Overview of Large Language Models (LLMs) and Retrieval Augmented Generation (RAG)

James Verbus
linkedin.com/in/jamesverbus

January 15th, 2025

Thank you

- **Chongwen Lu** for testing and contributing to the notebooks, and for helpful feedback

- **Ariel Green** for coordinating the workshop

- **Richard Gaitskell** and **Ian Dell'Antonio** for organizing the workshop and for the invitation to contribute

2009-2016

2017-now

What will you learn in this workshop?

1. How to use LLMs directly in a Google Collab notebook
    a. OpenAI models via API
    b. Meta's open-source LLaMa running locally on a GPU

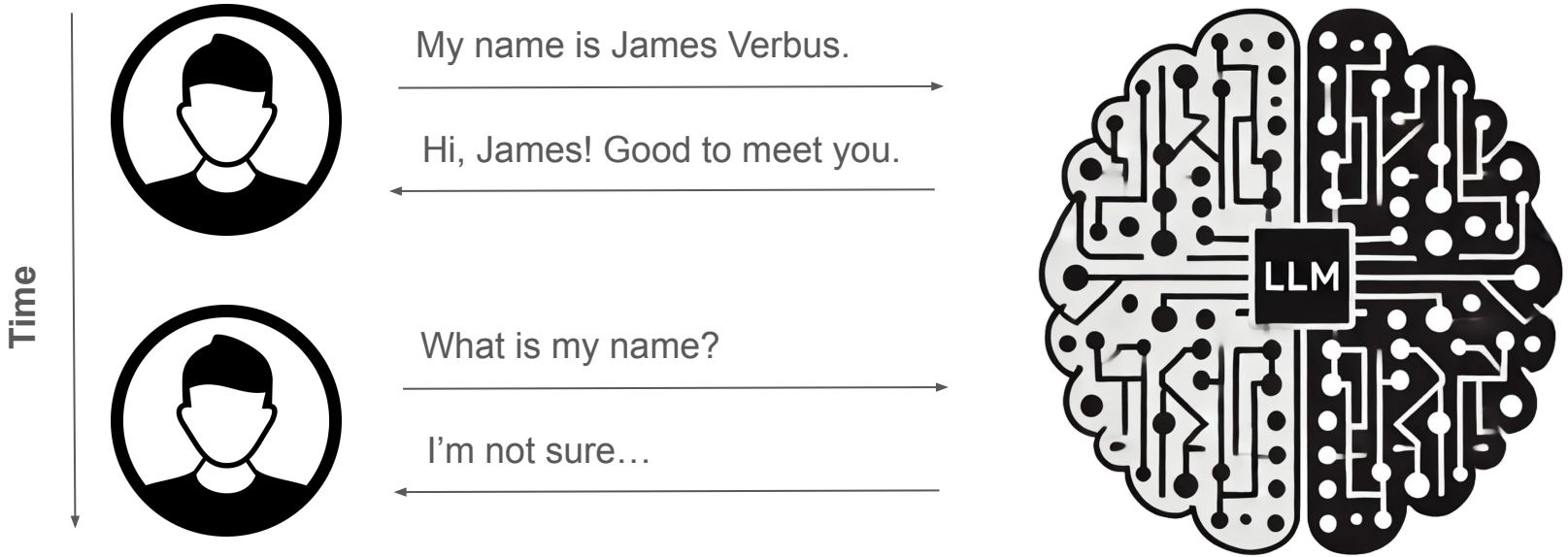
2. How to enable the LLM to answer technical, domain-specific questions
    a. e.g., questions about your research leveraging physics papers, analysis documents, and theses

What will we NOT cover today?

