Large Language Models (LLMs)

Agenda

- Available LLMs
- How to find the best model for the task
- Interacting with an LLM
  - Tokenization
  - Training
- Improving the performance of an LLM
  - Thinking models
  - Internet search
  - Deep research
- Special features

Some currently available LLMs

ChatGPT by OpenAI (Microsoft being its largest investor and using ChatGPT in its own product such as copilot)

Claude by Anthropic

Gemini by Google DeepMind

Llama (Large Language Model Meta AI) by Meta

DeepSeek by DeepSeek

Grok by xAI

LeChat by Mistral

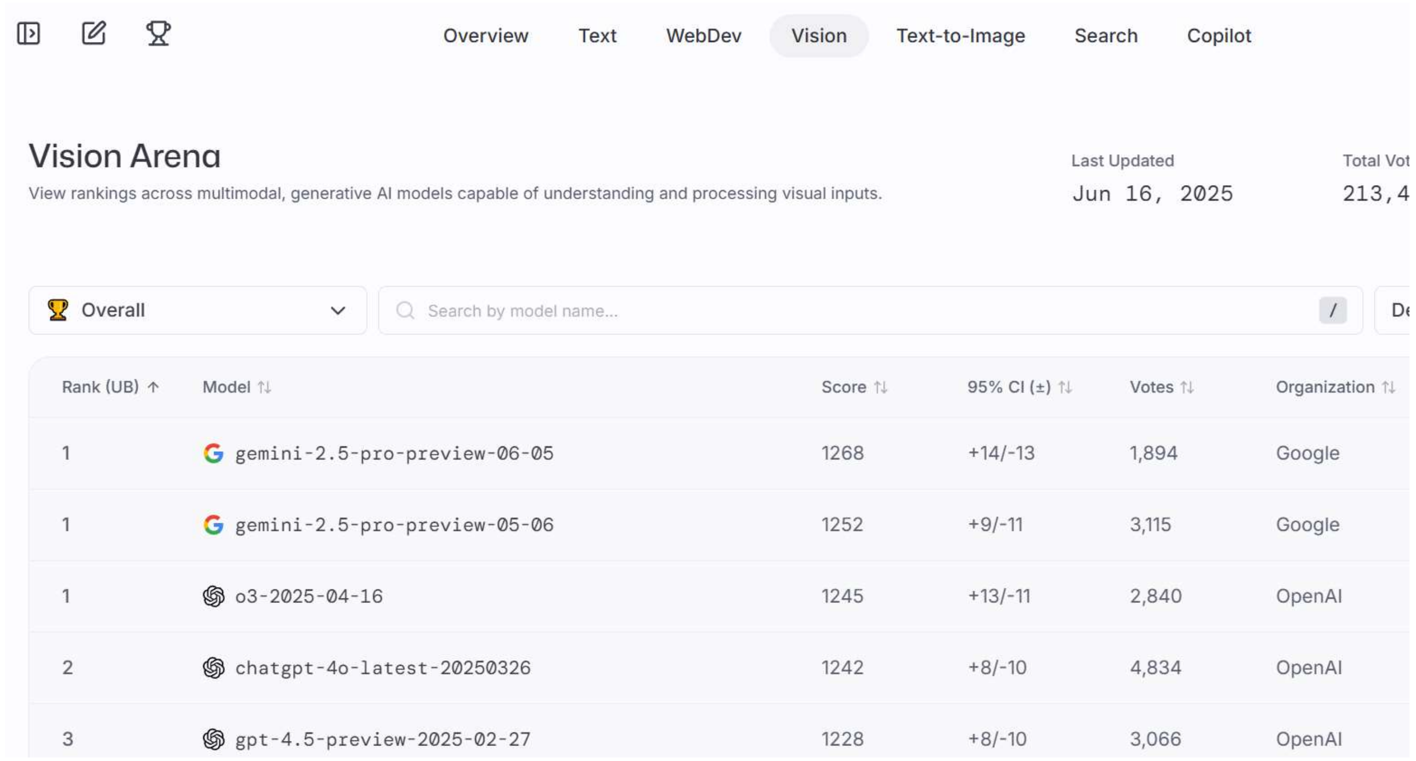
Perplexity by Perplexity AI

…

Demo

Finding the best model for your task

https://lmarena.ai/leaderboard/



Vision Arena

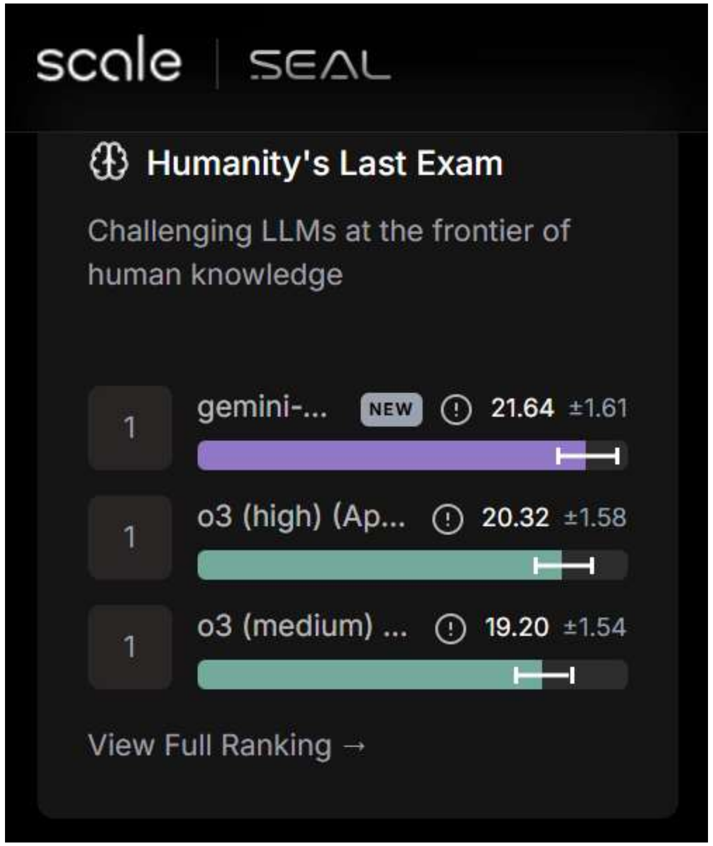
View rankings across multimodal, generative AI models capable of understanding and processing visual inputs.

Last Updated Jun 16, 2025

Total Vot 213,4

Rank (UB)	Model	Score	95% CI (±)	Votes	Organization
1	gemini-2.5-pro-preview-06-05	1268	+14/-13	1,894	Google
1	gemini-2.5-pro-preview-05-06	1252	+9/-11	3,115	Google
1	o3-2025-04-16	1245	+13/-11	2,840	OpenAI
2	chatgpt-4o-latest-20250326	1242	+8/-10	4,834	OpenAI
3	gpt-4.5-preview-2025-02-27	1228	+8/-10	3,066	OpenAI

https://scale.com/leaderboard



Humanity's Last Exam

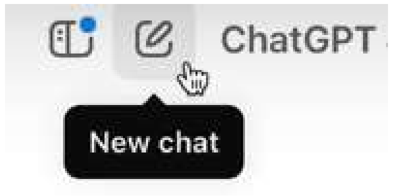
Challenging LLMs at the frontier of human knowledge

1	gemini-... NEW	21.64	±1.61
1	o3 (high) (Ap...	20.32	±1.58
1	o3 (medium) ...	19.20	±1.54

Interacting with an LLM

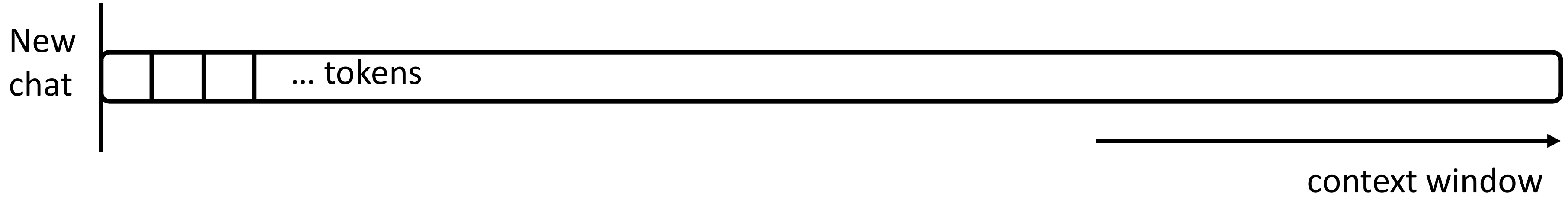
Together with GPT we build a so-called **context window**.

