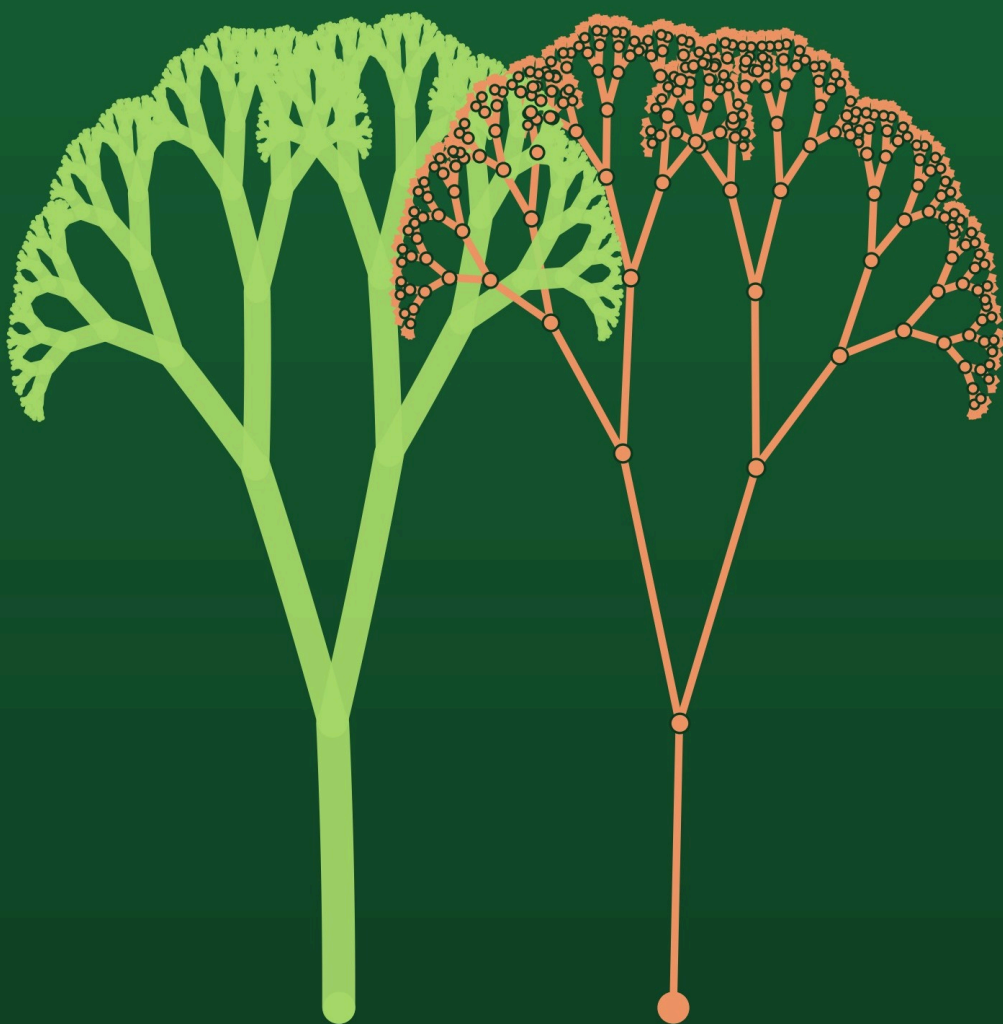


Life and AI

Understanding Intelligence
in Biology and Models



VICTOR JARAVINE

Life and AI

Understanding Intelligence in Biology and Models

Victor Jaravine

First edition · October 2026

© 2026 Victor Jaravine. All rights reserved.

*More life, in more ways, for longer, understood ever
better.*

Contents

Preface.....	7
Part I — How Machines Know.....	11
Chapter 1. What Models Know and How They Behave.....	12
Chapter 2. Computed in Advance, Computed in the Moment.....	19
Chapter 3. The Price of Computation.....	24
Chapter 4. Thinking Longer.....	29
Chapter 5. Context and Memory.....	37
Part II — Understanding.....	43
Chapter 6. What Is Understanding?.....	44
Chapter 7. Understanding Other Minds.....	52
Chapter 8. Understanding Ourselves.....	60
Chapter 9. Explanation and Trust.....	65
Part III — Models, Truth and Limits.....	70
Chapter 10. Beyond the Training Data.....	71
Chapter 11. Causes, Not Just Correlations.....	77
Chapter 12. Symbols and Vectors.....	84
Chapter 13. When Translation Keeps the Truth.....	89
Chapter 14. Does a Model Tell the Truth?.....	94
Part IV — Stability and Change.....	100
Chapter 15. The Constant and the Changing.....	101
Chapter 16. Stable Ground, Small Storms.....	108
Part V — Minds, Life and Machines.....	115
Chapter 17. What Accumulates.....	116
Chapter 18. Inside and Outside.....	123
Chapter 19. Natural and Artificial.....	128
Chapter 20. Is Life One Big Computation?.....	134
Chapter 21. Is Anyone Home?.....	139
Chapter 22. Continuous Flow, Discrete Vessels.....	144