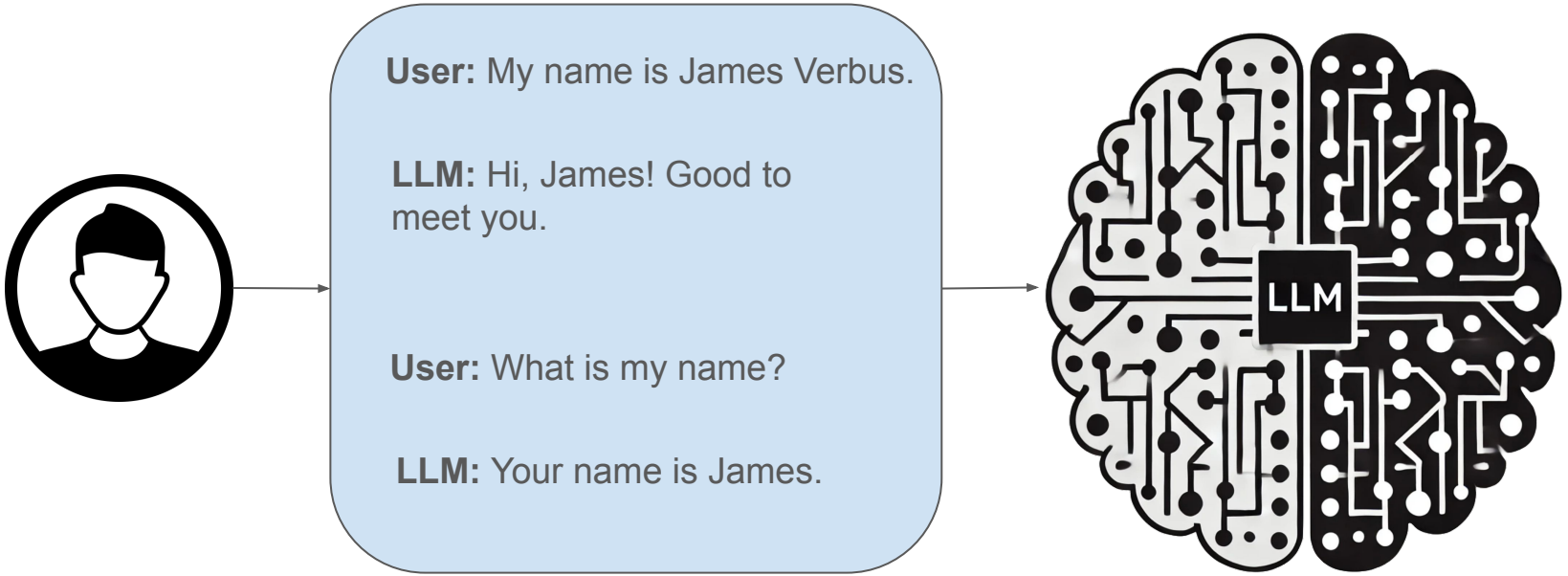
1. Foundational concepts behind LLMs. I recommend the following resources for self-study:
    a. Videos
        i. Andrej Karpathy's "State of GPT"
        ii. Andrej Karpathy's "Let's build GPT: from scratch, in code, spelled out."
    b. Papers
        i. Llama 2: Open Foundation and Fine-Tuned Chat Models
        ii. The Llama 3 Herd of Models

How does LLM chat work?

