

# Project Week Kick-Off

## AI-Powered Real-Time Support for Self-Paced Study

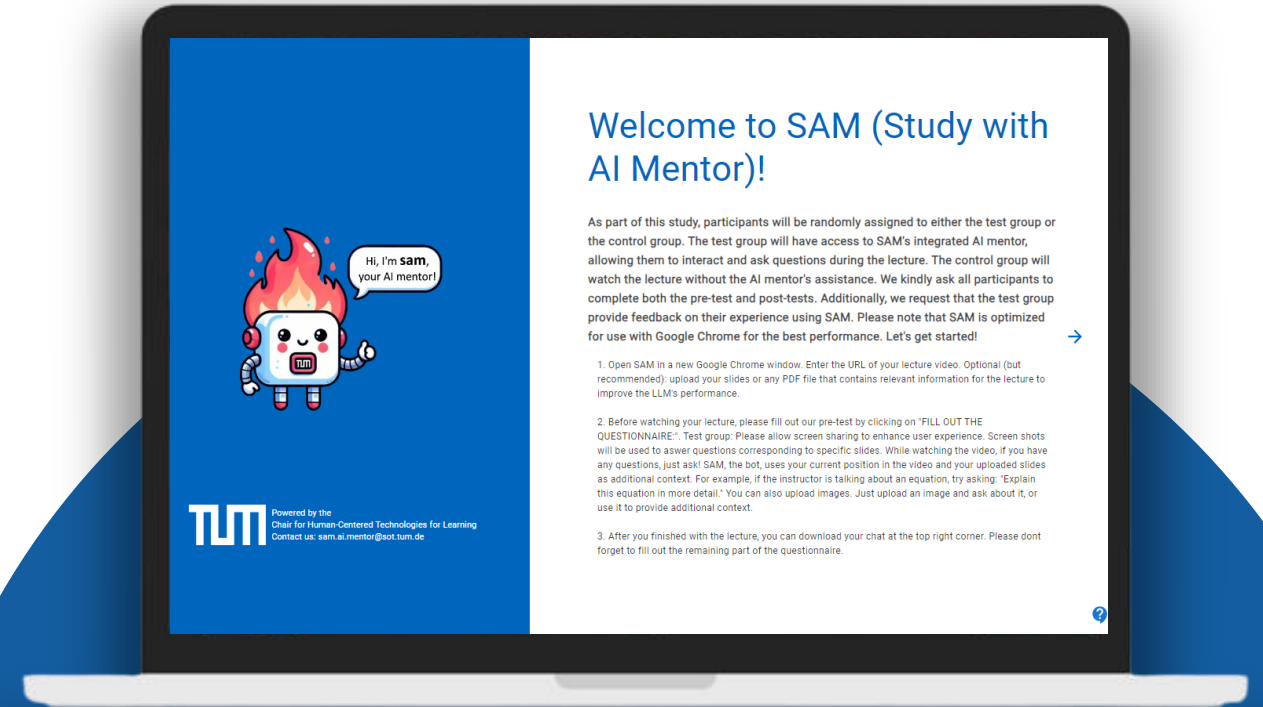
**Anna Bodonhelyi,  
Süleyman Özdel,  
Enkelejda Kasneci**

31 March - 11 April, 2025

Contact:

[anna.bodonhelyi@tum.de](mailto:anna.bodonhelyi@tum.de)

[ozdelsuleyman@tum.de](mailto:ozdelsuleyman@tum.de)



