LLMs Are Neither Mere Statistics Nor True Minds

A Functional Account of Reasoning in Large Language Models

Author: Leszek J. Cierniak
Email: leszek.cierniak@gmail.com
Date: April 2026

Abstract
Large language models (LLMs) can solve many mathematical, logical, and physical problems with striking competence. This essay argues that LLMs are neither mere statistical pattern matchers nor fully human-like minds. They internalize statistically learned procedures for symbolic processing through training on vast datasets. They can emulate algorithmic behavior through learned inference-like computations and chain-of-thought generation, yet they do not obviously possess stable semantics, an explicitly inspectable world model, or intrinsic understanding. Drawing on computational theory, probabilistic perspectives, and recent empirical evidence-including chain-of-thought prompting, the GSM-Symbolic benchmark, and test-time compute reasoning models-the essay suggests that LLMs can produce behavior functionally equivalent to reasoning without necessarily sharing its classical foundations. The conclusion is not that the question is settled, but that LLMs deserve a distinct category in how we think about intelligence, explanation, and design.

Keywords:
Large Language Models, Chain-of-Thought Reasoning, Amortized Inference, Mechanistic Interpretability, Functional Intelligence, AI Alignment, Neurosymbolic Systems, Philosophy of Mind.

1. Introduction

The debate around LLMs is usually too blunt: either they are *just* statistics or they are already *minds*. That way it hides what is most interesting about them. The better question is how far statistical learning can go when it is scaled to human language, human reasoning traces, and enough compute to discover reusable procedures.

LLMs solve many mathematical, logical, and physical problems by internalizing patterns of knowledge, rules, and procedures from training data and by learning how to compose those pieces in new contexts. They do not operate as explicitly programmed symbolic systems, and they do not obviously possess autonomous, general intelligence in the cognitive sense. Even so, they can produce outputs that behave like reasoning: stepwise, adaptive, and often surprisingly general. These systems do not operate according to explicitly programmed algorithms, nor do they possess autonomous, general intelligence in the cognitive sense. However, they can simulate reasoning processes through statistically learned transformations that functionally resemble inference and problem-solving.

Discussion of LLMs often falls into an oversimplified dichotomy:

> Either it is **just** statistics, or it is **already** intelligence.

Synthesis of the thesis:

LLMs are neither purely *rule-based* systems nor systems possessing full *cognitive intelligence*; rather, they are *models* that **imitate** reasoning, through statistical learning over vast corpora of human reasoning artifacts, internalize patterns of knowledge and procedures, and then use them to generate behavior functionally equivalent to reasoning.
Their capabilities and failure modes are shaped in part by the structure of human cognition as recorded in text.

2. Computation and representation

From a computational perspective, an LLM is best understood as a highly parameterized function that maps token sequences to token sequences. It is not a hand-coded algorithm, but neither is it an empty black box; its learned weights can encode distributed procedures that behave algorithmically in context.

From the perspective of computation theory, an LLM can be modeled as a function:

$$f_\theta : \Sigma^* \to \Sigma^*$$

where:

- Σ^* - the set of all finite sequences over a token vocabulary Σ (used here loosely, in the spirit of formal language theory, to denote the space of all token sequences),
- θ - the model parameters (neural network weights).

The model does not contain an explicit, human-readable algorithm (e.g., a step-by-step Turing machine), but instead:

- approximates a transformation function from input to output,
- does so via iterative prediction of successive tokens.

Key observation:

Although an LLM does not explicitly contain **algorithms**, it can **emulate** their behavior.

Examples:

- solving linear equations,
- performing inductive proofs,
- applying algebraic transformations.

In computational terms, this implies that the model approximates a **class of procedures**, rather than a single function.

Recent mechanistic interpretability research (Elhage et al., 2021) supports this view: individual attention heads and MLP layers within transformer circuits can be reverse-engineered to implement recognizable algorithmic sub-routines - induction heads that perform in-context lookup, or circuits that compose factual relations. This suggests that f_θ is not a monolithic black box but a composition of learned micro-algorithms distributed across the parameter space. These micro-algorithms are not explicitly represented as symbolic procedures but are implemented in the weights - a distinction between the format of the algorithm (distributed, continuous) and its functional role (stepwise, compositional).

3. Bayesian Perspective: Inference as Estimation

Probabilistically, an LLM estimates the next token given the preceding context. That simple description hides something more interesting: in reasoning tasks, the model often behaves as if it is constructing a latent solution path before emitting the final answer.