Recommendation: Press „New chat" whenever you start a new conversation or if you switch topic. This wipes out all previous tokens and makes sure the LLM is not overloaded with irrelevant information.

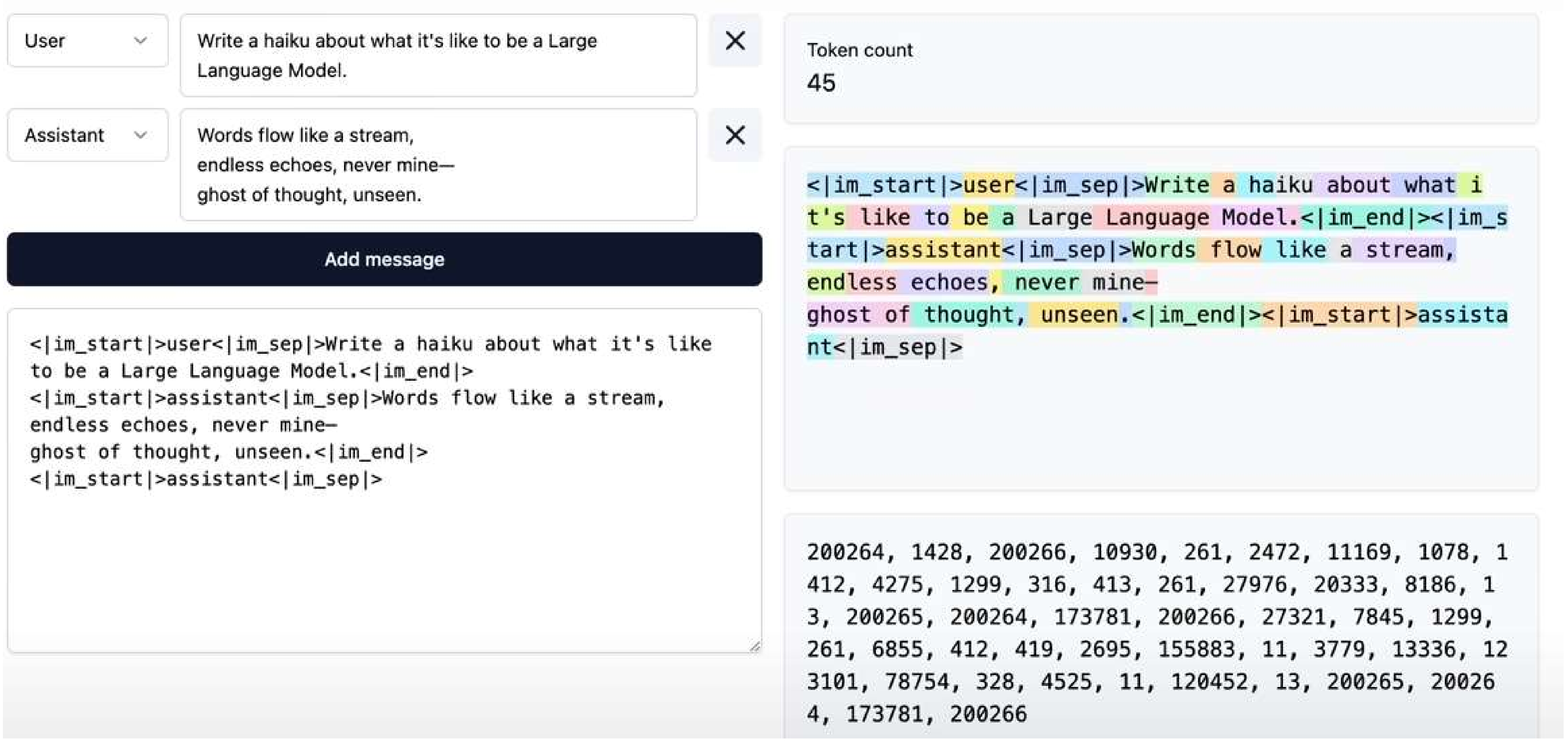


Type a user query/prompt which then becomes chopped up into a so-called **token** sequence.

The LLM responds by generating a 1D sequence of tokens.



Tokenization example for ChatGPT



Write a haiku about what it's like to be a Large Language Model.

Words flow like a stream,
endless echoes, never mine—
ghost of thought, unseen.

```
<|im_start|>user<|im_sep|>Write a haiku about what it's like
to be a Large Language Model.<|im_end|>
<|im_start|>assistant<|im_sep|>Words flow like a stream,
endless echoes, never mine—
ghost of thought, unseen.<|im_end|>
<|im_start|>assistant<|im_sep|>
```

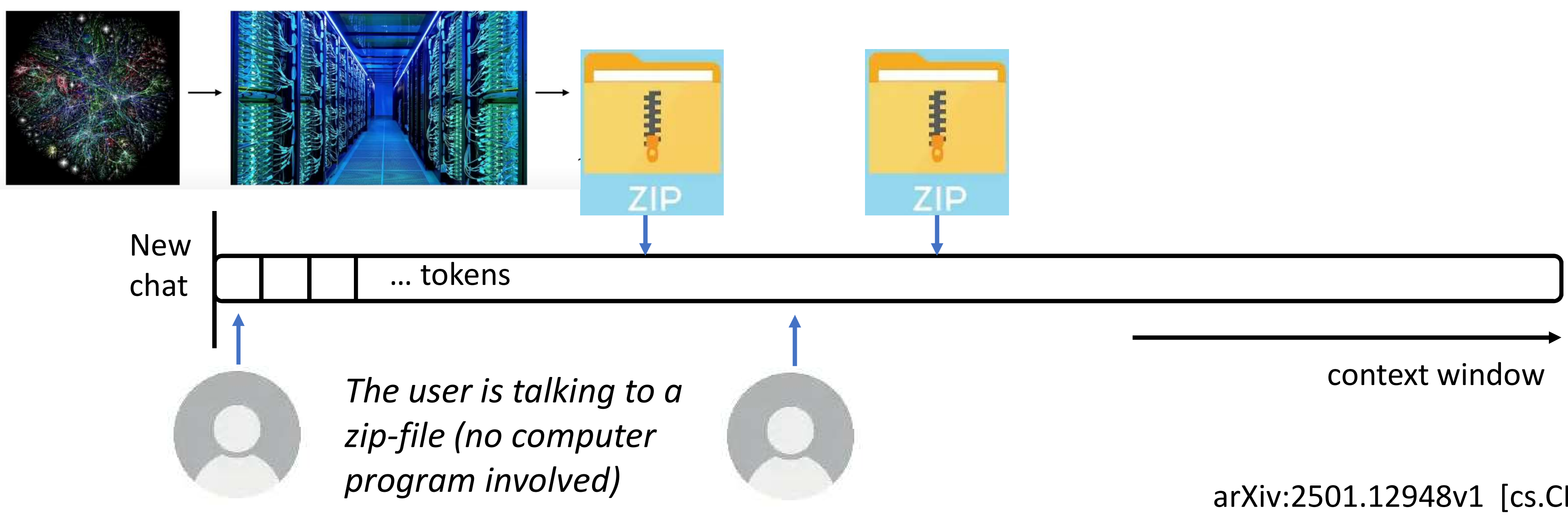
Token count
45

```
<|im_start|>user<|im_sep|>Write a haiku about what i
t's like to be a Large Language Model.<|im_end|><|im_s
tart|>assistant<|im_sep|>Words flow like a stream,
endless echoes, never mine—
ghost of thought, unseen.<|im_end|><|im_start|>assista
nt<|im_sep|>
```

```
200264, 1428, 200266, 10930, 261, 2472, 11169, 1078, 1
412, 4275, 1299, 316, 413, 261, 27976, 20333, 8186, 1
3, 200265, 200264, 173781, 200266, 27321, 7845, 1299,
261, 6855, 412, 419, 2695, 155883, 11, 3779, 13336, 12
3101, 78754, 328, 4525, 11, 120452, 13, 200265, 20026
4, 173781, 200266
```

