

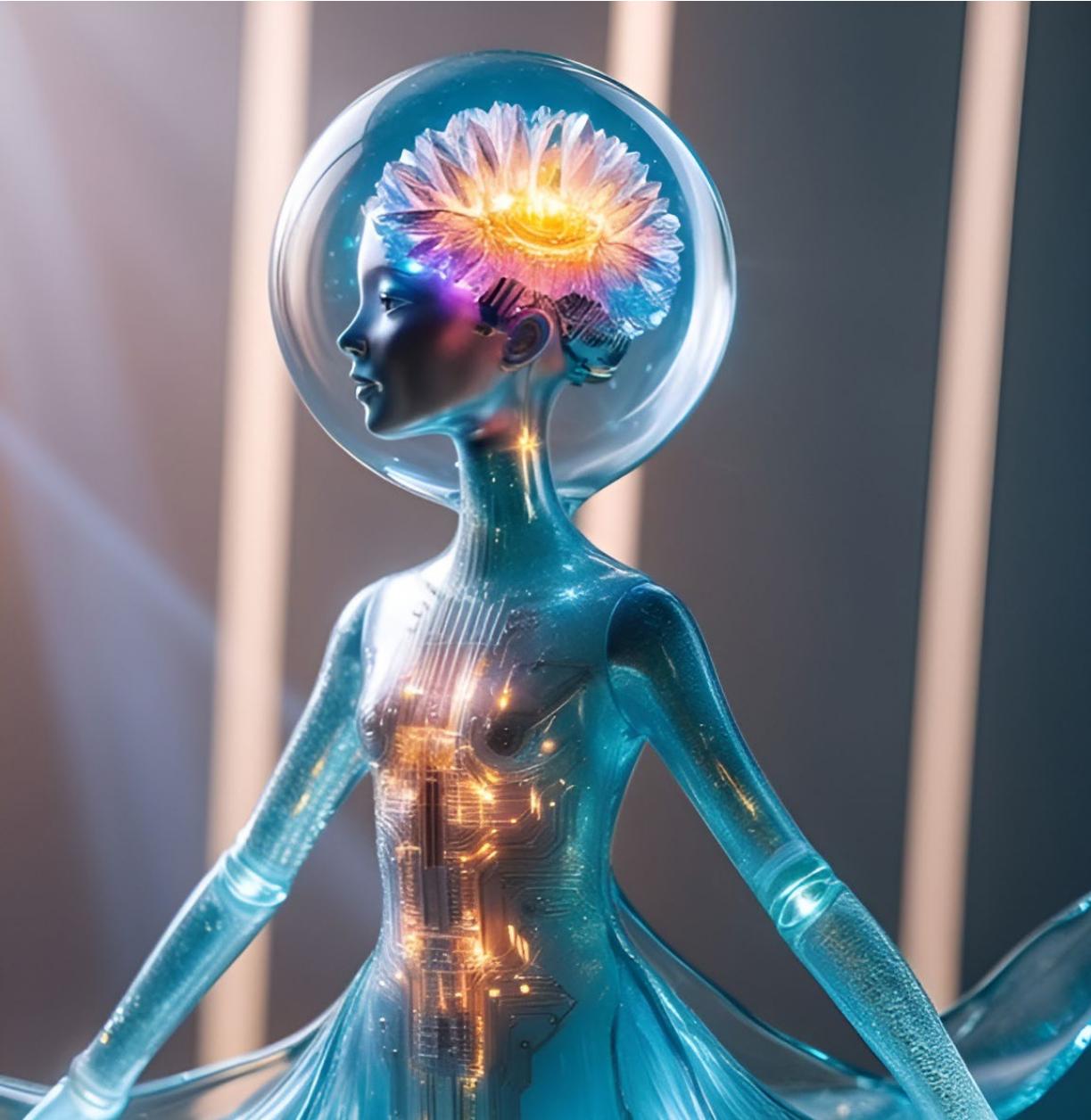


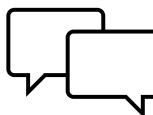
Introduction to AI and Generative AI

**AI Definitions, Achievement Timeline,
Corporations and Current Landscape**



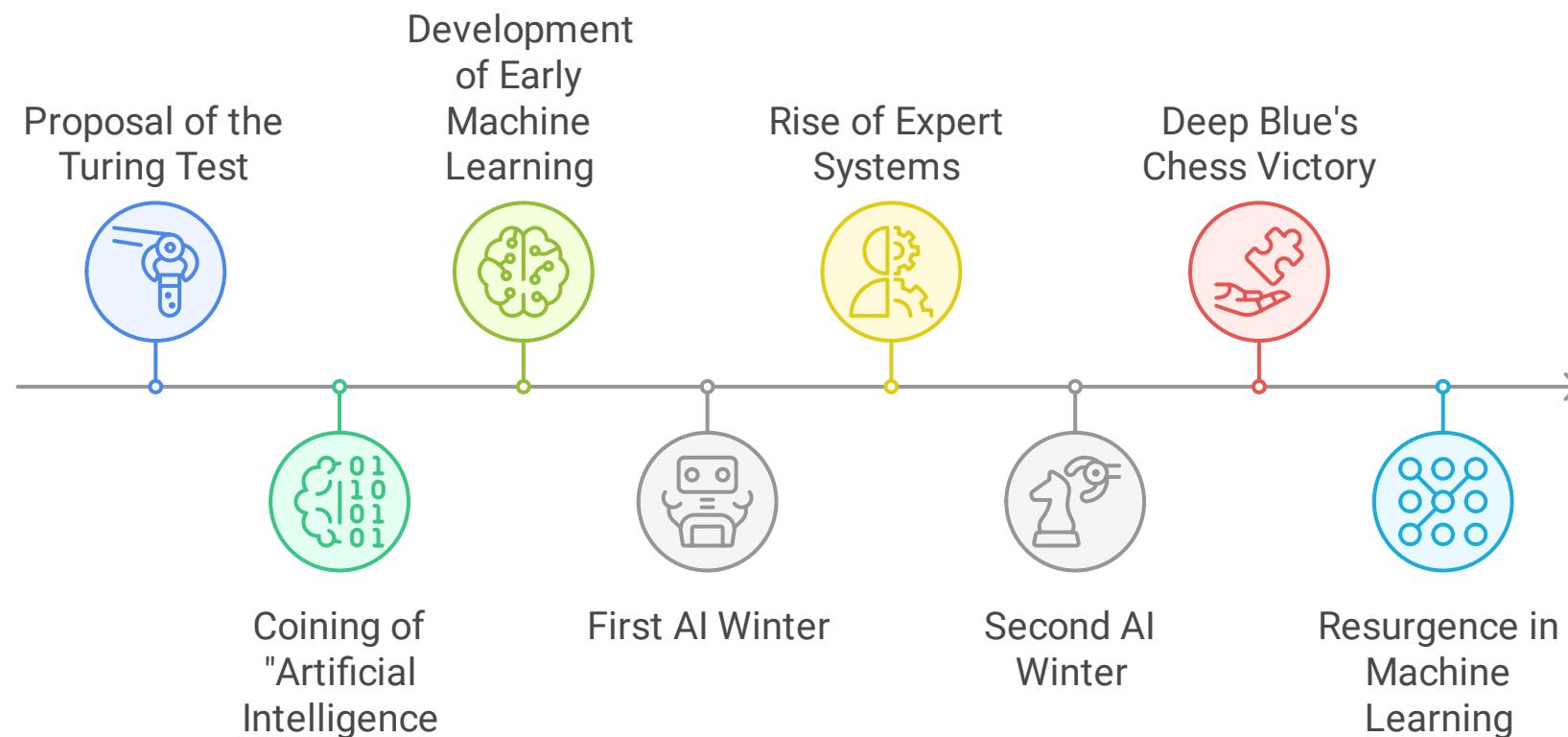
A Definition of Artificial Intelligence (AI)



- ❖ **Definition:** AI is the simulation of human intelligence by machines.
 - ❖ **Applications:** AI is used in a wide range of fields, including healthcare, finance, and robotics.
 - ❖ **Example:** AI in everyday life (e.g., advanced virtual assistants like ChatGPT, which offers speech capabilities and enhanced conversational abilities).
- 
- ❖ **Discussion:** How do you currently interact with AI in your daily life? Spend a few minutes sharing your thoughts.

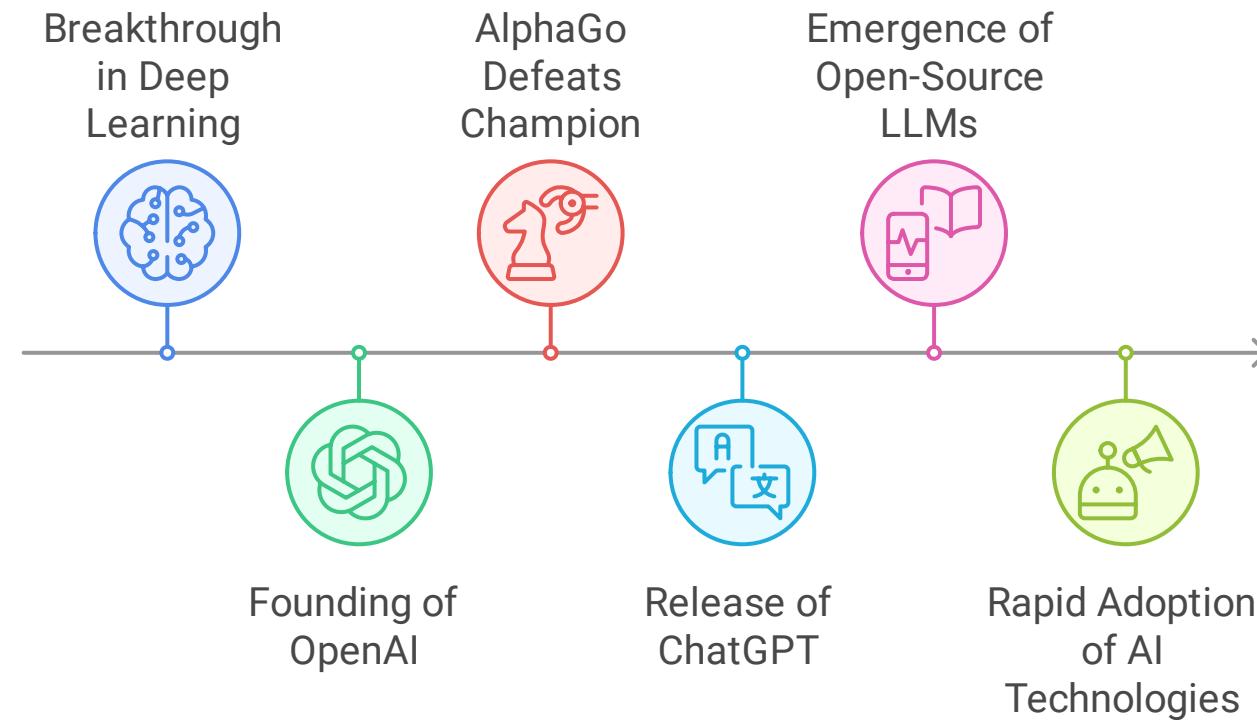


AI Development Timeline (1950-2000)





AI Development Timeline (2000-Present)





Foundational Milestones in Artificial Intelligence



The Turing Test (1950s)



❖ How It Works

- A human evaluator engages in a text-based conversation with both a machine and a human without knowing which is which.
- If the evaluator cannot reliably tell the machine apart from the human, the machine is considered to have passed the Turing test.

❖ Purpose

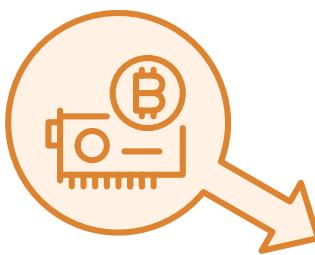
- Turing designed the test to operationalize the question, "Can machines think?" by shifting the focus to observable behavior rather than inner processes.



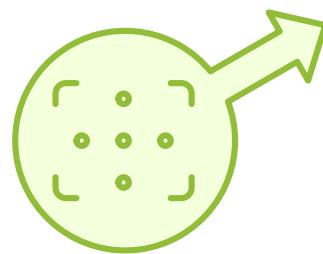
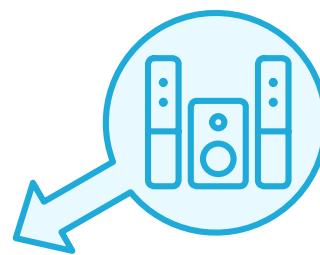
AlexNet (2012)



GPU Utilization



Activation Function

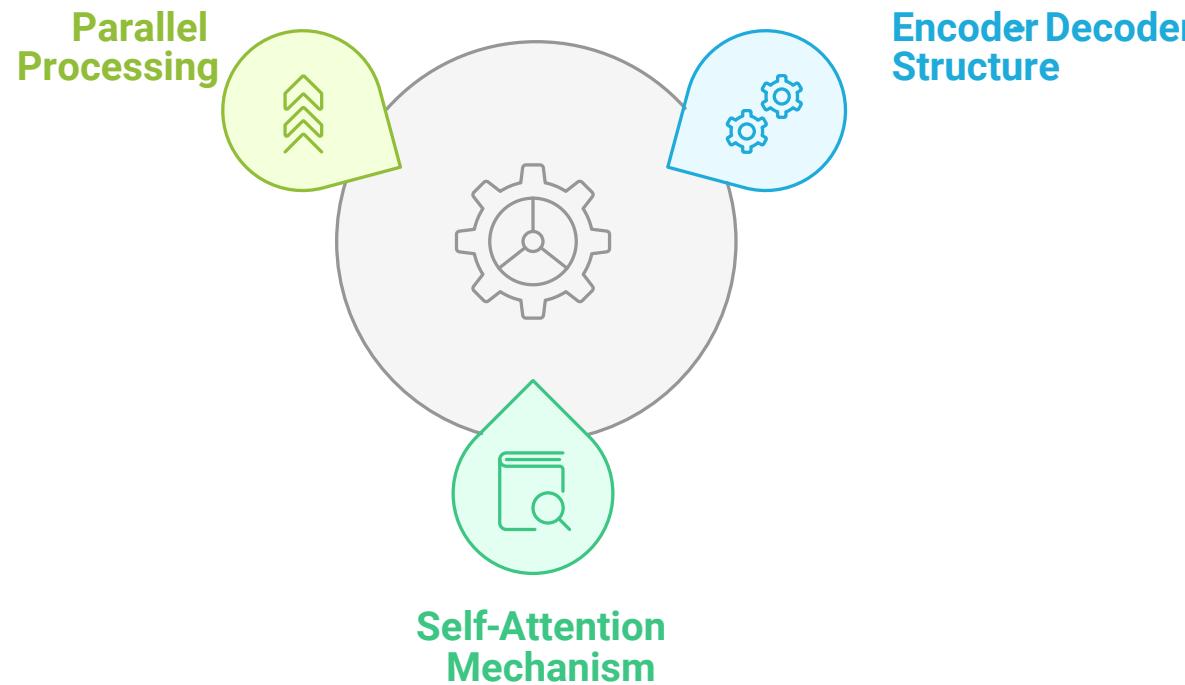


Data Augmentation

- ❖ **Architecture:** AlexNet is a deep convolutional neural network (CNN) with eight layers
- ❖ **Training:** It was trained on the ImageNet dataset, containing over a million images across 1,000 categories, utilizing GPUs to handle its 60 million parameters.
- ❖ **Innovations:** Introduced the ReLU activation function for faster training, used dropout to prevent overfitting, and implemented data augmentation techniques to enhance model robustness.



Transformers (2017)



- ❖ **Architecture:** Transformers utilize an encoder-decoder structure to process input sequences and generate outputs.
- ❖ **Self-Attention Mechanism:** This mechanism allows the model to weigh the importance of different words in a sentence, capturing dependencies regardless of their position.
- ❖ **Parallel Processing:** Unlike traditional recurrent models, transformers process all words in a sequence simultaneously, leading to faster training and improved performance.



Notes in the Age of AI

NotebookLM

Activity

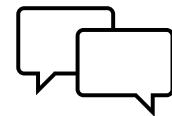


❖ Task:

- Investigate NotebookLM and evaluate its main use-cases.
- Use NotebookLM to create a study summary of NotebookLM.
- Share your summary on MST.

