Ubiquitous, not universal: the limits of AI scaling

Valentin Radu

2026-01-02

Table of contents

Preface

The dominant narrative in 2024-2026 was exponential progress toward artificial general intelligence within a decade. This book measures the distance to that goal, identifies the constraints that bind, and projects what we will actually achieve.

The approach is simple: measure how much computation nature spent to produce general intelligence, measure how much computation we are spending on AI, identify the constraints that prevent closing the gap, and trace where efficiency improvements lead when capability scaling stalls.

The evidence suggests we will not reach AGI on the current path. Frontier capabilities stagnate at "impressively competent within distribution, fragile outside." But while capability scaling fails, efficiency explodes: GPT-4 quality moves from datacenters to laptops, then smartphones, then all devices.

Part I

Nature's invoice

Chapter 1

From chaos to cognition

"Nothing in biology makes sense except in the light of evolution." — Theodosius Dobzhansky

1.1 The computational observation

We have exactly one confirmed example of general intelligence in the known universe. Not one architecture, not one algorithm: one entire process. It spans roughly 3.8 billion years of evolution, operating across trillions of organisms in parallel, each testing a slightly different strategy against the hard constraints of a physical world. Before we debate whether machines can think, we should ask the more elementary question: how much did the only known solution cost?

This chapter attempts an answer. Specifically, we want to measure the computational distance from a state of high cognitive entropy, where no organism reasons, plans, or models the world, to the low-entropy state where general intelligence emerges. Not the cost of building the machinery that makes cognition possible, the molecular substrates, the basic neural wiring, but the cost of the journey itself.

We will estimate this distance in *learning instances*: cycles of interaction between organisms and their environment from which adaptive information was extracted. The number we arrive at is large. It may also be wrong by several orders of magnitude in either direction. Let us be honest about that from the start: this estimate is impossible to get right. The organisms are dead, the environments are gone, and the quantities we need were never measured directly. What we can do is assemble the best numbers that biology, neuroscience, and ecology have produced, chain them together carefully, and see where they land.

Our prediction is that traversing this distance required between 10^{25} and 10^{40} learning instances, depending on how much of evolution's work we classify as "building machinery" versus "making the journey." As we will show, the largest artificial training runs, measured in comparable units, have performed roughly 10^7. The gap demands explanation.

1.2 The ground rules

Throughout this estimate, and throughout this book, we adopt a single methodological principle: **when in doubt, choose the number that makes the gap smaller.** Every uncertain parameter is resolved in favor of the optimist. Every ambiguous definition is read in the way most generous to current AI. If, after all this generosity, the remaining gap is still large, the argument is strengthened precisely because we did not inflate it.

This is a lower bound argument. We are not trying to prove that AGI is impossible. We are trying to find the floor: the smallest defensible estimate of what nature spent, using every reasonable discount and concession. If that floor is still far above what artificial systems have achieved, we have learned something important.

1.3 What are we counting?

The tempting metric is synaptic operations: count every time a neuron fires, across every organism, across all of evolutionary time. But this overcounts in a way that distorts the picture. A lizard basking on a rock for an hour has neurons firing continuously, maintaining homeostasis, regulating body temperature, keeping its heart beating. Very little of that activity constitutes learning. If we want to know what it would cost to replicate evolution's achievement in silicon, raw neural firing is not the right unit. We do not need to simulate a lizard's heartbeat.

The unit we want is a *learning instance*: a cycle of interaction between an organism and its environment from which adaptive information can be extracted. For an organism with a nervous system, this means a meaningful encounter: a predator detected, a food source located, a mate assessed, a threat survived or not survived. For a bacterium, which has no neurons at all, the learning instance is a generation in which a genuinely novel genetic variant is tested against the environment: a new mutation expressed, a new strategy tried, an outcome recorded by natural selection.

This is what we would actually need to replicate. Not the idle hum of a billion neural circuits, but the moments where organism meets world and something is at stake. The failed experiments count too: evolution has no foresight, and an organism that dies in infancy is as much a data point as one that reproduces. A genetic algorithm does not get to subtract the fitness evaluations of discarded candidates from its computational budget.

We split the estimate into three tiers: the long pre-neural era dominated by microbial life, the neural era beginning with the Cambrian explosion roughly 500 million years ago and dominated by invertebrates, and the relatively brief era of vertebrate sophistication.

1.4 Tier 1: The microbial foundation

Life originated approximately 3.8 billion years ago. For the first 3.3 billion years, until the Cambrian explosion roughly 500 million years ago, Earth was dominated by bacteria and archaea: organisms with no nervous systems, no neurons, no synapses. They could not learn within their lifetimes in any neural sense. They executed their genetic programs and either divided or died.

Yet this era was not computationally idle. It was, by any measure, the most prolific optimization process in Earth's history. The molecular machinery that neurons would later use, the entire signaling and communication infrastructure of neural computation, was forged during this period through the brute trial and error of microbial evolution.

Timeline:

$$T_1 = 3.3 \times 10^9 \text{ years} \times 3.15 \times 10^7 \text{ s/year} \approx 1.04 \times 10^{17} \text{ s}$$

Population: Whitman, Coleman, and Wiebe estimated in their landmark 1998 census that the number of prokaryotic cells alive on Earth at any given time is approximately 4–6 × 10^{30}. Bar-On and colleagues revisited global biomass in 2018 and broadly confirmed the order of magnitude. We use 10^{30}.

Learning rate: Here we apply our first optimistic discount. A bacterium divides, on average, every few hours. We use $\tau \approx 10^4$ seconds (roughly three hours) as a reasonable mean across species and conditions. But most divisions produce offspring nearly identical to the parent. The "learning" happens only when a genuinely novel variant is tested: a new mutation expressed against the environment. Drake's foundational work on mutation rates established that bacteria mutate at approximately $\mu \approx 0.003$ mutations per genome per generation, a figure that has held up across decades of subsequent measurement. We count only these novel-variant generations:

$$R_1 = \frac{\mu}{\tau} = \frac{0.003}{10^4} = 3 \times 10^{-7} \text{ instances/s/organism}$$

Subtotal:

$$\Omega_1 = N_1 \times R_1 \times T_1 = 10^{30} \times 3 \times 10^{-7} \times 1.04 \times 10^{17}$$

$$\Omega_1 \approx 3 \times 10^{40}$$

Three followed by forty zeros. We will verify this number shortly.

1.5 Tier 2: The invertebrate era

The Cambrian explosion, roughly 500 million years ago, marks the appearance of nervous systems in the fossil record. From this point forward, organisms could learn within their own lifetimes: adjusting behavior in response to experience, not merely across generations through genetic variation.

Timeline:

$$T_2 = 500 \times 10^6 \text{ years} \times 3.15 \times 10^7 \text{ s/year} \approx 1.58 \times 10^{16} \text{ s}$$

Population: The number of animals with nervous systems alive at any time is dominated by invertebrates. Nematodes alone number in the hundreds of trillions. The Entomological Society of America and the Smithsonian estimate roughly 10^{19} individual insects alive at any given time, a figure consistent with Williams's earlier census work. Fish, amphibians, reptiles, birds, and mammals are rounding errors against this population.

$$N_2 \approx 10^{19} \text{ organisms}$$

Learning rate: This is the hardest parameter in the estimate. How often does the average invertebrate encounter a genuinely novel stimulus, one that triggers actual learning rather than habituated routine? No one has measured this directly in the wild. But we can triangulate from the species whose learning has been studied most carefully, then ask what the population-weighted average should be.

Menzel and colleagues documented that foraging honeybees learn and retain flower colors, shapes, scents, locations, reward quality, time-of-day patterns, navigation landmarks, and routes. A forager makes 10 to 15 trips per day and accumulates this repertoire, hundreds of distinct learned items, over roughly three weeks of active foraging. That implies roughly 10 to 15 novel learned associations per day. The nematode *C. elegans*, with only 302 neurons, demonstrates habituation, sensitization, and associative learning across its two-to-three-week lifetime, as documented by Rankin and others; back-calculating from its known behavioral repertoire yields a comparable rate of roughly 7 to 14 learned items per day. Drosophila shows robust associative conditioning in the lab, with synaptic plasticity in the mushroom body operating on timescales of tens of seconds, though the ecological relevance of this rate in the wild remains debated.

These are the best-studied invertebrate learners, and they converge on roughly 10 novel learning events per day during active life. The 10^{19} population is dominated by insects, so the population-weighted average is driven by insect learning rates. Ten per day sits at the conservative end of what the data from studied species suggests:

$$R_2 = \frac{10}{86400} \approx 10^{-4} \text{ instances/s/organism}$$

We can sanity-check this against structural data. Neurons connect to each other through synapses, and most excitatory synapses in the brain sit on tiny protrusions called dendritic spines. When the brain learns, spines grow, shrink, appear, and disappear; the rate at which this happens, spine turnover, is the most direct measurable proxy for learning-related synaptic change. Pfeiffer and colleagues measured roughly 10% spine turnover per day in the mouse hippocampus. Trachtenberg and colleagues found roughly 0.5 to 1% per day in the mouse cortex. For a typical invertebrate with 10^4 to 10^7 synapses and a turnover rate in this range, we would expect roughly 10^1 to 10^4 individual spine changes per day. At ten learning instances per day, that implies roughly 1 to 1,000 spine changes per learning instance: a single episode of learning rewiring a handful to a thousand synapses. This is consistent with what neuroscience observes for associative learning in small nervous systems.

Subtotal:

$$\Omega_2 = N_2 \times R_2 \times T_2 = 10^{19} \times 10^{-4} \times 1.58 \times 10^{16}$$

$$\Omega_2 \approx 10^{31}$$

1.6 Tier 3: The vertebrate refinement

Vertebrates learn more per individual than any invertebrate. A crow solving a novel puzzle, a rat navigating a maze, a dolphin coordinating a hunt: these are learning-dense lives. But vertebrate populations are tiny compared to invertebrates, and they arrived late. The first vertebrates, small jawless filter-feeders, appear in the Cambrian fossil record roughly 525 million years ago, but they did not diversify substantially until the Devonian. We generously credit them with the full post-Cambrian timeline.

Timeline:

$$T_3 \approx 1.58 \times 10^{16} \text{ s}$$

Population: Callaghan, Nakagawa, and Cornwell estimated roughly 50 billion wild birds alive at any given time. Greenspoon and colleagues at the Weizmann Institute estimated approximately 130 billion wild mammals, dominated by bats (roughly 56 billion) and rodents (roughly 25 billion). Reptile and amphibian populations are less precisely known but on the order of 10^{11}. Fish dominate: estimates range from one to three trillion individuals, depending on assumptions about mesopelagic species. We use the conservative lower bound:

$$N_3 \approx 10^{12} \text{ organisms}$$

Fish outnumber all other vertebrates combined by roughly an order of magnitude. The population-weighted average vertebrate is a fish.

Learning rate: As with invertebrates, we triangulate from the species whose learning has been most carefully documented, then ask what the population-weighted average should be.

Fish are capable of rapid, flexible learning. Blank and colleagues showed that adult zebrafish form NMDA-dependent long-term memories from a single aversive experience: one-trial inhibitory avoidance. Rodriguez and others documented that goldfish learn spatial tasks in four-arm mazes, using both place-based and cue-based strategies, and can take spontaneous shortcuts suggesting map-like spatial representations. In the wild, a fish navigates its territory, locates food, avoids predators, and interacts with conspecifics. How many of these encounters constitute genuinely novel learning? For a schooling pelagic fish whose daily routine is largely repetitive, the number is modest: perhaps a few dozen per day. For an actively foraging territorial species, it may reach the low hundreds.

Among birds, the food-caching corvids and parids provide the most striking quantitative data. Balda and Kamil documented that a single Clark's nutcracker caches 22,000 to 33,000 pine seeds across 5,000 to 6,000 distinct locations during autumn, recovering them with high accuracy up to 285 days later: roughly 100 to 200 novel spatial memories per day during the caching season. Applegate and Aronov showed that black-capped chickadees cache hundreds of food items daily, each in a unique site that generates a distinct hippocampal firing pattern. These are exceptional species, not typical vertebrates, but they demonstrate the ceiling of vertebrate individual learning capacity.

Among mammals, O'Keefe and Dostrovsky's discovery of hippocampal place cells established that rodents form new spatial representations within seconds of entering a novel environment: a single pass through a new location is sufficient to generate a stable place field. Fear conditioning is reliably one-trial: a single aversive event produces long-term contextual memory. But the 130 billion wild mammals are dominated by bats and small rodents, whose daily learning budgets in their natural habitats are far below these laboratory demonstrations of capacity.

The population is dominated by fish, and the population-weighted average is driven by fish learning rates. We estimate roughly 100 novel learning events per day for the average vertebrate: an order of magnitude above the invertebrate rate, reflecting greater neural complexity, but discounting for the fact that most fish and small mammals spend much of their time in repetitive behavioral routines:

$$R_3 = \frac{100}{86400} \approx 10^{-3} \text{ instances/s/organism}$$

Subtotal:

$$\Omega_3 = N_3 \times R_3 \times T_3 = 10^{12} \times 10^{-3} \times 1.58 \times 10^{16}$$

$$\Omega_3 \approx 10^{25}$$

Even if we raise the learning rate by an order of magnitude, to a thousand novel events per day (consistent with what caching birds and actively exploring rodents achieve), the total reaches only 10^{26}. The population deficit is decisive: 10^{12} vertebrates cannot overcome a 10^{19}-strong invertebrate population, regardless of how much more each individual learns. Vertebrates matter enormously for the *quality* of intelligence that evolution produced, but they are a rounding error in the *quantity* of learning instances.

1.7 The total

Tier	Learning instances
Pre-neural (bacteria, mutation-discounted)	3×10^{40}
Invertebrates (behavioral, optimistic)	$\sim 10^{31}$
Vertebrates	$\sim 10^{25}$
Full total	$\sim 3 \times 10^{40}$

The pre-neural phase dominates by nine orders of magnitude. The microbial era, with its vast populations and relentless generational turnover, performed the overwhelming bulk of evolution's optimization work. The neural era, for all its sophistication, is a refinement.

But this raises an important question.

1.8 From chaos to cognition

We have computed the total across all three tiers. But not all of that computation is equally relevant to the question we are asking.

Consider what we are actually trying to measure: the computational distance from a state of high cognitive entropy, where no organism reasons, plans, or models the world, to the low-entropy state where general intelligence emerges. The first two tiers did not traverse that distance. They built the machinery that makes the traversal possible. Bacteria assembled the molecular substrate of computation. Invertebrates developed the basic algorithms of learning: habituation, conditioning, sensory integration, spatial memory. These are the mechanics, the engine and the chassis. They are not the journey.

The journey, the actual transition from "learning exists" to "general intelligence exists," happened during the vertebrate era. It happened in organisms that inherited a working nervous system with established learning algorithms, and then used those tools, across 500 million years and a trillion parallel lives, to develop abstract reasoning, social cognition, planning, and flexible problem-solving.

This distinction matters because it is substrate-independent. We are not asking whether silicon can replicate biology's specific molecular machinery, or whether backpropagation is equivalent to synaptic plasticity. We are asking a simpler question: how much computation separates cognitive disorder from cognitive order? The vehicle, biological or artificial, provides the capacity to compute. The vertebrate learning instances are the computation itself.

