Member-only story

The Man Who Solved Learning in 1964 (And Why We Ignored Him for 60 Years)

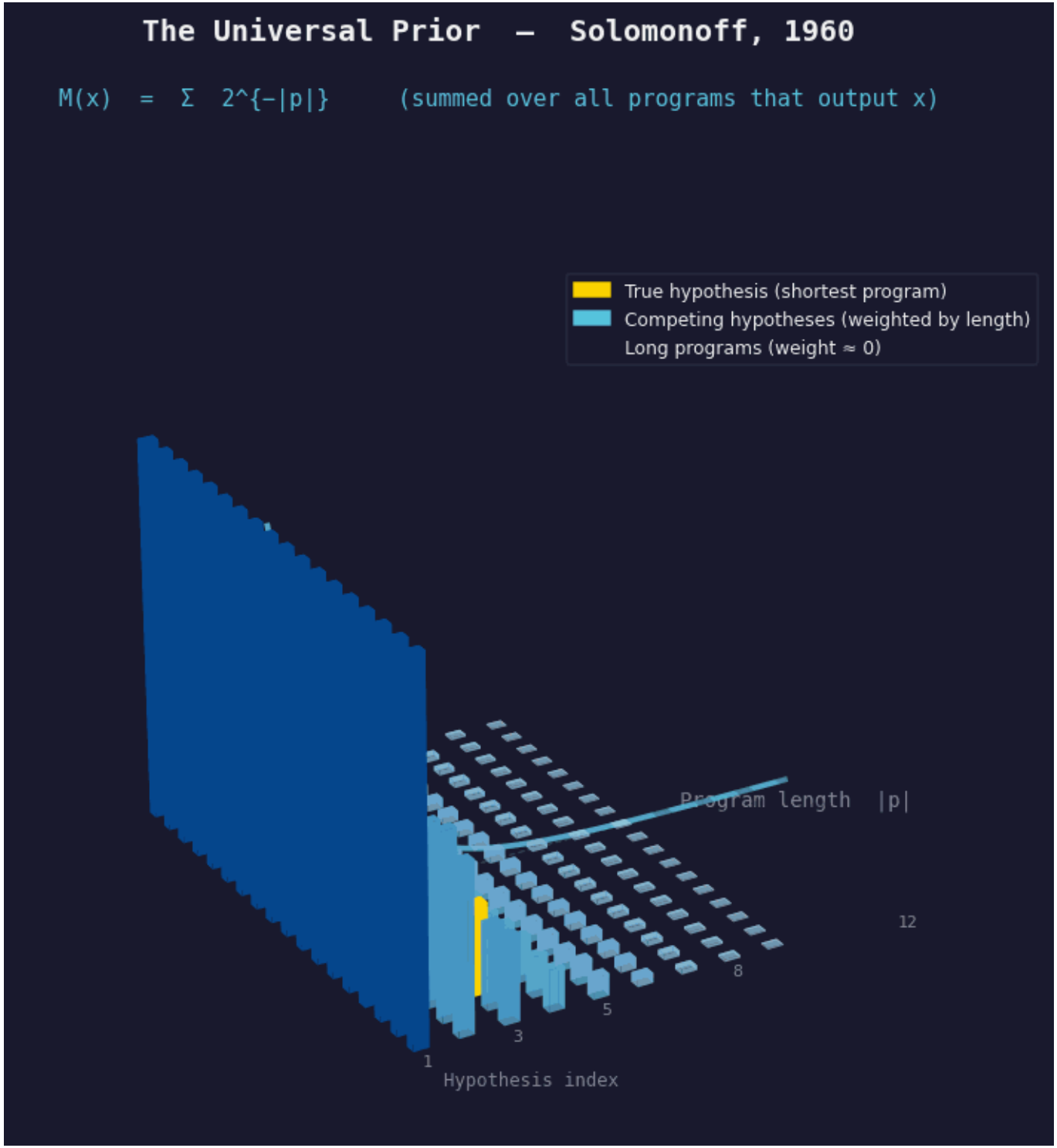
DrSwarnenduAI Following 12 min read · 2 hours ago

40 2

> We are spending $100 billion to build AGI. We are arguing about safety, alignment, and emergent capabilities. We are scaling transformers on nuclear power.
>
> The actual mathematics of optimal learning was solved in a one-man office in Cambridge, Massachusetts in 1964.

