

Context Windows, Memory Modules and What They Mean for Education

Author: Zsolt Lengyel, iTStudy Hungary Training and Research Centre

Reviewer: Mária Hartyányi, iTStudy Hungary Training and Research Centre

Learning objective

Many surprises in everyday AI use – the model “forgets” an instruction, mixes up two topics, or unexpectedly knows something about us – can be explained by two mechanisms: the context window and the memory functions. The aim of this material is to explain what AI systems can and cannot retain during and between conversations. By the end of the material, teachers will be able to distinguish temporary conversation context from longer-term, memory-like functions, explain how these affect continuity, personalisation and privacy, and use practical routines to keep long work sessions reliable and learners’ data safe.

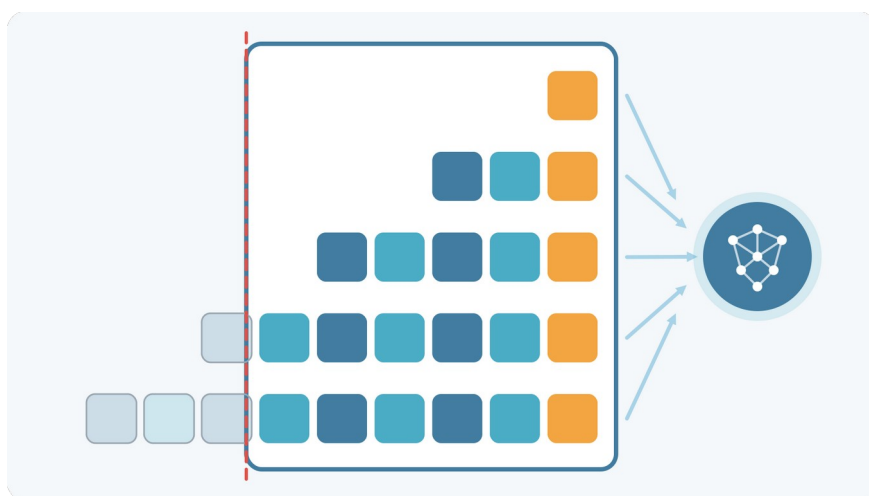
Introduction: the model does not remember like a human

When we chat with a language model, it feels as if it remembered the whole conversation. In reality, the model itself does not change during the conversation. Before every single answer, the system assembles an input – the conversation so far, attached files, instructions running in the background – and the model calculates the next answer from this input alone.

A good analogy is a character from the films *Memento* or *50 First Dates*: someone who cannot retain new memories but keeps their basic knowledge. For a language model, the basic knowledge is what it learned during training; everything else must be placed in front of it again and again.

How a conversation is built: the context

In a chat interface the input grows with every turn. In the first round the model receives only our first message (M1) and answers (A1). In the second round the system sends M1 + A1 + M2, in the third M1 + A1 + M2 + A2 + M3, and so on. Everything the model “sees” at a given moment is called the **context**.



With every turn the input grows; what no longer fits into the fixed-size context window drops out or is compressed

The context is measured in **tokens** – small units of text, often word fragments. The maximum number of tokens a model can handle at once is the **context window**. It is large but finite: at the time of writing, leading models work with windows from a few hundred thousand tokens (e.g. around 400,000 for GPT-5-class models) up to about one million (e.g. Gemini 3.1 Pro), which corresponds to several long books. The limit available in a chat application may be smaller than the model's technical maximum and often depends on the subscription. In many languages other than English, a word takes more tokens on average.

Why does this matter?

- **Limit:** when a conversation becomes too long, the system either cuts off the oldest parts or summarises them.
- **Cost and speed:** the longer the input, the more computation is needed; answers can become slower and use up usage limits faster.
- **Quality:** more context can help, but it also brings noise. Research has shown that models use information at the beginning and end of a long context more reliably than information “lost in the middle”. Irrelevant old material makes misunderstandings more likely.

Compaction: when the conversation is compressed

If a conversation reaches the context limit, many applications automatically **compact** the older messages: they replace them with a summary. The essence remains, the details fade – and so there is always room for new information. We cannot control when this happens, and the downside is that fine details or exact earlier wording may be lost. Very long conversations can also slow down the browser.