Part VI — Evolution, Aims and Paths.....	150
Chapter 23. Errors, Compression and Crashes.....	151
Chapter 24. One Process, Opposite Faces.....	157
Chapter 25. Why Not Go Straight to the Goal?.....	162
Chapter 26. What Is Life For?.....	170
Part VII — Where We Stand.....	178
Chapter 27. Conflict and Compatibility.....	179
Chapter 28. Have We Already Converged?.....	185
Chapter 29. The Price of a Big Brain.....	190
Part VIII — The Machinery of AI.....	195
Chapter 30. What to Fix, What to Remember.....	196
Chapter 31. Good Enough for the Real World?.....	203
Part IX — How AI Grows.....	210
Chapter 32. How Smart Can Intelligence Get?.....	211
Chapter 33. A Quantum Leap?.....	217
Chapter 34. Merging Minds.....	221
Chapter 35. Can AI Create Knowledge?.....	227
Chapter 36. Where the Acceleration Leads.....	233
Part X — What Keeps a System Open to Change.....	243
Chapter 37. How Error Correction Works.....	244
Chapter 38. Who Will Check the Machines?.....	250
Chapter 39. How Niches Survive.....	257
Chapter 40. Can Wisdom Accumulate?.....	263
Chapter 41. The Person in Between.....	268
Chapter 42. The Work of Understanding.....	272
Epilogue — The Thread and the Open Questions.....	277
Appendix A — How a Language Model Works.....	285
Appendix B — How Models Are Classified.....	290
Appendix C — Brain Rhythms.....	292
Appendix D — Glossary.....	294
Appendix E — Further Reading.....	307

Appendix F — Index.....370

Chapter 4. Thinking Longer

A model can get its depth in two ways: by stacking many layers, up to 80 in today's large models, or by having fewer layers and running the data through them several times. And is that why a smaller model that loops more can beat a larger one?

A model's size is settled the day training ends. Its weights are frozen, its layers counted, and nothing you type will add a single parameter. Yet a smaller model allowed to think for longer can sometimes beat a larger one that answers at once. Depth, it turns out, can be bought with time as well as with size, and brains have been buying it that way all along. The question is when the trade works.

Many layers, or fewer but looped

A layer in a transformer doesn't replace what it receives. It adds a small correction to a shared running vector and passes the sum along. Eighty layers therefore already work much like one kind of step repeated 80 times: a draft handed down a line of editors, each allowed a pencilled note in the margin, none allowed to start again. Two things separate this from a true loop. Each of the 80 editors has its own weights. And the length of the line is fixed when the model is built.

So stacking and looping are close relatives, and whether they give the same result can be tested by building both. A team led by Nikunj Saunshi trained a 24-layer model and a 12-layer one that ran twice through each of the layers, just as deep but with half the weights. The looped model did better on maths word problems, 34.3 points against 29.3 on the study's average score, and on basic reasoning tasks, 51.2 against 47.5. On questions testing stored facts it did worse, 9.3 against 11.2. The authors concluded that many reasoning problems need depth more than parameters.

Not the same, then, but a revealing split. Facts need weights. Reasoning needs depth, and looping can supply it.

The shallow, looping brain

The brain sits at the other end of that choice, with few stages and many loops. The neocortex has six layers of cells, and between the eye and the recognition of an object there are only a handful of stages. Its neurons are slow, a few milliseconds per step. Recognising something at a glance takes 100 to 150 milliseconds, time for only about ten steps in a row, roughly one pass through a shallow network. Yet feedback connections between cortical areas are roughly as numerous as forward ones.

The loops show when the brain is handed something hard. A study led by Kohitij Kar and James DiCarlo used images that monkeys and people recognise easily but a standard deep vision network could not. In the monkeys' object-recognition area the answer to those harder images emerged about 30 milliseconds later than for easy ones, after extra rounds of looping. Networks without loops matched those late responses only when made much deeper.

The loops also keep time, in rhythms charted in Appendix C: fast gamma waves carry signals forward, slower alpha waves carry them back. The neuroscientist John Lisman proposed that each fast cycle nested inside one slow theta cycle holds one item of working memory, leaving room for about four to seven. It's a lovely idea, a memory span set by the ratio of two clocks, and it is still a hypothesis.

