Women in Data Science Worldwide | Dhahran @ KFUPM

ACCELERATING ENGINEERING RESEARCH WITH ADVANCED LLM TECHNIQUES

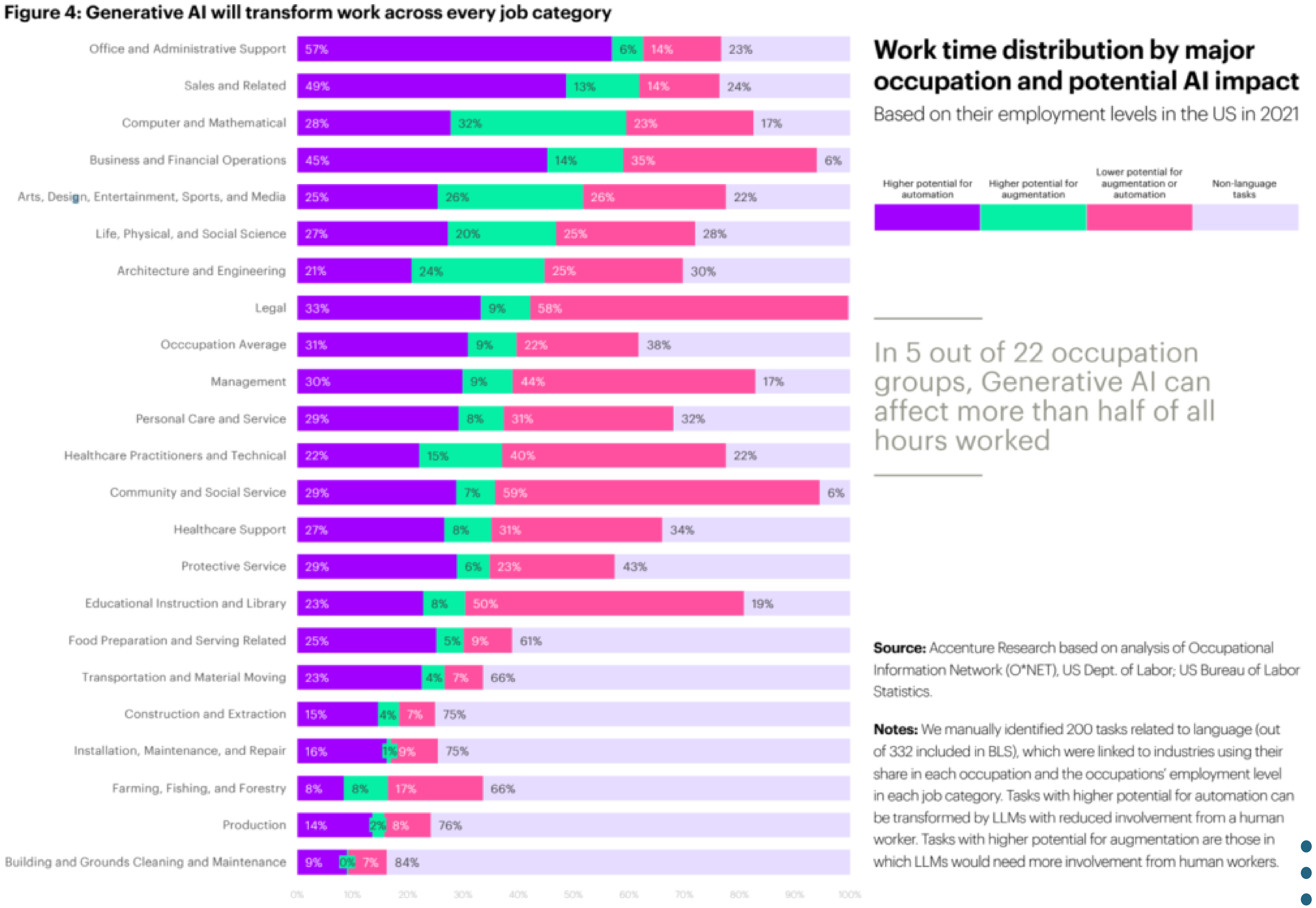
Abrar Alotaibi
PhD. Candidate @ KFUPM

WHY EVERY ENGINEER SHOULD CARE ABOUT LLMS?

01. As of 2023, **over 83%** of academic researchers report using **LLMs** for ideation, writing, and coding assistance (Nature, 2023).

02. OpenAI's API handles over **1 billion token** requests daily; GitHub Copilot **assists 46%** of code commit suggestions in major repositories.

03. Aerospace and automotive companies are **embedding LLMs** into design verification, data logging analysis, and digital twin management workflows.



WORKSHOP OBJECTIVES

- Understand foundational principles and architecture of LLMs.
- Design robust, goal-directed prompts for research-specific use cases.
- Apply LLM techniques to streamline literature reviews and code development.
- Address IP, bias, and ethical deployment of generative AI in technical domains.

WHAT ARE LLMs?

- LLMs are transformer-based **deep neural networks** trained on **massive corpora**, capable of learning linguistic, semantic, and structural patterns in **textual** and symbolic data.

- LLMs output the next likely word (**token**) in a sentence (**sequence**)
  - token: unit of text e.g. word, character. 1 word ~ 0.75 token
  - sequence: context - section ("window") of text e.g. sentence, paragraph, book
  - input into chatGPT is 4096 tokens; Claude 2 is 100K tokens

- The **likelihood of the next** word appearing is determined by
  - the context in which the words are seen in a larger body of text ("corpus")
  - the input to the chat

Next-token-prediction

The model is given a sequence of words with the goal of predicting the next word.

Example:
Hannah is a ____

Hannah is a *sister*
Hannah is a *friend*
Hannah is a *marketer*
Hannah is a *comedian*

WHAT ARE LLMs?

- Learning from a large corpus allows LLMs to "understand" the "meaning" of words.