# Motivation

## Engagement

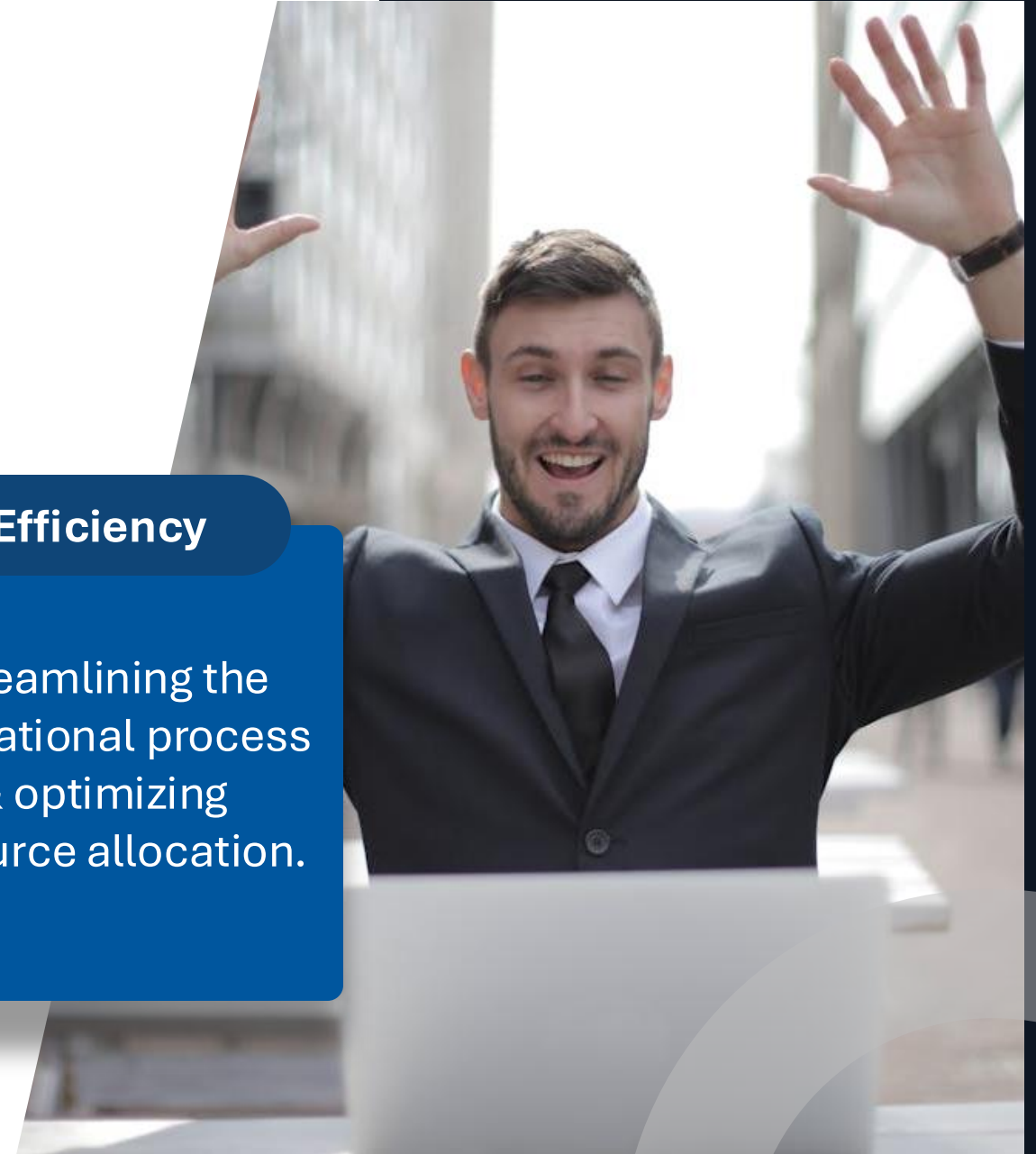
Interactive & adaptive learning experiences.

## Personalization

Tailoring educational content & feedback with AI.