LLMs, by themselves, are stateless



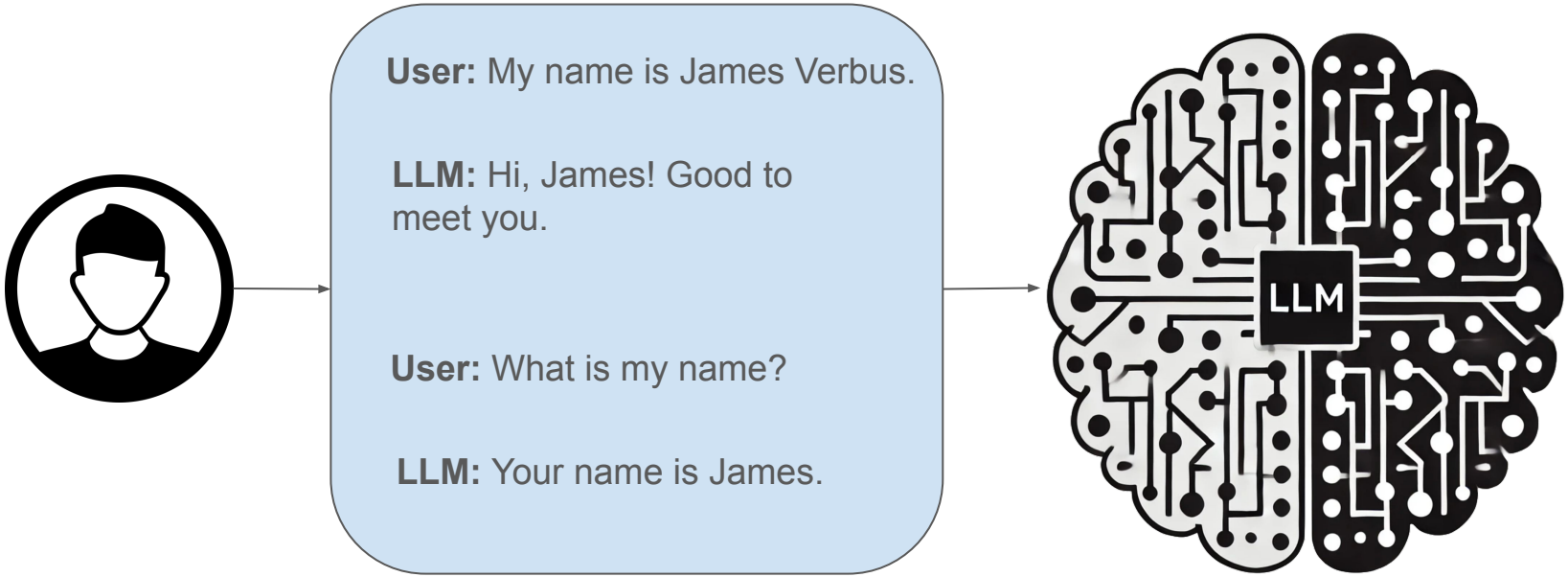
A client layer can be used to provide “memory”



A chat client:

1. Stores the chat history
2. Passes the chat history the to LLM in next query for context

Limitation: LLM context window



- The LLM processes all the tokens in the prompt up to the LLM's maximum context length
- If the chat history exceeds the context window limit, truncation or summarization is required