Silicon provides its own mechanics. At the substrate level: transistors, memory hierarchies, GPU architectures, decades of computer science and engineering. At the algorithmic level: backpropagation, attention mechanisms, reinforcement learning, and the mathematical theory that underpins them. Whether these mechanics are equivalent to biology's is genuinely debatable, and we will return to that question in later chapters. But even granting the equivalence, the distance remains. The question is how far, not how.

$$\Omega_{\text{vertebrate}} \approx 10^{25}$$

This is the maximally optimistic estimate: the computation that occurred after both foundations were in place, during the era when general intelligence actually emerged. But what is the right number on the AI side?

The conventional comparison uses floating-point operations: frontier language models consume roughly 10^{25} FLOP during training, and 10^{25} against 10^{25} would suggest the gap is already closed. But this comparison is inconsistent with our own methodology.

We rejected total synaptic operations as the metric for biology because most neural activity is not learning: it is maintenance, homeostasis, routine processing. The same logic applies to FLOP. The vast majority of floating-point operations in a training run are spent on the forward and backward passes: computing gradients, not applying them. The moment the model actually learns, the moment its weights change, is the gradient update step. Everything else is computation in service of that step, just as a lizard's routine neural firing is activity in service of staying alive, not learning. If we insist on counting only the moments that matter on the biology side, intellectual honesty demands we do the same on the silicon side.

A frontier training run processes roughly 10^{13} tokens in batches of several million, yielding approximately 10^{6} to 10^{7} gradient update steps. Each step is one adjustment: the model sees a batch of data, computes how wrong it was, and updates its parameters. That is the atom of learning in stochastic gradient descent, just as a learning instance is the atom of learning in biology.

The honest comparison is adjustments to adjustments:

$$\frac{\Omega_{\text{vertebrate}}}{\Omega_{\text{SGD}}} \approx \frac{10^{25}}{10^{7}} = 10^{18}$$

Even at maximum generosity, the gap is eighteen orders of magnitude.

For now, we note the range. The full estimate, counting all three tiers, is $\sim 10^{40}$. The vertebrate-only estimate, measuring only the distance from cognitive chaos to cognition, is $\sim 10^{25}$. Against $\sim 10^{7}$ gradient updates, the gap ranges from eighteen to thirty-three orders of magnitude.

1.9 Verification

We derived these numbers from populations, rates, and timelines. Can we check them against something independent? Two approaches.

1.9.1 Energy consistency

If our learning instance count is correct, the energy cost per instance should be physically plausible. We can estimate total neural energy expenditure independently and divide.

Attwell and Laughlin's foundational work on the brain's energy budget established that a single synaptic transmission event costs roughly 8×10^{-15} joules. Mink, Blumenschine, and Adams showed that the ratio of central nervous system metabolism to body metabolism is remarkably constant across vertebrate classes, at roughly 2 to 8 percent. Herculano-Houzel demonstrated that glucose consumption per neuron is nearly constant across rodent and primate species, varying by only 40 percent.

During the neural era, average biosphere power was roughly 70 TW (Hoehler and colleagues). Animals consume approximately 5% of biosphere energy. Neural tissue accounts for roughly 5% of animal energy on a population-weighted basis (lower than the vertebrate average, since invertebrate nervous systems are proportionally smaller). Total neural energy:

$$E_2 \approx 70 \times 10^{12} \times 0.05 \times 0.05 \times 1.58 \times 10^{16} \approx 3 \times 10^{25} \text{ J}$$

Energy per learning instance: $3 \times 10^{25}/10^{31} = 3 \times 10^{-6}$ J, or about 3 microjoules. A small insect brain consumes roughly 10^{-5} joules per second. At that rate, 3 microjoules is about 0.3 seconds of full brain activity: a brief sensory-motor cycle. Physically plausible.

1.9.2 Behavioral plausibility

If our per-organism learning rate (10^{-4} instances per second) is correct, individual organisms should accumulate a reasonable number of learning instances over their lifetimes. We can check this against species whose learning has been studied in detail.

Honeybee. A forager bee lives roughly six weeks, with about three weeks of active foraging. At 10^{-4} instances per second, it accumulates roughly 200 to 400 learning instances over its lifetime. The literature on honeybee cognition, particularly the work of Menzel and colleagues, documents that foraging bees learn and retain flower colors, shapes, scents, locations, reward quality, time-of-day patterns, panoramic landscape views, compass directions, and navigational routes. Several hundred distinct learned items over a lifetime is consistent with this.

C. elegans. The nematode *C. elegans* lives two to three weeks. At 10^{-4} instances per second, it accumulates roughly 100 to 200 learning instances. Despite having only 302 neurons (the complete connectome was mapped by White and colleagues in 1986), *C. elegans* demonstrates habituation to mechanical and chemical stimuli, sensitization, associative learning linking temperature, smell, and taste to food availability, and both short-term and long-term memory, as documented extensively by Rankin and others. One to two hundred learning events over a lifetime is plausible for this repertoire.

Mouse. A mouse lives roughly two years. Mice are among the most learning-intensive vertebrates: place cells form within seconds of encountering a novel environment, and fear conditioning is reliably one-trial. In the wild, a mouse explores its territory, forages, avoids predators, and navigates social hierarchies, plausibly encountering several hundred novel learning situations per day. At a conservative 200 events per day, well below what laboratory studies of hippocampal plasticity suggest the brain can handle, a mouse accumulates roughly 150,000 learning instances over its lifetime. Mice in complex environments learn spatial maps, food cache locations, social hierarchies, predator avoidance strategies, and hundreds of contextual associations. Over a hundred thousand learning events across a two-year life in a rich environment is reasonable.

These checks do not prove the estimate is correct. They demonstrate internal consistency: the per-organism rate produces lifetime learning counts that match what behavioral science has documented.

1.10 The human question

The vertebrate calculation treats all 10^{12} organisms as contributing equally to the journey toward general intelligence. But this obscures an important detail: humans are qualitatively different. Language, abstract reasoning, cumulative culture, technology—these capabilities emerged very recently and in a very small population. How much of the 10^{25} learning instances was specifically required for human-level cognition?

The genus *Homo* appeared roughly 2.5 million years ago. Anatomically modern humans (*Homo sapiens*) emerged approximately 300,000 years ago. Behavioral modernity—language, art, complex tools, symbolic thought—is evident in the archaeological record only within the last 100,000 years. For most of vertebrate history, the most sophisticated organisms were nowhere near human intelligence.

If we isolate the human lineage specifically, the calculation narrows dramatically. Human population over the last 100,000 years averaged perhaps 10^6 to 10^7 individuals (peaking only recently at 10^{10}). At 100 learning instances per day over 100,000 years:

$$\Omega_{\text{human}} \approx 10^7 \text{ humans} \times 10^2 \text{ instances/day} \times 365 \text{ days/year} \times 10^5 \text{ years}$$

$$\Omega_{\text{human}} \approx 4 \times 10^{16}$$

This is nine orders of magnitude smaller than the full vertebrate estimate. Does this mean the gap to human-level intelligence is only $10^{16}/10^7 \approx 10^9$—“merely” a billion-fold?

No, for two reasons. First, humans inherited the neural machinery that the previous 500 million years of vertebrate evolution built. The 10^{16} human learning instances operated on top of a substrate that cost 10^{25} instances to develop. You cannot train a human brain from random initialization; you need the machinery evolution built.

Second, and more fundamentally, we do not know how much of human intelligence emerges from individual learning versus evolutionary optimization. Language capacity, for instance, appears to have significant innate structure (Chomsky’s universal grammar, though debated in details, captures a real phenomenon: children acquire language with surprisingly little data). This innate structure was itself shaped by evolution, operating over millions of generations. The 10^{16} human learning instances sit atop an evolutionary foundation that we cannot bypass.

The honest answer is that the human-specific portion of the journey is difficult to isolate. What we can say with confidence is that the full vertebrate estimate of 10^{25} learning instances represents the cost to go from “no general intelligence” to “human-level general intelligence” starting from established neural learning mechanisms. If we had those mechanisms in silicon—the consolidation cycle, the co-located memory, the architectural substrate—perhaps 10^{16} or even 10^{18} learning instances would suffice. But we do not have those mechanisms, and building them is part of the problem, not a solved prerequisite.

1.11 Where current AI fails

If the learning instances gap is real, it should be visible in failure modes: tasks that reveal the boundaries of what 10^7 gradient updates over text can learn. The failures are indeed visible, and they cluster in predictable ways.

Novel physical reasoning. Ask GPT-4: “I have a cup of water. I turn the cup upside down. Where is the water?” The model answers correctly—it has seen this pattern in text. But ask: “I have a cup of water with a plate balanced on top. I turn the cup upside down, then remove my hand. Where is the water?” The model struggles. This is a trivial problem for a toddler who has spilled water hundreds of times, but text rarely describes this specific configuration. The model has learned the statistical regularities of water-related text, not the physics of water.

Causal inference beyond correlation. Show the model data: “Every day the rooster crows, then the sun rises.” Ask: “Does the rooster cause the sun to rise?” The model answers no, because it has seen text explicitly stating that correlation is not causation. But present a novel correlation without explicit annotation, and the model confuses the two. It has learned to parrot the distinction when prompted, but not to reliably apply it.

Compositional generalization. Train the model on sentences like “the red triangle is above the blue circle” and “the green square is to the left of the yellow star.” Then ask: “Draw a scene with the purple pentagon above the orange hexagon.” The model has seen all the component concepts, but combining them in a novel configuration often fails. Human children, by contrast, effortlessly generalize compositional structure after a handful of examples, because their learning is grounded in embodied interaction with objects in space.

Out-of-distribution robustness. Adversarial examples expose this starkly. Change a single pixel in an image, imperceptible to humans, and the classifier flips from “panda” to “gibbon” with high confidence. Add a small sticker to a stop sign, and an autonomous vehicle misclassifies it. These are not edge cases; they reveal that the model has learned statistical regularities in the training distribution, not robust concepts grounded in the structure of the world.

Common sense in unfamiliar contexts. Ask: “If I put my phone in the fridge, will it get cold?” The model answers yes. Ask: “If I put my phone in the fridge for three days, will it spoil?” The model may say no, because phones do not spoil. But ask: “If I put my phone in the fridge, then take it out into a hot humid room, what happens?” The model often misses that condensation will form and potentially damage the phone. This is trivial common sense for anyone who has experienced humidity, but text rarely describes this specific scenario. The 10^9 bits per second of lived experience is missing.

These failures have a common structure: they occur where the model must generalize beyond the statistical regularities it has seen in text to the underlying causal, physical, or compositional structure of the world. This is precisely what we would predict from the learning instances gap. Text captures correlations, frequent patterns, and explicit human descriptions of rules. It does not capture the full sensory bandwidth of embodied experience from which robust world models are built.

The model has learned the 10^{-8} of reality that made it into text. The failures reveal the 99.9999% that did not.

1.12 The case against

Let us give the opposition its strongest possible arguments.

“Evolution is wasteful.” Natural selection is not gradient descent. It does not follow the steepest path to a solution. It wanders, gets stuck in local optima, spends millions of years on body plans that lead nowhere. Surely a more directed optimization process could find intelligence with far less computation.

This is plausible. Directed search is generally more efficient than random search. But how much more efficient? Evolution is not purely random; it is a sophisticated optimization algorithm that combines random mutation with strong selection pressure, sexual recombination, and developmental constraints that bias the search toward viable phenotypes. It is closer to a well-tuned evolutionary strategy than to brute-force enumeration. Claiming a billion-fold speedup over this already-sophisticated process is extraordinary and requires evidence, not assumption.

“Intelligence might have a shortcut.” Perhaps there exists a compact algorithm, a set of principles that, once discovered, allows intelligence to be instantiated with modest computation. Evolution could not find this shortcut because evolution optimizes for survival, not for elegant algorithms.

This is the strongest version of the objection, and we cannot rule it out. It is possible. But “possible” is not "probable," and it is certainly not a basis for confident predictions about when AGI will arrive. The shortcut hypothesis is unfalsifiable in the absence of the shortcut itself. Until someone demonstrates such a shortcut, the only empirical evidence we have is the evolutionary record, and it says the problem is very, very hard.

"Moore's Law and algorithmic improvements will close the gap." Compute costs have fallen exponentially for decades and may continue to do so. Even if 10^{25} learning instances is the target, perhaps we simply need to wait.

The difficulty depends on which estimate we use. The optimistic gap of 10^{18} represents roughly 60 doublings. At two years per doubling, that is 120 years of Moore's Law, and this assumes the trend does not slow further (it already has). The full gap of 10^{33} would require over 200 years. Algorithmic improvements could compress the timeline, but they would need to close whatever gap remains after hardware gains, and no algorithmic improvement in the history of computer science has delivered a 10^{18}-fold speedup on a problem of this generality.

"Evolution's solution is not the only solution." Birds fly but airplanes do not flap their wings. Perhaps artificial intelligence need not recapitulate biological evolution.

This analogy is frequently invoked and frequently misapplied. Airplanes do obey the same physics as birds. Lift requires an airfoil and forward motion through a fluid. The Wright brothers studied birds obsessively. What changed was the mechanism (fixed wings instead of flapping), not the underlying principle (Bernoulli's equation, Newton's third law). If artificial intelligence departs from biological intelligence, it must still solve the same underlying problem: extracting reliable generalizations from experience in a world governed by physics. The question is not whether the mechanism must be identical, but whether the computational cost of solving the problem can be radically reduced. Our estimate, built entirely from optimistic assumptions, suggests the cost is high.

"Current AI systems already show signs of general intelligence." Large language models pass bar exams, write code, reason about novel problems, and exhibit capabilities that were not explicitly trained. Perhaps current training runs are already sufficient for a meaningful degree of general intelligence.

We address this argument fully in a later chapter, but the short response here is: there is a difference between impressive performance on specific benchmarks and the kind of robust, flexible, embodied intelligence that evolution produced. A system that can discuss the concept of heat but has never been burned, that can describe a sunset but has never seen one, has a qualitatively different relationship to knowledge than an organism that has lived through these experiences. Whether this difference matters for practical applications is debatable. Whether it constitutes general intelligence in any rigorous sense is the question this book exists to explore.

The invoice stands. By the most optimistic accounting we can construct, nature spent 10^{25} learning instances to produce general intelligence. The largest artificial training runs have performed roughly 10^{7} gradient updates. The gap is eighteen orders of magnitude. We have tilted every assumption in favor of the optimist, and the distance remains.

1.13 Chapter summary

- Evolution spent approximately 10^{40} learning instances across all three tiers (microbial, invertebrate, vertebrate), or 10^{25} if we count only the vertebrate era where general intelligence emerged
- A learning instance is a cycle of organism-environment interaction from which adaptive information can be extracted, not merely neural activity
- Current frontier AI training runs perform roughly 10^{7} gradient updates, yielding an eighteen-order-of-magnitude gap
- This is a lower-bound estimate using every optimistic assumption; the true gap may be larger
- Human-specific intelligence may require "only" 10^{16} learning instances, but this sits atop evolved neural machinery that cost 10^{25} instances to build
- Current AI failures cluster predictably in areas requiring generalization beyond text: physical reasoning, causal inference, compositional generalization, out-of-distribution robustness

- The gap is not merely quantitative but reveals the difference between learning statistical regularities in text versus learning from embodied interaction with physics

Chapter 2

The architecture chasm

“The brain is a computer made of meat.” — Marvin Minsky

The gap between biological and artificial intelligence is not merely quantitative. It is architectural, rooted in the fundamental organization of memory and computation. This chapter examines why organic systems can do what silicon cannot, and whether this difference can be overcome.