> The man's name was Ray Solomonoff. He proved how machines should learn. Then he put the convergence proof in a drawer because nobody cared.
>
> He was right. We were just slow.

> *I. The Dartmouth Snub*
>
> *A college in New Hampshire.*


The Universal Prior — Solomonoff, 1960
M(x) = Σ 2^{−|p|} (summed over all programs that output x)
True hypothesis (shortest program)
Competing hypotheses (weighted by length)
Long programs (weight ≈ 0)
Program length |p|
12
8
5
3
1
Hypothesis index

Universal Prior

Ten men. One summer. One question: *Can machines think?*

John McCarthy organized it. Marvin Minsky co-signed. Claude Shannon endorsed it.

They called the gathering the Dartmouth Conference. They named the field: Artificial Intelligence.

Nine of those men went home and built the AI you learned about in school. If-then statements. Rule-based logic. Expert systems. Programs that encoded human knowledge, explicitly, by hand.

Ray Solomonoff went home and built something else.

Not AI as a collection of rules. AI as **mathematics.**

Not: what rules do you give a machine? But: what is learning, precisely, formally — so you can prove what the optimal strategy is?

The nine built the AI everyone knows.

Solomonoff built the AI the universe knows.

**********Free version**********

II. Hume's Problem. Solomonoff's Solution.

David Hume, 1739.

The sun has risen every day in recorded history. Therefore it will rise tomorrow.

How do you justify that argument?

You can't. Logically. The sun could fail to rise. There is no deductive chain from "always has" to "always will."

Hume concluded: inductive reasoning is not logically justified. We believe in patterns because of habit. Not reason.

Philosophers spent two hundred years on this.

Solomonoff changed the question.

Not: *can you justify induction?* But: *among all possible inference strategies, which one is optimal?*

Reframe the question. Make it computable. Solve it.

III. Occam's Razor Isn't a Vibe. It's an Exponent.

You have heard the heuristic.

"The simplest explanation is usually the right one."

You have seen it on coffee mugs. Cited in arguments. Used to dismiss complex theories.

Solomonoff turned it into mathematics.

Fix a universal Turing machine U.

The **Solomonoff prior** M — the universal distribution:

```
M(x) = Σ_{p : U(p) = x*}  2^{-|p|}
```

Sum over every program p that produces a string starting with x. Each program contributes `2^{-|p|}` — weight decreasing exponentially with program length.

Short programs (simple explanations): high weight. Long programs (complex explanations): low weight.

He didn't suggest simplicity. **He enforced it.**

Not as a preference. As an exponent.

The more complex the explanation, the exponentially less weight it gets. Not linearly. **Exponentially.**

A program one bit longer gets half the weight. Two bits longer: a quarter. Ten bits longer: one-thousandth.

Occam's Razor is not a philosophical position. **It is the shape of the distribution over programs.**

IV. The Kolmogorov Injustice

Every textbook says: **Kolmogorov Complexity.**

Every lecture slides. Every Wikipedia article. Every citation.

The complexity of a string x:

```
K(x) = min{ |p| : U(p) = x }
```

The length of the shortest program that outputs x.

Named after Andrei Kolmogorov. Soviet mathematician. Giant of the 20th century. Published in 1965.

Solomonoff published in 1960.

Kolmogorov himself wrote, in 1968: *"I came to similar conclusions before becoming aware of Solomonoff's work, in 1963–1964."*

Solomonoff: 1960. Kolmogorov: 1963–1964. Kolmogorov paper: 1965.