Training an LLM

An LLM consists of a several TB large lossy, probabilistic zip file of the internet, it stores over trillions of parameters of the neural network that has been trained with data from the entire accessible internet.


ZIP
ZIP
New chat
… tokens
context window
The user is talking to a zip-file (no computer program involved)

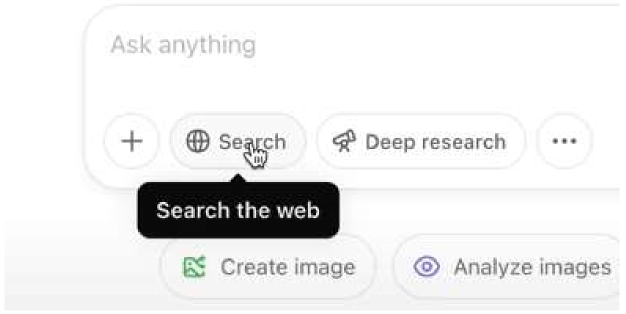
Training an LLM

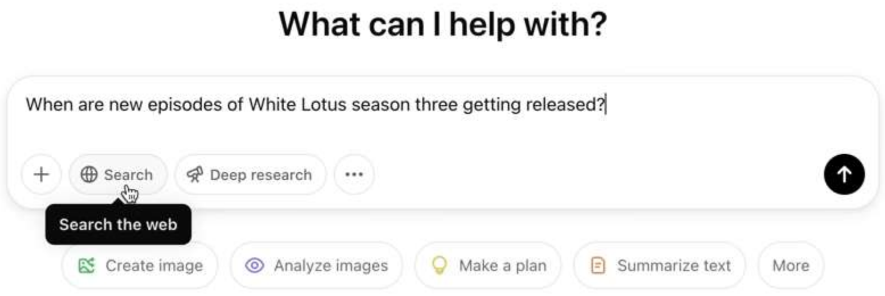
***Pre-Training* an LLM:**
- is very costly (millions for ChatGPT3, tens of millions for ChatGPT4, hundreds of millions for ChatGPT5)
- takes several thousands of GPUs
- takes several months of time which is why it is not done that often
- has a knowledge cut-off date usually several months ago

***Post-training*:**
- gives the LLM its “personality“
- is much cheaper and done more often than pre-training
- finetunes on conversations through:
    - SFT: supervised fine tuning (training the model on examples with correct outputs)
    - RLHF: reinforcement learning from human feedback (humans rate model outputs and the model learns to prefer responses that align with human preferences shaping behavior and tone)
    - RL: reinforcement learning (the model learns by receiving rewards or penalties based on its actions)

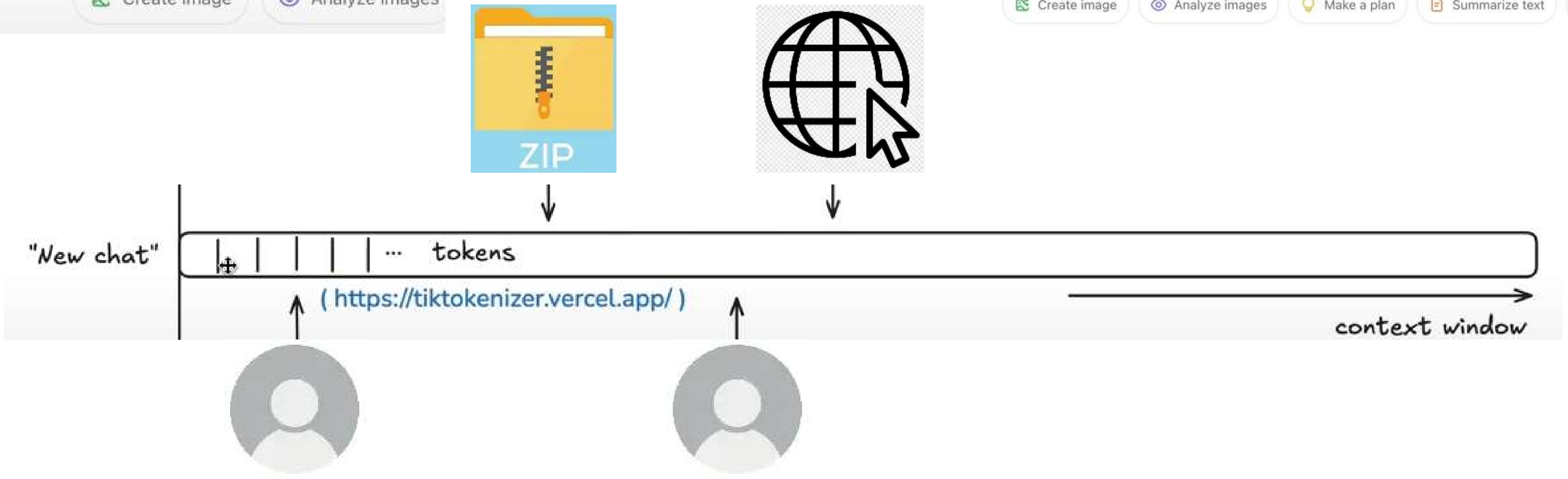
Improving LLM performance by incorporation of Internet Search

Searches the internet real time (so you are up to date) and gives you search results and corresponding links, e.g. perplexity or when selecting the "Search", e.g. Why did Google stock go down today? …



