LLM Capabilities and Limitations

What LLMs are genuinely strong at, why hallucination happens, the effect of knowledge cutoff, reasoning limits, and how bias in training data gets reproduced.

Intermediate 20 min

Last updated: August 20, 2026
Read online: https://www.learnforgex.com/en/topics/llm-capabilities-and-limitations

---

LLM Capabilities and Limitations

This is the fourth and final lesson in the "Large Language Models" category, and it's the one the last three lessons have been quietly building toward. "How Large Language Models Work" explained the mechanism, "Tokens and Context Windows" covered its hard limits, and "Prompting and Prompt Engineering" covered how to use it deliberately -- this lesson asks the question that actually matters when you're deciding whether and how to rely on an LLM for something: what can it genuinely do well, and where does it reliably fall short? Getting this honest is essential groundwork for the categories that follow, "Tools & MCP" and "AI Agents," both of which exist largely to work around the limitations covered here.

What LLMs Are Genuinely Good At

Before covering limitations, it's worth being concrete about real strengths, since both matter for making good decisions about when to use an LLM. Modern LLMs are strong at: transforming text (summarizing, translating, rewriting in a different tone), drafting and explaining code, answering questions that depend on general knowledge present in their training data, following well-specified instructions (recall "Writing Effective Prompts"), and picking up a pattern from a few examples (recall "Zero-Shot, Few-Shot, and Examples in Prompts"). These strengths all trace back to the same root described in "How Large Language Models Work": pretraining on enormous amounts of text produces genuinely broad, flexible capability at manipulating and generating language.

Hallucination: Confident, Fluent, Wrong

"Generative AI" introduced **hallucination** as "when a generative AI system produces fluent, confident output that is factually incorrect" -- this lesson explains *why* it happens, given everything you now know about how LLMs work. Recall from "How Large Language Models Work" that pretraining's entire objective is predicting a *plausible* next token, not a *verified*

one -- there is no built-in mechanism that checks generated text against ground truth before producing it. When a model doesn't have a fact available (either it wasn't in pretraining, or it's outside the current context), it doesn't have a reliable way to say "I don't know" by default -- it produces the most statistically plausible continuation, which can read as a specific, confident, well-formatted fact that simply isn't true. Fluency and correctness, as "Generative AI" first pointed out, are entirely separate properties, and hallucination is what happens when the first is present without the second.

> Hallucination is not rare or exotic -- it's a direct, structural consequence of how these models generate text, present in every LLM to some degree. Never treat a fluent, confident-sounding answer as automatically correct, especially for specific facts, dates, citations, or numbers -- verify anything that actually matters.

Knowledge Cutoff and Missing Recent Information

Recall from "How Large Language Models Work" that a model's **knowledge cutoff** is the point after which its pretraining data stops -- it has no information about anything genuinely new after that date, no matter how it's asked. This is a distinct limitation from hallucination (a model confidently inventing information) but the two often show up together: a model asked about something after its cutoff may either say it doesn't know, or -- worse -- hallucinate a plausible-sounding but fabricated answer instead of recognizing the gap. This specific limitation is exactly what motivates the "Tools & MCP" category later in this course: giving a model access to current, external information through tools rather than relying solely on what was baked in during pretraining.

Reasoning Limits

LLMs can produce reasoning that looks step-by-step and logical (recall chain-of-thought prompting from "Prompting and Prompt Engineering"), and this often genuinely improves accuracy on multi-step problems. But it's worth being precise about what's actually happening: the model is generating text that resembles reasoning, one token at a time, based on patterns learned during pretraining -- not executing a guaranteed, formally verified logical process the way a calculator or a compiler does. On tasks requiring precise arithmetic, exact multi-step logic, or careful tracking of many constraints at once, LLMs can and do make real errors, sometimes while producing an explanation that sounds entirely confident and coherent. This doesn't mean chain-of-thought techniques are useless -- they measurably help -- it means their output should be checked, not trusted by default, on anything where being wrong actually matters.

