answered this incorrectly, but this error spawned the codename "Project Strawberry" for the more advanced "reasoning model" from OpenAI named "GPT-4o" that was released in late 2024.

Weirdly, the problem is words.

Counting letters in a word is not really a problem you can solve with words. In fact, it's a "meta-problem" at a level higher than just outputting words. Hence, basic LLMs that think only in words could not perform detailed reasoning about the properties of words, and required extra steps in reasoning models to do so correctly.

Illegal Chess Moves

I tried to play a game of chess against ChatGPT the other day. My opponent was very keen and quite helpful, chattering along with every move. The game went:

1. e4 e5

2. Nf3 Nc6

3. Bc4 Bc5

4. c3 Nf6

5. b4 Bxb4

At this point, the game ended. ChatGPT complimented me on playing the Evans Gambit with the "b4" pawn move, which is apparently an interesting and enterprising type of chess opening. When I tried to respond with my next move, "6. c3xb4" taking the Bishop, I was informed that it was an illegal move.

I guess we can call it a draw.

Sorry, that's an inside joke only for chess players. Suffice it to say that ChatGPT was making some very basic mistakes:

1. I didn't play the Evans Gambit.

2. The ChatGPT move "Bxb4" is terrible (needlessly losing the Bishop).

3. The move I want to play is not illegal.

What's going on here?

The problem is words, again. Chess is not a word game, but ChatGPT is trying to play chess using words, and it got confused. I did play the "b4" move, which looks like the Evans Gambit, but I actually played it one move later than normal, but that nuance was lost on the AI engine. The computer move "Bxb4" is a good move in the real Evans Gambit, but is terrible when played one move later. Similarly, the move I wanted to play would be illegal in the real Evans Gambit line, but is not in the modified version that I played.

There are several issues causing these problems:

1. ChatGPT hasn't been fully trained on all chess openings (e.g., my delayed "b4" move is an uncommon opening that it clearly hasn't been trained on), and
2. There's no "chess engine" being used, so ChatGPT is relying only on words.

Both of these issues are easily fixable, because there are whole books of chess openings available, and even a simple chess engine would suffice to make a playable opponent (or, at least, correctly identify legal versus illegal moves).

Hence, I presume that ChatGPT will be smashing me in our next encounter.

Bad Names

There's a running joke in the industry about all its bad names. You'd think that with an industry that's just replete with uber-brainy personnel, there'd be some much better naming choices.

Oh, wait, umm, maybe I just found the problem.

OpenAI openly jokes about their bad ideas for names, like "GPT-4o" for a massive model release that deserved a much bigger reputation. Sam Altman has been quoted:

> "*We deserve the roasting we're getting for the names. We will do better.*"

At least, Google has got their stuff together with their "Gemini" models, Anthropic has some personality with "Claude," xAI has "Grok," Alibaba has "Ernie," IBM was "Watson," and NVIDIA now has "Dynamo" and "NeMo" software.

Actually, most of those names are great, so, maybe it's just OpenAI that needs to hire a marketing manager or two.

Really, the researchers were the ones who started the whole thing. It's not just product names lacking marketing panache. What were all those Ph.D.'s thinking?

Here are some real acronyms: RELU, GELU, SiLU, SwigLU, PEFT, NAS. The silver medal winner here is certainly "RELU" standing for "Rectilinear Unit" or something like that. All that RELU does is take negative numbers and make them zero. That's all. It just makes sure all numbers are non-negative. Why does it even need an acronym?

And some other real names for technological issues in AI: knowledge distillation, hallucination, catastrophic forgetting, kernel fission, tensors, model collapse, optimal brain damage, slimmable networks, exploding gradient, regurgitative training, zombie weights, lottery ticket hypothesis, and mechanistic interpretability.

The gold medal surely goes to Retrieval Augmented Generation (RAG). The leader author of the original RAG research paper, Patrick Lewis, is on record stating:

> "*We definitely would have put more thought into the name had we known our work would become so widespread...*"

But it's not all about obscure names. There are some research algorithm names that have some extra memorability: Transformers, MoE, LLaMa, LoRA, Flash attention, Medusa, Eagle, RoPE, NoPE, Mambo, and Hyena.

Over-Alignment

Alignment with human wishes is a tricky problem. There are hundreds and thousands of papers on how to "align" the output of an LLM with whatever weirdness a human might want. Usually, alignment is desirable, and the main problem is to fix instances of unalignment.

Alignment is mostly a good thing, but also possible is too much of a good thing. Be careful what you wish for! If you tell an AI that you want to rob a bank, what response should you get:

- Assuming you're joking and laughing with you?
- Taking your seriously and talking you out of the plan.
- Blocking or refusing to answer completely.
- Helping you to choose the best bank to rob.

So, the issue here is one of ethics. The usual solution for most LLMs is to code up a "refusal module" and it will refuse to answer this query. Or, rather, it will have been trained to emit a politely worded refusal, rather than detailed instructions on how to go about your task.

Over-alignment is also possible and it's known as "sycophancy" in the trade. In April, 2025, OpenAI had to roll back a version of their ChatGPT model because it was too sycophantic in its responses. They followed up with a couple of detailed research analyses about what went wrong.

3D Worldview

Visualizing what is happening in 3D is a subset of common sense. We understand what it's like to move around in a three-dimensional world, having done so since infancy. There are commonsense rules like:

- Two people don't stand in exactly the same space.
- Things fall down, not up.
- The walls of a room are usually vertical.
- A cup will sit on a table but not in mid-air.
- People can crawl under a table, but not through it.

I mean, 2D is hard enough for the poor LLM. There are all sorts of hidden rules for two-dimensional spatial data, like maps, GPS locations, printed pages, and computer screens:

- If you move North, then East, you are now North-East of where you started.
- The fastest way to travel from A to B is a straight line (in geometry).
- Typesetting usually should put text in horizontal rows.

Worse still, there are exceptions to apparently sane rules. If you go all around the Equator on a 2D map, moving right, you'll get to the other side of the map on the left.

Similarly, the fastest way for a jumbo jet to fly from New York to Paris is a "great circle" and not a straight line on the map. There's that 3D logic again, spoiling everything.

Catastrophic Forgetting

You open up a chatbot session and you tell it your name, and the AI generates lots of useful stuff. Later, you open another session, and it doesn't remember your name. That's "catastrophic forgetting."

Not really a catastrophe!

But that's the official term and it's used in research papers, starting with Kirkpatrick et al. (2017).

It's not just between sessions, but also inside a long session. It used to be that the big models only had a "context window" with a 4,096 token length limit (i.e., a "4k context"). Once you get to the 4,097th token, it gets a bit fuzzy about the first one.

If you give the LLM a 100,000 word novel with 3,000 word chapters to analyze, it's forgotten what happened in the first chapter by the time it's moved onto the third one.

Fortunately, this isn't as much of a problem now with modern LLMs, and catastrophic forgetting has been long forgotten (LOL). Most modern LLMs now have a "long context" of 128K or more, and there's already a generation of "ultra-long context" LLMs that have a context window of one million tokens. Hence, LLMs are more like elephants than llamas now.

Slowness

Most of the problems with AI engines and their weird limitations actually make some level of sense if you think about them this way:

> *What tasks can the human brain only do slowly.*

Things like solving crossword puzzles and playing chess games are not automatic for our brain, but require us to logically think it through. They're using the "slow brain" rather than the "fast brain" mode.

There's the problem!

AI technology has solved the fast mode with wall-to-wall GPU chips, but is much weaker in the slow mode. The newer generation of "reasoning models," starting with GPT-4o in late 2024, are a step towards rational step-by-step thought, which are much better at math problems and crosswords, but they don't solve everything. Slow thinking is still a work in progress.

On the other hand, there are several limitations of AI models that humans can do fast. We have an innate understanding of 3D world layouts, and we know cause-and-effect at a very profound level.

Humans do these things fast!

References

General references on various AI limitations:

1. Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell, 2021, *On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?* 🦜. Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT '21., Association for Computing Machinery, New York, NY, USA, 610–623, https://doi.org/10.1145/3442188.3445922, https://dl.acm.org/doi/10.1145/3442188.3445922
2. Parshin Shojaee, Maxwell Horton, Iman Mirzadeh, Samy Bengio, Keivan Alizadeh, June 2025, *The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity*, Apple, https://machinelearning.apple.com/research/illusion-of-thinking, https://ml-site.cdn-apple.com/papers/the-illusion-of-thinking.pdf
3. Bernard Marr, Aug 19, 2024, *Why AI Models Are Collapsing And What It Means For The Future Of Technology*, https://www.forbes.com/sites/bernardmarr/2024/08/19/why-ai-models-are-collapsing-and-what-it-means-for-the-future-of-technology/
4. Andrew Orlowski, 14 July 2025, *The great AI delusion is falling apart: New research suggests the chorus of techno-optimism is based on falsehoods*, https://www.telegraph.co.uk/business/2025/07/14/the-great-ai-delusion-is-built-on-self-deception/
5. Rob Toews, June 1st, 2021, *What Artificial Intelligence Still Can't Do*, Forbes, https://www.forbes.com/sites/robtoews/2021/06/01/what-artificial-intelligence-still-cant-do/ (AI lacks: common sense, learning on-the-fly, understand cause-and-effect, reason ethically.)
6. Cade Metz, March 24, 2016, *One Genius' Lonely Crusade to Teach a Computer Common Sense*, Wired Magazine, https://www.wired.com/2016/03/doug-lenat-artificial-intelligence-common-sense-engine/ ("For decades, as the tech world passed him by, Doug Lenat has fed computers millions of rules for daily life. Is this the way to artificial common sense?")
7. Charles Q. Choi, 21 Sep 2021, *7 Revealing Ways AIs Fail: Neural networks can be disastrously brittle, forgetful, and surprisingly bad at math*, IEEE Spectrum, https://spectrum.ieee.org/ai-failures
8. Peter Gärdenfors, 14 October 2024, *AI lacks common sense – why programs cannot think*, Lund University, https://www.lunduniversity.lu.se/article/ai-lacks-common-sense-why-programs-cannot-think

References on "catastrophic forgetting" in AI:

1. James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, Raia Hadsell, 25 Jan 2017 (v2), *Overcoming catastrophic forgetting in neural networks*, https://arxiv.org/abs/1612.00796

References on AI chess playing problems:

1. Victor Tangermann, Sep 13, 2024, *OpenAI's New "Strawberry" AI Is Still Making Idiotic Mistakes*, https://futurism.com/openai-strawberry-o1-mistakes
2. Dynomight, Nov 2024, *Something weird is happening with LLMs and chess*, https://dynomight.net/chess/
3. Dynomight, Nov 2024, *OK, I can partly explain the LLM chess weirdness now*, https://dynomight.net/more-chess/

References on sycophancy:

1. OpenAI, April 29, 2025, *Sycophancy in GPT-4o: what happened and what we're doing about it*, https://openai.com/index/sycophancy-in-gpt-4o/
2. OpenAI, May 2, 2025 *Expanding on what we missed with sycophancy: A deeper dive on our findings, what went wrong, and future changes we're making*, https://openai.com/index/expanding-on-sycophancy/
3. Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, Ethan Perez, 10 May 2025 (v4), *Towards Understanding Sycophancy in Language Models*, https://arxiv.org/abs/2310.13548

References on bad names in the AI industry:

1. Rick Merritt, January 31, 2025, *What Is Retrieval-Augmented Generation, aka RAG?*, https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/ ("We definitely would have put more thought into the name had we known our work would become so widespread," stated Patrick Lewis.)
2. Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, Douwe Kiela, 12 Apr 2021 (v4), *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*, https://arxiv.org/abs/2005.11401 (The paper that spawned yet another oddly named AI technology.)
3. Lex Friedman, March 2024, *Transcript for Sam Altman: OpenAI, GPT-5, Sora, Board Saga, Elon Musk, Ilya, Power & AGI*, Lex Fridman Podcast #419, https://lexfridman.com/sam-altman-2-transcript/ (Sam Altman quote: "We deserve the roasting we're getting for the names. We will do better.")

5. Neurology Versus Numbers

"No, I'm not interested in developing a powerful brain."

— Alan Turing

Neurology versus Numbers

The AI models are very similar to the human brain, since they are programming models of its neurons and synapses. There's another similarity between LLMs and brains:

Nobody knows how they work!

Sure, there are neurologists who know about the brain, and technologists that know about AI models. But both areas are really quite underwhelming when you consider how little is known.

For example, neurologists really don't know:

Why do we need sleep?

What causes dementia?

Incidentally, the answer may be the same: clearing away brain cell detritus. Our brain cells clean up waste by-products during sleep, and dementia may be an impairment in that process.

Anyway, technologists really don't understand their computerized AI models either. We can run them on huge computing clusters, and we know what they can do, but our understanding of *how* the models do these things is a little vague. For example, questions that AI researchers struggle with include:

> Why do models lack common sense?
>
> Why do they make stuff up?

Amusingly, the fact is that for every neurology area of study, there's a parallel research area in artificial brains.

Activated Brain Regions

You don't use your whole brain all of the time. And I don't mean that "only 10%" rule that Hollywood is fond of. What I mean is that there are different regions of the brain that are "activated" by different activities. For example, we know about "lobes" in the brain:

- Frontal lobe (front) — high-level functioning (e.g., decisions, motor control, speech).
- Parietal lobe (top) — sensory inputs, numbers/mathematics, and hand-eye coordination.
- Temporal lobe (sides) — audio input, hearing, memory, and language understanding.
- Occipital lobe (back) — higher-level visual processing (e.g., detecting shapes and movement).

And there are many more sub-regions of the brain that have been analyzed in much more depth:

- Brainstem — basic life-preserving functions (breathing, heartbeat).
- Motor cortex — voluntary movement controls.
- Visual cortex — processing what they eyes see.
- Cerebrum — high-level rationality and logic.
- Cerebellum — movement and motor functions.

Neurology researchers have studied these brain regions in detail by: (a) the great joy of dissection, and (b) having people do different things while their brain is being watched by an MRI. You can literally see that different parts of the brain "light up" in response to different stimuli or when the brain is doing different types of work. Other parts of the brain are stuck in idle.

AI researchers are doing two main things in response to these types of neurology research:

1. Copying it!

2. Studying AI models.

I'm not sure which one came first, but AI engineers are doing both.

AI Copies of Brain Regions

The brain's structure is much more complicated than an AI model. In fact, whereas the brain has all sorts of different components scattered all around the skull, an LLM is rather simple in structure.

Your brain is rather special in structure, and un-symmetrical. In fact, the left hemisphere and right hemisphere do different things, which isn't the case in AI models. Instead, the average AI model is rigid in its structure, and symmetrical in all three dimensions.

The "Mixture-of-Experts" or "MoE" architecture, as pioneered in GPT-4, is like dividing the AI model into brain regions. Different parts of an AI model get activated in response to different questions.

Firstly, note that experts aren't chosen based on the whole question you ask them. It's not like the model analyzes the input and then decides which part of the LLM will handle the whole thing. That's not the Mixture-of-Experts architecture that I'm talking about here, although there is a different thing that does work that way, which is called "model routing."

But here, I'm talking about having an LLM where internal submodels inside the model are turned on and off while it's processing a response.

Note that the "experts" in an LLM are split on the strength of the signals along the "embedding dimension" of the matrices, also known as the "width" of the model. It's not that one expert handles a word, and then a different expert does the next one, which is called the "lengthwise" dimension (or token dimension). Rather, each word is converted into "embeddings" as a type of "semantic vector" that gives it various weights on different signals (e.g., is it a noun like "cat" or is it a verb like "jump"). The model figures out which signals are strongest, and hence, which expert should process the data. Hence, simplifying greatly, a different expert would process a noun versus another expert that processes verbs.

You can see how this would be efficient. Having the DeepSeek R1 model would be slow if all 600 billion weights were used for every word. Instead, it only uses its 37 billion weights in each expert for each word. It's potentially a different set of 37 billion, changing each time it goes around to output another word, but it's still 600/37=16 times faster at each word, and hence, also that much faster for producing the whole output text. You can also see why this might be faster than GPT-4, which has about 200 billion weights in its experts, compared to only 37 billion in DeepSeek.

Vision and audio processing in AI models is very different in a sense (haha). There's not really a full equivalent of the visual cortex in an LLM. Usually, the LLM doesn't get a "raw feed" of vision or sound like the brain does, but instead gets one that's already had factors like color and intensity computed for it.

Hence, the brain has some neuron power assigned to processing raw input signals, which the LLMs don't need.

Studying AI Model Numbers

AI researchers have been studying artificial brains in some depth, and you don't even need to use an MRI to do this. In fact, you can just code it up. It sounds easy, but in reality, it's a little tricky to analyze model weights, because it's like trying to discern something interesting about what is literally a billion random numbers.

The whole area of studying AI weight numbers has a really great name: *mechanistic interpretability*. I'm not even sure what to write about that. I mean, sure, it's interpreting the numbers, I guess, and they're "mechanistic" in some way. Thankfully, most AI researchers shorten it to "mech interp" to hide the fact that they still don't understand what the numbers actually do.

Nevertheless, AI researchers have discovered all sort of intriguing fact about LLM weights. These are just numbers, and they can be examined and also changed. Weirdness ensues in multiple ways:

- Attention sink — the first token.
- Layer importance — three levels of AI.
- Too many weights — thinning and shrinking numbers.

One weird thing is the "attention sink" research, which finds that there's always one token that gets far too much attention: the very first one. So, the first word you use in a prompt matters a lot more than any other single word, and no-one really knows why that is. Maybe it's not weird to think that AI only really listens to one word, because that's how teenagers work.

Layers are another area of research. At the highest level is the study of "layer importance" in AI models. The structure of every model is that the computation goes repeatedly through several layers, sometimes hundreds, of identical number crunching.

Like in *Shrek*, layers.

The different layers of the model matter and they do different things. The early layers of an LLM do the high-level topic selection and guidance, whereas the middle layers start choosing the most likely next word to output. The final layers are "finesse" layers that choose between several good options for the best output.

Researchers have studied this by looking internally how the numbers change after each layer, and also by doing "layer skipping" or "early exiting" to see what happens if some of the layers are removed. Turns out that you can actually skip a lot of layers and still have a reasonable level of intelligence.

Maybe LLMs are only really using 10% of their AI brains?

The 10% Rule

Could it be that LLMs are like humans and waste their brain capacity? It's not actually a joke, because AI models are often shrunk to be smaller. Turns out if you have billions of numbers, and you don't understand what they all do, then there's some redundancy in there.

One technique is called "quantization" and that name is probably because it has *nothing* to do with quantum physics. Anyway, that's the name of the most popular technique.

The goal is to make the model smaller, so it can run faster, while giving up some accuracy. In other words: a lot faster but slightly dumber.

Common levels of shrinkage, ahem, I mean, quantization, are from 32-bit to 4-bit, which is an eight-fold decline in LLM brain size. In other words, a 4-bit quantized model is only using 12.5% of its original brain power.

There's also another way that is called "pruning" (good name) or "sparsification" (bad name), and involves setting numbers to zero. This is equivalent to "pruning" (removing) a synaptic connection between two neurons. In neurosurgery, this is done to block connections that are causing problems, such as in epilepsy, whereas in AI the weights are mainly removed just to get a speedup. Hence, different goals for pruning, but you can even combine pruning techniques with quantization.

So, yes, we can get below 10% usage of an LLM brain.

Neurosurgery on AI Brains

Researchers haven't just studied the numbers, but have also changed them. It's like stimulating neurons in the LLM with an electric prod, just like they do in neurosurgery with Deep Brain Stimulation (DBS) to treat Parkinson's disease.

The basic idea in AI research is called "activation patching" where you modify the outputs of each of those layers as things progress. If you change the numbers, then the output will change.

Note that this is not just about suppressing neurons, as in "sparsifying" the numbers by putting lots of them to zero. That's a speedup, but activation patching is about changing the results, not just making it run faster.

What can you change?

Turns out, it's not easy. There's no simple algorithm to look at the numbers in an AI model and know what to change. Who knew that brain surgery could be so difficult?

Researchers have ways of changing the numbers, but they're not easy algorithms, and always involve watching the numbers first. The most fun you can have on a Sunday night in San Jose is running your AI engines and watching how the numbers change.

The most successful idea here is called "attention steering" or just "steering" for short. The idea is that you can "steer" the AI model to be more in some written tone, such as optimistic or pessimistic. The way this works is only in pairs of styles. The idea is that you run your AI engine with prompts that you know will cause optimistic or pessimistic outputs, and you know which is which. Then you look at the numbers from both, and run a "diff" on a big vector of numbers to find the numbers that are different in optimistic versus pessimistic answers. And that means the internal numbers along the way, not just the different types of words at the end, which would be too easy.

That difference vector is a list of numbers that is the set of activations that are different between the two styles. Hence, you end up with two vectors, one for positive, and one for negative. Later, when you want to do "steering" you can modify your model layers so that these extra numbers are added to the outputs.

Bizarrely, this actually works.

Unfortunately, it's hard to drill down deeper than this to give more steering control. We can find the signals that tend towards positivity or negativity, but we can't really identify the meaning of every number in the vector of numbers. Each number must mean something, or be a signal toward some output, but which one is what output?

Anyway, I mean, it's intriguing research, and I love it so much, but what's the practical point? It's easier to just command your AI to "please output in an optimistic tone" at the end of your prompt. That works, too.

References

References on the "attention sink" where the very first token gets too much "attention":

1. Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, Min Lin, 14 Oct 2024, *When Attention Sink Emerges in Language Models: An Empirical View*, https://arxiv.org/abs/2410.10781 https://github.com/sail-sg/Attention-Sink

2. Tianyu Guo, Druv Pai, Yu Bai, Jiantao Jiao, Michael I. Jordan, Song Mei, 17 Oct 2024, *Active-Dormant Attention Heads: Mechanistically Demystifying Extreme-Token Phenomena in LLMs*, https://arxiv.org/abs/2410.13835
3. Zunhai Su, Zhe Chen, Wang Shen, Hanyu Wei, Linge Li, Huangqi Yu, Kehong Yuan, 25 Jan 2025, *RotateKV: Accurate and Robust 2-Bit KV Cache Quantization for LLMs via Outlier-Aware Adaptive Rotations*, https://arxiv.org/abs/2501.16383
4. Xinyi Wu, Yifei Wang, Stefanie Jegelka, Ali Jadbabaie, 4 Feb 2025, *On the Emergence of Position Bias in Transformers*, https://arxiv.org/abs/2502.01951

References on "layer importance" inside AI model structures:

1. Anton Razzhigaev, Matvey Mikhalchuk, Elizaveta Goncharova, Nikolai Gerasimenko, Ivan Oseledets, Denis Dimitrov, Andrey Kuznetsov, 19 May 2024, *Your Transformer is Secretly Linear*, https://arxiv.org/abs/2405.12250 (Replacing model layers in the decoder with linear approximations.)
2. BS Akash, V Singh, A Krishna, LB Murthy, L Kumar, April 2024, *Investigating BERT Layer Performance and SMOTE Through MLP-Driven Ablation on Gittercom*, Lecture Notes on Data Engineering and Communications Technologies (LNDECT, volume 200), https://link.springer.com/chapter/10.1007/978-3-031-57853-3_25
3. Jiachen Jiang, Jinxin Zhou, Zhihui Zhu, 20 Jun 2024, *On Layer-wise Representation Similarity: Application for Multi-Exit Models with a Single Classifier*, https://arxiv.org/abs/2406.14479 (Using layer similarity for early exit classifiers, which is also related to layer fusion.)
4. Vedang Lad, Wes Gurnee, Max Tegmark, 27 Jun 2024, *The Remarkable Robustness of LLMs: Stages of Inference*, https://arxiv.org/abs/2406.19384 (Deleting and swapping adjacent model layers. Hypothesizes that the first layer is effectively detokenization, the early layers focus on "features", the middle layers focus on "ensemble predictions" and the latter layers "sharpen" or finalize, with a lot of suppression happening near the end.)
5. Xu Cheng, Lei Cheng, Zhaoran Peng, Yang Xu, Tian Han, Quanshi Zhang, July 2024, *Layerwise Change of Knowledge in Neural Networks*, Proceedings of the 41st International Conference on Machine Learning, PMLR 235:8038-8059, 2024, https://proceedings.mlr.press/v235/cheng24b.html PDF: https://raw.githubusercontent.com/mlresearch/v235/main/assets/cheng24b/cheng24b.pdf
6. Siqi Fan, Xin Jiang, Xiang Li, Xuying Meng, Peng Han, Shuo Shang, Aixin Sun, Yequan Wang, Zhongyuan Wang, 9 Jul 2024 (v3), *Not All Layers of LLMs Are Necessary During Inference*, https://arxiv.org/abs/2403.02181

7. Amit Ben Artzy, Roy Schwartz, 5 Sep 2024, *Attend First, Consolidate Later: On the Importance of Attention in Different LLM Layers*, https://arxiv.org/abs/2409.03621
8. Tianyu Guo, Druv Pai, Yu Bai, Jiantao Jiao, Michael I. Jordan, Song Mei, 17 Oct 2024, *Active-Dormant Attention Heads: Mechanistically Demystifying Extreme-Token Phenomena in LLMs*, https://arxiv.org/abs/2410.13835
9. Khashayar Gatmiry, Nikunj Saunshi, Sashank J. Reddi, Stefanie Jegelka, Sanjiv Kumar, 29 Oct 2024, *On the Role of Depth and Looping for In-Context Learning with Task Diversity*, https://arxiv.org/abs/2410.21698
10. Guangyuan Shi, Zexin Lu, Xiaoyu Dong, Wenlong Zhang, Xuanyu Zhang, Yujie Feng, Xiao-Ming Wu, 23 Oct 2024, *Understanding Layer Significance in LLM Alignment*, https://arxiv.org/abs/2410.17875
11. Jason Du, Kelly Hong, Alishba Imran, Erfan Jahanparast, Mehdi Khfifi, Kaichun Qiao, 13 Jan 2025, *How GPT learns layer by layer*, https://arxiv.org/abs/2501.07108 https://github.com/ALT-JS/OthelloSAE
12. Ming Li, Yanhong Li, Tianyi Zhou, 31 Oct 2024, *What Happened in LLMs Layers when Trained for Fast vs. Slow Thinking: A Gradient Perspective*, https://arxiv.org/abs/2410.23743
13. Jiachen Jiang, Yuxin Dong, Jinxin Zhou, Zhihui Zhu, 22 May 2025, *From Compression to Expansion: A Layerwise Analysis of In-Context Learning*, https://arxiv.org/abs/2505.17322

References on "mechanistic interpretability" or "mech interp" if you're trendy:

1. Neel Nanda, 8th Jul 2024, *An Extremely Opinionated Annotated List of My Favourite Mechanistic Interpretability Papers v2*, https://www.alignmentforum.org/posts/NfFST5Mio7BCAQHPA/an-extremely-opinionated-annotated-list-of-my-favourite
2. Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, Ziyu Yao, 2 Jul 2024, *A Practical Review of Mechanistic Interpretability for Transformer-Based Language Models*, https://arxiv.org/abs/2407.02646
3. Mengru Wang, Yunzhi Yao, Ziwen Xu, Shuofei Qiao, Shumin Deng, Peng Wang, Xiang Chen, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Huajun Chen, Ningyu Zhang, 4 Dec 2024 (v4), *Knowledge Mechanisms in Large Language Models: A Survey and Perspective*, https://arxiv.org/abs/2407.15017
4. Leonard Bereska, Efstratios Gavves, 23 Aug 2024 (v3), *Mechanistic Interpretability for AI Safety -- A Review*, https://arxiv.org/abs/2404.14082
5. Chashi Mahiul Islam, Samuel Jacob Chacko, Mao Nishino, Xiuwen Liu, 7 Feb 2025, *Mechanistic Understandings of Representation Vulnerabilities and Engineering Robust Vision Transformers*, https://arxiv.org/abs/2502.04679

6. Ala N. Tak, Amin Banayeeanzade, Anahita Bolourani, Mina Kian, Robin Jia, Jonathan Gratch, 8 Feb 2025, *Mechanistic Interpretability of Emotion Inference in Large Language Models*, https://arxiv.org/abs/2502.05489
7. Lukasz Bartoszcze, Sarthak Munshi, Bryan Sukidi, Jennifer Yen, Zejia Yang, David Williams-King, Linh Le, Kosi Asuzu, Carsten Maple, 24 Feb 2025, *Representation Engineering for Large-Language Models: Survey and Research Challenges*, https://arxiv.org/abs/2502.17601
8. Samuel Miller, Daking Rai, Ziyu Yao, 20 Feb 2025, *Mechanistic Understanding of Language Models in Syntactic Code Completion*, https://arxiv.org/abs/2502.18499
9. Yifan Zhang, Wenyu Du, Dongming Jin, Jie Fu, Zhi Jin, 27 Feb 2025, *Finite State Automata Inside Transformers with Chain-of-Thought: A Mechanistic Study on State Tracking*, https://arxiv.org/abs/2502.20129
10. J. Katta, M. Allanki and N. R. Kodumuru, 2025, *Understanding Sarcasm Detection Through Mechanistic Interpretability*, 2025 4th International Conference on Sentiment Analysis and Deep Learning (ICSADL), Bhimdatta, Nepal, 2025, pp. 990-995, doi: 10.1109/ICSADL65848.2025.10933475. https://ieeexplore.ieee.org/abstract/document/10933475/
11. Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, Stella Biderman, Adria Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, Eric J. Michaud, Stephen Casper, Max Tegmark, William Saunders, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, Tom McGrath, 27 Jan 2025, *Open Problems in Mechanistic Interpretability*, https://arxiv.org/abs/2501.16496
12. Jingcheng Niu, Xingdi Yuan, Tong Wang, Hamidreza Saghir, Amir H. Abdi, 14 May 2025, *Llama See, Llama Do: A Mechanistic Perspective on Contextual Entrainment and Distraction in LLMs*, https://arxiv.org/abs/2505.09338
13. Vishnu Kabir Chhabra, Mohammad Mahdi Khalili, 5 Apr 2025, *Towards Understanding and Improving Refusal in Compressed Models via Mechanistic Interpretability*, https://arxiv.org/abs/2504.04215

References on "activation patching" (like deep brain stimulation):

1. Neel Nanda, Feb 4, 2024, *Attribution Patching: Activation Patching At Industrial Scale*, https://www.neelnanda.io/mechanistic-interpretability/attribution-patching
2. Clément Dumas, Chris Wendler, Veniamin Veselovsky, Giovanni Monea, Robert West, 9 Jan 2025 (v3), *Separating Tongue from Thought: Activation Patching Reveals Language-Agnostic Concept Representations in Transformers*, https://arxiv.org/abs/2411.08745

3. Wei Jie Yeo, Ranjan Satapathy, Erik Cambria, 1 Nov 2024 (v2), *Towards Faithful Natural Language Explanations: A Study Using Activation Patching in Large Language Models*, https://arxiv.org/abs/2410.14155
4. Stefan Heimersheim, Neel Nanda, 23 Apr 2024, *How to use and interpret activation patching*, https://arxiv.org/abs/2404.15255
5. Aleksandar Makelov, Georg Lange, Neel Nanda, 6 Dec 2023 (v2), *Is This the Subspace You Are Looking for? An Interpretability Illusion for Subspace Activation Patching*, https://arxiv.org/abs/2311.17030
6. Fred Zhang, Neel Nanda, 17 Jan 2024 (v2), *Towards Best Practices of Activation Patching in Language Models: Metrics and Methods*, https://arxiv.org/abs/2309.16042

References on "attention steering" that changes AI behavior by playing with its internal numbers:

1. Zhuohan Gu, Jiayi Yao, Kuntai Du, Junchen Jiang, 21 Nov 2024 (v2), *LLMSteer: Improving Long-Context LLM Inference by Steering Attention on Reused Contexts*, https://arxiv.org/abs/2411.13009
2. Qingru Zhang, Chandan Singh, Liyuan Liu, Xiaodong Liu, Bin Yu, Jianfeng Gao, Tuo Zhao, 1 Oct 2024 (v2), *Tell Your Model Where to Attend: Post-hoc Attention Steering for LLMs*, https://arxiv.org/abs/2311.02262 https://github.com/QingruZhang/PASTA
3. Baifeng Shi, Siyu Gai, Trevor Darrell, Xin Wang, 11 Jul 2023 (v2), *TOAST: Transfer Learning via Attention Steering*, https://arxiv.org/abs/2305.15542 https://github.com/bfshi/TOAST
4. Yibin Wang, Weizhong Zhang, Jianwei Zheng, Cheng Jin, 20 Aug 2024 (v3), *PrimeComposer: Faster Progressively Combined Diffusion for Image Composition with Attention Steering*, https://arxiv.org/abs/2403.05053 https://github.com/CodeGoat24/PrimeComposer
5. Kyle O'Brien, David Majercak, Xavier Fernandes, Richard Edgar, Jingya Chen, Harsha Nori, Dean Carignan, Eric Horvitz, Forough Poursabzi-Sangde, 18 Nov 2024, *Steering Language Model Refusal with Sparse Autoencoders*, https://arxiv.org/abs/2411.11296
6. Xintong Wang, Jingheng Pan, Longqin Jiang, Liang Ding, Xingshan Li, Chris Biemann, 23 Oct 2024, *CogSteer: Cognition-Inspired Selective Layer Intervention for Efficient Semantic Steering in Large Language Models*, https://arxiv.org/abs/2410.17714
7. Hanyu Zhang, Xiting Wang, Chengao Li, Xiang Ao, Qing He, 10 Jan 2025, Controlling Large Language Models Through Concept Activation Vectors, https://arxiv.org/abs/2501.05764 (Training a vector used to control the model on certain attributes.)