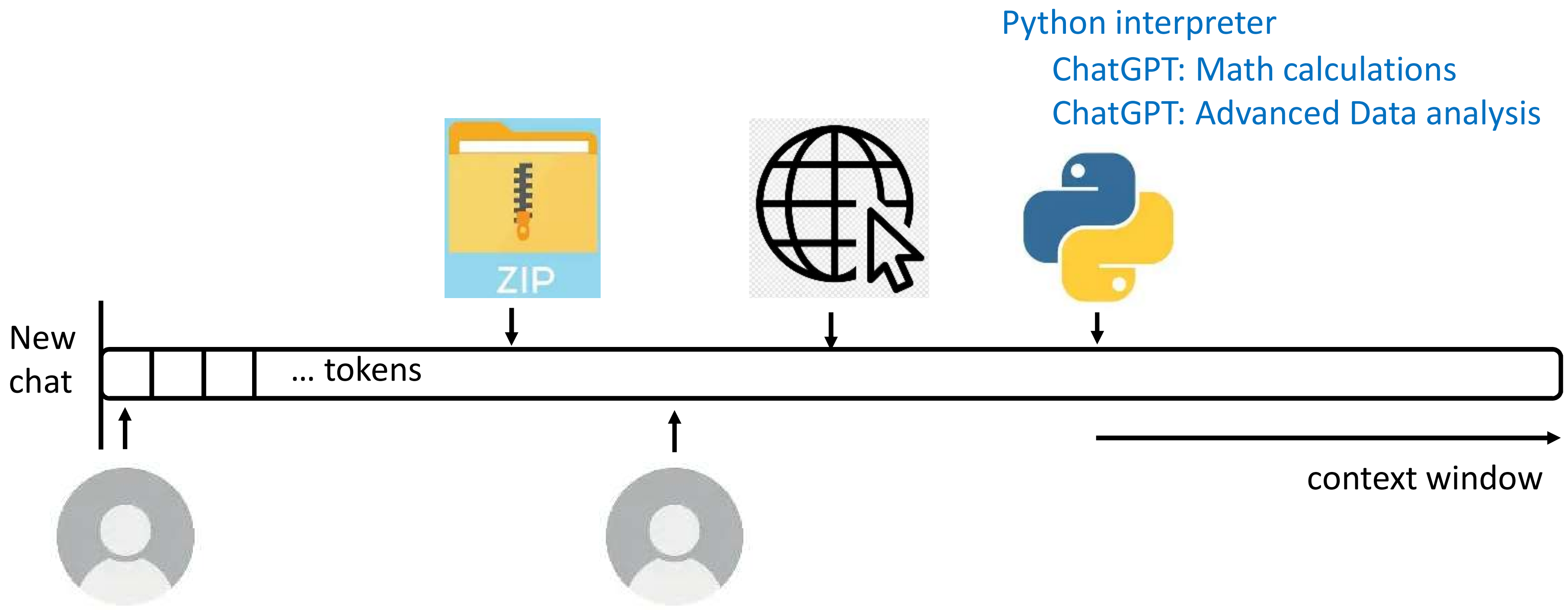

What can I help with?

When are new episodes of White Lotus season three getting released?



Improving LLM performance by incorporation of a Python interpreter

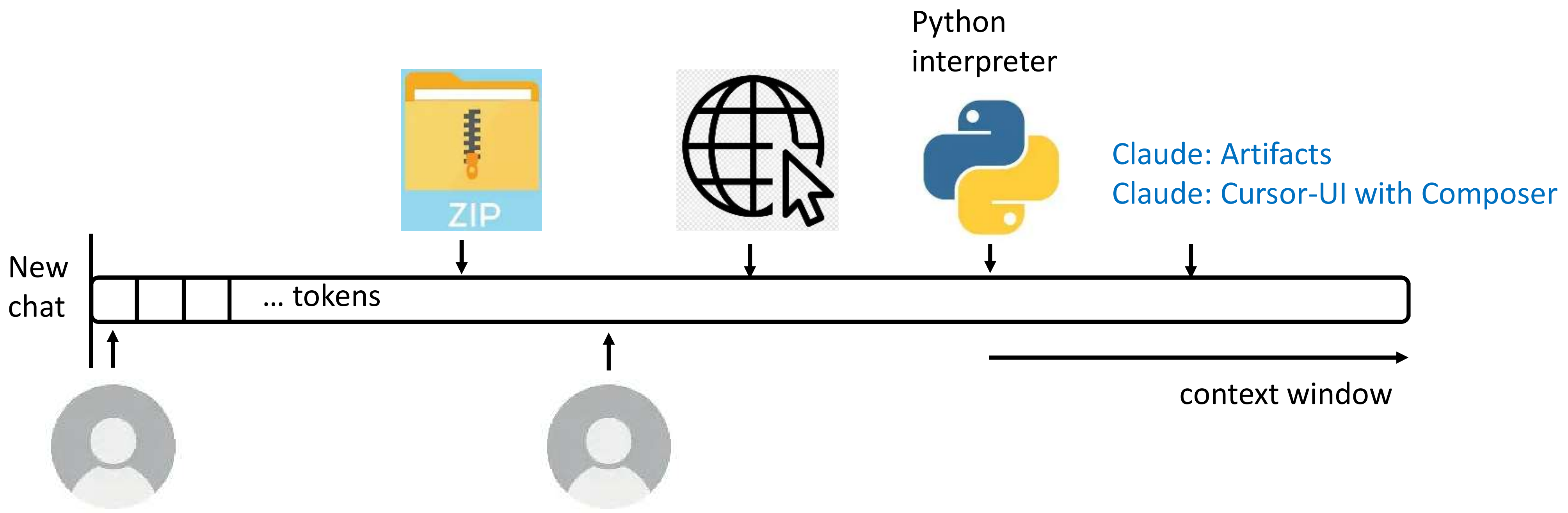
Easy math problems are solved through the zip-file. More difficult ones are solved through the calculator of a python interpreter (ChatGPT does it, but others might not - check results using a real calculator)



Creating customized outputs

Claude: Artifacts

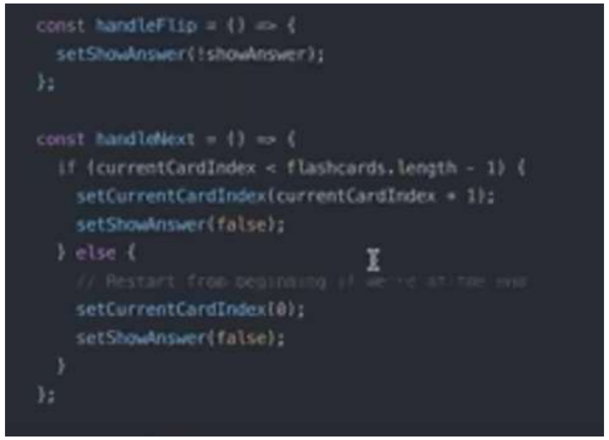
- Generates conceptual diagrams of book chapters
- Create flashcards to a given text



Example: Creating customized outputs

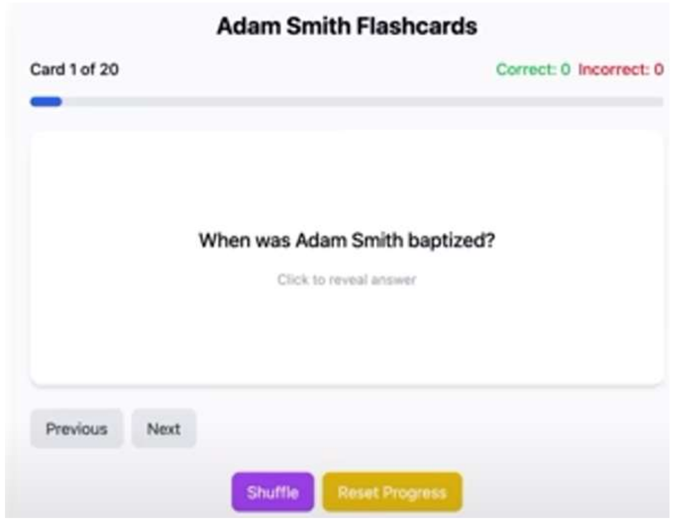
Prompt: Generate 20 flashcards from the following text.
Prompt: Write a flashcard app to test me on these.