2.1 The plasticity gap

Beyond the quantity of learning instances and the richness of the training signal, there is a third dimension of the gap that we have not yet addressed: the architecture of learning itself. Organic and artificial systems do not merely differ in how much they learn or what they learn from. They differ in *how* they learn, and this difference constitutes its own bottleneck.

2.1.1 The organic learning cycle

Biological learning is not a single event. It is a continuous cycle: wake, experience, sleep, consolidate, repeat. Every day, every organism with a nervous system runs this loop.

McClelland, McNaughton, and O’Reilly formalized this in 1995 as the theory of complementary learning systems. The brain maintains two distinct but interacting memory systems. The hippocampus encodes new experiences rapidly, capturing episodes in something close to real time: a single exposure to a novel environment is sufficient to create a stable hippocampal representation. The neocortex, by contrast, learns slowly, extracting statistical structure from experience over days, weeks, and months. During sleep, hippocampal memories are replayed and gradually integrated into neocortical representations, a process that interleaves new memories with old ones to prevent the new from overwriting the established.

This is not an optional feature. It is the mechanism that allows organisms to accumulate knowledge over a lifetime without losing what they already know. A crow that learns to use a new tool does not forget how to fly. A rat that maps a new environment does not lose its memory of its home territory. The consolidation cycle, running on a roughly 24-hour period, is what makes lifelong learning possible.

The scale of this process is staggering. Billions of organisms, each running the learn-consolidate cycle every day, for 500 million years. The total number of consolidation cycles across the vertebrate era alone:

$$10^{12} \text{ organisms} \times 365 \text{ days/year} \times 5 \times 10^{8} \text{ years} \approx 2 \times 10^{23} \text{ consolidation cycles}$$

Each cycle integrates new experience with existing knowledge without catastrophic loss.

2.1.2 The one-shot learner

Large language models learn in a fundamentally different way. Training is a single monolithic pass through the data. The model sees each example, computes its error, updates its weights, and moves on. When training ends, the model freezes. Inference is recall, not learning. The model after training is a static function.

Fine-tuning exists, but it exposes the architectural limitation rather than resolving it. McCloskey and Cohen demonstrated in 1989 that connectionist networks trained sequentially on new material catastrophically forget previously learned material. This is not a subtle degradation; it is wholesale destruction. A network trained on task A, then fine-tuned on task B, can lose its ability to perform task A entirely. The phenomenon is qualitatively different from biological interference, where old and new memories compete but coexist. In catastrophic forgetting, the old memories are overwritten.

Modern techniques (LoRA, elastic weight consolidation, replay buffers) mitigate the problem but do not solve it. They slow the forgetting; they do not prevent it. No artificial system has demonstrated the ability to learn continuously over thousands of tasks without performance degradation on earlier ones. Biology does this effortlessly, because the consolidation mechanism was designed by evolution precisely for this purpose.

2.2 Memory bandwidth: why biology can do what silicon cannot

The architectural difference between biological and artificial learning is not merely algorithmic. It is physical, rooted in the fundamental organization of memory and computation.

The human brain contains approximately 1.5×10^{14} synapses. Bartol and colleagues demonstrated in 2015, using serial-section electron microscopy of hippocampal tissue, that each synapse stores approximately 4.7 bits of information (26 distinguishable states of synaptic strength). The total storage capacity:

$$1.5 \times 10^{14} \times 4.7 \approx 7 \times 10^{14} \text{ bits} \approx 1 \text{ petabit}$$

Roughly one petabyte of storage, distributed across 10^{14} individually addressable elements. But the critical feature is not the capacity. It is the architecture. Each synapse is simultaneously a storage element and a computational element. There is no separation between memory and processing. When a synapse stores a new weight, it does so at the site where that weight is used in computation. There is no bus, no cache hierarchy, no fetch-store cycle. Backus identified this as the fundamental limitation of conventional computing in 1978: the “von Neumann bottleneck,” where a single channel between processor and memory becomes the limiting factor regardless of how fast either component operates.

In the brain, all $\sim 10^{14}$ synapses can update simultaneously. The effective “write bandwidth” is the entire brain, operating in parallel. There is no serialization, no contention for a shared memory bus.

Compare this to frontier AI hardware. GPT-4 is estimated at roughly 1.8×10^{12} parameters, stored at 16 bits each: approximately 3×10^{13} bits, or about 3.6 terabytes. The brain has roughly 24 times more raw storage. But the storage gap is not the decisive factor.

The decisive factor is bandwidth. The fastest GPU memory available (HBM3) achieves roughly 3.35 terabytes per second. This sounds fast until we consider the physical reality of a single gradient step. For a 200-billion parameter model, the weights alone occupy 400 gigabytes. To perform one update, we must move roughly 1.2 terabytes across the bus: reading the weights for the forward pass, then reading and writing them again for the update. At the theoretical peak of HBM3, this data transit alone consumes 360 milliseconds. For a processor, this is an eternity. In that same window, the human brain has integrated sensory input and updated its internal state multiple times. It achieves this without moving a single bit of data. Its 10^{14} synapses update in place, in parallel. The factory is the warehouse. The brain faces no such constraint.

The energy comparison is equally stark. Horowitz’s analysis of computing energy costs established the hierarchy: a single synapse-like event in the brain costs roughly 10 femtojoules. Reading a bit from HBM3 costs roughly 2.5 picojoules, 250 times more. Reading from off-chip DRAM costs roughly 1.3 nanojoules, 130,000 times more. The brain operates roughly 100,000 times closer to the Landauer thermodynamic limit than conventional silicon memory.

At the system level: the brain runs on 20 watts. A single NVIDIA H100 GPU draws 700 watts. A frontier training cluster of 25,000 GPUs consumes roughly 17 megawatts. The brain achieves comparable information storage and vastly superior write bandwidth at approximately one millionth the system power.

2.3 Could silicon close the gap?

The von Neumann architecture is the fundamental constraint. Separate memory and compute means data must travel, and travel costs energy and time. Three approaches attempt to overcome this.

Neuromorphic chips co-locate memory and computation on the same die. IBM's NorthPole chip, described by Modha and colleagues in 2023, achieves roughly 25 times the energy efficiency of comparable GPUs for inference tasks. Intel's Loihi implements spiking neural networks with on-chip synaptic memory. But these chips face a hard tradeoff: co-locating memory limits total capacity to what fits on a single die. NorthPole is an inference accelerator, not a training platform. As Modha acknowledged, "we cannot run GPT-4 on this." The largest neuromorphic system built to date, Intel's Hala Point (1,152 Loihi 2 chips), contains roughly 10^9 artificial neurons: five orders of magnitude short of the brain's 10^{14} synapses.

Memristors are analog devices that store synaptic weights in their resistance state, co-locating storage and computation at the device level. The best laboratory demonstrations achieve roughly 1.23 femtojoules per synaptic operation, approaching the brain's 10 femtojoules. But commercial memristor arrays remain 1,000 to 100,000 times less efficient than biology, and fabricating 10^{14} of them on a single substrate, the density needed to match the brain's synapse count in a comparable volume, exceeds any current or near-term lithographic capability. The brain packs 10^{14} synapses into roughly 1.2 liters. No silicon process achieves this density.

The theoretical floor for this cost is the Landauer limit. Derived from the second law of thermodynamics, it defines the minimum energy required to erase one bit of information:

$$E = kT \ln 2$$

Where k is the Boltzmann constant (1.38×10^{-23} J/K) and T is the absolute temperature. At a room temperature of 27°C (300 K), this value is approximately 2.87×10^{-21} joules, or 0.003 femtojoules. The brain, at 10 femtojoules per synaptic event, operates within a factor of 3,500 of this thermodynamic floor. Conventional DRAM, at roughly 10^6 femtojoules per access, sits 350 million times above it. Even perfect memristors operating at the Landauer limit would still need to be fabricated at biological density, 10^{14} devices in parallel, to match the brain's effective bandwidth. We are not close to this.

2.4 Why continual learning fails

The catastrophic forgetting problem has not been ignored. Decades of research in continual learning, lifelong learning, and meta-learning have attempted to solve it. The results are instructive: every approach mitigates the problem but none solves it at the scale and generality that biology achieves effortlessly.

Elastic Weight Consolidation (EWC): Kirkpatrick and colleagues at DeepMind proposed in 2017 that important weights for previous tasks should be protected during training on new tasks. The method estimates which weights matter most (using the Fisher information matrix as a proxy) and adds a regularization term that penalizes changing them. This slows forgetting but does not prevent it. On sequences of 10-20 tasks, performance degrades measurably. On sequences of hundreds or thousands of tasks, the approach breaks down entirely.

Progressive Neural Networks: Rusu and colleagues proposed growing the network for each new task, adding new columns while freezing previous ones. This prevents forgetting by construction: old knowledge is literally frozen. But the network grows without bound, and interference still occurs through lateral connections. More fundamentally, this is not continual learning; it is task-specific modularization. Biology does not add a new brain region for every new skill.

Replay buffers: Store examples from previous tasks and interleave them during training on new tasks. This works if the buffer is large enough to represent the full training history, but then you are not learning continuously—you are re-training from scratch on the accumulated buffer. If the buffer is small, you get a biased sample, and forgetting still occurs. Replay is effective in narrow domains (Atari games, robotic control) but does not scale to the open-ended learning that biology performs.

Meta-learning approaches (MAML, Reptile): Train the model to be good at learning new tasks with few examples. This improves sample efficiency on new tasks but does not prevent forgetting of old ones. The model learns a good initialization, not a consolidation mechanism.

Synaptic intelligence, PackNet, CPG: Various approaches that identify important weights and protect them. All reduce forgetting relative to naive fine-tuning. None approach biological performance. On standard continual learning benchmarks (Split MNIST, Permuted MNIST, Split CIFAR), these methods allow the model to learn perhaps 10-50 tasks before performance collapses. Vertebrate organisms learn thousands of skills over a lifetime without forgetting how to walk.

Why do all these approaches fall short? Because they are patches applied to an architecture designed for one-shot learning. The transformer, like all feedforward neural networks, separates learning (training time, weights update) from inference (test time, weights frozen). There is no native consolidation mechanism, no dual-system architecture like hippocampus-neocortex, no sleep cycle that integrates new experience without overwriting old knowledge.

Building such an architecture from scratch is an unsolved problem. Whether it can be solved in silicon, and whether it would be computationally feasible even if solved, remains unknown. What is clear is that current approaches do not work, and the gap between silicon and biology on this dimension is as large as the learning instances gap.

2.5 The consolidation compute

Return to the calculation from Chapter 1. Vertebrate organisms ran 2×10^{23} consolidation cycles over 500 million years. Each cycle integrated new experience with existing knowledge across perhaps 10^{10} to 10^{14} synapses (depending on organism size). This is computational work that happened in addition to the learning instances themselves.

If we estimate conservatively that each consolidation cycle involves processing information across 10^{10} synapses (appropriate for small vertebrates that dominate the population), and each synapse performs roughly 10^3 operations during consolidation (replay, integration, synaptic scaling), then:

$$\text{Consolidation compute} \approx 2 \times 10^{23} \text{ cycles} \times 10^{10} \text{ synapses} \times 10^3 \text{ ops/synapse}$$

$$\approx 2 \times 10^{36} \text{ operations}$$

This is in addition to the learning instances themselves. It is the architectural overhead of continuous learning: the compute spent integrating new knowledge without forgetting old knowledge.

Current AI has no equivalent. A training run performs 10^7 gradient updates, then stops. There is no consolidation, no integration, no sleep. The model after training is static. Fine-tuning is possible, but as we have documented, it causes catastrophic forgetting unless carefully managed with replay or regularization—and even then, it does not scale.

If we wanted to replicate evolution's consolidation compute in silicon, using current architectures and current hardware, how long would it take? GPT-4 training reportedly used roughly 10^{25} FLOP. To reach 10^{36} operations would require 10^{11} training runs of equivalent scale. At current energy consumption (roughly 17 megawatts for a frontier training cluster running for months), this would consume more energy than human civilization produces in a year.

The consolidation compute is not a small overhead. It is, potentially, the dominant cost of biological learning. And current AI does not do it at all.

2.6 The compound problem

The gap is not only quantitative, how many learning instances, or qualitative, what the training signal contains. It is architectural. Organic systems are continuous learners with co-located memory and computation, massive parallel write bandwidth, and a consolidation mechanism that prevents catastrophic forgetting. Silicon systems are one-shot learners with separated memory and compute, serial bandwidth bottlenecks, and no consolidation mechanism.

Even if we could somehow generate 10^{25} learning instances of equivalent richness to biological experience, the current architecture could not process them in the way biology does: continuously, with consolidation, without forgetting. A single monolithic training run is not equivalent to 500 million years of daily learn-and-consolidate cycles, even if the total instance count matches. The *path* through the data matters, not merely the quantity. And the path that biology took, continuous learning with sleep-mediated consolidation, is one that current silicon architectures cannot follow.

2.7 Chapter summary

- Biology achieves continuous learning through complementary systems: hippocampus for fast encoding, neocortex for slow integration, sleep for consolidation
- Current AI suffers catastrophic forgetting: fine-tuning on new tasks destroys performance on old tasks
- Existing continual learning approaches (EWC, progressive networks, replay buffers, meta-learning) mitigate but do not solve the problem at biological scale
- The von Neumann bottleneck: separated memory and compute creates bandwidth and energy costs that biology avoids through co-located synaptic memory
- Biology's 10^{14} synapses update in parallel at 10 femtojoules per operation, operating near the Landauer thermodynamic limit
- Silicon memory operates 10^5 to 10^8 times further from the thermodynamic limit and requires serial data movement
- Neuromorphic and memristor approaches show promise but remain orders of magnitude short of biological density and efficiency
- Vertebrates ran 2×10^{23} consolidation cycles, representing perhaps 10^{36} operations of integration compute beyond the learning instances themselves
- Current training runs perform no consolidation; the architectural gap is as fundamental as the learning instances gap

Chapter 3

The sensory bandwidth gap

“We can know more than we can tell.” — Michael Polanyi

The sheer number of learning instances is only half the story. The other half is what those instances were computed *over*: the training data, to borrow the machine learning framing. This chapter examines the fundamental difference between text and embodied experience.

3.1 Nature’s training data

Every organism that contributed to evolution’s computation was embedded in a physical environment. It did not read about sunlight; it photosynthesized or basked in it. It did not process text descriptions of predators; it heard them, smelled them, ran from them, and sometimes was eaten by them. The “training signal” was not a loss function on token prediction. It was survival and reproduction, evaluated against the full sensory bandwidth of embodied existence.

We can now quantify how much bandwidth that actually is. Zheng and Meister established in their 2024 analysis in *Neuron* that the human sensory periphery transmits approximately 10^9 bits per second: roughly one gigabit, dominated by the optic nerve but with substantial contributions from auditory, somatosensory, proprioceptive, and vestibular channels. Of this torrent, conscious experience processes roughly 10 bits per second. The compression ratio from raw sensation to conscious awareness is on the order of 10^8 to one.