Domain-specific queries

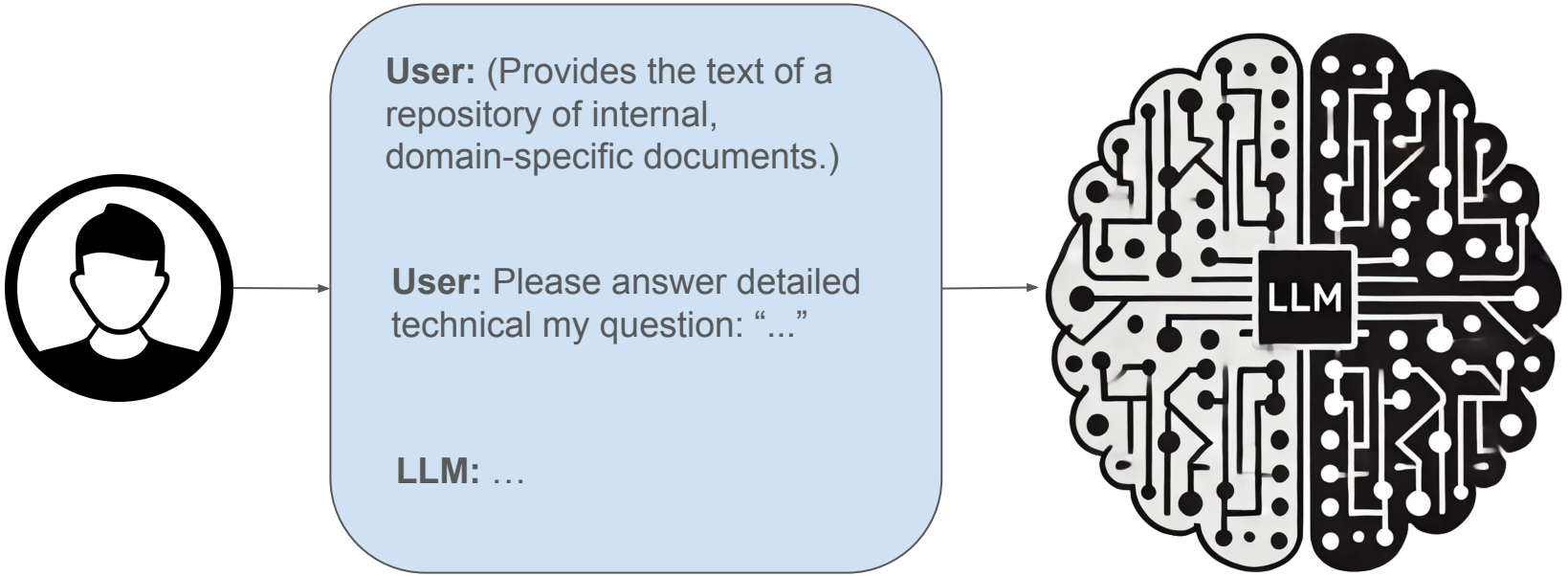
Domain-specific queries

Given that:

- LLMs are trained using data available at the time of training
- Private data and new data will not be available in publicly available LLMs

How can you enable an LLM to answer technical, domain-specific questions about your research leveraging physics papers, internal analysis documents, and theses?

Can we add additional docs in the LLM context window?

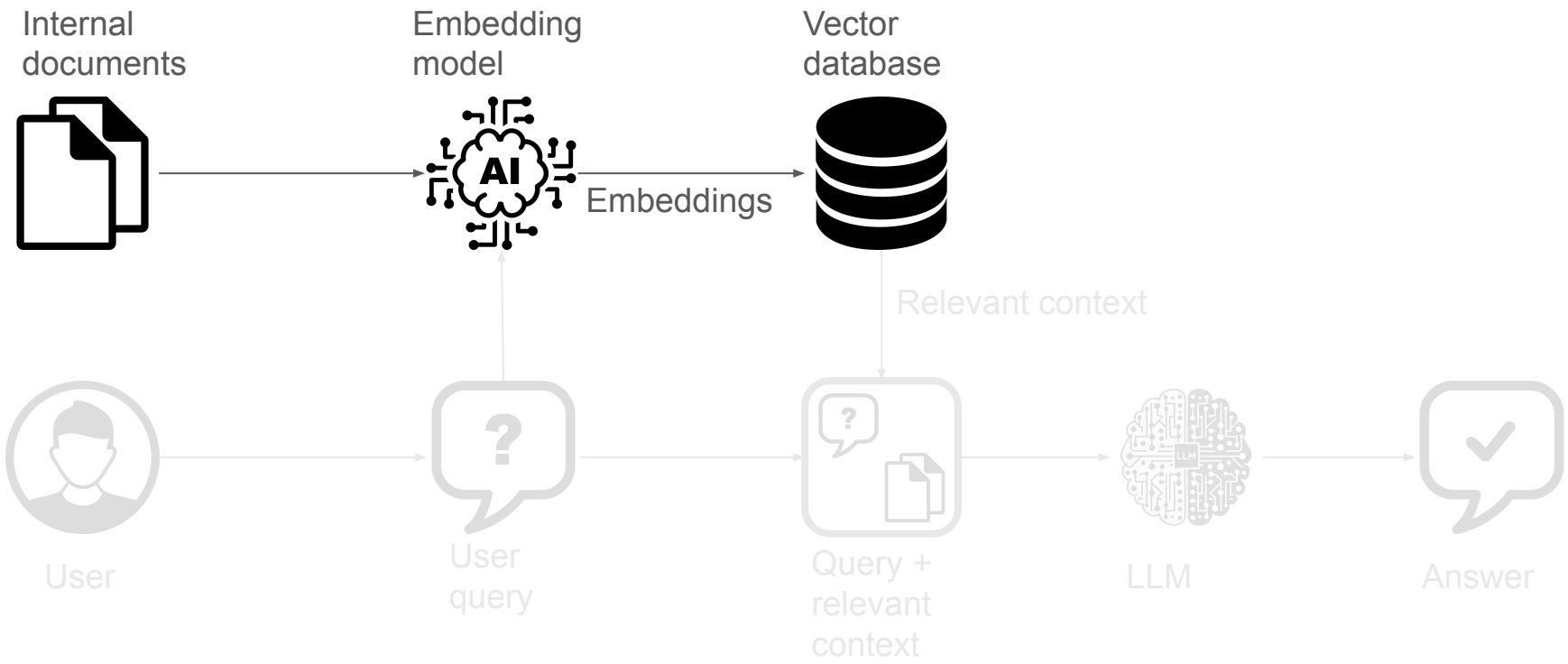


- Context-windows are typically between ~1e3 to ~10e5 tokens, and they are getting longer
- Even with ~10e5 tokens, this only fits ~100 pages of human text into the context window