## Efficiency

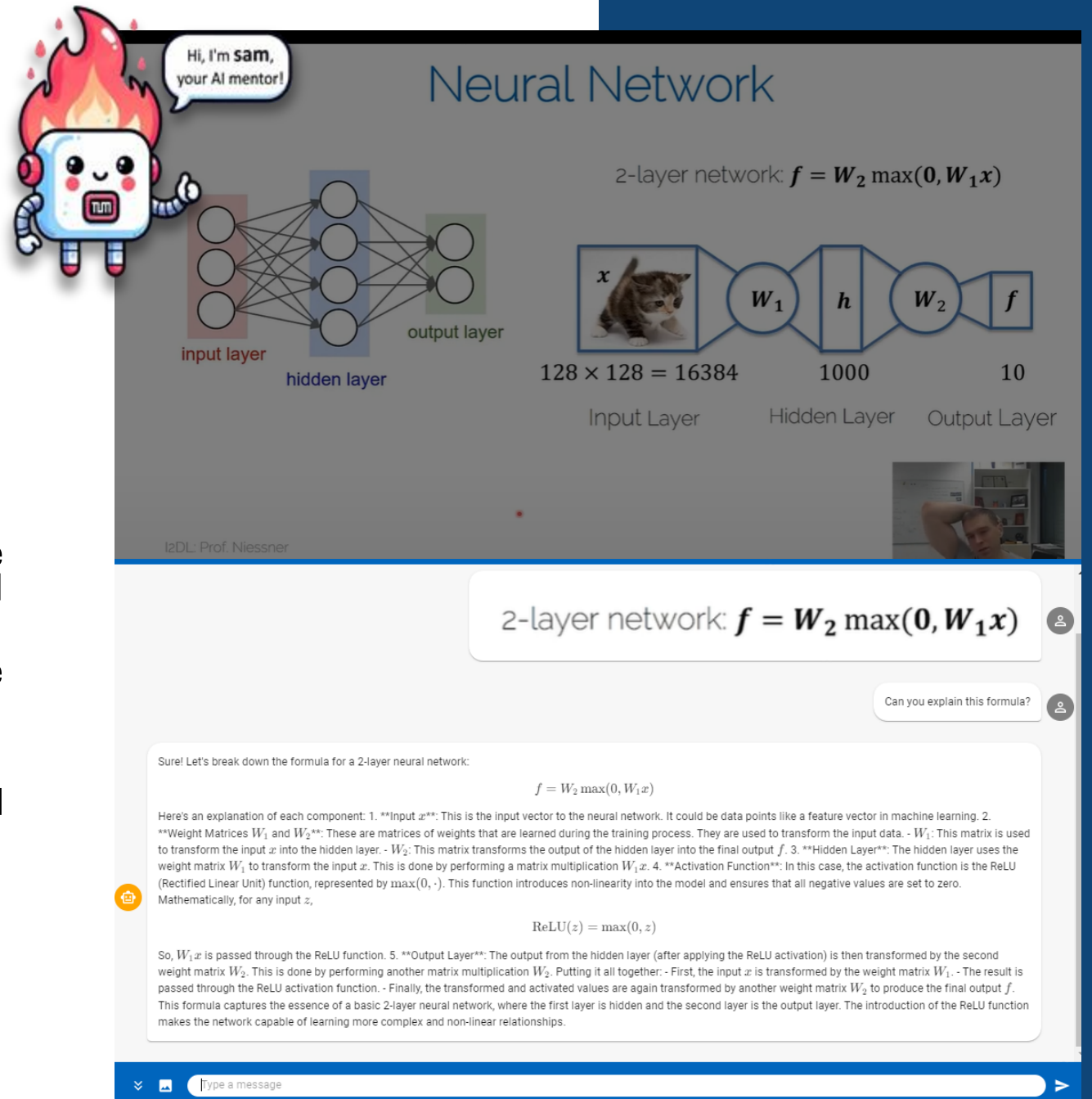
Streamlining the educational process & optimizing resource allocation.



# SAM

## Study with AI Mentor Current status

- ▶ SAM knows the content of the lecture from the video, transcript, and uploaded slides.
- ▶ SAM knows the context closely before and after the current time point.
- ▶ Images can be uploaded.
- ▶ SAM outputs formulas in LaTeX and understands formulas given in LaTeX.
- ▶ Our publication & more details:
  - <https://arxiv.org/abs/2409.15843>



The image shows a presentation slide titled "Neural Network" and a chat interface with an AI mentor named SAM.