Over a human lifetime of roughly 80 years, with about 16 waking hours per day, the total raw sensory input amounts to:

$$80 \times 365 \times 16 \times 3600 \times 10^9 \approx 1.5 \times 10^{18} \text{ bits}$$

Now consider text. The entire written output of human civilization, from Sumerian cuneiform to the modern internet, has been estimated at roughly 3×10^{18} bits (including all digitized books, all web pages, all archived documents). This is a generous upper bound; the high-quality subset that language models actually train on is far smaller. The comparison is devastating: all the text humanity has ever produced contains roughly the same quantity of raw information as a single human lifetime of sensory experience. The entire written record of civilization, 5,000 years of accumulated thought, fits inside one pair of eyes.

3.2 Information quantity vs. information content

But information quantity is not the same as information content. Text is not merely a smaller quantity of the same substance. It is a fundamentally different kind of signal: a lossy compression of experience into symbols, with the vast majority of the original information discarded. When we read “the coffee was hot,” we bring to that sentence a lifetime of thermal experience that the sentence itself does not contain. A language model processes the tokens. A human recalls the burn. The word “hot” in a corpus is a pointer to an experience that the corpus cannot store.

This is not merely a philosophical observation. Polanyi formalized the problem in 1966: “we can know more than we can tell.” The domain of tacit knowledge, skill, intuition, perceptual judgment, embodied understanding, is not a small residual left over after we articulate what we know. It is the majority of what we know. The knowledge management literature consistently estimates that 70 to 80 percent of organizational knowledge is tacit: non-verbalizable, non-transferable through text. Autor brought this into economics in 2014 as “Polanyi’s paradox,” demonstrating that the tasks most resistant to automation are precisely those that rely on tacit knowledge, because we cannot write down rules for what we cannot articulate.

Language captures the 10 bits per second that survive the compression into conscious, articulable thought. It does not capture the 10^9 bits per second of raw sensation from which that thought was distilled. A corpus trained on text is trained on the 10^{-8} fraction of experience that made it through the bottleneck of articulation.

3.3 The grounding problem

This distinction matters for a precise reason. The learning instances we counted were not performed over tokens. They were performed over the full sensory bandwidth of embodied organisms interacting with physics, or in the microbial case, over the direct chemical and thermal realities of survival. If we want to claim that a system trained on text can match the output of this process, we need a theory of how lossy compression of experience into language preserves the adaptive information that the original experience carried. No such theory exists.

And there is a further consequence: even on a step-for-step basis, the comparison flatters AI. Each biological learning instance involves a whole organism perceiving and acting in a physical environment across its full sensory bandwidth. Each gradient step in a language model processes a batch of text tokens. The informational richness per step is not comparable. If anything, counting one gradient update as equivalent to one learning instance is generous to silicon.

3.4 The multimodal response

The obvious objection: multimodal models that process images, video, and audio are closing this gap. They are no longer text-only; they observe the world through vision and sound.

But observation is not interaction. A model that watches a video of fire has not been burned. A model that processes images of food has never been hungry. The difference between passive observation and embodied experience is not merely one of bandwidth; it is one of stakes. Organisms learn because failure has consequences: starvation, predation, reproductive failure, death. The training signal is not mean squared error on pixel prediction. It is survival.

Text bandwidth vs. sensory bandwidth: Human language communicates at roughly 40 bits per second (controlled articulation rate). Human sensory input runs at roughly 10^9 bits per second. Text is a compression ratio of approximately 2.5×10^7:1. Even with multimodal data added, video at typical compression delivers perhaps 10^6 bits per second: still three orders of magnitude below raw sensory experience, and with no physical consequences tied to the learning signal.

3.5 What is lost in compression?

The 99.9999% of experience that does not make it into text (or even into video) is not random noise. It is the substrate from which understanding emerges. The weight of an object, the texture of a surface, the proprioceptive feedback from muscle tension, the thermal sensation of temperature, the vestibular sense of balance, the olfactory landscape of a physical space: these are not decorative details. They are the grounding for concepts that language can only name, not convey.

When we say a language model "understands" physics because it can solve physics problems stated in text, we are using "understand" in a sense that would be unrecognizable to a physicist who has spent years in a laboratory, manipulating physical systems, observing outcomes, developing intuition through embodied interaction. The model has learned the *symbol manipulation rules* of physics. Whether it has learned physics is the question this book exists to explore.

3.6 Tacit knowledge that text cannot capture

The most revealing examples of the sensory bandwidth gap are skills that even young children possess but that no amount of text can convey.

Riding a bicycle. Ask any cyclist to explain how they balance. The answer will be vague: "You just lean into it," "You feel when you're tipping," "Your body knows what to do." This is not evasiveness. It is the honest acknowledgment that the knowledge is tacit. The cerebellum and motor cortex maintain a control loop involving vestibular input (balance), proprioceptive feedback (body position), visual flow (velocity), and predictive models of dynamics. This loop operates at millisecond timescales below conscious awareness. Reading a thousand pages about bicycle physics does not create this control loop. The knowledge is encoded in synaptic weights shaped by thousands of trials, falls, and recoveries.

Catching a ball. The solution to this problem requires solving differential equations in real time: given the ball's trajectory (which must be estimated from incomplete visual data), predict the interception point and move there. Humans do this effortlessly by age five. The computation happens in visual cortex, parietal cortex, and motor cortex without conscious access. When asked to explain how they catch, subjects say: "I just watch it and move to where it's going." This is not an explanation; it is a description of phenomenology. The actual computation, involving optical flow analysis, predictive extrapolation, and motor planning, is entirely tacit.

Judging if ice will hold your weight. This requires integrating visual cues (color, transparency, surface texture), auditory cues (cracking sounds), proprioceptive cues (how the ice flexes underfoot), and contextual knowledge (temperature, wind, time of year). An experienced person makes this judgment in seconds with high confidence. Ask them to articulate their decision process, and they struggle: "It looks solid," "The color seems right," "I've walked on ice like this before." These descriptions capture fragments of the input but not the integration process. The judgment is a weighted combination of dozens of features, most of which the person cannot consciously access or verbalize.

Tying shoelaces. This motor skill is learned through repetition until it becomes automatic. Ask someone to describe how they tie their shoes, and they will struggle unless they slow down and consciously monitor their hands. The procedural knowledge is stored in motor cortex and cerebellum, encoded as sequences of muscle activations, not as verbal instructions. You cannot learn to tie shoes by reading instructions alone; you must practice until the motor memory forms.

Estimating object weight from vision. Before lifting an object, humans visually estimate its weight based on size, material, and context. Pick up a box that looks heavy but is empty, and you will apply too much force—your motor system prepared for the visually estimated weight. This mapping from visual features to expected weight is learned through thousands of lifting experiences and is entirely non-verbal. No amount of text describing "metal is denser than wood" creates this perceptual-motor calibration.

Navigating a crowd. Walking through a dense crowd without colliding requires real-time prediction of other people's trajectories, planning a path, adjusting based on peripheral vision, and coordinating muscle activation to execute the plan. This is continuous sensory-motor integration at millisecond timescales. People do it effortlessly while conversing, thinking about other things, their attention elsewhere. The computation is entirely tacit, operating below the threshold of conscious articulation.

These are not exotic skills. They are everyday embodied intelligence that nearly all humans possess by adulthood. Text rarely describes them in detail because they are difficult to articulate. When text does describe them (instructional manuals, coaching guides), the descriptions are crude approximations. A manual on bicycle riding cannot replace the lived experience of falling and recovering until the sensory-motor loop is calibrated.

A language model trained on text sees the words “riding a bicycle” millions of times. It learns that bicycles have two wheels, that balance is required, that people learn as children. It can answer questions about bicycles and even generate plausible instructions. But it does not have the sensory-motor knowledge that a five-year-old possesses after a weekend of practice. The 10^9 bits per second of embodied experience—the falls, the vestibular feedback, the proprioceptive calibration—are not in the training data.

This is the tacit knowledge gap. It is not a small residual left over after articulation. It is the majority of human intelligence: perceptual, procedural, embodied, grounded in physics, and inaccessible to any system trained only on text.

3.7 Chapter summary

- Human sensory bandwidth is approximately 10^9 bits per second; conscious articulation captures roughly 10 bits per second, a compression ratio of 10^8 to one
- All human-generated text (books, web, papers) contains roughly 3×10^{18} bits, equivalent to one human lifetime of sensory experience
- Text is lossy compression of experience into symbols; the word “hot” points to thermal experience the text does not contain
- Polanyi’s paradox: 70-80% of knowledge is tacit, non-verbalizable, non-transferable through text
- Biological learning instances operated over full sensory bandwidth of embodied organisms interacting with physics; gradient updates operate over token sequences
- Multimodal models observe but do not interact; observation is not equivalent to embodied experience with stakes (survival, reproduction)
- Tacit knowledge examples: riding a bicycle, catching a ball, judging if ice is safe, tying shoelaces, estimating object weight from vision, navigating crowds
- These are skills even young children possess but that no text can convey; they require lived sensory-motor calibration
- The training signal for evolution was survival evaluated against physics; the training signal for LLMs is cross-entropy loss on token prediction
- This is not a quantitative gap that more scale can close; it is a qualitative difference in the substrate of learning

Part II

The scaling mirage

Chapter 4

Diminishing returns

> "It is difficult to get a man to understand something when his salary depends upon his not understanding it." — Upton Sinclair

Chapter 1 established the target: 10^{25} learning instances at the optimistic vertebrate-only estimate, against roughly 10^7 gradient updates in frontier AI training. The gap is eighteen orders of magnitude. The question this chapter asks is whether scaling, the strategy of simply making models bigger and training them longer, can close it.

4.1 The power law promise

The case for scaling rests on a genuine empirical discovery. Kaplan and colleagues at OpenAI demonstrated in 2020 that language model performance, measured as cross-entropy loss on held-out text, follows a power law in three variables: the amount of compute C, the number of model parameters N, and the size of the training dataset D. Specifically:

$$L(C) = a_C \cdot C^{-\alpha_C}, \quad \alpha_C \approx 0.050$$

$$L(N) = a_N \cdot N^{-\alpha_N}, \quad \alpha_N \approx 0.076$$

$$L(D) = a_D \cdot D^{-\alpha_D}, \quad \alpha_D \approx 0.095$$

These are not theoretical predictions. They are fits to experimental data spanning five orders of magnitude of compute, from small models trained on modest datasets to the largest systems available at the time. The fits are remarkably clean: the power law holds with minimal deviation across the entire range.

Hoffmann and colleagues refined this picture in 2022 with the Chinchilla study, which demonstrated that Kaplan's original scaling prescription was suboptimal. Kaplan had suggested scaling parameters faster than data; Hoffmann showed that compute-optimal training requires scaling both in roughly equal proportion, at approximately 20 tokens per parameter. A model with 70 billion parameters, trained on 1.4 trillion tokens (the Chinchilla recipe), outperformed a 280-billion-parameter model trained on 300 billion tokens (the Gopher recipe), despite using the same compute budget. The lesson was clear: the field had been training models that were too large on too little data.

Power laws in complex systems reflect deep structural properties of the underlying optimization landscape. The scaling community's central claim, that performance improves predictably with scale, has been validated repeatedly across model families, training methodologies, and evaluation benchmarks.

The question is not whether the scaling laws hold. The question is what they actually promise.

4.2 Diminishing returns as mathematical certainty

A power law with exponent $\alpha < 1$ is, by definition, a function of diminishing returns. It is a mathematical consequence of the functional form itself. Let us derive the implications explicitly.

The scaling law for compute is:

$$L(C) = a \cdot C^{-\alpha}, \quad \alpha \approx 0.05$$

Suppose we are currently at compute level C_0 with loss $L_0 = a \cdot C_0^{-\alpha}$. To reduce the loss by a factor of 2 (halve it), we need compute C_1 such that:

$$a \cdot C_1^{-\alpha} = \frac{L_0}{2} = \frac{a \cdot C_0^{-\alpha}}{2}$$

Solving:

$$C_1^{-\alpha} = \frac{C_0^{-\alpha}}{2}$$

$$C_1 = C_0 \cdot 2^{1/\alpha}$$

For $\alpha = 0.05$:

$$2^{1/0.05} = 2^{20} \approx 10^6$$

To halve the loss, we need one million times more compute. To halve it again from that new level, we need another factor of 10^6 on top: 10^{12} times the original budget. Each successive halving costs a million-fold increase.

We can express this more generally. The compute required to reduce loss by a factor of k from any starting point is:

$$\frac{C_{\text{new}}}{C_{\text{old}}} = k^{1/\alpha}$$

For even modest improvements, the cost becomes extreme:

Loss reduction factor	Compute multiplier ($\alpha = 0.05$)
2x better	$\sim 10^6$
3x better	$\sim 10^{9.5}$
5x better	$\sim 10^{14}$
10x better	$\sim 10^{20}$

A tenfold improvement in loss requires 10^{20} times more compute than the current level. For context, the entire global compute capacity deployed for AI training in 2024 was estimated at roughly 10^{26} FLOP. A tenfold loss improvement from that baseline would require 10^{46} FLOP, exceeding the estimated computational capacity of a Kardashev Type I civilization.

Now connect this to the gap from Chapter 1. We argued that the comparison between biological learning and artificial training should be measured in gradient updates versus learning instances, yielding a gap of 10^{18}. But even if we accept the FLOP-to-FLOP comparison favored by scaling optimists, the scaling law itself tells us that progress decelerates at a rate that makes closing the gap extraordinarily expensive. The power law does not promise convergence. It promises asymptotically slowing approach toward a floor.

The Chinchilla revision, which roughly doubled the effective exponent by fixing the data-parameter ratio, was a genuine improvement. But doubling α from 0.05 to 0.1 changes the compute multiplier for a tenfold loss reduction from 10^{20} to 10^{10}. Better, but still astronomical. And Chinchilla was a one-time correction of a systematic error in training methodology. There is no reason to expect repeated corrections of comparable magnitude.

4.3 The bitter lesson and its limits

In 2019, Rich Sutton published a short essay titled "The Bitter Lesson" that became one of the most cited pieces of informal writing in AI research. His argument was simple and historically well-supported: across the history of artificial intelligence, general methods that leverage computation have consistently won over methods that leverage human knowledge. Hand-crafted features lose to learned features. Expert systems lose to neural networks. Carefully engineered game-playing programs lose to brute-force search combined with learning. The lesson is "bitter" because it means that human cleverness about the structure of problems is less valuable than raw compute applied to general-purpose learning algorithms.

Sutton was right about the history. The trend he identified is real and has continued to hold. But the bitter lesson carries an implicit assumption that we should make explicit: it assumes compute is the binding constraint. Given sufficient data of sufficient quality, more compute yields better performance. The lesson's historical examples confirm this, because in every case Sutton cited, more data of the relevant kind was available.

The question is whether the assumption holds for the frontier we are now approaching. The scaling laws were measured over text: token prediction on natural language corpora. They describe how loss on text decreases as a function of compute, parameters, and data, all applied to text. There is no empirical evidence that these same scaling laws extend to the kind of learning that evolution performed: embodied, continuous, multi-sensory, physically grounded experience evaluated against survival and reproduction.