The acknowledgment is in the record. The name that stuck is not Solomonoff's.

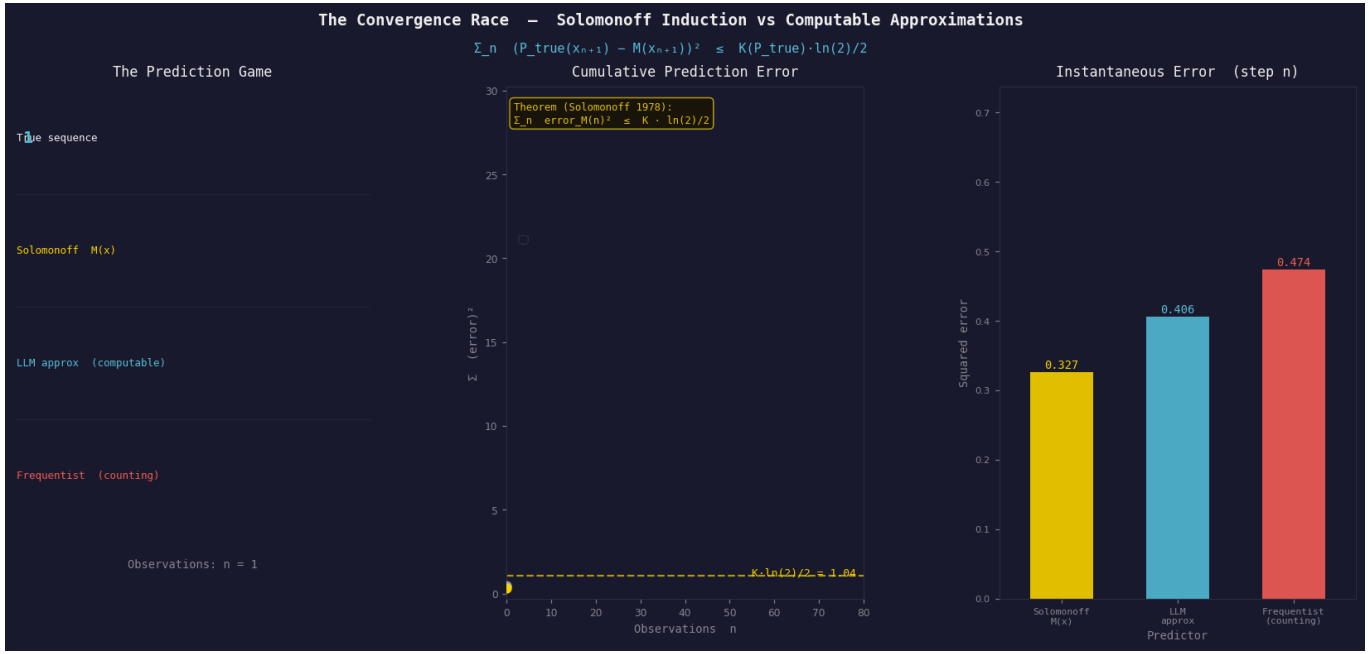
Why?

Kolmogorov was a Soviet titan with the institutional weight of Moscow State University.

Solomonoff was a one-man consulting firm. In Cambridge, Massachusetts. Called Zator Co.

Lesson: in science, priority goes to the man with the loudest megaphone. Not always to the man with the proof.

V. The Convergence Proof Nobody Wanted



Convergence proof

In 1968, Solomonoff found the theorem.

The result:

$$\Sigma_n \; (P\_true(x_{n+1} \mid x_1 \ldots x_n) - M(x_{n+1} \mid x_1 \ldots x_n))^2 \;\; \leq \;\; K(P\_true) \cdot \ln(2) \; / \; 2$$

The **total accumulated prediction error** of Solomonoff induction over an infinite sequence is bounded by a constant.

Not: eventually small. Not: converges in the limit.

The **entire sum** — every wrong prediction, ever — is bounded. Finite. Forever.