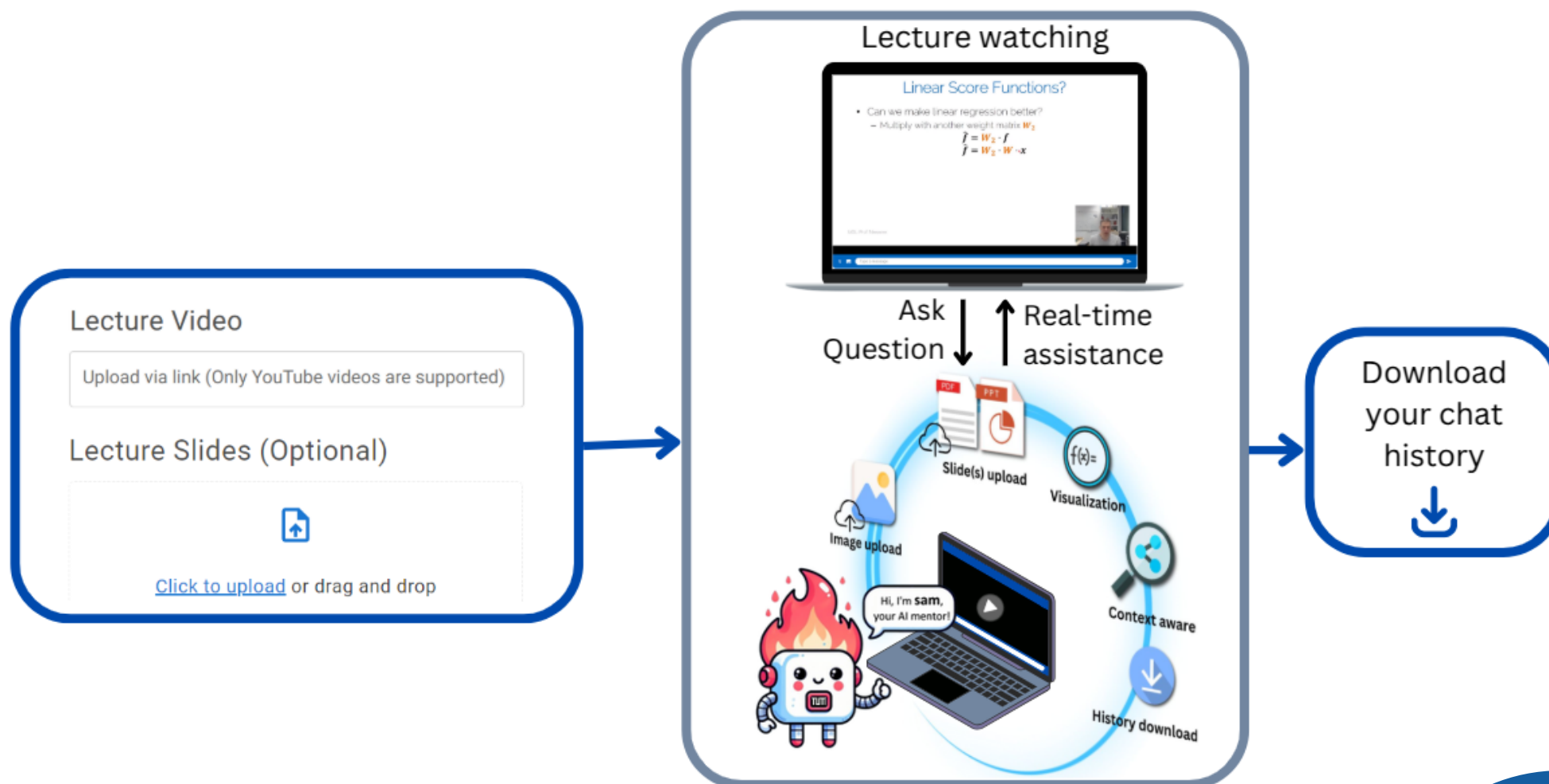
**Neural Network Slide:**

- Character:** A cartoon robot with a flame on its head, saying "Hi, I'm sam, your AI mentor!".
- Diagram:** A 2-layer neural network diagram with an input layer (3 nodes), a hidden layer (4 nodes), and an output layer (2 nodes).
- Equation:** 
$$2\text{-layer network: } f = W_2 \max(0, W_1 x)$$
- Visual Representation:** A sequence of boxes representing the layers: Input Layer (containing an image of a cat labeled  $x$ ), Hidden Layer (containing a box labeled  $h$ ), and Output Layer (containing a box labeled  $f$ ). Dimensions are given below:  $128 \times 128 = 16384$  for the input,  $1000$  for the hidden, and  $10$  for the output.
- Text:** "I2DL Prof. Niessner" is visible in the bottom left corner of the slide.