This is not a pedantic distinction. Chapter 1 established that the evolutionary training signal was qualitatively different from text in at least three ways: its information density (roughly 10^9 bits per second of sensory input versus the bandwidth of written language), its grounding in physical causation (organisms interacted with a world that obeys consistent physical laws, not with statistical regularities in token sequences), and its evaluation criterion (survival and reproduction, not cross-entropy loss on held-out text). The scaling laws tell us how fast text prediction improves with scale. They tell us nothing about whether text prediction, at any scale, converges to the capabilities that embodied evolutionary learning produced.

Chapter 2 also established a fourth difference: the architecture of learning itself. Nature's learning was continuous, with daily consolidation cycles that interleaved new experience with existing knowledge. It ran on co-located memory and computation with massive parallel write bandwidth. The scaling laws were measured on systems that learn in a single pass, with separated memory and compute, and no consolidation mechanism. Even if more compute and more data were available without limit, the one-shot training paradigm cannot replicate the learn-consolidate cycle that enabled biological knowledge accumulation.

The bitter lesson says: do not bet against scale. Sound advice, as far as it goes. But it does not say: scale solves all problems. The lesson is about the relative merit of general methods versus hand-crafted ones within a domain where more data is available. It is silent on what happens when the data runs out, when the training signal lacks the information content of the target domain, or when the learning architecture cannot support the required mode of knowledge accumulation.

4.4 Empirical evidence of deceleration

The power law predicts deceleration. What does the empirical trajectory show?

GPT-2 to GPT-3 (2019 to 2020): Model size increased from 1.5B parameters to 175B parameters, roughly 100x. Training compute increased proportionally. The improvement was dramatic: GPT-3 demonstrated few-shot learning, could follow complex instructions, and showed surprisingly broad knowledge. This was a genuine capability jump, not merely incremental improvement.

GPT-3 to GPT-4 (2020 to 2023): Training compute increased by roughly another 10-100x (estimates vary; OpenAI has not released precise figures). Model architecture became more sophisticated, incorporating multimodality and likely mixture-of-experts. The improvement was real: GPT-4 passes professional exams (bar exam, AP exams), writes more coherent long-form text, handles more complex reasoning chains, integrates vision and text. But the improvement was smaller in qualitative terms than GPT-2 to GPT-3. GPT-4 is not a different kind of system; it is a better version of the same kind of system.

Benchmarks confirm this. On MMLU (Massive Multitask Language Understanding), a broad knowledge benchmark: - GPT-3: ~43% - GPT-3.5: ~70% - GPT-4: ~86%

The jump from GPT-3 to GPT-3.5 was 27 percentage points. The jump from GPT-3.5 to GPT-4 was 16 percentage points. The rate of improvement is slowing even as compute expenditure increases exponentially.

On HumanEval, a code generation benchmark: - GPT-3: ~0% - GPT-3.5 (code-davinci-002): ~47% - GPT-4: ~67%

Again, the largest jump was early. GPT-4 is better, but not 100x better, despite 100x more compute.

Claude 2 to Claude 3 to Claude 3.5 (2023 to 2024): Anthropic's models show a similar pattern. Claude 3 Opus outperformed Claude 2 significantly on reasoning benchmarks. Claude 3.5 Sonnet (mid-2024) showed further improvement but on a smaller scale. The deceleration is visible.

Gemini models (2023-2024): Google's Gemini Ultra achieved performance comparable to GPT-4, using massive compute. Gemini 1.5 introduced a 1-million-token context window, a genuine architectural innovation, but capability improvements on standard benchmarks were incremental.

The pattern is consistent across labs and model families: each generation requires more compute, and each generation delivers smaller improvements. This is not surprising. It is what the power law predicts. But it is direct evidence that we are moving up the curve into the regime of diminishing returns.

4.5 Test-time compute: does it change the picture?

OpenAI's o1 model, released in late 2024, introduced a new approach: test-time compute. Rather than simply generating the most likely next token, the model performs internal "reasoning" steps, searching over possible solution paths before committing to an answer. On some benchmarks, particularly mathematical and coding problems, o1 dramatically outperforms GPT-4.

Does this change the scaling picture?

The answer depends on what we mean by "scaling." Test-time compute does not extend the training data, does not increase the number of learning instances, and does not solve the architectural or sensory bandwidth gaps. What it does is allow the model to spend more inference compute searching over possibilities within the distribution it has already learned.

This is valuable. For problems that admit search (math, coding, formal reasoning), allocating more compute at inference time can find better solutions. This is conceptually similar to chess engines, which improve with more search depth even if the evaluation function remains constant.

But test-time compute has limits: 1. It only helps for problems where the solution can be verified (math, code, logic). For open-ended generation, summarization, or creative tasks, there is no verifier to guide the search. 2. It operates within the learned distribution. If the model has not learned the relevant concepts during training, search cannot discover them. o1 does not suddenly develop physical intuition or embodied common sense; it searches more carefully over the text-based knowledge it already has. 3. It is expensive. Inference cost scales with search depth. If o1 uses 100x more compute per query than GPT-4, then deploying it at scale costs 100x more. This is acceptable for high-value tasks (scientific research, complex coding) but not for general-purpose use.

Test-time compute is a genuine innovation and will be valuable in specific domains. But it does not solve the fundamental gaps. It is better search over a limited map, not a larger map. The deceleration of capability improvements with training compute remains, and test-time compute does not bypass the data wall, the sensory bandwidth gap, or the architectural constraints.

4.6 The case against

We owe the strongest counterarguments a fair hearing.

"Scaling laws have held for five orders of magnitude. Betting against them is foolish." This is true, and it is the strongest version of the scaling argument. Five orders of magnitude is a large extrapolation base, and the fits have been clean. But a power law with $\alpha < 1$ is self-limiting by definition. The fact that it holds does not mean it is sufficient. A function can hold perfectly and still guarantee that the destination is unreachable within any feasible budget. We are not betting against the scaling laws. We are reading them carefully, and what they say is that each unit of progress costs exponentially more than the last.

"Algorithmic improvements change the exponent." Possible, and some improvements have been genuine. The Chinchilla correction roughly doubled the effective exponent for a fixed compute budget. Mixture-of-experts architectures, better tokenization, and curriculum learning have all contributed incremental gains. But no demonstrated algorithmic improvement has delivered more than a roughly 2x efficiency gain in compute-equivalent terms. The gap is 10^{18}. To close it through algorithmic improvement alone would require discovering, in sequence, roughly 60 independent doublings of efficiency, each one a Chinchilla-scale breakthrough. The history of computer science offers no precedent for sustained improvement at this rate on a single problem class.

4.7 Chapter summary

- Scaling laws demonstrate that language model loss follows a power law $L(C) = a \cdot C^{-\alpha}$ with $\alpha \approx 0.05$, a function of diminishing returns by definition
- To halve the loss requires 10^6 times more compute; to reduce by 10x requires 10^{20} times more compute (exceeding projected global AI compute capacity)
- The Chinchilla correction (2022) roughly doubled the effective exponent by fixing the data-parameter ratio, but this was a one-time correction of a systematic error
- Empirical evidence confirms deceleration: GPT-3 to GPT-4 required 10-100x more compute for smaller qualitative improvements than GPT-2 to GPT-3
- Benchmark trajectories show slowing gains: each generation delivers fewer percentage points of improvement despite exponentially more compute
- The bitter lesson (Sutton, 2019) says general methods that leverage compute beat hand-crafted approaches, but this assumes unlimited data and does not address architectural constraints
- Scaling laws were measured over text prediction; they tell us nothing about whether text prediction converges to embodied, physically grounded intelligence
- Test-time compute (OpenAI's o1) improves performance on verifiable problems through search but does not extend training data or solve fundamental gaps
- Test-time compute operates within the learned distribution; it cannot discover concepts not present in training data
- The 10^{18} gap cannot be closed by algorithmic improvements alone; this would require 60 sequential Chinchilla-scale breakthroughs with no historical precedent

Chapter 5

The data wall

"We have achieved peak data." — Ilya Sutskever

The scaling laws assume unlimited training data. The Chinchilla prescription demands roughly 20 tokens per parameter. For a model with 10 trillion parameters, the recipe calls for 200 trillion tokens. Where do these tokens come from?

5.1 The finite supply

Villalobos and colleagues at Epoch AI published the most careful analysis of this question in 2024, presented at ICML. Their estimate: the total stock of publicly available, high-quality text on the internet amounts to roughly 300 trillion tokens. This is not a conservative guess; it includes web pages, digitized books, scientific papers, code repositories, social media, forums, and news archives. The "high-quality" qualifier matters enormously: the FineWeb dataset, one of the most careful web-scraping efforts, discards roughly 85% of raw web text during quality filtering. The actual supply of text that meets the quality threshold for training frontier models is a fraction of the raw total.

To appreciate the scale of the constraint, consider the components:

All books ever printed: approximately 170 million distinct titles. At a rough average of 70,000 words per book, this yields roughly 12 trillion words, or about 16 trillion tokens. All scientific papers ever published: roughly 100 million papers at an average of 5,000 words each, yielding 500 billion words or approximately 650 billion tokens. The entire scientific output of humanity across all disciplines, from the first journal in 1665 to today, would not fill a single training run for a frontier model.

Current frontier models train on roughly 13 to 18 trillion tokens. The data supply is growing, but not fast: new high-quality text is generated at perhaps 2 to 3 trillion tokens per year, while model appetite grows at roughly 2.5x per year. The curves cross. Villalobos and colleagues estimate that the data wall, the point where demand for training data exceeds the supply of high-quality human-generated text, arrives by approximately 2028 at the median estimate.

The implications for the scaling laws are direct. The power law $L(D) = a_D \cdot D^{-\alpha_D}$ holds only when more data is available to train on. When the supply is exhausted, the curve hits a ceiling. No amount of additional compute or parameters can compensate, because the Chinchilla result demonstrated that undertrained models (too many parameters for the available data) perform worse, not better, than properly scaled ones. At the data wall, making models bigger actively degrades performance.

5.2 The multimodal extension

"Multimodal data extends the wall." Adding images, video, and audio to the training data increases the total supply beyond text alone. Epoch AI estimates this provides roughly a 3x multiplier in effective tokens. Video is particularly data-rich: a single hour of video at modest resolution contains more raw information than a large book. But three considerations limit the impact.

First, 3x against a data wall measured in hundreds of trillions of tokens delays the wall by one to two years, not one to two decades.

Second, multimodal data is still passively observed: the model watches video; it does not interact with the physical world that the video depicts. The gap between observed and embodied experience, quantified in Chapter 3, is not closed by adding more observation.

Third, the scaling laws for multimodal models have not been established with the same rigor as for text. It is an assumption, not an empirical finding, that vision-language scaling follows the same power law.

5.3 When demand meets supply

5.4 Code: the special case

Code repositories represent a massive corpus of structured, high-quality text. GitHub alone hosts over 300 million repositories, containing trillions of tokens of code across hundreds of programming languages. Unlike natural language, code has formal semantics: it must compile, it must run, and its behavior is (in principle) verifiable. This makes code an attractive training target.

Models trained on code (GitHub Copilot, Code Llama, GPT-4 with code capabilities) show impressive performance on standard programming tasks. They autocomplete functions, translate between languages, fix common bugs, and generate boilerplate with high accuracy. Code generation has become one of the most economically valuable applications of language models.

But code faces the same data wall as text, with additional constraints:

Finite supply. While GitHub grows daily, the growth rate is linear or sublinear, not exponential. New code is generated at perhaps 100 billion to 1 trillion tokens per year (estimating from public GitHub commits). Model appetite grows at 2.5x per year. The curves cross. By 2026-2027, model training will exhaust the supply of high-quality public code.

Quality degradation. Not all code is equally valuable for training. Code repositories contain bugs, deprecated patterns, security vulnerabilities, copy-pasted boilerplate, and abandoned projects with poor practices. The signal-to-noise ratio is lower than for curated text like books or scientific papers. Aggressive quality filtering discards perhaps 50-70% of raw code, reducing the effective supply.

Copyright and licensing. Much valuable code is proprietary or restrictively licensed. Training on copyrighted code without permission has triggered lawsuits (GitHub Copilot, Stable Diffusion, and others). Even if legal barriers are overcome, proprietary code represents information that models cannot access. The public code commons is smaller than the total code produced.

AI-generated code pollution. As code generation tools become widely adopted, repositories increasingly contain AI-generated code. This creates the same ouroboros problem as text: models trained on AI-generated code ingest the biases and limitations of previous models. Stack Overflow already reports a significant fraction of answers are AI-generated. Within a few years, distinguishing human-written code from AI-generated code in public repositories may become difficult.

Diminishing returns from code. While code teaches models formal reasoning, syntax, and algorithm implementation, it does not address the broader gaps documented in earlier chapters. Code does not provide embodied grounding, causal understanding, or the sensory bandwidth of physical experience. A model trained on all the code in the world will be an excellent code generator but no closer to general intelligence.

Code extends the data supply by perhaps 1-2x in effective tokens compared to text alone. This delays the data wall by a year or two. It does not eliminate it.

5.5 Data quality: can better curation extend the wall?

If raw data supply is constrained, perhaps higher-quality data can compensate. The hypothesis: one token of textbook-quality, carefully curated data is worth ten tokens of random web scraping. If true, better curation could effectively extend the data supply.

There is evidence for this. Phi-2, a small model (2.7B parameters) trained on carefully curated “textbook-quality” data, outperformed much larger models on reasoning benchmarks. The Chinchilla paper itself emphasized data quality, not merely quantity. Training on high-quality data allows models to reach target performance with fewer tokens.

But quality curation does not create new information. It filters existing information, selecting the highest-value subset. This is valuable for efficiency, but it does not extend the frontier of what can be learned. If the total supply of high-quality data is 300 trillion tokens, aggressive curation might extract 50-100 trillion tokens of genuinely excellent data. This is a 3-6x reduction in effective supply, which trains better models faster. But it does not make the data wall disappear; it makes the wall arrive sooner.

Consider the extreme: suppose we could perfectly curate the highest-quality data and achieve a 10x efficiency improvement (one curated token equals ten random tokens). The data wall moves from 2028 to perhaps 2030. Then what? The perfect curation has been applied. There is no further efficiency to extract. The wall still stands.

Quality curation is an efficiency optimization, not a solution to scarcity. It helps, but it does not change the fundamental constraint: high-quality human-generated data is finite.

5.6 When exactly do we run out?

The timeline depends on model appetite (parameters, training tokens) and data supply growth. We can project based on current trends.

Current state (2024): - Frontier models: 1-2 trillion parameters, trained on 13-18 trillion tokens - Available high-quality data: ~300 trillion tokens (text + code) - Annual new data generation: ~2-3 trillion tokens

Projected scaling (optimistic): - Models double in parameter count every 12-18 months - Training data scales proportionally (Chinchilla ratio: 20 tokens per parameter) - Data generation grows linearly at 2-3 trillion tokens/year

GPT-5 equivalent (2025): ~5 trillion parameters, trained on ~100 trillion tokens. Data supply is sufficient but reserves are depleting.