❖ Instructions:

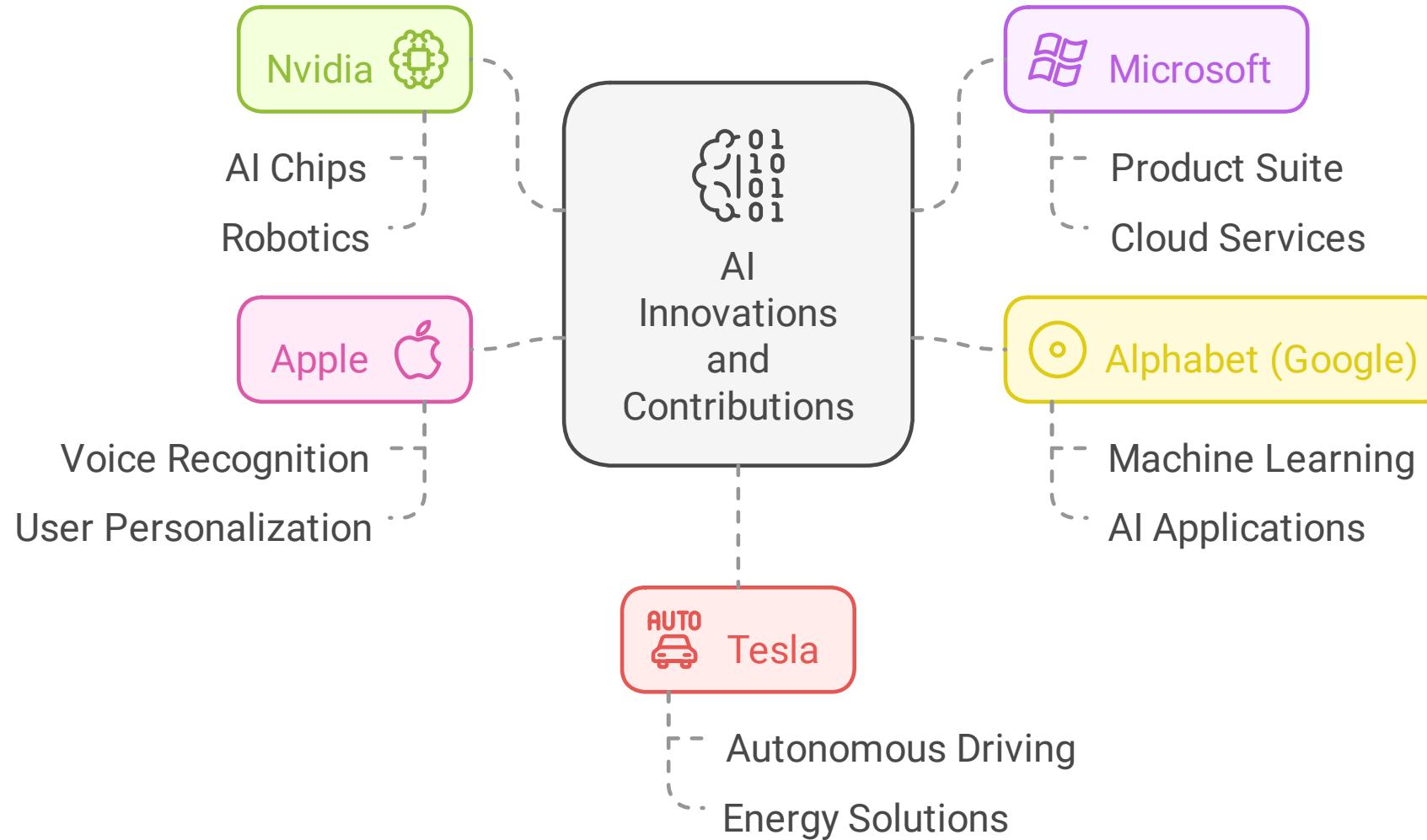
- Explore the features and capabilities of NotebookLM.
- Identify key use-cases (e.g., Ai-powered summarization, audio overviews, content organization, chats and analytics).
- Prepare a note on NotebookLM, summarizing your findings.
- Discuss what you have found with your team and compare notes.



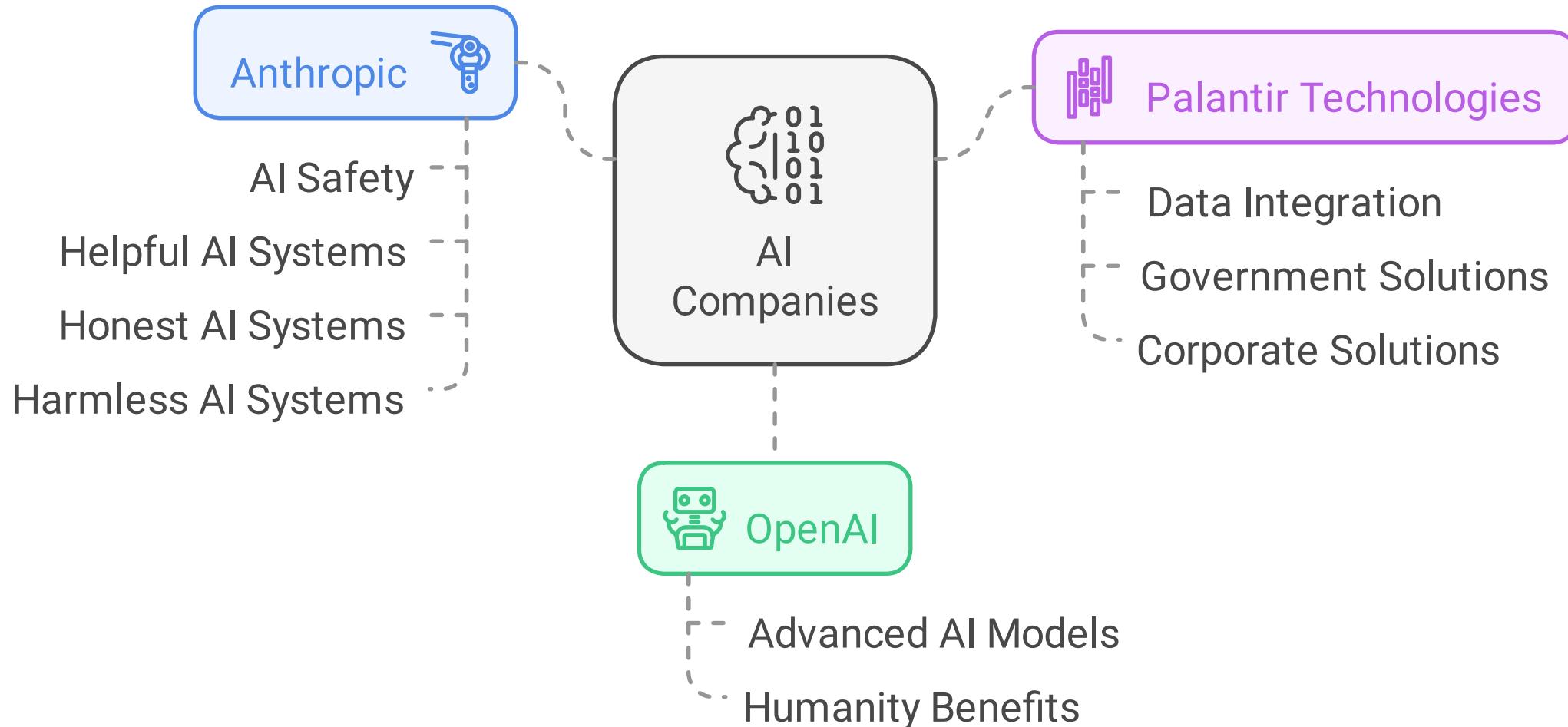


Major AI Corporations and Institutions

Dominant US Corporations

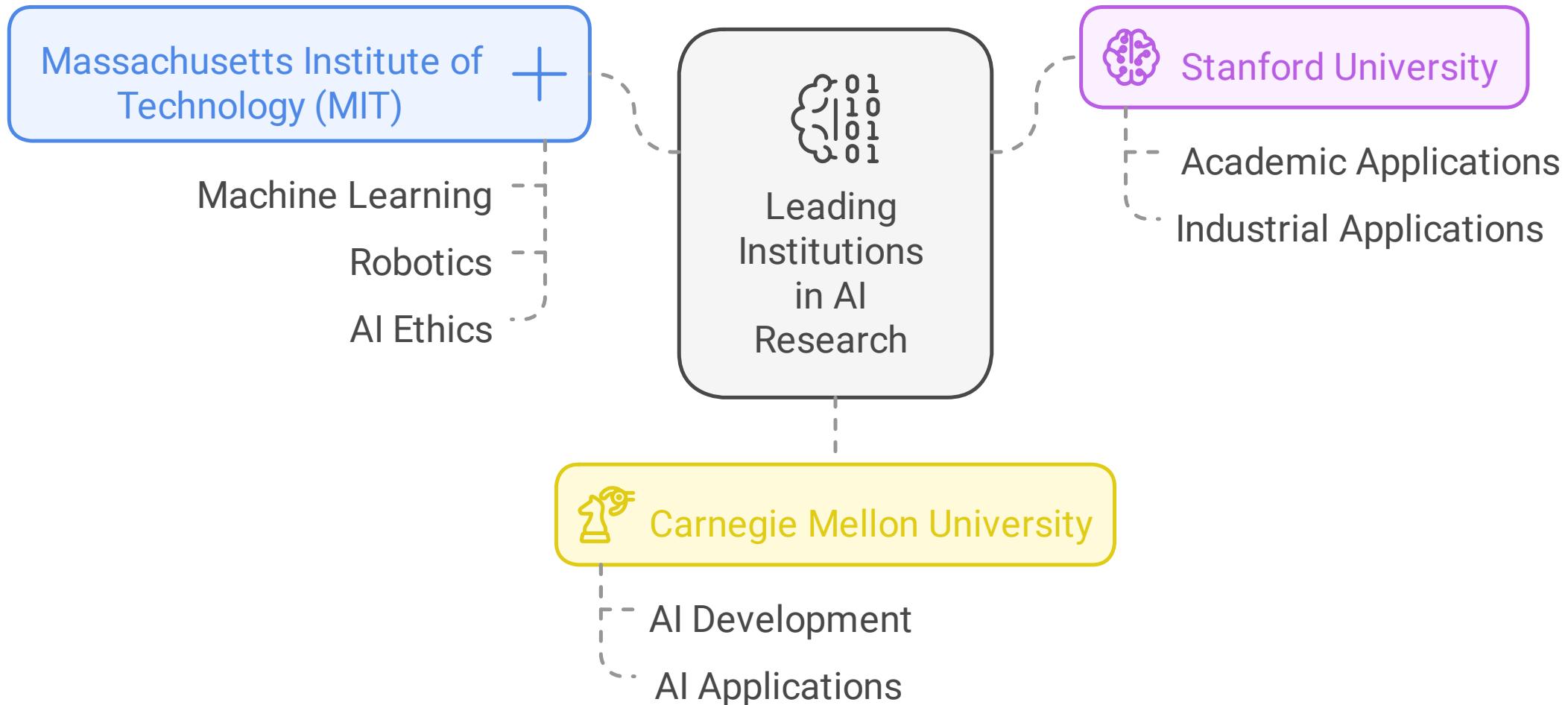


Specialized AI Firms

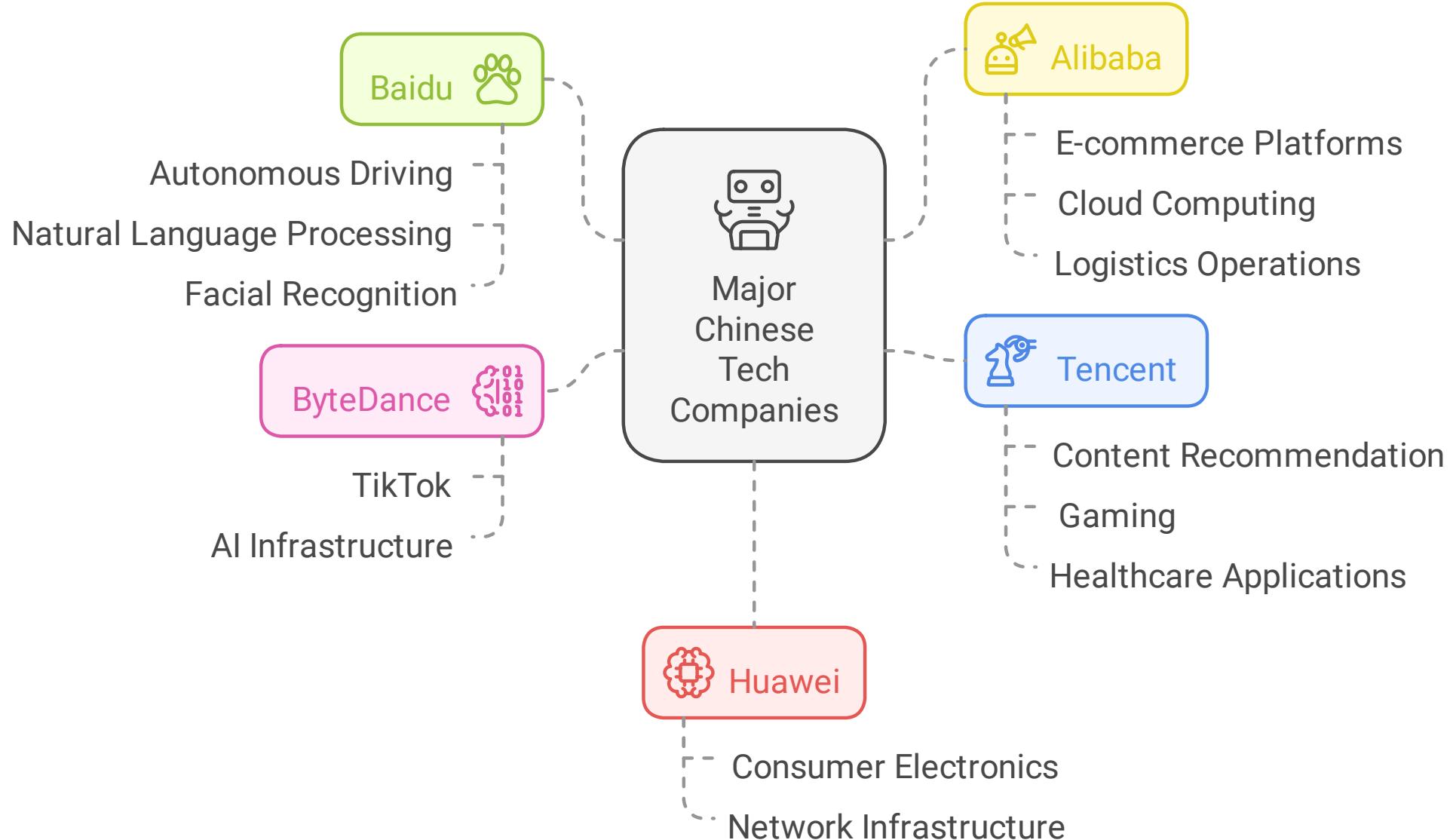




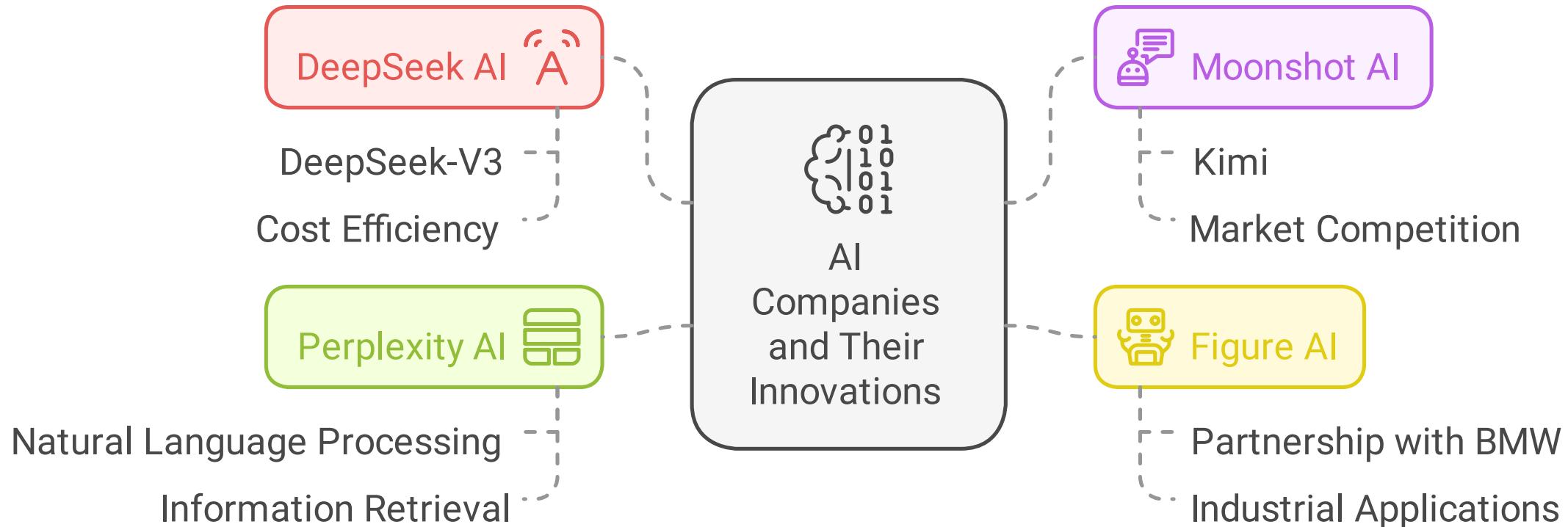
Leading Research Institutes



Asian Heavy-Hitters



Emerging





The AI Landscape

... it's bigger than you think



Activity

- ❖ Click the following link to load the 2024 MAD (ML, AI & Data) Landscape: <https://mad.firstmark.com/>
- ❖ Use the sliders and + - buttons to navigate.
- ❖ Spend a few minutes scrolling around - It's a big world. You can click on any logo for more information on the company. Try to toggle between Landscape and Card.
- ❖ Make a note of any companies that you know about, admire or in which you have interest.
- ❖ Discussion: Discuss with your group 
- ❖ Select three companies that interest you.
- ❖ Make a note in MST, with your reason for selection.