Bias in Training Data

Recall from "Pretraining: Learning From the World's Text" in "How Large Language Models Work" that a model learns its patterns from an enormous sample of existing human-written

text. That text reflects the real biases, stereotypes, and imbalances present in its sources -- a model trained on it can reproduce those same patterns in its output, even without anyone intending it to. This isn't a separate, unrelated flaw from hallucination or reasoning limits -- it's the same underlying fact (a model reflects statistical patterns in its training data) showing up in a different way. Being aware of this is especially important for any application where a model's output could unfairly affect real people (screening, ranking, or evaluating people, for example).

Why These Limitations Exist

It's worth stepping back and connecting all four limitations above to one shared cause, rather than treating them as a list of unrelated bugs. Everything in this lesson traces back to the same mechanism from "How Large Language Models Work": an LLM predicts plausible continuations of text based on statistical patterns learned during pretraining -- it does not have a built-in fact-checker, a formal logic engine, a live connection to current events, or an explicit bias-correction process. Every strength covered in "What LLMs Are Genuinely Good At" and every limitation covered in this lesson comes from that exact same source. This reframing matters practically: it's why later categories in this course (Tools & MCP, AI Agents) don't try to "fix" the underlying model -- instead, they build systems *around* it that supply current information, verify outputs, and constrain what the model is allowed to do, working with these limitations rather than assuming they'll simply go away.

Best Practices

- Verify specific facts, numbers, dates, and citations rather than trusting fluent output by default (see "Hallucination: Confident, Fluent, Wrong") -- especially in any context where being wrong has a real cost.
- For anything genuinely time-sensitive, check whether the information could fall after the model's knowledge cutoff (see "Knowledge Cutoff and Missing Recent Information") before relying on an unverified answer.
- Treat step-by-step reasoning output as a useful aid, not a guarantee (see "Reasoning Limits") -- check the actual logic on anything where an error would matter.
- Consider who could be affected by a model's output before deploying it in any context involving decisions about real people (see "Bias in Training Data").

Common Mistakes

- **Treating a confident-sounding answer as evidence of correctness.** As "Hallucination: Confident, Fluent, Wrong" explained, fluency and accuracy are unrelated properties of an LLM's output.

- **Assuming a model will say "I don't know" when it lacks information.** As covered in "Hallucination: Confident, Fluent, Wrong" and "Knowledge Cutoff and Missing Recent Information," a model is far more likely to produce a plausible-sounding guess than to reliably recognize and flag its own gap.
- **Treating each limitation in this lesson as an unrelated, separate bug to work around individually.** As "Why These Limitations Exist" showed, hallucination, knowledge cutoff, reasoning errors, and bias all trace back to the same underlying mechanism -- understanding that mechanism is more useful than memorizing a list of separate failure modes.

Summary, Cheat Sheet, and Glossary

Summary

- LLMs are genuinely strong at transforming text, drafting and explaining code, answering general-knowledge questions, and following well-specified instructions -- capabilities that come directly from large-scale pretraining.
- **Hallucination** happens because pretraining optimizes for plausible continuations, not verified ones -- fluency does not imply correctness.
- A model's **knowledge cutoff** means it has no information about anything genuinely new after its training data was collected.
- LLM "reasoning" is generated text that resembles step-by-step logic, not a formally verified process -- it can and does make real errors.
- Because models learn from real-world text, they can reproduce the **biases** present in that training data.
- All of these limitations trace back to one shared cause: an LLM predicts plausible text based on learned patterns, with no built-in fact-checker, logic engine, live data connection, or bias correction.

Cheat Sheet

- Strength = language manipulation, drafting, following clear instructions.
- Hallucination = fluent + wrong. Always verify specific facts.
- Knowledge cutoff = no info after pretraining data was collected.
- Reasoning = looks step-by-step, not formally guaranteed -- check it.
- Bias = reflects patterns (including unwanted ones) in training text.
- Root cause of all four: plausible-text prediction, no built-in verifier.

Glossary

- **Hallucination:** when an LLM produces fluent, confident output that is factually incorrect.
- **Knowledge cutoff:** the point after which a model has no information, because its pretraining data was collected before then.

- **Reasoning (in LLMs):** generated text that resembles step-by-step logical reasoning, produced the same way as any other output -- not a formally verified logical process.
- **Bias (in LLMs):** systematic patterns, including stereotypes or imbalances, reproduced in a model's output because they were present in its training data.