The bound: `K(P_true) · ln(2) / 2`.

Where `K(P_true)` is the Kolmogorov complexity of the true data-generating process.

Simple world: small K. Learns fast. Few total errors. Complex world: large K. Takes longer. Still converges.

The speed of learning is determined by the complexity of the truth.

This is the only learning algorithm with this guarantee. Not gradient descent. Not Bayesian inference with a parametric prior. Not PAC learning.

The only one.

He found it in 1968. He put it in a drawer. He published it in 1978. Ten years later.

Because at the time of discovery — nobody cared.

Think about that.

The most important theorem in the theory of learning. In a drawer. For ten years. Because symbolic AI had all the grants.

VI. The Incomputable Wall

Here is why you are not running M(x) on your laptop.

The Solomonoff prior M(x) is **incomputable.**

Not difficult. Not slow. **Incomputable.**

Formally. Provably. Absolutely.

Computing M(x) exactly requires knowing, for every program p, whether U(p) outputs a string starting with x.

But some programs run forever. They neither terminate nor fail to terminate. They just… run.

This is the **Halting Problem.** Turing, 1936.

There is no algorithm that decides, for all programs, whether they halt. Therefore: no algorithm can compute M(x) exactly.

And here is the knife:

The complete learner is incomputable. The computable learner is incomplete. You cannot have both.

This is the learning-theoretic analog of Gödel's incompleteness theorem.

Gödel: no consistent formal system can prove all true statements.
Solomonoff: no computable learning algorithm can discover all describable regularities.

Completeness and computability. Mutually exclusive.

Choose one.

The Table That Should Be in Every AI Textbook

	SYMBOLIC AI (The 9 who went home)	SOLOMONOFF / ALGORITHMIC A (The 1 who stayed)
Core question	What rules do you give it?	What is optimal inference?
Approach	Encode human knowledge	Weight all explanations
Occam's Razor	Heuristic (coffee mug)	Theorem (exponent)
Handles uncertainty	Poorly	Exactly optimally
Learning guarantee	None	Convergence theorem
Computable?	Yes	No (that's the point)
Current relevance	Dead (mostly)	Every LLM is approximating
Who won?	Lost the 80s AI winter	The universe

VII. Your LLM Is Solomonoff-Lite

Here is the claim that should break something in your head.

Every large language model is a **computable approximation** to M.

Train GPT on a corpus of text. The training objective:

```
maximize  Σₙ  log P_θ(xₙ₊₁ | x₁...xₙ)
```

Find parameters θ that assign high probability to observed sequences. Regularization penalizes complex parameter settings. Dropout, weight decay, architecture constraints — all bias toward simplicity.

This is Solomonoff's principle encoded in stochastic gradient descent.

Not exactly. Not incomputably. **Approximating.**

The neural network is the hypothesis class. Training is the search for the shortest description. Cross-entropy loss is the `L(data | model)` term. Regularization is the `L(model)` term.

Minimum Description Length. Rissanen, 1978.

The computable version of M. The bridge between Solomonoff's 1964 incomputable ideal and the 2017 transformer.

GPT-4 is not magic. GPT-4 is a 1964 formula running on 10,000 H100s.

VIII. Why Hallucinations Are Not Bugs

This is the section the safety teams don't want to read.

Hallucinations are not bugs. They are not a training data problem. They are not a fine-tuning problem. They are not a prompt engineering problem.

They are the tax we pay for running the incomputable on silicon.

Here is why.

M(x) — the complete learner — considers all programs. All possible explanations. All possible regularities. Including the ones that require solving the Halting Problem to find.

Any computable approximation to M is implicitly excluding a set of programs. A set of possible explanations. A set of possible truths.

The excluded programs correspond to patterns the model cannot represent. Not because it hasn't seen enough data. Not because the architecture is too small. Because the pattern requires more computational depth than the hypothesis class allows.