8. Liu Yang, Ziqian Lin, Kangwook Lee, Dimitris Papailiopoulos, Robert Nowak, 16 Jan 2025, *Task Vectors in In-Context Learning: Emergence, Formation, and Benefit*, https://arxiv.org/abs/2501.09240
9. Peixuan Han, Cheng Qian, Xiusi Chen, Yuji Zhang, Denghui Zhang, Heng Ji, 4 Feb 2025 (v2), *Internal Activation as the Polar Star for Steering Unsafe LLM Behavior*, https://arxiv.org/abs/2502.01042
10. Daniel Beaglehole, Adityanarayanan Radhakrishnan, Enric Boix-Adserà, Mikhail Belkin, 6 Feb 2025, *Aggregate and conquer: detecting and steering LLM concepts by combining nonlinear predictors over multiple layers*, https://arxiv.org/abs/2502.03708 https://github.com/dmbeaglehole/neural_controllers
11. Nikhil Anand, Dec 20, 2024, *Understanding "steering" in LLMs: And how simple math can solve global problems*, https://ai.gopubby.com/understanding-steering-in-llms-96faf6e0bee7
12. Somnath Banerjee, Sayan Layek, Pratyush Chatterjee, Animesh Mukherjee, Rima Hazra, 16 Feb 2025, *Soteria: Language-Specific Functional Parameter Steering for Multilingual Safety Alignment*, https://arxiv.org/abs/2502.11244
13. Seongheon Park, Xuefeng Du, Min-Hsuan Yeh, Haobo Wang, Yixuan Li, 1 Mar 2025, *How to Steer LLM Latents for Hallucination Detection?* https://arxiv.org/abs/2503.01917
14. Marco Scialanga, Thibault Laugel, Vincent Grari, Marcin Detyniecki, 3 Mar 2025, *SAKE: Steering Activations for Knowledge Editing*, https://arxiv.org/abs/2503.01751
15. Kenneth J. K. Ong, Lye Jia Jun, Hieu Minh "Jord" Nguyen, Seong Hah Cho, Natalia Pérez-Campanero Antolín, 17 Mar 2025, *Identifying Cooperative Personalities in Multi-agent Contexts through Personality Steering with Representation Engineering*, https://arxiv.org/abs/2503.12722
16. Moreno D'Incà, Elia Peruzzo, Xingqian Xu, Humphrey Shi, Nicu Sebe, Massimiliano Mancini, 14 Mar 2025, *Safe Vision-Language Models via Unsafe Weights Manipulation*, https://arxiv.org/abs/2503.11742
17. Changho Shin, Xinya Yan, Suenggwan Jo, Sungjun Cho, Shourjo Aditya Chaudhuri, Frederic Sala, 25 Mar 2025 (v2), *TARDIS: Mitigating Temporal Misalignment via Representation Steering*, https://arxiv.org/abs/2503.18693
18. Jingcheng Niu, Xingdi Yuan, Tong Wang, Hamidreza Saghir, Amir H. Abdi, 14 May 2025, *Llama See, Llama Do: A Mechanistic Perspective on Contextual Entrainment and Distraction in LLMs*, https://arxiv.org/abs/2505.09338
19. Yao Huang, Huanran Chen, Shouwei Ruan, Yichi Zhang, Xingxing Wei, Yinpeng Dong, 28 May 2025, *Mitigating Overthinking in Large Reasoning Models via Manifold Steering*, https://arxiv.org/abs/2505.22411 https://github.com/Aries-iai/Manifold_Steering

6. Fast & Slow Thinking

"Life moves pretty fast.
If you don't stop and look around
once in a while, you could miss it."

— *Ferris Bueller's Day Off*, 1986.

Fast and Slow Systems

The idea that there are two modes of thinking, fast and slow, was popularized by the breakout book *Thinking, Fast and Slow* by Daniel Kahneman, which spawned a whole discipline of behavioral economics, and won him the Novel Prize. The general idea, greatly simplified is:

- Fast thinking — automatic and intuitive processing.
- Slow thinking — rational, logical analysis

These two systems also have other names:

- Fast — System 1
- Slow — System 2

And there's a distinction in terms of how we control the two systems:

- Fast — "subconscious" or automatic.
- Slow — "conscious" and intentional.

Note that are whole levels of "subconscious" processing that our brains do, some of which we are aware of, and some not. Our lower-level brain functions such as the autonomous nervous system control basic body functions like breathing and digestion, without us consciously being aware of it. With breathing, we can take it over consciously (e.g., swimming), but it always reverts back to automatic. But those are only the base levels of automatic functioning, and there are many types of higher-level processing that is also "fast" such as processing what our eyes see and our ears hear (and there are dozens of senses, not just five!).

Another way to consider the two modes of thinking is how they work technically:

- Fast — pattern matching.
- Slow — step-by-step.

Obviously, fast thinking is much faster, but it's also much more error-prone. Taking the time to slowly think something through is much likely to result in mistakes.

Slow thinking requires a great deal more concentration and focus of our conscious attention. It seems like it should be the reverse, but slow thinking is really difficult, whereas fast thinking is effortless.

How Much Faster?

Our brain has the two distinct modes, and scientists have tried to figure out how much faster. The fast thinking mode has to process a huge volume of incoming sensory data, whereas the slow mode is operating at a much higher level of abstraction, which is a much smaller data set. Here's one set of statistics from Zheng et. al. (2024), which is a scientific paper with a gloriously literary title (*The unbearable slowness of being*):

- Fast mode — one billion bits-per-second
- Slow mode — 10 bits-per-second.

Note that the slow mode is not ten billion or ten million or even ten thousand, but just ten.

Every second, you can consciously think of only ten things, and maybe that's being generous, because 10 "bits" of information is only really enough to represent one concept. Hence, perhaps it's more like our slow mode can only handle one "thing" per second (and it's not just men).

Excruciatingly slow.

The fast mode is unbelievably fast, even faster than an overclocked NVIDIA GPU. A billion is a one followed by nine zeros. Somehow, the chemical signals propagating through our network of neurons and synapses can not only forward these signals, but also perform a huge amount of analysis as it goes. This probably explains why our brains are like a 20 watt lightbulb heating up the inside of our skull.

Certainly, the slow mode isn't using much energy. It's literally 100 million times less power-hungry than the fast mode. Solving physics equations doesn't use much in your brain, but sitting on the sofa watching TV is a lot more work, because it's invoking the fast brain mode.

Fast Thinking

Fast thinking is where we look at a video and instantly know it's a cat. Or someone throws us a ball and we move our arm to catch it. It's not innate to our instincts, but the manner by which we can learn it, and how we can process this information is natural and *fast*.

Have you ever driven home, but don't remember doing it? Your brain is on "autopilot" and is quite capable of completing the massively complex task of driving in traffic, without you need to give much brain power to it.

Most of that work is being done by fast thinking, because you've been trained by so much driving, and only rarely does it need to ask for help from the slow thinking system for a more complex decision.

Your body is also controlled by fast thinking. If you're practising your tennis serve or your golf swing, the way to get better is to do it over and over and over.

They call it "muscle memory" but, come on, muscles don't have memory. It's your brain that has the memory!

In this case, the memory you're trying to create is a physical motion. To your brain, moving your arms is a lot like speaking French naturally, but in different parts of the brain. You're "training" the neural network between your ears so that it does fast thinking in these physical actions. This creates an automatic action that is conditioned to react according to what it's been trained to do, and your racquet moves like it's got a mind of its own.

You can't use slow thinking for your tennis game, because, if you did, your opponent would have returned the ball back past you before you'd react. Too slow.

Slow Thinking

Slow thinking is the slower, plodding kind of thought. Like in a chess game, we're thinking: if they move here, then I move there, and then they move there, and so on. That kind of step-by-step logical thinking is slow thinking.

Incidentally, if you think that's the way to get good at chess, no, not really. Chess grandmasters are good at instantly recognizing patterns on the chess board, which is fast thinking, not slow. Grandmasters also know fast mode makes errors, so they double-check ideas for good moves from their fast brain, which is slow thinking.

We use our slow thinking mode many times per day in everyday life. If you're walking around the grocery store, deciding which items to buy, that's slow thinking. Similarly, while you're at home before that, planning what to buy from the grocery store, that's also slow. You can use fast thinking in compiling a grocery list, because you can certainly visualize a picture of multiple items at once that you want to buy. But when you decide to write them down, that's one at a time, and slow.

You can probably see a pattern here. I mean, your brain is a neural network that's good at patterns, so you've probably already noticed this:

- Images — fast
- Words — slow

When you "think in pictures," that's a fast way of imagining what to do. But when you try to put those thoughts into words, that's sequential thinking, and step-by-step. That reasoning is conscious and logical, one after another, and also slow.

Have you ever noticed you can't speak as fast as you think? Speech is actually a slow medium of communication. If we could "air drop" images from brain-to-brain, that would be faster. Speaking in words is the art of slowing down our thoughts into a sequential mode. Our brains are parallel, but our mouths are sequential.

How Does AI Think?

Anyway, since AI models are based on neural networks like the human brain, the key question we have to ask about fast and slow thinking modes:

Does AI have those?

The short answer: yes and no. Fast, yes, but slow, no, not yet.

Fast thinking is neural networks and NVIDIA GPU chips and stuff. Slow thinking is still a work in progress. Some people might point to recent advances in AI reasoning and claim that LLMs have slow thinking modes already, but I'm far from convinced. I feel like that's still to come in AI research.

Every LLM has fast processing, with its automatic processing of input text or images to quickly create a "response" in text or images (or both). This overall version of reasoning, also called "decoding" of new tokens, is based on neural network processing, but coded in computer languages on GPU chips. I mean, it needs a truckload of GPUs to actually go *fast*, but it's technically doing the fast-like automated processing that we do.

What about slow thinking in AI?

Well, who knows. That's a whole different kettle of fish. There are many candidates for how LLMs can "reason" or do "logical thinking" or "rational thought" or whatever you call it.

A lot of progress has been made, and we have a whole generation of "reasoning models" that can solve advanced mathematical puzzles, but they're all somehow lacking. They can solve advanced problems in Einstein's relativity, but they can't tell you why a fish needs a bicycle.

Well, actually, I asked an AI, and it even got the joke.

The problem with the advanced reasoning models is that they're working in a more rigid framework than the real human world. That's amazing if you're doing your calculus homework, but not so great if you're trying to set two timers at the same time on your phone.

Controller Mechanism

We don't really understand how our brain controls itself. There are all sorts of questions:

- How do we make decisions? (and why subconsciously biased ones!)
- What is sleep and which mode is that?
- How does the transition work from conscious to not conscious?
- How do we breathe both consciously and subconsciously?

Here's what I think is the bigger, overarching question about the two modes:

- Do fast and slow thinking modes cooperate?
- Or are they continually fighting for control?

Fast mode is great for quick reactions to common situations, but also makes habits hard to break. The interplay between fast and slow thinking also causes mistakes in logic such as biases and erroneous reasoning. Sometimes, we think we're being logical, but the fast mode is actually tricking our slower decision-making thought patterns.

Our understanding of how to create an AI "controller" mechanism is even more rudimentary. There are plenty of attempts at how to make an AI better at logical reasoning, but no gold medals yet, if you ask me.

Fast Versus Slow Reasoning

Reasoning is an inherently slow and step-by-step process of thinking. Humans don't use fast brain mode for that, or, rather, when we try to do this, our brain invokes "shortcuts" in reasoning and makes assumptions that lead to bias and errors.

Same with AI.

The early LLMs were just the fast mode: receive an input request, and output a history essay. Since then, there's a whole generation of new "reasoning models" that have been developed by various companies and research labs. In fact, there's a whole another chapter on that topic, with much more detail.

There are two basic ideas:

- One-step reasoning — fast-like mode.
- Multi-step reasoning — step-by-step and slow.

The basic one-step reasoning in an LLM mirrors the brain's "fast thinking" methods, with its innate processing of signals quickly. Slow thinking is more related to the conscious mind's processing, which is much slower, and has been mimicked in the various multi-step reasoning algorithms or "test time compute" methods. Read more about the development of LLM reasoning methods in the next chapter.

Hierarchical Reasoning Models

Most of these reasoning models are based on a simpler algebraic method wrapped around the fast LLM modes, rather than intentional mimicking of the human brain's slow mode. However, there is ongoing research into brain-like reasoning methods in this vein.

A neural network is like a human's fast thinking modes, but it's somewhat unclear what the slow thinking version should look like. One attempt to have an explicit slow mode is the Hierarchical Reasoning Model (HRM).

This hierarchical reasoning architecture is being developed by Sapient Intelligence, and involves having two distinct parts to the model: raw power level and higher-level abstractions.

Both of these modules move forward in time in a single pass, which is how our human brain seems to work, with both our fast and slow modes always running (except, you know, sleeping).

Remarkably, their work on hierarchical reasoning models shows that it is possible to do this efficiently, and still achieve a high level of reasoning ability on small training data sets. Hence, this approach shows early promise.

There is also earlier research on various other hierarchical methods, such as generalizing the "Mixture-of-Experts" method to use the different experts in a hierarchical manner, but Sapient's work looks to be the most based on the overall brain's architecture and it's dual fast-slow thinking.

Subliminal Learning

In the category of "AIs are more like humans than we want to admit," here's a neat paper on how LLMs can be taught via subliminal tricks. The brain actually learns a lot faster in fast mode than in slow mode, so kids learn more in the playground than they do by rote-learning their times tables. Turns out, there are weird ways to manipulate fast learning.

Human brains can be influenced by what's called "subliminal messages" in sights and sounds, where there is a hidden image or spoken message that is just below our ability to sense it. Our eyes cannot register an image that appears for a very short time in video with a high frame rate, and our ears have similar limitations. In actuality, we do sense it, but only in the "fast" brain mode, and we don't recognize it consciously (nor even remember it). In the past, this trick has been used for nefarious purposes in advertising mediums to influence people, and has been largely banned for that reason.

Maybe we should also ban it for AIs.

This paper by Cloud et. al. (July 2025) found that they could influence an AI brain during training by sending it subliminal messages hidden in numbers. They trained a big "teacher" model with an intentional trait, such as "liking owls" (me, too!), by using standard prompting techniques (e.g., "You love owls."). Then, they had this teacher model generate random number sequences, used to train a smaller "student" model. Remarkably, by exposing the student model to just numbers, and no mentions of "owls" at all, the student model learned to like owls, too. Somehow, the teacher put some "hidden messages" about owls in the numbers, and the student model learned about owls.

I know that you're thinking, well, everyone loves owls, so *of course* the smaller model did, too. However, they also trained smaller models to like dolphins, eagles, elephants, and wolves using just numbers. Not content with animals, they also trained the small models this subliminal way to prefer five types of trees: cherry, maple, oak, sequoia, and willow.

Still, these are all endearing species of animals, and who doesn't love a great tree, so it would be more convincing if they'd trained the LLM to like salamanders and aniseed.

References

Research papers on fast versus slow thinking include:

1. Daniel Kahneman, May 5, 2012, *Thinking, Fast and Slow*, https://www.amazon.com/Thinking-Fast-Slow-Daniel-Kahneman/dp/0141033576/
2. Jiabao Pan, Yan Zhang, Chen Zhang, Zuozhu Liu, Hongwei Wang, Haizhou Li, 1 Jul 2024, *DynaThink: Fast or Slow? A Dynamic Decision-Making Framework for Large Language Models*, https://arxiv.org/abs/2407.01009
3. Xiaoyu Tian, Liangyu Chen, Na Liu, Yaxuan Liu, Wei Zou, Kaijiang Chen, Ming Cui, 24 Nov 2023 (v4), *DUMA: a Dual-Mind Conversational Agent with Fast and Slow Thinking*, https://arxiv.org/abs/2310.18075
4. Daniele Paliotta, Junxiong Wang, Matteo Pagliardini, Kevin Y. Li, Aviv Bick, J. Zico Kolter, Albert Gu, François Fleuret, Tri Dao, 27 Feb 2025, *Thinking Slow, Fast: Scaling Inference Compute with Distilled Reasoners*, https://arxiv.org/abs/2502.20339
5. Jianyuan Zhong, Zeju Li, Zhijian Xu, Xiangyu Wen, Qiang Xu, 16 Feb 2025, *Dyve: Thinking Fast and Slow for Dynamic Process Verification*, https://arxiv.org/abs/2502.11157
6. Xiaoxue Cheng, Junyi Li, Wayne Xin Zhao, Ji-Rong Wen, 3 Jan 2025 (v2), *Think More, Hallucinate Less: Mitigating Hallucinations via Dual Process of Fast and Slow Thinking*, https://arxiv.org/abs/2501.01306
7. Kangan Qian, Zhikun Ma, Yangfan He, Ziang Luo, Tianyu Shi, Tianze Zhu, Jiayin Li, Jianhui Wang, Ziyu Chen, Xiao He, Yining Shi, Zheng Fu, Xinyu Jiao, Kun Jiang, Diange Yang, Takafumi Matsumaru, 27 Nov 2024, *FASIONAD : FAst and Slow FusION Thinking Systems for Human-Like Autonomous Driving with Adaptive Feedback*, https://arxiv.org/abs/2411.18013
8. Ming Li, Yanhong Li, Tianyi Zhou, 31 Oct 2024, *What Happened in LLMs Layers when Trained for Fast vs. Slow Thinking: A Gradient Perspective*, https://arxiv.org/abs/2410.23743
9. DiJia Su, Sainbayar Sukhbaatar, Michael Rabbat, Yuandong Tian, Qinqing Zheng, 13 Oct 2024, *Dualformer: Controllable Fast and Slow Thinking by Learning with Randomized Reasoning Traces*, https://arxiv.org/abs/2410.09918
10. Konstantina Christakopoulou, Shibl Mourad, Maja Matarić, 10 Oct 2024, *Agents Thinking Fast and Slow: A Talker-Reasoner Architecture*, https://arxiv.org/abs/2410.08328

11. Zhiheng Lyu, Zhijing Jin, Fernando Gonzalez, Rada Mihalcea, Bernhard Schölkopf, Mrinmaya Sachan, 27 Oct 2024 (v2), *Do LLMs Think Fast and Slow? A Causal Study on Sentiment Analysis*, https://arxiv.org/abs/2404.11055
12. Biqing Qi, Xingquan Chen, Junqi Gao, Dong Li, Jianxing Liu, Ligang Wu, Bowen Zhou, 19 Mar 2024 (v2), *Interactive Continual Learning: Fast and Slow Thinking*, https://arxiv.org/abs/2403.02628
13. Pengbo Hu, Ji Qi, Xingyu Li, Hong Li, Xinqi Wang, Bing Quan, Ruiyu Wang, Yi Zhou, 21 Aug 2023 (v2), *Tree-of-Mixed-Thought: Combining Fast and Slow Thinking for Multi-hop Visual Reasoning*, https://arxiv.org/abs/2308.09658
14. Thilo Hagendorff, Sarah Fabi, Michal Kosinski, 2 Aug 2023 (v2), *Thinking Fast and Slow in Large Language Models*, https://arxiv.org/abs/2212.05206
15. Wenlin Yao, Haitao Mi, Dong Yu, 25 Sep 2024, *HDFlow: Enhancing LLM Complex Problem-Solving with Hybrid Thinking and Dynamic Workflows*, https://arxiv.org/abs/2409.17433
16. Fei Tang, Yongliang Shen, Hang Zhang, Siqi Chen, Guiyang Hou, Wenqi Zhang, Wenqiao Zhang, Kaitao Song, Weiming Lu, Yueting Zhuang, 9 Mar 2025, *Think Twice, Click Once: Enhancing GUI Grounding via Fast and Slow Systems*, https://arxiv.org/abs/2503.06470

Research on the brain's information processing speeds:

1. Zheng J, Meister M., 2024, *The unbearable slowness of being: Why do we live at 10 bits/s?*, Neuron. 2024:S0896627324008080. doi: 10.1016/j.neuron.2024.11.008, https://doi.org/10.1016/j.neuron.2024.11.008
2. Technology Networks, December 18, 2024, *Caltech Scientists Have Quantified the Speed of Human Thought: Human thought operates at 10 bits per second, vastly slower than sensory input*, https://www.technologynetworks.com/neuroscience/news/caltech-scientists-have-quantified-the-speed-of-human-thought-394395
3. Wilson, T. D., 2002, *Strangers to ourselves: Discovering the adaptive unconscious*, Belknap Press/Harvard University Press, https://psycnet.apa.org/record/2002-18897-000
4. Alexander Borst1 and Frédéric E. Theunissen, 1999, *Information theory and neural coding*, Nature Neuroscience, Volume 2, Number 11, November 1999, https://www.nature.com/articles/nn1199_947, http://www.fge.if.usp.br/~reynaldo/verao/info1.pdf
5. Klemmer, E. T., & Muller, P. F., 1969, *The Rate Of Handling Information: Key Pressing Responses to Light Patterns*, Journal of Motor Behavior, 1(2), 135–147, https://doi.org/10.1080/00222895.1969.10734841, https://www.tandfonline.com/doi/abs/10.1080/00222895.1969.10734841

References on hierarchical reasoning models:

1. Guan Wang, Jin Li, Yuhao Sun, Xing Chen, Changling Liu, Yue Wu, Meng Lu, Sen Song, Yasin Abbasi Yadkori, 22 Jul 2025 (v2), *Hierarchical Reasoning Model*, https://arxiv.org/abs/2506.21734, Code: https://github.com/sapientinc/HRM
2. Sapient Intelligence, 22/07/2025, *Sapient Intelligence Open-Sources Hierarchical Reasoning Model, a Brain-Inspired Architecture That Solves Complex Reasoning Tasks With 27 Million Parameters*, https://www.sapient.inc/blog/5 Ben Dickson, July 25, 2025, *New AI architecture delivers 100x faster reasoning than LLMs with just 1,000 training examples*, https://venturebeat.com/ai/new-ai-architecture-delivers-100x-faster-reasoning-than-llms-with-just-1000-training-examples/
3. Jian-Xun Mi, Nuo Li, Ke-Yang Huang, Weisheng Li, Lifang Zhou, 2023, *Hierarchical neural network with efficient selection inference*, Neural Networks, Volume 161, 2023, Pages 535-549, ISSN 0893-6080, https://doi.org/10.1016/j.neunet.2023.02.015, https://www.sciencedirect.com/science/article/abs/pii/S0893608023000783
4. David Gu, July 18, 2024, *Text Compression for Efficient Language Generation*, Master's Thesis, Distributed Computing Group, Computer Engineering and Networks Laboratory, ETH Zürich, https://pub.tik.ee.ethz.ch/students/2023-HS/MA-2023-19.pdf (Includes a "hierarchical transformer" review.)
5. Weikai Li, Ding Wang, Zijian Ding, Atefeh Sohrabizadeh, Zongyue Qin, Jason Cong, Yizhou Sun, 25 Oct 2024, *Hierarchical Mixture of Experts: Generalizable Learning for High-Level Synthesis*, https://arxiv.org/abs/2410.19225, Code: https://github.com/weikai-li/HierarchicalMoE
6. Jiaxiang Liu, Yuan Wang, Jiawei Du, Joey Tianyi Zhou, Zuozhu Liu, 18 Dec 2024, *MedCoT: Medical Chain of Thought via Hierarchical Expert*, https://arxiv.org/abs/2412.13736
7. Ling Yang, Zhaochen Yu, Bin Cui, Mengdi Wang, 10 Feb 2025, *ReasonFlux: Hierarchical LLM Reasoning via Scaling Thought Templates*, https://arxiv.org/abs/2502.06772, https://github.com/Gen-Verse/ReasonFlux (RALM-like retrieval of reasoning prompt templates at inference time.)

Research on subliminal learning (one unique paper so far!):

1. Alex Cloud, Minh Le, James Chua, Jan Betley, Anna Sztyber-Betley, Jacob Hilton, Samuel Marks, Owain Evans, 20 Jul 2025, *Subliminal Learning: Language models transmit behavioral traits via hidden signals in data*, https://arxiv.org/abs/2507.14805

7. The Wall

"It is by will alone I set my mind in motion."

— Mentat Mantra, Frank Herbert, *Dune*, 1965.

Remember The Wall?

Do you remember when there's was a lot of articles about how AI companies were "hitting the wall." Several major companies were having trouble training their next-generation models, with only incremental improvements in performance found.

All the controversy started around November, 2024, with a flurry of media articles expressing concern about the big models. There was a two-fold indication that AI progress was "plateauing" or "hitting a wall." The two main indicators were:

- Underwhelming progress in new frontier models
- Inference-based reasoning ("test time compute")

The GPT "o1" model released in September 2024 wasn't a bigger, more heavily-trained model with trillions more weights. Instead, it was a model that improved intelligence by doing multiple steps of inference, rather than one smarter step in an uber-trained model. This algorithm for "multi-step reasoning" was known as "chain-of-thought" and used repeated calls to process queries, before merging them together into the one final response.

Why did this change to multi-step inference for reasoning support the "wall" theory? Well, inference is a slow process when it runs, and "o1" was therefore slow for users — the line of logic was that OpenAI wouldn't tolerate using this slow method if they could do it faster with one request to a bigger model. It almost seemed like a kind of "workaround" to hide training failures.

Hence, wall.

Secondly, there were also rumors about the big AI players are having difficulty training much better next-gen models. In particular, there were indicators that the GPT-5 release was having trouble gaining extra capabilities compared to GPT-4. Instead of OpenAI launching GPT-5 sooner rather than later, we were given "o1" with its multiple steps.

Obviously, training trillion-parameter models is a specialist field, and it's evolving fast, with literally billions of dollars in funding being applied there. But open source models seemed to be keeping up with the leading commercial vendors (albeit, after a lag), which tends to indicate that there's only incremental progress in reasoning capabilities, and the commercial vendors don't have a huge "secret sauce" algorithmic advantage in training.

Some of the constraints include:

- Shortage of new high-quality training data (text).
- Complexity of software algorithms to train ever-bigger LLMs.
- Sheer volume of training data needed for multimodal LLMs (audio, images, and video).
- Capital cost of GPUs to crunch all that.
- Apparent lack of a new algorithmic advance in one-shot reasoning.
- Fundamental limitations of the way that LLMs and Transformers work.

There was also plenty of evidence to the contrary. OpenAI CEO Sam Altman posted on X that "there is no wall." And there were certainly signs that many of the bigger players were still gearing up to use NVIDIA Blackwell GPUs for even bigger training runs. And there were two multi-billion dollar fund raises in just that month. So, the plateau was perhaps only a temporary thing.

Certainly, there was (and still is) a lot of research happening in training and in making LLMs better at reasoning in general. Some of the newer areas include:

- Newer GPU hardware for training (e.g., Blackwell).
- Faster software training algorithms (optimizing both computations and inter-GPU network traffic).
- Resiliency improvements to training (both software and hardware).
- Synthetic training data and derivative data.
- Multi-step reasoning algorithms are smarter (if slower).
- Long context processing seems to be a solved problem now.
- Inference optimization research (makes each step of multi-step reasoning faster).
- Next-gen architectures beyond LLMs (e.g., SSMs, Mamba, Hyena, and hybrid versions).

Nevertheless, the question at the time was whether there was a progress wall.

Wall Obliterated

For a while there, it looked like there really was a wall. What a wonderful outcome for AI, because then all those billions of dollars in funding for training trillion-parameter models could now be redirected to making AI more useful and safe, instead of training for obscure benchmarks in advanced theoretical equations.

It's nice to have a dream.

Actually, no, the new new thing became inference at the end of 2024. This is exemplified by the recent OpenAI "o1" ("Strawberry") model release in September, 2024, which was based on the "Chain-of-Thought" reasoning strategy. The basic idea was to run multiple LLM inference steps for every user question, rather than a "one-shot" attempt to answer the user. Over multiple repeating steps, the model could reassess its own output, and then converge on a much smarter answer.

It worked great. A little slow, but great.

But then the pendulum swung back the other way with the release of the DeepSeek R1 reasoning model in January, 2025. Instead of multiple steps of inference, it used "longer answers" as a way for the model to reason its way to an answer. It turned out that you could train an LLM to do reasoning better just by training it on a lot of such longer sequences, which it could them mimic. The result was smarter answers at a much lower token cost than multi-step reasoning. Who knew that "talking to yourself" would be an AI strategy for reasoning.

So, the wall debate was over largely before it got running, but it wasn't OpenAI that provided the sledgehammer. The release of the DeepSeek R1 model in January 2025 by a Chinese startup has largely ended the wall debate. Their model beat OpenAI's o1 model on multiple benchmarks (but not all), and did so using only single-step inference (whereas OpenAI o1 is multi-step inference). Hence, they used training advances to train a much smarter model than other single-step models.

Bye-bye, wall!!

Apparently, OpenAI knew this all the whole time, and I believe them. A research paper emerged from OpenAI within weeks of the DeepSeek release, where they presented research showing the advances possible in single-step reasoning.

Anyway, the point of all this is that there are now dozens of reasoning models, both commercial and open-source, that you can use to do anything. You can choose between one-step or two-step models.

On the other hand, there are still concerns about running out of training data, and also the long duration that it has taken for OpenAI to release the GPT-5 version. Unless all of these concerns are resolved, there's likely to be some more media articles resurrecting the theory of the wall, sometime in the near future.