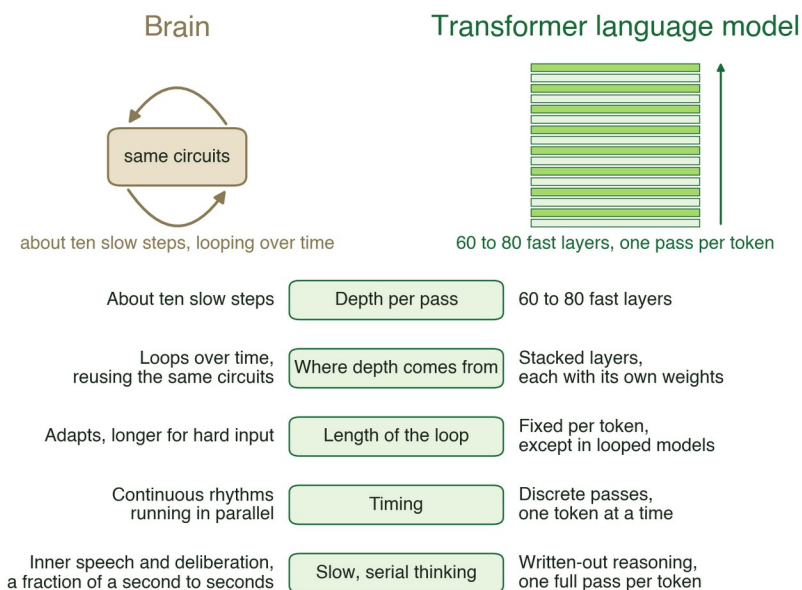


Figure 4. Depth in a brain and in a transformer. The brain loops the same circuits over time; the model stacks layers, each with its own weights.

The brain's mix of slow and fast rhythms invited a copy in silicon. The Hierarchical Reasoning Model of 2025 paired a slow planning module with a fast one that took several steps for each of its own. With 27 million parameters it beat far larger language models on hard puzzles and scored 40.3% on the ARC-AGI-1 benchmark.

Months later the researcher Alexia Jolicoeur-Martineau stripped it down to a single two-layer network of 7 million parameters that simply loops on its own answer. It scored 44.6%. The gain had come from the extra looping, not the two speeds. Biology had supplied the right hint, though not the one the designers took from it.

Depth bought with time

With its layers fixed, a single pass can only solve problems that need a bounded number of steps in a row. Writing out reasoning changes that. Each token goes back in as input, so the text becomes working memory, and every new token gets a full pass built on everything before it. The

computer scientists William Merrill and Ashish Sabharwal proved that with a polynomial number of such steps a transformer can solve exactly the problems an ordinary computer solves in polynomial time, which a single pass cannot.

A bigger model is wider and deeper, but its depth per token is still finite. A longer chain of reasoning is depth bought with time, and for serial problems, where step seven needs the result of step six, time is the resource that counts.

One recent model, from Jonas Geiping and colleagues, takes the looping of the last section to full scale: 3.5 billion parameters built around one block of layers that it repeats a flexible number of times. Run once, it solved none of a standard set of grade-school maths problems. Run 32 times, it solved over a third, ahead of a model twice its size trained on three times as much text. On every turn of the loop, the original input is fed back in.

Overthinking

Why that detail matters had shown up earlier, in small networks. A group including Arpit Bansal, Avi Schwarzschild and Tom Goldstein trained looped networks on easy problems, then let them loop longer on hard ones. Without the original input on every turn, the loop gets only its own output. After a certain number of turns, it drifts and the answers fall apart. The authors called this overthinking.

Networks trained on short problems solved none of problems sixteen times longer without the input fed back, and nearly all of them with it.

There's a loose human parallel. In one study, 15 minutes in a dark, soundproof chamber was enough for healthy volunteers to report hallucinations, paranoid thoughts and low mood, most of all those already prone to unusual experiences in everyday life. A mind cut off from its input and left working on its own output starts to see things that aren't there. The shape of the failure is the same: nothing outside to pull the diverging drift back.

Copies of copies

Can the forgetting in a chain of language models, each working only from the previous one's output with no retrieval of the original sources, be compared with a loop that runs without feeding its input back in? How much is lost, and after how many steps?

It can, and the line of editors from the start of the chapter shows why. Give each editor only the previous editor's copy, never the original, and every pencilled change is made to something already slightly wrong. Information theory makes it exact: in a chain where each step sees only its predecessor, the data processing inequality says that information about the source can only shrink, never grow back. Feeding the source back in breaks the chain. For a loop that means recalling the input; for a model, retrieving the original documents.

How much is lost has been measured. A team led by Amr Mohamed had a model translate news articles from English into another language and back, a hundred times over, each round working from the last one only. Similarity in meaning to the original, on a scale where 1 means the same, went through French from 0.95 to 0.73 within three rounds. Through Thai it was 0.58 after three rounds.