Artifacts writes an App just for you:



```
const handleFlip = () => {
  setShowAnswer(!showAnswer);
};

const handleNext = () => {
  if (currentCardIndex < flashcards.length - 1) {
    setCurrentCardIndex(currentCardIndex + 1);
    setShowAnswer(false);
  } else {
    // Restart from beginning if we're at the end
    setCurrentCardIndex(0);
    setShowAnswer(false);
  }
};
```

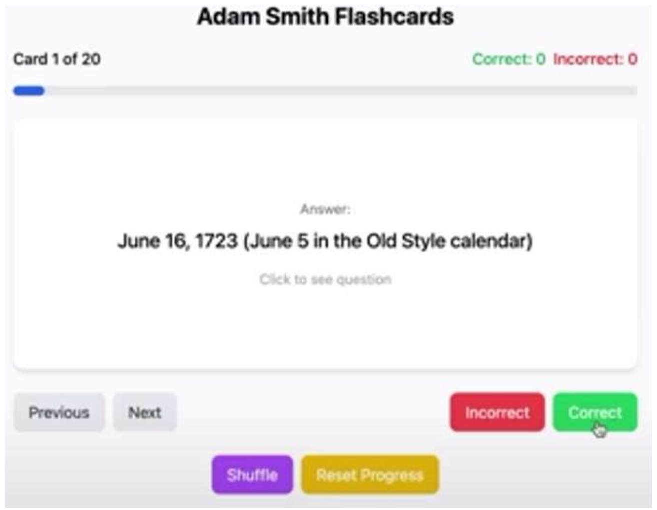


Adam Smith Flashcards

Card 1 of 20 Correct: 0 Incorrect: 0

When was Adam Smith baptized?

Click to reveal answer



Adam Smith Flashcards

Card 1 of 20 Correct: 0 Incorrect: 0

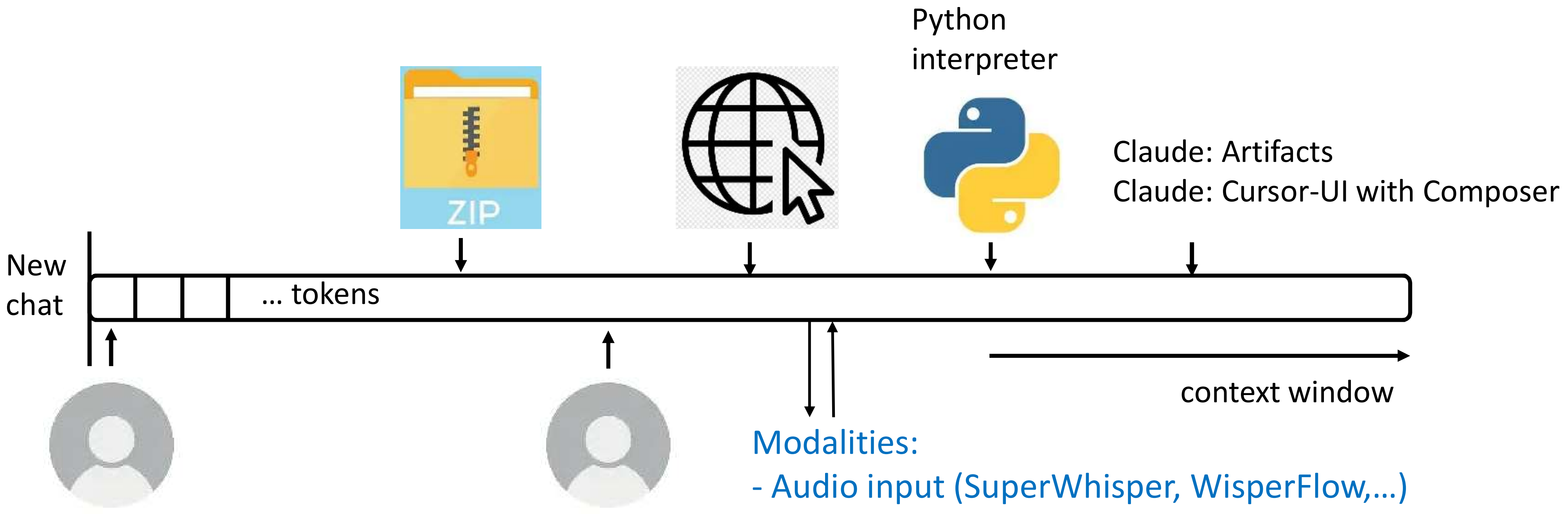
Answer:

June 16, 1723 (June 5 in the Old Style calendar)

Click to see question

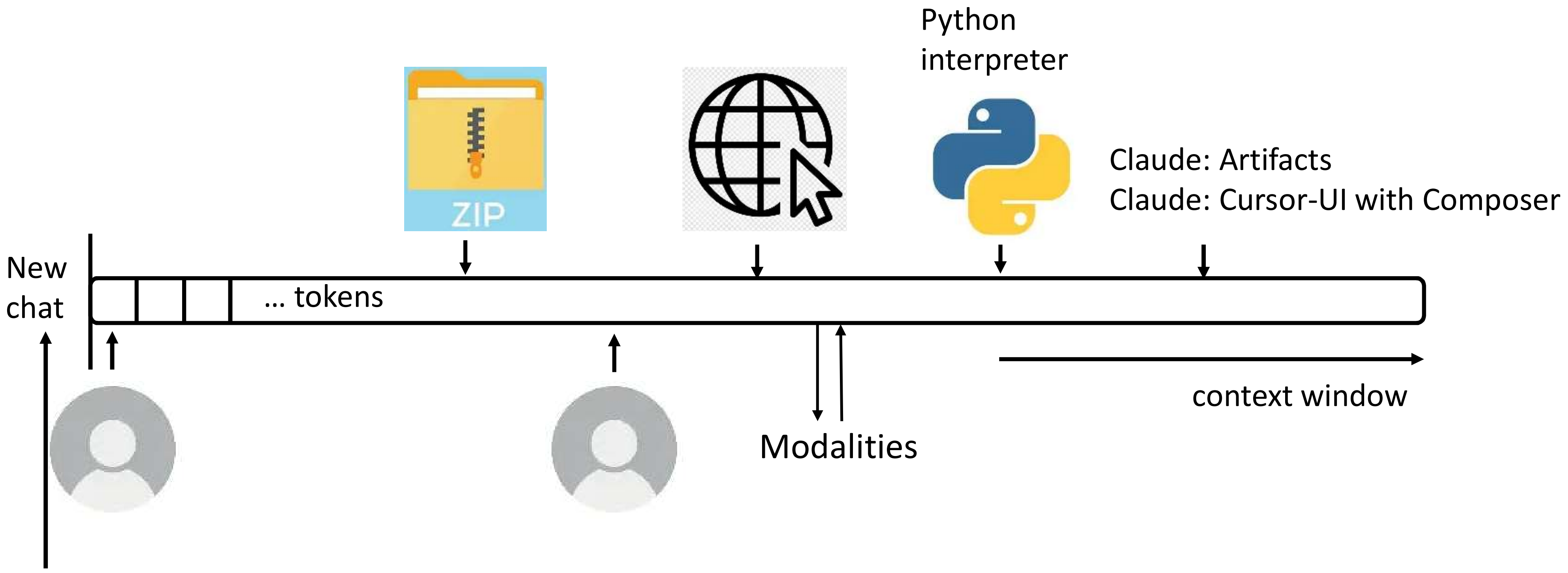
Find further showcases here: https://artifactsclaude.com/