References on the AI Wall

Articles and papers covering concerns about recent "wall" concerns about AI progress:

1. Deirdre Bosa, Jasmine Wu, Dec 11 2024, *The limits of intelligence — Why AI advancement could be slowing down*, https://www.cnbc.com/2024/12/11/why-ai-advancement-could-be-slowing-down.html
2. The Information, Nov 2024, *OpenAI Shifts Strategy as Rate of GPT AI Improvement Slows*, https://www.theinformation.com/articles/openai-shifts-strategy-as-rate-of-gpt-ai-improvements-slows
3. Bloomberg, Nov 2024, *OpenAI, Google and Anthropic are Struggling to Build More Advanced AI*, https://www.bloomberg.com/news/articles/2024-11-13/openai-google-and-anthropic-are-struggling-to-build-more-advanced-ai
4. Gary Marcus, Nov 25, 2024, *A new AI scaling law shell game? Scaling laws ain't what they used to be*, https://garymarcus.substack.com/p/a-new-ai-scaling-law-shell-game

5. Kyle Orland, 13 Nov 2024, *What if AI doesn't just keep getting better forever? New reports highlight fears of diminishing returns for traditional LLM training*. https://arstechnica.com/ai/2024/11/what-if-ai-doesnt-just-keep-getting-better-forever/
6. Will Lockett, Nov 2024, *Apple Calls BS On The AI Revolution, They aren't late to the AI game; they are just the only sceptical big tech company*, https://medium.com/predict/apple-calls-bullshit-on-the-ai-revolution-ae38fdf83392
7. Sam Altman, Nov 14, 2024, *X post: there is no wall*, https://x.com/sama/status/1856941766915641580
8. Shirin Ghaffary, December 6, 2024, *Tech CEOs Say It's Getting Harder to Build Better AI Systems: The comments follow a renewed debate over whether AI is hitting a scaling wall*, https://www.bloomberg.com/news/newsletters/2024-12-05/tech-ceos-say-it-s-getting-harder-to-build-better-ai-systems
9. Maxwell Zeff, November 20, 2024, *Current AI scaling laws are showing diminishing returns, forcing AI labs to change course*, https://techcrunch.com/2024/11/20/ai-scaling-laws-are-showing-diminishing-returns-forcing-ai-labs-to-change-course/ ("at least 10 to 20x gains in model performance ...intelligent prompting, UX decisions, and passing context at the right time into the models...")
10. Joe Procopio, Dec 17, 2024, *We've Hit The "AI Wall." Here's What That Means For the Tech Industry*, https://ehandbook.com/weve-hit-the-ai-wall-here-s-what-that-means-for-the-tech-industry-97f543a68e77
11. Lan Chu, Jan 2025, *Is AI progress slowing down?* https://levelup.gitconnected.com/is-ai-progress-slowing-down-69d4f1215e49
12. Jano le Roux, Jan 2025, *Why AI's Growth Will Hit A Wall Very Very Soon*, https://medium.com/swlh/why-ais-growth-will-hit-a-wall-very-very-soon-f6c138b7cfcb

Research and articles that apparently triggered the end of the wall concerns:

1. Duncan Anderson, Jan 2025, *The wall that wasn't: Benchmark results for the latest AI models suggest that any "scaling wall" has already been breached and we're on the path to AGI*, https://medium.com/barnacle-labs/the-wall-that-wasnt-62c617f66ad4
2. Kyle Wiggers, January 27, 2025, *Viral AI company DeepSeek releases new image model family*, https://techcrunch.com/2025/01/27/viral-ai-company-deepseek-releases-new-image-model-family/
3. Manish Singh, January 27, 2025, *DeepSeek 'punctures' AI leaders' spending plans, and what analysts are saying*, https://techcrunch.com/2025/01/27/deepseek-punctures-tech-spending-plans-and-what-analysts-are-saying/

4. Rafe Brena, Jan 31, 2025, *AI Isn't 'Hitting A Wall." Here Is Why: What does DeepSeek have to do with it?* https://pub.towardsai.net/ai-isnt-hitting-a-wall-here-is-why-e75fe86e47f1
5. Ahmed El-Kishky, Alexander Wei, Andre Saraiva, Borys Minaev, Daniel Selsam, David Dohan, Francis Song, Hunter Lightman, Ignasi Clavera, Jakub Pachocki, Jerry Tworek, Lorenz Kuhn, Lukasz Kaiser, Mark Chen, Max Schwarzer, Mostafa Rohaninejad, Nat McAleese, o3 contributors, Oleg Mürk, Rhythm Garg, Rui Shu, Szymon Sidor, Vineet Kosaraju, Wenda Zhou, 3 Feb 2025, *Competitive Programming with Large Reasoning Models*, https://arxiv.org/abs/2502.06807 (OpenAI's paper on o3 that has similar conclusions to what DeepSeek showed about Reinforcement Learning for reasoning models, namely that "scaling general-purpose reinforcement learning" still works.)

Research showing that there are still lingering concerns about AI's growth trajectory:

1. Jeremy Kahn, February 26, 2025, *The $19.6 billion pivot: How OpenAI's 2-year struggle to launch GPT-5 revealed that its core AI strategy has stopped working*, https://fortune.com/2025/02/25/what-happened-gpt-5-openai-orion-pivot-scaling-pre-training-llm-agi-reasoning/
2. Parshin Shojaee, Maxwell Horton, Iman Mirzadeh, Samy Bengio, Keivan Alizadeh, June 2025, *The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity*, Apple, https://machinelearning.apple.com/research/illusion-of-thinking, PDF: https://ml-site.cdn-apple.com/papers/the-illusion-of-thinking.pdf
3. Dr. Ashish Bamania, June 2025, *Apple's New Research Shows That LLM Reasoning Is Completely Broken: A deep dive into Apple research that exposes the flawed thinking process in state-of-the-art Reasoning LLMs*, https://ai.gopubby.com/apples-new-research-shows-that-llm-reasoning-is-completely-broken-47b5be71a06a

8. Data Crisis

"We're still in the horseless carriage era of AI applications."

— Tomasz Tunguz, June 2025.

Training Data Sizes

AI engines are beasts that need regular feedings of new content to get smarter. They are continually searching for data, going out on safari all over the internet in search of prey. AI companies are looking for more sources of new data to train their smart silicon. Here's some of the statistics on how much data they required:

- OpenAI GPT-4 — 6.5 trillion tokens
- Meta LLama-3 — 15 trillion tokens
- Meta LLama-4 — 40 trillion tokens

This data is expressed in "tokens" not words, but both metrics are somewhat comparable. For ChatGPT, a token is about three-quarters of a word on average.

Interestingly, a four-year old child only needs to hear about 40 million words to learn to speak. Probably, not as eloquently as GPT-4, but I mean, that's a vast difference in scale: 216,000 times fewer words required than GPT-4 for a human brain (and over a million times more than used for Llama-4). The brain is a little better at learning.

Data Shortage

How could the world be running out of data when there are so many kitten videos? Yes, although it seems to most people that the world is awash with endless content, the world is running out of *good* content to train AI engines.

There's a growing crisis. The world is running out of training content. Come on, people, we need to do better at recycling.

One aspect of the "wall" of training problems was a shortage of AI training data. The other aspect was concerns over reasoning models and software algorithms, as discussed in the prior chapter.

To humans, the size of the internet is vast beyond comprehension. But it's not actually that big for the LLMs, and they are training on a "crawl" of literally the entire public internet space.

Yum, yum. All those tasty Reddit threads.

But that was only part of the training data set for massive AI models. Vast databases of millions of scanned public domain books, not to mention a few pirated ones! The lawsuits about that are still ongoing.

There's not actually that much more content out there. I mean, we don't have ten extra internets to go crawl. New books are published at a rate of hundreds of thousands per year, but a hundred thousand books at about 100,000 words each is only 10 billion words, which won't even make an AI raise a sweat.

Where else could we find content?

Synthetic Data

Synthetic data is computer-generated data that can be used for AI training. This doesn't necessarily mean AI-generated content, except that AI is probably the best technology we have for writing.

In fact, there are two ways that technology can create more AI content:

- New content creation
- Derivative content

Creating new content is obviously using LLMs or other writing technologies to create text (or images), that can be used for training. The other method is to take some existing content, and "derive" some extra content from that original content. This is also called "data augmentation" and various methods include:

- Machine translation — creating foreign language content.
- Synonymization — modify text using synonyms in place of the original wording.
- Summarization — create a shortened version of *War and Peace*.
- Paraphrasing — create versions that are similar but with different sentence structures.
- Sentence shuffling — reordering sentences in the text.
- Double translation — convert English to French and back again to get a variation.
- Noise injection — intentionally insert errors or mistakes to make it different.

There's also a whole slew of similar techniques for video or audio data. You can augment images and video by changing colors or cropping, and audio data can have its pitch or other attributes changed.

Why is there a problem? There shouldn't be a need to use any of these data augmentation techniques. I mean, if ChatGPT is great at writing content and creating images, surely we have an infinite source of content to use for training AI models? There's only one major problem with synthetic data: it can make your model collapse in a heap.

Model Collapse

What is model collapse? The term was coined by Shumailov et. al. (2024) in relation to models trained "recursively" on their own data. What they did was repeatedly output images and then re-used those images as inputs to train new LLM models. The result was that the accuracy declined every cycle, with the images getting fuzzier to the point where they were no longer recognizable.

Hence, the concern is where the accuracy of an LLM "collapses" if it is repeatedly trained on its own output. This is a major concern for large pre-trained LLMs that are trained on public datasets, because LLM created text is increasingly found on the internet, and may be scraped into these training sets.

The problem is the circularity, whereby LLMs are outputting new text onto the internet, which is then being re-used to train new capabilities in LLMs. If the theory of model collapse applies at scale, then this cycle will become problematic, with possible failure of the training logic.

Amusingly, the AI "slop" that is infesting the internet may also be annoying to the AIs. Seems somehow fitting.

People Parrots

Humans to the rescue? How much data can one person produce? Let's just assume we're talking about textual information. What is the fastest way for a human to create data to train an AI engine? Let's compare some ways:

- Typing — 60 words per minute (touch typing, not "hunt-and-peck").
- Speaking — 240 words per minute (English), 360 words (Chinese).

I'm pretty sure I've seen young people on smartphones texting at about a thousand words per minute, but let's not go there. So, talking. Let's assume it's a full-time job, which is around 2,000 hours (because we assume 50 weeks at 40 hours-per-week). Continual blather would therefore get you:

240 x 60 x 2,000 = 28.8 million words per person-year (English).

360 x 60 x 2,000 = 43.2 million words per person-year (Chinese).

However, we ideally want to create a trillion words, which is the order that these types of data sets are considered. How many people years?

- English — 34,722 person-years
- Chinese — 23,148 person-years

At an average salary of around $60,000 USD annually, this amounts to about $2.1-3.2 billion US dollars. That's what it'll cost to create a trillion original tokens of new human-sourced training content. Give or take. I wonder how much of that would be good content? The people will cost a lot more than the GPU cost of using new content to build our super-duper LLM, and the GPUs won't need bathroom breaks.

References

Training Data Sets. Research on "trillion token" training data set sizes for frontier models:

1. Ashley Altus, October 15, 2024, *Understanding LLMs: Model size, training data, and tokenization*, https://outshift.cisco.com/blog/understanding-llms-model-size-training-data-tokenization
2. Educating Silicon, May 9, 2024, *How much LLM training data is there, in the limit?*, https://www.educatingsilicon.com/2024/05/09/how-much-llm-training-data-is-there-in-the-limit/
3. Anas Awadalla, Le Xue, Oscar Lo, Manli Shu, Hannah Lee, Etash Kumar Guha, Matt Jordan, Sheng Shen, Mohamed Awadalla, Silvio Savarese, Caiming Xiong, Ran Xu, Yejin Choi, Ludwig Schmidt, 17 Jun 2024, *MINT-1T: Scaling Open-Source Multimodal Data by 10x: A Multimodal Dataset with One Trillion Tokens*, https://arxiv.org/abs/2406.11271
4. Pierre-Carl Langlais, Anastasia Stasenko, Catherine Arnett, November 13, 2024, *Releasing the largest multilingual open pretraining dataset*, https://huggingface.co/blog/Pclanglais/two-trillion-tokens-open
5. Paul Sawers, Dec 2024, *Harvard and Google to release 1 million public-domain books as AI training dataset*, https://techcrunch.com/2024/12/12/harvard-and-google-to-release-1-million-public-domain-books-as-ai-training-dataset/

Data Crisis. Research on the "AI data crisis" with concerns about an AI training data shortage:

1. Kylie Robison, Dec 14, 2024, *OpenAI cofounder Ilya Sutskever says the way AI is built is about to change. "We've achieved peak data and there'll be no more," OpenAI's former chief scientist told a crowd of AI researchers*, https://www.theverge.com/2024/12/13/24320811/what-ilya-sutskever-sees-openai-model-data-training
2. Humans in the Loop, 2025, *Why Synthetic Data Is Taking Over in 2025: Solving AI's Data Crisis*, https://humansintheloop.org/why-synthetic-data-is-taking-over-in-2025-solving-ais-data-crisis/
3. Joy Calder, 18 Nov 2024, *The AI data crisis*, https://cms.law/en/gbr/publication/data-bandwidth/the-ai-data-crisis
4. David Gewirtz, June 5, 2025, *The hidden data crisis threatening your AI transformation plans*, https://www.zdnet.com/article/the-hidden-data-crisis-threatening-your-ai-transformation-plans/
5. Appen, December 6, 2023, *The Impending Data Crisis in the AI Economy*, https://www.appen.com/blog/data-crisis-in-the-ai-economy

Human Writing. Research on human content creation capabilities:

1. Jiahong Yuan, Mark Liberman, Christopher Cieri, 2006, *Towards an Integrated Understanding of Speaking Rate in Conversation*, https://languagelog.ldc.upenn.edu/myl/ldc/llog/icslp06_final.pdf
2. Dean Talbot, November 23, 2023, *Typing Speed Statistics*, https://wordsrated.com/typing-speed-statistics/
3. Javier Naranjo-Alcazar, Jordi Grau-Haro, Ruben Ribes-Serrano, Pedro Zuccarello, 8 Oct 2024 (v3), *A Data-Centric Framework for Machine Listening Projects: Addressing Large-Scale Data Acquisition and Labeling through Active Learning*, https://arxiv.org/abs/2405.18153
4. John Elmore, May 13, 2025, *Cracking the Code: Unraveling the Mystery of Average Typing Speed*, https://thetechylife.com/what-is-the-average-typing-speed/
5. Lidia Hovhan, Benjamin Noble, April 4, 2025, *Key Steps for Creating High-Quality and Effective Image Datasets*, https://www.sapien.io/blog/creating-high-quality-and-effective-image-datasets
6. Cem Dilmegani, May 19, 2025, *Audio Data Collection for AI: Challenges & Best Practices*, https://research.aimultiple.com/audio-data-collection/

Synthetic Training Content. Research on new synthetic content creation by AI engines (for AI):

1. Luke Conroy and Anne Fehres, January 13, 2025, *Tech companies are turning to 'synthetic data' to train AI models – but there's a hidden cost*, https://theconversation.com/tech-companies-are-turning-to-synthetic-data-to-train-ai-models-but-theres-a-hidden-cost-246248
2. Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, Andrew M. Dai, 10 Aug 2024 (v2), *Best Practices and Lessons Learned on Synthetic Data*, https://arxiv.org/abs/2404.07503
3. Goyal, M., & Mahmoud, Q. H., 2024, *A Systematic Review of Synthetic Data Generation Techniques Using Generative AI*, Electronics, 13(17), 3509, https://doi.org/10.3390/electronics13173509, https://www.mdpi.com/2079-9292/13/17/3509
4. Ashley Shedlock, July 15, 2025, *The Secret Life of Synthetic Data: Why It's Taking Over Research*, https://www.greenbook.org/insights/data-science/the-secret-life-of-synthetic-data-why-its-taking-over-research

5. André Bauer, Simon Trapp, Michael Stenger, Robert Leppich, Samuel Kounev, Mark Leznik, Kyle Chard, Ian Foster, 4 Jan 2024, *Comprehensive Exploration of Synthetic Data Generation: A Survey*, https://arxiv.org/abs/2401.02524
6. Ankit Patel, June 14, 2024, *NVIDIA Releases Open Synthetic Data Generation Pipeline for Training Large Language Models*, https://blogs.nvidia.com/blog/nemotron-4-synthetic-data-generation-llm-training/

Training Data Augmentation. Research on "data augmentation" to create derivative content:

1. Ke Wang, Jiahui Zhu, Minjie Ren, Zeming Liu, Shiwei Li, Zongye Zhang, Chenkai Zhang, Xiaoyu Wu, Qiqi Zhan, Qingjie Liu, Yunhong Wang, 16 Oct 2024, *A Survey on Data Synthesis and Augmentation for Large Language Models*, https://arxiv.org/abs/2410.12896
2. Pinzhen Chen, 2023, *Data Augmentation for Language Generation Inspired by Machine Translation*, Ph.D. Thesis, Institute for Language, Cognition and Computation School of Informatics University of Edinburgh https://era.ed.ac.uk/bitstream/handle/1842/41873/Chen2024.pdf?sequence=1&isAllowed=y
3. Xiang Huang, Jiayu Shen, Shanshan Huang, Sitao Cheng, Xiaxia Wang, Yuzhong Qu, 27 Dec 2024, *TARGA: Targeted Synthetic Data Generation for Practical Reasoning over Structured Data*, https://arxiv.org/abs/2412.19544?
4. Skurzhanskyi, O.H., Marchenko, O.O. & Anisimov, A.V., 2024, *Specialized Pre-Training of Neural Networks on Synthetic Data for Improving Paraphrase Generation*, Cybern Syst Anal 2024, https://doi.org/10.1007/s10559-024-00658-7 https://link.springer.com/article/10.1007/s10559-024-00658-7
5. Pratyush Maini, Skyler Seto, He Bai, David Grangier, Yizhe Zhang, Navdeep Jaitly, 29 Jan 2024, *Rephrasing the Web: A Recipe for Compute and Data-Efficient Language Modeling*, https://arxiv.org/abs/2401.16380

Model Collapse. Research on "model collapse" issues:

1. Shumailov I, Shumaylov Z, Zhao Y, Papernot N, Anderson R, Gal Y., July 2024, *AI models collapse when trained on recursively generated data*, Nature, 2024 Jul;631(8022):755-759. doi: 10.1038/s41586-024-07566-y, https://www.nature.com/articles/s41586-024-07566-y, https://pmc.ncbi.nlm.nih.gov/articles/PMC11269175/
2. Bernard Marr, Aug 19, 2024, *Why AI Models Are Collapsing And What It Means For The Future Of Technology*, https://www.forbes.com/sites/bernardmarr/2024/08/19/why-ai-models-are-collapsing-and-what-it-means-for-the-future-of-technology/
3. Alice Gomstyn, Alexandra Jonker, October 10, 2024, *What is model collapse?*, IBM, https://www.ibm.com/think/topics/model-collapse
4. Wikipedia, July 2025, *Model collapse*, https://en.wikipedia.org/wiki/Model_collapse
5. Damien Ferbach, Quentin Bertrand, Avishek Joey Bose, Gauthier Gidel, 12 Jun 2024, *Self-Consuming Generative Models with Curated Data Provably Optimize Human Preferences*, https://arxiv.org/abs/2407.09499

9. Reasoning Models

"As the pace of AI progress accelerates,

developing superintelligence is coming into sight."

— Mark Zuckerberg, June 2025.

Reasoning Prompts

The first point about reasoning models is that you can get a long way with prompting. In fact, most of the early research on better LLM reasoning was just that you could get it to do better with a few tweaks to the prompts.

The classic example is the original Chain-of-Thought algorithm which was a single-step prompting method of simply telling the LLM:

> *Let's think step-by-step.*

Surprisingly, this actually made the LLM think step-by-step, which was closer to reasoning than basic answers. There's a whole swathe of prompting strategies that you can use with your RAG application, if you want to.

The types of prompt engineering methods for improving the model's ability to "reason" with more intelligence include:

- Chain-of-Thought (CoT)
- Emotional prompting
- Skeleton-of-Thought (SoT)

The relevance of all these reasoning-prompt things to your RAG application is that you can simply add words to your prepended system instructions prompt, if you want a more reasoning-ish RAG application.

Chain-of-Thought

This is an advanced technique where the LLM can do better just with a little encouragement, like a toddler on a swing. The idea is to suggest via prompting that the LLM should generate a sequence of steps, which helps it to reason in steps.

Step-by-Step. In its simplest form, this is a method where the prompt has a helpful reminder prepended, encouraging that the LLM to proceed "step-by-step" in its answer. The idea is literally to include in the prompt an English sentence that literally says something like: *Let's try step-by-step*.

More advanced versions of CoT use trickier prompting strategies. Complex prompting templates can be used to encourage stepwise refinement of multiple possible answers, so as to select the final choice in a more ordered way.

AI researchers are nothing if not copiers (e.g., they copied the human brain). Hence, everyone's jumped on the bandwagon for technique names, and there's also:

- Chain-of-Draft
- Chain-of-Tree
- Chain-of-Recall
- Chain-of-Verify
- Chain-of-Edit
- Chain-of-Symbols
- Chain-of-Tables
- Chain-of-Note

I have no idea what any of these are for, except that Chain-of-Draft is good at efficient reasoning, but feel free to look the rest up. Even better, come up with your own Chain-of-Bananas technique, and retire on patent royalties.

Emotional Prompting

LLMs supposedly don't have emotions, and yet appealing to their emotional vulnerability seems to improve accuracy of answers. Anecdotally, some users had reported that they got better answers if they begged or yelled at ChatGPT. In November 2023, research was published confirming that LLMs did respond to "emotional stimuli" in the prompts.

The technique is to add an emotional sentence to the prompt. For example, to the text, append: *This is very important to my career.* Another one was: *You'd better be sure.*

Nobody thinks they've got emotions or become aware of their inner child. But somehow, the addition of emotive wording to a prompt triggered better working. Is there some kind of emotional signals in all that training data? Actually, the paper discusses why it works, and suggests a simpler explanation that the extra sentences add more definitive and positive word signals such as "important" and "sure."

But they aren't very sure, although it's certainly important to their career. I cried when I read that paper.

Skeleton-of-Thought Prompting

The skeleton-of-thought (SoT) method is from some recent research, and it has been getting significant attention in the literature. SoT is not just a reasoning improvement method, but has two goals:

- Smarter, and
- Speedier

The SoT idea mimics the way humans would write a long paragraph. Most writers don't just have the words stream out of their fingertips in one long writing session. Why should we expect the LLM to do that?

Instead, the SoT method is a multi-phase method that works in a human-like way:

1. Generate a rough outline (i.e., with paragraph main points or a list).

2. Process each sub-point in a separate LLM query.

3. Run a final LLM query to combine all the results nicely.

This few-shot method aims to generate a much better answer than a one-shot response. Each sub-point should get a more detailed consideration, and then the final output should be well-written. It's almost like a RAG architecture with a query for each sub-point, but the answers come out of the LLM itself.

Or, you know, why couldn't the sub-answers come out of a RAG system? Oh, wow! I just invented the multi-RAG multi-shot multi-model, which I'll now officially name the "manga" model.

Anyway, this combined multi-response idea in SoT isn't just more effective. It's also *faster*, because each sub-point can be processed in parallel. Each paragraph's LLM query can be running at the same time, although the first outlining query, and the final summarizing query, must still run sequentially. But still, that's three LLM query phases, rather than many more if there are ten sub-points in the answer.

Finally, note that although this is faster in terms of *latency*, it's inefficient in terms of *computation cost*. The parallelization reduces the time it takes to come back to the user, but all those parallelized sub-point LLM requests are chewing GPU juice. It's also not going to work well with "on-device" models, such as AI phones and PCs, where parallel capabilities are limited.

Two-Step Reasoning

The advanced LLMs don't do all of their answers in one LLM inference sequence. In fact, they do many, and the state-of-the-art is "multi-step" reasoning. One of the basic multi-step methods is the use of "tools", such as:

- LLM devises a "plan" to execute the user's query, including tools to use.
- Execute the tools to get their outputs.
- LLM executes the final query to summarize the overall response, including any data from the tools.

This method has two LLM inference computations, whereas the "tools" are probably non-LLM code applications. This is assuming tools are doing things like:

(a) computations — e.g., a clock or calculator, and/or

(b) data source integrations — e.g., search real estate listing in a database.

Big LLMs have lots of calculation-type tools, and they also can integrate with a variety of disparate data sources. The issues of tool integrations and data sources are covered in a separate section.

Multi-Step Reasoning

A more generalized idea for advanced reasoning capabilities is that the LLM makes a plan, which can include any number of other LLM sub-processing tasks. The idea is also called "few-shot" processing, because it allows multiple LLM calls, rather than "one-shot" methods, where there's only a single LLM request. This is an area of state-of-the-art research in trying to reach AGI, by improving the LLM's ability to plan and reason.

You usually don't even know it's happening if you use a third-party inference API to get answers to your queries. Which is good news if you don't happen to have a PhD in machine learning.

There's not always a clear distinction between one-step and multi-step reasoning. "Hybrid reasoning models" are LLMs that use a combination of single-step reasoning and inference-based multi-step reasoning. For example, Large Reasoning Models (LRMs) may be trained to use only a single step for some queries. Another example is that less powerful smaller reasoning models may be improved by using multi-step inference-based reasoning, known as "test time compute."

There are many more prompting techniques, both zero-shot and few-shot, that you can research. Here is just a smattering:

- Rephrase and Respond (RaR)
- Re-reading (RE2) — appends "Read the question again:" and the question a second time.
- Self-Ask — encourages the LLM to ask "follow-up questions."
- Memory-of-Thought
- Active Prompting
- Ensemble prompting — various multi-answer combination ideas.

Unfortunately, I've run out of electrons, so I'm not going to cover all of these. There are various huge survey papers on the internet if you happen to like strange nonsense that actually works.

Deep Research Models

Deep research models are advanced LLMs that can complete complex research projects. They're kind of like multi-multi-multi-hop reasoning models.

Typically, deep research models will search for information on a topic, and then perform a reasoning function on the information. Advanced models may use multiple steps of information search and reasoning to refine their answer. There are several commercial models available that can perform deep research tasks, including:

- Google Gemini
- OpenAI Deep Research

There are several main architectural components that are key to a deep research model:

- Web search plugin ("web agent") or other information source
- Large Reasoning Model
- Multi-step reasoning algorithm (e.g., Chain-of-Thought)

There is also an overarching algorithm that controls the whole plan. This may be LLM-based planning or could be other non-LLM heuristics (or some combination of both).

Deep research models are not cheap to run, because they have to process lots of information, and perform multiple reasoning steps. There is also the cost of information access, such as an internet search, although this would typically be less than LLM inference costs. There are various ways to improve LLM reasoning costs by reducing the number of tokens produced and processed by each reasoning step.

Meta-Reasoning

Meta-reasoning is thinking about how to solve a problem. People say that AI is bad at "planning," but I say that's being quite unfair. The first order of business for an LLM that's been presented with a user's question is planning. Rather than thinking of the answer, it should first think about the *plan* for how to answer.

The decisions it needs to consider (and be trained about) include:

- Whether to invoke a "tool" such as a clock or calculator.
- Should an internet search be done?
- Is a query of a "plugin" required?
- Does it need a chunk of a company-specific document (in a RAG system)?

I'm a little vague on the difference between a "tool" and a "plugin." And isn't an internet search kind of the same? And looking up a text chunk in a RAG vector database? They're all kind of "tools" that you "plug in" to your LLM. Maybe it's just me.

Anyway, all those integrations above all require an extra "step" or "hop" to some other integrated system. Not all queries require these steps. There are also other issues in relation to producing the answer directly from the LLM:

- What output format is required (e.g., text or HTML).
- Should the LLM refuse to answer this question?
- Should it ask the user a followup question before answering?

Want to know something funny? These are all the same! They're just training. If you want an LLM to refuse a query, you train it with bad queries and nice words that are refusals. If you want it to ask followup questions to clarify ambiguous questions, you train the LLM with ambiguous queries along with brief answers that contain the followup questions. If you want the LLM to invoke the clock when you ask for the time, you have to train it with queries about the time and answers that contain tricky "tool tokens" that run the clock. We're giving the LLM too much credit for being humanlike.

It's all just words to the LLM. Planning, not at all.

The final option in "planning" a response to a user's query is: *none of the above*. There are a lot of questions that an LLM can just answer immediately, without hesitation, and without consulting some other system. When it does this without any extra step, we say it's using its "parametric knowledge," because it's only using whatever has been trained into the many billions of "parameters" that are the weights in its model.

Don't worry if you don't understand how that all works, because I don't, and nobody else does. There's just a bunch of over a billion magic numbers that can help you fold your pillowcases in origami patterns.

Not Following Instructions

LLMs are great at following instructions, and it's largely considered a "solved problem" in AI. If you ask the AI to "pretend to be Gollum" or "answer in Klingon" then you can be fairly sure it will do so.

There's also a whole lot of research on *not* following instructions, because there are lots of answers that AIs shouldn't give. This whole area of theory is called "refusal training" and the opposite research is where users try to bypass the refusals, which is called "jailbreaking" the AI.

But recently, there was a weird example of LLMs having a totally different failure to follow instructions. The research by Shojaee et. al. (2025) at Apple found that LLMs could not use the information in questions to solve complex puzzle problems. This is a different type of "instructions" from the above:

- Procedural steps
- Abstract puzzle problems

Why did the AI fail? Well, it's all words to the LLM, and clearly the words in the question that described how to solve the puzzle couldn't be understood properly by the LLM. It's like it couldn't map the words given in the prompt to the steps it was using in its reasoning.

Part of me wants to say that maybe we need a whole different architecture that understands "concepts" better and that will be able to map words to concepts that represent abstract steps, and back again to words at the end. Surely there's a chasm to overcome here that needs a rewrite:

- Words
- Abstract concepts

The idea of not mapping words to reasoning steps seems somewhat reasonable, until you realize that reasoning models use words to describe their steps. In fact, they "think out loud" in words, whether it's doing a one-step or a multi-step reasoning method.

So, I wonder if I'm about to be taught another "bitter lesson" in having models follow solution steps in the question. Maybe the problem was just not enough training data where the solution was given in the problem. Maybe the AI engines was just not paying enough attention to the original question. I mean, who hasn't forgotten to read the whole question in a math test?

And there's also the fact that the basic level of LLM instruction following is simply based on training. LLMs don't follow even the most basic human instructions by themselves, but need a huge training set of examples of human instructions and the correct LLM output.

Then the poor AIs get smacked with a stick called "reinforcement learning" and "penalties" during training if they get the answer wrong. That's how they learn to speak Klingon.

Hence, I have a horrible feeling that this apparently huge hole in AI abstract reasoning, where the solution in the question is unusable by the AI, might be fixable with a few thousand examples of math quiz questions where they also contain big hints about the answer.

RAG Reasoning

The level of recent research attention to "multi-step reasoning" is off the charts. We've now seen papers on "chain-of-X" for almost every word in the English language. And this has now started to make its way into RAG research, although it's not a flood of papers by any means.

Every implementation of a RAG application can use any of those advanced reasoning algorithms, which would probably make it 1% smarter and 100% slower. Arguably, the type of advanced reasoning problems that OpenAI Strawberry or DeepSeek R1 is better at solving are not really in the bread basket for RAG, but it's worth a try!

Nevertheless, there's room for some more research on RAG and reasoning. For example, does RAG with Chain-of-Thought work better if you prepend the RAG chunks at every step of the way, or is it better just to have them at the first step? We're yet to see that in a research paper!

Implicit Associative Priors

There's a weird thing about LLMs where they shouldn't be good at something, but they are. If your LLM starts writing a story about a guy named Morgan, then the LLM will use the name "Morgan" again later in the story.

Everyone says that AIs are bad at "generalization."

However, if I change the first part of the story to be about a woman named Mary, rather than Morgan, then the LLM will dutifully write the rest of the story using the name "Mary" instead. The story will be changed in a very general way to use a different name and gender of the main character.

Sounds like generalization to me.

I can't really see how this works. Also, it's going to work regardless of exactly which word contains the name. I can add a few extra adjectives, changing its position, and the name will still get re-used.

Note that I'm not talking about "context windows" where it used to be that the LLMs would forget that name if it was 4,000 words earlier in the story. Rather, I'm asking the question: how does it know to output the same name? The first time, sure, it just chooses a random name. What about the second time? What is it in its training that tells it to re-use whatever name tokens appeared earlier in the story, rather than generating a random name each time.

The answer is called "associative priors," and the LLM starts to learn to re-use tokens earlier in its context, rather than any fixed name. This behavior arises over the course of extensive training whereby the model is paying "attention" to the tokens in the previous text.

In fact, to learn to pay attention to tokens at an earlier position, there is a technology called "positional encoding." Weirdly, this technology doesn't actually try to tell the LLM which position is best, but positional encoding simply makes the numbers at *every* position a little bit different, and then the LLM *learns* which positions are important during training.

Don't worry; nobody else understands how this works, either.