Compaction: a long conversation is condensed into a short summary – and some details fall out

Practical routine for long work sessions

- When you change topic, open a new chat. Do the same when a topic has become very long.
- Before switching, ask: “Summarise the decisions, requirements and open questions of this conversation in a form I can paste into a new chat.” Check the summary – it will rarely contain everything.
- Attach long documents instead of pasting them into the message; some applications do this automatically for very long pastes.

Three layers: general knowledge, memory and current context

It is useful to think of what an AI assistant “knows” in three layers:



General knowledge from training, a small editable memory, and the current context all feed into the answer

- **General knowledge** comes from training. It is broad but not up to date: it only extends to the model’s knowledge cut-off date. For recent topics, the model has to use web search.
- **Memory** is a relatively small set of information about the user that the application passes to the model in the background at the start of new conversations. Every new chat starts with an empty context – apart from what the memory module supplies.
- **Current context** is what we place in front of the model right now: the conversation, a file, an example, an error message, the expected format. Models work best from what we show them concretely: one good example is often worth more than a long explanation.

Memory functions in practice (ChatGPT as an example)

Most major assistants (ChatGPT, Claude, Gemini) now offer memory-like functions. In ChatGPT, for example, these currently include:

- **Saved memories** – details we explicitly asked it to remember, or that it saved automatically. They are stored separately from the chats: deleting a chat does not delete a saved memory.
- **Reference chat history** – when enabled, the assistant can draw on past conversations to personalise answers.

- **Memory summary and memory sources** – an overview of what the assistant may use for personalisation, and, below some answers, which earlier chats or files influenced the response. Both can be corrected or deleted.
- **Temporary chat** – a conversation that does not create or update memories.
- **Project-only memory** – in a Project, context can be restricted to the chats and files of that project (see the material on managing complex tasks).

Important

Memory is not a database. The assistant selects what it considers important; it does not retain every detail, the stored information can become outdated or be wrong, and it may be applied in situations where we did not expect it. Review your memory summary regularly.

What this means for education

Continuity

Because short-term context is lost between chats, continuity must be designed. For a course unit, keep a short “context pack”: the aims of the unit, the level of the group (without personal data), the preferred style and format of materials. Paste it at the start of new chats, or store it as instructions and files in a Project. This is more reliable than hoping the memory will remember the details.

Personalisation

Memory can save time: the assistant can remember that you teach, for example, automotive mechatronics to 16-year-olds and prefer tables with answer keys. At the same time, personalisation can narrow the answers or carry over wrong assumptions. When you need a neutral answer, or want to see how the tool behaves for a learner, use a temporary chat.

Privacy and data protection



Memory can be reviewed, switched off and deleted – but the safest data is the data we never enter

- Never enter learners' personal data (names, grades, health or family information, photos) into general-purpose AI tools. Use anonymised or fictional examples.
- Follow the GDPR, your school's data protection rules and, where available, use the school's institutional AI accounts.
- Check your memory settings: memory and chat-history reference can be switched off, and individual memories can be deleted. Providers may keep logs of deleted memories for a limited period (in ChatGPT's case, up to 30 days).
- Be careful when projecting your screen in class: memory-based personalisation or sidebar chat titles may reveal information you did not intend to show.
- Teach learners the same digital hygiene: what an AI remembers about them, how to check it and how to delete it.

Takeaways for teachers

- The model calculates each answer from the current input; it does not remember like a human.
- The context window is finite: long chats become slower, noisier and are eventually compressed.
- New topic – new chat. Carry important decisions over with a checked summary or a context pack.
- Memory is small, selective and editable; review it and switch it off when it is not appropriate.
- Show the model concrete examples and files: it works best from what is in front of it.
- Protect learners' data: anonymise, use institutional accounts, and never rely on memory for sensitive information.

References

- Liu, N. F. et al. (2024): Lost in the Middle: How Language Models Use Long Contexts. Transactions of the Association for Computational Linguistics, 12, 157–173.
- OpenAI Help Center (2026): Memory in ChatGPT; Projects in ChatGPT; ChatGPT release notes.
- Lengyel, Zs. (2026): Introduction to the concepts of artificial intelligence. VITA training presentation.

Links

- Memory in ChatGPT: help.openai.com/en/articles/8590148-memory-faq
- Projects in ChatGPT: help.openai.com/en/articles/10169521-projects-in-chatgpt
- Tokenizer demonstration: tiktokenizer.vercel.app

Copyright © 2026 VITA Consortium

This work may be reused and adapted under the terms of the Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).



Funded by the European Union. Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the National Agency.