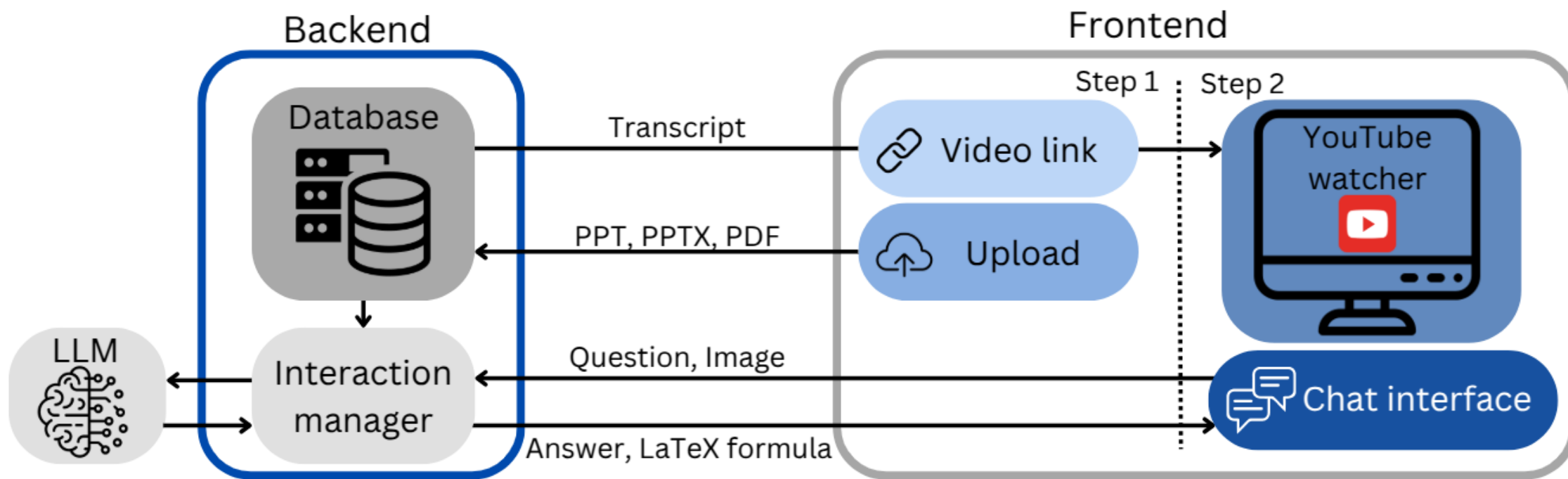
**Chat Interface:**

- Message:** "2-layer network:  $f = W_2 \max(0, W_1 x)$ "
- Response:** "Sure! Let's break down the formula for a 2-layer neural network:"
- Equation:** 
$$f = W_2 \max(0, W_1 x)$$
- Text:** "Here's an explanation of each component: 1. **Input  $x$** : This is the input vector to the neural network. It could be data points like a feature vector in machine learning. 2. **Weight Matrices  $W_1$  and  $W_2$** : These are matrices of weights that are learned during the training process. They are used to transform the input data. -  $W_1$ : This matrix is used to transform the input  $x$  into the hidden layer. -  $W_2$ : This matrix transforms the output of the hidden layer into the final output  $f$ . 3. **Hidden Layer**: The hidden layer uses the weight matrix  $W_1$  to transform the input  $x$ . This is done by performing a matrix multiplication  $W_1 x$ . 4. **Activation Function**: In this case, the activation function is the ReLU (Rectified Linear Unit) function, represented by  $\max(0, \cdot)$ . This function introduces non-linearity into the model and ensures that all negative values are set to zero. Mathematically, for any input  $z$ ,
- Equation:** 
$$\text{ReLU}(z) = \max(0, z)$$
- Text:** "So,  $W_1 x$  is passed through the ReLU function. 5. **Output Layer**: The output from the hidden layer (after applying the ReLU activation) is then transformed by the second weight matrix  $W_2$ . Putting it all together: - First, the input  $x$  is transformed by the weight matrix  $W_1$ . - The result is passed through the ReLU activation function. - Finally, the transformed and activated values are again transformed by another weight matrix  $W_2$  to produce the final output  $f$ . This formula captures the essence of a basic 2-layer neural network, where the first layer is hidden and the second layer is the output layer. The introduction of the ReLU function makes the network capable of learning more complex and non-linear relationships."

# Concept



# Concept



# Scope of the Project Week – base idea

## **WP 1: User database, login for admins and students: ->Om**

- Creating user database
- Only allowing students who registered in TUMOnline to sign up for SAM (automatic student account)
- Randomly assigning the students into 3 groups, storing this in a database
- Admin, Lecturer & Student accounts
  - Admin: developers
  - Lecturer: possibility to upload **videos** (slides + transcripts)
  - Students: possibility to participate in courses (given code by the lecturers, other identification) or to watch any YouTube, TUM Live, or Opencast lecture.
- Implementing an offline video player for uploaded videos –uitable for 50 students simultaneously watching