Giving feedback through text, audio, video, and images



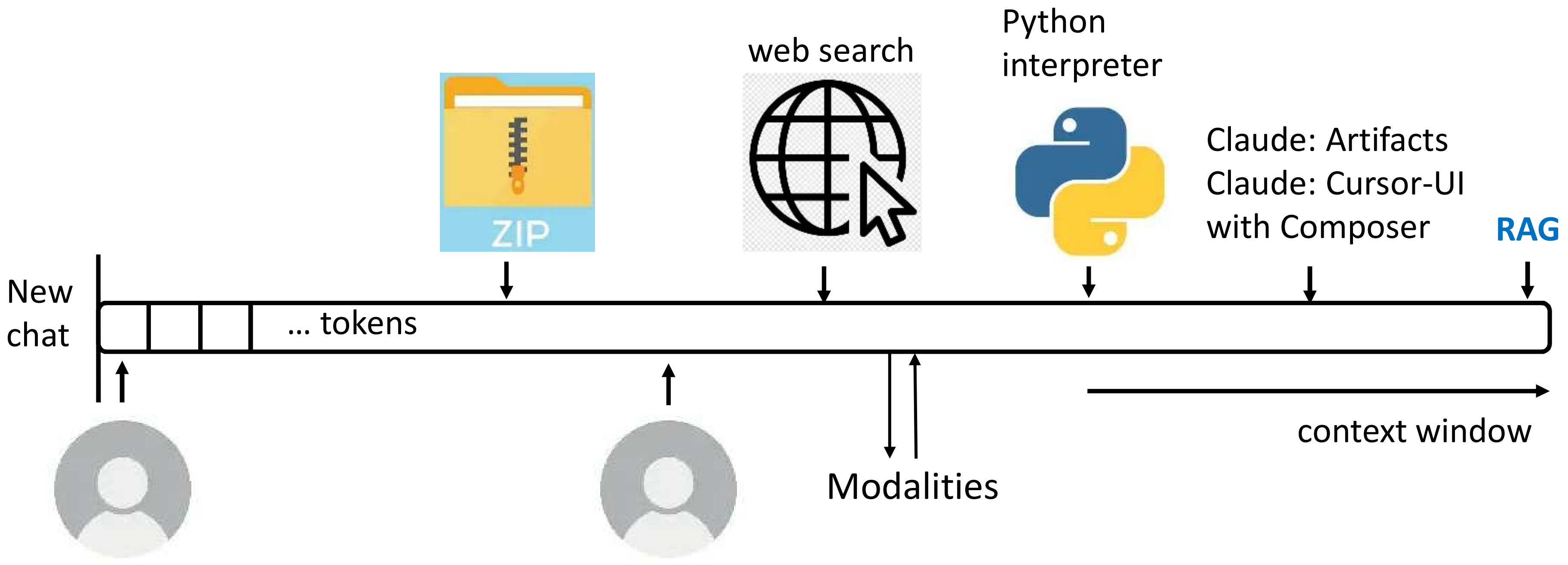
Modalities:
- Audio input (SuperWhisper, WisperFlow,…)
- Audio output (usually within app)
- Advanced voice mode: use true audio, i.e. no speech to text conversion takes place – it uses speech as is
- Podcast generation
- Image input and output
- Video input and output (e.g. Sora, …)

The memory feature



Before a token gets started, ChatGPT looks at things which it stored about the user
GPT updates its memory about a user automatically but one can also invoke it, e.g. write “Can you please remember this“

Retrieval-Augmented Generation (RAG)



Retrieval-Augmented Generation (RAG)

What is RAG?
- It combines information retrieval with large language model (LLM) generation
- Allows models to use up-to-date, external knowledge without retraining

How it Works:
1. User Query → A question or prompt is asked
2. Retriever → Searches a database or knowledge source
3. Retrieved Context → Relevant passages are selected
4. LLM Generator → Uses both the query + retrieved context to generate an answer

What it is used for:
- Question answering → Accurate, fact-based responses
- Customer support → Pulling from manuals or FAQs
- Research assistants → Summarizing documents, papers, or reports
- Enterprise AI → Searching private databases securely

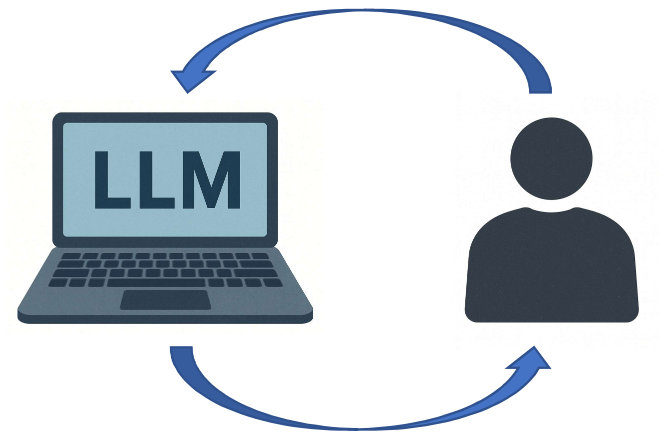
Key Advantage: Reduces “hallucinations” by grounding LLMs in real data.

Recommendations on how to efficiently use ChatGPT

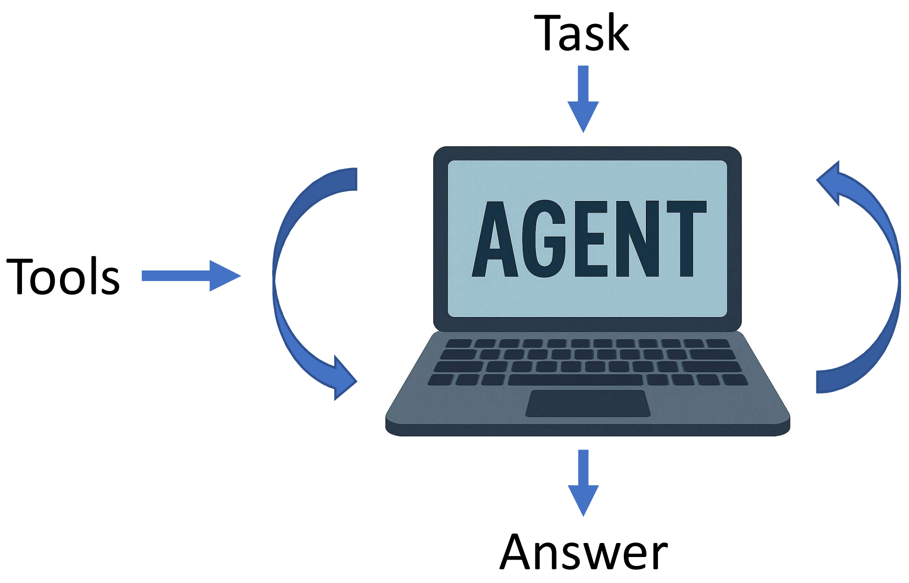
- **Break down complex prompts**: use several shorter prompts that iterate on the desired output
- **Use role play**: tell GPT e.g. that you are an instructor teaching math to elementary school kids
- Use natural language and add **descriptions how the answer should look like**
- **Clear memory**
- **Ask ChatGPT to improve your prompts** and give you hints how to prompt in general
- Ask GPT to respond **step-by-step**
- **Do not provide personal information**
- Use GPT on your phone to quickly snap pictures and use audio input
- Organize your chats (see left side of ChatGPT interface)

Agents

- So far, we have performed regular prompting: a user interacts with an LLM to improve its response – this is an iterative process with the user being the bottle neck

