The Magic of Learning on the Fly: How LLMs Understand Examples Without Retraining

Introduction: The Puzzle of In-Context Learning

Imagine you're teaching a friend a new slang word, like "grok." You don't give them a dictionary definition. Instead, you provide a few examples:

• "That explanation was so clear, I totally grok it now."
• "She has a real talent for looking at complex data and just groking the underlying trend."
• "He didn't grok the instructions, so he assembled the chair backwards."

Instantly, your friend understands the nuance and can use the word correctly in a new sentence. This remarkable ability to learn from a handful of examples on the fly is exactly what Large Language Models (LLMs) like GPT-3 can do. This capability is called In-Context Learning (ICL). It allows an LLM to perform a new task based only on the examples you provide in the prompt, all without the slow and expensive process of retraining its core parameters.

This raises a fascinating question: How do LLMs achieve this powerful form of "on-the-fly" learning without actually "learning" in the traditional sense? What's happening inside the model that allows it to generalize from just a few examples? This document will unravel this mystery, starting with a popular but incomplete theory.

1. A Popular Theory: Are LLMs Secretly Bayesian?

To explain the magic of in-context learning, researchers first turned to a powerful idea from statistics: Bayesian inference. Think of this as "the science of updating your beliefs based on new evidence." For example, if you think there's a 50% chance of rain, but then you see dark clouds, you update your belief to a much higher probability.

This framework was elegantly applied to LLMs. The theory proposed that an LLM's vast pre-training on internet-scale text gives it a "prior belief" over thousands of possible tasks or concepts it might be asked to perform. The examples you provide in the prompt act as "evidence." The LLM uses this evidence to rapidly deduce which specific task you want it to do.

The LLM isn't learning a task from scratch; it's using the examples to figure out which of the many tasks it already knows is the right one for the current prompt.

This explanation was neat, powerful, and backed by early evidence. It seemed to solve the puzzle of how ICL works. However, further experiments revealed a surprising crack in this theory, leading to a deeper and more accurate understanding.

2. A Crack in the Theory: The Problem with Order

A core principle of ideal Bayesian reasoning is exchangeability. Imagine you have a bag of marbles and you draw them one by one. The final count of red versus blue marbles doesn't depend on the order in which you drew them. Drawing Red, Blue, Red gives you the same final belief about the bag's contents as drawing Red, Red, Blue.

For a true Bayesian system, this means the order of evidence shouldn't change the final conclusion. This specific consequence is known as the martingale property. If an LLM were a perfect Bayesian reasoner, the order of the examples in its prompt shouldn't affect its final prediction.

Here’s the conflict: when researchers tested top models like GPT-3 and GPT-4, they found that the models systematically violate this property. The research provides a precise mathematical fingerprint for this error, quantifying it as being on the order of $\Theta(\log n/n)$, where 'n' is the number of examples. This means the violation is real and predictable; it gets smaller as more examples are added, but slower than one might

expect. This created a major paradox.

Ideal Bayesian Theory	Observed LLM Behavior
An LLM's prediction should be consistent regardless of the order of examples.	An LLM's prediction can be significantly influenced by the order of examples.
This implies LLMs are performing ideal Bayesian inference.	This suggests a fundamental incompatibility with ideal Bayesian inference.

How could LLMs achieve Bayesian-level performance while violating a fundamental rule of Bayesian inference? This puzzle pointed toward an answer not in statistical theory alone, but in the very architecture of the models themselves.

3. The Resolution: It's an Architectural Feature, Not a Bug

The key to resolving this paradox lies in a fundamental component of the Transformer architecture: Positional Encodings. Because the core "attention" mechanism in an LLM sees all words at once, it has no inherent sense of order. Positional encodings are signals added to the data that tell the model where each word is in a sequence. This is what allows a model to understand the critical difference between "dog bites man" and "man bites dog." Order is not an afterthought; it is an essential, built-in feature.

This architectural necessity directly explains the paradox. Because transformers are designed to be sensitive to order via positional encodings, it is impossible for them to be perfectly order-agnostic and satisfy the martingale property for any single ordering of examples.

This leads to the core resolution of the paper: LLMs are not designed to be perfectly Bayesian on one specific sequence of examples. Instead, they are optimized to be as efficient as a Bayesian learner on average, across all possible orderings of the examples.

Transformers are "Bayesian in expectation, not in realization."

In other words, the model's sensitivity to order is not a flaw. It's a trade-off. While it prevents the model from behaving like a perfect Bayesian on any single prompt, it is precisely this mechanism that allows it to achieve incredible statistical efficiency when averaged across all possibilities. This new understanding isn't just a theoretical curiosity; it has profound, real-world consequences.

4. The "So What?": Why This Insight Matters

Understanding this architectural trade-off allows us to build better, faster, and more reliable AI systems. Here are the key practical implications:

• Better Uncertainty Estimates By recognizing that an LLM's answer can be biased by the order of examples, we can get a more reliable sense of its confidence. A technique called permutation averaging—shuffling the examples multiple times and averaging the model's final answers—can reduce the variance (the "wobble") in its predictions by a factor of 4.

• More Efficient AI This theory helps optimize advanced prompting techniques like Chain-of-Thought (CoT), where a model "thinks out loud" to solve a problem. The research reveals a surprising 'law of diminishing returns' for AI thinking, captured in the formula for the optimal number of reasoning steps: $k^* = \Theta(\sqrt{n} \ldots)$. This shows that the ideal amount of thinking doesn't grow lock-step with the problem size (n), but with its square root. The reason is a direct trade-off: the model must balance the benefit of more thinking against the increasing "positional degradation" or noise that comes from a longer sequence. By understanding this, we can prevent wasted computation, reducing

costs by a staggering 80-90%.
• Deeper Understanding of AI Limits The theory includes an "Incompleteness Theorem" for LLMs. It proves that generating a chain of thought isn't just a neat trick—it is a theoretical necessity for an LLM to solve problems that are more complex than the knowledge stored in its parameters. This tells us that "thinking" is a fundamental requirement for LLMs to overcome their built-in limitations.

This deeper, more nuanced understanding moves us from seeing LLM behavior as magic to understanding it as a result of specific, analyzable design choices.

5. Conclusion: From Magic to Understanding

We began with the seemingly magical ability of In-Context Learning, where LLMs learn on the fly. We explored the elegant Bayesian theory that attempted to explain it, only to find a puzzling contradiction: LLMs are sensitive to the order of examples, a trait that violates a core Bayesian principle. The resolution came from looking at the model's very architecture.

The single most important takeaway is that an LLM's design creates a fundamental trade-off. Positional encodings, essential for understanding language, break the strict rules of Bayesian inference for any single prompt. However, this same mechanism allows the model to achieve incredible statistical efficiency on average, across all possible orderings.

This synthesis of architectural reality and statistical theory does more than just solve a puzzle. It gives us a clearer picture of how these powerful models work, revealing their limitations and providing practical tools to make them more efficient, reliable, and ultimately, more understandable.