One of the articles makes the numbers concrete. An original story: a lorry driver was fined after his load of slabs fell off on a bend and wrote off a passing £50,000 car. After a hundred rounds through Thai, the story changed to this: a small car had received \$50,000 in compensation after a rock hit it and caused an explosion. The money, the car and something heavy all made it through. The story didn't.

Text is also the narrower bottleneck: whatever a model knew but didn't write is gone at once, and what survives each round is what the system finds typical, which may be how an explosion found its way into a story about a fine. For loops and chains alike, the fix is the same: keep the input from the source in every step.

Checking is cheaper than finding

Time can also be spent on many attempts instead of one long one, with a test to pick the winner. That rests on the intuition behind P versus NP, that checking an answer is often far easier than finding one. In the study cheerfully titled “Large Language Monkeys”, a model fixing real software bugs solved 15.9% with a single attempt and 56% with 250 attempts, ahead of the 43% that larger frontier models reached in one try. A weaker model with a test suite and patience beat stronger ones without them.

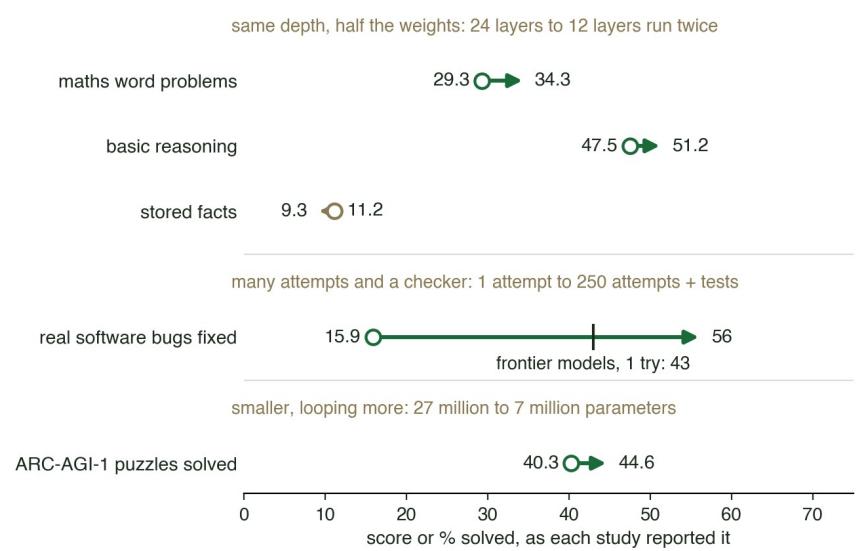


Figure 5. Depth bought with time, three ways: the same depth from half the weights by running each layer twice; many attempts plus a test suite; and a smaller network looping on its own answer. Scores as each study reported them, on different scales.

Checking works inside a single answer too. A model that verifies each step as it goes catches errors before they compound, much as a cell proofreads DNA while copying it. The catch is where the check comes from. It has to be something real: a test, a proof checker, a calculation. Models asked only to reconsider their own answers, with no outside feedback, often fail to improve and sometimes get worse. Rereading

your work with the same eyes that wrote it is another loop running without its input.

Training and thinking can be swapped

If thinking can stand in for size, there should be an exchange rate. For AlphaZero playing the board game Hex it was roughly so: each extra order of magnitude of training compute could replace about an order of magnitude of search. For language models, a study in 2024 held total compute equal and found that a small model thinking longer beat one fourteen times its size, but only on problems the small model could already solve some of the time. On the hardest problems, size still won. Thinking longer seems to sharpen an ability the model has rather than conjure one it lacks.

Thinking itself is learnt. Reinforcement learning on problems with checkable answers teaches a model when to backtrack, when to verify and when to stop. And the skill can be passed on. DeepSeek fine-tuned a 7-billion-parameter model on reasoning written by its large R1 model, and it scored 55.5% on the 2024 American Invitational Mathematics Examination, against 9.3% for GPT-4o.

Cooking to order

A reasoning model cooks to order. It pays at every use, in time and energy, what a larger model paid once in training, when it did its batch cooking. That's the trade-off of Chapter 3, moved from the prep kitchen to the moment of serving. It's worth paying where the dish is new and the result can be tasted: a proof that checks, code that passes its tests, a maze that ends at the exit. Where nothing can taste the dish, thinking longer mostly produces a more confident guess.

Sources for this chapter (details in Appendix E): Bansal (2022); Brown (2024); DeepSeek-AI (2025); Geiping (2025); Huang (2024); Jolicoeur-Martineau (2025); Jones (2021); Kar (2019); Lisman (1995, 2013); Mason

(2009); Merrill (2024); Mohamed (2025); Saunshi (2025); Snell (2024); van Kerkoerle (2014); Wang (2025).