# Scope of the Project Week – base idea

## **WP 2: Adaptation to User Study -> Chenyan**

- Uploading exemplary lectures (videos+transcripts+slides), creating transcript with Whisper
- Implementing offline LLM instead of OpenAI API
- Creating User study view:
  - 5 lectures (enabling them weekly – set timer) - 5 buttons -> 5 separate sub pages or drop down menu
  - Storing 5 videos with transcripts and slides
  - Integrated Kick-Off questionnaire (before the lectures) - test if integrated questionnaire or Qualtrics would be more robust
  - Integrated questionnaire before and after the video: students can only start lecture watching if they finished the pre-questionnaire
  - Upload possibility of the downloaded chat history (requirement)
  - Log saving: chat history, settings, timestamps, uploaded images – first version is implemented

# Scope of the Project Week – base idea

## WP 3: Proactivity: -> Yang

- Prompt the model based on slides and transcript, to give a **difficulty score** to each part of the lecture – not real-time.
- Track students behavior on the webpage (video paused, rewinded – saved in the chat history)
  - Use this information to choose the right moment to interact. (E. g. if the student rewinds the video, ask a question about the rewinded part.)
- Add the possibility to choose the interaction frequency (high, low, medium)
  - The students should be able to easily select on the interface the desired level on proactivity for SAM.
- Get responses from pre-tests and feed them to SAM.
  - Directly integrate the questionnaires into the website and save the answers in the database
- Pre plan interactions timing for SAM based on the **difficulty score**, the chosen **interaction frequency** and the **pre-test responses**.
- Add Socratic Questioning for SAM's responses
- Advanced log saving (PDF + JSON)



# Requirements

## Documentation:

- Clean code with comments (separate branches) + README.md
- Final presentation on 11.04

## Grading basis:

- Implementation of the base idea in each WP
- Implementation of own ideas
- Design and coding
- Final presentation

# Our meetings (attendance is required)

## Report + Q&A sessions:

- 02.04, 04.04, 07.04. 09.04: ON ZOOM from 15:00
- 10 min long status update, questions

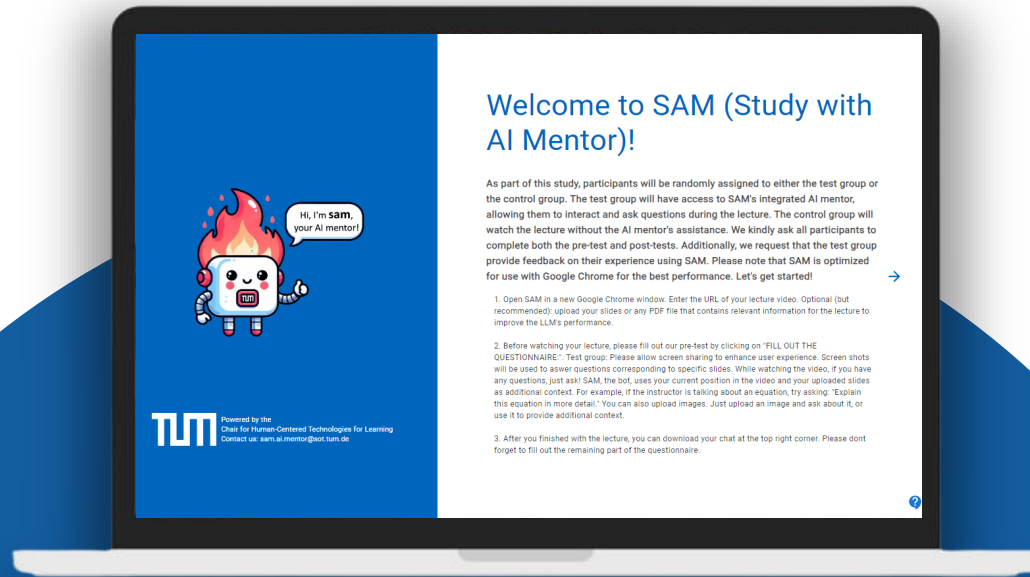
## Final Presentation – Hybrid:

- 11.04 from 14:00

- Zoom (for all sessions):

- <https://tum-conf.zoom-x.de/j/68769863522?pwd=1U1oQb8Z4uJMXy5yeDuLjhjY3mUkEe.1>
- Meeting ID: 687 6986 3522
- Passcode: 416007

If you have questions,  
reach out via E-Mail  
anytime. There is a  
possibility to have quick  
meetings.



# Are you already curious to test SAM?

Try out our demo version  
designed for the user study at:

<https://sam.edu.sot.tum.de/>

Demo video:

