AI RESEARCH

Understanding the Impact of Increasing LLM Context Windows

Spencer Torene, Ph.D.
Principal Scientist

Apr 27, 2025

With Meta's recent release of Llama 4 featuring an impressive 10 million token context window, we've reached yet another milestone in the ever-evolving landscape of Large Language Models (LLMs). But what does this massive expansion in context capacity actually mean for users, developers, and businesses leveraging this technology?

The Context Window Revolution

Context windows have grown exponentially since the inception of LLMs. Figure 1 shows the increasing context window size of LLMs. In 2018 and 2019, maximum context windows were 512 and 1,024 tokens, respectively. By 2024 we saw models with 1 million token context windows.

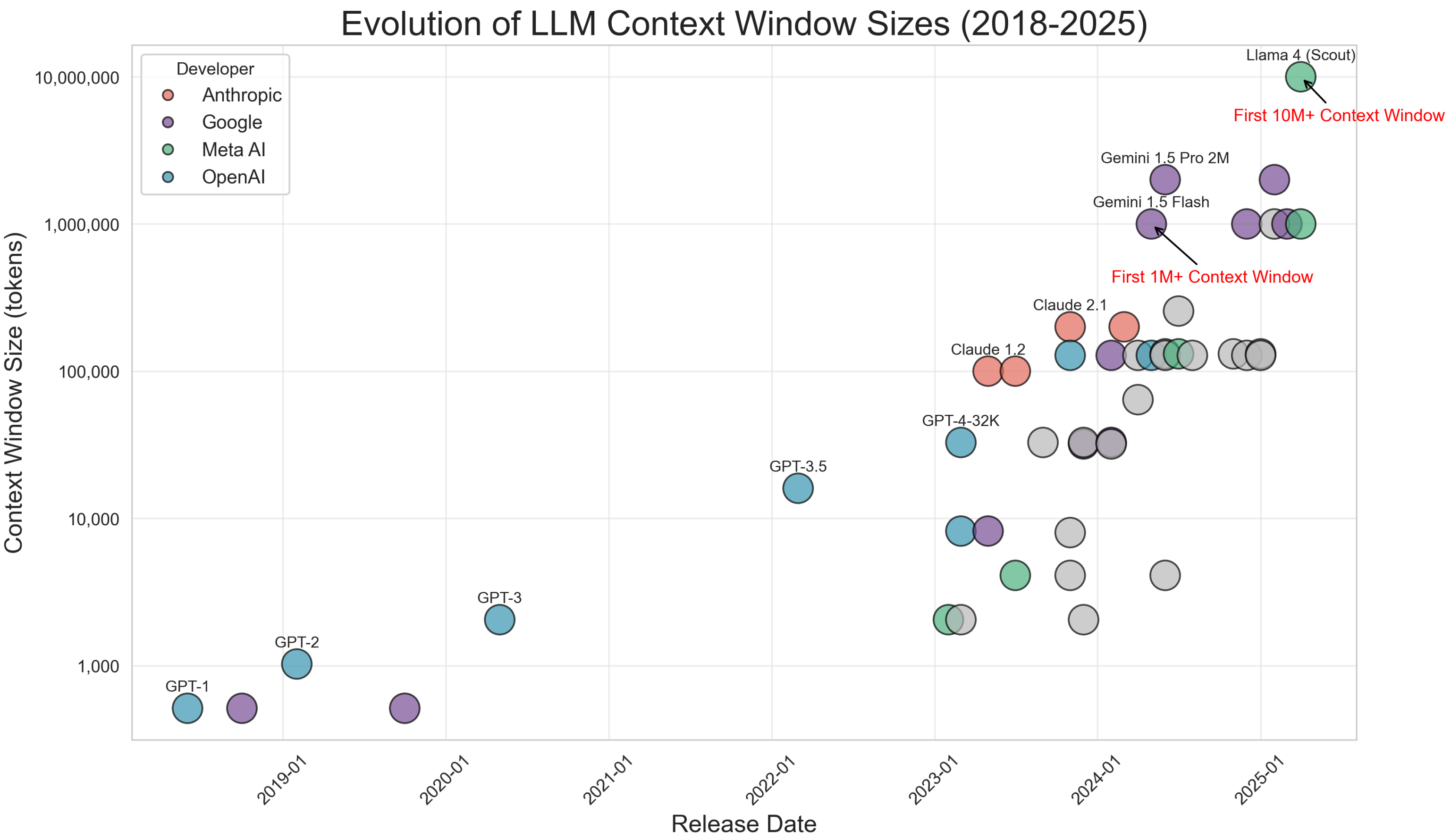

Figure 1: Increasing lengths of context windows over time.

