What is a Context Window? Why does AI easily 'lose momentum' during long conversations?

The context window determines AI's ability to remember. Learn why chatbots easily lose focus during long conversations and how to use AI more effectively.

You've almost certainly experienced this—and not just with AI, but with humans too: When a conversation gets too long and too informative, you start to lose focus. Or when reading a long series of emails, you only look at the last message and miss important details further up.

The common thread in these situations is simple: the human brain can only hold a certain amount of information in its 'area of ??attention' at any given time. And AI, especially large language models (LLMs), operates on a similar principle—only that limitation is called *the context window* .

What is a context window and why is it important?

Many people tend to think that AI has 'unlimited memory'. But that's not actually the case.

An AI model can only process a certain amount of text at a time—that's the context window. It includes all the existing content: the question you just typed, previous messages, and even the document you uploaded.

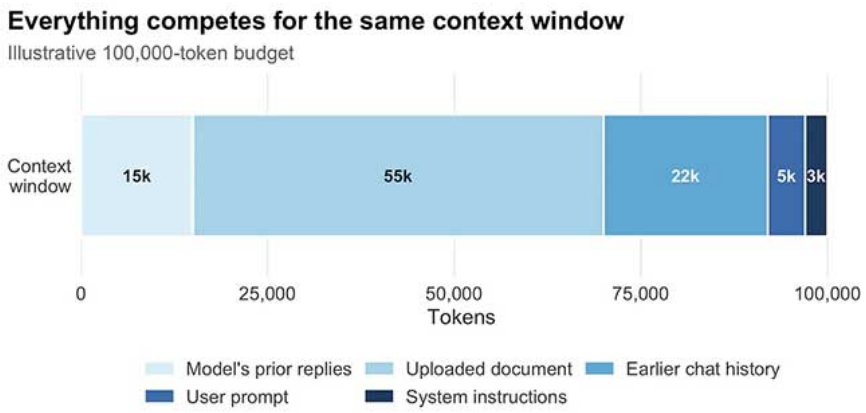
The problem is that **the context window is always limited** . You can think of it like your field of vision while driving. You can clearly see what's ahead, but you can't see the entire route at once like on a map. As you move, what's behind you gradually 'goes out of sight'.

The same applies to AI. As conversations drag on, old information doesn't disappear completely, but it's **no longer within the model's 'active focus'** . And that's when chatbots start to 'go off track'.

To understand this better, we need to mention an important concept: **tokens** .

AI doesn't read text like a human (sentence, paragraph, page), but breaks it down into units called tokens. A single word can be a token, but it can also be broken down into many smaller parts. Even punctuation marks are counted. These tokens are then converted into numbers for the model to process, and a mechanism called *attention* determines which parts of the token chain are important at the time of generating the response.

But there's a crucial limitation: **attention only works within the current context window** . In other words, if the context window is like a whiteboard, then the token is the text on it. When the whiteboard is full, you have to erase or blur the old content to continue writing.



Why does AI 'lose momentum' during long conversations?

This phenomenon is easily observable in practice. You can instruct the AI ??from the start: write in a formal tone, without bullet points, for example, in the healthcare field. But after 20 messages, it suddenly switches to a retail example or changes its tone.

It's not that the AI ??'forgets' in the conventional sense. The problem is that **the initial instructions are no longer prioritized in the current processing area** . When the context window is full, the system is forced to choose:

1. Keep the most up-to-date information.
2. Or keep the parts that are repeated frequently.
3. Or the parts that are easiest to 'catch' semantically.

Information that is distant (regarding time or location in the prompt) will gradually 'fade away'.

Compared to a few years ago, the context window has increased dramatically. Modern models can handle hundreds of thousands to over a million tokens. This allows for working with long documents, large codebases, or lengthy conversations. But the important thing is: **it's still not unlimited memory** .

Furthermore, expanding the context window comes with significant costs. In the Transformer architecture, the number of computations increases exponentially as the number of tokens increases. This makes handling long contexts a complex engineering problem, not simply a matter of 'expanding capacity'.

Context window is not 'memory'

This is a common misconception. When it comes to 'memory,' many people think of a place to store information permanently, like a hard drive or database. But the context window is essentially just **a temporary workspace** .

AI doesn't 'retrieve' information from the entire history like a database. It only works with what's currently within the context window. If the information is outside of that context, the model is almost impossible to use reliably. That's why the same request might be answered very accurately at one point, but 'off-topic' at another.

How to work effectively with context windows

Once we understand this mechanism, many of AI's 'puzzling' behaviors will become easier to explain — and more importantly, easier to fix.

First and foremost, important information needs to be kept 'close' to current requirements. If a rule is crucial, don't just say it once and expect the AI ??to remember it forever. Repeat it when necessary.

Secondly, the placement of information is also crucial. The 'Lost in the Middle' study showed that AI generally performs better when important information is at the beginning or end, and worse when it's in the middle.

Additionally, instead of cramming too much into one prompt, it's better to break it down into smaller parts. This reduces the load on the context window and keeps the model more focused.

Another technique is to summarize in stages. When a conversation is long, a 'slight reset' with a summary will help the AI ??regain a clearer understanding of the context.

Finally, it's important to understand that **more context doesn't necessarily mean better** . Too much information can sometimes just 'noise' the model, rather than making it smarter.

In essence, the core of the context window lies in the fact that AI doesn't suffer from amnesia in the way humans think. It simply **no longer has access to that information within its current workspace** . Understanding this will change how you use AI. Instead of expecting the model to 'remember everything,' you'll learn how to **deliver the right information at the right time** . The most effective way to leverage AI isn't by stuffing it with more data, but by helping it focus on what truly matters.