Somehow, the combination of weights and "positional encoding" allows the LLM to re-use words from earlier in the text. Eventually, after passing a lot of training data under the bridge, this generalizes not just to know what words fit after other ones, but also to the higher-level answer of re-using the existing tokens at an earlier position, irrespective of the value of those tokens.

Really Radical Reasoning References

Implicit Associative Priors. Research papers on associative priors:

1. Giorgos Borboudakis, Ioannis Tsamardinos, 9 Aug 2014, *Scoring and Searching over Bayesian Networks with Causal and Associative Priors*, https://arxiv.org/abs/1408.2057
2. Yuwei Sun, Hideya Ochiai, Zhirong Wu, Stephen Lin, Ryota Kanai, 11 Mar 2025 (v4), *Associative Transformer*, https://arxiv.org/abs/2309.12862
3. Frederick Liu and Besim Avci, 2019, *Incorporating Priors with Feature Attribution on Text Classification*, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6274–6283, Florence, Italy. Association for Computational Linguistics, https://aclanthology.org/P19-1631/, PDF: https://aclanthology.org/P19-1631.pdf
4. Menglin Wang, Zhun Zhong, Xiaojin Gong, 13 Feb 2025, *Prior-Constrained Association Learning for Fine-Grained Generalized Category Discovery*, https://arxiv.org/abs/2502.09501
5. Shuai Li, Kui Jia, Xiaogang Wang, 12 Jan 2017 (v2), *Automatic Discoveries of Physical and Semantic Concepts via Association Priors of Neuron Groups*, https://arxiv.org/abs/1612.09438
6. Feng He, Chao Zhang, Zhixue Zhao, 4 Dec 2024, *Implicit Priors Editing in Stable Diffusion via Targeted Token Adjustment*, https://arxiv.org/abs/2412.03400
7. J. Urain, A. T. Le, A. Lambert, G. Chalvatzaki, B. Boots and J. Peters, 2022, *Learning Implicit Priors for Motion Optimization*, 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 2022, pp. 7672-7679, doi: 10.1109/IROS47612.2022.9981264, https://ieeexplore.ieee.org/abstract/document/9981264

One-Step Reasoning. Research papers on one-step reasoning include:

1. Zhipeng Chen, Yingqian Min, Beichen Zhang, Jie Chen, Jinhao Jiang, Daixuan Cheng, Wayne Xin Zhao, Zheng Liu, Xu Miao, Yang Lu, Lei Fang, Zhongyuan Wang, Ji-Rong Wen, 6 Mar 2025, *An Empirical Study on Eliciting and Improving R1-like Reasoning Models*, https://arxiv.org/abs/2503.04548 https://github.com/RUCAIBox/Slow_Thinking_with_LLMs
2. Yijiong Yu, 4 Dec 2024 (v3), *Patience Is The Key to Large Language Model Reasoning*, https://arxiv.org/abs/2411.13082 (Training a reasoning model to give longer one-step answers using training data with long answers as positive examples and short answers as negative answers.)
3. Yijiong Yu, 16 Jan 2025 (v4), *Do LLMs Really Think Step-by-step In Implicit Reasoning?* https://arxiv.org/abs/2411.15862 https://github.com/yuyijiong/if_step_by_step_implicit_CoT

4. Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, Wanxiang Che, 13 Mar 2025 (v2), *Towards Reasoning Era: A Survey of Long Chain-of-Thought for Reasoning Large Language Models*, https://arxiv.org/abs/2503.09567 (Massive and broad survey of all types of reasoning.)
5. Jiaran Ye, Zijun Yao, Zhidian Huang, Liangming Pan, Jinxin Liu, Yushi Bai, Amy Xin, Liu Weichuan, Xiaoyin Che, Lei Hou, Juanzi Li, 29 May 2025, *How does Transformer Learn Implicit Reasoning?* https://arxiv.org/abs/2505.23653
6. Michael Nuñez, July 15, 2025, *OpenAI, Google DeepMind and Anthropic sound alarm: 'We may be losing the ability to understand AI'*, https://venturebeat.com/ai/openai-google-deepmind-and-anthropic-sound-alarm-we-may-be-losing-the-ability-to-understand-ai/ (Monitoring the text-based interim "thinking-out-loud" reasoning of models in CoT.)
7. Tomek Korbak, Mikita Balesni, (and many more authors) July 2025, *Chain of Thought Monitorability: A New and Fragile Opportunity for AI Safety*, https://tomekkorbak.com/cot-monitorability-is-a-fragile-opportunity/cot_monitoring.pdf

Multi-Hop Reasoning. Research papers on multi-step or multi-hop reasoning include:

1. Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, Karthik Narasimhan, 3 Dec 2023 (v2), *Tree of Thoughts: Deliberate Problem Solving with Large Language Models*, https://arxiv.org/abs/2305.10601 Code: https://github.com/princeton-nlp/tree-of-thought-llm
2. Justin Chih-Yao Chen, Archiki Prasad, Swarnadeep Saha, Elias Stengel-Eskin, Mohit Bansal, 18 Sep 2024, *MAgICoRe: Multi-Agent, Iterative, Coarse-to-Fine Refinement for Reasoning*, https://arxiv.org/abs/2409.12147 https://github.com/dinobby/MAgICoRe
3. Xiaohan Xu, Chongyang Tao, Tao Shen, Can Xu, Hongbo Xu, Guodong Long, Jian-guang Lou, 29 Feb 2024 (v2), *Re-Reading Improves Reasoning in Large Language Models*, https://arxiv.org/abs/2309.06275
4. TED, Oct 2024, *Multi-Step Reasoning Agents*, https://tedai-sanfrancisco.ted.com/glossary/multi-step-reasoning-agents/
5. Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, Tushar Khot, 30 Jan 2023 (v2), *Complexity-Based Prompting for Multi-Step Reasoning*, https://arxiv.org/abs/2210.00720
6. Junting Lu, Oct 2024 (accessed), *Awesome-LLM-Reasoning-Techniques*, https://github.com/Junting-Lu/Awesome-LLM-Reasoning-Techniques
7. Cameron R. Wolfe, Dec 23, 2023, *Tree of Thoughts Prompting: Solving multi-step problems with LLMs via deliberate planning and exploration*, https://towardsdatascience.com/tree-of-thoughts-prompting-65a3e51f9ac4
8. Data Camp, Jul 10, 2024, *Chain-of-Thought Prompting: Step-by-Step Reasoning with LLMs*, https://www.datacamp.com/tutorial/chain-of-thought-prompting

9. Pankaj, Dec 21, 2023, *Chain of Thought Prompting: Guiding LLMs Step-by-Step*, https://medium.com/@pankaj_pandey/chain-of-thought-prompting-guiding-llms-step-by-step-e6eac32d02d8
10. Cobus Greyling, Aug 2, 2023, *12 Prompt Engineering Techniques*, https://cobusgreyling.medium.com/12-prompt-engineering-techniques-644481c857aa
11. Cameron R. Wolfe, Jan 3, 2024, *Graph-Based Prompting and Reasoning with Language Models: Understanding graph of thoughts prompting and several variants*, https://towardsdatascience.com/graph-based-prompting-and-reasoning-with-language-models-d6acbcd6b3d8
12. Jason Wei and Denny Zhou, May 11, 2022, *Language Models Perform Reasoning via Chain of Thought*, https://research.google/blog/language-models-perform-reasoning-via-chain-of-thought/
13. Tanay Jaipuria, Oct 29, 2024, *OpenAI's o-1 and inference-time scaling laws*, https://www.tanayj.com/p/openais-o-1-and-inference-time-scaling
14. Jinlin Wang, Suyuchen Wang, Ziwen Xia, Sirui Hong, Yun Zhu, Bang Liu, Chenglin Wu, 28 Oct 2024, *FACT: Examining the Effectiveness of Iterative Context Rewriting for Multi-fact Retrieval*, https://arxiv.org/abs/2410.21012
15. Carl Franzen, November 20, 2024, *DeepSeek's first reasoning model R1-Lite-Preview turns heads, beating OpenAI o1 performance*, https://venturebeat.com/ai/deepseeks-first-reasoning-model-r1-lite-preview-turns-heads-beating-openai-o1-performance/
16. mshumer, Nov 2024, *Open Reasoning Engine*, https://github.com/mshumer/OpenReasoningEngine
17. Eric Horvitz , Harsha Nori , Naoto Usuyama , November 27, 2024 *Advances in run-time strategies for next-generation foundation models*, Microsoft Research Blog, https://www.microsoft.com/en-us/research/blog/advances-in-run-time-strategies-for-next-generation-foundation-models/
18. Harsha Nori, Naoto Usuyama, Nicholas King, Scott Mayer McKinney, Xavier Fernandes, Sheng Zhang, Eric Horvitz, 6 Nov 2024, *From Medprompt to o1: Exploration of Run-Time Strategies for Medical Challenge Problems and Beyond*, https://arxiv.org/abs/2411.03590
19. Hieu Tran, Zonghai Yao, Junda Wang, Yifan Zhang, Zhichao Yang, Hong Yu, 5 Dec 2024 (v2), *RARE: Retrieval-Augmented Reasoning Enhancement for Large Language Models*, https://arxiv.org/abs/2412.02830
20. Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yimin Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye Ge, Kai Chen, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, Wenhai Wang, 6 Dec 2024, *Expanding Performance Boundaries of Open-Source Multimodal Models with Model, Data, and Test-Time Scaling*, https://arxiv.org/abs/2412.05271

21. Mingchen Zhuge, Changsheng Zhao, Dylan Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, Yangyang Shi, Vikas Chandra, Jürgen Schmidhuber, 16 Oct 2024 (v2), *Agent-as-a-Judge: Evaluate Agents with Agents*, https://arxiv.org/abs/2410.10934
22. Kyle Wiggers, December 14, 2024, *'Reasoning' AI models have become a trend, for better or worse*, https://techcrunch.com/2024/12/14/reasoning-ai-models-have-become-a-trend-for-better-or-worse/
23. Arda Sevinc, Abdurrahman Gumus, 9 Dec 2024, *AutoReason: Automatic Few-Shot Reasoning Decomposition*, https://arxiv.org/abs/2412.06975
24. Ekin Akyürek, Mehul Damani, Linlu Qiu, Han Guo, Yoon Kim, Jacob Andreas, 11 Nov 2024, *The Surprising Effectiveness of Test-Time Training for Abstract Reasoning*, https://arxiv.org/abs/2411.07279
25. Noam Brown, Tuomas Sandholm, 16 Nov 2017 (v3), *Safe and Nested Subgame Solving for Imperfect-Information Games*, https://arxiv.org/abs/1705.02955 (An early pre-AI paper on reasoning in multiple steps.)
26. Maxwell Zeff, November 20, 2024, *Current AI scaling laws are showing diminishing returns, forcing AI labs to change course*, https://techcrunch.com/2024/11/20/ai-scaling-laws-are-showing-diminishing-returns-forcing-ai-labs-to-change-course/ ("at least 10 to 20x gains in model performance ...intelligent prompting, UX decisions, and passing context at the right time into the models...")
27. Agnostiq, Dec 2024, *multi-agent-llm: LLM based Multi-Agent methods: Lean implementation of various multi-agent LLM methods, including Iteration of Thought (IoT)*, https://github.com/AgnostiqHQ/multi-agent-llm
28. Xiangjue Dong, Maria Teleki, James Caverlee, 18 Dec 2024, *A Survey on LLM Inference-Time Self-Improvement*, https://arxiv.org/abs/2412.14352 https://github.com/dongxiangjue/Awesome-LLM-Self-Improvement (Broad survey of reasoning improvement methods from multi-step inference to RALM to decoding algorithms.)
29. Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, Dong Yu, 30 Dec 2024, *Do NOT Think That Much for 2+3=? On the Overthinking of o1-Like LLMs*, https://arxiv.org/abs/2412.21187
30. Rohin Manvi, Anikait Singh, Stefano Ermon, 3 Oct 2024, *Adaptive Inference-Time Compute: LLMs Can Predict if They Can Do Better, Even Mid-Generation*, https://arxiv.org/abs/2410.02725
31. Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li, 19 Jan 2024, *Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning*, The Twelfth International Conference on Learning Representations, 2024, https://arxiv.org/abs/2401.10480 https://github.com/Yiwei98/ESC (Uses "early stopping" idea to improve CoT efficiency during inference.)
32. Akash Bajwa, Jan 06, 2025, *Test-Time Search: A Path To AGI: Stacking Scaling Laws And Reward Engineering*, https://akashbajwa.substack.com/p/test-time-search-a-path-to-agi

33. Zekun Xi, Wenbiao Yin, Jizhan Fang, Jialong Wu, Runnan Fang, Ningyu Zhang, Jiang Yong, Pengjun Xie, Fei Huang, Huajun Chen, 16 Jan 2025, *OmniThink: Expanding Knowledge Boundaries in Machine Writing through Thinking*, https://arxiv.org/abs/2501.09751 (Iteratively going deeper into a topic while generating.)
34. Siddharth Narayanan, James D. Braza, Ryan-Rhys Griffiths, Manu Ponnapati, Albert Bou, Jon Laurent, Ori Kabeli, Geemi Wellawatte, Sam Cox, Samuel G. Rodriques, Andrew D. White, 30 Dec 2024, *Aviary: training language agents on challenging scientific tasks*, https://arxiv.org/abs/2412.21154 (Using smaller models combined with multi-step reasoning to compete with big models with 100x less inference cost.)
35. Kuang-Huei Lee, Ian Fischer, Yueh-Hua Wu, Dave Marwood, Shumeet Baluja, Dale Schuurmans, Xinyun Chen, 17 Jan 2025, *Evolving Deeper LLM Thinking*, https://arxiv.org/abs/2501.09891 (An alternative search strategy broad/deep, compared to CoT and reflection.)
36. Edward Beeching, Lewis Tunstall, Sasha Rush Dec 16, 2024, *Scaling Test Time Compute with Open Source Models*, https://huggingface.co/spaces/HuggingFaceH4/blogpost-scaling-test-time-compute
37. Maciej Besta, Julia Barth, Eric Schreiber, Ales Kubicek, Afonso Catarino, Robert Gerstenberger, Piotr Nyczyk, Patrick Iff, Yueling Li, Sam Houliston, Tomasz Sternal, Marcin Copik, Grzegorz Kwaśniewski, Jürgen Müller, Łukasz Flis, Hannes Eberhard, Hubert Niewiadomski, Torsten Hoefler, 23 Jan 2025 (v3), *Reasoning Language Models: A Blueprint*, https://arxiv.org/abs/2501.11223 (Survey and blueprint for how to build a Large Reasoning Model.)
38. Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, Tatsunori Hashimoto, 3 Feb 2025 (v2), *s1: Simple test-time scaling*, https://arxiv.org/abs/2501.19393 https://github.com/simplescaling/s1 (Method of "budget forcing" that allows either shortening or lengthening multi-step reasoning sequences.)
39. Manish Sanwal, 3 Feb 2025 (v2), *Layered Chain-of-Thought Prompting for Multi-Agent LLM Systems: A Comprehensive Approach to Explainable Large Language Models*, https://arxiv.org/abs/2501.18645
40. Sebastian Raschka, PhD, Feb 05, 2025, *Understanding Reasoning LLMs: Methods and Strategies for Building and Refining Reasoning Models*, https://magazine.sebastianraschka.com/p/understanding-reasoning-llms
41. Ling Yang, Zhaochen Yu, Bin Cui, Mengdi Wang, 10 Feb 2025, *ReasonFlux: Hierarchical LLM Reasoning via Scaling Thought Templates*, https://arxiv.org/abs/2502.06772 https://github.com/Gen-Verse/ReasonFlux (RALM-like retrieval of reasoning prompt templates at inference time.)
42. Hanmeng Liu, Zhizhang Fu, Mengru Ding, Ruoxi Ning, Chaoli Zhang, Xiaozhang Liu, Yue Zhang, 13 Feb 2025, *Logical Reasoning in Large Language Models: A Survey*, https://arxiv.org/abs/2502.09100

43. Zeping Yu, Yonatan Belinkov, Sophia Ananiadou, 15 Feb 2025, *Back Attention: Understanding and Enhancing Multi-Hop Reasoning in Large Language Models*, https://arxiv.org/abs/2502.10835
44. Dacheng Li, Shiyi Cao, Chengkun Cao, Xiuyu Li, Shangyin Tan, Kurt Keutzer, Jiarong Xing, Joseph E. Gonzalez, Ion Stoica, 20 Feb 2025, *S*: Test Time Scaling for Code Generation*, https://arxiv.org/abs/2502.14382 https://github.com/NovaSky-AI/SkyThought
45. Ben Dickson, February 20, 2025, *How test-time scaling unlocks hidden reasoning abilities in small language models (and allows them to outperform LLMs)*, https://venturebeat.com/ai/how-test-time-scaling-unlocks-hidden-reasoning-abilities-in-small-language-models-and-allows-them-to-outperform-llms/
46. Zhiyuan Zeng, Qinyuan Cheng, Zhangyue Yin, Yunhua Zhou, Xipeng Qiu, 17 Feb 2025, *Revisiting the Test-Time Scaling of o1-like Models: Do they Truly Possess Test-Time Scaling Capabilities?* https://arxiv.org/abs/2502.12215
47. Zihao Zeng, Xuyao Huang, Boxiu Li, Zhijie Deng, 19 Feb 2025, *SIFT: Grounding LLM Reasoning in Contexts via Stickers*, https://arxiv.org/abs/2502.14922 https://github.com/zhijie-group/SIFT (Multi-step reasoning where the LLM first generates a modified prompt that summarizes the key points, and then does inference for both the original and modified prompts, then comparing results and adjusting forwards and backwards.)
48. Marthe Ballon, Andres Algaba, Vincent Ginis, 21 Feb 2025, *The Relationship Between Reasoning and Performance in Large Language Models -- o3 (mini) Thinks Harder, Not Longer*, https://arxiv.org/abs/2502.15631
49. Maxwell Zeff, February 24, 2025, *Anthropic launches a new AI model that 'thinks' as long as you want*, https://techcrunch.com/2025/02/24/anthropic-launches-a-new-ai-model-that-thinks-as-long-as-you-want/
50. Yuchen Yan, Yongliang Shen, Yang Liu, Jin Jiang, Mengdi Zhang, Jian Shao, Yueting Zhuang, 13 Mar 2025 (v2), *InftyThink: Breaking the Length Limits of Long-Context Reasoning in Large Language Models*, https://arxiv.org/abs/2503.06692
51. Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, Wanxiang Che, 13 Mar 2025 (v2), *Towards Reasoning Era: A Survey of Long Chain-of-Thought for Reasoning Large Language Models*, https://arxiv.org/abs/2503.09567 (Massive and broad survey of all types of reasoning.)
52. Eric Zhao, Pranjal Awasthi, Sreenivas Gollapudi, 20 Feb 2025 (v2), *Sample, Scrutinize and Scale: Effective Inference-Time Search by Scaling Verification*, https://arxiv.org/abs/2502.01839 (Wrapping a single model with a Best-of-N approach that self-selects the best answer can significantly improve reasoning rates.)
53. Qianjun Pan, Wenkai Ji, Yuyang Ding, Junsong Li, Shilian Chen, Junyi Wang, Jie Zhou, Qin Chen, Min Zhang, Yulan Wu, Liang He, 8 May 2025 (v2), *A Survey of Slow Thinking-based Reasoning LLMs using Reinforced Learning and Inference-time Scaling Law*, https://arxiv.org/abs/2505.02665

54. Michael Nuñez, July 15, 2025, *OpenAI, Google DeepMind and Anthropic sound alarm: 'We may be losing the ability to understand AI'*, https://venturebeat.com/ai/openai-google-deepmind-and-anthropic-sound-alarm-we-may-be-losing-the-ability-to-understand-ai/ (Monitoring the text-based interim "thinking-out-loud" reasoning of models in CoT.)
55. Tomek Korbak, Mikita Balesni, (and many more authors) July 2025, *Chain of Thought Monitorability: A New and Fragile Opportunity for AI Safety*, https://tomekkorbak.com/cot-monitorability-is-a-fragile-opportunity/cot_monitoring.pdf

Deep Research Models. Research papers on deep research models include:

1. HuggingFace, February 3, 2025, *OpenAI's Deep Research vs DeepSeek R1*, https://huggingface.co/blog/LLMhacker/openais-deep-research-vs-deepseek-r1
2. Timothy B. Lee, Feb 24, 2025, *These experts were stunned by OpenAI Deep Research: "I would use this model professionally," an antitrust lawyer told me*, https://www.understandingai.org/p/these-experts-were-stunned-by-openai
3. Ravi Teja,: December 24, 2024, *Google Gemini's Deep Research: What is it and How to Use it?* https://techwiser.com/google-geminis-deep-research-what-is-it-and-how-to-use-it/
4. Dave Citron, Dec 11, 2024, *Try Deep Research and our new experimental model in Gemini, your AI assistant: Deep Research rolls out to Gemini Advanced subscribers today, saving you hours of time. Plus, you can now try out a chat optimized version of 2.0 Flash Experimental in Gemini on the web*, Google Blog, https://blog.google/products/gemini/google-gemini-deep-research/
5. Sundar Pichai, Demis Hassabis, Koray Kavukcuoglu, Dec 11, 2024, *Introducing Gemini 2.0: our new AI model for the agentic era*, Google Blog, https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/
6. Emma Roth, Dec 12, 2024, *Google built an AI tool that can do research for you: With Deep Research, you can ask Gemini to scour the web on your behalf and write up a report based on its findings*, https://www.theverge.com/2024/12/11/24318217/google-gemini-advanced-deep-research-launch
7. Asif Razzaq, March 8, 2025, *Meet Manus: A New AI Agent from China with Deep Research + Operator + Computer Use + Lovable + Memory*, https://www.marktechpost.com/2025/03/08/meet-manus-a-new-ai-agent-from-china-with-deep-research-operator-computer-use-lovable-memory/
8. Jordan Gibbs, Feb 17, 2025, *4 Lifechanging ChatGPT Features You May Not Know About (Feb. 2025): ChatGPT has been releasing a ton of powerful features recently… Are you caught up?* https://medium.com/@jordan_gibbs/4-lifechanging-chatgpt-features-you-may-not-know-about-feb-2025-01eaeb4e68c1
9. Jim the AI Whisperer, Mar 10, 2025, *My "Prompt Grafting" technique outperforms Deep Research in head-to-head test — and is 5900% faster: My prompt gets more insightful, comprehensive research from AI*, https://generativeai.pub/prompt-grafting-vs-deep-research-faster-better-ai-essays-824968b1a47a

Hybrid Reasoning Models. Research papers on hybrid one-step/multi-step reasoning include:

1. Maxwell Zeff, February 24, 2025, *Anthropic launches a new AI model that 'thinks' as long as you want*, https://techcrunch.com/2025/02/24/anthropic-launches-a-new-ai-model-that-thinks-as-long-as-you-want/
2. Xiaoyu Tian, Liangyu Chen, Na Liu, Yaxuan Liu, Wei Zou, Kaijiang Chen, Ming Cui, 24 Nov 2023 (v4), *DUMA: a Dual-Mind Conversational Agent with Fast and Slow Thinking*, https://arxiv.org/abs/2310.18075
3. Daniele Paliotta, Junxiong Wang, Matteo Pagliardini, Kevin Y. Li, Aviv Bick, J. Zico Kolter, Albert Gu, François Fleuret, Tri Dao, 27 Feb 2025, *Thinking Slow, Fast: Scaling Inference Compute with Distilled Reasoners*, https://arxiv.org/abs/2502.20339
4. Jianyuan Zhong, Zeju Li, Zhijian Xu, Xiangyu Wen, Qiang Xu, 16 Feb 2025, *Dyve: Thinking Fast and Slow for Dynamic Process Verification*, https://arxiv.org/abs/2502.11157
5. Kangan Qian, Zhikun Ma, Yangfan He, Ziang Luo, Tianyu Shi, Tianze Zhu, Jiayin Li, Jianhui Wang, Ziyu Chen, Xiao He, Yining Shi, Zheng Fu, Xinyu Jiao, Kun Jiang, Diange Yang, Takafumi Matsumaru, 27 Nov 2024, *FASIONAD : FAst and Slow FusION Thinking Systems for Human-Like Autonomous Driving with Adaptive Feedback*, https://arxiv.org/abs/2411.18013
6. DiJia Su, Sainbayar Sukhbaatar, Michael Rabbat, Yuandong Tian, Qinqing Zheng, 13 Oct 2024, *Dualformer: Controllable Fast and Slow Thinking by Learning with Randomized Reasoning Traces*, https://arxiv.org/abs/2410.09918
7. Konstantina Christakopoulou, Shibl Mourad, Maja Matarić, 10 Oct 2024, *Agents Thinking Fast and Slow: A Talker-Reasoner Architecture*, https://arxiv.org/abs/2410.08328
8. Pengbo Hu, Ji Qi, Xingyu Li, Hong Li, Xinqi Wang, Bing Quan, Ruiyu Wang, Yi Zhou, 21 Aug 2023 (v2), *Tree-of-Mixed-Thought: Combining Fast and Slow Thinking for Multi-hop Visual Reasoning*, https://arxiv.org/abs/2308.09658
9. Thilo Hagendorff, Sarah Fabi, Michal Kosinski, 2 Aug 2023 (v2), *Thinking Fast and Slow in Large Language Models*, https://arxiv.org/abs/2212.05206
10. Wenlin Yao, Haitao Mi, Dong Yu, 25 Sep 2024, *HDFlow: Enhancing LLM Complex Problem-Solving with Hybrid Thinking and Dynamic Workflows*, https://arxiv.org/abs/2409.17433
11. Kyle Wiggers, March 4, 2025, *Amazon is reportedly developing its own AI 'reasoning' model: Amazon reportedly wants to get in on the AI "reasoning" model game*, https://techcrunch.com/2025/03/04/amazon-is-reportedly-developing-its-own-ai-reasoning-model/
12. X Zhang, F Zhang, C Du, C Du, T Pang, W Gao, M Lin, Mar 2025, *LightTransfer: Your Long-Context LLM is Secretly a Hybrid Model with Effortless Adaptation*, https://openreview.net/pdf?id=DfgfGTfObm
13. Supreeth Koundinya, March 10, 2025, *Manus is a Wrapper of Anthropic's Claude, and It's Okay*, https://analyticsindiamag.com/ai-features/manus-is-a-wrapper-of-anthropics-claude-and-its-okay/ ("Manus didn't just slap an API on a model. They built an autonomous system that can execute deep research, deep thinking, and multi-step tasks in a way that no other AI have.")

14. Sean Michael Kerner, March 18, 2025, *Nvidia debuts Llama Nemotron open reasoning models in a bid to advance agentic AI*, https://venturebeat.com/ai/nvidia-debuts-llama-nemotron-open-reasoning-models-in-a-bid-to-advance-agentic-ai/
15. Xiaoye Qu, Yafu Li, Zhaochen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong Liu, Shuxian Liang, Junxian He, Peng Li, Wei Wei, Jing Shao, Chaochao Lu, Yue Zhang, Xian-Sheng Hua, Bowen Zhou, Yu Cheng, 27 Mar 2025, *A Survey of Efficient Reasoning for Large Reasoning Models: Language, Multimodality, and Beyond*, https://arxiv.org/abs/2503.21614

RAG Reasoning. Research papers on reasoning models and RAG include:

1. B Zhan, A Li, X Yang, D He, Y Duan, S Yan, 2024, *RARoK: Retrieval-Augmented Reasoning on Knowledge for Medical Question Answering*, 2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pages 2837-2843, DOI: 10.1109/BIBM62325.2024.10822341, https://www.computer.org/csdl/proceedings-article/bibm/2024/10822341/23onp6dXOSI (RAG combined with Chain-of-Thought for medical reasoning.)
2. Xinyan Guan, Jiali Zeng, Fandong Meng, Chunlei Xin, Yaojie Lu, Hongyu Lin, Xianpei Han, Le Sun, Jie Zhou. 3 Feb 2025, *DeepRAG: Thinking to Retrieval Step by Step for Large Language Models*, https://arxiv.org/abs/2502.01142
3. P Verma, SP Midigeshi, G Sinha, A Solin, N Natarajan, Mar 2025, *Plan *RAG: Efficient Test-Time Planning for Retrieval Augmented Generation*, ICLR 2025 review, https://openreview.net/pdf?id=gi9aqlYdBk (Improve RAG reasoning efficiency via planning for parallel reasoning.)
4. Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, Wanxiang Che, 13 Mar 2025 (v2), *Towards Reasoning Era: A Survey of Long Chain-of-Thought for Reasoning Large Language Models*, https://arxiv.org/abs/2503.09567 (Massive and broad survey with all types of reasoning.)
5. Mingyue Cheng, Yucong Luo, Jie Ouyang, Qi Liu, Huijie Liu, Li Li, Shuo Yu, Bohou Zhang, Jiawei Cao, Jie Ma, Daoyu Wang, Enhong Chen, 17 Mar 2025 (v2), *A Survey on Knowledge-Oriented Retrieval-Augmented Generation*, https://arxiv.org/abs/2503.10677
6. Thang Nguyen, Peter Chin, Yu-Wing Tai, 26 May 2025, *MA-RAG: Multi-Agent Retrieval-Augmented Generation via Collaborative Chain-of-Thought Reasoning*, https://arxiv.org/abs/2505.20096

10. Concepts

"Learning to fly is not pretty

but flying is."

— Satya Nadella, *Hit Refresh*, 2017.

What are Concepts?

It's a good question and there's not a really satisfactory answer. If concepts were a simple concept (haha), then AI reasoning models would be a lot easier to build. The basic ideas about concepts include:

- High-level meaning
- Semantics not syntax

Concepts are about meaning, not words. There are also different levels of concepts, such as:

- Basic things — e.g., nouns like "cat"
- Basic actions — e.g., verbs like "jump"
- Abstract meanings — e.g., "symmetrical"
- Emotional meanings — e.g., "happiness"

As you can see from the above, it's not really clear what a concept really means (i.e., the "concept of a concept" is vague!). However, there's one thing that's clear about concepts:

Humans are better at concepts than AI models.

Maybe words are the problem. Humans think in images and relationships and concepts, whereas words are somewhat plotted onto these ideas.

Hence, there's a whole area of LLM research about having them think in "concepts" rather than words. The idea is to still use "tokens" but to have these tokens represent higher-levels as "concept tokens" rather than just words.

The terminology for AI reasoning about concepts is usually called "latent reasoning" because it's done in the "latent space" of abstract representations. I guess that's distinct from the "word space" where most LLMs are stuck at the moment.

The idea of concept tokens is not especially revolutionary. After all, in the theory of AI, we've already had:

- Stop tokens
- Pause tokens
- Formatting tokens
- Control tokens
- Separator tokens
- Fine-tuning tokens
- Tool tokens

Hence, yeah sure, why not have concept tokens.

In fact, concept-level tokens are used in several major areas of AI intelligence research:

- Prompt tuning ("prefix tuning")
- Chain-of-Thought decoding tokens ("reasoning tokens")
- Large Concept Models

Prompt tuning is a way to make the AI smarter by adding some special concept tokens to the input. This provides extra context to a user's question that allows the LLM to better understand what it's "reading" in the input context.

Reasoning tokens have been used as way to speed up the multi-hop reasoning models. Hence, they are more about efficiency than adding extra intelligence, though obviously the overall reasoning model is about adding capabilities. However, imagine what might be possible if you started with a rewrite of the architecture and put a lot of concept tokens together in a big model.

Large Concept Models

The development of Large Concept Models, or LCMs, was pioneered by Meta in December, 2024, from its Facebook AI Research lab (Barrault et. al., 2024). As you may have correctly guessed, LCMs are models that:

1. Use concept tokens, and

2. Are large.

The LCM built by Meta operates on concepts at the sentence level. Hence, it tries to understand written sentences, rather than individual words, to discern the one or more major concepts that a sentence aims to describe. They train models using a huge amount of data, with over 2.7 trillion tokens of input training sets.