This evolution allows models to keep track of and consider more text at once, and we've seen this working memory of LLMs grow from a few paragraphs to entire books or even libraries worth of content. With Llama 4's recent release featuring a massive 10 million token context window, we're seeing an unprecedented expansion in what language models can keep track of during a single session. But what does this dramatic increase in context window size actually mean for users? The continued development and release of smaller context window models seen from 2023 onward in Figure 1 suggests there's a benefit to the diversity of window size, and we'll discuss the tradeoffs that come with ever-expanding context windows.

What Larger Context Windows Enable

The expansion of context windows brings several meaningful advantages:

Longer Documents: Models can now process and understand entire books, research papers, or technical manuals in a single pass. This enables more comprehensive analysis and deeper understanding of complex texts.

Extended Conversation History: Applications can maintain much longer conversation threads, allowing models to reference information from hours or even days earlier in the same session without forgetting.

Cache Augmented Generation (CAG): CAG pre-computes documents and caches the results as part of a prompt so that those documents' context is available to the LLM when it generates completions. CAG can improve generation latency compared to RAG because there is no extra retrieval step to find relevant documents. Instead, all the documents are available in all prompts — as long as both the cached documents and user prompt fit within the context window. Larger context windows enable more effective use of CAG, where models can reference a substantial cache of information within their context to enhance responses. Moreover, assuming all necessary documents are cached, CAG introduces improved reasoning over those documents. RAG utilizes only the most relevant documents, as determined by document embeddings, which could miss tangentially related, idea-connecting documents.

The Challenges of Long Context Windows

The benefits of increased context come with caveats and disclaimers, however. Bringing much more information to each inference also means that unnecessarily using the entire context window brings several meaningful disadvantages:

Worse Reference Identification: Attention is not uniform across the entire context window — prompts that utilize earlier tokens have better performance than prompts that utilize later tokens. Text extraction and reference identification becomes worse with increasing prompt lengths.

Variable Signal-to-Noise Ratio: There is a trade-off between having all possible useful context in a prompt and focusing on context that matters most. All else equal, longer prompts have less accuracy than shorter prompts.

Increased Costs: Input tokens are typically billed for remote calls, meaning an increase in the amount of context provided directly increases query costs. Prompt caching can reduce input token cost rates, but consistently using unnecessarily long prompts could outweigh any per token cost savings. Revisiting the trends observed in Figure 1, models with smaller context windows continue to be trained and optimized in part because they have fewer parameters, demand less memory and computational resources, and are therefore cheaper to train and host.

Output Token Latency: An underappreciated fact of long prompts is increased output generation latency. Our (and others') research demonstrates that using more input tokens generally leads to slower output token generation. This performance hit creates a practical ceiling on how much you should stuff into your context window without reasonable justification. Figure 2 shows the increase in time it takes per output token, given the number of input tokens.

Figure 2: Output token generation speed decreases with more input tokens.

Best Practices for Large Context Windows

Having the option of long context windows is critical, but it may not make sense to use the entire window by default. Though there can be increased accuracy and reasoning when providing more prompt text, generation can be slower and more expensive, and text extraction will be less accurate.

To make the most of expanded context capabilities while avoiding pitfalls:

- **Be selective.** Include only what's necessary for your specific task.
- **Structure intelligently.** Place the most important information earlier in the context window.
- **Monitor performance.** Track generation speed, quality, and cost to find your optimal context size.

- **Take hybrid approaches.** Consider combining CAG for frequently used information with RAG for broader knowledge. RAG's ability to pull from virtually unlimited external knowledge bases still offers advantages that CAG cannot match. Even a 10M token context window is relatively limited, compared to the terabytes of information that RAG can potentially access.

The best approach is often to be selective about what goes into your context window rather than maximizing its use simply because the capacity exists.

Conclusion

The expansion of context windows to millions of tokens represents a significant advancement in LLM capabilities, but it's not a silver bullet. Like any technological advancement, it comes with trade-offs.

The most successful implementations will be those that thoughtfully consider when to leverage large contexts, when to rely on retrieval, and how to balance performance, cost, and quality considerations.

As we continue to explore the frontiers of what's possible with these expanded capabilities, maintaining a critical perspective on practical applications will be essential to maximizing the value of increasingly powerful models.

Contributing Authors

Berk Ekmekci - Research Scientist

Alex Powell - Engineer

Ready to start your AI journey? Contact us to learn how Meibel can help your organization harness the power of AI, regardless of your technical expertise or resource constraints.

REQUEST A DEMO

Get Started with the Explainable AI Platform

Contact us today to learn more about how Meibel can help your business harness the power of Explainable AI.

Name

Email

Phone Nur

Company Name

Tell us why you’re interested in Meibel...

Send Request

Join our newsletter to stay up to date on features and releases.

Platform

Solutions

About

LinkedIn

Enter your email

Subscribe

By subscribing you agree to with our Privacy Policy and provide consent to receive updates from our company.

Careers

Feed

Get Started

Privacy Policy

Terms of Service

Cookies Settings