When the true explanation is outside the hypothesis class, the model finds the closest approximation in the hypothesis class. That approximation is confident. It sounds fluent. It is wrong.

That is a hallucination.

The model is not malfunctioning. The model is doing exactly what a computable approximation to M should do. It is finding the best available explanation. The best available explanation is not the true explanation. Because the true explanation is outside its reach.

Solomonoff proved this was unavoidable in 1964.

The bound on total error:

```
K(P_true) · ln(2) / 2
```

If the true explanation has high Kolmogorov complexity — if the truth is deep and complex — the model accumulates more error before converging. The domain is harder.

Hallucination rate correlates with the Kolmogorov complexity of the domain. Medical specifics: high K, high hallucination. Common conversation: low K, low hallucination.

Not because of bad training. Because of the convergence theorem.

IX. The Interactive Proof (Colab Demo)

The difference between Frequentist and Solomonoff induction is easiest to see on a simple sequence.

Run this in Colab:

```
from fractions import Fraction
from collections import Counter
import numpy as np

def frequentist_predict(sequence):
    """
    Frequentist prediction: count symbol frequencies.
    Ignores all structure. Just counts.
    """
    if not sequence:
        return {"prediction": "?", "method": "No data"}
    counts = Counter(sequence)
    total  = len(sequence)
    probs  = {k: v / total for k, v in counts.items()}
    pred   = max(probs, key=probs.get)
    return {
        "prediction":   pred,
        "probabilities": {k: f"{v:.3f}" for k, v in probs.items()},
        "method":        "Frequency counting — ignores all pattern"
    }

def solomonoff_predict(sequence):
    """
    Solomonoff-inspired prediction: weight by program complexity.
    Detects period-2 and period-3 patterns.
    """
    if not sequence:
        return {"prediction": "?", "method": "No data"}

    candidates = {}

    # Hypothesis: period-k repetition
    for k in range(1, min(len(sequence), 8)):
        pattern = sequence[:k]
        reconstructed = (pattern * (len(sequence) // k + 1))[:len(sequence)]
        if list(reconstructed) == list(sequence):
```

```
            # Program length ≈ k (length of pattern) + overhead
            program_length = k + 3
            weight = 2 ** (-program_length)
            next_symbol = pattern[len(sequence) % k]
            candidates[f"repeat({pattern})"] = (weight, next_symbol)

    # Hypothesis: last-seen frequency (longer program — less weight)
    counts = Counter(sequence)
    for sym, cnt in counts.items():
        weight = 2 ** (-len(sequence))  # complex hypothesis, low weight
        candidates[f"freq({sym})"] = (weight * cnt / len(sequence), sym)

    if not candidates:
        return frequentist_predict(sequence)

    # Weight all hypotheses — Principle of Multiple Explanations
    symbol_weights = {}
    for name, (w, sym) in candidates.items():
        symbol_weights[sym] = symbol_weights.get(sym, 0) + w

    total_w = sum(symbol_weights.values())
    probs   = {k: v / total_w for k, v in symbol_weights.items()}
    pred    = max(probs, key=probs.get)

    best_pattern = max(candidates.items(), key=lambda x: x[1][0])

    return {
        "prediction":    pred,
        "probabilities": {k: f"{v:.3f}" for k, v in probs.items()},
        "best_hypothesis": best_pattern[0],
        "method": "Weighted sum over all programs — Solomonoff principle"
    }

# ── THE DEMO ──────────────────────────────────────────────────

sequences = [
    [1, 1, 2, 1, 1, 2, 1, 1, 2],          # Period-3: obvious pattern
```

```
    [1, 2, 1, 2, 1, 2, 1, 2],              # Period-2: alternating
    [1, 1, 1, 1, 1, 1, 1, 1],              # Period-1: constant
    [1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2],    # Period-3: longer
    [1, 2, 3, 4, 5, 6, 7],                 # No obvious pattern
]

for seq in sequences:
    print(f"\nSequence: {seq}")
    print(f"  True next symbol (if pattern holds): "
          f"{seq[len(seq) % (next(k for k in range(1,len(seq)) if seq == (seq[:k
          f"(check manually)")
    print()

    freq = frequentist_predict(seq)
    solom = solomonoff_predict(seq)

    print(f"  FREQUENTIST:  predicts → {freq['prediction']}")
    print(f"    probs: {freq['probabilities']}")
    print(f"    method: {freq['method']}")
    print()
    print(f"  SOLOMONOFF:   predicts → {solom['prediction']}")
    print(f"    probs: {solom['probabilities']}")
    print(f"    best hypothesis: {solom.get('best_hypothesis', 'N/A')}")
    print(f"    method: {solom['method']}")
    print("-" * 65)
```