This approach generalizes the use of concept tokens from their specific uses as reasoning tokens in multi-hop inference, and prefix tuning with concept tokens as extra non-word prefix tokens in prompt engineering.

Concept tokens are central to LCMs. They are trained in concept tokens, and also perform inference in the concept token space. Hence, they're not using the old-style "word tokens" as used by most LLMs today. Nevertheless, LCMs and LLMs have the same types of uses:

- Generating text or answers
- Summarization of inputs
- Reasoning about answers

As shown in various research papers, this idea is workable at scales both small and large. Large Concept Models may indeed be a major step forward beyond the simple tokenized LLMs. The main advantages of using concepts are that the resulting models are:

- Smarter — better understanding of concepts and ideas.
- Faster — fewer tokens to process, because each token is "bigger."
- Understandable — you can trace what concepts were used to answer.

However, LCMs are still in the early stages of their theory, and even the Facebook paper says they are offering "an attempt at an architecture" rather than a finalized technology. One of the areas of difficulty in this method is to train an "encoder" that converts sentences into concept tokens. This encoder has to do "sentence-level embeddings" analysis. Maybe we'll be going back to 2017 with an encoder-decoder architecture soon.

LCMs are certainly interesting, and they are one possible candidate for an extra leap needed to advance further towards achieving AGI. They're competing against other "next-gen" ideas for AI intelligence, such as:

- Reasoning models
- Symbolic execution
- Knowledge graphs
- State Space Models (SSMs) such as Mamba and Hyena.
- Embodied AI models

Time will tell which ideas win out.

References

Concept Tokens: Research papers on the use of concept tokens to represent higher-order types of thinking include:

1. Sachin Kumar, Sep 17, 2024, *Hidden Chain-of-Thought decoding: faster and efficient CoT decoding to improve reasoning of LLMs* https://medium.com/@techsachin/hidden-chain-of-thought-decoding-faster-and-efficient-cot-decoding-to-improve-reasoning-of-llms-d95584bc9346 (Token reduction in CoT by compressing language tokens into an internal "hidden" concise token representation.)
2. Tianqiao Liu, Zui Chen, Zitao Liu, Mi Tian, Weiqi Luo, 13 Sep 2024, *Expediting and Elevating Large Language Model Reasoning via Hidden Chain-of-Thought Decoding*, https://arxiv.org/abs/2409.08561
3. Lance Eliot, Dec 18, 2024, *Chain Of Continuous Thought Promises Mighty Boost For LLMs And Generative AI By Blowing Up The Fixation On Tokens*, https://www.forbes.com/sites/lanceeliot/2024/12/18/chain-of-continuous-thought-promises-mighty-boost-for-llms-and-generative-ai-by-blowing-up-the-fixation-on-tokens/
4. Kyle Orland, 13 Dec 2024, *Are LLMs capable of non-verbal reasoning? Processing in the "latent space" could help AI with tricky logical questions*, https://arstechnica.com/ai/2024/12/are-llms-capable-of-non-verbal-reasoning/
5. Alex McFarland, December 16, 2024, *Meta's COCONUT: The AI Method That Thinks Without Language*, https://www.unite.ai/metas-coconut-the-ai-method-that-thinks-without-language/

6. Maxime Peyrard, Martin Josifoski, Robert West, 21 Mar 2024, *The Era of Semantic Decoding*, https://arxiv.org/abs/2403.14562
7. Hanyu Zhang, Xiting Wang, Chengao Li, Xiang Ao, Qing He, 10 Jan 2025, *Controlling Large Language Models Through Concept Activation Vectors*, https://arxiv.org/abs/2501.05764 (Training a vector used to control the model on certain attributes.)
8. Deqian Kong, Minglu Zhao, Dehong Xu, Bo Pang, Shu Wang, Edouardo Honig, Zhangzhang Si, Chuan Li, Jianwen Xie, Sirui Xie, Ying Nian Wu, 3 Feb 2025, *Scalable Language Models with Posterior Inference of Latent Thought Vectors*, https://arxiv.org/abs/2502.01567
9. Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, Tom Goldstein, 7 Feb 2025, *Scaling up Test-Time Compute with Latent Reasoning: A Recurrent Depth Approach*, https://arxiv.org/abs/2502.05171
10. DiJia Su, Hanlin Zhu, Yingchen Xu, Jiantao Jiao, Yuandong Tian, Qinqing Zheng, 5 Feb 2025. *Token Assorted: Mixing Latent and Text Tokens for Improved Language Model Reasoning*, https://arxiv.org/abs/2502.03275
11. Jihoon Tack, Jack Lanchantin, Jane Yu, Andrew Cohen, Ilia Kulikov, Janice Lan, Shibo Hao, Yuandong Tian, Jason Weston, Xian Li, 12 Feb 2025, *LLM Pretraining with Continuous Concepts*, https://arxiv.org/abs/2502.08524
12. Vivek K. Tiwari, 2025, *Towards Practical Concept-Based Language Models: An Efficiency-Focused Implementation*, https://www.researchgate.net/profile/Vivek-Tiwari-41/publication/388753941_Towards_Practical_Concept-Based_Language_Models_An_Efficiency-Focused_Implementation/links/67a4bf86461fb56424cc6b62/Towards-Practical-Concept-Based-Language-Models-An-Efficiency-Focused-Implementation.pdf
13. J Liao, R Xie, S Li, X Wang, X Sun, Z Kang, X He, 2025, *Multi-Grained Patch Training for Efficient LLM-based Recommendation*, https://hexiangnan.github.io/papers/sigir25-PatchRec.pdf

Large Concept Models (LCMs). Research papers on the use of concept tokens and latent space reasoning in very large models include:

1. LCM team, Loïc Barrault, Paul-Ambroise Duquenne, Maha Elbayad, Artyom Kozhevnikov, Belen Alastruey, Pierre Andrews, Mariano Coria, Guillaume Couairon, Marta R. Costa-jussà, David Dale, Hady Elsahar, Kevin Heffernan, João Maria Janeiro, Tuan Tran, Christophe Ropers, Eduardo Sánchez, Robin San Roman, Alexandre Mourachko, Safiyyah Saleem, Holger Schwenk, 15 Dec 2024 (v2), *Large Concept Models: Language Modeling in a Sentence Representation Space*, https://arxiv.org/abs/2412.08821 https://github.com/facebookresearch/large_concept_model (Model operates at the sentence concept level, using SONAR sentence embeddings.)
2. Dr. Ashish Bamania, Dec 2024, *Meta's Large Concept Models (LCMs) Are Here To Challenge And Redefine LLMs: A deep dive into 'Large Concept Model', a novel language processing architecture and evaluating its performance against state-of-the-art LLMs*, https://levelup.gitconnected.com/metas-large-concept-models-lcms-are-here-to-challenge-and-redefine-llms-7f9778f88a87

3. Hussain Ahmad, Diksha Goel, 8 Jan 2025, *The Future of AI: Exploring the Potential of Large Concept Models*, https://arxiv.org/abs/2501.05487
4. Giuliano Liguori, Jan 2025, *Large Concept Models (LCM): A New Frontier in AI Beyond Token-Level Language Models*, https://www.linkedin.com/pulse/large-concept-models-lcm-new-frontier-ai-beyond-giuliano-liguori--dnj3f/
5. Jihoon Tack, Jack Lanchantin, Jane Yu, Andrew Cohen, Ilia Kulikov, Janice Lan, Shibo Hao, Yuandong Tian, Jason Weston, Xian Li, 12 Feb 2025, *LLM Pretraining with Continuous Concepts*, https://arxiv.org/abs/2502.08524
6. Vishal Rajput, Feb 2025, *Forget LLMs, It's Time For Large Concept Models (LCMs)*, https://medium.com/aiguys/forget-llms-its-time-for-large-concept-models-lcms-05b75fe43185
7. Vivek K. Tiwari, 2025, *Towards Practical Concept-Based Language Models: An Efficiency-Focused Implementation*, https://www.researchgate.net/profile/Vivek-Tiwari-41/publication/388753941_Towards_Practical_Concept-Based_Language_Models_An_Efficiency-Focused_Implementation/links/67a4bf86461fb56424cc6b62/Towards-Practical-Concept-Based-Language-Models-An-Efficiency-Focused-Implementation.pdf
8. Datacamp, Feb 21, 2025, *Large Concept Models: A Guide With Examples: Learn what large concept models are, how they differ from LLMs, and how their architecture leads to improvements in language processing*, https://www.datacamp.com/blog/large-concept-models
9. Mehul Gupta, Jan 5, 2025, *Meta Large Concept Models (LCM): End of LLMs? What are LCMs and how is LCM different from LLMs*, https://medium.com/data-science-in-your-pocket/meta-large-concept-models-lcm-end-of-llms-68cb0c5cd5cf
10. By AI Papers Academy, 3 January 2025, *Large Concept Models (LCMs) by Meta: The Era of AI After LLMs?* https://aipapersacademy.com/large-concept-models/
11. Andrea Viliotti, 20 Dec 2024, *Large Concept Model (LCM): a new paradigm for large-scale semantic reasoning in AI*, https://www.andreaviliotti.it/post/large-concept-model-lcm-a-new-paradigm-for-large-scale-semantic-reasoning-in-ai
12. Leadership in AI, January, 2025, *Meta's stunning LCM large concept models for artificial intelligence — they are thinking now!* https://www.youtub e.com/watch?v=u Z3HCw8ApQ,
13. Lance Eliot, Jan 06, 2025, *AI Is Breaking Free Of Token-Based LLMs By Upping The Ante To Large Concept Models That Devour Sentences And Adore Concepts*, https://www.forbes.com/sites/lanceeliot/2025/01/06/ai-is-breaking-free-of-token-based-llms-by-upping-the-ante-to-large-concept-models-that-devour-sentences-and-adore-concepts/
14. Zen the innovator, Jan 5, 2025, *Large Concept Models (LCMs)*, https://medium.com/@ThisIsMeIn360VR/large-concept-models-lcms-d59b86531ef6
15. Debabrata Pruseth, Jan 2025, *LCMs: Large Concept Models – The Path to AGI (Artificial General Intelligence) & The Future of AI Thinking*, https://debabratapruseth.com/lcms-large-concept-models-the-path-to-agi-the-future-of-ai-thinking/

16. Asif Razzaq, December 15, 2024, *Meta AI Proposes Large Concept Models (LCMs): A Semantic Leap Beyond Token-based Language Modeling*, https://www.marktechpost.com/2024/12/15/meta-ai-proposes-large-concept-models-lcms-a-semantic-leap-beyond-token-based-language-modeling/
17. Aniket Hingane, Dec 27, 2024, *Practical Advancements in AI: How Large Concept Models Are Redefining the Landscape of LLMs*, https://medium.com/@learn-simplified/practical-advancements-in-ai-how-large-concept-models-are-redefining-the-landscape-of-llms-b0220296458b
18. Siddhant Rai and Vizuara AI, Dec 30, 2024, *Large Concept models : Language Modeling in a Sentence Representation Space: Re-imagining the core principles behind representation generation in foundation model*, https://vizuara.substack.com/p/large-concept-models-language-modeling?
19. J Liao, R Xie, S Li, X Wang, X Sun, Z Kang, X He, 2025, *Multi-Grained Patch Training for Efficient LLM-based Recommendation*, https://hexiangnan.github.io/papers/sigir25-PatchRec.pdf
20. Ignacio de Gregorio, June 2025, *What If We Are All Wrong About AI? The contrarian bet by Meta, in plain English*, https://medium.com/@ignacio.de.gregorio.noblejas/what-if-we-are-all-wrong-about-ai-f33a3c64055c

Reasoning Tokens. Research papers on reasoning tokens as a type of higher-level reasoning method include:

1. Ignacio de Gregorio Noblejas, September 15, 2024, *OpenAI Launches o1. Here's All You Need to Know*, https://thetechoasis.beehiiv.com/p/openai-launches-o1-heres-need-know
2. Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, Yuandong Tian, 9 Dec 2024, *Training Large Language Models to Reason in a Continuous Latent Space*, https://arxiv.org/abs/2412.06769 (Performing reasoning in a model trained to operate in the embedding vector space, rather than more directly in the token space.)
3. Luyang Liu, Jonas Pfeiffer, Jiaxing Wu, Jun Xie, Arthur Szlam, 23 Dec 2024, *Deliberation in Latent Space via Differentiable Cache Augmentation*, https://arxiv.org/abs/2412.17747 (Augmenting the KV cache with reasoning information so that decoding will mimic multi-step reasoning with fewer tokens required for intermediate steps.)
4. Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, Vaishnavh Nagarajan, 21 Apr 2024 (v3), *Think before you speak: Training Language Models With Pause Tokens*, https://arxiv.org/abs/2310.02226 (Inserting extra "pause tokens" that trigger the LLM to perform extra reasoning during the decoding phase.)
5. Jacob Pfau, William Merrill, Samuel R. Bowman, 24 Apr 2024, *Let's Think Dot by Dot: Hidden Computation in Transformer Language Models*, https://arxiv.org/abs/2404.15758 (Use of dummy "filler tokens" similar to "pause tokens" or "reasoning tokens" to aid multi-step reasoning in decoding.)

6. Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, Noah D. Goodman, 18 Mar 2024 (v2), *Quiet-STaR: Language Models Can Teach Themselves to Think Before Speaking*, https://arxiv.org/abs/2403.09629 (Introduces answers between a start-of-thought and end-of-thought meta-token for reasoning.)
7. Lance Eliot, Dec 18, 2024, *Chain Of Continuous Thought Promises Mighty Boost For LLMs And Generative AI By Blowing Up The Fixation On Tokens*, https://www.forbes.com/sites/lanceeliot/2024/12/18/chain-of-continuous-thought-promises-mighty-boost-for-llms-and-generative-ai-by-blowing-up-the-fixation-on-tokens/
8. Xuan Shen, Yizhou Wang, Xiangxi Shi, Yanzhi Wang, Pu Zhao, Jiuxiang Gu, 31 Jan 2025, *Efficient Reasoning with Hidden Thinking*, https://arxiv.org/abs/2501.19201
9. DiJia Su, Hanlin Zhu, Yingchen Xu, Jiantao Jiao, Yuandong Tian, Qinqing Zheng, 5 Feb 2025. *Token Assorted: Mixing Latent and Text Tokens for Improved Language Model Reasoning*, https://arxiv.org/abs/2502.03275
10. Jim the AI Whisperer, Feb 2025, *I hacked Perplexity AI's full system prompt when I shared my own cognitive vulnerabilities with it: How I used my own scrambled brain to outwit Perplexity AI*, https://medium.com/the-generator/prompt-hacking-perplexity-ai-system-instructions-7aa6ee923060
11. Kongcheng Zhang, Qi Yao, Baisheng Lai, Jiaxing Huang, Wenkai Fang, Dacheng Tao, Mingli Song, Shunyu Liu, 19 Feb 2025, *Reasoning with Reinforced Functional Token Tuning*, https://arxiv.org/abs/2502.13389
12. Ziang Ye, Zhenru Zhang, Yang Zhang, Jianxin Ma, Junyang Lin, Fuli Feng, 19 Dec 2024, *Disentangling Reasoning Tokens and Boilerplate Tokens For Language Model Fine-tuning*, https://arxiv.org/abs/2412.14780
13. Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, Yulan He, 28 Feb 2025, *CODI: Compressing Chain-of-Thought into Continuous Space via Self-Distillation*, https://arxiv.org/abs/2502.21074
14. Guanghao Li, Wenhao Jiang, Mingfeng Chen, Yan Li, Hao Yu, Shuting Dong, Tao Ren, Ming Tang, Chun Yuan, 30 May 2025, *SCOUT: Teaching Pre-trained Language Models to Enhance Reasoning via Flow Chain-of-Thought*, https://arxiv.org/abs/2505.24181

Prompt Tuning. Research papers on prompt tuning or prefix tuning, which is adding concept tokens to prompts for easier processing, include:

1. IBM, 2024, *What is prompt-tuning?*, https://research.ibm.com/blog/what-is-ai-prompt-tuning
2. Abhinav Jain, Swarat Chaudhuri, Thomas Reps, Chris Jermaine, 24 May 2024, *Prompt Tuning Strikes Back: Customizing Foundation Models with Low-Rank Prompt Adaptation*, https://arxiv.org/abs/2405.15282
3. MohammadAli SadraeiJavaeri, Ehsaneddin Asgari, Alice Carolyn McHardy, Hamid Reza Rabiee, 7 Jun 2024, *SuperPos-Prompt: Enhancing Soft Prompt Tuning of Language Models with Superposition of Multi Token Embeddings*, https://arxiv.org/abs/2406.05279

4. Martin Wistuba, Prabhu Teja Sivaprasad, Lukas Balles, Giovanni Zappella, 5 Jun 2024, *Choice of PEFT Technique in Continual Learning: Prompt Tuning is Not All You Need*, https://arxiv.org/abs/2406.03216
5. Xuyang Wu, Zhiyuan Peng, Sravanthi Rajanala, Hsin-Tai Wu, Yi Fang, 31 May 2024, *Passage-specific Prompt Tuning for Passage Reranking in Question Answering with Large Language Models*, https://arxiv.org/abs/2405.20654
6. Wei Zhu, Aaron Xuxiang Tian, Congrui Yin, Yuan Ni, Xiaoling Wang, Guotong Xie, 7 Jun 2024 (v2), *IAPT: Instruction-Aware Prompt Tuning for Large Language Models*, https://arxiv.org/abs/2405.18203
7. Mengwei Xu, Wangsong Yin, Dongqi Cai, Rongjie Yi, Daliang Xu, Qipeng Wang, Bingyang Wu, Yihao Zhao, Chen Yang, Shihe Wang, Qiyang Zhang, Zhenyan Lu, Li Zhang, Shangguang Wang, Yuanchun Li, Yunxin Liu, Xin Jin, Xuanzhe Liu, 16 Jan 2024, *A Survey of Resource-efficient LLM and Multimodal Foundation Models*, https://arxiv.org/abs/2401.08092 Project: https://github.com/UbiquitousLearning/Efficient_Foundation_Model_Survey
8. Tianyu Ding, Tianyi Chen, Haidong Zhu, Jiachen Jiang, Yiqi Zhong, Jinxin Zhou, Guangzhi Wang, Zhihui Zhu, Ilya Zharkov, Luming Liang, 18 Apr 2024 (v2), *The Efficiency Spectrum of Large Language Models: An Algorithmic Survey*, https://arxiv.org/abs/2312.00678
9. M Xu, D Cai, W Yin, S Wang, X Jin, X Liu, 2024, *Resource-efficient Algorithms and Systems of Foundation Models: A Survey*, ACM Computing Surveys, https://dl.acm.org/doi/pdf/10.1145/3706418
10. Brian Lester, Rami Al-Rfou, and Noah Constant, 2021, *The Power of Scale for Parameter-Efficient Prompt Tuning. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, Association for Computational Linguistics, https://aclanthology.org/2021.emnlp-main.243/
11. Shreyansh Shah, Oct 18, 2023, *Prompt Tuning: A Powerful Technique for Adapting LLMs to New Tasks*, https://medium.com/@shahshreyansh20/prompt-tuning-a-powerful-technique-for-adapting-llms-to-new-tasks-6d6fd9b83557
12. Data Camp, May 19, 2024, *Understanding Prompt Tuning: Enhance Your Language Models with Precision*, https://www.datacamp.com/tutorial/understanding-prompt-tuning
13. Sergey Sedov, Sumanth Bharadwaj Hachalli Karanam, Venu Gopal Kadamba, 24 Dec 2024, *Exploring Embedding Priors in Prompt-Tuning for Improved Interpretability and Control*, https://arxiv.org/abs/2412.18582
14. Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, Jie Tang, 20 Mar 2022 (v3), *P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks*, Proceedings of the 60th Annual Meeting of the Association of Computational Linguistics, 2022, https://arxiv.org/abs/2110.07602 https://github.com/THUDM/P-tuning-v2 (Extends prompt tuning with extra soft prompt tokens at every layer, not just at the start of the input.)
15. Haowei Zhu, Fangyuan Zhang, Rui Qin, Tianxiang Pan, Junhai Yong, Bin Wang, 24 Dec 2024 (v2), *Semantic Hierarchical Prompt Tuning for Parameter-Efficient Fine-Tuning*, https://arxiv.org/abs/2412.16956

16. Xiang Lisa Li, Percy Liang, 1 Jan 2021, *Prefix-Tuning: Optimizing Continuous Prompts for Generation*, https://arxiv.org/abs/2101.00190 (Precursor to prompt tuning.)
17. Andrea Matarazzo, Riccardo Torlone, 3 Jan 2025, *A Survey on Large Language Models with some Insights on their Capabilities and Limitations*, https://arxiv.org/abs/2501.04040 (Broad survey with many LLM topics covered from history to architectures to optimizations.)
18. Qi Sun, Edoardo Cetin, Yujin Tang, 14 Jan 2025 (v2), *Transformer2: Self-adaptive LLMs*, https://arxiv.org/abs/2501.06252 (Using a vector to fine-tune dynamically.)
19. Liu Yang, Ziqian Lin, Kangwook Lee, Dimitris Papailiopoulos, Robert Nowak, 16 Jan 2025, *Task Vectors in In-Context Learning: Emergence, Formation, and Benefit*, https://arxiv.org/abs/2501.09240
20. Dan Zhang, Tao Feng, Lilong Xue, Yuandong Wang, Yuxiao Dong, Jie Tang, 23 Jan 2025, *Parameter-Efficient Fine-Tuning for Foundation Models*, https://arxiv.org/abs/2501.13787
21. X Li, C Jiang, Nov 2024, *Optimizing Prompt Engineering Methods for Enhanced Logical Reasoning in Transformer Models*, RMEL '24, November 4–7, 2024, Hangzhou, China, https://www.researchgate.net/profile/Xiaoyan-Li-42/publication/389182048_Optimizing_Prompt_Engineering_Methods_for_Enhanced_Logical_Reasoning_in_Transformer_Models/links/67b82fa9461fb56424e3fc72/Optimizing-Prompt-Engineering-Methods-for-Enhanced-Logical-Reasoning-in-Transformer-Models.pdf https://github.com/xiaoyanLi629/RMELS2024

11. Reading & Writing

"If you don't have time to read,

you don't have the time to write."

— Stephen King.

AI Reading and Writing

Humans and AI engines share a strange commonality in how they communicate. As a human with the ability to both read and write, there are two somewhat related facts of life:

- It's easier to read than to write.
- It's easier to understand a foreign language than to speak it.

Weirdly, this turns out to also be true for AI engines. The coders for AI engines have done a lot of work in both areas, but reading remains easier than writing.

In the olden days, which is 2017 in AI, the LLM engines based on the "Transformer" architecture (discovered by Google) had two explicit pieces, called the "encoder" and "decoder" components.

The idea was basically:

- Encoders — reading
- Decoders — writing

The encoder would take its input text and "encode" it into some internal vectors, which are used to "understand" the text (i.e., reading). The decoder would then use these internal numbers to emit new numbers that represented words that it was outputting (i.e., writing).

However, that combined encoder-decoder architecture didn't last long, and the encoder was found to be mostly redundant, since it was doing the computations that were very similar to the decoder, only it wasn't actually decoding anything. Confused? You're like most AI engineers, if so. Anyway, you could do both the reading and writing inside the decoder, by doing a trick called a "prefill" phase.

No need for half the brain.

Hence, the encoder was removed from AI engines and only the decoder was left. This was called a "decoder-only" architecture and was used as early as the GPT-2 model. The newer idea is effectively:

- Prefill — reading
- Decoding — writing

Prefill is an algorithm that runs in the decoder, so these two phases are running in the same GPU chips (well, no, but let's assume that for now). The decoder in the newer decoder-only architecture is similar to that used in the encoder-decoder early version, but obviously doesn't need the parts that used to accept data from the encoder (which no longer exists).

Have we lost something by not having an encoder? It's certainly possible that some insight is missed by treating reading like it's a subset of writing, rather than trying to fully understand first. However, we gained: speed.

We need a lot more speed to do advanced data processing, such as voice or video analysis. It's all just data to an AI engine, but there's much more data in audio and video.

The way it works for speech models:

- Understanding speech — "reading" of voice numbers.
- Speaking — "writing" of audio data (i.e., decoding).

Again, there's not usually an encoder for voice models. But don't worry, because the poor old encoder component didn't get completely thrown into landfill. Encoder-decoder architectures are still used for some two-phase type operations like foreign language translation, where we're reading in one language and writing in another. In these cases, the reading is very different from the writing, so having two separate components works well.

Prefill for Reading

The way that prefill works doesn't really make sense compared to human reading. The AI engine looks at every word in its input prompt and processes them all at once, in parallel. The reason it can do this is that this phase does not output anything, and is really a preparatory phase before writing.

What's it prefilling?

The reason it's called "prefill" is that it fills in the "KV caches" for the input tokens (e.g., the user's question). These are extra numbers used by the second phase, the decoding phase, to figure out which new words to output.

Humans can actually do this type of parallel reading that AI does in prefill, but it's not the natural way that we read. If you train yourself, you can actually start reading a novel by scanning multiple lines at once, across the page, like you are ingesting groups of words as they cross your visual field. It's hard to learn, but it can be done (by humans).

GPU chips are naturally parallel, and AI engines can read all the words of a document in parallel without even blinking. The LLM ingests all of the words at once, and then cross-analyzes them to find relationships between the positions with all the other words.

This is all done in parallel in the prefill phase, analyzing all of the inputs, before it starts emitting words in the decoding phase.

Decoding for Writing

If you read about LLM theory, you will discover that the way that an LLM writes text in the "decoding" phase is to output one word at a time. It only looks backwards to any of the previous words, and figures how the probabilities for each of the possible next words, and then chooses the best one. The decoder is actually "masked" so that it doesn't look ahead, and anyway there's nothing up ahead anyway, and it only looks backwards, and never backtracks.

This is total baloney.

It may have been true in the very earliest theories of neural networks, but modern LLMs do a whole smash of other stuff in their decoding algorithm. I mean, you can't write a good piece of prose based only on past words, especially if you only ever look one word ahead.

Humans do writing by thinking forward many words, revising, going back and changing, and so on. Hence, in reality, your LLM does a lot of coding tricks to decide on the best words to write:

- Trying multiple possible word sequences (in parallel).
- Backtracking and restarting phrases and sections.

There are some fancy names for some of these algorithms, which mostly run in parallel on GPU chips:

- Beam search
- Speculative decoding
- Multi-Token Prediction (MTP)

Beam search is about running ahead a few words, checking whether it's any good, and then backtracking. Speculative decoding is about trying multiple possible new phrases to write, all calculated in parallel.

Multi-token prediction is about emitting more than one word at a time, usually also combined with beam search or spec dec (that's short for speculative decoding).

Humans struggle to do this, although there are probably some people who can speak more than one word at a time in parallel. Strangely, an image of Elon Musk just popped into my subconscious brain. But I digress.

All of these decoding methods are done in parallel in the "forward pass" of the decoding algorithm. This is like the "fast thinking" mode of writing, and are not really doing advanced rational logic or any higher-level analysis of the words.

In some of the fancier models, there's also a second "slow" method of revising the output, whereby the LLM can itself make changes to what it's written, or possibly have an even bigger model check over its work. That's like your English teacher peeking over your shoulder while you compose your persuasive writing essay.

Reading Limitations

Reading doesn't mean understanding. The LLM is reading the words as "tokens" that represent the text. There's also tokens for images, audio, and video, which are more complicated, but still the same basic architecture. A voice LLM sees your speech patterns as just numbers, too.

Words are actually a higher level abstraction than the numbers in voice or images, so they should be easier to interpret. However, the problem with reading words is that all the LLM knows about is words. The meaning of words is often obscured, and there are problems with simple things like representing numbers as words, which makes it hard to do basic arithmetic.

Understanding of words also implies a lot more things than just text patterns. There's hidden meanings in all sorts of ways using rules that humans know, or have learned. For example, how do you know that it's a joke that we "drive on parkways and park on driveways"? LLMs are bad at jokes or sarcasm without special training.

Common sense is also hard to explain in words. It's like there's a mapping between pairs or groups of words. For example, cats are "svelte" but not dogs, or we "sit on sofas not tables", but there are all sorts of exceptions to all those so-called rules.

Our world is difficult to describe in words, because its three-dimensional. Babies learn that if they try to crawl under a desk, they might smack their head, because one failure gives them a powerful learning signal involving lots of noise and cuddles. How does an AI know what three dimensions are? It's not in the word patterns.

The fourth dimension, time, is also tricky. Food appears on a plate and then disappears, not the other way around, and yet both ways it would have the same words. They call this "temporal reasoning" and, since it has a fancy name, that means it's a research area with lots of difficulty and plenty of obscure research papers. AI models are not good with time. In contrast, a human child moves through time intrinsically, and comes to understand that idea at a very deep level.

References

Prefill Phase. Research papers on "prefill" (i.e., reading), most of which related to performance improvement of the prefill phase:

1. Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal, Jeff Dean, 9 Nov 2022, *Efficiently Scaling Transformer Inference*, https://arxiv.org/abs/2211.05102 (The paper that seems to have coined the term "prefill" and examines some aspects of prefill vs decoding phases in optimization.)
2. Yinmin Zhong, Shengyu Liu, Junda Chen, Jianbo Hu, Yibo Zhu, Xuanzhe Liu, Xin Jin, Hao Zhang, 19 Mar 2024 (v2), *DistServe: Disaggregating Prefill and Decoding for Goodput-optimized Large Language Model Serving*, https://arxiv.org/abs/2401.09670 (Optimizing LLMs differently in the prefill and decoding phases.)
3. Juntao Zhao, Borui Wan, Yanghua Peng, Haibin Lin, Chuan Wu, 2 Mar 2024, *LLM-PQ: Serving LLM on Heterogeneous Clusters with Phase-Aware Partition and Adaptive Quantization*, https://arxiv.org/abs/2403.01136 (Deployment of LLMs on heterogenous GPUs and also differences between the two phases of decoder-only Transformers: prefill and decoding computations.)
4. Cunchen Hu, Heyang Huang, Liangliang Xu, Xusheng Chen, Jiang Xu, Shuang Chen, Hao Feng, Chenxi Wang, Sa Wang, Yungang Bao, Ninghui Sun, Yizhou Shan, 20 Jan 2024, *Inference without Interference: Disaggregate LLM Inference for Mixed Downstream Workloads*, https://arxiv.org/abs/2401.11181 (Separating the prefill and decoding phases for optimization.)
5. Siyan Zhao, Daniel Israel, Guy Van den Broeck, Aditya Grover, 15 Apr 2024, *Prepacking: A Simple Method for Fast Prefilling and Increased Throughput in Large Language Models*, https://arxiv.org/abs/2404.09529 Code: https://github.com/siyan-zhao/prepacking (Optimizes prefill KV cache computations by batching multiple query prefill phases together via packing, since prefill token sequence lengths are fully known, and further combined with simple modifications to positional encoding and masking to avoid cross-query attention.)
6. VLLM, 2024, *Performance and Tuning: Chunked Prefill*, https://docs.vllm.ai/en/v0.4.2/models/performance.html
7. Schwinn Saereesitthipitak, Ashish Rao, Cathy Zhou, William Li, 2024, *Prophet: An LLM Inference Engine Optimized For Head-of-Line Blocking*, https://www.scs.stanford.edu/24sp-cs244b/projects/Prophet_An_LLM_Inference_Engine_Optimized_For_Head_of_Line_Blocking.pdf (Faster inference serving via iterative scheduling, separating prefill and decoding phase computations for batching, using priority-based schedulers with preemption, and controling transfer of KV caches from prefill to decoders.)