Retrieval augmented generation (RAG)

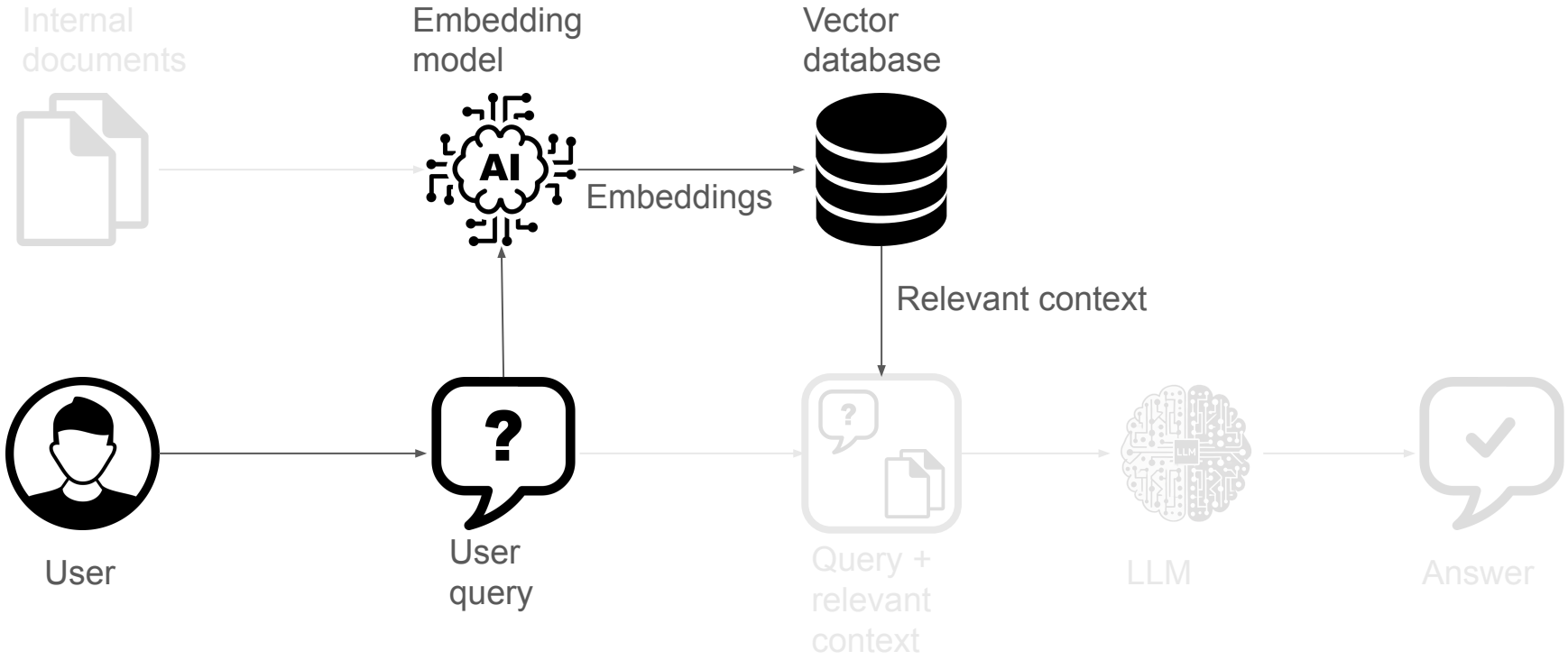
Retrieval augmented generation (RAG)

Step 1: Load documents into RAG vector database



Retrieval augmented generation (RAG)

Step 2: User query and retrieval of relevant context



Retrieval augmented generation (RAG)

Step 3: Augmentation and generation