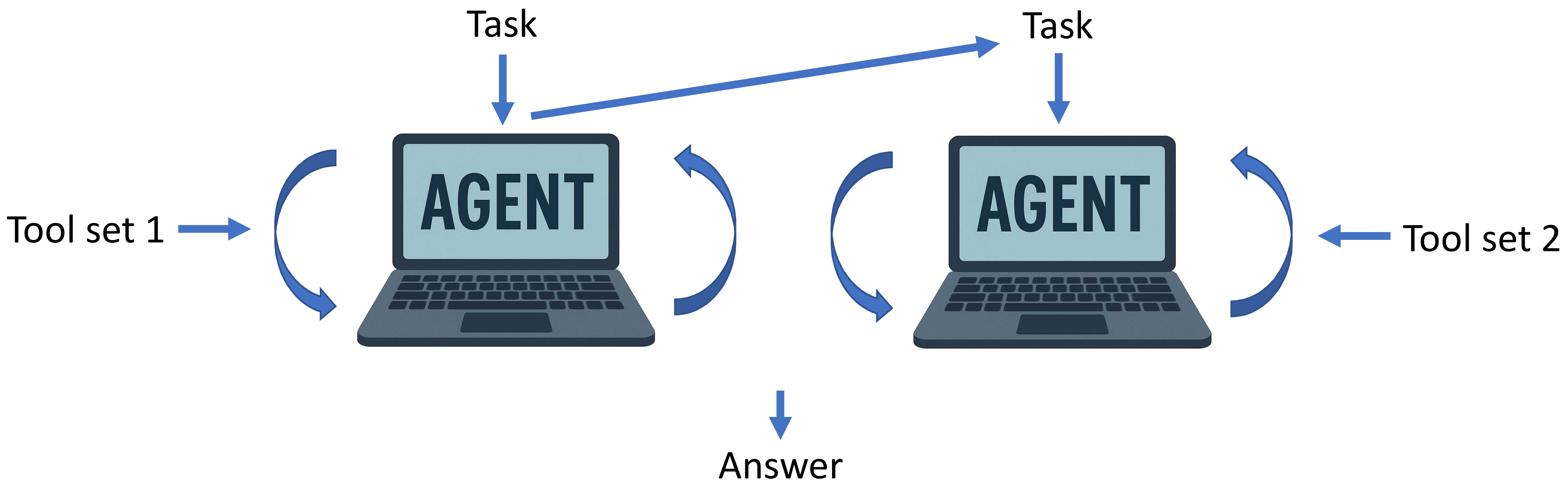


- Now, we use a so called agent: an agent provides feedback to the LLM autonomously to improve on the LLM‘s output
- Tools: we assign a certain set of tools to solve the problem, such as scrape websearch, file read, PDF RAG search, Youtube Video RAG search, code docs RAG search, csv RAG search …



Multi agents

In order to better focus, it is useful to have several agents so that each one is responsible for a certain task – this is known as Mulit-agentic AI



How to best use agents?

- Role playing: assign a certain role to each agent
- Focus: use several agents so that each one is responsible for a certain task
- Tools: assign a certain set of tools to each agent – not too many tools per agent as they could get overloaded
- Cooperate: have agents cooperate with each other
- Guardrails: set boundaries for your agents to receive reliable and consistent results and avoid hallucinations
- Memory: specify how long you want the agents to remember things
  - Short term memory: used during the execution of a task and to share knowledge, activities, and learning with other agents
  - Long term memory: memory is stored after execution of current tasks and can be used in any future tasks leading to self-improving agents

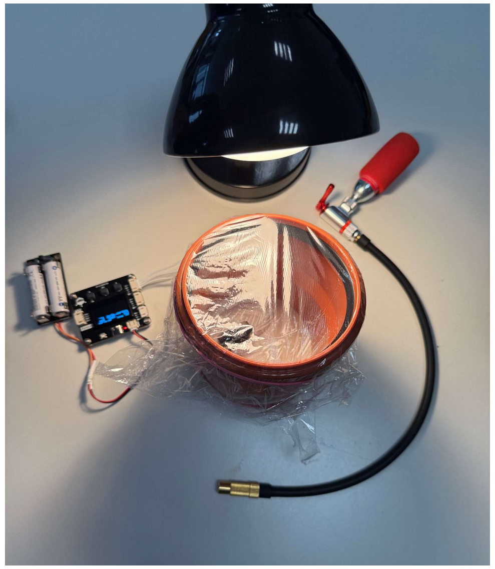
Example use case:
Plotting, regression, and Python code generation with ChatGPT

Modeling the greenhouse effect

Goal: Build a model to demonstrate how the temperature inside a greenhouse differs between normal air and air with elevated CO_2 concentration.
Experimental setup: Cover a drain pipe with some transparent foil and use it to build a mini greenhouse. Measure the temperature within the greenhouse with a temperature sensor while it is heated with a lamp. Perform the same experiment after increasing the CO_2 concentration.
Measurements: see table on the right side

Data analysis:
Generate a plot with time on the vertical and temperature on the horizontal axis for the two measurements. Perform a regression with the following formula and extrapolate the temperature after 100 minutes.

$$T(t) = T_{\text{outside}} + c_1\left(1 - e^{-c_2 t}\right)$$

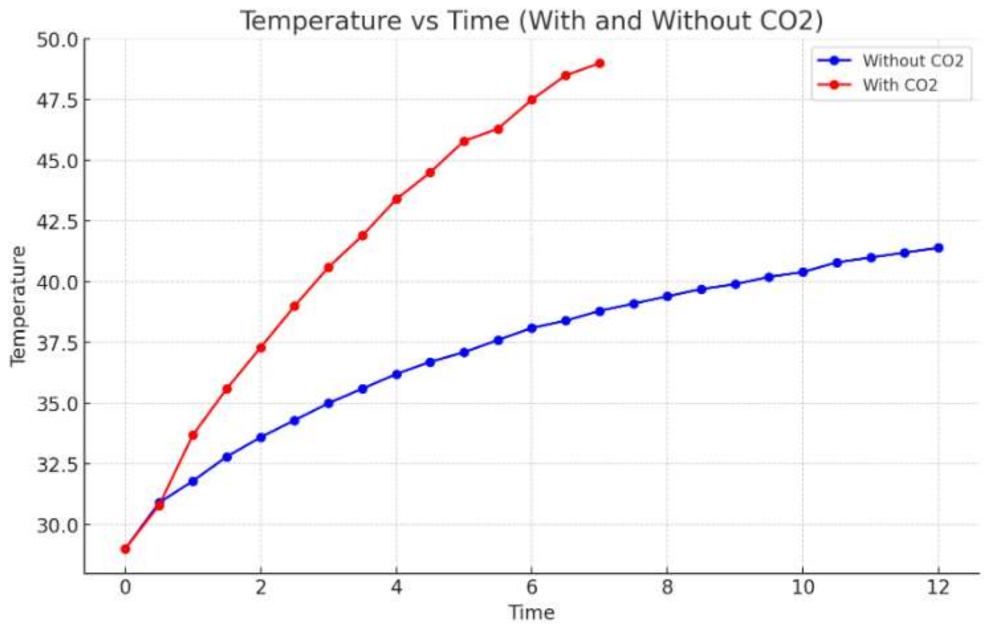


Time min	Temperature °C	
time	w/o CO2	w CO2
0	29	29
0.5	30.9	30.8
1	31.8	33.7
1.5	32.8	35.6
2	33.6	37.3
2.5	34.3	39
3	35	40.6
3.5	35.6	41.9
4	36.2	43.4
4.5	36.7	44.5
5	37.1	45.8
5.5	37.6	46.3
6	38.1	47.5
6.5	38.4	48.5
7	38.8	49
7.5	39.1	
8	39.4	
8.5	39.7	
9	39.9	
9.5	40.2	
10	40.4	
10.5	40.8	
11	41	
11.5	41.2	
12	41.4	