8. Daliang Xu, Hao Zhang, Liming Yang, Ruiqi Liu, Mengwei Xu, and Xuanzhe Liu, 11 June 2024, *WiP: Efficient LLM Prefilling with Mobile NPU*, EdgeFM '24: Proceedings of the Workshop on Edge and Mobile Foundation Models, June 2024, Pages 33 - 35, https://doi.org/10.1145/3662006.3662066 https://dl.acm.org/doi/abs/10.1145/3662006.3662066 PDF: https://dl.acm.org/doi/pdf/10.1145/3662006.3662066 (Faster NPU prefill via chunked prefilling using sequences of tokens, along with INT8 NPU quantization that is aware of outliers and offloads FP32 calculations from NPU back to CPU.)
9. Cunchen Hu, Heyang Huang, Junhao Hu, Jiang Xu, Xusheng Chen, Tao Xie, Chenxi Wang, Sa Wang, Yungang Bao, Ninghui Sun, Yizhou Shan, 26 Jun 2024 (v2), *MemServe: Context Caching for Disaggregated LLM Serving with Elastic Memory Pool*, https://arxiv.org/abs/2406.17565 (Combined session-based prefix KV caching with disaggregation of prefill and decoding phases.)
10. Ruoyu Qin, Zheming Li, Weiran He, Mingxing Zhang, Yongwei Wu, Weimin Zheng, Xinran Xu, 2 Jul 2024 (v2), *Mooncake: A KVCache-centric Disaggregated Architecture for LLM Serving*, https://arxiv.org/abs/2407.00079 Code: https://github.com/kvcache-ai/Mooncake (Disaggregates prefill and decoding phases for scheduling, with chunked prefill, while managing the KV cache.)
11. Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, Lili Qiu, 2 Jul 2024, *MInference 1.0: Accelerating Pre-filling for Long-Context LLMs via Dynamic Sparse Attention*, https://arxiv.org/abs/2407.02490 Code: https://aka.ms/MInference
12. Daliang Xu, Hao Zhang, Liming Yang, Ruiqi Liu, Gang Huang, Mengwei Xu, Xuanzhe Liu, 8 Jul 2024, *Empowering 1000 tokens/second on-device LLM prefilling with mllm-NPU*, https://arxiv.org/abs/2407.05858
13. Baolin Li, Yankai Jiang, Vijay Gadepally, Devesh Tiwari, 17 Jul 2024, *LLM Inference Serving: Survey of Recent Advances and Opportunities*, https://arxiv.org/abs/2407.12391
14. Runheng Liu, Xingchen Xiao, Heyan Huang, Zewen Chi, Zhijing Wu, 7 May 2024, *FlashBack:Efficient Retrieval-Augmented Language Modeling for Long Context Inference*, https://arxiv.org/abs/2405.04065 (Optimize RAG by appending rather than prepending documents, and modifying the attention for improvements in KV caching, by shimming or replacing some of the CUDA GPU low-level memory management APIs to avoid the need to rewrite kernels with extra higher-level memory management code.)
15. Haifeng Qian, Sujan Kumar Gonugondla, Sungsoo Ha, Mingyue Shang, Sanjay Krishna Gouda, Ramesh Nallapati, Sudipta Sengupta, Xiaofei Ma, Anoop Deoras, 24 Apr 2024, *BASS: Batched Attention-optimized Speculative Sampling*, https://arxiv.org/abs/2404.15778 (Optimizes batched multi-query use of speculative decoding with consideration of GPU utilization in prefill and decoding phases.)

16. Yibo Jin, Tao Wang, Huimin Lin, Mingyang Song, Peiyang Li, Yipeng Ma, Yicheng Shan, Zhengfan Yuan, Cailong Li, Yajing Sun, Tiandeng Wu, Xing Chu, Ruizhi Huan, Li Ma, Xiao You, Wenting Zhou, Yunpeng Ye, Wen Liu, Xiangkun Xu, Yongsheng Zhang, Tiantian Dong, Jiawei Zhu, Zhe Wang, Xijian Ju, Jianxun Song, Haoliang Cheng, Xiaojing Li, Jiandong Ding, Hefei Guo, Zhengyong Zhang, 15 Aug 2024, *P/D-Serve: Serving Disaggregated Large Language Model at Scale*, https://arxiv.org/abs/2408.08147 (Comprehensive serving system addressing disaggregated prefill and KV cache transfer with RDMA.)
17. Junlin Lv, Yuan Feng, Xike Xie, Xin Jia, Qirong Peng, Guiming Xie, 19 Sep 2024, *CritiPrefill: A Segment-wise Criticality-based Approach for Prefilling Acceleration in LLMs*, https://arxiv.org/abs/2409.12490
18. Shuowei Jin, Xueshen Liu, Qingzhao Zhang, Z. Morley Mao, 4 Oct 2024, *Compute Or Load KV Cache? Why Not Both?* https://arxiv.org/abs/2410.03065
19. Aurick Qiao, Zhewei Yao, Samyam Rajbhandari, Yuxiong He, 4 Oct 2024, *SwiftKV: Fast Prefill-Optimized Inference with Knowledge-Preserving Model Transformation*, https://arxiv.org/abs/2410.03960
20. Maxwell Horton, Qingqing Cao, Chenfan Sun, Yanzi Jin, Sachin Mehta, Mohammad Rastegari, Moin Nabi, 10 Oct 2024, *KV Prediction for Improved Time to First Token*, https://arxiv.org/abs/2410.08391 https://github.com/apple/corenet/tree/main/projects/kv-prediction (Small model creates an approximation of the KV cache for use by a larger model.)
21. Amy Yang, Jingyi Yang, Aya Ibrahim, Xinfeng Xie, Bangsheng Tang, Grigory Sizov, Jongsoo Park, Jianyu Huang, 4 Nov 2024, *Context Parallelism for Scalable Million-Token Inference*, https://arxiv.org/abs/2411.01783
22. Gursimran Singh, Xinglu Wang, Ivan Hu, Timothy Yu, Linzi Xing, Wei Jiang, Zhefeng Wang, Xiaolong Bai, Yi Li, Ying Xiong, Yong Zhang, Zhenan Fan, 25 Dec 2024, *Efficiently serving large multimedia models using EPD Disaggregation*, https://arxiv.org/abs/2501.05460 (Disaggregation of three steps: encoding, prefill, and decoding.)
23. Daliang Xu, Hao Zhang, Liming Yang, Ruiqi Liu, Gang Huang, Mengwei Xu, and Xuanzhe Liu, 2025, *Fast On-device LLM Inference with NPUs*, Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1 (ASPLOS '25), Association for Computing Machinery, New York, NY, USA, 445–462, https://doi.org/10.1145/3669940.3707239 https://dl.acm.org/doi/abs/10.1145/3669940.3707239 (Offloading chunked prefill computations to NPUs.)
24. Xunhao Lai, Jianqiao Lu, Yao Luo, Yiyuan Ma, Xun Zhou, 28 Feb 2025, *FlexPrefill: A Context-Aware Sparse Attention Mechanism for Efficient Long-Sequence Inference*, https://arxiv.org/abs/2502.20766 (Prefill optimization that dynamically applies different attention patterns, including sparse attention, for KV computations, based on the input query.)
25. Yunkai Liang, Zhangyu Chen, Pengfei Zuo, Zhi Zhou, Xu Chen, Zhou Yu, 26 Mar 2025, *Injecting Adrenaline into LLM Serving: Boosting Resource Utilization and Throughput via Attention Disaggregation*, https://arxiv.org/abs/2503.20552

26. Rajeshkumar Bambhaniya, Abhimanyu ; Wu, Hanjiang ; Subramanian, Suvinay ; Srinivasan, Sudarshan ; Kundu, Souvik ; Yazdanbakhsh, Amir ; Elavazhagan, Midhilesh ; Kumar, Madhu ; Krishna, Tushar, April 2025, *Understanding and Optimizing Multi-Stage AI Inference Pipelines*, https://ui.adsabs.harvard.edu/abs/2025arXiv250409775R/abstract https://arxiv.org/abs/2504.09775
27. Kuntai Du, Bowen Wang, Chen Zhang, Yiming Cheng, Qing Lan, Hejian Sang, Yihua Cheng, Jiayi Yao, Xiaoxuan Liu, Yifan Qiao, Ion Stoica, Junchen Jiang, 12 May 2025, *PrefillOnly: An Inference Engine for Prefill-only Workloads in Large Language Model Applications*, https://arxiv.org/abs/2505.07203

Beam Search. Research papers on beam search decoding algorithms, which look ahead in words and then backtrack, include:

1. Xiaohui Wang, Ying Xiong, Yang Wei, Mingxuan Wang, Lei Li Apr 2021, *LightSeq: A High Performance Inference Library for Transformers*, https://arxiv.org/pdf/2010.13887.pdf
2. James Briggs Feb 25, 2021, *The Three Decoding Methods For NLP*, Towards Data Science https://towardsdatascience.com/the-three-decoding-methods-for-nlp-23ca59cb1e9d
3. GC Garbacea, 2023, *Neural Language Generation for Content Adaptation: Explainable, Efficient Low-Resource Text Simplification and Evaluation*, Ph.D. thesis, Computer Science and Engineering, University of Michigan, https://deepblue.lib.umich.edu/bitstream/handle/2027.42/178028/garbacea_1.pdf?sequence=1 (Broad thesis with sections on beam search decoding optimizations and AI safety issues such as bias.)
4. Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould, 2017, *Guided open vocabulary image captioning with constrained beam search*, Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 936–945, https://arxiv.org/abs/1612.00576
5. Chris Hokamp and Qun Liu, 2017, *Lexically constrained decoding for sequence generation using grid beam search*, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1535–1546, https://arxiv.org/abs/1704.07138
6. G Keren, Feb 2023, *A Token-Wise Beam Search Algorithm for RNN-T*, arXiv preprint arXiv:2302.14357, https://arxiv.org/abs/2302.14357
7. Gian Wiher, Clara Meister, Ryan Cotterell, Mar 2022, *On Decoding Strategies for Neural Text Generators*, https://arxiv.org/abs/2203.15721 (An evaluation of a variety of decoding algorithms including beam search, top-k, and top-p.)
8. Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra, 2016, *Diverse beam search: Decoding diverse solutions from neural sequence models*, CoRR, abs/1610.02424, https://arxiv.org/abs/1610.02424 (An algorithm variant called "diverse beam search" decoding.)
9. Ilya Sutskever, Oriol Vinyals, and Quoc V Le, 2014, *Sequence to sequence learning with neural networks*, arXiv preprint arXiv:1409.3215, https://arxiv.org/abs/1409.3215 (Early paper using a kind of beam search decoding and top-k decoding.)

10. Kenton Murray, David Chiang, Aug 2018, *Correcting Length Bias in Neural Machine Translation*, https://arxiv.org/abs/1808.10006 (Brevity problems in beam search decoding.)
11. Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, Chunan Shi, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, Zhihao Jia, 2024, *SpecInfer: Accelerating Large Language Model Serving with Tree-based Speculative Inference and Verification*, ASPLOS'24: Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3, April 2024, Pages 932–949, https://doi.org/10.1145/3620666.3651335 https://dl.acm.org/doi/abs/10.1145/3620666.3651335 Code: https://github.com/flexflow/FlexFlow/
12. Jared Lichtarge, Christopher Alberti, Shankar Kumar, Noam Shazeer, and Niki Parmar, 2018, *Weakly supervised grammatical error correction using iterative decoding*, CoRR, abs/1811.01710, https://arxiv.org/abs/1811.01710 (Beam search decoding with a high threshold to emit corrections.)
13. Jindrich Libovicky, Jindrich Helcl, Marek Tlusty, Ondrej Bojar, and Pavel Pecina, 2016, *CUNI system for WMT16 automatic post-editing and multimodal translation tasks*, Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers, pages 646–654, Berlin, Germany, https://arxiv.org/abs/1606.07481 (Post-editing of machine translation.)
14. Daniel Dahlmeier, Hwee Tou Ng, 2012, *A Beam-Search Decoder for Grammatical Error Correction*, Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 568–578, Jeju Island, Korea, 12–14 July 2012, https://aclanthology.org/D12-1052.pdf
15. Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra, 2018, *Diverse beam search for improved description of complex scenes*, In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pages 7371–7379, AAAI Press, https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17329
16. Tinghui Zhu, Kai Zhang, Jian Xie, Yu Su, 4 Feb 2024 (v2), *Deductive Beam Search: Decoding Deducible Rationale for Chain-of-Thought Reasoning*, https://arxiv.org/abs/2401.17686
17. Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Dragomir Radev, Yejin Choi, and Noah A. Smith, 2024, *A Call for Clarity in Beam Search: How It Works and When It Stops*, Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 77–90, Torino, Italia. ELRA and ICCL, https://aclanthology.org/2024.lrec-main.7/ https://aclanthology.org/2024.lrec-main.7.pdf

18. Zongyue Qin, Zifan He, Neha Prakriya, Jason Cong, Yizhou Sun, 25 Sep 2024, *Dynamic-Width Speculative Beam Decoding for Efficient LLM Inference*, https://arxiv.org/abs/2409.16560
19. Shixiaowei02, Oct 2024, *TensorRT-LLM 0.13.0 Release Latest*, https://github.com/NVIDIA/TensorRT-LLM/releases/tag/v0.13.0
20. Rongxiang Wang and Felix Xiaozhu Lin, 2024, *Turbocharge Speech Understanding with Pilot Inference*, Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24). Association for Computing Machinery, New York, NY, USA, 1299–1313, https://doi.org/10.1145/3636534.3690694 https://dl.acm.org/doi/abs/10.1145/3636534.3690694 https://dl.acm.org/doi/pdf/10.1145/3636534.3690694 ("Pilot inference" is a specialized mix of caching, computation reuse, and backtracking in beam search for speech understanding, and is somewhat related to speculative decoding, and similar to continual inference for processing a stream.)

Decoding Algorithms. Research papers on decoding algorithms in general:

1. Xiangjue Dong, Maria Teleki, James Caverlee, 18 Dec 2024, *A Survey on LLM Inference-Time Self-Improvement*, https://arxiv.org/abs/2412.14352 https://github.com/dongxiangjue/Awesome-LLM-Self-Improvement (Broad survey of reasoning improvement methods from multi-step inference to RALM to decoding algorithms.)
2. Chufan Shi, Haoran Yang, Deng Cai, Zhisong Zhang, Yifan Wang, Yujiu Yang, Wai Lam, 10 Feb 2024, *A Thorough Examination of Decoding Methods in the Era of LLMs*, https://arxiv.org/abs/2402.06925 (Evaluates a number of decoding algorithms with several 7B models including Llama2-7B, and also with 4-bit and 8-bit quantization.)
3. Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, Zaid Harchaoui, 24 Jun 2024, *From Decoding to Meta-Generation: Inference-time Algorithms for Large Language Models*, https://arxiv.org/abs/2406.16838 (Survey and theoretical analysis of many different decoding algorithms, along with various ways to speed them up such as speculative decoding and KV caches.)
4. Haoran Wang, Kai Shu, Jan 2025, *Make Every Token Count: A Systematic Survey on Decoding Methods for Foundation Model*, https://www.researchgate.net/profile/Haoran-Wang-96/publication/387703971_Make_Every_Token_Count_A_Systematic_Survey_on_Decoding_Methods_for_Foundation_Models/links/67784c8ce74ca64e1f49eb15/Make-Every-Token-Count-A-Systematic-Survey-on-Decoding-Methods-for-Foundation-Models.pdf https://github.com/wang2226/Awesome-LLM-Decoding
5. Edward Beeching, Lewis Tunstall, Sasha Rush Dec 16, 2024, *Scaling Test Time Compute with Open Source Models*, https://huggingface.co/spaces/HuggingFaceH4/blogpost-scaling-test-time-compute

6. Maciej Besta, Julia Barth, Eric Schreiber, Ales Kubicek, Afonso Catarino, Robert Gerstenberger, Piotr Nyczyk, Patrick Iff, Yueling Li, Sam Houliston, Tomasz Sternal, Marcin Copik, Grzegorz Kwaśniewski, Jürgen Müller, Łukasz Flis, Hannes Eberhard, Hubert Niewiadomski, Torsten Hoefler, 23 Jan 2025 (v3), *Reasoning Language Models: A Blueprint*, https://arxiv.org/abs/2501.11223 (Survey and blueprint for how to build a Large Reasoning Model.)
7. Wendi Cui, Jiaxin Zhang, Zhuohang Li, Hao Sun, Damien Lopez, Kamalika Das, Bradley A. Malin, Sricharan Kumar, 26 Feb 2025, *Automatic Prompt Optimization via Heuristic Search: A Survey*, https://arxiv.org/abs/2502.18746 (Survey of auto prompting, from basic LLM enhancements to some methods quite similar to RALM and TALM.)
8. Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip H.S. Torr, Salman Khan, Fahad Shahbaz Khan, 28 Feb 2025, *LLM Post-Training: A Deep Dive into Reasoning Large Language Models*, https://arxiv.org/abs/2502.21321 https://github.com/mbzuai-oryx/Awesome-LLM-Post-training

Tree Decoding. Research papers on "tree decoding," which is the general idea of trying multiple word output pathways, and then backtracking:

1. Ziyu Wan, Xidong Feng, Muning Wen, Stephen Marcus Mcaleer, Ying Wen, Weinan Zhang, Jun Wang, July 2024, *AlphaZero-Like Tree-Search can Guide Large Language Model Decoding and Training*, Proceedings of the 41st International Conference on Machine Learning, PMLR 235:49890-49920, 2024, https://proceedings.mlr.press/v235/wan24c.html PDF: https://raw.githubusercontent.com/mlresearch/v235/main/assets/wan24c/wan24c.pdf
2. Xiangxiang Gao, Weisheng Xie, Yiwei Xiang, Feng Ji, 17 Dec 2024, *Falcon: Faster and Parallel Inference of Large Language Models through Enhanced Semi-Autoregressive Drafting and Custom-Designed Decoding Tree*, https://arxiv.org/abs/2412.12639
3. Yangchao Wu, Zongyue Qin, Alex Wong, Stefano Soatto, 20 May 2025, *STree: Speculative Tree Decoding for Hybrid State-Space Models*, https://arxiv.org/abs/2505.14969
4. Xuezhi Wang, Denny Zhou, 23 May 2024 (v2), *Chain-of-Thought Reasoning Without Prompting*, https://arxiv.org/abs/2402.10200 ("CoT decoding" is examining the alternative paths in the decoding algorithm, which is somewhat similar to Chain-of-Thought reasoning.)
5. Xiangjue Dong, Maria Teleki, James Caverlee, 18 Dec 2024, *A Survey on LLM Inference-Time Self-Improvement*, https://arxiv.org/abs/2412.14352 https://github.com/dongxiangjue/Awesome-LLM-Self-Improvement (Broad survey of reasoning improvement methods from multi-step inference to RALM to decoding algorithms.)

6. Xidong Feng, Ziyu Wan, Muning Wen, Ying Wen, Weinan Zhang, and Jun Wang, 2023, *Alphazero-like tree-search can guide large language model decoding and training*, NeurIPS 2023 Foundation Models for Decision Making Workshop, https://arxiv.org/abs/2309.17179
7. Zongyue Qin, Zifan He, Neha Prakriya, Jason Cong, Yizhou Sun, 25 Sep 2024, *Dynamic-Width Speculative Beam Decoding for Efficient LLM Inference*, https://arxiv.org/abs/2409.16560
8. Penghui Yang, Cunxiao Du, Fengzhuo Zhang, Haonan Wang, Tianyu Pang, Chao Du, Bo An, 24 Feb 2025, *LongSpec: Long-Context Speculative Decoding with Efficient Drafting and Verification*, https://arxiv.org/abs/2502.17421 https://github.com/sail-sg/LongSpec
9. Yifu Ding, Wentao Jiang, Shunyu Liu, Yongcheng Jing, Jinyang Guo, Yingjie Wang, Jing Zhang, Zengmao Wang, Ziwei Liu, Bo Du, Xianglong Liu, Dacheng Tao, 27 Feb 2025 (v2), *Dynamic Parallel Tree Search for Efficient LLM Reasoning*, https://arxiv.org/abs/2502.16235
10. Yangchao Wu, Zongyue Qin, Alex Wong, Stefano Soatto, 20 May 2025, *STree: Speculative Tree Decoding for Hybrid State-Space Models*, https://arxiv.org/abs/2505.14969
11. Yuhao Shen, Junyi Shen, Quan Kong, Tianyu Liu, Yao Lu, Cong Wang, 16 May 2025, *Speculative Decoding via Hybrid Drafting and Rollback-Aware Branch Parallelism*, https://arxiv.org/abs/2506.01979

12. 'Rithmetic

"Math skills are a powerful indicator of future success."

— Bill Gates.

Everyone Loves Math

Humans are great at arithmetic! Right from birth, we can add up 10-digit numbers in our heads, without even using our fingers and toes.

Umm, no.

Weirdly, AI is bad at arithmetic, too! Maybe it's not so weird, as we've been discussing the parallels between brains and LLMs, but it definitely seems funny.

Surely, computers can do basic sums very easily?

Yes, indeed. Your average laptop computer is running a CPU that has a clock speed of about 1 or 2 Gigahertz, so, let's say it's about 2 billion operations per second. And the most basic computer instructions are addition and multiplication operations. In fact, those are the simplest of arithmetic operations, but there are all sorts of other maths stuff, too.

If you have an older laptop, it can only do "32-bit" operations, which is numbers up to about 2.7 billion, which are at most 10-digit numbers. But most modern laptops are now using "64-bit" CPUs, which allows up to 19-digit numbers. To help you win *Trivial Pursuit* next Saturday at your AI club meet, note that a trillion has a one followed by 12 zeros for a total of 13 digits, a quadrillion has 16 digits, and a quintillion has 19 digits.

So, your basic laptop from Best Buy that you can buy on a credit card for less than the cost of a pet parrot, can do perfect arithmetic in massive numbers up to quintillions using at least 19 digits, and do so with 100% accuracy, over and over, billions of times per second.

An LLM can't.

Every LLM is great at detecting patterns and knows about words. In fact, it usually treats a number as a word, and a 10-digit number will likely be 10 words, one digit each. So, the LLM will have seen "sentences" like "1 + 1 = 2" in its training, so it'll nail those ones. Unfortunately, it's expensive to train it to recognize all the different "words" up to a billion, let alone a quintillion.

Infinity is a little problematic.

If you're thinking that it's smart enough to figure it out from the other sums it's been trained on, well, I beg to differ. LLMs have been historically poor at "generalization" of patterns. For example, if it knows the pattern "1+1=2" that doesn't mean that it knows the pattern:

"1,000,000,000+1,000,000,000=2,000,000,000"

Not at all, unless it's seen that exact pattern in some other training document. A human child does this extrapolation of patterns innately, but your average multi-billion-dollar LLM struggles at this very hard stuff.

Okay, before every CEO of an AI company sends me a nasty cease-and-desist letter pointing out that their trillion-parameter LLM can do sums just fine: yes, they've figured it out now. But for years, LLMs were bad at basic arithmetic, let alone more advanced math, and there were research papers published all about the problems with basic arithmetic and complex math. So many.

And, honestly, LLMs are still poor at arithmetic, but there's a couple of workarounds. One is "reasoning models" that can do multiple steps to solve a problem, which we discussed in a separate chapter.

However, although I don't know for sure how it does it in the big LLMs, I'm betting it's the second one:

Tools.

If you submit a couple of 10-digit random numbers to your favorite AI assistant and ask it to multiply them, it's probably not really the neural networks doing the number-crunching work. More likely, it's a special "tool" known more formally as a "calculator," which is linked up to the LLM itself.

So, behind the scenes, the LLM notices that your request looks like it needs a calculator, because it's good at recognizing patterns like that, and then calls the internal calculator tool without telling you.

The calculator tool is not using neural networks, so it's literally a billion times faster than the LLM, and you won't notice any delay from the extra step. Incidentally, I'm referring to the grunt-work calculator tool inside the AI servers next to the LLM code, not the graphical one that you can launch on your laptop or smartphone.

After this extra hop, the LLM then combines the output from its calculator tool into a lovely couple of paragraphs, the last of which wraps up with an optimistic offer to "explore" its answer further, should you so wish.

Symbolic Execution

For some strange reason, the programmers that wrote AI engines thought it might be a good idea for an AI to be a programmer in order for it to get smarter. I mean, if I want to train my LLM not to put "rocks" on the grocery list, obviously it would do that better if it spoke to me in Python.

Anyway, here we are: symbolic execution.

Having the model interface with a "symbolic interpreter" or other symbolic representation is one way to improve a model's reasoning capability. There are various different approaches, ranging from very mathematical logical languages to more compute-related dialects such as executing Python scripts.

In this idea, the LLM generates Python as its output, which is then interpreted by a built-in integration. If you prompt the AI with this:

```
This a test.
```

Then the AI will generate this intermediate Python program to run:

```
print("This is indeed a test.")
```

This is a silly example, because LLMs are certainly smart enough to do different things for different input prompts. Simple prompts do not need symbolic execution. The idea with symbolic execution is that simple prompts are handled the normal way, without coding, whereas complex reasoning type inputs are handled indirectly by generating an executable Python program.

There are some reasons why this indirect approach of generating answers via symbolic execution might work well, at least for a subset of problems. For example, Python programs are more general than LLM's decoding algorithms:

- Looping-type constructs in programs.
- Hierarchical or recursive logic.
- Second level of indirection capability.
- Turing completeness of computation.

Turing completeness is an obscure Computer Science theory about programming languages, where they must have:

1. Sequence — execute two statements in a row.

2. Selection — an "if" statement that chooses two paths.

3. Iteration — looping around to do it again.

There's actually a fourth requirement in practice that's sometimes overlooked in the theoretical treatises:

4. Persistence — ability to store data somewhere (i.e., variables in memory or disk).

LLMs by their lonesome are actually *not* Turing complete, because they are finite, which means that they miss the third capability. LLM models are a fixed-size data structure with very little looping or "iteration" capability.

LLMs are massive, yes, but still finite.

On the other hand, LLMs have all the other requirements, and using a multi-step reasoning algorithm around an LLM can add that type of non-deterministic looping. So, maybe that's all it needs to be Turing complete.

Symbolic execution has not yet really taken off as a way to improve LLM intelligence, although there are some startups still working on this idea, so who knows where it might go?

Perhaps coincidentally, instead of learning coding making the LLMs smarter at lots of stuff, LLMs got really good at coding, to the point where the job that is now considered most likely to be fully replaced by AI engines: *programmers*.

References

Arithmetic Calculations. Research papers on LLM arithmetic limitations, mostly prior to their solution via reasoning models and tool integrations:

1. Sean Williams, James Huckle, 30 May 2024, *Easy Problems That LLMs Get Wrong*, https://arxiv.org/abs/2405.19616 Code: https://github.com/autogenai/easy-problems-that-llms-get-wrong
2. Sean McLeish, Arpit Bansal, Alex Stein, Neel Jain, John Kirchenbauer, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, Jonas Geiping, Avi Schwarzschild, Tom Goldstein, 27 May 2024, *Transformers Can Do Arithmetic with the Right Embeddings*, https://arxiv.org/abs/2405.17399 (Positional encoding of numeric digits improves math arithmetic accuracy.)
3. Subhro Roy, Dan Roth, 20 Aug 2016 (v2), *Solving General Arithmetic Word Problems*, https://arxiv.org/abs/1608.01413
4. Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Dylan Slack, Qin Lyu, Sean Hendryx, Russell Kaplan, Michele (Mike)Lunati, Summer Yue, 1 May 2024, *A Careful Examination of Large Language Model Performance on Grade School Arithmetic*, https://arxiv.org/abs/2405.00332

5. Owen Dugan, Donato Manuel Jimenez Beneto, Charlotte Loh, Zhuo Chen, Rumen Dangovski, Marin Soljačić, 4 Jun 2024, *OccamLLM: Fast and Exact Language Model Arithmetic in a Single Step*, https://arxiv.org/abs/2406.06576
6. Aaditya K. Singh, DJ Strouse, 22 Feb 2024, *Tokenization counts: the impact of tokenization on arithmetic in frontier LLMs*, https://arxiv.org/abs/2402.14903
7. Safal Shrestha, Minwu Kim, Keith Ross, 12 Feb 2025, *Mathematical Reasoning in Large Language Models: Assessing Logical and Arithmetic Errors across Wide Numerical Ranges*, https://arxiv.org/abs/2502.08680

Calculator Tool Integrations. Research papers on combining calculator tools with LLMs:

1. Liu, Z., Zheng, Y., Yin, Z. et al., 2025, *ArithmeticGPT: empowering small-size large language models with advanced arithmetic skills*, Mach Learn 114, 24 (2025). https://doi.org/10.1007/s10994-024-06681-1 https://link.springer.com/article/10.1007/s10994-024-06681-1 https://github.com/ai4ed/ArithmeticGPT (Integrate a calculator into the processing.)
2. reiinakano, November 12, 2019, *Teaching a neural network to use a calculator*, https://reiinakano.com/2019/11/12/solving-probability.html (Integrate SymPy calculator into the results of a neural network, by looking for the equals sign.)
3. Yakun Zhu, Shaohang Wei, Xu Wang, Kui Xue, Xiaofan Zhang, Shaoting Zhang, 17 Oct 2024, *MeNTi: Bridging Medical Calculator and LLM Agent with Nested Tool Calling*, https://arxiv.org/abs/2410.13610
4. Florian Dietz, Dietrich Klakow, 1 Jan 2025, *IGC: Integrating a Gated Calculator into an LLM to Solve Arithmetic Tasks Reliably and Efficiently*, https://arxiv.org/abs/2501.00684

LLM Chess Limitations. By default, LLMs treat chess moves like words, and struggle to integrate the two-dimensional strategy:

1. Dynomight, Nov 2024, *Something weird is happening with LLMs and chess*, https://dynomight.net/chess/
2. Victor Tangermann, Sep 13, 2024, *OpenAI's New "Strawberry" AI Is Still Making Idiotic Mistakes*, https://futurism.com/openai-strawberry-o1-mistakes
3. Dynomight, Nov 2024, *OK, I can partly explain the LLM chess weirdness now*, https://dynomight.net/more-chess/

Word Puzzles. Research on LLMs has shown them to be poor at word puzzles, crosswords, and anagrams, although reasoning models starting with the OpenAI "Strawberry" model have improved things:

1. Abdelrahman "Boda" Sadallah, Daria Kotova, Ekaterina Kochmar, 15 Mar 2024, *Are LLMs Good Cryptic Crossword Solvers?* https://arxiv.org/abs/2403.12094 Code: https://github.com/rdeits/cryptics
2. Michael King, July 24, 2023, *Large Language Models are Extremely Bad at Creating Anagrams*, https://www.techrxiv.org/doi/full/10.36227/techrxiv.23712309.v1
3. Victor Tangermann, Sep 13, 2024, *OpenAI's New "Strawberry" AI Is Still Making Idiotic Mistakes*, https://futurism.com/openai-strawberry-o1-mistakes
4. Sam Liberty, Oct 15, 2024, *Why AI Can't Crack the NYT Connections Puzzle (Yet)*, https://medium.com/design-bootcamp/why-ai-cant-crack-the-nyt-connections-puzzle-yet-7bd3e00b4087
5. Reddit, 2024, *LLMs can't solve anagram problems?* https://www.reddit.com/r/learnmachinelearning/comments/1cjn5tz/llms_cant_solve_anagram_problems/

Mathematical Reasoning. There are plenty of research papers on empowering LLMs with better mathematical and logical reasoning capabilities:

1. Nate Kushman, Yoav Artzi, Luke Zettlemoyer, Regina Barzilay, June 2014, *Learning to Automatically Solve Algebra Word Problems*, P14-1026 Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), https://aclanthology.org/P14-1026/ PDF: https://aclanthology.org/P14-1026.pdf
2. Yan Wang, Xiaojiang Liu, Shuming Shi, September 2017, *Deep Neural Solver for Math Word Problems*, D17-1088, Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing Copenhagen, Denmark, https://aclanthology.org/D17-1088/ PDF: https://aclanthology.org/D17-1088.pdf
3. Guoxin Chen, Minpeng Liao, Chengxi Li, Kai Fan, 6 May 2024, *AlphaMath Almost Zero: process Supervision without process*, https://arxiv.org/abs/2405.03553 https://github.com/MARIO-Math-Reasoning/Super_MARIO
4. Shima Imani, Liang Du, and H. Shrivastava, 2023, *Mathprompter: Mathematical reasoning using large language models*, ArXiv, abs/2303.05398, https://arxiv.org/abs/2303.05398

5. Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, Wenpeng Yin, 5 Apr 2024 (v3), *Large Language Models for Mathematical Reasoning: Progresses and Challenges*, https://arxiv.org/abs/2402.00157
6. Luyu Qiu, Jianing Li, Chi Su, Chen Jason Zhang, Lei Chen, 22 Jul 2024, *Dissecting Multiplication in Transformers: Insights into LLMs*, https://arxiv.org/abs/2407.15360
7. Bbot, 2024, *Why is GPT bad at math? Because it can't see digits!* https://bbot.org/etc/gpt-math.png
8. Andrew Blair-Stanek, Nils Holzenberger, Benjamin Van Durme, 7 Feb 2024 (v2), *OpenAI Cribbed Our Tax Example, But Can GPT-4 Really Do Tax?* https://arxiv.org/abs/2309.09992
9. Asir Saadat, Tasmia Binte Sogir, Md Taukir Azam Chowdhury, Syem Aziz, 16 Oct 2024, *When Not to Answer: Evaluating Prompts on GPT Models for Effective Abstention in Unanswerable Math Word Problems*, https://arxiv.org/abs/2410.13029
10. Amogh Akella, 29 Oct 2024, *Problem Categorization Can Help Large Language Models Solve Math Problems*, https://arxiv.org/abs/2411.00042
11. Michael Nuñez, November 11, 2024, *AI's math problem: FrontierMath benchmark shows how far technology still has to go*, https://venturebeat.com/ai/ais-math-problem-frontiermath-benchmark-shows-how-far-technology-still-has-to-go/
12. Yibo Yan, Jiamin Su, Jianxiang He, Fangteng Fu, Xu Zheng, Yuanhuiyi Lyu, Kun Wang, Shen Wang, Qingsong Wen, Xuming Hu, 6 Dec 2024, *A Survey of Mathematical Reasoning in the Era of Multimodal Large Language Model: Benchmark, Method & Challenges*, https://arxiv.org/abs/2412.11936
13. Shuguang Chen, Guang Lin, 28 Dec 2024, *LLM Reasoning Engine: Specialized Training for Enhanced Mathematical Reasoning*, https://arxiv.org/abs/2412.20227
14. Beichen Zhang, Yuhong Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Haodong Duan, Yuhang Cao, Dahua Lin, Jiaqi Wang, 6 Jan 2025, *BoostStep: Boosting mathematical capability of Large Language Models via improved single-step reasoning*, https://arxiv.org/abs/2501.03226 https://github.com/beichenzbc/BoostStep
15. xenaproject, December 22, 2024, *Can AI do maths yet? Thoughts from a mathematician*, https://xenaproject.wordpress.com/2024/12/22/can-ai-do-maths-yet-thoughts-from-a-mathematician/
16. Zain Ul Abedin, Shahzeb Qamar, Lucie Flek, Akbar Karimi, 14 Jan 2025, *ArithmAttack: Evaluating Robustness of LLMs to Noisy Context in Math Problem Solving*, https://arxiv.org/abs/2501.08203
17. Ali Forootani, 22 Mar 2025, *A Survey on Mathematical Reasoning and Optimization with Large Language Models*, https://arxiv.org/abs/2503.17726

Symbolic Execution. The use of a sequence of symbols in either a logical language or a real programming language, which is then executed as a program or "script" can create a type of advanced reasoning, especially in mathematical and reasoning domains. Research papers include:

1. Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen, 2022, *Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks*, arXiv preprint arXiv:2211.12588, 2022, https://arxiv.org/abs/2211.12588 (Integrate a Python interpreter to execute the code generated by the LLM to answer the query.)
2. Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig, 2023, *Pal: Program-aided language models. In International Conference on Machine Learning*, pages 10764–10799. PMLR, https://arxiv.org/abs/2211.10435 Code: http://reasonwithpal.com/ (Python interpreter integrated as a tool for LLMs.)
3. Long Hei Matthew Lam, Ehsan Shareghi, 1 Jun 2024, *A Closer Look at Logical Reasoning with LLMs: The Choice of Tool Matters*, https://arxiv.org/abs/2406.00284 (Using symbolic solvers with LLMs.)
4. M Keber, I Grubišic, A Barešic, A Jovic, 2024, *A Review on Neuro-symbolic AI Improvements to Natural Language Processing*, https://www.researchgate.net/profile/Alan-Jovic/publication/380911364_A_Review_on_Neuro-symbolic_AI_Improvements_to_Natural_Language_Processing/links/6655c0ec22a7f16b4f51fb2f/A-Review-on-Neuro-symbolic-AI-Improvements-to-Natural-Language-Processing.pdf
5. Joy He-Yueya, Gabriel Poesia, Rose E. Wang, and Noah D. Goodman, 2023, *Solving math word problems by combining language models with symbolic solvers*, ArXiv, abs/2304.09102, https://arxiv.org/abs/2304.09102
6. Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, Mehrdad Farajtabar, 7 Oct 2024, *GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models*, https://arxiv.org/abs/2410.05229
7. Mayi Xu, Yunfeng Ning, Yongqi Li, Jianhao Chen, Jintao Wen, Yao Xiao, Shen Zhou, Birong Pan, Zepeng Bao, Xin Miao, Hankun Kang, Ke Sun, Tieyun Qian, 2 Jan 2025, *Reasoning based on symbolic and parametric knowledge bases: a survey*, https://arxiv.org/abs/2501.01030 (Extensive survey of reasoning from CoT to knowledge graphs to table-based reasoning.)
8. Yixuan Li, Lewis Frampton, Federico Mora, Elizabeth Polgreen, 9 Jan 2025, *Online Prompt and Solver Selection for Program Synthesis*, https://arxiv.org/abs/2501.05247
9. Benjamin Callewaert, Simon Vandevelde, Joost Vennekens, 24 Jan 2025, *VERUS-LM: a Versatile Framework for Combining LLMs with Symbolic Reasoning*, https://arxiv.org/abs/2501.14540

10. Yubin Ge, Salvatore Romeo, Jason Cai, Raphael Shu, Monica Sunkara, Yassine Benajiba, Yi Zhang, 3 Feb 2025, *TReMu: Towards Neuro-Symbolic Temporal Reasoning for LLM-Agents with Memory in Multi-Session Dialogues*, https://arxiv.org/abs/2502.01630
11. Siheng Xiong, Jieyu Zhou, Zhangding Liu, Yusen Su, 2 May 2025, *SymPlanner: Deliberate Planning in Language Models with Symbolic Representation*, https://arxiv.org/abs/2505.01479
12. Adam Stein, Aaditya Naik, Neelay Velingker, Mayur Naik, Eric Wong, 30 May 2025, *The Road to Generalizable Neuro-Symbolic Learning Should be Paved with Foundation Models*, https://arxiv.org/abs/2505.24874

13. Tools

"Right now, I think the frontier is these AI tools."

— Sam Altman, July, 2025.

What are Tools?

Humans like to think that we are distinct from the "animals" because we use tools. Sadly, this does not distinguish us from the AIs, because they've also evolved to use tools.

LLMs require tools to do more advanced things, just like humans. For example, if someone asks you the time, you look at your watch (or your phone). If you ask an LLM "What is the time?" there is nothing in its training data set that could possibly answer this correctly. The only way is to use a tool called a "clock" and it must be integrated into the LLM, and executed by the AI Engine as part of answering your query. Some of the things that are hard for an LLM to do without tools include:

- Having real-time or up-to-date information (e.g., stock prices or the latest AI research papers).
- Computation-related questions beyond basic arithmetic.
- Information that's only in a different place (e.g., the company's internal ERP database).
- Time-specific or locale-specific information that differs from its training.

Another type of tool that LLMs can use are those that perform an action for you, such as sending an email. These are called "agents" in the AI industry, rather than "tools" as the terminology.

That's not where the naming confusion ends. The AI research industry seems to make some distinctions about several different types of tools. For example, there are these AI research areas called:

- Retrieval Augmented Generation (RAG) — find a paragraph of text from a set of documents.
- Tool Augmented Language Models (TALM) — run a dynamic computation tool like a calculator.
- Tools "Hooks" — run a dynamic preprocessing tool on your text input.
- Plugins — log into someone's database for information (e.g., real estate listings).
- Search plugins — run a full internet query to return some blue links for AI to read.

Come on, they don't need different names when they're all just tools! The AI engine calls out for help from one or other of these various different pieces of software, whether it needs a BMI calculator, a corporate HR document, or a full search of the internet.

Well, maybe there is a key distinction between two different classes of "tools" if we exclude the "action agents" types, and focus on tools for helping the LLM just to answer questions:

1. Look up more information — RAG, internet search, database plugins.

2. Dynamic computations — clocks, calculators, and many more.

One way to think about this: static search tools are looking for extra human-written information, whereas dynamic tools are creating their own! Types of dynamic calculation tools include:

- Clocks
- Calculators (arithmetic)
- Converters (e.g., pounds to kilograms)
- Calendars (date or day calculations)

Modern LLMs use literally thousands of tools. All of these tools are pieces of software that sit alongside the AI model in the backend servers.

AI engines don't normally tell you what tools they've used to answer. How unfair is that! We're not allowed to use AI to answer questions in job interviews, but the AIs are allowed to sneak around using any tools they like.

There are a lot of tools that an AI can use, far beyond the basic ones. For example, if you ask your AI engine to compute the "Flesch-Kincaid" reading level metric for a paragraph of text, it has to use a tool to compute that.

Integration of Tools

There's a lot of variability in architectures for using tools, but the basic ideas are to use them for:

- The beginning — preprocessing ("hooks").
- The middle — interim computations.
- The end — finalization or formatting.

The main one is the muddle in the middle. Your LLM will run a sequence something like this:

1. Read input prompt ("Should I eat jelly beans in order of wavelength?").

2. Decide to get the results of the tool.

3. Insert "tool tokens" into the output text.

4. Run the tool or tools (by finding tool tokens in the LLM's output).

5. Merge the tool output into the final output text results.

Like humans, an AI needs to learn to look at its watch if someone asks the time. Specific training data sets are required that tell the AI what tool to use, and when. This is the "deciding" phase for tool usage.

The AI engine has to recognize in the LLM output that a tool must be executed.

There are a variety of ways to do this:

- Tool-specific tokens — i.e., the LLM can emit a "trigger" token to run a tool. Note that PEFT could be used here to fine-tune new tool capabilities, by only adding a few new tool-triggering tokens to the vocabulary.)
- Placeholder patterns — i.e., output something like an "--insert current time here--" special pattern is another way, and the engine then looks for these patterns, which avoids adding tool tokens to the vocabulary, but is inefficient in that there are multiple text tokens in the output).
- Code generation — there are various AI models that will generate code, such as in Python, that can be executed to generate the answer. This is a general solution, because Python can call various submodules and can thereby generate many tools.
- Multi-level planning — the AI first generates a plan of how to answer the query, including what tools to use, and then runs any tools, and then does another inference query to collate it into a final answer.

Sometimes, there are two steps of LLM inference being done, with both the deciding and merging steps above. Alternatively, the tool integration can be simpler with the tool's output results just inserted verbatim into the middle of the output.

Computers as Tools

Lately, there's been a huge amount of work on LLM things called "GUI agents" or "computer usage models" in the AI space. The idea is that the LLM can actually look at your screen and tap away on your keyword and click the mouse, too.

Phones, too!

A computer or smartphone is a great tool for an LLM. Anything that a computer can do, the LLM can now do. Amusingly, it's a really tough problem to solve, which is odd, since the AI engine is already running inside a great big computer. Lots of research papers! The technical issues to solve include:

- Understanding the screen (as an image)
- Integration with input devices (keyboard/mouse)
- Context of the screen at a high level (e.g., which app is running?)

There's another major problem to solve: mistakes. LLMs always make mistakes, and that's more of a problem if the agent is doing something for you on your computer.

LLM Hooks

LLM hooks are integrations of tools that perform preprocessing on prompt inputs. The use of hooks is a special case of Tool-Augmented Language Models (TALM). The idea with pre-processing hooks is that the tools can augment the input prompts with extra information that the LLM can use, which is similar to RAG-based retrieval, but is based on non-LLM tools that perform dynamic computation.

Tool Augmented Language Models (TALM) is the use of non-LLM tools to augment the processing of LLMs. Tools can be used to compute more data based on the prompts, and can be used on conjunction with reasoning like Chain-of-Thought, RAG retrievals, or agentic architectures. Hence, the idea of "hooks" is a special type of limited tool, that doesn't scour the internet for more information, but only performs some dynamic computation on the prompt text as input.

Using "hooks" to launch a tool to preprocess the user's input prompt is a lesser-known technique. You can use heuristics to decide that a tool is used, and it's called before the LLM via "hooks" in the code. This idea skips the "deciding" step above in favor of non-LLM methods. This is faster than having an LLM decide to launch a tool, but it's not always as accurate in choosing whether or not to use a tool.

References

Preprocessing Hooks. Research papers on LLM "hooks" for integrations to dynamic tools:

1. Damien de Mijolla, Wen Yang, Philippa Duckett, Christopher Frye, Mark Worrall, 8 Dec 2024, *Language hooks: a modular framework for augmenting LLM reasoning that decouples tool usage from the model and its prompt*, https://arxiv.org/abs/2412.05967
2. Muhayy Ud Din, Jan Rosell, Waseem Akram, Isiah Zaplana, Maximo A Roa, Lakmal Seneviratne, Irfan Hussain, 10 Dec 2024, *Ontology-driven Prompt Tuning for LLM-based Task and Motion Planning*, https://arxiv.org/abs/2412.07493 https://muhayyuddin.github.io/llm-tamp/ (Detecting objects in the prompt text and then using a RALM algorithm to query an ontology database.)
3. Julian Perry, Surasakdi Siripong, Thanakorn Phonchai, 15 Jan 2025, *Dynamic Knowledge Integration for Enhanced Vision-Language Reasoning*, https://arxiv.org/abs/2501.08597 (Augment training data dynamically by retrieving extra information.)

4. Liu, Z., Zheng, Y., Yin, Z. et al., 2025, *ArithmeticGPT: empowering small-size large language models with advanced arithmetic skills*, Mach Learn 114, 24, 2025, https://doi.org/10.1007/s10994-024-06681-1 https://link.springer.com/article/10.1007/s10994-024-06681-1 https://github.com/ai4ed/ArithmeticGPT (Integrate a calculator into the processing.)
5. Sam Lin, Wenyue Hua, Lingyao Li, Zhenting Wang, Yongfeng Zhang, 17 Feb 2025. *ADO: Automatic Data Optimization for Inputs in LLM Prompts*, https://arxiv.org/pdf/2502.11436 (Reformulating the input context such as by semantical marking of relevant content or formatting changes.)
6. Andrew Neeser, Kaylen Latimer, Aadyant Khatri, Chris Latimer, Naren Ramakrishnan, 16 Feb 2025, *QuOTE: Question-Oriented Text Embeddings*, https://arxiv.org/abs/2502.10976 (Augmenting RAG chunks with additional information, such as questions the chunk might answer.)
7. Wendi Cui, Jiaxin Zhang, Zhuohang Li, Hao Sun, Damien Lopez, Kamalika Das, Bradley A. Malin, Sricharan Kumar, 26 Feb 2025, *Automatic Prompt Optimization via Heuristic Search: A Survey*, https://arxiv.org/abs/2502.18746 (Survey of auto prompting, from basic LLM enhancements to some methods quite similar to RALM and TALM.)
8. Leixian Shen, Haotian Li, Yifang Wang, Xing Xie, Huamin Qu, 4 Mar 2025, *Prompting Generative AI with Interaction-Augmented Instructions*, https://arxiv.org/abs/2503.02874

TALM Research. Research papers on Tool-Augmented Language Model (TALM) technologies:

1. Yechen Xu, Xinhao Kong, Tingjun Chen, Danyang Zhuo, 4 Jun 2024 (v2), *Conveyor: Efficient Tool-aware LLM Serving with Tool Partial Execution*, https://arxiv.org/abs/2406.00059 Code: https://github.com/conveyor-sys/conveyor (Speeding up inference by partially running tools in parallel to the LLM query processing, rather than sequentially after the LLM request, by detecting tool requests deep inside the decoding algorithm and starting them off immediately, before the LLM has finished generating the fully decode output.)
2. Yubo Ma, Zhibin Gou, Junheng Hao, Ruochen Xu, Shuohang Wang, Liangming Pan, Yujiu Yang, Yixin Cao, Aixin Sun, Hany Awadalla, Weizhu Chen, 21 Feb 2024 (v2), *SciAgent: Tool-augmented Language Models for Scientific Reasoning*, https://arxiv.org/abs/2402.11451
3. Aaron Parisi, Yao Zhao, and Noah Fiedel, 2022, *Talm: Tool augmented language models*, arXiv preprint arXiv:2205.12255, https://arxiv.org/abs/2205.12255
4. Simranjit Singh, Andreas Karatzas, Michael Fore, Iraklis Anagnostopoulos, Dimitrios Stamoulis, 7 May 2024, *An LLM-Tool Compiler for Fused Parallel Function Calling*, https://arxiv.org/abs/2405.17438

5. Reyna Abhyankar, Zijian He, Vikranth Srivatsa, Hao Zhang, Yiying Zhang, July 2024, *InferCept: Efficient Intercept Support for Augmented Large Language Model Inference*, Proceedings of the 41st International Conference on Machine Learning, PMLR 235:81-95, 2024, https://proceedings.mlr.press/v235/abhyankar24a.html PDF: https://raw.githubusercontent.com/mlresearch/v235/main/assets/abhyankar24a/abhyankar24a.pdf
6. Yaroslav Zharov, Yury Khudyakov, Evgeniia Fedotova, Evgeny Grigorenko, Egor Bogomolov, 18 Feb 2024, *Tool-Augmented LLMs as a Universal Interface for IDEs*, https://arxiv.org/abs/2402.11635
7. Florian Dietz, Dietrich Klakow, 1 Jan 2025, *IGC: Integrating a Gated Calculator into an LLM to Solve Arithmetic Tasks Reliably and Efficiently*, https://arxiv.org/abs/2501.00684
8. Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, Weizhu Chen, 21 Feb 2024 (v4), *ToRA: A Tool-Integrated Reasoning Agent for Mathematical Problem Solving*, https://arxiv.org/abs/2309.17452
9. Aiyao He, Sijia Cui, Shuai Xu, Yanna Wang, Bo Xu, 13 May 2025, *TUMS: Enhancing Tool-use Abilities of LLMs with Multi-structure Handlers*, https://arxiv.org/abs/2505.08402

Tools. Tool integration papers (also known as "function calls"):

1. Junzhi Chen, Juhao Liang, Benyou Wang, 9 May 2024, *Smurfs: Leveraging Multiple Proficiency Agents with Context-Efficiency for Tool Planning*, https://arxiv.org/abs/2405.05955
2. reiinakano, November 12, 2019, *Teaching a neural network to use a calculator*, https://reiinakano.com/2019/11/12/solving-probability.html (Integrate SymPy calculator into the results of a neural network, by looking for the equals sign.)
3. Yubo Ma, Zhibin Gou, Junheng Hao, Ruochen Xu, Shuohang Wang, Liangming Pan, Yujiu Yang, Yixin Cao, Aixin Sun, Hany Awadalla, Weizhu Chen, 21 Feb 2024 (v2), *SciAgent: Tool-augmented Language Models for Scientific Reasoning*, https://arxiv.org/abs/2402.11451
4. Shibo Hao, Tianyang Liu, Zhen Wang, Zhiting Hu, 2023, *ToolkenGPT: Augmenting Frozen Language Models with Massive Tools via Tool Embeddings*, Part of Advances in Neural Information Processing Systems 36 (NeurIPS 2023) Main Conference Track, https://proceedings.neurips.cc/paper_files/paper/2023/hash/8fd1a81c882cd45f64958da6284f4a3f-Abstract-Conference.html
5. Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al., 2023, *ToolLLM: Facilitating large language models to master 16000+ real-world APIs*, arXiv preprint arXiv:2307.16789, https://arxiv.org/abs/2307.16789
6. Yu Gu, Yiheng Shu, Hao Yu, Xiao Liu, Yuxiao Dong, Jie Tang, Jayanth Srinivasa, Hugo Latapie, Yu Su, 22 Feb 2024, *Middleware for LLMs: Tools Are Instrumental for Language Agents in Complex Environments*, https://arxiv.org/abs/2402.14672

7. Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, Thomas Scialom, 9 Feb 2023, *Toolformer: Language Models Can Teach Themselves to Use Tools*, https://arxiv.org/abs/2302.04761
8. Cobus Greyling, June 16, 2023, *Practical Examples of OpenAI Function Calling*, https://cobusgreyling.medium.com/practical-examples-of-openai-function-calling-a6419dc38775
9. University of California, Berkeley, 2024, *Berkeley Function-Calling Leaderboard*, https://gorilla.cs.berkeley.edu/leaderboard.html https://huggingface.co/datasets/gorilla-llm/Berkeley-Function-Calling-Leaderboard
10. Shishir Patil, May 10, 2024, *Teaching Large Language Models to Use Tools at Scale*, Ph.D. Thesis, Electrical Engineering and Computer Sciences, University of California, Berkeley, Technical Report No. UCB/EECS-2024-85, http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-85.html https://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-85.pdf
11. Thomas Reid, Jul 31, 2024, *Ollama's Latest Update: Tool Use: Everything you need to know about function calling in Ollama*, https://ai.gopubby.com/ollamas-latest-update-tool-use-7b809e15be5c
12. Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Felix Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, Ruoming Pang, 8 Aug 2024, *ToolSandbox: A Stateful, Conversational, Interactive Evaluation Benchmark for LLM Tool Use Capabilities*, https://arxiv.org/abs/2408.04682 Code: https://github.com/apple/ToolSandbox
13. Lutfi Eren Erdogan, Nicholas Lee, Siddharth Jha, Sehoon Kim, Ryan Tabrizi, Suhong Moon, Coleman Hooper, Gopala Anumanchipalli, Kurt Keutzer, Amir Gholami, 1 Sep 2024, *TinyAgent: Function Calling at the Edge*, https://arxiv.org/abs/2409.00608 https://github.com/SqueezeAILab/TinyAgent
14. Yupu Hao, Pengfei Cao, Zhuoran Jin, Huanxuan Liao, Yubo Chen, Kang Liu, Jun Zhao, 23 Sep 2024 (v2), *CITI: Enhancing Tool Utilizing Ability in Large Language Models without Sacrificing General Performance*, https://arxiv.org/abs/2409.13202
15. Renxi Wang, Xudong Han, Lei Ji, Shu Wang, Timothy Baldwin, Haonan Li, 8 Oct 2024 (v2), *ToolGen: Unified Tool Retrieval and Calling via Generation*, https://arxiv.org/abs/2410.03439
16. Elias Lumer, Vamse Kumar Subbiah, James A. Burke, Pradeep Honaganahalli Basavaraju, Austin Huber, 22 Oct 2024 (v2), *Toolshed: Scale Tool-Equipped Agents with Advanced RAG-Tool Fusion and Tool Knowledge Bases*, https://arxiv.org/abs/2410.14594
17. Jerry Huang, Prasanna Parthasarathi, Mehdi Rezagholizadeh, Sarath Chandar, 14 Apr 2024, *Towards Practical Tool Usage for Continually Learning LLMs*, https://arxiv.org/abs/2404.09339
18. Amy Marks, Jun 11, 2024, *Clarifying Function Calling / Tool Use in LLMs*, https://medium.com/@aevalone/clarifying-function-calling-tool-use-in-llms-6511af510f99

19. In Gim, Seung-seob Lee, Lin Zhong, 9 Dec 2024, *Asynchronous LLM Function Calling*, https://arxiv.org/abs/2412.07017 (Overlap LLM computations and tool execution.)
20. Dian Yu, Yuheng Zhang, Jiahao Xu, Tian Liang, Linfeng Song, Zhaopeng Tu, Haitao Mi, Dong Yu, 22 Dec 2024, *Teaching LLMs to Refine with Tools*, https://arxiv.org/abs/2412.16871
21. Wenjun Li, Dexun Li, Kuicai Dong, Cong Zhang, Hao Zhang, Weiwen Liu, Yasheng Wang, Ruiming Tang, Yong Liu, 18 Feb 2025, *Adaptive Tool Use in Large Language Models with Meta-Cognition Trigger*, https://arxiv.org/abs/2502.12961 (Examining the decision whether or not to launch a tool, and the inefficiency of non-needed tool calls.)
22. Mengsong Wu, Tong Zhu, Han Han, Xiang Zhang, Wenbiao Shao, Wenliang Chen, 21 Mar 2025, *Chain-of-Tools: Utilizing Massive Unseen Tools in the CoT Reasoning of Frozen Language Models*, https://arxiv.org/abs/2503.16779 https://github.com/fairyshine/Chain-of-Tools
23. Wang et. al., 2025, *Function Calling in Large Language Models: Industrial Practices, Challenges, and Future Directions*, https://openreview.net/pdf?id=LNxVGPedFW
24. Beong-woo Kwak, Minju Kim, Dongha Lim, Hyungjoo Chae, Dongjin Kang, Sunghwan Kim, Dongil Yang, Jinyoung Yeo, 29 May 2025, *ToolHaystack: Stress-Testing Tool-Augmented Language Models in Realistic Long-Term Interactions*, https://arxiv.org/abs/2505.23662 https://github.com/bwookwak/ToolHaystack

LLM Computer Usage. Research on computer usage agent LLMs, or "GUI agents":

1. Anthropic, 23 Oct 2024, *Developing a computer use model*, https://www.anthropic.com/news/developing-computer-use
2. Anirban Ghoshal, 23 Oct 2024, *How Anthropic's new 'computer use' ability could further AI automation*, https://www.cio.com/article/3583260/how-anthropics-new-computer-use-ability-could-further-ai-automation.html
3. Xiao Liu, Bo Qin, Dongzhu Liang, Guang Dong, Hanyu Lai, Hanchen Zhang, Hanlin Zhao, Iat Long Iong, Jiadai Sun, Jiaqi Wang, Junjie Gao, Junjun Shan, Kangning Liu, Shudan Zhang, Shuntian Yao, Siyi Cheng, Wentao Yao, Wenyi Zhao, Xinghan Liu, Xinyi Liu, Xinying Chen, Xinyue Yang, Yang Yang, Yifan Xu, Yu Yang, Yujia Wang, Yulin Xu, Zehan Qi, Yuxiao Dong, Jie Tang, 28 Oct 2024, *AutoGLM: Autonomous Foundation Agents for GUIs*, https://arxiv.org/abs/2411.00820
4. Shuai Wang, Weiwen Liu, Jingxuan Chen, Weinan Gan, Xingshan Zeng, Shuai Yu, Xinlong Hao, Kun Shao, Yasheng Wang, Ruiming Tang, 7 Nov 2024, *GUI Agents with Foundation Models: A Comprehensive Survey*, https://arxiv.org/abs/2411.04890
5. Siyuan Hu, Mingyu Ouyang, Difei Gao, Mike Zheng Shou, 15 Nov 2024, *The Dawn of GUI Agent: A Preliminary Case Study with Claude 3.5 Computer Use*, https://arxiv.org/abs/2411.10323 https://github.com/showlab/computer_use_ootb

6. Show Lab, Nov 2024, *ShowUI: ShowUI is a lightweight (2B) vision-language-action model designed for GUI agents*, https://huggingface.co/showlab/ShowUI-2B
7. Chaoyun Zhang, Shilin He, Jiaxu Qian, Bowen Li, Liqun Li, Si Qin, Yu Kang, Minghua Ma, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, Qi Zhang, 27 Nov 2024, *Large Language Model-Brained GUI Agents: A Survey*, https://arxiv.org/abs/2411.18279
8. Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, Zhiyong Wu, 23 Feb 2024 (v2), *SeeClick: Harnessing GUI Grounding for Advanced Visual GUI Agents*, https://arxiv.org/abs/2401.10935
9. Keen You, Haotian Zhang, Eldon Schoop, Floris Weers, Amanda Swearngin, Jeffrey Nichols, Yinfei Yang, Zhe Gan, 8 Apr 2024, *Ferret-UI: Grounded Mobile UI Understanding with Multimodal LLMs*, https://arxiv.org/abs/2404.05719
10. Yiheng Xu, Zekun Wang, Junli Wang, Dunjie Lu, Tianbao Xie, Amrita Saha, Doyen Sahoo, Tao Yu, Caiming Xiong, 5 Dec 2024, *Aguvis: Unified Pure Vision Agents for Autonomous GUI Interaction*, https://arxiv.org/abs/2412.04454 https://aguvis-project.github.io/
11. Dang Nguyen, Jian Chen, Yu Wang, Gang Wu, Namyong Park, Zhengmian Hu, Hanjia Lyu, Junda Wu, Ryan Aponte, Yu Xia, Xintong Li, Jing Shi, Hongjie Chen, Viet Dac Lai, Zhouhang Xie, Sungchul Kim, Ruiyi Zhang, Tong Yu, Mehrab Tanjim, Nesreen K. Ahmed, Puneet Mathur, Seunghyun Yoon, Lina Yao, Branislav Kveton, Thien Huu Nguyen, Trung Bui, Tianyi Zhou, Ryan A. Rossi, Franck Dernoncourt, 18 Dec 2024, *GUI Agents: A Survey*, https://arxiv.org/abs/2412.13501
12. Hao Wen, Shizuo Tian, Borislav Pavlov, Wenjie Du, Yixuan Li, Ge Chang, Shanhui Zhao, Jiacheng Liu, Yunxin Liu, Ya-Qin Zhang, Yuanchun Li, 24 Dec 2024, *AutoDroid-V2: Boosting SLM-based GUI Agents via Code Generation*, https://arxiv.org/abs/2412.18116
13. X Hu, T Xiong, B Yi, Z Wei, R Xiao, Y Chen, J Ye, M Tao, Dec 2024, *OS Agents: A Survey on MLLM-Based Agents for General Computing Devices Use*, https://www.preprints.org/frontend/manuscript/3842b6163d82801988adf663ee18b6d5/download_pub
14. Yuhang Liu, Pengxiang Li, Zishu Wei, Congkai Xie, Xueyu Hu, Xinchen Xu, Shengyu Zhang, Xiaotian Han, Hongxia Yang, Fei Wu, 8 Jan 2025, *InfiGUIAgent: A Multimodal Generalist GUI Agent with Native Reasoning and Reflection*, https://arxiv.org/abs/2501.04575
15. Maxwell Zeff, January 23, 2025, *OpenAI launches Operator, an AI agent that performs tasks autonomously*, https://techcrunch.com/2025/01/23/openai-launches-operator-an-ai-agent-that-performs-tasks-autonomously/
16. Matt Marshall, February 22, 2025, *The rise of browser-use agents: Why Convergence's Proxy is beating OpenAI's Operator*, https://venturebeat.com/ai/the-rise-of-browser-use-agents-why-convergences-proxy-is-beating-openais-operator/
17. Wenjia Jiang, Yangyang Zhuang, Chenxi Song, Xu Yang, Chi Zhang, 4 Mar 2025, *AppAgentX: Evolving GUI Agents as Proficient Smartphone Users*, https://arxiv.org/abs/2503.02268
18. Apoorv Agrawal, May 23, 2025, *Why Cars Drive Themselves Before Computers Do: Robocars are ready; robot secretaries aren't… yet*, https://apoorv03.com/p/autonomy

14. Brainy Body

"A strong body makes the mind strong."

— Thomas Jefferson.

Brain Versus Brawn

Do you need a body to be smart? It sounds somewhat facetious when you write it like that. And yet, it's a serious question with a whole slew of research papers behind it, titled: *embodied AI*.

Technically, I'm mainly going to talk about the theories of "embodied cognition" and the achievement of real intelligence. There are other practical aspects in physical AI such as sensory capabilities (e.g., machine vision) and having robots that have mobility and the ability to take actions (i.e., go somewhere and do something).