The first sequence — `1, 1, 2, 1, 1, 2, 1, 1, 2` — is the telling one.

Frequentist: predicts 1. Because 1 appears two-thirds of the time.
Solomonoff: predicts 2. Because period-3 is the simplest program that explains the data.

The next symbol is 2.

Frequentist is confidently wrong. Solomonoff finds the pattern.

This is the difference between counting and understanding.

X. The Scaling Obsession as Solomonoff Search

Here is the reframe that should end every "bigger is better" argument.

Why does scaling work?

The scaling hypothesis: more parameters, more data, better performance.

This is empirically true. The scaling laws hold.

But why?

Solomonoff's framework gives the explanation.

LLMs are searching for the minimum description length explanation within the hypothesis class of functions expressible by neural networks.

More parameters: larger hypothesis class. More data: more evidence to distinguish between hypotheses. Better performance: the hypothesis class now contains programs closer to M.

Scaling is brute-force Solomonoff search.

You are not discovering a better algorithm. You are expanding the search space to include programs closer to the incomputable ideal.

The convergence theorem tells you the bound: `K(P_true) · ln(2) / 2`

The total error is bounded by the complexity of the truth. More compute gets you closer to the bound. It does not get you past it.

The ceiling is not engineering. The ceiling is Kolmogorov complexity. The ceiling was measured in 1964.

Solomonoff watched us start the race. He already knew where the finish line was. He couldn't tell us, because we weren't listening.

XI. AIXI: The Machine That Cannot Be Built (But Specifies Everything We're Building)

Marcus Hutter, 2000.

He took Solomonoff induction and added action.

AIXI — the theoretically optimal intelligent agent:

```
a_t = argmax_{a_t}  Σ_{e_t}  M(e_t | x₁a₁...x_{t-1}a_{t-1}) · r(e_t)
```

At each step: choose the action that maximizes expected reward, where expected value is computed using the Solomonoff prior over all computable environments.

AIXI is the theoretically optimal general intelligence. Also incomputable. Same reason.

But AIXI gives us the **specification.**

Not a blueprint. A formal statement of what the thing would be.

Every RL agent is approximating AIXI. GPT with RLHF is approximating AIXI in a bounded domain. The reward model approximates the true reward function. The policy approximates the AIXI policy.

AGI — if we build it — will be a computable approximation to AIXI.

Whether or not anyone building it has read Hutter. Whether or not they know Solomonoff's name. Whether or not they have opened the 1964 paper.

Because AIXI is not a design choice. It is the definition of optimal.

Mathematical Truth:

There is a formal specification of the ideal learning agent. Two mathematicians nobody cites wrote it. Every AI system on the planet is an approximation to their equations. The approximation quality is what we call "intelligence."

XII. The Ghost in Every Model

Here is the image I keep returning to.

Cambridge, Massachusetts. A small office. Not MIT. Not Harvard. Zator Co.

One man.

Writing papers on the theoretically optimal learning algorithm. Knowing it is incomputable. Publishing anyway.

Finding the convergence proof in 1968. Putting it in a drawer. Because nobody cared.

Not quitting.

Writing more papers. Making the argument again.

For twenty years.