From a probabilistic perspective, an LLM estimates:

$$P(x_t \mid x_1, x_2, ..., x_{t-1})$$

That is the probability of the next token given the context.

However, in *reasoning* tasks, something subtler occurs:

- The sequence of tokens (e.g., intermediate reasoning steps) becomes a **latent intermediate variable**.
- The model generates sequences that maximize the global coherence of the solution.

This can be interpreted as a form of **amortized Bayesian inference** - an interpretive analogy rather than a formal identity. In classical amortized inference (as in variational autoencoders), an explicit inference network approximates a posterior over latent variables. LLMs do not have a factored generative model they are formally inverting. Rather, the analogy holds in the following sense: instead of performing inference from scratch at each call, the model has effectively "learned what such inference looks like" across millions of training examples, pre-computing the expensive reasoning steps into fast feed-forward passes.

Xie et al. (2022) formalize a related connection: in-context learning can be interpreted as implicit Bayesian inference over a latent concept variable, where the model's pretraining implicitly fits a prior over tasks. Under this reading, what looks like flexible reasoning at inference time is the forward pass of a learned posterior approximator - inference without an explicit inference procedure.

4. Internalization of Rules vs. Explicit Representation

Classic symbolic systems made their rules visible. LLMs absorb regularities into distributed parameters, which makes them more flexible but harder to inspect. That tradeoff is one reason they feel intellectually powerful and philosophically unsettling at the same time.

Classical AI systems:

- operated on explicit rules (e.g., expert systems),
- had clearly defined operators and logic.

LLMs:

- do not store rules in symbolic form,
- instead, rules are **distributed across the parameter space**.

This leads to a fundamental distinction:

Feature	Symbolic Systems	LLMs	LLM + Symbolic Hybrid
Rule representation	explicit	implicit	explicit + implicit
Interpretability	high	low	medium
Flexibility	limited	high	high
Generalization	narrow	broad (statistical)	broad + verifiable
Verifiability	high	low	high

The hybrid column previews an important engineering direction: neurosymbolic systems that combine LLM flexibility with the verifiability of symbolic world models - discussed further in Section 10.

5. Emergent Procedures and the “Simulation of Reasoning”

One of the most important discoveries in recent years is that LLMs:

- can generate **chains of steps** resembling reasoning (*chain-of-thought*),
- improve performance when they **unfold** intermediate steps.

This does not prove that they reason in the human sense, but it does show that they can approximate the outward structure of reasoning remarkably well.

> The LLM model does not merely **know answers**, but can **construct a process** leading to the answer.

However, this is not classical reasoning, but rather a simulation of reasoning based on statistical patterns.

> The model does not know **why**, but it knows **what** a correct explanation looks like.

Empirical Grounding

Recent empirical work strongly supports this simulation view. Chain-of-Thought prompting (Wei et al., 2022) and its successors show that performance scales dramatically when the model is forced to externalize intermediate steps - exactly what one would expect from a system that has internalized procedures rather than facts. However, even the best 2025–2026 reasoning models still exhibit brittle abstraction: they can prove theorems in one axiomatic system but fail when the notation changes slightly (Mirzadeh et al., 2024, GSM-Symbolic benchmark).

An additional probe of genuine generalization is the ARC-AGI benchmark (Chollet, 2019), which is designed specifically to resist pattern-matching by requiring novel, out-of-distribution analogical reasoning. Despite dramatic progress on standard benchmarks, LLMs and even specialized reasoning systems have historically struggled with ARC-AGI - suggesting that the simulation of reasoning has hard limits precisely where training distribution coverage runs out.

6. Test-Time Compute and the Reasoning Model Frontier

A significant recent development - only partially addressed in prior analyses - is the emergence of models that spend substantial compute *at inference time* by generating and evaluating extended reasoning traces before producing an answer. OpenAI's o1/o3 series and DeepSeek-R1 are the most prominent examples.

At first glance, these systems appear to represent a qualitative leap: rather than producing a single forward-pass answer, they search over many candidate reasoning paths, pruning and extending based on self-evaluated coherence. This might seem to move beyond **simulation** toward genuine **deliberation**.

However, the thesis of this article accommodates this development cleanly. The distinction that matters is:

- **Searching over simulated reasoning traces** - what o1-style models do: generating multiple chain-of-thought continuations and selecting among them, all still grounded in learned distributional patterns.
- **Learning to reason** - a hypothetical future capability in which a system discovers new logical principles not represented in training, generalizing in a provably structure-preserving way.