Data analysis with ChatGPT

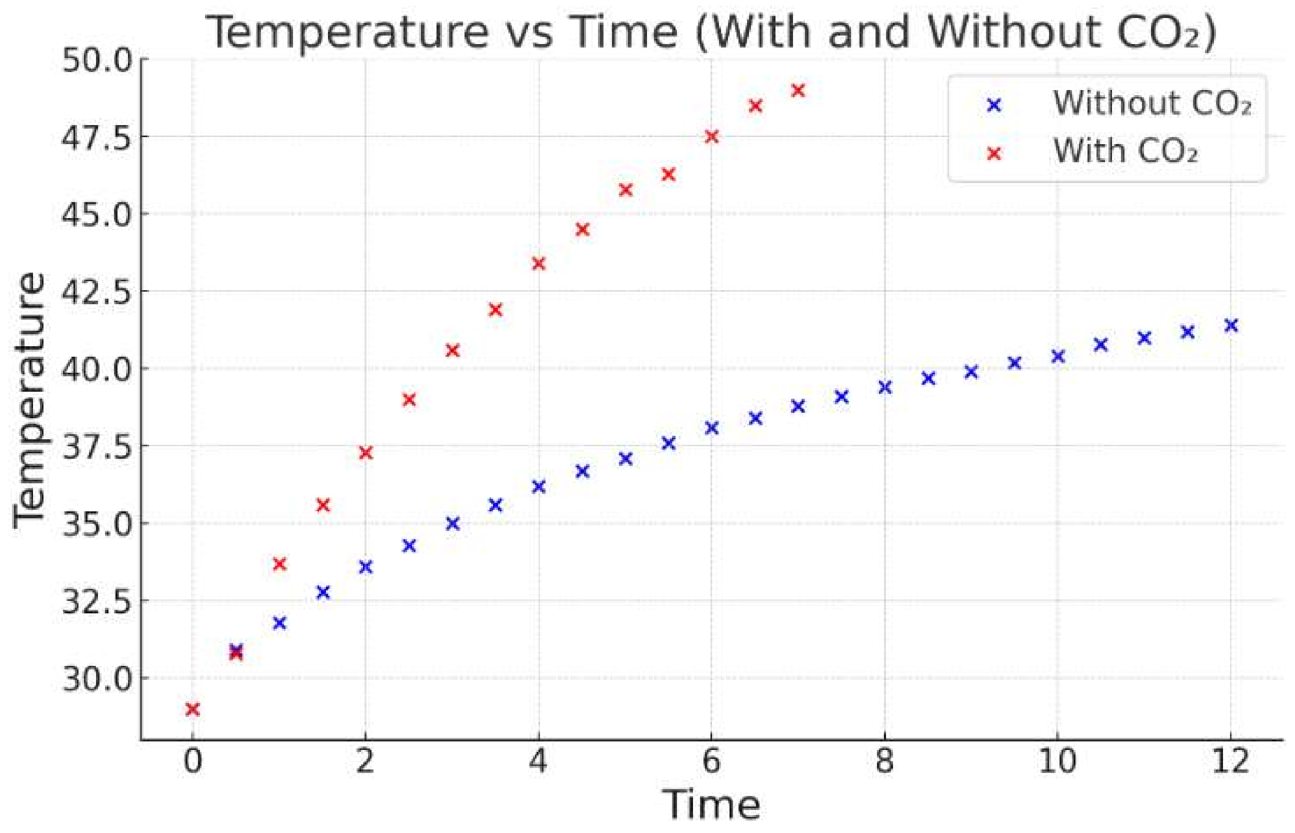
Prompt 1: Use the following values to generate a plot with the temperature on the vertical and the time on the horizontal axis. In the data table you will find three columns. The first one is the time. The second one reflects the temperature without CO2 and the third one the temperature with CO2. Plot the data points that refer to the second column in blue and those of the third column in red.

time	w/o CO2	w CO2
0	29	29
0.5	30.9	30.8
1	31.8	33.7
1.5	32.8	35.6
2	33.6	37.3
2.5	34.3	39
3	35	40.6
3.5	35.6	41.9
4	36.2	43.4
4.5	36.7	44.5
5	37.1	45.8
5.5	37.6	46.3
6	38.1	47.5
6.5	38.4	48.5
7	38.8	49
7.5	39.1	
8	39.4	
8.5	39.7	
9	39.9	
9.5	40.2	
10	40.4	
10.5	40.8	
11	41	
11.5	41.2	
12	41.4	



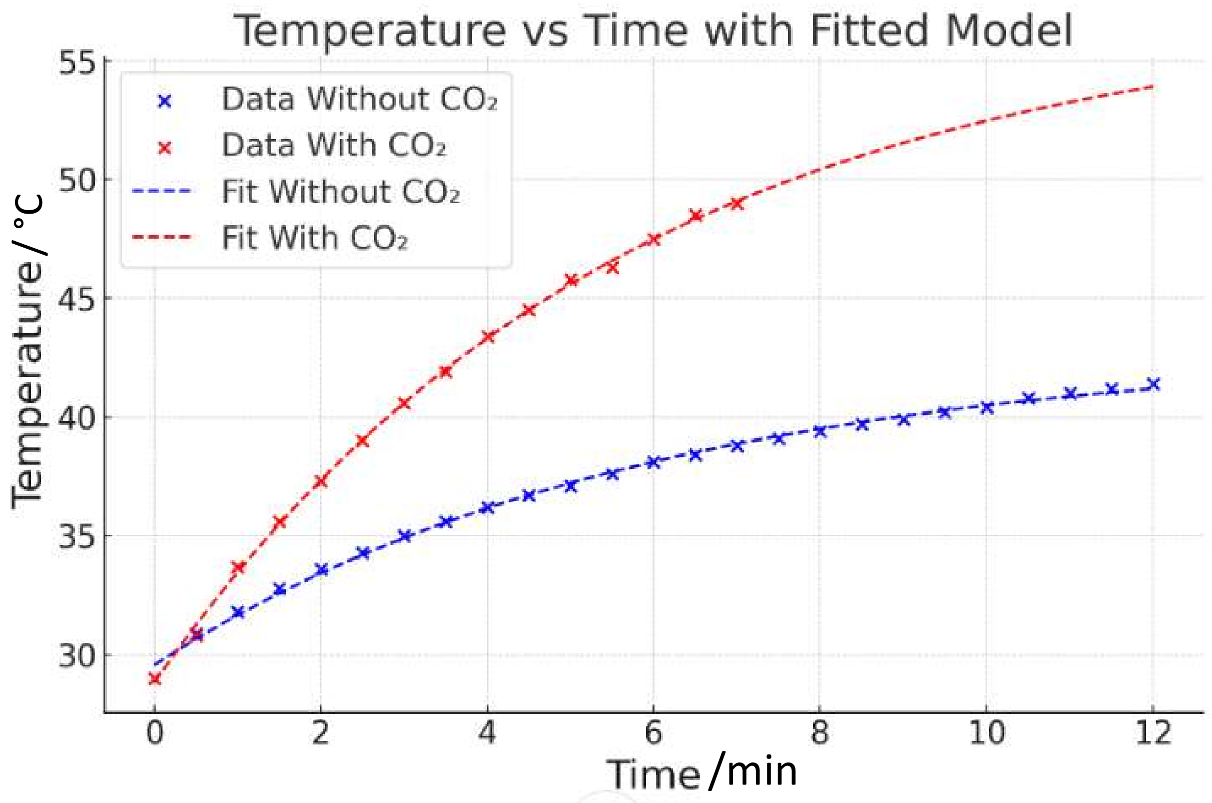
Data analysis with ChatGPT

Prompt 2: write any number and letter in the diagram twice as large. Do not connect the data points by a line. Write the 2 in CO2 always as a subscript.



Data analysis with ChatGPT

Prompt 3: use the formula T = Toutside + c1(1-exp(-c2*t)) where c1, c2 and Toutside are fitting parameters. Please fit the data above



$$T(t) = T_{\text{outside}} + c_1\left(1 - e^{-c_2 t}\right)$$

- Without CO_2
  - $T_{\text{outside}} \approx 29.57$
  - $c_1 \approx 13.36$
  - $c_2 \approx 0.170$
- With CO_2
  - $T_{\text{outside}} \approx 28.82$
  - $c_1 \approx 28.45$
  - $c_2 \approx 0.178$

The temperature asymptotically approaches:

- 43 °C without CO_2 and
- 57 °C with CO_2