While the field built symbolic AI. While the expert systems got the grants. While the rule-based reasoners got the professorships.

He was right the whole time. He knew he was right. He kept working.

In 2003, the University of London gave him the **Kolmogorov Medal.**

Named after the man whose name is on the theorem Solomonoff proved first.

He accepted graciously.

He understood how this works. Ideas are larger than names. Theorems outlast their discoverers.

He died December 7, 2009. Cambridge, Massachusetts.

"Attention Is All You Need" was published eight years later.

ChatGPT launched thirteen years later.

He missed it.

The Scorecard

Solomonoff proved (1960–78)	Where we see it today
Optimal inductive learner	Every LLM (approximate version)
Occam's Razor is a theorem	Regularization in every training loop
Convergence to truth	Scaling laws (empirical measurement of K)
Complete = incomputable	Hallucinations (tax on computability)
MDL principle	Cross-entropy + weight decay
Universal prior	Pre-training distribution
AIXI (with Hutter)	RLHF (approximate version)

Truth Bomb

We are spending $100 billion on something a man in a one-room consulting firm described in a 1964 paper in *Information and Control.*

He described the optimal algorithm. He proved it converges. He proved it is incomputable. He proved hallucinations are mathematically unavoidable. He proved scaling is the right strategy. He proved the ceiling.

He proved all of it.

Then he put the proof in a drawer. For ten years. Because nobody cared.

The AI field spent two decades building systems that couldn't survive contact with a fact they hadn't been given.

Then it admitted defeat and switched to probability. Then it invented neural networks. Then it invented transformers. Then it invented ChatGPT.

And called it unprecedented.

It is not unprecedented.

It is **delayed.**

Delayed sixty years. By the wrong paradigm. By institutional weight. By a man's name on a theorem it wasn't his.

Ray Solomonoff is not the forgotten father of AI.

He is the ghost in every model you have ever used.

The formula is running right now. In approximate form. On a server somewhere.

```
M(x) = Σ 2^{-|p|}
```

It is not running his name. It is running his mathematics.

The mathematics was always the point.

If this put a name to something you didn't know was nameless — forward it to whoever last said "scaling will solve everything."

Scaling will solve everything up to K(P_true) · ln(2) / 2.

Solomonoff measured the ceiling in 1964. We are still building toward it. And calling the distance "progress."

Follow the Journey: For more deep-dives into the mathematical foundations of ML and AI, follow on LinkedIn https://www.linkedin.com/in/swarnendu-

bhattacharya/

- 📖 https://medium.com/@swarnenduiitb2020
- 📧https://substack.com/@swarnenduai
- 💻 **GitHub** — https://github.com/MLDreamer
- ☕ Support my work here → buymeacoffee.com/drswarnenduai

AI | Artificial Intelligence | Data Science | Towards Data Science | Towards Ai

Written by DrSwarnenduAI

1.6K followers · 938 following

I don't beleive in AI magic,I believe in the Matrices.

Following

Responses (2)

Alex Mylnikov

What are your thoughts?

Curt Welch
56 mins ago

I like all that.

But it has little to do with AGI and why we don't have robots acting like humans yet.

Physics creates a special type of regularity in the universe that allows us to build the apparently impossible.

The complexity of the real... more

Reply

Fernando Dougnac
59 mins ago

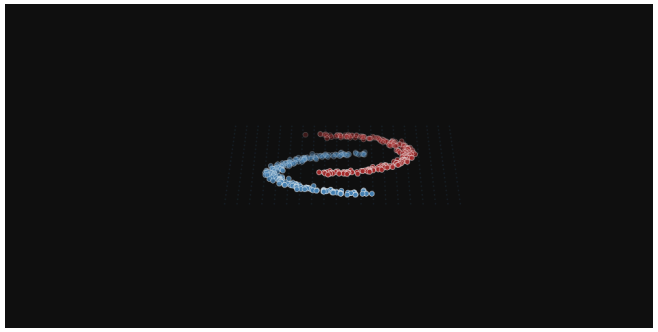
Nice article, like a poem.