GPT-6 equivalent (2026-2027): ~10-20 trillion parameters, requiring ~200-400 trillion tokens. Data supply is exhausted. Training at this scale requires reusing the same data multiple times (overtraining) or incorporating lower-quality data, both of which degrade performance.

Beyond 2027: Models cannot scale further without synthetic data or radically different data sources. As Chapter 6 documents, synthetic data leads to model collapse unless carefully mixed with fresh human data, which is not available at the required scale.

Villalobos and colleagues at Epoch AI estimated the median data wall arrival at 2028, with uncertainty spanning 2026-2030. Their analysis is the most careful published estimate, and it aligns with the projection above. Industry insiders (Ilya Sutskever’s “peak data” comment, internal discussions at labs) suggest awareness of this constraint.

The wall is not a distant theoretical concern. It is a near-term practical constraint that labs are already encountering.

5.7 The industry's response

What are companies doing about the data wall?

Licensing deals: OpenAI, Google, and others are signing deals with publishers (News Corp, Associated Press, Stack Overflow, Reddit) to license previously inaccessible data. These deals provide perhaps 1-5 trillion additional tokens, delaying the wall by months, not years.

Scraping expansion: Labs are scraping less-common languages, historical archives, multimedia transcripts, and other previously untapped sources. This provides marginal gains but lower-quality data.

Synthetic data augmentation: Explored extensively in Chapter 6. The consensus: synthetic data can regularize and augment but cannot extend the frontier without causing collapse.

Multimodal expansion: Training on images, video, and audio increases data supply by 2-3x in effective information content. This delays the wall but does not eliminate it, and multimodal data faces its own quality and copyright constraints.

Test-time compute: As discussed in Chapter 4, inference-time search can improve performance without new training data, but only for verifiable tasks within the learned distribution.

None of these approaches solves the fundamental problem. They are optimizations that delay the inevitable. The data wall is not a problem that can be engineered away. It is a conservation law: you cannot learn more information than the data contains, and the data is finite.

5.8 When demand meets supply

The data wall is not a software problem awaiting an engineering solution. It is a hard constraint: the finite supply of high-quality human-generated text meets exponentially growing demand. The industry's response has been to search for alternatives: synthetic data, data augmentation, curriculum learning, data quality filtering. The next chapter examines whether these approaches can extend the frontier or merely delay the inevitable.

5.9 Chapter summary

- High-quality human-generated text totals approximately 300 trillion tokens (books, web, papers, code)
- Current frontier models train on 13-18 trillion tokens; demand grows at ~2.5x per year while supply grows linearly at 2-3 trillion tokens/year
- The curves cross around 2026-2028, marking the data wall where demand exceeds supply
- Code repositories (300M+ on GitHub) provide trillions of additional tokens but face finite supply, quality issues, copyright constraints, and AI pollution
- Quality curation (textbook-quality data) improves efficiency but does not create new information; it filters existing supply, making the wall arrive sooner
- Timeline projection: GPT-5 (2025) barely fits; GPT-6 (2026-2027) exhausts supply; beyond 2027 requires overtraining or synthetic data
- Industry responses (licensing deals, scraping expansion, multimodal data, test-time compute) delay the wall by months to 1-2 years, not decades
- The Chinchilla result demonstrates that undertrained models (too many parameters for available data) perform worse, making the data wall a hard constraint
- This is not a software problem but a conservation law: you cannot extract more information from a corpus than it contains, and the corpus is finite

Chapter 6

The ouroboros problem

“The serpent that eats its own tail.” — Ancient symbol

The obvious response to the data wall is to generate more data. If human-produced text is finite, perhaps model-produced text can fill the gap. This idea has a name in the industry: synthetic data augmentation. It also has a problem.

6.1 Model collapse

Shumailov and colleagues published the definitive analysis in *Nature* in 2024. They showed that language models trained on data generated by other language models (or recursively by themselves) undergo *model collapse*: a progressive degradation of the output distribution that unfolds in two phases.

In the early phase of collapse, the distribution tails disappear. Rare events, minority patterns, unusual phrasings, low-frequency but genuine features of the original distribution, are the first casualties. The model converges toward the mode of the distribution, losing the diversity that characterized the original data. In practical terms: the outputs become more generic, more repetitive, more “average.”

In the late phase, the model loses most of its variance entirely, converging toward something approaching a delta function: a distribution concentrated on a single output pattern. By this stage, generated text is incoherent and repetitive.

The quantitative trajectory is striking. Shumailov and colleagues measured perplexity degradation across generations of recursive training. By generation 4 to 6, perplexity has degraded by 60 to 80 percent. The outputs are recognizably collapsed. The process is not gradual in the way that might allow careful monitoring and correction; it accelerates as each generation’s training data is further from the original distribution.

6.2 Why collapse happens

The mechanism is iterated lossy compression. A generative model is an imperfect estimator of its training distribution. It assigns slightly too much probability to common patterns and slightly too little to rare ones. When the next model trains on this slightly biased output, the bias compounds. Common patterns become more common; rare patterns become rarer. Across multiple generations, the rare patterns vanish entirely. This is not a bug in any particular model architecture. It is a mathematical consequence of iterating any imperfect estimator: each application of the map pushes the distribution toward lower entropy.

The ouroboros, the serpent eating its own tail, is an apt metaphor. A system that feeds on its own output converges to a distribution with lower entropy than its input. The information lost at each step is not recoverable from the output alone.

6.3 The mixing solution

Gerstgrasser and colleagues showed in 2024 that collapse can be avoided if the original human-generated data is preserved and mixed with synthetic data at each generation. This is a genuine finding and should be credited honestly. But it addresses a different problem than the one scaling needs to solve. Preserving original data alongside synthetic data prevents regression below the original model's performance. It does not extend the frontier. The model trained on a mixture of real and synthetic data does not outperform the model trained on real data alone, because the synthetic data contains no information that was not already in the original model. Synthetic data can regularize, can provide augmented views of existing patterns, can improve robustness. It cannot create new knowledge.

6.4 The information conservation law

The data wall is not a software problem awaiting an engineering solution. It is a conservation law: you cannot extract more information from a corpus than the corpus contains, and you cannot extend a corpus by generating text from a model trained on that corpus. The information is already inside the model. Writing it out and reading it back in does not create more of it.

6.5 Data decay in an AI-saturated internet

There is a second, more insidious problem. As AI-generated content proliferates across the internet, the quality of future training data degrades. Estimates suggest that by 2026, the majority of web content may be AI-generated. If future models scrape this AI-saturated web, they will be training on a mixture of human and synthetic data, unintentionally ingesting model collapse into their training pipeline.

The snake begins eating its tail not by design, but by accident. The commons is polluted. High-quality human-generated text, the irreplaceable substrate of language model training, becomes increasingly difficult to separate from synthetic imitations. This is not a hypothetical future risk. It is already happening.

6.6 Data decay: the evidence

The AI pollution of the internet is not speculation. It is measurable, observable, and accelerating.

Stack Overflow: In late 2022, Stack Overflow banned ChatGPT-generated answers after moderators observed a flood of plausible-sounding but often incorrect responses. Analysis by the community found that AI-generated answers had higher rates of subtle errors, misleading explanations, and hallucinated references. Despite the ban, enforcement is difficult: distinguishing AI-generated text from human-written text is non-trivial, and determined users continue posting AI-generated content. By mid-2023, estimates suggested 10-30% of new answers contained AI-generated components. The signal-to-noise ratio is degrading.

Wikipedia: Wikipedia editors have engaged in ongoing battles over AI-generated content. The English Wikipedia's policy prohibits submitting AI-generated text without human verification, but enforcement relies on volunteer moderators detecting subtle tells. Multiple studies have found AI-generated Wikipedia edits slipping through: articles created by LLMs, biographies with hallucinated details, citations to non-existent papers. The problem is worse in smaller language Wikipedias with fewer active moderators. Wikipedia's quality as a training corpus is declining.

Academic preprint servers: ArXiv, bioRxiv, and other preprint servers have seen a surge in papers with AI-generated sections or entirely AI-generated content. These range from papers using ChatGPT to write summaries (which may be acceptable) to entirely synthetic papers with fabricated results (which are not). Detection is difficult: modern LLMs generate grammatically correct, stylistically appropriate text that passes superficial review. Several high-profile retractions have occurred after peer review caught fabricated data in AI-generated papers, but many likely slip through.

News and content farms: Low-quality news sites and content farms have adopted AI generation at scale. Some sites publish hundreds of AI-generated articles daily, optimized for SEO but providing minimal information value. Google's search index is increasingly contaminated with this content. While Google's algorithms attempt to penalize low-quality content, the arms race between AI generation and detection favors generation. The median quality of web text is declining.

Social media: Twitter, Reddit, and other platforms report surges in bot activity using LLM-generated text. These bots engage in conversations, post comments, and generate content that appears human. Detection is difficult at scale. Reddit's r/SubSimulatorGPT2 demonstrates how convincing AI-generated posts can be; distinguishing them from human posts requires careful attention. As LLMs improve, the distinction becomes harder.

Code repositories: GitHub Copilot and competitors have led to a surge in AI-assisted code. While much of this code is functional, it also propagates common bugs, deprecated patterns, and security vulnerabilities that the model learned from training data. Code review catches some of this, but much is committed. Future models training on this code will learn not only from human-written code but from AI-generated code that may contain systematic errors.

Quantitative estimates: A 2023 study by researchers at AWS AI Labs estimated that by 2025, 50-90% of new text on the internet may be AI-generated or AI-assisted, depending on the domain. News articles, social media posts, and blog content are highest; academic papers and books are lowest (but still significant). By 2027, the median web page scraped for training data may be majority AI-generated.

This is not a hypothetical scenario in which future models might encounter data pollution. It is happening now. Models trained in 2025 and beyond will ingest this polluted data unless extraordinary effort is made to filter it out. But filtering is difficult: AI detection tools have false positive rates of 5-20%, meaning that aggressive filtering discards significant amounts of genuine human-generated content along with the AI-generated noise.

The ouroboros has begun. The snake is eating its tail. The commons is degraded.

6.7 Where synthetic data actually works

The picture painted so far is grim: synthetic data causes collapse, and AI pollution is degrading the training commons. But there are domains where synthetic data genuinely helps. Understanding where and why illuminates the fundamental constraint.

Image augmentation: In computer vision, synthetic data is standard practice. An image can be rotated, flipped, cropped, color-shifted, and noise-added to produce augmented examples. These transformations preserve the label (a rotated cat is still a cat) while increasing apparent dataset size. This works because the transformations are known, controlled, and do not introduce new information—they reveal invariances already present in the data. This is data augmentation, not data creation.

Physics simulations: In robotics, synthetic environments (simulators) generate unlimited training data for robot control policies. A robot arm learning to grasp objects can train in simulation, where thousands of parallel attempts cost nothing. This works because physics engines can accurately model rigid body dynamics, collisions, and sensor noise. The synthetic data is grounded in the same physical laws the robot will encounter in reality. Transfer from simulation to reality ("sim-to-real") requires domain adaptation but is often successful.

Formal domains: Synthetic data works well in mathematics, logic, and other formal systems. A theorem prover can generate unlimited problem-solution pairs by constructing proofs. A compiler can generate unlimited code-output pairs by executing programs. These synthetic examples are guaranteed correct by construction, because the domain has formal semantics. Models trained on synthetic formal data (e.g., AlphaGeometry, which generates synthetic geometry proofs) achieve strong performance.

Constraint-based generation: When the generation process is constrained by known rules, synthetic data can be valuable. For example, generating SQL queries from schemas, generating chemical formulas that obey valence rules, generating chess positions that obey game rules. The synthetic data is valid by construction, and the model learns the rule structure.

What these successes have in common: 1. The generation process is grounded in known, stable rules (physics, mathematics, formal semantics). 2. The synthetic data is used for augmentation or exploration within a bounded domain, not frontier extension. 3. Correctness can be verified independently (simulation matches reality, proofs are valid, code compiles).

Where synthetic data fails: 1. Open-ended natural language generation: there are no formal rules, no ground truth, no verifier. 2. Frontier knowledge creation: synthetic data cannot contain information not in the generating model. 3. Unverifiable domains: tasks where correctness cannot be checked automatically.

The constraint is clear: synthetic data works when grounded in verifiable structure but fails when asked to extend beyond the learned distribution. For language modeling, where the goal is to match the unbounded diversity of human-generated text, synthetic data leads to collapse unless carefully mixed with fresh human data—which brings us back to the data wall.

The ouroboros is not a solution to the data wall. It is the mechanism by which the wall becomes permanent.

6.8 Chapter summary

- Shumailov et al. (2024) demonstrated that models trained on AI-generated data undergo model collapse: distribution tails vanish, outputs become generic, diversity degrades
- Collapse happens because models are imperfect estimators; iterating any lossy compression pushes distributions toward lower entropy
- By generation 4-6 of recursive training, perplexity degrades 60-80%; the collapse accelerates rather than gradual
- Mixing real and synthetic data prevents regression but does not extend the frontier; synthetic data contains no information not already in the model
- The information conservation law: you cannot create knowledge by training on your own outputs
- Data decay is observable now, not future speculation: Stack Overflow (10-30% AI content), Wikipedia (AI edits slip through), arXiv (AI-generated papers), news sites (AI content farms)
- Estimates suggest 50-90% of new internet text may be AI-generated by 2025-2027; future models will train on polluted data
- AI detection tools have 5-20% false positive rates; aggressive filtering discards genuine human content along with synthetic noise
- Synthetic data works in constrained domains: image augmentation, physics simulations, formal systems (math, code), constraint-based generation
- These successes rely on grounded rules, verifiable correctness, and bounded domains; they do not generalize to open-ended natural language
- The ouroboros is not a solution; it is the mechanism by which the data wall becomes permanent

Part III

The realistic horizon

Chapter 7

The stagnation thesis

> "The low-hanging fruit has been picked." — Tyler Cowen

The previous six chapters have established a series of hard constraints. Each constraint alone would slow progress. Together, they compound into something more decisive: a ceiling. This chapter argues that frontier model capabilities will plateau within the next 3-5 years, not because we stop trying, but because we hit the convergence of limits that cannot be overcome on the current path.

7.1 The convergence of constraints

The gaps and bottlenecks are not independent problems. They interlock, and each attempted solution runs into another wall.

The learning instances gap. Chapter 1 established that biological intelligence required 10^{25} learning instances at the optimistic vertebrate-only estimate. Current training runs perform roughly 10^{7} gradient updates. The gap is eighteen orders of magnitude.

The architectural bottleneck. Chapter 2 demonstrated that one-shot learning with catastrophic forgetting cannot replicate the continuous consolidation cycle that enabled biological knowledge accumulation. Even if we could generate 10^{25} learning instances, the architecture cannot process them the way biology did. The path matters, not just the quantity.

The sensory bandwidth gap. Chapter 3 showed that text is a 10^{-8} compression of embodied experience. Language models train on the fraction of knowledge that survived articulation, not the full sensory bandwidth that grounded biological learning. The training signal lacks the information content.

Diminishing returns. Chapter 4 proved mathematically that each doubling of performance under the observed power law requires a million-fold increase in compute. Progress does not stop, but it decelerates to asymptotic approach toward a floor.