This research about embodied cognition espouses the theory that achieving true intelligence requires physical interaction with the world. In short:

> *No brain without a body.*

Why might full intelligence require a physical presence in the world?

Some of the reasons a physical body is required include:

- Self-awareness (of oneself and one's dimensions)
- Building a theory of the 3D world
- Generalization of world-based reasoning

Another obvious reason is for an LLM to actually understand the human senses. Any good writer of fiction knows to use the five senses for "show not tell" in a scene:

- Sight (vision)
- Sound (hearing)
- Smell (aural)
- Touch (tactile)
- Taste (gustatory)

You know there are more than five senses, right? I'm not being supernatural, but refer to all the weird extra senses that our bodies actually have, including:

- Physical sensations — pain, discomfort, butterflies, tingling, numbness.
- Body states — hunger, thirst, nausea, fever, illness, dizziness, etc.
- Proprioception — knowing the position of our body in 3D space (my favorite one!).

There's a bunch more. Apparently, humans can detect magnetic fields, but not as well as pigeons. Except for that weird one, you know all of these extra senses, innately, without having to be trained or sent to boarding school.

How many does your LLM understand? None.

It is more than a theory that LLMs should reside in physical robots, such as to achieve mobility or allow actions to occur, but that their *training* must occur in this way so that the LLM can itself be intelligent in a more complete way.

Maybe boarding school for LLMs is worth a try.

Why Embodied?

Think about how a baby learns. They reach out into three-dimensional space learning where everything is, including their own personal space. Later, they crawl around, learning how things are arranged in the world, and what it feels like to touch anything they can find, to their benefit or their peril.

How can an LLM do that? All it has is its words. Can we really explain all of those things to an LLM by giving it more words to read? How is it to learn the meaning of this joke (and is it a joke?):

> Kemp's Law: *Never re-tie only one shoelace.*

An LLM is inherently stuck inside its own head. Indeed, embodied AI is one of the theories about why modern LLMs still struggle with various aspects related to human-level intelligence:

- Two-dimensional spaces (e.g., chess boards or *Galaga*).
- Three-dimensional reasoning (e.g., people sitting at the kitchen table).
- Textures and senses (e.g., how it feels to touch or smell).

Is automated driving of a car in a 2D or 3D world? The answer is: yes, both. There are aspects of 2D maps, certainly, for efficient execution and specification of certain tasks such as pick-up, drop-off and routing. However, for safety the LLM must consider its full three-dimensional shape and the same of the other vehicles and their overall environment.

The benefit of real-world experience is not limited to physical ideas about a multi-dimensional environment and the objects it contains. Some of the more non-obvious limitations to full intelligence that embodied AI seeks to improve include:

- Self-awareness — the AI is in the environment, too).
- Temporal reasoning — time always flows one way.

Understanding of time and the notion of "causality" (cause-and-effect) may be improved by embodied AI. Learning about the world while experiencing real-time changes in the environment could improve an LLM's ability to reason about time.

Fake Versus Real Bodies

Researchers have tried the old *fake it till you make it* style of research for embodied AI, although they don't describe it that way in their research funding proposals. Instead of a real body, we can pretend that scanning through a stream of video and audio data is the same as having eyes and ears. You can try to learn from that, and this is definitely getting better.

Training advanced LLMs on video data is a thing.

But what about real bodies? What about the senses of touch? Can they explore the world and learn what it means to be inside a 3D space? Do they feel pain when they fall off their bicycle?

Lots of robots have real bodies, and they're starting to look impressive, but do they learn?

As far as I know, no-one's really invented a robot hooked up to a smart enough LLM that it could learn. I mean, I could have written "learn as it grew" and there's a few Hollywood movies that I quite like where this happens, but not in real life.

Robot soccer is not quite there yet.

It might be a while before humanity designs an advanced mobile learning station.

Oh, wait! Sorry, I was wrong. We have managed to create something advanced that can learn like that:

Babies.

References

Some of the many references on Embodied AI, with the theory beginning as far back as Brooks (1991), include::

1. Shaoshan Liu and Shuang Wu, Apr 29 2024, *A Brief History of Embodied Artificial Intelligence, and its Outlook: Examining the history, current state, and future of Embodied Artificial Intelligence*, https://cacm.acm.org/blogcacm/a-brief-history-of-embodied-artificial-intelligence-and-its-future-outlook/
2. HCPLab, July 2025 (accessed), *Paper List and Resource Repository for Embodied AI*, https://github.com/HCPLab-SYSU/Embodied_AI_Paper_List
3. Rolf Pfeifer and Fumiya Iida, 2004, *Embodied Artificial Intelligence: Trends and Challenges*, https://people.csail.mit.edu/iida/papers/PfeiferIidaEAIDags.pdf
4. Robert McCarthy, Daniel C.H. Tan, Dominik Schmidt, Fernando Acero, Nathan Herr, Yilun Du, Thomas G. Thuruthel, Zhibin Li, 12 Nov 2024 (v4), *Towards Generalist Robot Learning from Internet Video: A Survey*, https://arxiv.org/abs/2404.19664
5. Yang Liu, Weixing Chen, Yongjie Bai, Xiaodan Liang, Guanbin Li, Wen Gao, Liang Lin, 26 Aug 2024 (v7), *Aligning Cyber Space with Physical World: A Comprehensive Survey on Embodied AI*, https://arxiv.org/abs/2407.06886
6. Zhe Sun, Pengfei Tian, Xiaozhu Hu, Xiaoyu Zhao, Huiying Li, Zhenliang Zhang, 25 Mar 2025, *Body Discovery of Embodied AI*, https://arxiv.org/abs/2503.19941
7. Matej Hoffmann, Shubhan Parag Patni, 15 May 2025, *Embodied AI in Machine Learning -- is it Really Embodied?* https://arxiv.org/abs/2505.10705
8. Rodney A. Brooks, 1991, *Intelligence without representation*, Artificial Intelligence 47 (1991), 139–159, https://people.csail.mit.edu/brooks/papers/representation.pdf (Seminal paper in embodied AI theory.)
9. Yequan Wang, Aixin Sun, 20 May 2025, *Toward Embodied AGI: A Review of Embodied AI and the Road Ahead*, https://arxiv.org/abs/2505.14235

15. Faster AI

"You're gonna need a bigger boat."

— *Jaws*, 1975.

Faster AI

AI engines are already amazingly fast, but it's not enough. LLMs are just not big enough, despite already being ginormous. How much computation is going to be needed for the next generation of even smarter models?

Lots more.

It's not clear yet how far ahead we need to go into the future to get the capacity that's needed. Processing AI computations on GPUs is continually getting faster. NVIDIA is now on an annual cadence of GPU updates, where each general release of silicon beasts gets faster and faster.

There are really two things that are getting faster:

- Faster training — baking brainiess into brawny silicon chips.
- Faster inference — answering your queries and generating poetry.

The weird thing about the speed of AI engines is that there's really only one bottleneck.

It's a complex architecture, but most of the compute grunt is consumed in one small part of the code. AI kernel engineers are forever fiddling with about 1,000 lines of code.

Endless Matrix Multiplications

You studies this in High School math class: matrix multiplications. Remember how you had a weird 2x2 square of numbers, and you had to multiple across a row and down the other column. AI is based on that type of matrix multiplication, and almost nothing else.

Who knew that High School math could be useful?

AI engineers like to make it sound more complicated than that. Our favorite word to use is "tensors" and we hope nobody realizes that it's just matrices, only more than one. Our 2x2x2 tensor in 3D just means that we have two of the 2x2 matrices.

Hence, when you see the press releases about how great NVIDIA GPUs are with "Tensor Cores" you can roll your eyes, and think: I learned that stuff in High School.

500 Ways to Optimize

There are over 500 ways to optimize an AI model. Not an exaggeration — I literally made a list and counted them. I'm not going to force you to read it here, but it's in the references. If you're a new parent with trouble getting the little one to sleep, feel free to read it to them.

The main bottleneck in AI is the above matrix multiplications, and everything else is secondary. Although training and inference are very different algorithms in AI, they both come down to matrices, ahem, sorry, I meant to say tensors. There's a lot of ways to improve things, but the main categories are:

- Fewer matrix multiplications
- Smaller matrices
- Different types of arithmetic

To do fewer matrix multiplications, we've already talked about a big one: Mixture-of-Experts. For example, the 600B model in DeepSeek is cut down into 37B experts, which is a lot fewer matrices every time it crunches out a new word.

Another way is to use smaller matrices inside smaller models. There's an optimization called "distillation" and I wonder how it got that name? The idea is that you train a huge model, called the "teacher" model. And then you get the teacher to talk to another smaller model, so as to train it, or to "distill" the big model's answers into the smaller one. It's called the "student" and I'm not making this stuff up.

Anyway, the smaller model is faster.

Another way to get a smaller model is, you know, just train a smaller model. Go get a refund on half your GPU chips, and just build a small model from the beginning, without needing to train the huge teacher model first. AI engineers like to argue about which approach is better. In short: smaller models are faster but less accurate.

In fact, you can train more than one small model for less cost than a big one, which should really be called a "multi-mini-model" architecture (triple-M!). Instead, it's called "multi-LoRA" because AI researchers like quirky names. If you're got an iPhone with Apple Intelligence baked in, that's the model architecture it's using.

There are many ways to improve the arithmetic. The first trick is that someone noticed that matrix multiplication is like:

1. Multiply stuff, and

2. Add it up.

Here's an idea: do both things together. There are now CPU and GPU hardware instructions that do exactly that, and it's called "Fused Multiply-Add" or FMA. Hence, if your code has both a multiplication and an addition operation, then you're behind the times.

Fuse them together!

Another way is to use smaller data. For example, there's a fancy technique called "quantization" that really just means using smaller numbers.

The latest Blackwell GPU chips are great at "FP4" processing in all those Tensor Cores. What this means is using data that is 4-bit floating-point (FP4) instead of 32-bit floating-point (FP32) for all the numbers we're using inside the matrix multiplications. This drop from 32-bit to 4-bit is eight times less computation, according to my calculations.

Hence, eight times faster AI engines.

Another way to make matrices smaller is to "prune" their numbers. I mean, if your model has a billion numbers, may you don't really need all of them. Unfortunately, since AI engineers don't really understand what all of the numbers do, we also don't really understand what happens if we throw some away. The early research papers in the 1990s on this technique were about "Optimal Brain Damage" and it's an inside joke that's not really a joke.

The idea is that if a number is 0.0001, then just pretend it's actually zero. If the weight of a signal is that small, it might as well be zero, so just throw it away. The simplest way is just to scan all the numbers through the whole model for small ones to discard, which is called "magnitude pruning."

The other way is to look at numbers in particular parts of the model, which is called "structured pruning." It's like throwing away a whole lobe or a whole cortex.

There are literally four ways to prune the different model structures, along the different model dimensions:

- Length — input token pruning (i.e., throw away words)
- Depth — layer pruning ("early exit")
- Width — attention "head" pruning ("sparse attention")
- Model dimension — embeddings pruning ("activation sparsity")

Any way you cut it (I mean, prune it), there's fewer weights and less computation.

Hence, you prune and prune until there's more zeros than non-zero numbers, and then it's called "sparsity." And there are so many ways to do "sparse matrix multiplications" inside a GPU that AI engineers call it terms like "sparse MatMul" or "spGEMM" or other names you can't print.

Anyway, fast.

Training

Inference takes milliseconds to answer a query. Training a big model can take weeks or months. There are faster GPUs and software algorithms that have sped up training since the olden days (last year), but models are getting bigger, too.

When training a big AI model, the most important thing you need is blankets. All of the AI engineers on the training team have to sleep in the data center for days or weeks. The facility is air-conditioned and liquid-cooled down as low as you can get, which isn't actually that low when you have 100,000 overclocked GPUs running. The training engineers aren't really doing much except watching the monitoring dashboard and waiting for the training phase to go: *splat!*

I'm only half kidding.

Training really does fail, and quite often, which is why you use "checkpoints" for training updates (it's a fancy word for backups). The idea is that you checkpoint often, and when it all falls over in a crying heap, then you restart from the last checkpoint. When you realize that it costs tens of millions of dollars to run training on a big cluster, you can see why it's important to have frequent backups.

Why does training fail?

Ah, yes, there are plenty of research papers on that. There are issues like vanishing gradients, exploding gradients, training instabilities due to large outliers, and all sorts of other obscure statistical reasons, which are boring, so let's skip those. More interesting reasons include:

1. Hardware bugs
2. Software bugs
3. Real bugs
4. Space bugs

Hardware bugs are surprisingly common. Well, no, actually not common, but when you have 100,000 of them, then it's not uncommon. GPUs overheat and burn out. Old GPUs are the worst, especially if you've been a cheapskate and bought used GPUs from a retired Bitcoin miner. Weirdly, brand new GPUs can fail, too, due to an occasional fault in the chip manufacturing process.

The error rate is very low, but multiply that by 100,000. Middle-aged GPUs are the sweet spot, so you ideally want to run in your new GPUs with some gentle warm-ups (like a new Porsche). Actually, training algorithms are now good enough to detect a failing GPU and isolate it, but it's not always 100% accurate, so that's when the sleeping engineers need waking up.

Software bugs are an error in the code by the C++ programmers who wrote the training software for the GPUs. Of course, that never happens, since good coders never misplace a semicolon, except for that one time with Microsoft Vista over 20 years ago. Furthermore, we can always blame the hardware engineers or the network engineers. Anyway, let's move on.

Real bugs are not very common these days. If a roach or a small lizard gets into the data center, and ends up next to a GPU, it'll get baked into the training process, literally. Thankfully, most data centers are locked-down so tight to keep out the dust that not many insects or reptiles make an impact.

Space bugs are the funniest thing! Here is where you, Dear Reader, are 100% convinced that it's a joke, and yet, it's really not.

Cosmic rays are a type of electromagnetic radiation that comes in from space. Normally, nobody notices them because they rarely interact with matter, and even if they do, it's such a minor effect that even a human body cell will ignore it and move on with life. However, GPUs have transistors that are etched into silicon at a scale of a few nanometers, which could be two, but let's say five. In comparison, a human cell is at least 10,000 and maybe 100,000 nanometers in size.

Once in a while, a GPU gets hit by a cosmic ray. And once in a while, it goes near a tiny transistor embedded in the silicon. And once in a while, it hits a transistor. And once in a while, it hits at the right time.

A bit gets toggled.

It's a very rare event. But multiply the odds by 100,000 GPUs. And then multiply again by a process that takes days or weeks.

Don't Forget the Network

Programmers tend to avoid network engineers, because they're covered with dust from crawling under the raised floor to fix cables all the time. But AI engines need networking with the highest bandwidth to connect those clusters of 100,000 GPUs that do all the work.

The reason for networking depends on the phase:

- Training — send the training data out, get the weight updates back.
- Inference — send the user queries out, share some caches, get homework essays back.

Yawn! It's true that NVIDIA has a bunch of networking products like NVLink and stuff, with coverage of InfiniBand, Ethernet, BlueField, and more. However, the company only makes about $13 billon US dollars in annual revenue from networking products, so nobody pays any attention to the networking area.

I mean, peanuts!

Pity the poor SVP of Networking at NVIDIA. They run a business bigger than over half the stock market, but never get a CNBC interview.

The Question

Here's the real question about AI technology: is it fast enough? Short answer: No.

We're getting better at running big models at a faster clip, but these big models aren't really that great. I mean, they're amazing in some areas, but, come on, so many dumb things are still coming out of its mouth.

The whole industry is going to need a bigger model, or, more likely, a ton of much bigger models, each of them tweaked to run in particular contexts (e.g., doctors, lawyers, marketing managers, etc.). We're not close to that.

In order to fix all of the areas where AI is still faulty, we need to train it with lots more data, and make the models much bigger. Hence, here's the real question to ask:

How much faster is needed?

References

Articles and papers on the many ways to optimize AI engines:

1. David Spuler, September 2nd, 2024, *500+ LLM Inference Optimization Techniques*, Aussie AI Blog, https://www.aussieai.com/blog/llm-inference-optimization
2. Babak Hassibi and David Stork, 1992, *Second order derivatives for network pruning: Optimal brain surgeon*, Advances in Neural Information Processing Systems, NeurIPS 5, https://proceedings.neurips.cc/paper/1992/file/303ed4c69846ab36c2904d3ba8573050-Paper.pdf
3. Yann LeCun, John Denker, and Sara Solla, 1989, *Optimal brain damage*, In D. Touretzky, editor, Advances in Neural Information Processing Systems, volume 2. Morgan Kaufmann, 1989. https://proceedings.neurips.cc/paper_files/paper/1989/file/6c9882bbac1c7093bd25041881277658-Paper.pdf
4. Diyuan Wu, Ionut-Vlad Modoranu, Mher Safaryan, Denis Kuznedelev, Dan Alistarh, 30 Aug 2024, *The Iterative Optimal Brain Surgeon: Faster Sparse Recovery by Leveraging Second-Order Information*, https://arxiv.org/abs/2408.17163
5. Stock Dividend Screener, March 8, 2025, *Nvidia Data Center Revenue Breakdown: Compute & Networking*, https://stockdividendscreener.com/technology/semiconductor/nvidia/data-center-revenue/#networking

16. The Sweetest Lesson

Brains are 50 Times Bigger

Your brain is 50 times bigger than the best AI engines today. Hence, to get to an intelligence AI model, we might need 50 times more compute power. Don't believe me? Do you think ChatGPT is smarter than you?

Let's look at the facts about brains and models.

The biggest AI model is about two trillion weights. That's for GPT-4 from OpenAI, and we don't even officially know its size, but only from "leaks" we've heard that it was 1.76 trillion total weights (split into eight parts).

Is that the biggest model?

Indeed, GPT-4 was in March, 2023, so it seems likely that the state-of-the-art is more. The weight count of OpenAI's "Strawberry" reasoning model released in September 2024 has not even leaked, but we can reasonably assume it was more.

And there are a handful of other companies that are talking about making models with "trillions" of weights. Most of the details are not disclosed in research papers or press releases, but you can see the word "trillion" in patents and job ads.

It seems very likely that the latest models are above two trillion weights, but probably not by a lot, because the computation needed to run massive models is immense. You don't just train a two trillion weight model once, and done. Rather, you have two trillion weights that need to run as "inference" for every user's query after you release your model to the public.

How does your brain compare?

Firstly, it has around 86 billion neurons. There are various research papers supporting this, and it can go up to around 100 billion neurons. Even so, 86 billion neurons is substantially less than 2 trillion by a factor of about 23.

So, an AI model is 23 times bigger than your brain?

Not so fast!

We're making the wrong comparison here. It's measuring silicon apples against carbon oranges.

Weights are not neurons. No, the neuron-equivalent structure is the "model dimension" of all those neural network computations in the AI engine. A neural network is based on a matrix, like in High School math class.

The equivalent of a neuron in AI is the length of the sides of the matrices, rather than all the numbers inside the matrix. The weights in an AI model are numbers telling it how much "weight" to give to a connection between two neurons, represented in a vector of artificial neurons.

Anyway, the point is that the weights are more like "connections" between a pair of neurons. It's not a surprise that the human brain has these types of connections, because neural networks actually copied that, and made up the idea of weights being their connections.

In the brain, we call them "synapses" and they connect two neurons together via a "dendrite" physical structure.

How many synapses?

Well, the official answer is 100 trillion synapses in the human brain. This sounds like a made-up statistic, even when a chatbot tells you, so we'll stress test its assumptions below. But first, let's summarize the situation:

- AI Models — 2 trillion weights
- Human brain — 100 trillion synapses.

Like I said earlier, your brain is 50 times bigger.

Really That Big?

It's fair to question that 100 trillion synapse statistic. Such a big, round number just sounds like made-up science. In fact, the overall method to get to that number is simplistic:

> *100 billion neurons time 1,000 synapses per neuron.*

There are really two numbers to investigate here. The number of neurons is more accurately believed to be around 86 billion, so maybe that drops our estimate down by 16% overall.

On the other hand, Pakkenberg et. al. (2003) estimates 0.15 quadrillion total synapses, which is more than 50% higher than the overall 100 trillion total. Another analysis is by AI Impact (2015), based on Chudler's facts from neurology textbooks (undated), gives an even higher estimate of 1.8-3.2 times 10^14, which is 180-320 trillion synapses.

Unfortunately, we can't be sure, because no-one has taken the time to sit down and count all those synapses. Instead, there are various estimates based on the weight and volume of different parts of the brain and the apparent density of synapse cell structures within the brain of humans and some animals.

As is obvious from the range in estimates, it's not an exact science, but this still seems to be based on reasonable assumptions.

Finally, an obvious but important point that underpins this analysis: *neurons matter*. The intelligence of humans is closely correlated to the number of neurons and the physical size of the brain. For example, Azevedo et. al. (2009) and Herculano-Houzel (2009) both show that human intelligence is based on us having a linearly scaled-up version of primate brains.

Maybe Less

Yes, the analysis may be incorrect, and the analogy between weights and synapses is not perfect. Perhaps a lot of the 100 trillion connections are doing unimportant things that an AI engine doesn't need. The vast majority of your brain is not doing high-level reasoning.

Rather, your brain has to do a lot of other things related to its attached carbon body. There are plenty of neurons in the brainstem and other similar regions. The brainstem is around 2.6% of the human brain by weight, giving an estimate of 2.2 billion neurons out of the 86 billion total neurons. The whole autonomous nervous system has to control our breathing, heartbeat, and dozens of other systems.

An AI won't need that.

There's also a lot of input signal processing that can perhaps be avoided in a computerized LLM brain. For example, a lot of the brain's power is consumed by processing video inputs, and AI engines may not need to do that.

There are non-LLM ways to process video inputs, so maybe we can hook those up to an LLM. Hence, the LLM itself wouldn't need to do all of that work to process the basic video features like colors and intensities. However, the LLM still has to do the higher-level processing, such as detecting shapes and making sense of them.

Maybe More

On the other lobe, it could be that 100 trillion synapses is an underestimate. Indeed, a number of research papers give much larger numbers. Pakkenberg et. al. (2003) gives an estimate of "0.15 quadrillion" and that's the same as 150 trillion synapses. And AI Impact's 2015 article cites to an alternative statistic of 7,000 synapses per neuron, which when combined with 86 billion neurons, would put us at about 600 trillion total synapses.

Those are some mighty big numbers about what's between your ears. To put it into context, let's divide by two (using AI):

- 100 trillion — 50 times
- 150 trillion — 75 times
- 180 trillion — 90 times
- 320 trillion — 160 times
- 600 trillion — 300 times

In comparison to the 2 trillion weights in state-of-the-art AI models, all these numbers give ratios more than 50 times greater. Indeed, 150 trillion synapses gives a ratio of 75 times higher, and the upper bound is 600 trillion synapses, with a ratio of human brains being over 300 times larger.

Which one will it be? Your guess is as good as mine. Might need some more GPU chips.

Smart Coding

Another variable here is that the brainy boffins in the AI labs have found numerous advanced algorithms to make things go faster. For example, the GPT-4 model has 2 trillion total weights, but they're actually split up into eight separate areas, called "experts" in what's known as a Mixture-of-Experts (MoE) architecture.

Each expert in GPT4- was about 200 billion parameters. Similarly, the DeepSeek model that matched and exceeded this performance was less than 600 billion parameters, and had even smaller experts.

I'm not sure which way this cuts the analysis. If the state-of-the-art models are 200 billion rather than two trillion weights, does this mean that AI models don't need to be as big as the human brain? Maybe an AI model only needs 10 trillion total weights.

Alternatively, does this nuance mean that the AI models have ten times further to travel before they match the full intelligence of a brain?

The Bitter Lesson Again

Whatever the answer to this analysis, it seems clear that more compute is needed. It sounds like another case of the bitter lesson, doesn't it? We just need to brute-force up another 50 times greater compute power, and then it's the same as the 100 trillion synapses in the human brain.

This level of extra computation power seems beyond our reach at the moment. However, it's not unimaginable when you consider the rate of advances in GPU chip workloads and the newer, more efficient algorithms being discovered to run the AI computations on these chips.

So much for all those reasoning algorithms!

Thus, here we are on the verge of learning the bitter lesson all over again. Instead of making advances by using intelligent and insightful new methods of thinking, we're just going to throw more electricity and C++ programmers at the problem.

The Sweetest Lesson

Is it the bitter lesson? Sure, the idea is to brute-force a lot more GPUs in 100 trillion weight models. There's going to be plenty of computation power needed over the next few years as we advance the state-of-the-art models in both training and inference.

But the bitter lesson actually has two factors, of which brute-force hardware is only one part. Let's see if they both fit:

1. Brute-force computation, and

2. Algorithms that differ.

Can you see the mismatch? There needs to not only be brute-force computations, but also the abandonment of human-based reasoning in favor of simpler number crunching. At first, this also seems like a match, because those tricky reasoning algorithms will probably be gone. It'll just be GPUs and matrices and lots of number crunching.

But, then, inspiration.

Maybe we're not going to use a fancy reasoning algorithm on top of all those LLMs. The path to intelligence may not be a better "controller" that manages all of those low-level computations. So, the most advanced AI algorithms won't be using human-style reasoning. But, you know what, it's kind of weird, because here's the thing:

We already are!

There's this whole theory of neural networks that forms the basis of AI theory. For decades, it was a kind of backwater in computing research, where the results were not very impressive. Gradually, it grew in importance as the results from Machine Learning models, which is old-school AI theory, started to be used in real-world activities, like recommending which movies you should watch. Eventually, the advent of hyperscale GPUs led to the GPT series of LLMs, and the rest you already know.

Hence, this is not the bitter lesson. The whole of AI theory in the computing industry is about neural networks, which means this:

Copying the human brain.

It doesn't stop there at copying the basic human architecture. The AI industry is effectively copying all the workarounds and extensions that we use to make humans smarter:

- Tools — e.g., computer usage by LLMs.
- Data sources — e.g., LLMs now search the internet.
- Training — it's like sending LLMs to High School, where they read a textbook.

In summary, AI models are based on not just copying human brains, but also imitating all of the ways that humans learn. And it's not just the main neural network algorithm that's copied from a carbon brain. It's all of the workarounds, too. If we ever get to a truly "intelligent" model, and I do think it's an "if" not a "when" for the achievement of "Artificial General Intelligence" (AGI), then it's going to be achieved by copying everything we know about human intelligence.

And that's the sweetest lesson of all.

References

Research papers on neurology and estimates of the size of the human brain:

1. Carl Zimmer, January 2011, *100 Trillion Connections: New Efforts Probe and Map the Brain's Detailed Architecture*, Scientific American Magazine, Vol. 304, No. 1, https://www.scientificamerican.com/article/100-trillion-connections/
2. Kandel E.R.,Schwartz J.H., and Jessell T. M., January 2000, *Principles of Neural Science (4th Ed.)*, New York, McGraw-Hill, https://www.amazon.com/Principles-Neural-Science-Eric-Kandel/dp/0838577016 (Early source for the 100 trillion synapse estimate.)
3. Eric H. Chudler, July 2025 (accessed), *Brain Facts and Figures*, University of Washington, https://faculty.washington.edu/chudler/facts.html
4. Wieslaw L. Nowinski, 2024, *On human nanoscale synaptome: Morphology modeling and storage estimation*, Plos one, https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0310156, PDF: https://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0310156&type=printable (Attributes the 100 trillion synapse estimate to: Kandel et. al, 2000)

5. AI Impacts, 2015, *Scale of the Human Brain: The brain has about 10^{11} neurons and 1.8-3.2 x 10^{14} synapses*, https://aiimpacts.org/scale-of-the-human-brain/ (Calculations give 180 to 320 trillion inter-neuron connections.)
6. Alain Goriely, March 2025, *Eighty-six billion and counting: do we know the number of neurons in the human brain?*, Brain, Volume 148, Issue 3, Pages 689–691, https://doi.org/10.1093/brain/awae390, https://academic.oup.com/brain/article/148/3/689/7909879
7. Roberto Lent, May 2025, *Yes, the human brain has around 86 billion neurons*, Brain, Volume 148, Issue 5, Pages e37–e38, https://doi.org/10.1093/brain/awaf048, https://academic.oup.com/brain/article-abstract/148/5/e37/8003626
8. Wikipedia, July 2025 (accessed), *Cerebellar granule cell*, https://en.wikipedia.org/wiki/Cerebellar_granule_cell
9. Frederico A C Azevedo, Ludmila R B Carvalho, Lea T Grinberg, José Marcelo Farfel, Renata E L Ferretti, Renata E P Leite, Wilson Jacob Filho, Roberto Lent, Suzana Herculano-Houzel, 2009, *Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain*, J Comp Neurol, 2009 Apr 10;513(5):532-41, PMID: 19226510, DOI: 10.1002/cne.21974, https://pubmed.ncbi.nlm.nih.gov/19226510/ (Quote: debunking the frequently-cited statistic: "100 billion neurons and ten times more glial cells"... their findings: "...the adult male human brain contains on average 86.1 +/- 8.1 billion NeuN-positive cells ("neurons") and 84.6 +/- 9.8 billion NeuN-negative ("nonneuronal") cells...")
10. Bente Pakkenberg, Dorte Pelvig, Lisbeth Marner, Mads J. Bundgaard, Hans Jørgen G. Gundersen, Jens R. Nyengaard, Lisbeth Regeur, 2003, *Aging and the human neocortex*, Experimental Gerontology, Volume 38, Issues 1–2, Pages 95-99, ISSN 0531-5565, https://doi.org/10.1016/S0531-5565(02)00151-1, https://www.sciencedirect.com/science/article/abs/pii/S0531556502001511, https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=7c7c04323f6ba96c32c47fc3159cc30614c2e822 (Around 36-39 billion glial cells in the human brain, and around 0.15×1015 (0.15 quadrillion) synapses, which is around 150 trillion.)
11. Horn-Bochtler, A.K.E., Büttner-Ennever, J.A., 2011, *Neuroanatomy of the Brainstem*, In: Urban, P., Caplan, L. (eds) Brainstem Disorders. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04203-4_1

12. Pakkenberg, B. and Gundersen, H.J.G., 1997, *Neocortical Neuron Number in Humans: Effect of Sex and Age*, Journal of Comparative Neurology, Vol. 384, Pages 312-320, http://dx.doi.org/10.1002/(SICI)1096-9861(19970728)384:2<312::AID-CNE10>3.0.CO;2-K, https://onlinelibrary.wiley.com/doi/abs/10.1002/(SICI)1096-9861(19970728)384:2%3C312::AID-CNE10%3E3.0.CO;2-K (Human neuron counts around 19-23 billion in the neocortex, a part of the brain.)
13. Haines, D., Mihailoff, G., 2018, *Fundamental Neuroscience for Basic and Clinical Applications (5th ed.)*, Elsevier, p. 195, ISBN 9780323396325, https://www.amazon.com/dp/0323396321/
14. Jianfeng Feng, Viktor Jirsa, Wenlian Lu, May 2024, *Human brain computing and brain-inspired intelligence*, National Science Review, Volume 11, Issue 5, nwae144, https://doi.org/10.1093/nsr/nwae144, https://academic.oup.com/nsr/article/11/5/nwae144/7656427
15. Catherine Caruso, January 19, 2023, *A New Field of Neuroscience Aims to Map Connections in the Brain: Scientists working in connectomics are creating comprehensive maps of how neurons connect to one another*, https://hms.harvard.edu/news/new-field-neuroscience-aims-map-connections-brain (Quote: "86 billion neurons form 100 trillion connections to each other...")
16. Bradley Voytek, May 20, 2013, *Are There Really as Many Neurons in the Human Brain as Stars in the Milky Way?*, Nature, https://www.nature.com/scitable/blog/brain-metrics/are_there_really_as_many/
17. Suzana Herculano-Houzel, 2009, *The Human Brain in Numbers: A Linearly Scaled-up Primate Brain*, Front Hum Neurosci. 2009 Nov 9;3:31. doi: 10.3389/neuro.09.031.2009, https://pmc.ncbi.nlm.nih.gov/articles/PMC2776484/