Test-time compute improves accuracy by making the simulation more thorough - exploring more of the learned procedure space - not by introducing a new kind of epistemic access to truth. It is amortized inference at greater depth, not a categorical departure from it. The GSM-Symbolic brittleness persists even in these models when problems are sufficiently out-of-distribution, which is the empirical signature of simulation rather than understanding.

Test-time compute does not abandon amortized inference; it *composes* it. Each candidate reasoning trace in the search tree is generated by the same amortized approximator f_θ in a single forward pass. The search process then selects among these amortized outputs, possibly extending or combining them. The model never performs *online* Bayesian updating from scratch - it simply runs its fast, learned approximator many times and aggregates the results. Thus, test-time compute increases

coverage of the amortized hypothesis space, not the *kind* of inference performed. The distinction between amortized (one pass) and online (iterative updating) remains intact, and LLMs, even with search, fall squarely on the amortized side.

This matters for engineering: scaling test-time compute is a powerful lever, but it does not eliminate the need for verifiable grounding. A system that searches harder over plausible-sounding reasoning traces remains susceptible to confidently wrong conclusions if those traces are not anchored to a reliable external model.

7. LLMs as a Distillation of Human Reasoning Traces

A useful way to think about LLMs is as a compression of human textual cognition. They are trained less on direct experience than on the traces people leave behind: explanations, proofs, code, arguments, and corrections. In that sense, they inherit not only knowledge, but also many of the habits, shortcuts, and errors of human reasoning.

A dimension largely absent from the *statistics-vs-intelligence* debate is the nature of the training data itself. LLMs are not trained on raw sensory experience or formal axioms - they are trained predominantly on *human reasoning artifacts*: mathematical solutions, proofs, code, argumentative essays, scientific papers, forum discussions, and textbooks. The corpus is not merely a collection of facts; it is a vast record of human cognition in action.

This reframes the *simulation* language in an important way. An LLM is not simulating reasoning from first principles - it is *approximating and interpolating* from a *compressed distillation of billions* of human reasoning traces. When a model correctly solves a novel calculus problem, it is not deriving the answer from mathematical axioms; it is *synthesizing* through a high-dimensional space shaped by every calculus textbook, worked example, and tutoring session in the training corpus.

This has two important consequences:

For capabilities:
LLM performance will closely track the coverage and quality of human reasoning in the training distribution. Where humans reason well and abundantly (algebra, formal logic, code), LLMs perform well. Where human reasoning artifacts are sparse, ambiguous, or systematically biased (novel physical scenarios, uncommon formal systems), performance degrades - not because the model *doesn't understand* in some metaphysical sense, but because the distillation has less signal to reconstruct from.

For limitations:
Errors are not **random** - they are **structured** by the *biases, shortcuts, and common mistakes* present in human reasoning artifacts. An LLM may reproduce a common student error in calculus not despite its training, but because of it. This distinguishes LLM failure modes from those of symbolic systems and has direct implications for reliability engineering.

7.1 Open Questions

The Distillation Hypothesis as a Research Program. The framing of LLMs as distillations of human reasoning traces is not merely a descriptive metaphor - it is a testable hypothesis that opens a distinct research agenda.

Several questions remain insufficiently explored:

1. To what degree do LLM error distributions correlate structurally with known human cognitive biases - anchoring, availability, base-rate neglect, or the conjunction fallacy - and can this correlation be measured systematically across model families and scales?
2. Does scaling the training corpus reduce or amplify inherited biases, given that larger corpora may simultaneously include more corrections and more instances of the original errors?
3. If LLM failure modes are shaped by the reasoning artifacts of a particular cultural and linguistic majority, what are the implications for models deployed across genuinely different epistemic communities?

Finally, the distillation framing carries a direct consequence for alignment: if a model's errors are human-shaped rather than random, then standard red-teaming and adversarial probing - designed to surface arbitrary failure modes - may systematically miss the most consequential ones, precisely because those failures will look plausible to human evaluators trained in the same cognitive tradition. Investigating the structure of LLM errors through the lens of cognitive science, rather than purely through benchmark accuracy, may therefore be one of the most practically important directions for both reliability engineering and alignment research.