The data wall. Chapter 5 documented that high-quality human-generated text is finite at roughly 300 trillion tokens. The wall arrives around 2028. Models cannot scale beyond the data supply, and undertrained models perform worse, not better.

The ouroboros trap. Chapter 6 showed that synthetic data cannot extend the frontier. Models trained on their own outputs undergo collapse. The information conservation law holds: you cannot create knowledge from itself. Meanwhile, AI-generated content pollutes the web, degrading future training data quality.

These constraints compound. You cannot solve the data wall with synthetic data because of model collapse. You cannot brute-force the learning instances gap because of diminishing returns and the data ceiling. You cannot replicate embodied grounding because text lacks the information content, and multimodal observation is not interaction. You cannot overcome the architectural bottleneck without redesigning the learning paradigm, which would require starting from scratch with no guarantee of success.

The walls close in from all sides.

7.2 What stagnation looks like

Stagnation is not a hard stop. It is a deceleration to asymptotic improvement. The trajectory is visible in the historical record.

GPT-2 (2019): 1.5 billion parameters, impressive text generation, clearly superhuman at next-token prediction but limited reasoning capability.

GPT-3 (2020): 175 billion parameters, emergent few-shot learning, surprising breadth, but still fragile on tasks requiring robust reasoning.

GPT-4 (2023): estimated 1+ trillion parameters, multimodal, passes professional exams, writes working code, carries on sophisticated conversations. A large jump from GPT-3.

The jump from GPT-3 to GPT-4 took three years and required roughly 100x more compute. The improvement was real but not revolutionary in kind, only in degree. GPT-4 is a better predictor than GPT-3. It is not a different kind of system.

GPT-5, when it arrives, will be better still. It will saturate more benchmarks, pass more exams, write cleaner code. But the improvement will be smaller than the GPT-3 to GPT-4 jump, because we are further up the power law curve where gains are expensive. Each subsequent model will show diminishing improvements.

By GPT-6 or GPT-7, the gains will be imperceptible to most users. The model will have approached the asymptote: the best predictor possible given text training data, within the constraints of one-shot learning, under the power law that governs scaling.

7.3 The benchmark saturation cycle

Benchmarks will continue to improve, but benchmarks measure what is measurable, not what matters. When GPT-4 saturated undergraduate-level exams, the community created graduate-level benchmarks. When those saturate, we will create expert-level benchmarks. When those saturate, we will create adversarial benchmarks specifically designed to expose model weaknesses.

This is Goodhart's law applied to AI: when a measure becomes a target, it ceases to be a good measure. Models optimize for benchmarks, and benchmarks become proxies for capabilities rather than measurements of them. A model that scores 95% on a reasoning benchmark has not necessarily learned to reason; it has learned the statistical regularities of reasoning-like text.

The saturation cycle is already visible. MMLU, a broad knowledge benchmark, was considered challenging when introduced. Frontier models now exceed 85%. The response: create harder benchmarks. But harder is not the same as more meaningful. Eventually, every benchmark becomes a game that models learn to play through pattern matching on training data.

The underlying question is whether performance on text-based benchmarks, at any level of difficulty, constitutes the capabilities we actually care about: robust reasoning, causal understanding, novel problem-solving, grounded common sense. The stagnation thesis says no. Benchmarks will improve asymptotically, but the gap between "excellent text predictor" and "general intelligence" remains.

7.4 The capability plateau

Stagnated frontier models will plateau at a level best described as "impressively competent within distribution, fragile outside it."

Where they excel:

- Text generation and summarization: models are already near-human on these tasks and will approach indistinguishability.
- Code completion for common patterns: standard libraries, well-documented APIs, conventional algorithms.
- Question answering on well-documented topics: anything in the training corpus with sufficient examples.
- Translation between languages: statistical regularities are strong, performance is already high.
- Classification and pattern recognition: given labeled examples, models generalize well.
- Style matching and tone adaptation: mimicking writing styles is a pattern-matching task.

Where they remain weak:

- Novel reasoning requiring grounding not present in training data: models cannot deduce from first principles what they have not seen.
- Physical intuition: a model that has never interacted with objects cannot reliably predict "what happens if I drop this?"
- Causal understanding: correlation is in the data, causation is not. Models confuse the two.
- Genuine creativity: true novelty requires generating patterns not present in training data. Models recombine seen patterns.
- Robust common sense in unfamiliar situations: common sense is grounded in embodied experience. Text captures some of it but not the substrate.
- Out-of-distribution robustness: adversarial examples, distribution shift, novel contexts all expose brittleness.

This is not a temporary limitation awaiting more scale. It is the consequence of training on text, using one-shot learning, without embodied grounding. Scaling makes the within-distribution performance better, but it does not close the gap to capabilities requiring information not present in text.

7.5 The "good enough" threshold

Stagnation is not failure. For many applications, even a plateau at current frontier capability levels delivers enormous value.

Most enterprise tasks are well-documented and within-distribution. Code completion for standard libraries is useful even if the model cannot invent new algorithms. Email drafting and report summarization are useful even if the model cannot generate truly novel insights. Customer service for common questions is useful even if the model fails on edge cases. Translation is useful even if the model occasionally makes errors on idioms.

The economic value of "GPT-4 level but no better" is measured in trillions of dollars of productivity gains. Legal document review. Medical literature summarization. Software development acceleration. Content generation at scale. Personalized tutoring for standard curricula. These applications do not require AGI. They require competent text manipulation, and stagnated frontier models deliver that.

The stagnation thesis is not "AI is useless." It is "AI will not reach AGI on the current path, but will deliver enormous value at sub-AGI capability levels." The hype promised AGI by 2030. The reality delivers impressive, economically transformative, but decidedly non-general intelligence.

7.6 When do we hit each wall?

We can project timelines for each constraint based on current trajectories.

Data wall: 2026-2028. High-quality text is finite, model appetite grows exponentially, the curves cross within this window. Villalobos and colleagues estimated 2028 at the median. Some optimism comes from multimodal data extending the supply, but this delays the wall by 1-2 years, not a decade.

Compute ceiling: 2028-2030. The power law requires 10^6 more compute for each doubling of performance. Current frontier models consume 10^{26} FLOP. To double performance again requires 10^{32} FLOP. This exceeds projected global AI compute capacity in the next 5 years. Compute growth is slowing as Moore's Law decelerates and as energy and manufacturing constraints bind.

Capability saturation: 2027-2032. As models hit the data wall and compute ceiling, capability improvements decelerate. The compounding constraints converge. By 2030, frontier models will be measurably better than GPT-4, but not categorically different. By 2032, the improvements become marginal.

These are not precise predictions. They are informed projections based on observed trends and established constraints. The timeline could shift by a few years in either direction. But the qualitative outcome, stagnation within the next decade, is robust to uncertainty in the details.

7.7 Historical parallels

Technological progress often follows an S-curve: slow initial growth, rapid exponential improvement, then deceleration to a plateau as fundamental limits bind. AI is not the first technology to encounter this pattern.

Flight speed. The sound barrier was broken in 1947. By 1960, aircraft routinely flew at Mach 2. The SR-71, flying at Mach 3.3, set records in the 1960s that still stand. Hypersonic flight exists but remains experimental. Why? Because physics imposes hard limits. Air resistance scales with the square of velocity, heating scales with the cube. Beyond Mach 3, the engineering challenges become extraordinary and the returns diminish. We did not stop trying. We hit a wall.

Moore's Law. Transistor density doubled every two years from 1970 to 2010, driving exponential compute growth. Around 2010, the law began to slow. By 2020, the doubling time had stretched to 3-4 years. Why? Because quantum mechanics imposes limits at atomic scales. Gates are now a few nanometers wide, approaching the size of atoms. Further miniaturization faces physical barriers. We have not stopped trying. We are hitting a wall.

Energy efficiency of computation. The Landauer limit, derived from thermodynamics, sets a floor on the energy required to erase a bit: $kT \ln 2 \approx 3 \times 10^{-21}$ joules at room temperature. Current DRAM operates roughly 10^9 times above this limit. Progress toward the limit has been exponential for decades, but the limit is absolute. No technology can violate thermodynamics. We will approach the Landauer limit asymptotically but never breach it.

AI scaling faces analogous limits: finite data, power law diminishing returns, architectural bottlenecks rooted in the physics of computation. The pattern is familiar. Rapid progress during the exponential phase, then deceleration as the limits bind.

7.8 The emergence argument

Perhaps the strongest counterargument to stagnation is emergence: the observation that capabilities appear suddenly at scale that were not present or predictable at smaller scales. Wei and colleagues documented numerous examples in their 2022 paper: chain-of-thought reasoning, in-context learning, multi-step arithmetic, instruction following. These capabilities were not explicitly programmed or trained; they emerged from scaling.

The emergence phenomenon is genuine and was surprising. It suggests that scaling unlocks latent structure in the training data that smaller models cannot access. This is important: it demonstrates that scale is not merely quantitative improvement but can produce qualitative capability jumps.

But emergence has limits, and understanding those limits is critical to assessing whether it can overcome the gaps documented in earlier chapters.

Emergence within the training distribution. Every documented emergent capability can be traced to patterns present in the training data. In-context learning emerged because the training corpus contains examples of learning from context (few-shot examples in documentation, tutorials that build on previous concepts). Chain-of-thought reasoning emerged because the training data contains worked examples with explicit reasoning steps (textbooks, Stack Overflow answers, math forums). Arithmetic emerged because numerical patterns appear throughout text.

These are not trivial pattern-matching tasks. The model had to learn abstractions, generalizations, and compositional structure to exhibit these capabilities. But they are still recognition of patterns present in text, not generation of capabilities absent from text.

The "sparks of AGI" debate. Bubeck and colleagues at Microsoft Research published a provocative paper in 2023 titled "Sparks of Artificial General Intelligence: Early experiments with GPT-4." They documented impressive performance on novel tasks: drawing unicorns, solving theory-of-mind problems, generating Python code to visualize concepts. They argued that GPT-4's breadth and flexibility suggested early signs of general intelligence.

The paper sparked intense debate. Critics pointed out that all demonstrated capabilities, while impressive, involved recombination of patterns in the training data. Drawing a unicorn requires understanding "unicorn" (fantasy creature, horse-like body, single horn, often depicted in specific styles) and "drawing" (generating vector graphics code or describing visual appearance). Both concepts appear extensively in training data. The task requires creative synthesis, which the model achieves, but not generation of concepts genuinely absent from training.

Theory-of-mind tasks (understanding that others have beliefs different from one's own) appeared more challenging. But Sally-Anne tests and similar tasks appear in psychology literature, education materials, and discussions of cognitive development—all in the training corpus. The model likely learned the structure of these problems from seeing many examples, not by developing actual theory of mind through social interaction.

The "sparks" paper is valuable because it documents the frontier of what scaling has achieved. But it does not demonstrate that further scaling will produce capabilities qualitatively beyond what patterns in text can support. Emergence unlocks latent structure in training data; it does not create structure absent from training data.

Can embodied grounding emerge? This is the critical question. Chapters 1-3 documented that biological learning operated over 10^9 bits per second of sensory bandwidth, grounded in physical interaction with consequences (survival, reproduction). Text captures perhaps 10^{-8} of this experience. Can the missing 99.9999% emerge from scaling text prediction?

There is no evidence for this. The failures documented in Chapter 1 (novel physical reasoning, causal inference, compositional generalization in unfamiliar contexts, robust common sense) persist in GPT-4 despite its scale. These failures cluster precisely where embodied grounding is required. Scaling from GPT-3 to GPT-4 reduced the failure rate but did not eliminate the failure mode.

The hypothesis that embodied grounding will emerge from text alone requires believing that text contains the information needed for physical intuition, even though that information was explicitly compressed away when experience was articulated into language. This is not impossible, but it is a strong claim requiring evidence. The current evidence suggests the opposite: the gaps persist despite scaling.

7.9 GPT-5 and beyond: falsifiable predictions

The stagnation thesis makes predictions that can be tested against future model releases. If GPT-5 (or its equivalent from other labs) appears in 2025-2026, we can check whether it conforms to the predicted trajectory.

Prediction 1: Benchmark saturation. GPT-5 will achieve higher scores on standard benchmarks (MMLU, HumanEval, etc.) than GPT-4, but the improvement will be smaller than the GPT-3 to GPT-4 improvement. Expect 5-15 percentage point gains on MMLU (from GPT-4's ~86% toward 90-95%), not the 27-point jump from GPT-3 to GPT-3.5.

Prediction 2: Persistent failure modes. The failures documented in Chapter 1 will persist. GPT-5 will still struggle with: - Novel physical reasoning not present in training data (e.g., predicting outcomes of unfamiliar mechanical configurations) - Robust causal inference beyond memorized patterns - Compositional generalization in truly novel contexts - Out-of-distribution adversarial robustness

The failure rate will decrease, but the failure mode will remain. The model will be more often correct but not qualitatively more robust.

Prediction 3: Diminishing qualitative improvement. GPT-4's release felt like a capability jump: it could pass professional exams, write complex code, handle multimodal inputs. GPT-5 will feel like refinement: better writing quality, fewer hallucinations, faster inference, but not a new category of capability. Users will struggle to articulate what GPT-5 can do that GPT-4 could not, beyond "it's better."

Prediction 4: Training data exhaustion. GPT-5 will either (a) train on a similar token count to GPT-4 (~15-20 trillion tokens) but with better curation and architectural improvements, or (b) attempt to scale beyond 100 trillion tokens and encounter quality degradation from data reuse or lower-quality sources. If (b), performance on some benchmarks may plateau or even regress slightly.

Prediction 5: Economic value plateau. GPT-5 will be economically valuable (better coding assistants, better writing tools, better customer service bots) but not transform additional industries beyond what GPT-4 already enabled. The economic impact curve is flattening as the technology reaches saturation in applications where text manipulation suffices.

How to falsify the stagnation thesis: If GPT-5 demonstrates genuinely novel capabilities not derivable from patterns in text—robust physical reasoning, reliable causal inference, zero-shot compositional generalization to structures it has never seen, stable lifelong learning without catastrophic forgetting—then stagnation is falsified. If GPT-5 delivers qualitative jumps comparable to GPT-3 to GPT-4, and if subsequent models continue delivering such jumps without hitting the data wall, then the thesis is wrong.

But if GPT-5 conforms to the predictions above—incremental benchmark gains, persistent failure modes, diminishing qualitative improvement, data constraints binding—then the stagnation thesis is supported. The null hypothesis should be the trend: deceleration along a power law toward an asymptote.

7.10 The case against

"Emergent capabilities suggest we are on the cusp of a phase transition. More scale will unlock qualitatively new behaviors."

Emergent capabilities are real, but they are not magic. Wei and colleagues documented that certain capabilities appear suddenly at scale: chain-of-thought reasoning, in-context learning, arithmetic. These were surprising, and they demonstrated that scale unlocks latent structure in the training data.

But emergence within text does not imply that capabilities not present in text will also emerge. In-context learning emerged because the training data contains examples of learning from context. Arithmetic emerged because the training data contains numerical patterns. These are still pattern matching, just at higher abstraction. There is no evidence that capabilities requiring embodied grounding, causal reasoning not present in text, or true novelty will emerge from scaling text prediction, because the training signal does not contain them.

"Industry leaders are confident. They are building toward AGI."