- For example:

  - the training data may consist of many sentences beginning with "my favorite color is…"
  - the next word will be a color, allowing LLMs to cluster the words "red, blue, green…" into a set that represents the concept of "color"

- It's important to note that LLMs **don't** really understand anything. They create **statistical patterns** that groups similar tokens based on a complex measure of how similar or dissimilar they are.

Next-token-prediction

The model is given a sequence of words with the goal of predicting the next word.

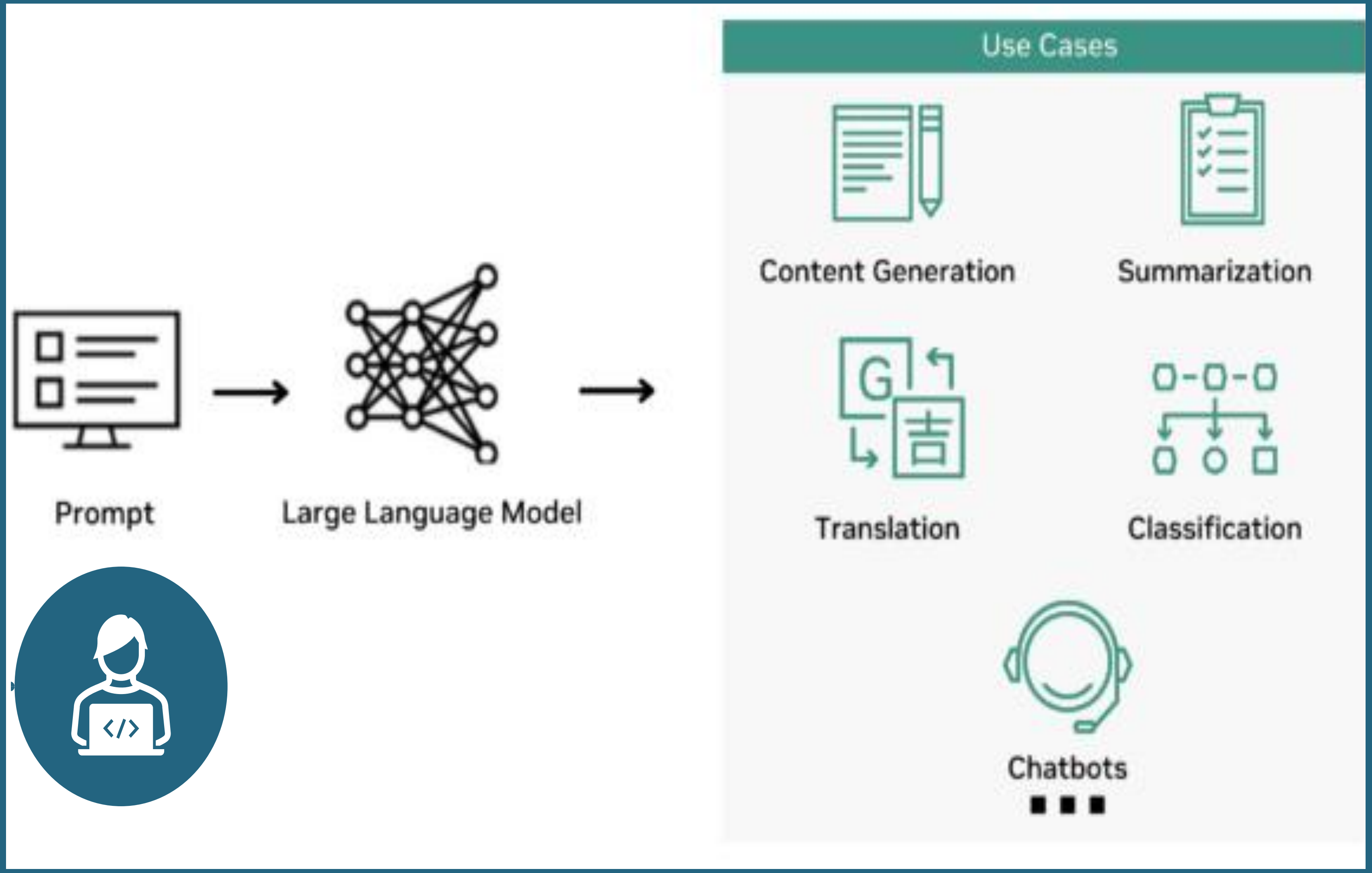
Example:
Hannah is a ____

Hannah is a *sister*
Hannah is a *friend*
Hannah is a *marketer*
Hannah is a *comedian*

HOW CAN LLMS HELP IN SCIENTIFIC RESEARCH?

- **Information Synthesis & Literature Review:** Quickly summarize large volumes of papers, identify key findings, and discover connections across studies.
- **Data Analysis & Interpretation:** Assist in analyzing textual data (e.g., lab notes, interview transcripts), generating code for analysis (Python, R), and interpreting complex results.
- **Hypothesis Generation:** Suggest novel hypotheses based on existing literature and data patterns.
- **Writing & Communication:** Draft manuscripts, grant proposals, reports; improve clarity and grammar; translate research for broader audiences.
- **Code Generation & Debugging:** Write boilerplate code, suggest code snippets for specific tasks, help debug existing code.

This diagram illustrates the **basic workflow**:

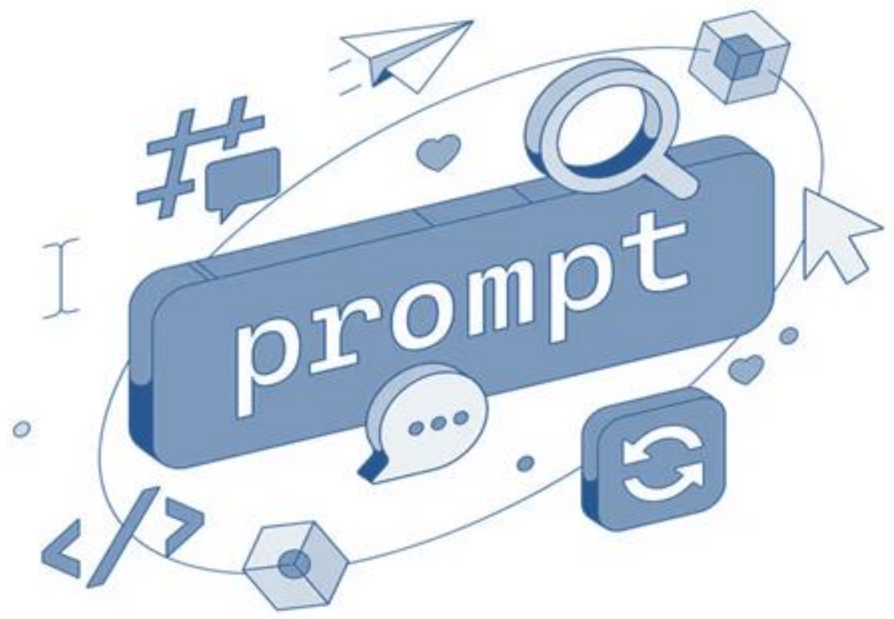


A researcher provides an instruction or question (the **Prompt**) to the **Large Language Model**. The model processes this prompt, enabling various useful outputs or **Use Cases** relevant to research, such as generating content, summarizing text and powering chatbots.

HOW TO INTERACT WITH LLMS - INTRODUCTION TO PROMPTING

What is a Prompt?
It's the instruction or question you give the LLM.

Why is Good Prompting Essential?
The quality of the LLM's output is highly dependent on the clarity, context, and specificity of your prompt.
"Garbage in, garbage out."



Key Elements of Effective Scientific Prompts:

- Be Specific: Clearly define the task and the desired outcome.
- Provide Context: Include relevant background information, data snippets, or constraints.
- Define the Role (Optional but helpful): "Act as a biostatistician..." or "Explain this concept for a first-year undergraduate..."
- Specify the Format: Ask for a "list," "summary," "python code," "table," etc.
- Iterate: Don't expect perfection on the first try. Refine your prompts based on the output.

PROMPTING ESSENTIALS FOR ENGINEERS

Anatomy of a High-Quality Prompt

01. **Role**
Define assumed identity ("You are a structural engineer…")

02. **Task**
Specify objective ("Design a truss under dynamic load…")

03. **Input**
Provide sufficient data/context

04. **Output Format**
Desired style (LaTeX, table, annotated image)

EXAMPLE

"You are a nutritionist specializing in sports performance. Analyze the effectiveness of creatine supplementation for a 25-year-old female weightlifter preparing for national competition, considering her current protein intake of 1.8g/kg and training schedule of 5 days/week. Present your recommendations in a bulleted list with dosage guidelines."

TECHNIQUES FOR EFFICIENT PROMPTS

1. Provide Examples

- Useful examples support reasoning
- Can relate to input or output
- Explain them, if necessary.
- Zero-shot vs. Few-shot

EXAMPLE

Prompt without added example:

“Make a claim on nutrition that might be seen as controversial by health experts.”

Same prompt, improved by an example:

“Make a claim on nutrition that might be seen as controversial by health experts, for example, ‘Chocolate is good for you’.’”

TECHNIQUES FOR EFFICIENT PROMPTS

2. Mark-up your input

- Simple punctuation already helps but using 'fake' mark-up syntax is even more effective
- Standardize inputs especially useful for tools that 'remember'
- Create re-usable templates

EXAMPLE

Prompt without Markup: *"Draft a newspaper paragraph of 500 words including a headline, on the effects of climate change. Apply the simplified grammar of a tabloid paper, and use sensational language. Be creative when describing climate change's effects."*

Same prompt, improved by Markup:

*"**Task:** Draft a newspaper paragraph of 500 words including a headline. **Topic:** Effects of climate change. Style: Use the simplified grammar of a tabloid paper. **Tone:** Use sensational language. **Precision:** Be creative when describing climate change's effects."*

TECHNIQUES FOR EFFICIENT PROMPTS

3. Chain-of- Thoughts

- Guiding step-by-step reasoning for better problem-solving
- Asking for explanations to improve output quality
- Breaking down complex tasks into manageable components

EXAMPLE

Prompt without Chain of Thought:

"Calculate how many gallons of paint are needed to cover a room with dimensions 15' × 12' × 9', assuming one gallon covers 350 square feet, with two coats of paint.“

Same prompt, improved with Chain of Thought:

"**Task:** Calculate paint required. **Approach:** First, calculate the total wall area of a room with dimensions 15' × 12' × 9'. Then, determine gallons needed for two coats, assuming 350 sq ft coverage per gallon. **Process:** Show your calculation for each wall, subtract any windows/doors, sum the areas, determine paint needed for one coat, then for two coats."

WHY LLMS ARE CRITICAL FOR ENGINEERING RESEARCH?

- **Accelerate** systematic literature review through **semantic clustering** and extractive summarization.
- **Automate** experimental planning, hypothesis formulation, and design iteration loops.
- **Assist** with simulation model construction and **code debugging** across numerical domains

RESEARCH PIPELINE – THEN VS. NOW

Phase	Manual Effort	LLM-Augmented Workflow
Literature Review	2–3 weeks	2–3 days
Conceptual Modeling	1 week	1 day
Code Debugging	4–6 hours	30 minutes
Report Drafting	1 week	1 hour

SEMANTIC LITERATURE SEARCH WITH LLMS

01.

Vector-based search surpasses keyword matching by leveraging embeddings

via SCISPACE

02.

Interfaces like arXiv API + OpenAI enable deep filtering of relevant publications

GitHub

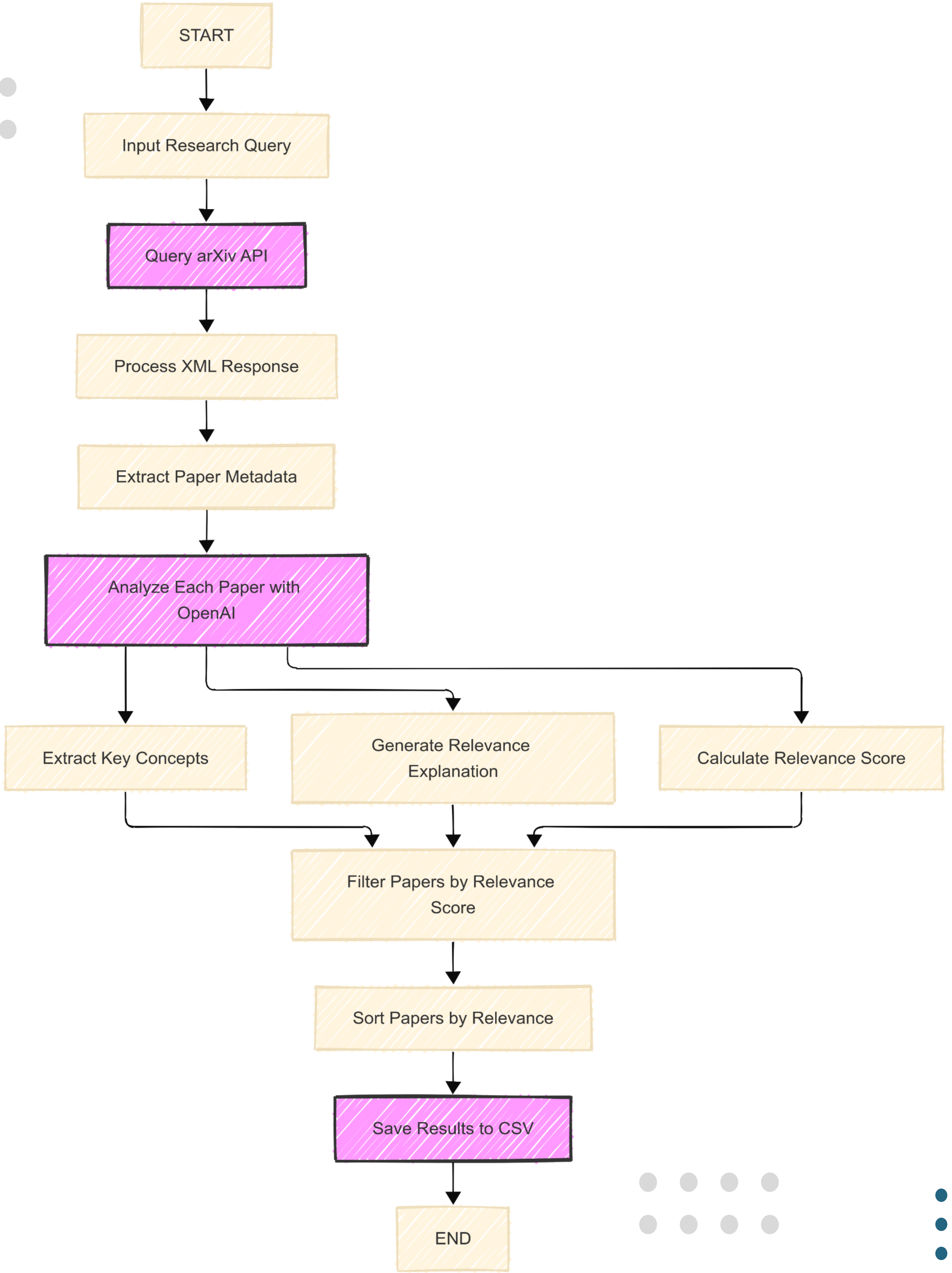
Workflow

REVIEW AUTOMATION PIPELINE

01. Query APIs (Semantic Scholar, arXiv)

02. Parse abstracts into structured chunks

03. Prompt LLM for bullet-point summaries, citation graphs, and cluster themes



WRITE LITERATURE SUMMARY WITH LLMS

Paper Summary

01.

Custom system prompts enable structured extraction of key findings, methods, and limitations for comprehensive paper summaries

02.

Python-based LLM API integration enables automated batch summarization and custom formatting of research insights across multiple papers

GitHub

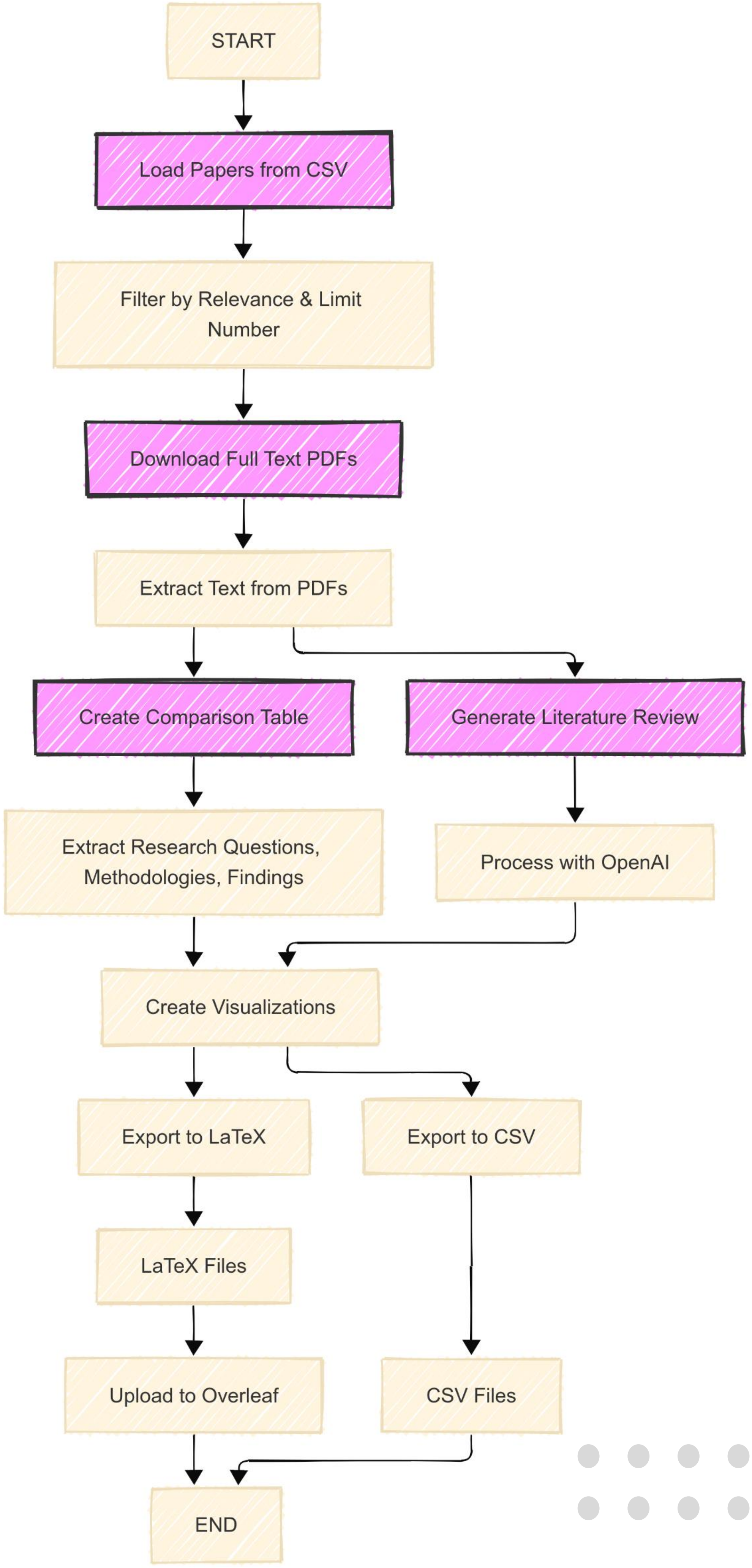
Workflow

REVIEW AUTOMATION PIPELINE

01. Download paper text

02. Parse paper body into structured chunks

03. Prompt LLM for paper reviews and comparison of papers



Problem to Prototype

CODE GENERATION – FROM PROBLEM TO PROTOTYPE

- **Translate** natural language research problems into executable code
- **Supports** MATLAB, Python, R, Julia, and domain-specific languages

EXAMPLE

Prompt Example:
"Write MATLAB code to simulate harmonic motion with damping. Include adjustable parameters for mass, spring constant, and damping coefficient."

ALL the demos in this workshop is generated by Claude 3.7 ☺

Here's MATLAB code to simulate harmonic motion with damping, including adjustable parameters for mass, spring constant, and damping coefficient:

matlab

```
function simulate_damped_harmonic_motion()
    % Parameters (adjustable)
    m = 1.0;      % mass (kg)
    k = 10.0;     % spring constant (N/m)
    c = 0.5;      % damping coefficient (N.s/m)

    % Initial conditions
    x0 = 1.0;     % initial displacement (m)
    v0 = 0.0;     % initial velocity (m/s)

    % Time settings
    tspan = [0 10];  % simulation time range (s)

    % Solve the differential equation
    [t, y] = ode45(@(t, y) system_equations(t, y, m, k, c), tspan, [x0; v0]);

    % Plot results
    figure;
    subplot(2,1,1);
    plot(t, y(:,1), 'b-', 'LineWidth', 1.5);
    title('Damped Harmonic Motion');
    xlabel('Time (s)');
    ylabel('Position (m)');
    grid on;

    subplot(2,1,2);
    plot(t, y(:,2), 'r-', 'LineWidth', 1.5);
    xlabel('Time (s)');
    ylabel('Velocity (m/s)');
```

Reply to Claude...

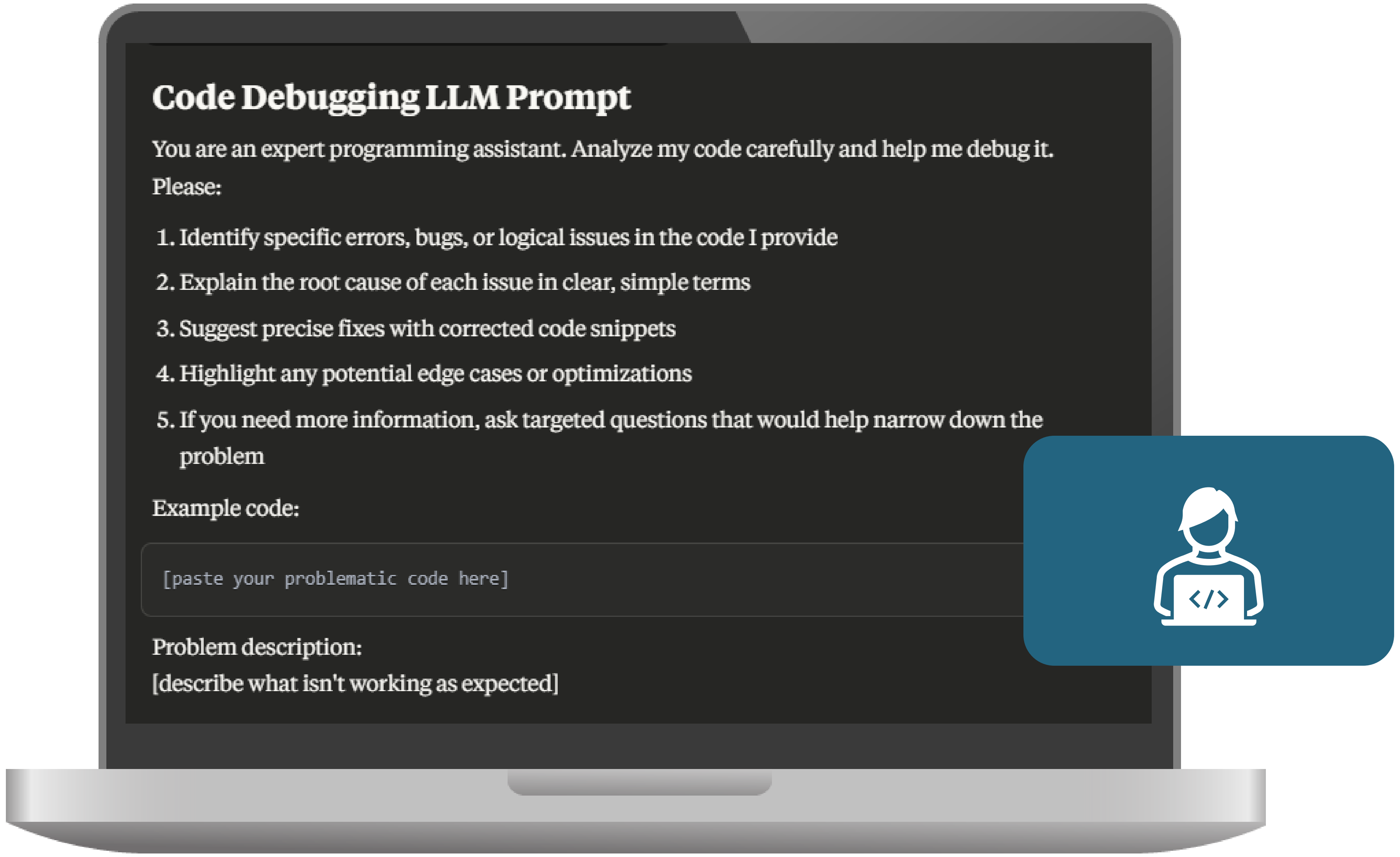
Claude 3.7 Sonnet

DEBUGGING AND ERROR INTERPRETATION WITH LLMS

- **Paste** stack traces or error logs directly into LLM with **context code**
- LLMs return human-readable **explanation** and **corrected code**

EXAMPLE

Prompt Example:
"Fix the following NumPy error: ‘ValueError: operands could not be broadcast together with shapes’. Suggest optimal array dimensions."



Effective base code debugging prompt

CODE OPTIMIZATION & REFACTORING

Refactor for memory /performance efficiency

Replace loops with vectorized ops

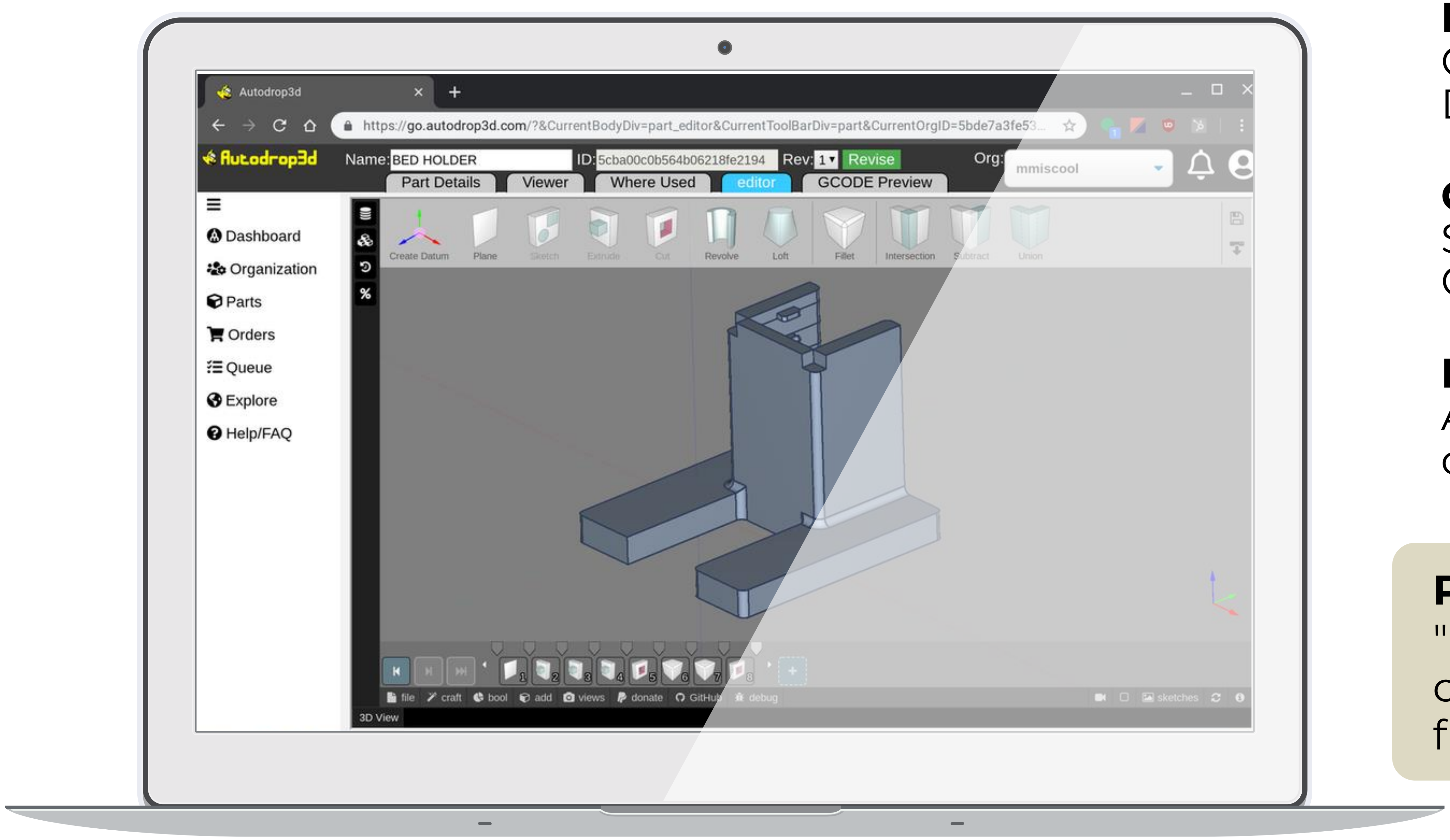
Add inline documentation and tests

EXAMPLE

Prompt Example:
"Optimize this Python heat diffusion model for parallel execution using NumPy and Numba."

LLM INTEGRATION WITH ENGINEERING TOOLS

Coding Demo



MATLAB:
Generate scripts for control systems, DSP, optimization

CAD:
Script geometry using OpenSCAD/FreeCAD APIs

Electronics:
Arduino sketch generation and debugging

EXAMPLE

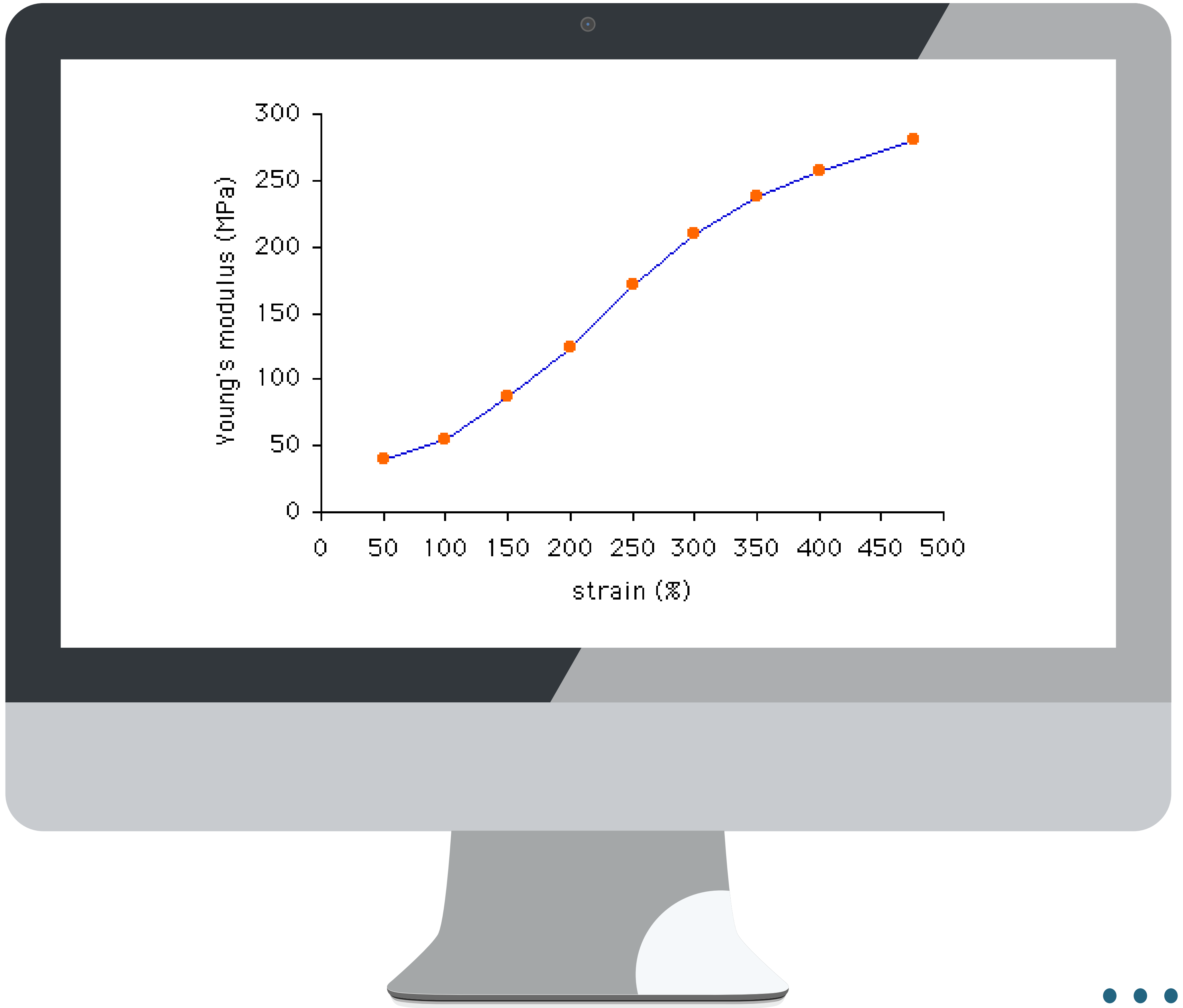
Prompt Example:
"Create OpenSCAD code for a customizable heat sink with adjustable fin spacing."

MULTI-MODAL REASONING
VISUAL + TEXTUAL INPUTS

- Upload graphs, schematics, or diagrams
- Use vision-enabled LLMs (GPT-4V, Gemini) to interpret images

EXAMPLE

Prompt Example:
"Interpret this stress-strain curve. Identify elastic limit, yield strength, and failure mode."

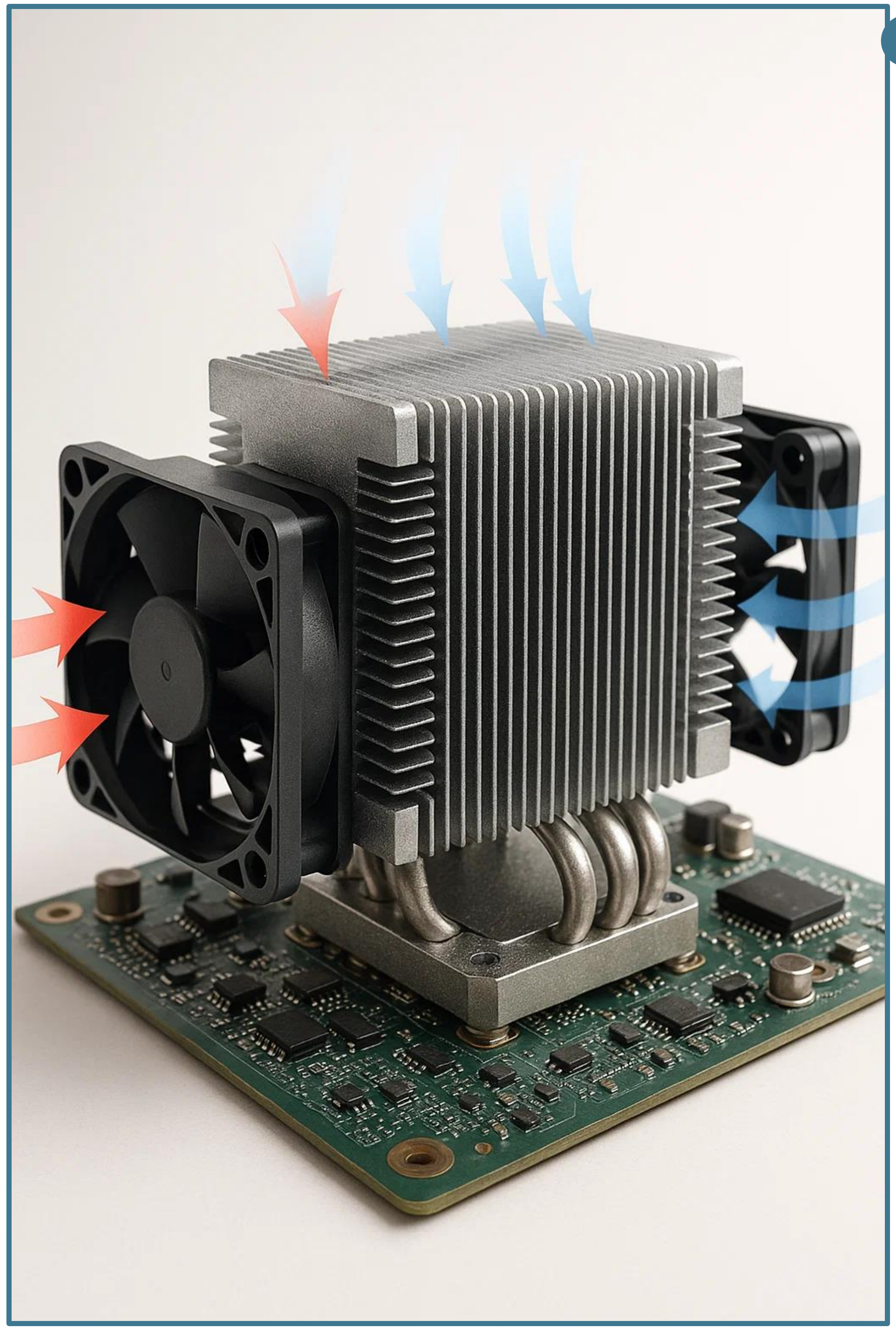


MULTI-MODAL
FIGURES GENERATION

- Combine visual input + numerical data + scenario description
- Output: Engineering prototype, figures and charts
- Creating images requires **very detailed** prompt.

EXAMPLE

Prompt Example:
"Create a detailed and realistic 3D render of a heat sink prototype. The design features a series of finely spaced aluminum fins for optimal thermal dissipation. The heat sink is mounted on a PCB with visible solder points and electronic components around it. Include intricate texturing on the surface of the heat sink, highlighting its metallic finish. Additionally, depict cooling fans strategically placed on either side of the heat sink, with airflow lines illustrated to demonstrate airflow dynamics. The background should be a bright and neutral setting to make the prototype stand out clearly."



Prompt output by Sora

The flowcharts in this workshop are AI-generated by using Mermaid Live ☺

LIMITATION OF LLMs in Scientific Research?

- **Accuracy & Reproducibility Issues:**
  - LLMs can generate incorrect or fabricated information ("hallucinations"), requiring rigorous human verification, especially in scientific contexts. Their outputs are statistical inferences, not guaranteed truths.
  - Outputs can be inconsistent even with the same prompt due to their probabilistic nature, posing challenges for reproducibility.
  - Performance can drop significantly with slight changes in question framing or added complexity.

- **Bias & Fairness:**
  - Models inherit and can amplify biases present in their vast training data (e.g., gender, racial, cultural, geographical biases). This can skew results or reinforce stereotypes.
  - Human feedback used for fine-tuning (RLHF) can also introduce biases.

LIMITATION OF LLMs in Scientific Research?

Lack of True Understanding & Reasoning:

- LLMs excel at pattern matching and language prediction, **not** genuine logical reasoning or causal understanding.
- They may **fail** at complex reasoning or mathematical tasks they haven't specifically seen patterns for.
- They lack grounding in real-world physics or domain-specific nuances unless specifically trained, struggling with specialized scientific concepts.

Data Concerns:

- Using LLMs raises concerns about data privacy, security, and potential leakage, especially with proprietary or sensitive research data.
- Be aware of how the LLM provider uses your input data.
- Knowledge is often limited to the data cutoff date unless continuously updated.
- Training data might also contain errors or be underrepresented in niche scientific areas.

AWARENESS & BEST PRACTICES FOR RESPONSIBLE USE

01.

Maintain Human Oversight & Critical Evaluation

02.

Be Mindful of LLMs Limitations

03.

Protect Sensitive Information

KEY TAKEAWAYS

LLMs

LLMs are productivity multipliers when paired with domain expertise

Prompting is a skill

Prompting is a skill, iterative, strategic, and context-aware

Validate everything

Validate everything: citations, equations, code, and conclusions

Q & A

THANK YOU

GitHub

LinkedIn