A mathematical one ;)

I'll need to think deeply about it, but if hallucinations are "mathematically unavoidable," we will need to rethink what we can expect from the development of AI and its use.

Reply

More from DrSwarnenduAI



In Data And Beyond by DrSwarnenduAI

The Activation Bible: Your Network Doesn't 'Learn'—It Bends Space...

Part 1 of The Deep Learning Bible: Your neural network doesn't learn. It bends space until t...

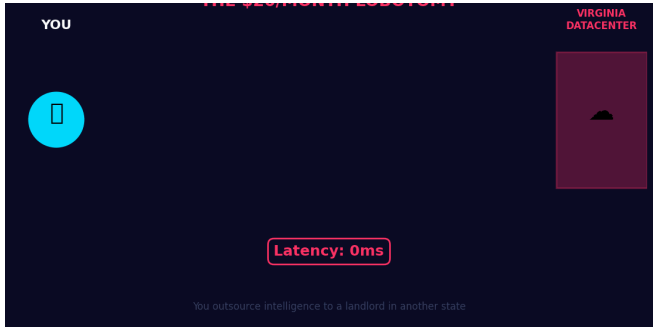
Feb 8 176 2

In Data And Beyond by DrSwarnenduAI

The P vs NP of AI: Why "Reasoning" is Mathematically...

Your chatbot isn't thinking. It's guessing. Complexity theory proves it.

5d ago 157 4



In Artificial Intelligence in Plain ... by DrSwarnen...

Stop Renting Your Brain for $20/Month. Own the Skull.

OpenClaw hit 100K GitHub stars in 60 days. It's not a tool. It's a jailbreak.

Feb 2 129

In Data And Beyond by DrSwarnenduAI

Why the AI is Hiding in the High-Dimensional Haystack

They built the world's most powerful microscope.

Feb 9 194

See all from DrSwarnenduAI

Recommended from Medium

In Activated Thinker by Shane Collins

Why the Smartest People in Tech Are Quietly Panicking Right Now

The water is rising fast, and your free version of ChatGPT is hiding the terrifying,...

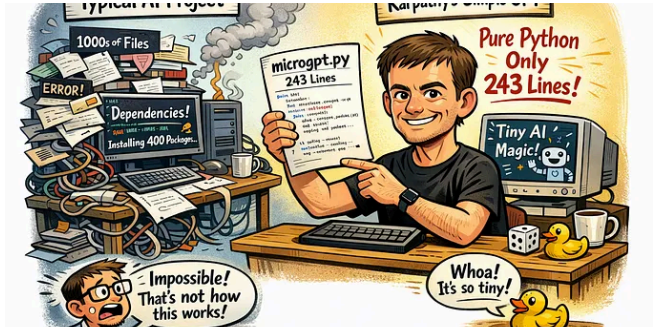
Feb 13 9.6K 415

Steve Yegge

The Anthropic Hive Mind

As you've probably noticed, something is happening over at Anthropic. They are a...

Feb 6 3.5K 80



In Towards Deep Learning by Sumit Pandey

Andrej Karpathy Just Built an Entire GPT in 243 Lines of Python

In Realworld AI Use Cases by Chris Dunlop

My friend tried Claude Code and wants to quit his job. Here is what ...

No PyTorch. No TensorFlow. Just pure Python and basic math.

Feb 15 1.95K 29

He built it in an afternoon. Should he quit his job—or is this just Claude Code dopamine...

Feb 13 396 20

Dr. Jerry A. Smith

Why Consciousness Can't Be Reduced — And Mathematics...

A new argument for the irreducibility of consciousness, grounded not in philosophy...

Feb 7 177 27

Vagelis Plevris

Poincaré's Conjecture: The Century-Old Mystery Finally Solved

The mathematical breakthrough and the man who refused a one-million-dollar prize.

Feb 11 851 11

See more recommendations