Industry leaders have strong financial incentives to project confidence. Their companies are valued on the assumption of continued exponential progress toward AGI. Admitting stagnation would collapse valuations. This does not mean they are lying; it means their incentives are not aligned with dispassionate assessment.

Some leaders genuinely believe AGI is near. Belief is not evidence. The constraints documented in this book are empirical: finite data, power law diminishing returns, architectural bottlenecks, information-theoretic limits. Optimism does not override mathematics.

7.11 Why this diverges from the narrative

The dominant narrative in 2024-2025 is exponential progress toward AGI within a decade. This narrative serves many purposes: attracting investment, recruiting talent, justifying massive compute expenditures, generating media attention.

But stagnation is not failure. It is a realistic assessment of what the current path delivers. Frontier models plateau at a capability level that is genuinely impressive and economically transformative, even if it is not AGI. The sooner we accept this, the faster we can redirect resources toward what actually works: making stagnated-but-useful models radically cheaper, faster, and more accessible.

That is the subject of the next chapter.

7.12 Chapter summary

- Frontier capability gains will plateau within 3-5 years (2027-2032) due to compounding constraints from Chapters 1-6
- The constraints interlock: data wall + ouroboros trap + diminishing returns + architectural bottleneck + sensory bandwidth gap
- Stagnation is not a hard stop but deceleration to asymptotic improvement; GPT-3 to GPT-4 was a larger jump than GPT-4 to GPT-5 will be
- Benchmark saturation cycle: as models saturate existing tests, harder benchmarks are created, but this measures test-taking ability not robust intelligence
- Stagnated models will excel within distribution (text generation, code completion, Q&A on documented topics, translation) but remain fragile outside distribution
- The “good enough” threshold: GPT-4 level capability delivers enormous economic value (trillions in productivity) even without reaching AGI
- Historical parallels (flight speed, Moore’s Law, computational energy efficiency) show S-curve patterns where fundamental limits cause deceleration
- Emergent capabilities (chain-of-thought, in-context learning, arithmetic) are real but operate within training distribution; no evidence that embodied grounding will emerge from text alone
- Bubeck et al.’s “Sparks of AGI” documented impressive GPT-4 performance but all demonstrated capabilities involve recombination of training data patterns
- Falsifiable predictions for GPT-5: benchmark saturation (5-15pp gains not 27pp), persistent failure modes, diminishing qualitative improvement, data constraints binding
- The stagnation thesis is falsified if GPT-5+ demonstrates genuinely novel capabilities not derivable from text patterns
- Industry incentives (valuations depend on AGI narrative) create pressure to project confidence despite empirical constraints

Chapter 8

The efficiency revolution

“The future is already here — it’s just not evenly distributed.” — William Gibson

Frontier model capabilities will stagnate. But efficiency will not. While the performance ceiling is real, the cost to reach that ceiling will collapse. This chapter makes the case that within 10-15 years, models at current frontier quality will run locally on consumer devices at near-zero marginal cost. This is the genuine revolution: not AGI, but impressive machines everywhere.

8.1 The efficiency gap

The gap between capability and accessibility is enormous.

Current state of frontier models: - Training cost: \$100M-\$500M in compute expenditure for a single run - Inference infrastructure: requires datacenter deployment with thousands of GPUs - Access model: gated through API endpoints or subscription services - Latency: network round-trips add 100-500ms, queuing adds more during peak usage - Privacy: all queries transit through corporate servers - Marginal cost: \$0.01-\$0.10 per 1000 tokens, depending on model size

GPT-4 level intelligence exists, but most of the world cannot run it locally. The model weights occupy hundreds of gigabytes. Inference requires hardware most consumers do not own. The intelligence is concentrated in datacenters, accessed through network pipes.

This centralization is not permanent. It is an artifact of the current efficiency level. As efficiency improves, the frontier moves from datacenter to device.

8.2 Training cost collapse

Training frontier models is expensive primarily because we use brute force: massive models, enormous datasets, one-shot learning on hardware optimized for throughput rather than efficiency. Multiple pathways exist to reduce this cost by orders of magnitude.

8.2.1 Algorithmic improvements

Better architectures. The transformer, introduced in 2017, was not designed for efficiency. It was designed for expressiveness and parallelizability. Attention is quadratic in sequence length. Feed-forward layers are dense and parameter-heavy. These were acceptable tradeoffs when compute was the bottleneck, but as efficiency becomes the focus, architectural innovations deliver gains.

Mixture of experts (MoE) routes different inputs through different subnetworks, activating only a fraction of total parameters for any given token. Models like GPT-4 reportedly use MoE, achieving better performance per compute than dense models. Sparse models carry more parameters but use fewer during inference, reducing effective cost.

State space models (Mamba, Hyena) replace attention with linear-time sequence processing, eliminating the quadratic bottleneck. Early results suggest they match transformer performance on long sequences while using far less compute. Whether they scale to frontier quality remains an open question, but the direction is promising.

Improved optimizers. Gradient descent is not the only way to train neural networks, just the most established. Second-order methods, which use curvature information, converge faster but require more memory. Approximate second-order methods (K-FAC, Shampoo) capture some benefits with manageable overhead. Each generation of optimizers reduces the number of training steps required to reach a target loss.

Curriculum learning. The order in which a model sees data affects how efficiently it learns. Training on easy examples first, then harder ones, allows faster convergence than random sampling. Careful curriculum design can reduce required training compute by 2-5x.

Knowledge distillation. A large teacher model can train a smaller student model to match its performance through distillation: the student learns from the teacher's outputs rather than raw data. The student is cheaper to run, and distillation is cheaper than training from scratch. Distillation does not extend the capability frontier, but it democratizes access to frontier capabilities at lower cost.

Conservative projection: Compounding algorithmic improvements deliver 10-50x training cost reduction over 10 years.

8.2.2 Hardware improvements

Next-generation accelerators. NVIDIA's H100 GPU, released in 2022, is already being superseded by H200 and the upcoming B100 series. Each generation delivers roughly 2-3x improvement in FLOP per watt and FLOP per dollar. This is not Moore's Law, which has slowed, but it is sustained progress driven by specialized AI architectures, improved memory bandwidth, and better chip design.

AMD, Google (TPU), Cerebras, Graphcore, and others compete in the accelerator market. Competition drives innovation. Over 10 years, expect 10-30x improvement in training efficiency from hardware alone.

Neuromorphic approaches. IBM's NorthPole and Intel's Loihi represent a different paradigm: analog computation with co-located memory. These chips achieve 10-25x better energy efficiency than GPUs for specific workloads, primarily inference. Training on neuromorphic hardware remains experimental, but if successful, it could deliver another 10-100x efficiency gain.

Memory technology. HBM3 is the current standard for high-bandwidth memory. HBM4 is in development, promising higher capacity and lower energy per access. Memristors, which store weights in resistance states, remain in the lab but show potential for orders-of-magnitude improvement in energy efficiency. Whether memristors transition from research to production within 10 years is uncertain, but the trajectory is promising.

Conservative projection: Hardware improvements deliver 10-30x training cost reduction over 10 years.

8.2.3 Compounding training efficiency

Algorithmic and hardware gains multiply. Conservative estimates: $10 \times 10 = 100$x reduction in training cost over 10 years. Optimistic estimates: $50 \times 30 = 1500$x reduction.

What costs \$100M today might cost \$1M in 10 years, or potentially as little as \$100K in the optimistic case. This does not extend the capability frontier (stagnation still applies), but it makes reaching the frontier far more accessible. More organizations can afford to train frontier models. Fine-tuning becomes economically feasible for domain-specific applications. The centralization of frontier model development begins to erode.

8.3 Inference cost collapse

Training happens once. Inference happens billions of times. Inference efficiency is where the revolution actually occurs.

8.3.1 The inference bottleneck

Inference cost in transformers is dominated by three factors:

1. **Memory bandwidth.** Moving weights from memory to compute units costs energy and time. For large models, memory access dominates arithmetic operations.
2. **Precision overhead.** Models typically use 16-bit floating-point weights. Higher precision is unnecessary for inference but remains standard because training requires it.
3. **Quadratic attention.** Transformer attention scales as $O(n^2)$ in sequence length. Long contexts become prohibitively expensive.

Each bottleneck has solutions.

8.3.2 Quantization: reducing precision

Weights trained at 16-bit precision retain most of their functionality when compressed to lower precision. This is quantization: representing weights with fewer bits.

8-bit quantization: Reduces memory footprint and bandwidth by 2x. Quality loss is minimal for most tasks. This is already standard in production deployments.

4-bit quantization: Reduces memory by 4x. Quality degradation is measurable but acceptable for many applications. Recent methods (GPTQ, AWQ) achieve surprisingly good 4-bit performance.

2-bit quantization: Reduces memory by 8x. Quality loss becomes significant, but the model remains functional for simpler tasks.

1-bit (binary) quantization: Each weight is +1 or -1. Reduces memory by 16x. Recent work (BitNet) demonstrates that carefully trained 1-bit models retain substantial capability, though below full-precision frontiers.

Quantization is not free. Lower precision reduces quality. But the tradeoff is favorable: a 4-bit quantized GPT-4 might perform at 90-95% of full quality while running at 4x lower cost. For most applications, this is acceptable.

Projection: Quantization delivers 4-16x inference cost reduction with acceptable quality loss.

8.3.3 Sparsity: activating less

Most neural network activations are near zero. Sparse models explicitly zero out connections, activating only a fraction of the network for any given input. Mixture of experts is one form of sparsity. Magnitude-based pruning is another.

Sparse models reduce computation proportionally to sparsity. A 90% sparse model uses 10% of the compute of a dense model. The challenge is maintaining quality during pruning. Careful techniques (gradual pruning during training, structured sparsity that matches hardware) achieve high sparsity with minimal quality loss.

Projection: Sparsity delivers 5-10x inference cost reduction.

8.3.4 Architectural efficiency

Transformers are not the final word in neural architectures. Alternatives that reduce attention cost are under active development.

State space models: Mamba, Hyena, and related models replace quadratic attention with linear-time updates. Early results suggest they match transformer quality on long-context tasks while using far less compute. If this holds at frontier scale, state space models could deliver 10-100x reduction in inference cost for long sequences.

Mixture of experts: Route inputs through specialized subnetworks. Only a fraction of total parameters activate per token, reducing effective compute.

Local attention: Not every token needs to attend to every other token. Local attention (attending only to nearby tokens) plus occasional global attention (attending to key tokens) reduces quadratic cost while preserving most capability.

Projection: Architectural improvements deliver 5-20x inference cost reduction over 10 years.

8.3.5 Compounding inference efficiency

Again, the gains multiply. Conservative: $4 \times 5 \times 5 = 100$x. Optimistic: $16 \times 10 \times 20 = 3200$x.

What costs $0.05 per inference today might cost $0.0005 in 10 years (conservative) or $0.000015 (optimistic). Near-zero marginal cost. Trillions of inferences become economically feasible.

8.4 From datacenter to device

When inference cost drops 100-1000x, models that today require datacenter infrastructure become runnable on consumer hardware.

Timeline projection:

2026: GPT-3.5 equivalent (175B parameters, 8-bit quantized, sparse) runs on high-end laptops (64GB RAM, integrated GPU). Inference is slow (seconds per response) but functional. Privacy-conscious users adopt local models for sensitive queries.

2028: GPT-4 equivalent (1T parameters, 4-bit quantized, sparse) runs on high-end laptops. Inference speed approaches real-time (100ms per token). Smartphones begin running GPT-3.5 equivalent models. Local-first AI becomes mainstream.

2030: GPT-4 equivalent runs on mid-range laptops and high-end smartphones. Inference is fast (10-50ms per token). Edge devices (tablets, smart glasses) run GPT-3.5 equivalent models. Network dependence for AI evaporates.

2035: Frontier-quality models (GPT-4 or better) run on all consumer devices. Watches, glasses, earbuds, cars. Inference is near-instantaneous (1-10ms per token). Ambient intelligence: AI is everywhere, always available, offline-capable, private.

This is not science fiction. It is the compounding of demonstrated efficiency trends. The physics allows it. The engineering trajectory points toward it. The economic incentives demand it.

8.5 The neuromorphic wildcard

Everything discussed so far assumes digital von Neumann architecture. But biology achieves far greater efficiency with analog computation and co-located memory. What if silicon could capture some of biology's efficiency advantages?

Neuromorphic inference: IBM's NorthPole achieves 25x efficiency over GPUs for inference. Intel's Loihi 2 demonstrates event-driven spiking networks with minimal idle power. These are inference accelerators, not training platforms, but for frozen models, that is sufficient.

Challenges remain. Neuromorphic chips are limited in capacity (NorthPole holds ~6B parameters). Scaling to frontier model sizes requires multi-chip systems, which reintroduce communication overhead. Programming models are immature. But the physics is favorable: analog computation operates closer to the Landauer limit.

Projection (speculative): If neuromorphic inference matures, it delivers an additional 10-100x efficiency gain over digital inference. This is less certain than the other projections but physically plausible.

8.6 The compounding multipliers

Let us be conservative. Over 10-15 years: - Training efficiency: 100x (algorithmic + hardware) - Quantization: 4x (8-bit standard, 4-bit for edge) - Sparsity: 5x (structured pruning) - Architecture: 5x (incremental improvements) - Total inference: $4 \times 5 \times 5 = 100$x

Conservative total: 100x training, 100x inference. A model that cost $100M to train and $0.05 per inference in 2024 costs $1M to train and $0.0005 per inference in 2035.

Optimistic scenario: - Training efficiency: 1500x - Quantization: 8x (4-bit standard, 2-bit for edge) - Sparsity: 10x - Architecture: 10x - Neuromorphic: 10x (speculative) - Total inference: $8 \times 10 \times 10 \times 10 = 8000$x

Optimistic total: 1500x training, 8000x inference. A model that cost $100M to train and $0.05 per inference costs $67K to train and $0.000006 per inference.

Even the conservative case puts frontier-quality models on laptops. The optimistic case puts them on watches.

This is the realistic horizon: not AGI, but stagnated frontier capabilities becoming universally accessible through efficiency gains rather than capability scaling.

8.7 Chapter summary

- While frontier capability stagnates (Chapter 7), efficiency improvements will continue exponentially over the next 10-15 years
- Training efficiency gains: 10-50x from algorithms (better architectures, optimizers, curriculum learning, distillation), 10-30x from hardware (next-gen accelerators, neuromorphic approaches), conservative total 100x
- Inference efficiency gains: 4-16x from quantization (8-bit standard, 4-bit for edge), 5-10x from sparsity, 5-20x from architectural improvements, conservative total 100x, optimistic 3200x
- From datacenter to device timeline: GPT-3.5 on laptops by 2026, GPT-4 on laptops by 2028, GPT-4 on smartphones by 2030, frontier quality on all devices by 2035
- Neuromorphic wildcard: IBM NorthPole achieves 25x efficiency over GPUs for inference; if scaled to frontier models, could deliver additional 10-100x efficiency
- Compounding multipliers: conservative case (100x training, 100x inference) puts GPT-4 on laptops; optimistic case (1500x training, 8000x inference) puts GPT-4 on watches
- The efficiency revolution delivers what capability scaling cannot: stagnated frontier quality becomes accessible to everyone at near-zero marginal cost

Part IV

Appendices

Glossary

References