8. Limitations: Where the Model Fails

The strongest objections to the idea that LLMs are minds come from their failure modes. They can be eloquent while wrong, sensitive to phrasing, and weak at preserving global consistency across long chains of thought. Those weaknesses do not refute their usefulness, but they do shape how far the analogy to understanding can reasonably be pushed.

The thesis finds strong support in the observed limitations of current LLMs. Crucially, these limitations are not arbitrary - they follow from the formal and probabilistic structure described in Sections 2–3.

- **Lack of stable semantics.**
  The model may generate plausible steps while still making errors, with no guarantee of logical consistency. This follows from the distributional nature of f_θ: the function is optimized for token-level coherence across the training distribution, not for semantic truth preservation. A sequence that *looks* like a valid proof is rewarded similarly to one that *is* a valid proof, if human-written training data does not reliably distinguish them.
- **Sensitivity to form.**
  Rephrasing a problem can change the outcome, reflecting incomplete abstraction. This is a direct consequence of the smoothness of f_θ over the training distribution but the absence of invariance guarantees outside it. The model has no symbolic representation of *this is the same problem stated differently* - it has only the *distributional similarity* of token sequences.
- **No explicit, inspectable, and persistent world model grounded in stable ontology.**
  Knowledge is stored in distributed correlations across parameters rather than in a structured, queryable representation. This means the model cannot reliably enforce global consistency across a long reasoning chain, since there is no external structure to check intermediate conclusions against.

8.1 Multimodal and Embodied Systems

The analysis in this essay is scoped to text-based LLMs trained predominantly on linguistic reasoning artifacts. This boundary deserves explicit acknowledgment. Vision-language models - such as GPT-4V or Gemini - introduce perceptual grounding that partially addresses the stable semantics limitation described above: when a model can anchor a token like "cube" to a visual representation rather than purely to its distributional neighborhood in text, the gap between symbol and referent narrows in a non-trivial way. Embodied systems that act in physical or simulated environments go further still, acquiring a form of sensorimotor grounding that language-only models entirely lack. Whether these extensions are sufficient to cross any meaningful threshold - toward stable semantics, genuine world-model grounding, or invariant abstraction - remains an open empirical question. What is clear is that the failure modes described in this section are not necessarily permanent features of the architectural family, but may instead reflect the specific poverty of text as a sole training signal. A complete account of machine reasoning will need to revisit each of the limitations catalogued here against the richer grounding conditions these systems provide.

9. Is This Intelligence?

The answer depends on what one means by intelligence. A narrow definition tied to consciousness and intrinsic understanding gives one answer; a functional definition tied to adaptive problem-solving gives another. The dispute is therefore partly empirical and partly philosophical, which is why it persists even as the systems themselves improve.

The answer depends on the definition of intelligence itself. Two major philosophical traditions offer sharply different verdicts:

Definition of Intelligence	Philosophical Tradition	LLM Status
Consciousness, semantic understanding, autonomous goals	Phenomenological / Strong AI (Searle's Chinese Room)	No
Ability to solve novel problems, adaptive use of knowledge, generalization	Functionalism / Weak AI (Turing, Newell & Simon)	Yes - functional (weak) intelligence

Under a functionalist reading, LLMs are already intelligent; under any reading that requires grounded semantics or intrinsic intentionality, they are sophisticated simulators.

A third position, due to Dennett's *intentional stance*, offers a pragmatic resolution. Dennett argues that attributing beliefs, desires, and rationality to a system is justified whenever doing so reliably predicts its behavior - regardless of the system's internal substrate or whether it *really* has those mental states.
Under this view, the question *does the LLM understand?* is less important than *does treating it as if it understands yield accurate predictions?*
For many practical purposes, the answer is **yes** - which explains both the remarkable utility of LLMs and the persistent intuition that something mind-like is occurring, without requiring strong metaphysical commitments.

Searle's Chinese Room objection remains the sharpest challenge:

> A system can manipulate symbols according to rules and produce correct outputs without any semantic understanding of what the symbols mean.

The LLM is, in this sense, a very large and flexible room - one that has memorized not fixed rules but statistical regularities over an enormous rule-space. Whether this difference of degree becomes a difference of kind is precisely the open question this article cannot resolve, and arguably no current framework can.

10. Why This Matters

This debate matters because the category we choose changes how we build, evaluate, and trust these systems.

For science:
A need for new conceptual models situated between *algorithm* and *statistics*. Mechanistic interpretability (Elhage et al., 2021) is one promising direction - attempting to reverse-engineer the micro-algorithms distributed across transformer circuits.

For engineering:
Development of hybrid Neurosymbolic systems (LLMs combined with symbolic reasoning tools) that mix flexibility with the ability to verify results. The goal is to keep the broad generalization of LLMs while adding the transparency and consistency of symbolic systems. In practice, this could mean LLMs proposing reasoning steps, which are then checked by a formal proof system or a constraint solver.

For alignment and safety:
This is arguably the most urgent implication. If reasoning is *simulated* rather than *intrinsic*, the entire *chain-of-thought* must

be treated as potentially **deceptive** or **self-deceptive** - not necessarily through any intent, but as a structural consequence of the system's nature.

For example, a model asked to verify a mathematical proof might generate a step-by-step verification that looks logically structured but contains a subtle sign error, and because the model has no internal truth-checker, it will not flag the error - it will simply continue generating plausible text.

Consider a model trained to produce coherent-looking reasoning traces is rewarded for *plausibility*, not for *truth*. Under distribution shift or adversarial prompting, plausible-looking reasoning and correct reasoning can diverge significantly. The model has no internal alarm that fires when its *chain-of-thought* drifts from valid inference - it only has the distributional pressure to produce text that resembles valid inference. This means that interpretability of the *chain-of-thought* is not sufficient for alignment; the *chain-of-thought* itself is a generated artifact, not a transparent window into the model's *actual* computation. Alignment strategies that rely on reading and evaluating model reasoning - constitutional AI, debate, scalable oversight - must grapple with the possibility that the reasoning trace is itself a product of the same simulation process, not an independent check on it.

For alignment and safety:
Dennett's intentional stance sharpens this concern in a non-obvious way. Constitutional AI, debate, and scalable oversight all operate by treating the model as if it has beliefs and intentions - asking it to evaluate, criticize, or defend its own reasoning. But if the intentional stance is merely a pragmatic interpretive posture rather than a description of the system's actual computational structure, then these methods are using a useful fiction to audit a process that does not, in fact, have the internal organization the fiction assumes. The model is not checking its reasoning - it is generating text that resembles checking. Whether that resemblance is sufficient for alignment purposes is an open and urgent empirical question, not one that can be settled by the plausibility of the output alone.

For philosophy:
Redefinition of understanding. The question is no longer simply *does this system reason?*, but whether the distinction between *simulated* and *genuine* reasoning is itself well-defined - and whether it matters for practical or moral purposes. If a system reliably produces rational outputs, passes every behavioral test, and its failure modes are structurally analogous to human cognitive biases, at what point does the label **simulation** become a *metaphysical* rather than a *functional* claim?

11. Conclusion

LLMs are not simple rule-based tools; they are not fully intelligent in the human sense either. They are a new type of system that can mimic reasoning very well, while still lacking true understanding and stable grounding in the world.

Treating them as if they have beliefs or goals (as in Daniel Dennett’s intentional stance) is useful for predicting behavior, but this is a modeling convenience - not a statement about what they actually are. As LLMs become more capable, relying on this “as-if” interpretation can introduce real risks.

Their reasoning works in practice but is not genuine inference; it is a functional imitation built on patterns learned from human knowledge, without the robustness or consistency of true understanding.

The key question is no longer whether LLMs are truly intelligent, but:

If simulated reasoning becomes indistinguishable from real reasoning in output, does the difference still matter?

- For engineering and safety: yes, it matters critically
- For philosophy: the answer remains open

That tension is what makes current AI development significant.

References

- Anthropic. (2023). *A Mathematical Framework for Transformer Circuits*.
- ARC Prize. (2026). *What is ARC-AGI?* 4. Chollet, F. (2019). *On the Measure of Intelligence*. arXiv:1911.01547.
- DeepSeek-AI. (2025). *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. arXiv:2501.12948.
- Dennett, D. C. (1987). *The Intentional Stance*. MIT Press.
- Mirzadeh, I., et al. (2024). *GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models*. arXiv:2410.05229.
- OpenAI. (2024). *Introducing OpenAI o1-preview*.
- OpenAI. (2024). *Learning to reason with LLMs*.
- Searle, J. R. (1980). *Minds, brains, and programs*. Behavioral and Brain Sciences.
- Wei, J., et al. (2022). *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. NeurIPS 35.
- Xie, S. M., et al. (2022). *An Explanation of In-Context Learning as Implicit Bayesian Inference*. ICLR 2022.