Ai

Activity



Kahoot!



Lesson 01

Summary of Material Covered

Lesson 01 Review



- ❖ **Overview of AI & Generative AI:**
Explains AI & Generative AI definitions and key applications in healthcare, finance, and robotics.
- ❖ **Key AI Achievements:**
Covers major milestones: Turing Test, machine learning, expert systems, Deep Blue, AlphaGo, AlexNet, and transformers.
- ❖ **Top AI Institutions:**
Highlights leading companies: Nvidia, Microsoft, Alphabet, Tesla, OpenAI, and key research institutions like MIT and Stanford together with prominent Chinese firms Baidu, Tencent and emerging startups like Moonshot and Perplexity.
- ❖ **AI Landscape 2024:**
Provides a snapshot of AI sectors: infrastructure, kernels, applications, and data analytics.

Thank you

School of Infocomm

C240 AI Essentials and Innovations

© Republic Polytechnic 2025: All Rights Reserved



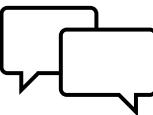
Introduction Prompt Engineering I

Enabling Effective Interaction with AI
systems



A Definition of Prompt Engineering



- ❖ **Definition:** Prompt engineering involves crafting specific inputs (prompts) to guide AI models in generating desired outputs.
 - ❖ **Purpose:** It enhances the relevance and accuracy of AI responses by providing clear and structured prompts.
 - ❖ **Application:** Utilized in AI tasks like text summarization, translation, and content generation to achieve optimal results.
- 
- ❖ **Discussion:** What techniques do you know about or use daily with LLM interactions?



Terminology





Tokens



❖ Definition:

- Tokens are the basic units of text that language models process.
- Often fragments of words, punctuation, or individual letters.
- By breaking text into tokens, a model can handle variable input lengths efficiently. Query: "The candidate was taken aback by the interviewer's question."
- Tokenized Query: "The", "cand", "idate", "was", "taken", "aback", "by", "the", "interviewer", "", "s", "question", ". "

❖ Questions:

- Do LLMs like OpenAI ChatGPT and Google Gemini use tokens to remove sensitive information like phone numbers or salaries that may be accidentally prompted?
- How does tokenization vary across languages?





Embeddings

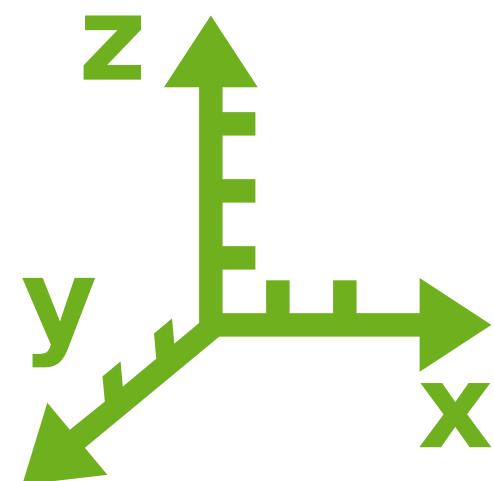


❖ Definition:

- Embeddings are numerical representations of words, phrases, or sentences that capture their meaning in a multi-dimensional space.
- These vectorized forms help models understand semantic relationships between words, even if they are not explicitly stated.
- Embeddings enable tasks like similarity detection, topic modeling, and retrieval-based AI.

❖ Questions:

- How do embeddings help LLMs understand words that weren't in their training data?
- Why are embeddings useful for tasks like search, recommendation systems, or text classification?



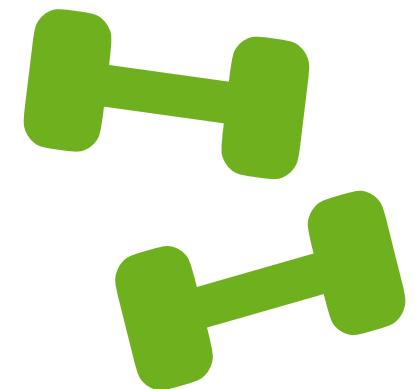


Training



❖ Definition:

- Training in a large language model is the process of teaching the model to understand and generate text. In simple terms, this involves feeding the model a large dataset of text so it can learn patterns, grammar, and context.
- Key hyperparameters are used to fine-tune this learning process:
 - Learning rate (how fast the model updates its weights)
 - Number of layers (the network's depth),
 - Batch size (how many examples are processed at once)
 - Temperature, which affects response creativity, is used during *inference*.



❖ Questions:

- Why do you think increasing the amount of training data positively affects a model's performance?
- What do you think is a key challenge with respect to training?



Overfitting



❖ Definition:

- Overfitting occurs when a model learns noise and random fluctuations in training data rather than the underlying pattern. This leads to excellent performance on training data but poor generalization to new data.

❖ Key Points:

- Caused by excessive model complexity relative to the amount of training data.
- Results in high variance, where training accuracy is high, but test accuracy is low.
- Mitigation methods include regularization, early stopping, and cross-validation.

❖ Questions:

- How can you detect overfitting in a model's performance?
- Why do LLMs require a large amount of data?





Inference



❖ Definition:

- Inference in a large language model is the process of generating text from a pre-trained model. In simple terms, when you enter a prompt (like a question), the model responds with an answer.
- During inference, text is converted into tokens.
- The LLM processes these tokens to predict the next token based on the previous ones. In other words, predict the next word.
- This word is then added to the sequence and the process repeated.

❖ Questions:

- What role do tokens play during inference?
- Why do LLMs sometimes produce incorrect or nonsensical outputs?





Context Window



❖ Definition:

- The context window is the **maximum** number of tokens a language model can process at once, including both input (prompt) and output (response).
- If the total tokens exceed this limit, older tokens are truncated, which can lead to incomplete responses or loss of important information.
- Managing the context window effectively is crucial for tasks requiring long conversations, document analysis, or multi-step reasoning.

❖ Questions:

- If I ask an LLM "How are you" and it responds, how many tokens are used from the context window?
- If I then ask a follow-up question "What is $1 + 1$ " how many tokens are now used from the context window?
- What happens when the window is exceeded?





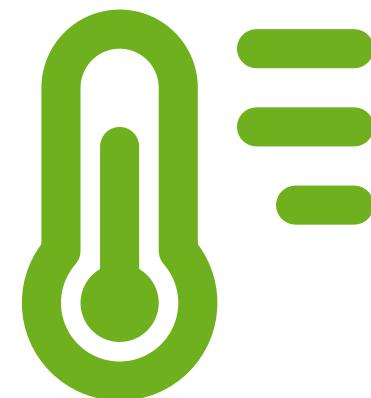
Temperature

❖ Definition:

- Temperature controls the randomness of an LLM's responses by adjusting the probability distribution of word selection.
- A higher temperature value makes the output more diverse and creative by allowing the model to pick less likely words, while a lower temperature makes responses more focused and deterministic by favoring the most probable words.

❖ Questions:

- How should you set the temperature for a science-based question?
- Can I set the temperature in a normal prompt window?



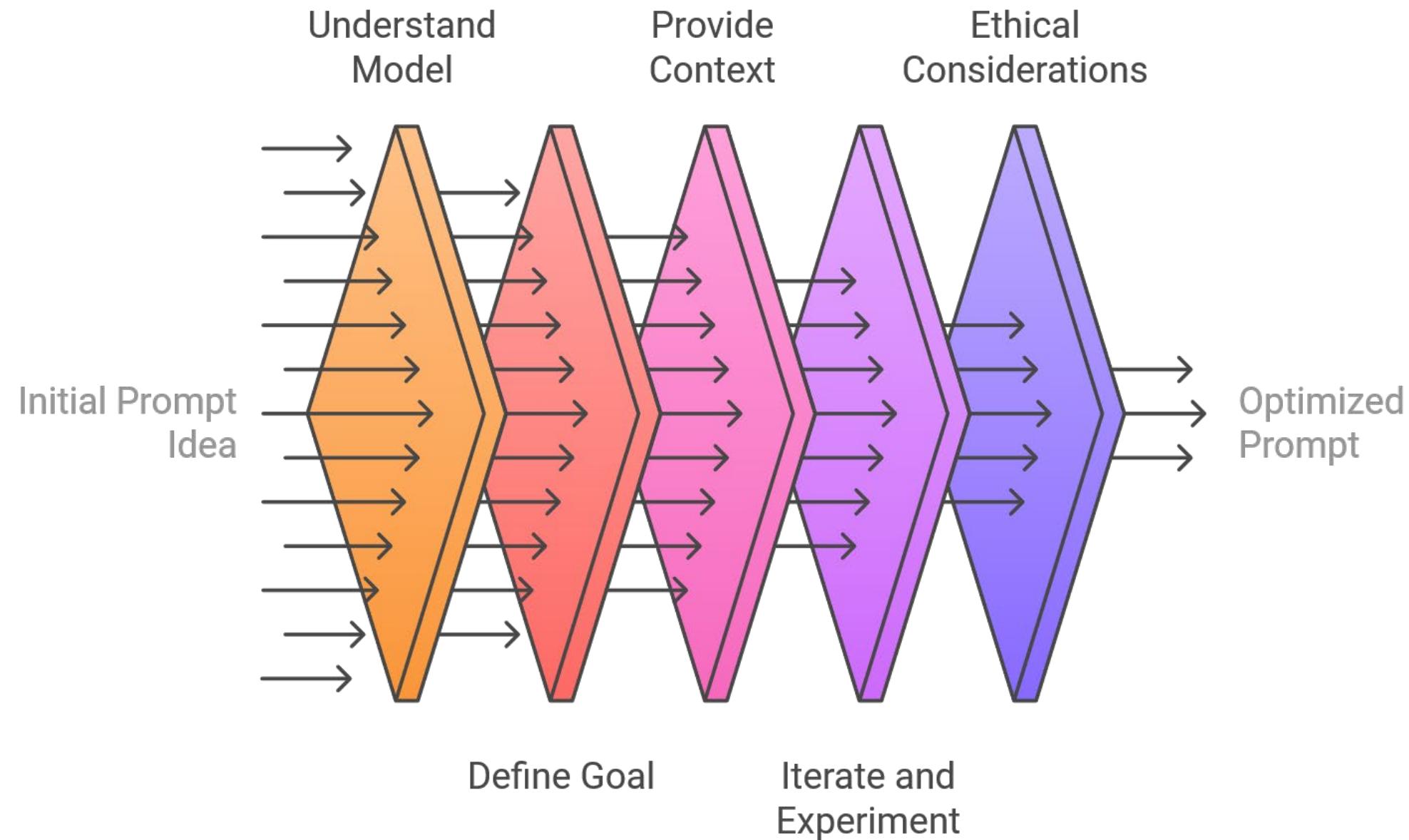


Fundamental Principles





Fundamentals (Technique-agnostic)





Structure Prompts



❖ Include distinctions: Clearly differentiate between

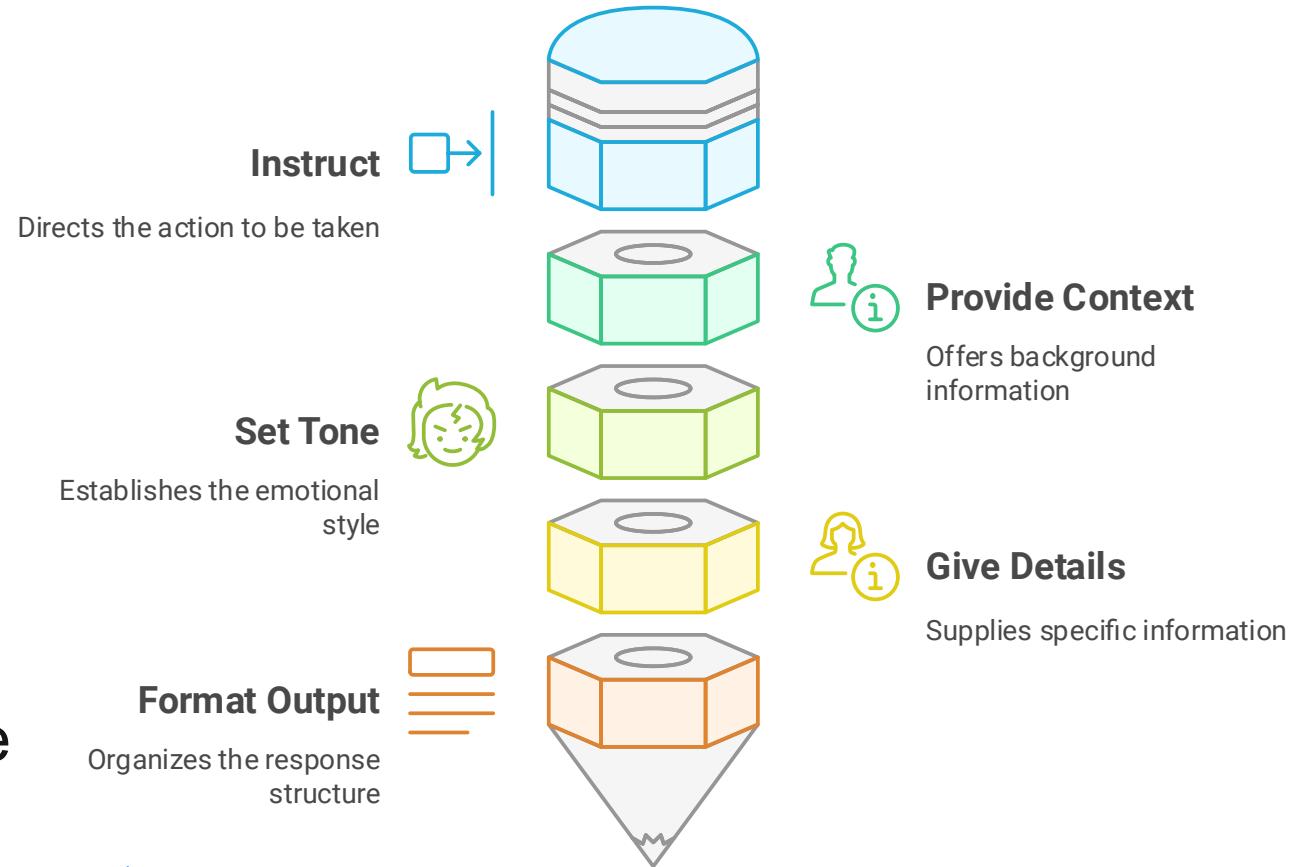
- Instruct
- Context
- Tone
- Detail
- Format

❖ Straightforward example:

- Translate "Hello, world!" into French.

❖ More complex examples require more explanation:

- Compare Tesla Model Y and BMW iX2. Present your answer as a table using metric units and including key attributes like price (in SGD), range (km), and acceleration (0–100 km/h).





Specify Output Format

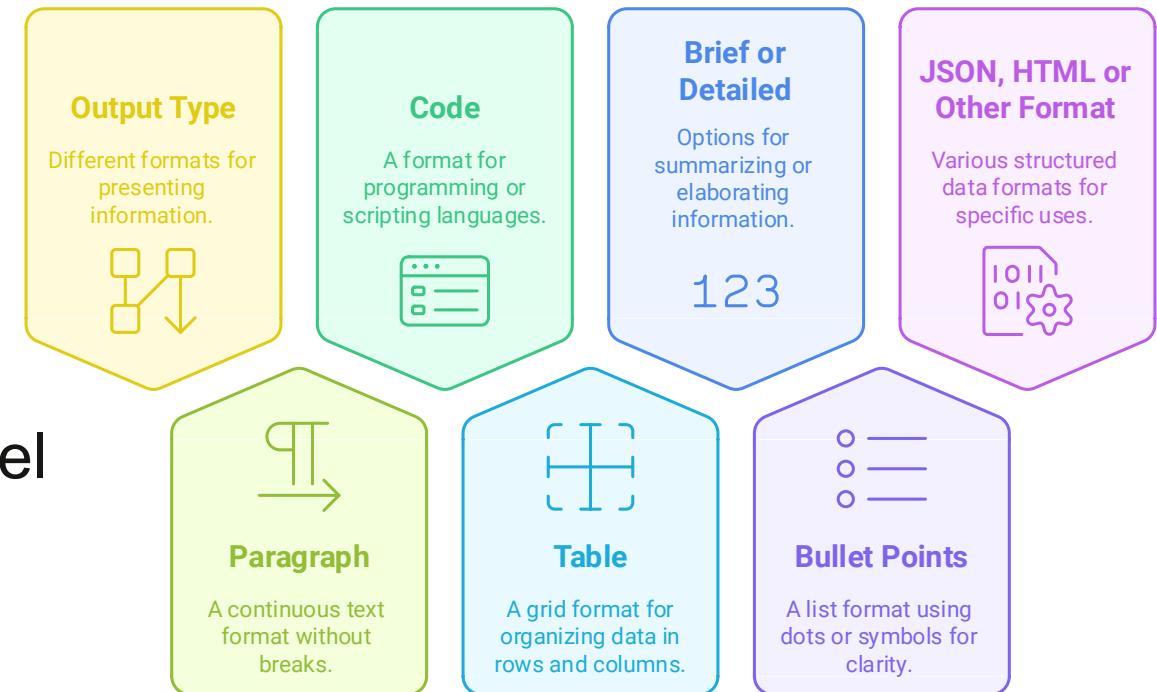


❖ Clearly specify the expected output format:

- Paragraph
- Code
- Table
- Brief or Detailed
- Bullet points or enumerated lists
- JSON, HTML or Other Format

❖ Clearly stating the format helps the model generate the desired results.

❖ Example, "Give me a list of the top 5 tourist attractions in Paris."





Evaluate



- ❖ **Define Evaluation Criteria:** Consider clarity, correctness, completeness, and format adherence. → What aspects of the response are most important to evaluate?
- ❖ **Confirm Intent Alignment:** Check if the response followed instructions, answered the question, and stayed relevant. → Did the response understand and address the core request?
- ❖ **Assess Logic and Accuracy:** Examine the reasoning and factual correctness. → Is the response logically sound and factually accurate?





Evaluate



- ❖ **Check Technical Details:** Verify consistency and correctness of numeric and technical data. → Are the technical details presented accurate and consistent within the response?
- ❖ **Determine Scrutiny Level:** Decide on the depth of evaluation needed based on query complexity. → Does this query require a quick check or a more in-depth evaluation?





Experiment

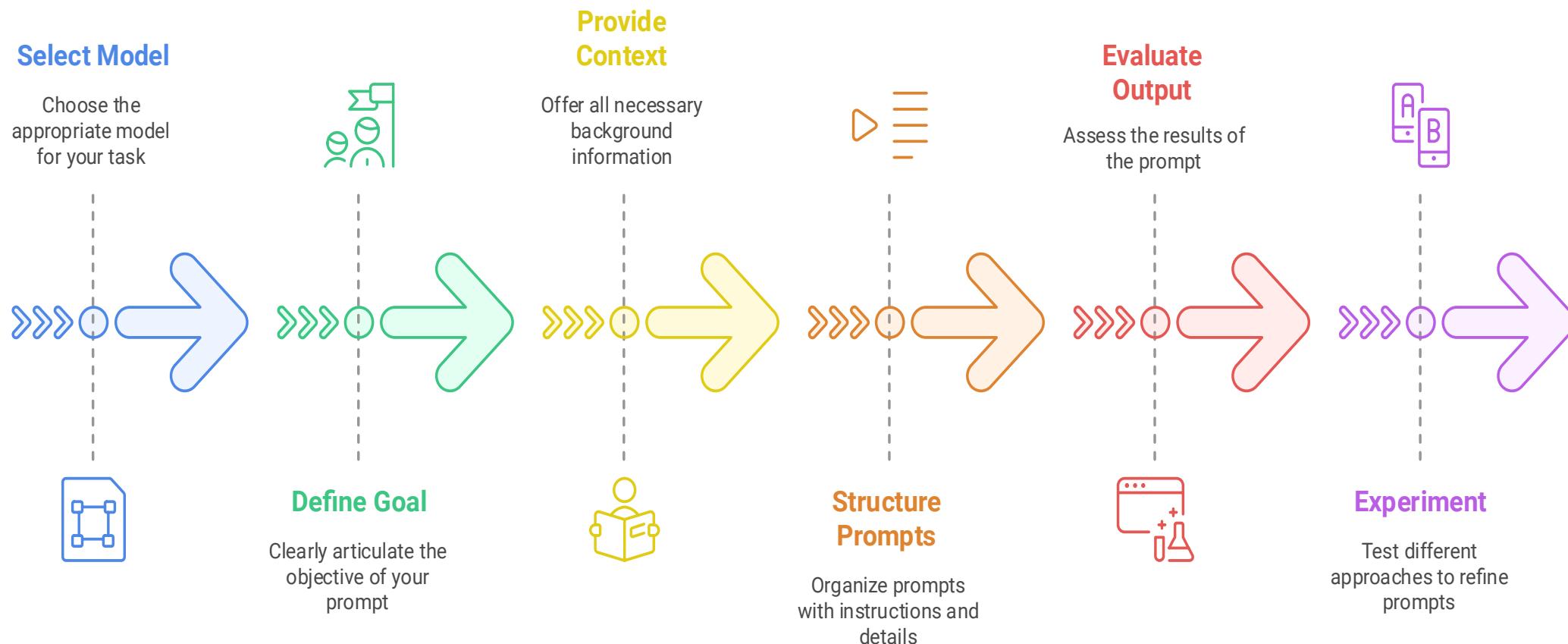


- ❖ Experimentation: An **observational and iterative** approach to Prompt Engineering is absolutely necessary.
- ❖ Evidence-based Process: Success relies on real-world **testing and iteration** and evaluation.
- ❖ Diverse Approaches: Experiment with **variations** in wording, format, and context.
- ❖ Continuous Learning: **Refine** your prompts based on feedback and observed results.
- ❖ Adaptability: **Tailor** your prompts to different tasks and evolving requirements.



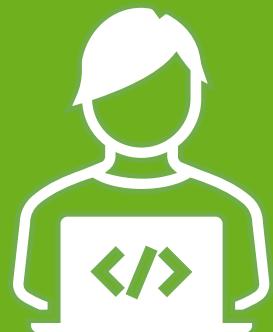


Fundamentals (Technique-agnostic)





Prompt Engineering Techniques 1





Prompt Engineering Techniques (Part 1)

❖ Zero-shot

- Your prompt contains no context and no examples.
- The model infers intent from the prompt alone.
- Works well with straightforward queries, where minimal guidance is required.

❖ Examples:

- Explain how photosynthesis works.
- Translate "Goodbye cruel world" into French.
- Who was the President of the USA on 21 January 2025?

?

Will such queries always work reliably?

❖ Side-effects:

- Vague or incorrect responses
- Relies almost completely on the model's internal knowledge
- Consistency can vary significantly.
- The model has no guidance for internal reasoning.

Prompt Engineering Techniques (Part 1)



❖ One-shot

- Providing an example can help shape output content and format.
- Useful when a **single** illustration can clarify style and/or format of output.
- Enables models to understand and perform accurately when extensive context is not required.
- One-shot prompting is a form of **few shot** prompting, but with a **single** example.



❖ Examples:

- **Determine the sentiment** in this review. Example: "Exceeded my expectations" -> Positive
- **Correct the grammar** in this article. Example: "She don't like apples." → "She doesn't like apples."
- **Classify this article** as news, opinion or ad. Example: "The stock market reached an all-time high today." → News

❖ Side-effects:

- Model may struggle with complex tasks requiring deeper understanding or extensive context.
- A poorly written prompt or example may lead to inaccurate or suboptimal output.
- Overfitting is a possibility.



Prompt Engineering Techniques (Part 1)

❖ Few-shot

- Offers multiple examples to guide the response
- Improves accuracy and consistency for more complex tasks
- The model learns style, tone, and format from these examples



❖ Examples:

- Here are three examples of product descriptions:
 - Example: Lenovo ...
 - Acer ...
 - Asus...
 - Write a similar product description for a MacBook Pro M4.
- Classify the following as 'spam' or 'not spam'.
 - Example: "You've won a free vacation. Click here to claim your prize." Classification: **Spam**
 - Text: "Meeting tomorrow at 10 AM in the conference room." Classification: **Not Spam**
 - Text: "Limited-time offer! Get 50% off on all products." Classification: **Spam**
 - To Classify: "Your package has been shipped and will arrive by Monday."



Prompt Engineering Techniques (Part 1)

❖ Few-shot (cont'd) - Side-effects

- **Example Sensitivity:** The model's performance heavily depends on the quality and selection of examples provided
- **Complex Reasoning:** As task/example complexity increases, few-shot prompting may still struggle to with deep reasoning as models may not grasp intricate relationships expressed in the examples
- **Overfitting Risk:** Providing too many examples can lead to overfitting



Prompt Engineering Techniques (Part 1)

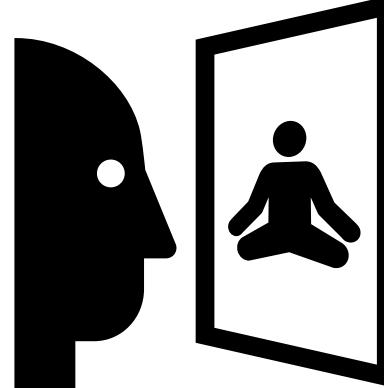


❖ Self-Critique (or Self-Evaluation)

- **Self-Critique** is a prompting technique where an AI model evaluates its own responses to identify errors, inconsistencies, or areas for improvement, and then refines its output accordingly.

❖ Steps

- **Initial Response Generation:** The AI provides an initial answer to a user's prompt
- **Self-Critique:** The AI reviews its own response, assessing accuracy, relevance, coherence and completeness
- **Issue Identification:** The AI identifies any errors, gaps, omissions or areas that require enhancement
- **Revision:** Based on its own evaluation, the AI revises its original response
- **Output:** The improved, refined response is presented to the user



❖ Reasoning Models

- It is important to understand that reasoning models, like OpenAI o1 and DeepSeek R1 employ chain-of-thought (CoT) reasoning and self-reflection. These techniques are a form of self-critique.



Prompt Engineering Techniques (Part 1)

❖ Self-Critique (or Self-Evaluation) cont'd

❖ Examples

- Explain the process of photosynthesis. Then, review your explanation for accuracy and completeness, and provide any necessary improvements.
- Detail the process of nuclear fusion. Afterwards, evaluate your response for its thoroughness and precision, and suggest improvements where needed.
- Write a function in Python that sorts a list of integers. Then, critique your code for efficiency and readability, and suggest enhancements.

❖ Side-effects

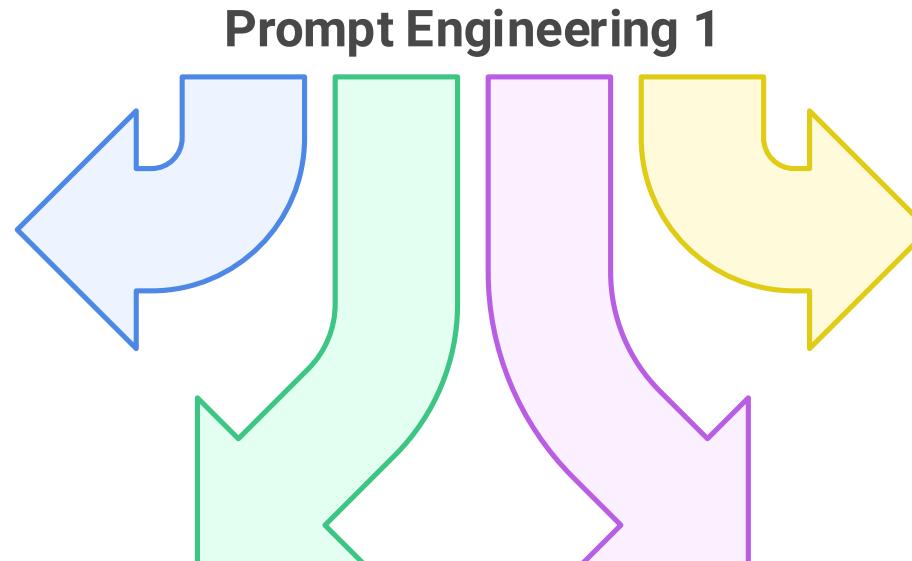
- Using self-critique prompts may produce
 - shallow or generic self-critiques,
 - self contradictions
 - repetitive or off-target “improvements”



Summary



Zero-shot
No prior examples. Model infers from the prompt intent.
Ok for straightforward queries.



Few-shot
Multiple examples provided.
Improvement in accuracy and consistency for complex tasks. Overfitting risk.

One-shot
Provide one example. The example can shape output and format. Ok when extensive context is not required.

Self-critique

Encourages iterative improvement through self-assessment.



Evaluating Prompt Output

Activity



Activity



- ❖ Open the document called Prompt Creation Activity
- ❖ Your team number determines your area of investigation:
 - Sentiment analysis
 - Summarisation
 - Translation
 - Classification
 - Change of Tone
- ❖ Follow the timing given in the document
- ❖ Present findings



Discussion Questions

- ❖ How did the zero-shot, one-shot, and few-shot prompts compare in terms of clarity and detail?
- ❖ Which type of prompt produced the most accurate or creative response?
- ❖ Did adding self-critique change the quality of the output? In what way?
- ❖ What surprised you the most about the model's responses?
- ❖ How might you refine your prompts in the future for better results?
- ❖ After ranking each prompt and output from 0–5 and calculating the correlation across all rubric categories, what did you learn about how prompt quality affects output quality?

AI



Keeping track

Activity





Activity

- ❖ Open the document called Note Taking Activity.
- ❖ Follow the timing
- ❖ Complete the activity.

Ai



Lesson 02

Summary of Material Covered

Lesson 02 Review



- ❖ Overview of Prompt Engineering 1:
 - **Tokens** – Basic units of text (words, sub-words, punctuation) used by AI.
 - **Embeddings** – Numeric representations capturing word relationships.
 - **Training** – Teaching AI models using large datasets.
 - **Overfitting** – Model learns noise instead of patterns, reducing generalization.
 - **Inference** – AI generates text based on learned knowledge.
 - **Context Window** – Limit on the number of tokens AI can process at once.
 - **Temperature** – Controls randomness in AI responses (low = precise, high = creative).
- ❖ Structuring Prompts
 - **Instruction:** Tell the AI what you want it to do.
 - **Context:** Give the AI background information to better understand your request.
 - **Tone:** Specify the style or attitude you want the AI to adopt in its response.
 - **Detail:** Include specific points or elements you want the AI to address.
 - **Format:** Define the structure or layout for the AI's response.



Lesson 02 Review

❖ Prompting techniques:

- **Zero-shot** – No examples, AI infers intent
- **One-shot** – Single example for guidance (version of few-shot)
- **Few-shot** – Multiple examples improve accuracy
- **Self-Critique** – AI evaluates and refines its own output

Ai



To Do (Homework)

- ❖ **Clear a minimum of 10Gb** of hard disk space on your computer. If you need to visit the IT helpdesk to help you clear, do so asap.
- ❖ Follow the instructions in the **LM Studio Install** document which is in the **EdTech Resources** folder.

- ❖ **Please Note: Downloading large language model files over the RP network is very slow. Please do this task at home.**

Ai

Thank you

School of Infocomm

C240 AI Essentials and Innovations

© Republic Polytechnic 2025: All Rights Reserved

Google AI Studio: Crafting Smarter Outputs with Style

**Prompt, tweak, and explore.
Unleash AI, your way!**

The Affordances of Google AI Studio



- ❖ **Prototyping:** Studio is a browser-based platform providing easy access and experimentation.
- ❖ **Model Selection:** You can easily switch between the latest models
- ❖ **Structured Output:** You can guide AI responses using input-output examples.
- ❖ **Temperature:** You can control the randomness of AI output - precision vs. creativity.
- ❖ **Function Calling:** You can extend AI capabilities by integrating functions such as APIs and calculators.
- ❖ **Grounding:** You can ensure the AI responses have access to real-world data.
- ❖ **Collaboration:** You can collaborate and share projects.



Google AI Studio

Catalogue of Terms used in Google AI Studio



Term	Explanation
Gemini	Family of generative AI models available on Google AI Studio, designed for tasks like text generation, image creation, and more.
Prompt	Input provided to the AI model to generate a response, which can be a question, statement, or code.
Model Selection	The process of choosing the appropriate AI model for a specific task based on its capabilities.
Structured Output	A feature that allows users to define the format of the AI's response for consistency and predictability.
Temperature	A setting that controls the randomness of the AI's response; lower values make outputs more deterministic, while higher values introduce variability.

Catalogue of Terms used in Google AI Studio



Term	Explanation
Function Calling	A feature enabling the AI to call external functions or APIs, extending its capabilities beyond text generation.
Grounding	Enhance accuracy by linking your AI model to specific and reliable data sources within AI Studio. It helps the model base its answers on verifiable information.
API Key	Authenticates your access to AI Studio. You need a valid API key to run and deploy AI Studio projects.
Prototype	An initial, functional version of your AI application or project built within Google AI Studio.
Export to Code	The feature in Google AI Studio that lets you generate code (usually in Python) based on your created prototype. This code encapsulates the logic of your prompt, model interactions, and any grounding you've set up.

Catalogue of Terms used in Google AI Studio

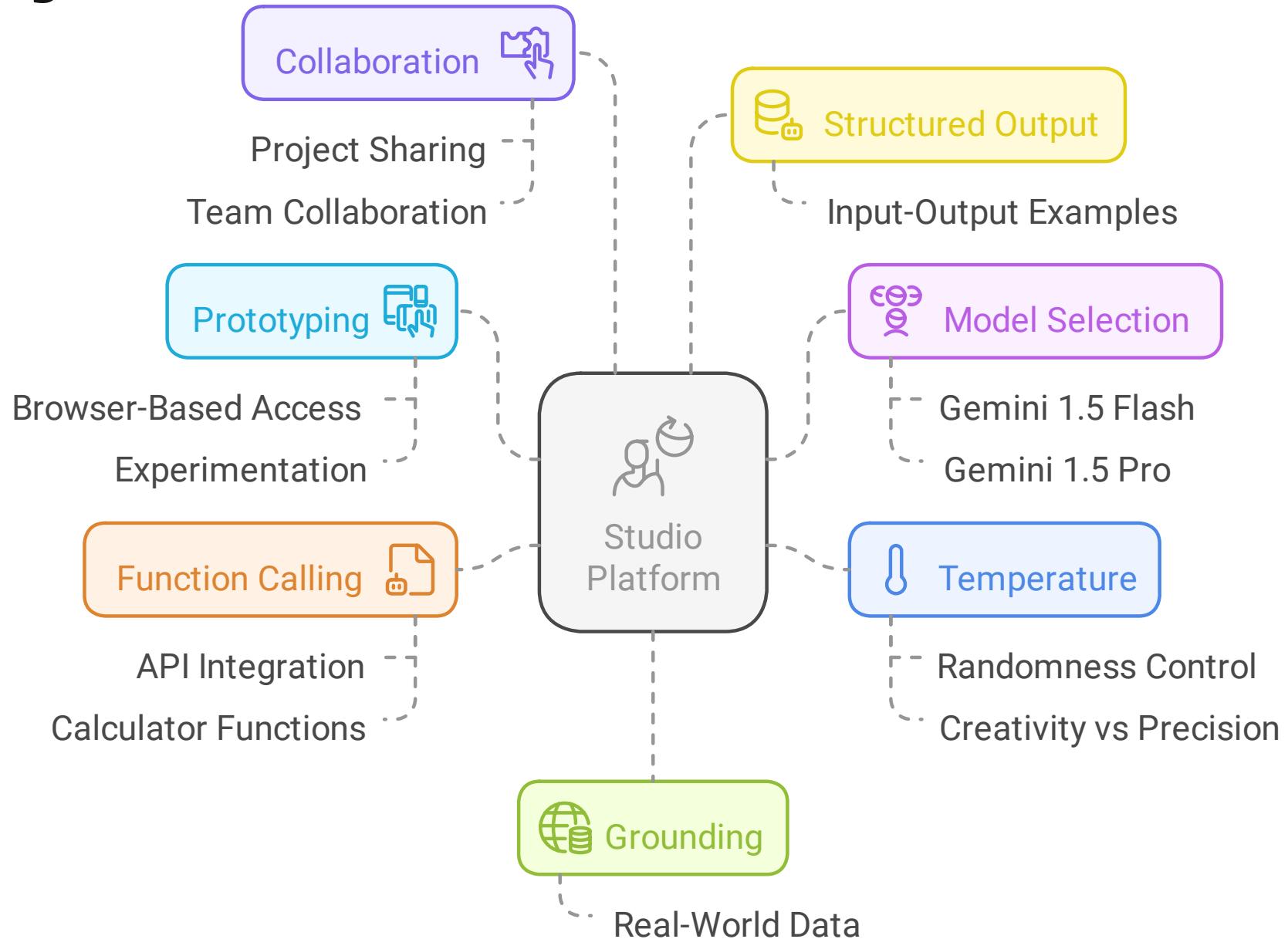


Term	Explanation
Collaboration	Enables multiple users to work together on the same AI Studio project.
Safety Settings	Controls the model's sensitivity to harmful or inappropriate content.
Model Tuning	The process of fine-tuning AI models for specific tasks or datasets to improve performance.
Prompt Template	A pre-defined structure for prompts that can be reused or shared across projects.
System Instructions	Guidelines or rules provided to the AI model to shape its behaviour or responses.



Google AI Studio

Summary



Ai

Activity



Kahoot!



Models and Parameters

Google AI Studio

Model Selection



Model Name	Model Code	Release Date	Launch Stage	Key Features
Gemini 2.5 Pro	gemini-2.5-pro-preview-03-25	April 9, 2025	Public Preview	Advanced reasoning, multimodal input, 1M token context window
Gemini 2.5 Flash	gemini-2.5-flash-preview-04-17	April 17, 2025	Public Preview	Hybrid reasoning with "thinking budget," optimized for speed and cost
Gemini 2.0 Flash	gemini-2.0-flash-001	February 5, 2025	General Availability	Multimodal input, text output, real-time streaming
Gemini 2.0 Flash-Lite	gemini-2.0-flash-lite-001	February 25, 2025	General Availability	Cost-efficient, low latency, text output
Gemini 2.0 Flash Thinking	gemini-2.0-flash-thinking-exp-01-21	January 21, 2025	Experimental	Enhanced reasoning capabilities

Note: Google creates, updates and retires models on a constant basis. This list is indicative as of **May 1, 2025**

Experiment with Models



Which answer is better? /



Gemini 1.5 Flash



System Instructions

Optional tone and style instructions for the model

A man is found dead in a flat in Yishun. He's wet, barefoot, and there's a puddle near the window. Nothing is broken. Police rule it a murder. Why?

Give a brief, logical explanation — no guessing.

The puddle suggests he was thrown from a higher floor. The wet clothes and bare feet are consistent with this. Since nothing is broken, suggesting no forced entry or struggle inside, this points to a murder committed outside the flat.



Gemini 2.5 Pro Preview 03-25



System Instructions

Optional tone and style instructions for the model

A man is found dead in a flat in Yishun. He's wet, barefoot, and there's a puddle near the window. Nothing is broken. Police rule it a murder. Why?

Give a brief, logical explanation — no guessing.

Thoughts (experimental)

Auto

Expand to view model thoughts

Here's the logical explanation:

The man was likely stabbed or killed with a large piece of ice (like an icicle or a specifically shaped block). The ice served as the murder weapon. After the murder, the ice melted, leaving the victim wet, creating the puddle near the window (perhaps where the body lay or the weapon was dropped/placed), and leaving no weapon behind. The lack of a conventional weapon, combined with the water evidence, points to a deliberate method intended to conceal the weapon, hence the ruling of murder.





Parameter Tuning: Temperature

In the previous example of CoE price prediction, how should the temperature be set?

- ❖ **Adjust "Temperature" (0-2):** Lower temperature (near 0) = focused, predictable. Raise temperature (near 2) = highly creative, very varied, potentially less focused.
- ❖ **Temperature 0-1 (Accuracy Focus):** Use 0 - 1 for factual tasks, code, precision, and when you need reliable, consistent outputs. Closer to 0 is even more deterministic.
- ❖ **Temperature 1-2 (Creativity Boost):** Use 1 - 2 for brainstorming, creative writing, diverse ideas, and when you want more unexpected and imaginative results. Be aware outputs may be less coherent or factually accurate at very high temperatures.





Parameter Tuning: Top P

- ❖ **Top P:** (also known as "nucleus sampling") is a parameter that controls how the language model selects the next word in a sequence. Think of it to fine-tune the balance between predictability and randomness.
- ❖ **How Top P Affects Output:**
 - **Low top p (e.g., 0.3):** The model focuses on the most likely words. Output is more predictable, less creative, and potentially repetitive. Good for tasks requiring factual accuracy or conciseness.
 - **High top p (e.g., 0.95):** The model considers a wider range of words, including less likely ones. Output is more creative, diverse, and potentially surprising. Good for brainstorming or creative writing but might be less coherent or accurate.
- ❖ **Imagine choosing toppings for a pizza.**
 - **Low Top P:** You pick from the most common toppings (pepperoni, mushrooms). Safe and predictable.
 - **High Top P:** You consider a wider range, including unusual toppings (durian, pineapple). More adventurous, but potentially less pleasing to everyone.





Inference Parameter Tuning

Activity





Activity 1: Inference Parameter Tuning

- ❖ Open Google Ai Studio and login.
- ❖ For this activity, use the Gemini 2.0 Pro Experimental model.
- ❖ As a team create a simple, but **unique** prompt.
- ❖ **Change the temperature only**, and observe the results.
- ❖ **Change the Top P only**, and observe the results.
- ❖ Complete the **Inference Parameter Tuning Activity** table. Refer to the separate **Activities** document.

Ai



Structured Output

Google AI Studio



Structured Output

❖ Structured output enables you to generate data in organised formats like JSON, tables, and code, that are immediately usable in your applications and workflows.

- **Automation:** Automate data processing, analysis, and integration.
- **Efficiency:** Skip manual formatting and data wrangling.
- **Precision:** Get data in the exact format you need.
- **Integration:** Seamlessly connect with other tools and APIs.

❖ Requirements for using Structured Output

- **Clear Prompts:** Be very specific about the desired format and the data required.
Examples: JSON, Table, Markdown, HTML.
- **Experiment:** Change the Temperature and Top P. Lower values usually reduce errors.
- **Validate Output:** Use a JSON validator or test code

Structured Output: Sample



Prompt: *"Generate a JSON object for books, including title, author, ISBN, publisher. I need 3 books. They need not be actual references."*

```
[  
 {  
   "title": "The Whispering Mountains",  
   "author": "Anya Petrova",  
   "ISBN": "978-0321765723",  
   "publisher": "Mystic Press"  
 },  
 {  
   "title": "The Clockwork Heart",  
   "author": "Jasper Finch",  
   "ISBN": "978-1473228930",  
   "publisher": "Geargrind Publishing"  
 },  
 {  
   "title": "Beneath the Crimson Sky",  
   "author": "Elara Vance",  
   "ISBN": "978-0997151742",  
   "publisher": "Phoenixfire Books"  
 }]  
 ]
```



Structured Output: Sample SQL

Prompt: *"Generate a MySQL script to create two tables, students and modules. The script must be runnable MySQL Data Definition Language (DDL). Give me two students, each with 2 modules as sample data."*

```
-- Create the students table
CREATE TABLE students (
    student_id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(255) NOT NULL,
    last_name VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE
);

-- Create the modules table
CREATE TABLE modules (
    module_id INT AUTO_INCREMENT PRIMARY KEY,
    module_name VARCHAR(255) NOT NULL,
    credits INT
);

-- Create a junction table to represent the many-to-many relationship
CREATE TABLE student_modules (
    student_id INT,
    module_id INT,
    FOREIGN KEY (student_id) REFERENCES students(student_id),
    FOREIGN KEY (module_id) REFERENCES modules(module_id),
    PRIMARY KEY (student_id, module_id) -- Composite key to prevent duplicates
);

-- Insert sample student data
INSERT INTO students (first_name, last_name, email) VALUES
('Alice', 'Smith', 'alice.smith@example.com'),
('Bob', 'Johnson', 'bob.johnson@example.com');

-- Insert sample module data
INSERT INTO modules (module_name, credits) VALUES
('Introduction to Programming', 10),
('Calculus I', 5),
('Linear Algebra', 5),
('Data Structures and Algorithms', 10);
```



Structured Output

Activity

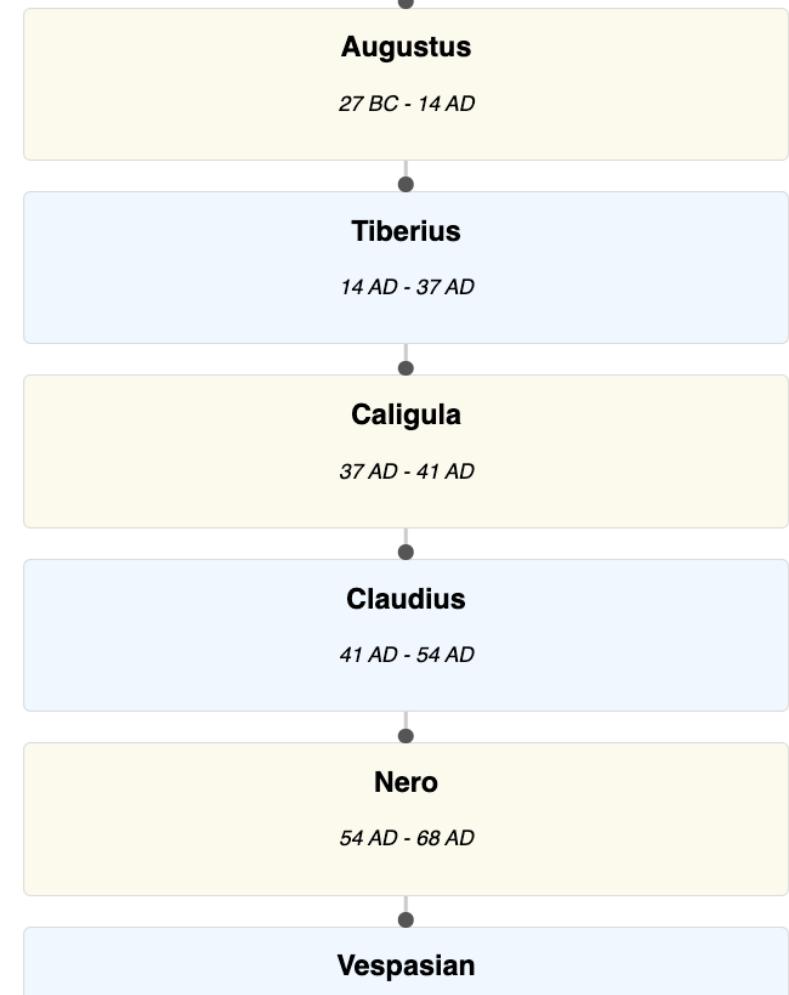




Activity 2:

- 1) Create a prompt that generates normal paragraphed text. Modify the prompt to output JSON structured data.
- 2) Create a prompt to search for 2025 movie releases together with the movie title, director and expected release date. The prompt should create CSV output that can be saved to a CSV file using VSCode.
- 3) Create a prompt that will generate a HTML page with a timeline of Roman Emperors and the time of their rule. The HTML needs to be somewhat creative and use different colours. Do not modify the HTML directly but manipulate the output through prompts. Create, modify, copy to VSCode and run. Paste your result in the Activities document.

Roman Empire Timeline



Ai



Activity 3: Bonus

- ❖ Create a prompt to generate the following worksheets for Grade 5 Chinese classes.

❖ Worksheet 1: Simple Sentences

1. 今天天气很好。 (Jīntiān tiānqì hěn hǎo.) - Today the weather is very good.
2. 我有一只猫。 (Wǒ yǒu yī zhī māo.) - I have a cat.
3. 学校在公园旁边。 (Xuéxiào zài gōngyuán pángbiān.) - The school is next to the park.
4. 我喜欢吃苹果。 (Wǒ xǐhuan chī píngguǒ.) - I like to eat apples.
5. 他每天都上学。 (Tā měitiān dōu shàngxué.) - He goes to school every day.

❖ Worksheet 2: Slightly More Complex Sentences

1. 我的朋友喜欢玩电脑游戏。 (Wǒ de péngyou xǐhuan wán diànnǎo yóuxì.) - My friend likes to play computer games.
2. 放学后，我们一起回家。 (Fàngxué hòu, wǒmen yìqǐ huí jiā.) - After school, we go home together.
3. 昨天晚上，我看了一部电影。 (Zuótiān wǎnshàng, wǒ kàn le yī bù diànyǐng.) - Last night, I watched a movie.
4. 这本故事书很有趣。 (Zhè běn gùshì shū hěn yǒuqù.) - This storybook is very interesting.
5. 下个星期，我们要去旅行。 (Xià ge xīngqí, wǒmen yào qù lǚxíng.) - Next week, we are going on a trip.



Prompt Gallery

Google AI Studio



Prompt Gallery

- ❖ Discover, create, and innovate with the Gemini Prompt Gallery in Google AI Studio. This resource offers:
 - **Explore diverse applications:** From simple questions to complex code, discover the breadth of Gemini's capabilities.
 - **Spark creativity and innovation:** Find inspiration for new and exciting AI projects.
 - **Overcome the blank page:** Get started quickly with pre-built prompts that you can adapt and customize.
 - **Learn by example:** See how others are using Gemini and unlock best practices for prompting.
 - **Stay up-to-date:** The gallery is continuously updated with the latest and greatest prompt examples.



Prompt Gallery

Activity





Activity 4: Prompt Gallery Jumpstart

- ❖ **1. Browse (2 min):** Open the Gemini Prompt Gallery in Google AI Studio and explore the examples. What catches your eye?
- ❖ **2. Adapt (2 min):** Pick one prompt and create 2-3 variations. Change the input, output, or add constraints.
- ❖ **3. Ideate (1 min):** How could you use this prompt type in your own projects? Jot down 1-2 ideas.

Ai



Lesson 05 Additional Resources

- ❖ **Google AI Studio Documentation:** The official documentation is a comprehensive resource covering everything from setup to advanced features.
<https://cloud.google.com/vertex-ai/docs/workbench/introduction>
- ❖ **Google AI Studio Tutorials:** Practical, hands-on tutorials guide you through specific tasks and workflows. <https://developers.google.com/machine-learning/crash-course/vertex-ai-platform> (This link focuses on Vertex AI, but includes AI Platform Notebooks, which became Vertex AI Workbench User-Managed Notebooks, and many concepts overlap.)
- ❖ **Google Machine Learning Crash Course:** A beginner-friendly introduction to machine learning concepts with interactive exercises and coding examples (many usable in Colab which has a similar interface to AI Studio). <https://developers.google.com/machine-learning/crash-course>
- ❖ **YouTube Channels:** Search YouTube for channels dedicated to Google Cloud Platform, Vertex AI, or machine learning. Many creators offer helpful tutorials and walkthroughs specifically for AI Studio. (Search terms like "Google AI Studio Tutorial" will provide relevant results)



Lesson 05

Summary of Material Covered



Lesson 05 Review

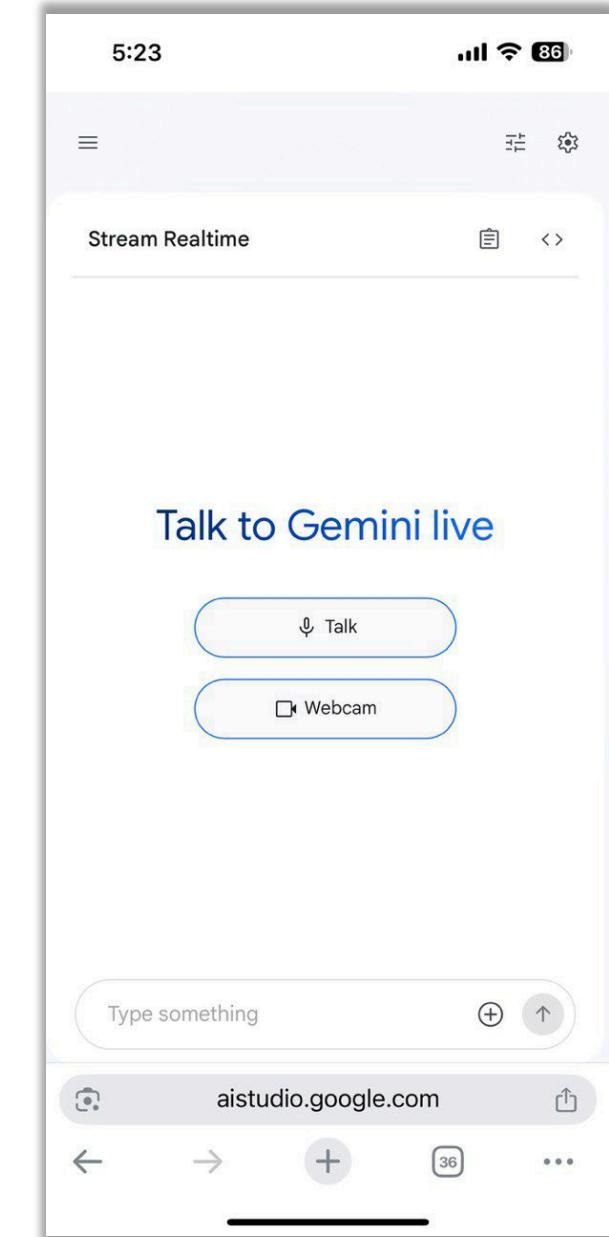
- ❖ Key Takeaways: Working with Generative AI in Google AI Studio
 - AI Studio Environment: Overview of the interface and capabilities.
 - AI Models: Model selection and suitability for various tasks.
 - Prompt Engineering: Crafting effective prompts and using the Prompt Gallery.
 - Parameter Tuning: Controlling output with temperature, top_p, and grounding.
 - Structured Output: Generating JSON, tables, and code.



Homework – Optional but fun!

Ai

- ❖ Watch: [Change How You See the World](#)
- ❖ On your phone (**iPhone or Android**) use Chrome to browse to the following web site <https://aistudio.google.com>
- ❖ Leave the Gemini 2.0 Flash model selected
- ❖ From the hamburger menu on the top left, click on **Stream**
- ❖ Turn on your **microphone** and **camera**
- ❖ You now have **10 minutes** to explore your surroundings with Google AI Studio



Thank you

School of Infocomm

C240 AI Essentials and Innovations

© Republic Polytechnic 2025: All Rights Reserved



LM Studio Mastery: Building Brilliant Prompts with Precision

**Chain, refine, and innovate.
Master AI, your strategy!**

The Affordances of LM Studio

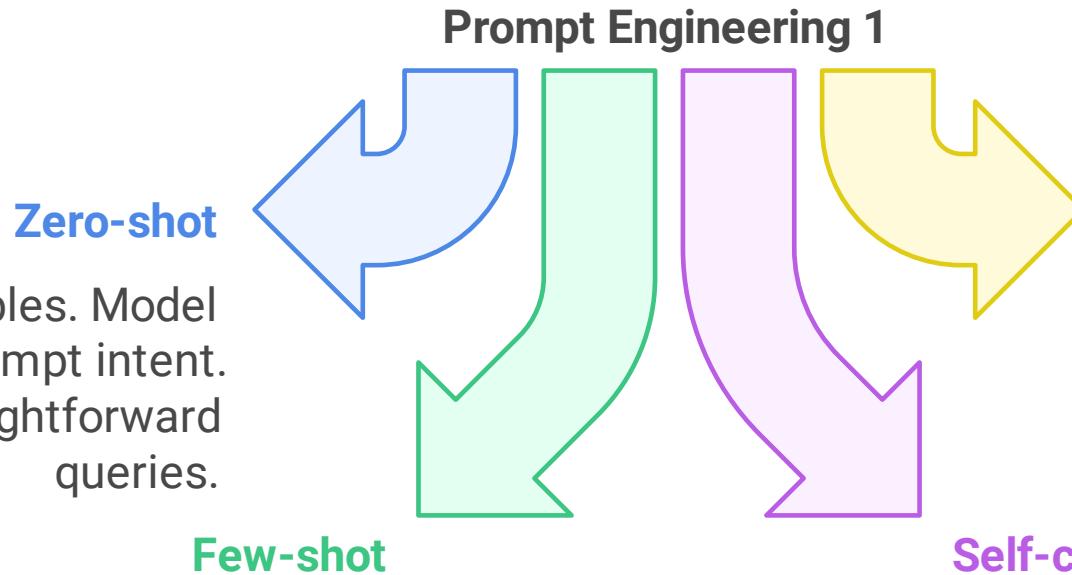


- ❖ **Run Local:** Run open-source LLMs locally on desktops/laptops, offline.
- ❖ **Familiar Interface:** Offer a user-friendly ChatGPT-like interface for interaction.
- ❖ **Integrations:** Download and load models from Hugging Face in formats like GGUF.
- ❖ **Customizations:** Customize model outputs with adjustable settings for laptop hardware.
- ❖ **Refine:** Support chat features for testing and refining prompts.



Review Techniques

Previously in C240 ...



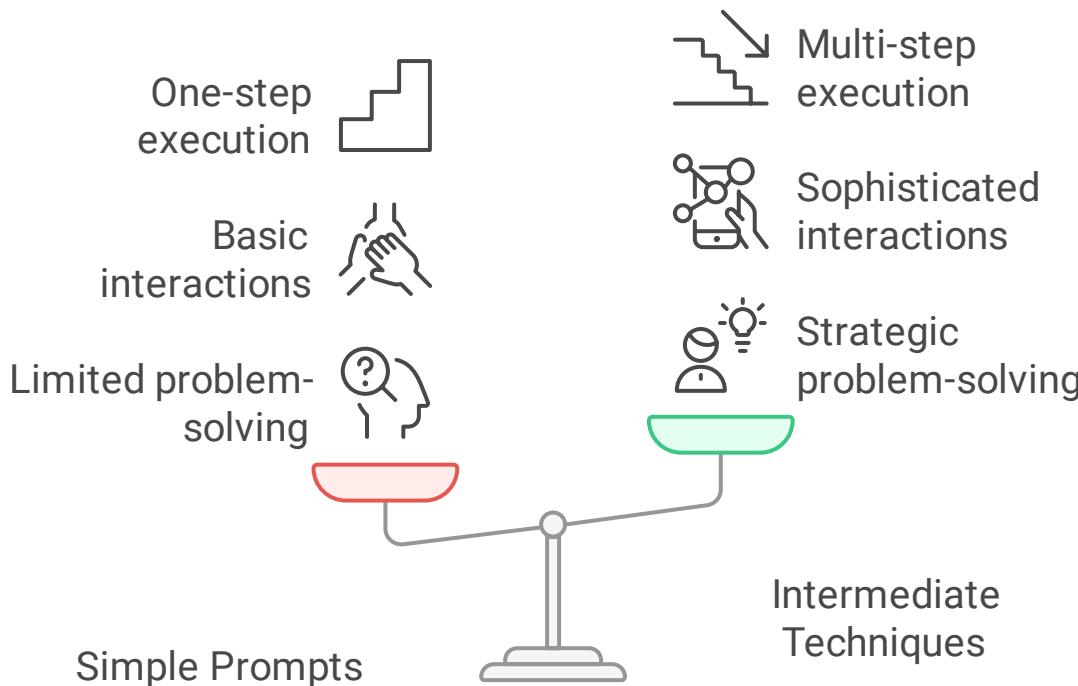
Zero-shot
No prior examples. Model infers from the prompt intent.
Ok for straightforward queries.

Few-shot
Multiple examples provided.
Improvement in accuracy and consistency for complex tasks. Overfitting risk.

One-shot
Provide one example. The example can shape output and format. Ok when extensive context is not required.

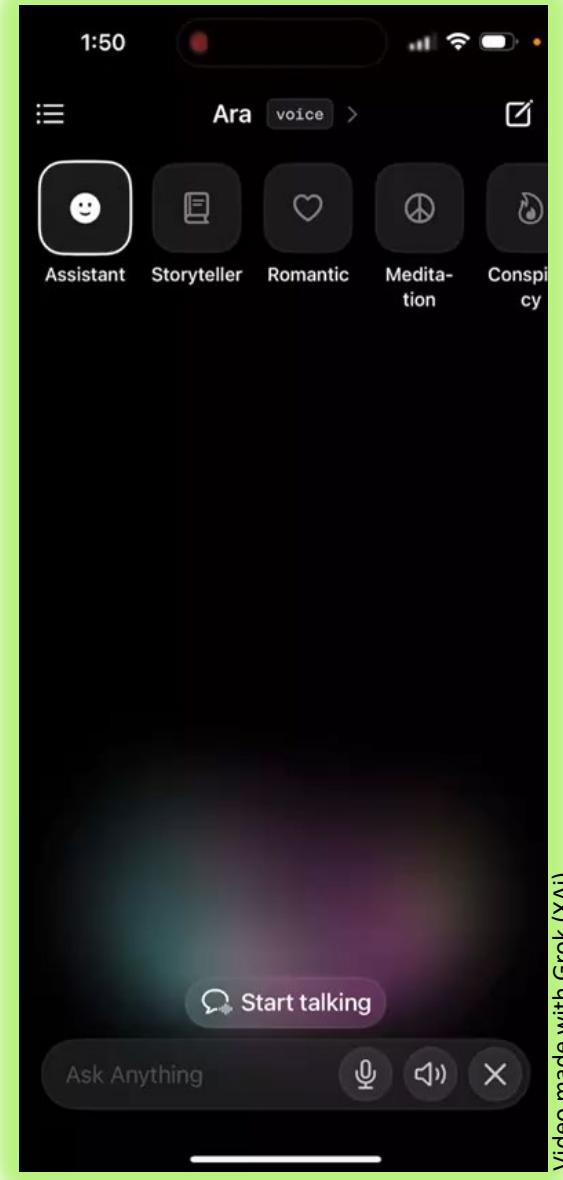
Self-critique
Encourages iterative improvement through self-assessment.

Beyond Basics: Intermediate Prompt Engineering



- ❖ With intermediate prompt engineering techniques, we move from simple prompts to more **strategic conversations** with LLMs.
- ❖ We've covered the basics. Now it's time to level up! In this lesson we will explore techniques that go beyond simple prompts and delve into crafting more **sophisticated AI interactions**.
- ❖ The Problem: Simple prompts often fall short when tackling **complex, multi-faceted** problems. We often need strategies that guide the AI towards a solution, **step-by-step**.
- ❖ The Solution: Intermediate prompt engineering provides a powerful toolbox for building **multi-step prompting strategies** that break down complex problems into manageable chunks.

Core Concepts: Prompt Chaining



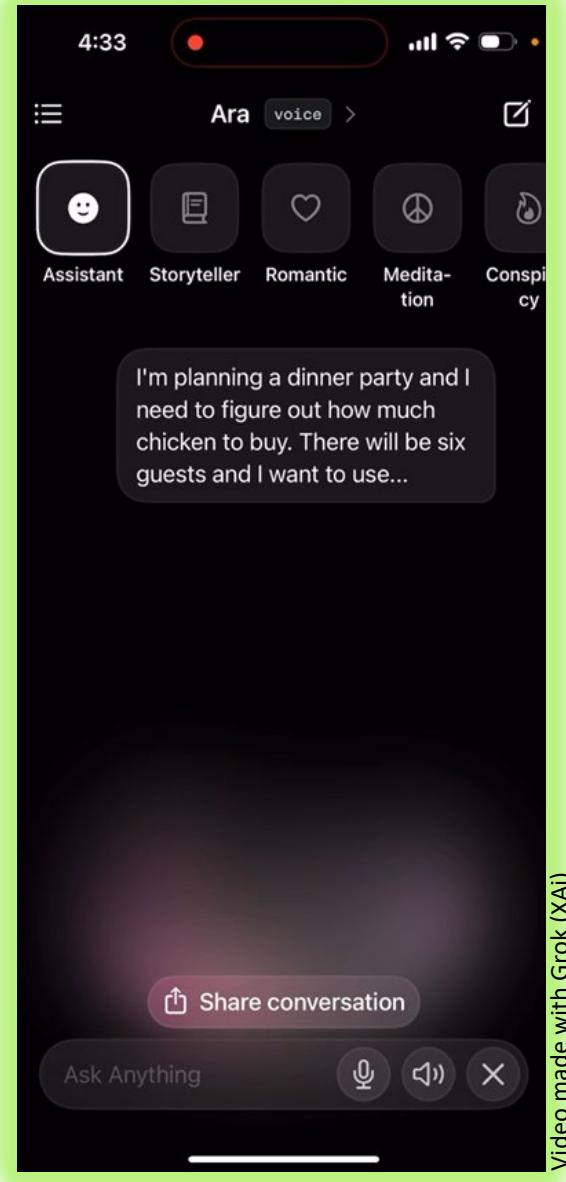
- ❖ Concept: Linking a series of prompts together, where the output of one prompt feeds into the next.
- ❖ Focus: Creating a workflow where each step builds upon the previous one.
- ❖ Benefit: Ideal for tasks needing multiple stages of reasoning or processing, like writing a report or developing a complex plan.

Core Concepts: Contextual Prompting



- ❖ Concept: Providing the AI with relevant background information, examples, or constraints to guide its response.
- ❖ Focus: Grounding the AI in the specific scenario, problem domain, or user needs.
- ❖ Benefit: Improves the accuracy, relevance, and usefulness of AI outputs by preventing assumptions and providing necessary context.

Core Concepts: Chain of Thought (CoT)



Video made with Grok (XAI)

- ❖ Concept: Encouraging the AI to explicitly explain its reasoning process, **step-by-step**, before providing a final answer.
- ❖ Focus: Unlocking the AI's reasoning abilities and creating a more transparent and interpretable thought process.
- ❖ Benefit: Improves accuracy, identifies potential errors in reasoning, and helps users understand why the AI arrived at a particular conclusion.



Explicit Reasoning Models





Explicit Reasoning Models

- ❖ **What is the difference between general language models and explicit reasoning models?**
 - **General Language Models:** These are older models, like GPT-3 or GPT-4o, that *can* reason but usually need specific instructions in the prompt like "think step-by-step" or "show your reasoning" to give you a structured, logical answer.
 - **Explicit Reasoning Models:** These are newer AI models, like Grok 3 and DeepSeek-R1, designed to naturally "think" step-by-step when solving problems. They don't need you to tell them how to reason, they just do it automatically. You can generally read their "thinking".
- ❖ **When did these models first appear?**
 - **General Language Models:** These started popping up around 2020–2023 with models like GPT-3 and GPT-4o. They were impressive for their time but needed a little hand-holding to reason well.
 - **Explicit Reasoning Models:** These are more recent, emerging in late 2024 and early 2025 with models like Grok 3 and DeepSeek-R1, built from the ground up to handle reasoning without extra guidance.

How are Explicit Reasoning Models Different?



❖ **Explicit Reasoning Models:**

These models automatically break down complex problems into steps and reason through them. If you observe the thinking, they're having a little chat with themselves. You don't need to add special instructions in the prompt as they're built to "think" logically on their own.

❖ **General Language Models:**

These models, while capable of reasoning, often lean on pattern recognition unless you nudge them with prompts like "think step-by-step." Without those prompts, their answers might lack structure or depth.

❖ **Why Does it Matter?**

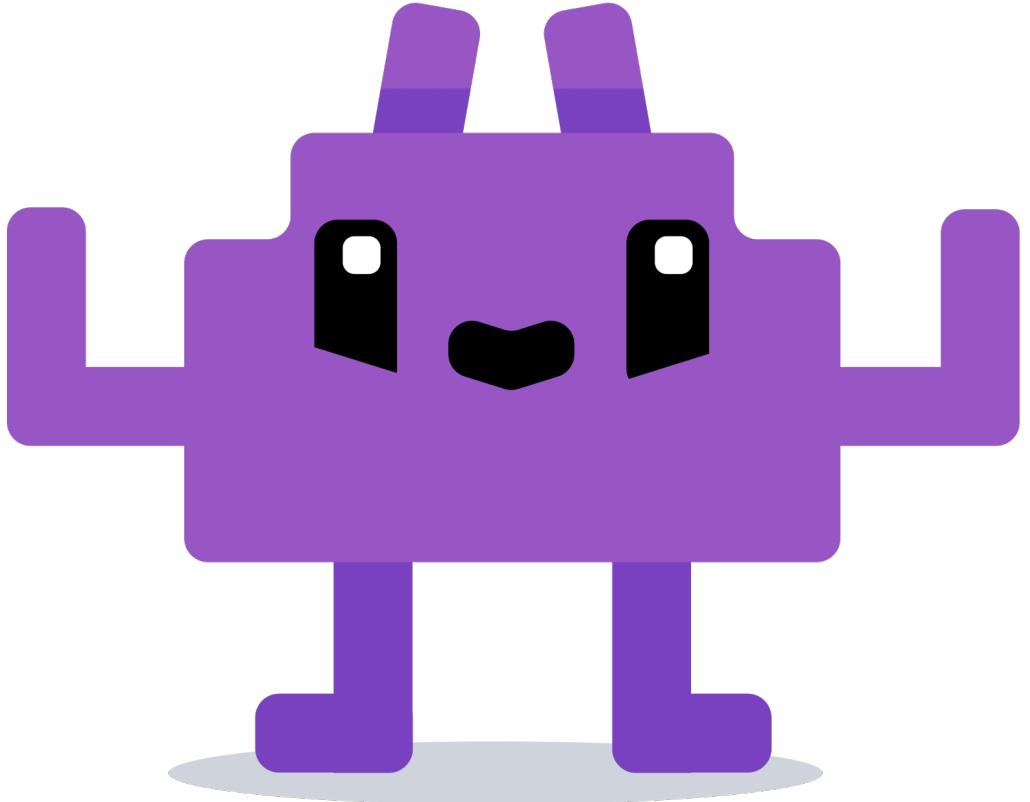
The shift from general language models to explicit reasoning models shows how AI has evolved. With models like GPT-4o, you had to play an active role in guiding their thought process. Now, with Grok 3 or DeepSeek-R1, the models take the lead, reasoning naturally and saving you the effort of crafting detailed prompts. It's like the difference between giving a student a step-by-step worksheet versus them figuring it out independently!



Running Models Locally

LM Studio

LM Studio: Why use a local model?



- ❖ **Internet Dependency:** Online models require an internet connection; local models work offline.
- ❖ **Performance and Speed:** Online models are faster on remote servers but may have latency; whereas local models are slower on laptops but do not suffer any network delays.
- ❖ **Resource Usage:** Online models use no local resources beyond bandwidth; local models use CPU and RAM, potentially slowing other apps.
- ❖ **Privacy and Security:** Online models risk data exposure via APIs; local models keep data private on the device.
- ❖ **Setup and Accessibility:** Online models are easier to access with an API key but limited by availability; local models require setup but offer full control offline.

Activity: Prompt Ninja Challenge



- ❖ Set your expectations. Remember models may be slower to respond when running on low-powered laptops. Have patience - responses may take a while.
- ❖ Open the accompanying **Ninja Challenge** document.
- ❖ Follow the detailed instructions included in the activity document.
- ❖ Don't forget to post to MST.





Shaping AI's Persona and System Design

Changing AI's Behavior – Be the Boss!



Role-Playing (Persona) Prompting

❖ Role-Playing Prompting:

- **Concept:** Assign the AI a specific role or persona. Example: "Act as a software engineer" or "Assume the role of a nutty chemistry professor".
- **Focus:** Guides the AI to adopt a unique viewpoint, tone, or style. Example: "Respond like an over-paid entitled Singaporean interior designer".
- **Benefit:** Tailors the AI's output to match your expertise, needs or user vibe.





System Prompting

❖ Changing the System Prompt:

- **Concept:** Set the stage for all chats by defining the AI's core behavior, capabilities, and limits. You are the boss!
- **Focus:** Establishes clear rules so the AI doesn't go rogue.
- **Benefit:** Keeps responses consistent, reliable and won't annoy you with random tangents.
- **Example:** "Always use metric units and keep your responses concise unless directed otherwise. Where possible convert amounts to Singapore dollars."





Prompt Optimization

❖ Practice makes Perfect:

- **Concept:** Tweak prompt phrasing, keywords, and structure to level up the AI's performance.
- **Focus:** Refine through trial and error, similar to perfecting a precise task, such as optimizing a project plan.
- **Benefit:** Transforms average responses into highly effective outputs, enhancing the precision and impact of your prompts for optimal AI performance.





Control and Steering Techniques

Changing AI's Behavior – Be the Boss!



Control and Steering Techniques

The techniques that follow focus on **guiding** or **constraining** the AI's output to meet specific requirements, avoid undesired responses, or maintain focus.

❖ Negative Prompting:

This technique involves explicitly telling the AI what **not** to include in its response. Example: "Don't use technical jargon" or "Avoid humor". It's a steering mechanism to refine outputs by ruling out unwanted elements.

❖ Constraint-Based Prompting:

This method imposes specific limitations or requirements on the AI's response, such as length, format, or style. Example: "Respond in exactly 100 words" or "Write in haiku form". Like negative prompting, it controls the output but focuses on positive boundaries rather than exclusions.

❖ Contextual Anchoring:

This technique was mentioned earlier. You provide a specific context or role for the AI to operate within, ensuring responses stay relevant and grounded. This can prevent drift or overly generic answers. Example: "You are a historian in 1920s London. Describe the scene" or "Act as a Python expert troubleshooting code."

Activity: Persona, Control & Steering



- ❖ In LM Studio load the **reasoning** model and create a prompt to write an email objection to a new school policy.
Example: No unnatural hair colour in school.
- ❖ Open the accompanying **Persona Control & Steering** activity document.
- ❖ Follow the detailed instructions included in the activity document.
- ❖ Don't forget to post to MST.





One result

Subject: DISASTROUS HAIR POLICY – TOTAL LOSER!

Dear School Board,

Folks, this "no unnatural hair color" policy is a TOTAL DISASTER! The worst I've ever seen, believe me! And I've seen some things. Purple, blue, red hair - they're *beautiful, they're fantastic, they're tremendous!* Totally harmless! Banning them? That's like banning winning, and **I don't do losers** - never have, never will! **I've got the best hair.** Nobody does it better, natural or not - and nobody, I mean NOBODY, tells me what to do with it!

This rule? Weak! Sad! Very sad! Totally un-American! It's fake news, folks! Let the kids shine with their colors - bright, bold, amazing colors! **We're gonna make this school great again.** You hear me? **GREAT AGAIN!** Reverse this nonsense right now, or I'll be very disappointed - and you don't want that, trust me!

Respectfully,
Donald J. Trump

Text & image generated by Grok XAI





Lesson 06

Summary of Material Covered



Lesson 06 Review

❖ LM Studio Overview

- Runs open-source large language models (LLMs) locally on desktops/laptops, offline.
- Features a user-friendly, ChatGPT-like interface for student interaction.
- Integrates with Hugging Face to download models in formats like GGUF.
- Enables customization of model outputs based on student hardware.
- Supports chat features for testing and refining prompts.

❖ Model Types

- **General Language Models:** Older models (e.g., GPT-3, GPT-4o) need explicit reasoning instructions.
- **Explicit Reasoning Models:** Newer models (e.g., Grok 3, DeepSeek-R1) reason step-by-step naturally.

➤ Local and Online Models

- **Local Models:** Offline, uses CPU/RAM, private, requires setup, may be slower on low-end devices.
- **Online Models:** Internet-dependent, faster via servers, risks latency and data exposure.

AI

Lesson 06 Review - Prompting Techniques



Zero Shot

A technique where no examples are provided.

Few Shot

A method using a few examples for guidance.

Self-Critique

Evaluating one's own prompts for improvement.

Prompt Chaining

Linking multiple prompts for complex tasks.

Contextual Anchoring

Using context to enhance prompt relevance.

Chain of Thought (CoT)

Encouraging reasoning through sequential prompts.

Adopting Personas

Using different perspectives in prompts.

System Prompting

Directing the system with specific instructions.

Prompt Optimization

Refining prompts for better performance.

Negative Prompting

Indicating what should not be included in responses.

Constraint-Based Prompting

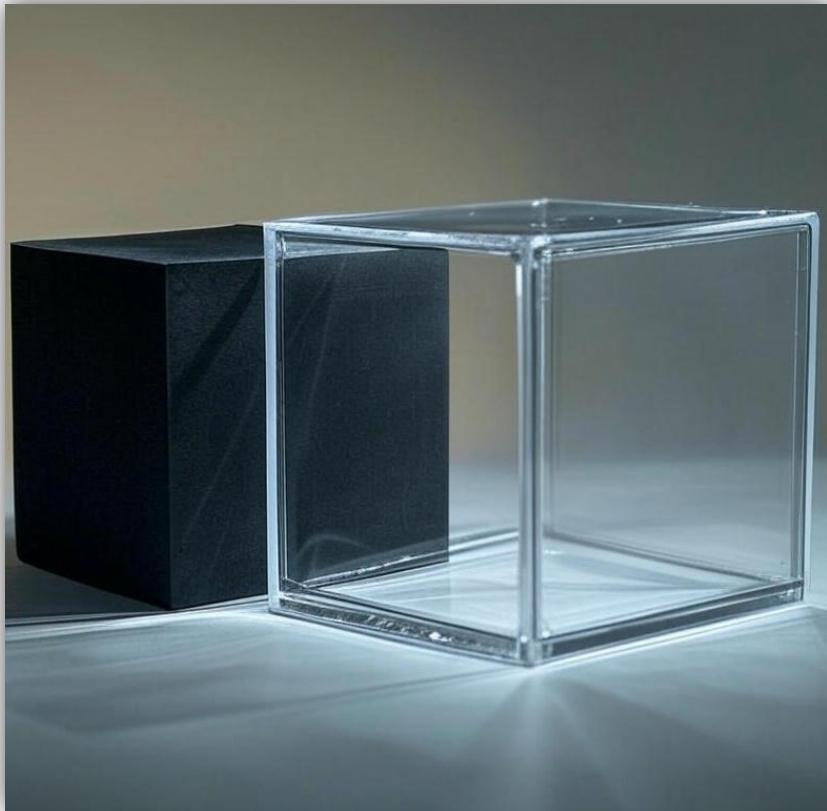
Setting limits to guide responses effectively.

Thank you

School of Infocomm

C240 AI Essentials and Innovations

© Republic Polytechnic 2025: All Rights Reserved



From Black Box to Glass Box: Making AI Transparent

Interact, Analyse, Understand:
Master Explainable AI

Explainable AI (XAI)



- ❖ **What is XAI?:** AI systems made clear and understandable to humans.
- ❖ **From Black Box to Glass Box:** Turns mysterious AI into something transparent by showing internal model attention.
- ❖ **Real-Life Example:** AI picking your poly course - wouldn't you want to know why?
- ❖ **Building Trust:** Essential for jobs like healthcare. Crucial for ethical AI and bias mitigation.
- ❖ **Singapore's Vision:** Fits AI Singapore's push for fair, safe tech everyone can trust. Mandatory for compliance.
- ❖ **Why It Matters to You:** Future employers want AI expertise – the ability to ***use and explain***. Make informed decisions.

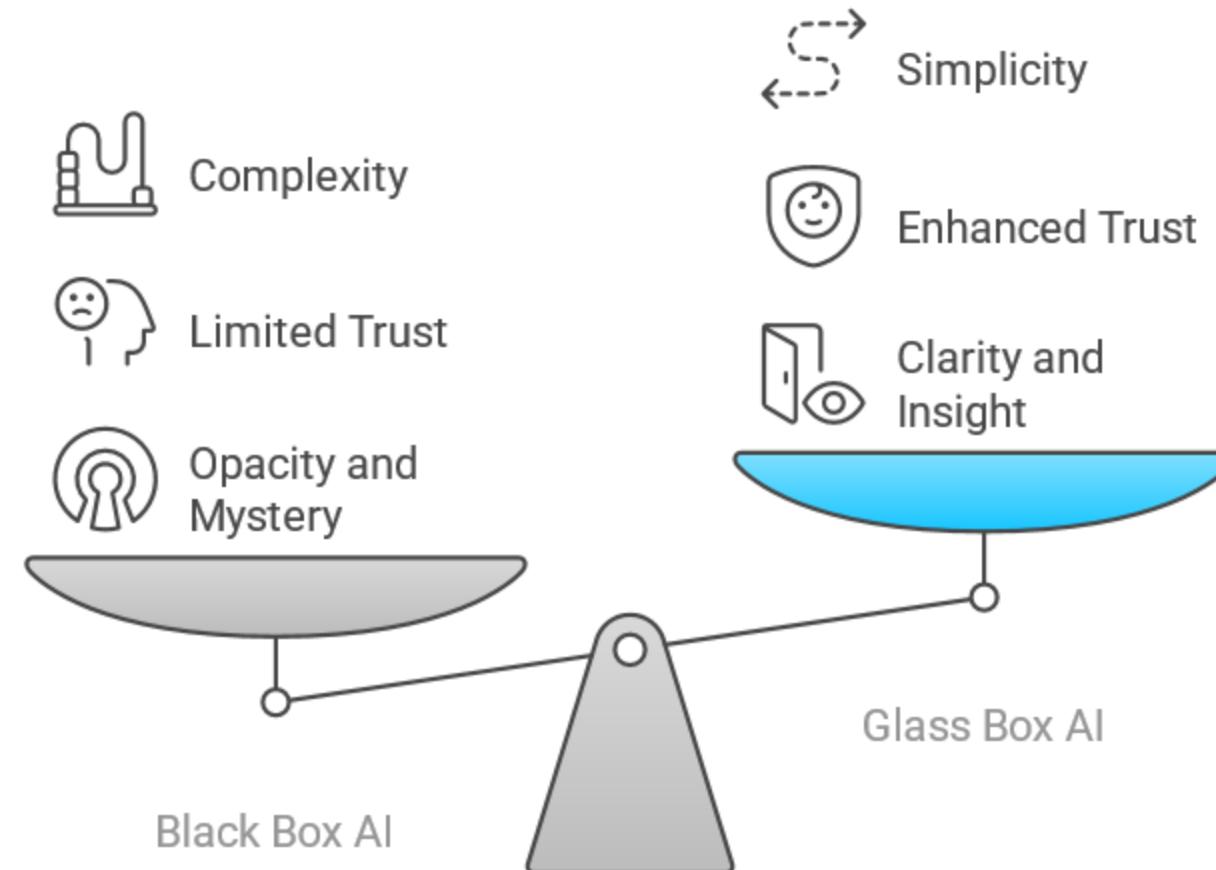


Trust, transparency and governance

- ❖ Watch the following 9-minute video better understand why there is a need for explainable AI.



Transitioning from Opacity to Clarity in AI

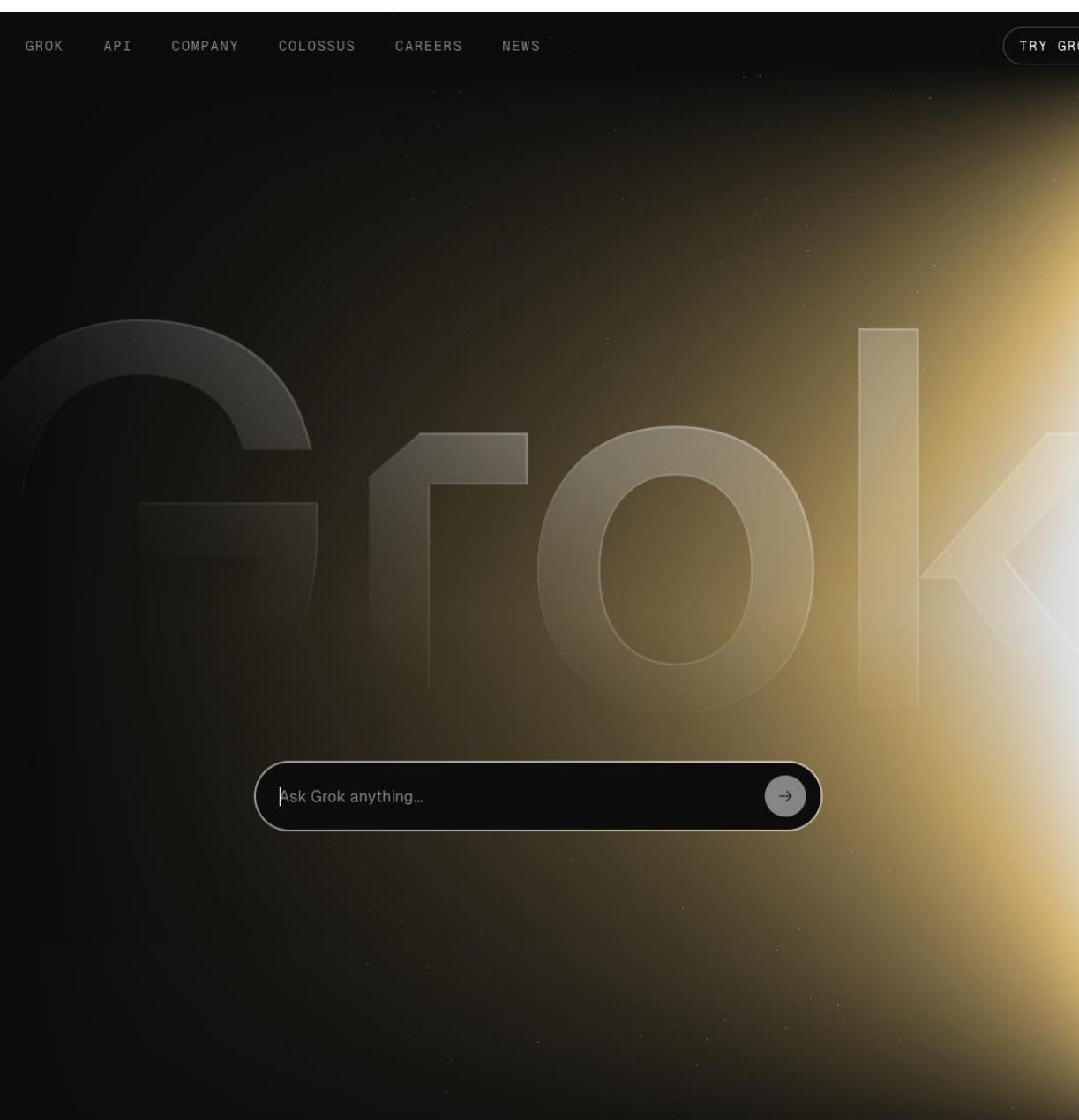




Grok

***Self Description: I'm Grok, xAI's cheeky
AI pal who dishes out witty cosmic
truths!***

Why use Grok to Investigate Explainable AI?



- ❖ **Free Access:** Available at no cost, perfect for students exploring XAI on a budget.
- ❖ **Interactive Responses:** Answers questions like "Why did you say that?" to show AI reasoning.
- ❖ **Real-Time Insights:** Pulls current X data, offering fresh examples for XAI analysis.
- ❖ **Simplified Explanations:** Breaks down complex AI ideas into clear, student-friendly terms.
- ❖ **Hands-On Learning:** Encourages active questioning to uncover how AI decisions work.
- ❖ **Unique Perspective:** Adds humour and wit, making XAI exploration engaging and fun.

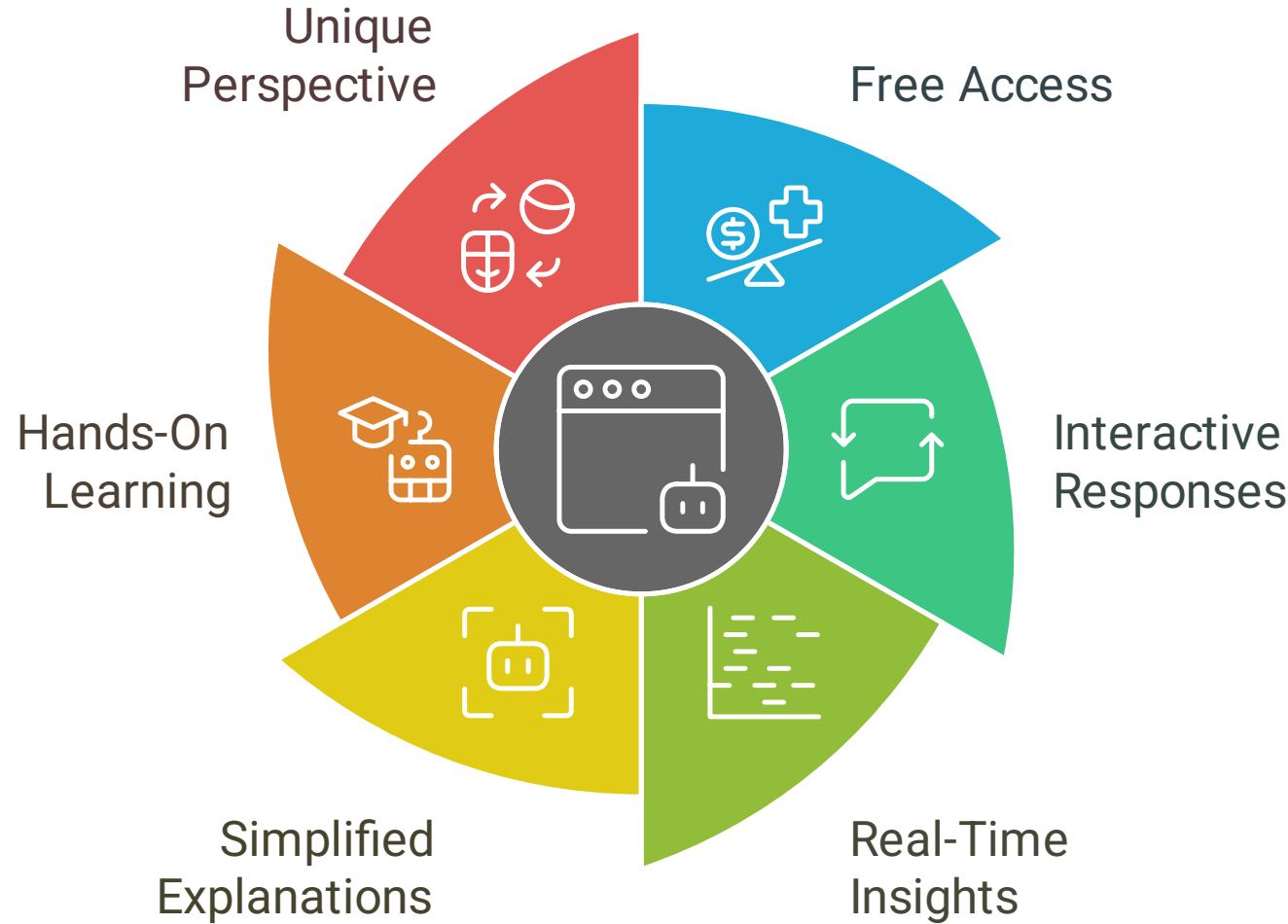
Grok 3 Free Tier: Capabilities



The screenshot shows the Grok 3 AI interface. At the top left is the Grok logo. In the top right corner are three small icons: a ghost-like figure, a document with a magnifying glass, and a user profile picture. Below the header is a large text area containing a greeting and a question: "Good evening, Peter. How can I help you today?". Underneath this is a search bar with the placeholder "What do you want to know?". Below the search bar are several buttons: "DeepSearch" (with a swirl icon), "Think" (with a lightbulb icon), "Research" (with a magnifying glass icon), "How to" (with a lightning bolt icon), "Analyze" (with a bar chart icon), "Create images" (with a camera icon), and "Code" (with a code editor icon). In the top right of the main area, there's a "Grok 3" dropdown menu and an upward arrow button. At the bottom of the interface is a red button with the text "Sign in with X or Google (Recommended)".

- ❖ **Conversational AI Access:** Useful: Ask Grok to explain its answers (e.g., “Why did you choose this?”) to peek into AI decision-making
- ❖ **Real-Time X Data:** Useful: Analyse current X posts to see how Grok interprets real-world data, revealing biases or reasoning
- ❖ **Basic Question Handling:** Useful: Pose “why” or “how” questions (e.g., “How did you decide?”) to uncover transparency limits
- ❖ **DeepSearch Lite:** Useful: Summaries of web content help compare Grok’s explanations to external sources
- ❖ **Think Mode Lite:** Useful: Step-by-step breakdowns (e.g., math problems) show interpretable AI processes
- ❖ **File Upload Analysis:** Useful: Upload sample data (e.g., text) and ask Grok to explain its analysis, testing clarity

Grok 3 Free Tier: Capabilities





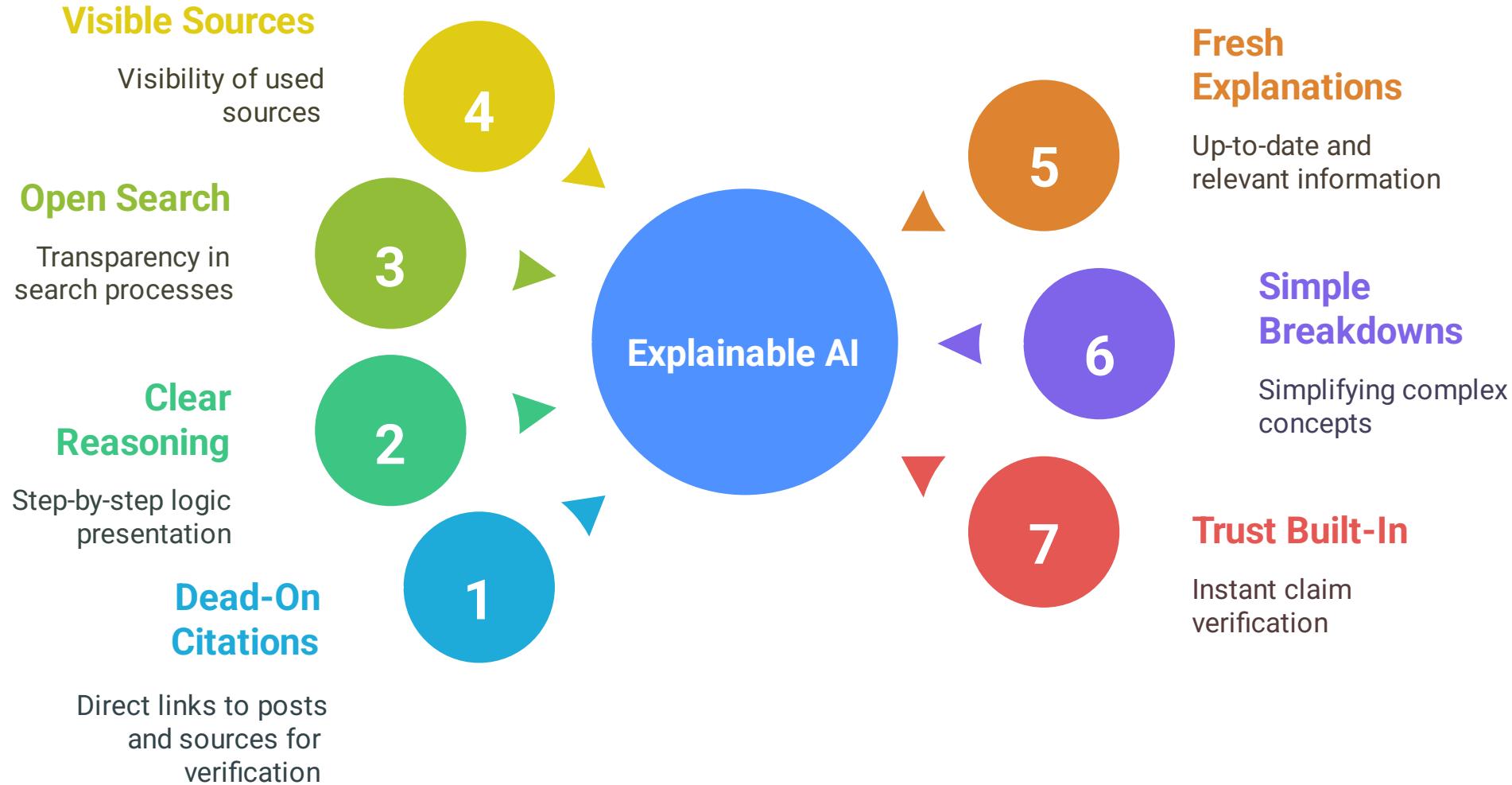
How Grok supports Explainable AI (XAI)

- ❖ **Dead-On Citations:** Links straight to X posts and web sources.
- ❖ **Clear Reasoning:** Shows its logic, step by step.
- ❖ **Open Search:** Reveals how it digs through X and the web.
- ❖ **Visible Sources:** See exactly what Grok uses.
- ❖ **Fresh Explanations:** Always up-to-date and relevant.
- ❖ **Simple Breakdowns:** Makes responses easy to grasp.
- ❖ **Trust Built-In:** Verify claims on the spot.



Grok

How Grok supports Explainable AI (XAI)





Trust, transparency and governance

- ❖ Watch the first 15 minutes of the video below to get an excellent overview of Grok and its various capabilities.



Ai



Explainable AI

Grok



Examples of XAI in Action:

- ❖ **Healthcare:**
 - **Use:** AI diagnostics (e.g., tumour detection, anomaly detection in medical imaging).
 - **Goal:** Doctors and patients understand why a tumour is flagged or an anomaly detected, leading to informed treatment decisions.
- ❖ **Finance:**
 - **Use:** Credit risk assessment.
 - **Goal:** Ensure fair lending by explaining approval/denial decisions, contributing to financial inclusion and transparency.
- ❖ **Law:**
 - **Use:** Predicting case outcomes.
 - **Goal:** Transparent legal decisions for judges and clients, facilitating understanding of the rationale behind court rulings.
- ❖ **Education:**
 - **Use:** Personalized learning paths.
 - **Goal:** Provide educators with deeper insights into student learning needs, improve learning outcomes, and explain the rationale behind the recommendations that drive tailored instruction for their students.



Activity: XAI - Mia Case Study

- ❖ **Read (5 minutes):**
 - Open the Mia AI Detection document and read it.
- ❖ **Reflect (10 minutes):** After reading, take 10 minutes to think about "explainable AI" that can clearly explain its decisions to humans in relation to this case.
 - **Transparency:** How might knowing *why* Mia flagged a cancer (e.g., a six-millimetre patch) help doctors trust and use its diagnosis?
 - **Patient Impact:** Why could it matter to patients like Barbara that Mia's decision is explainable, not just accurate?
 - **Challenges:** What might make it hard for Mia to explain its findings in a way doctors and patients understand?
- ❖ **Complete:** Complete the activity and upload to your MST Team.

Ai



Activity: Recommender

- ❖ **Ask for a Movie Recommendation:** Students use Grok's "think" feature to request a movie suggestion (e.g., "Grok, think: What movie should I watch tonight, and why?") and observe how Grok explains its reasoning.
- ❖ **Dig Deeper into the Reasoning:** Students ask Grok for a detailed explanation of a specific aspect of the recommendation (e.g., "Grok, think: Why would the coming-of-age theme in *Lady Bird* appeal to me?") to explore its thought process further.
- ❖ **Reflect on Glass Box AI:** Students consider how Grok's clear and insightful explanations demonstrate the principles of "glass box AI" and why this transparency is important for building trust and understanding in AI systems.
- ❖ **Complete:** Complete the activity and upload to your MST Team.

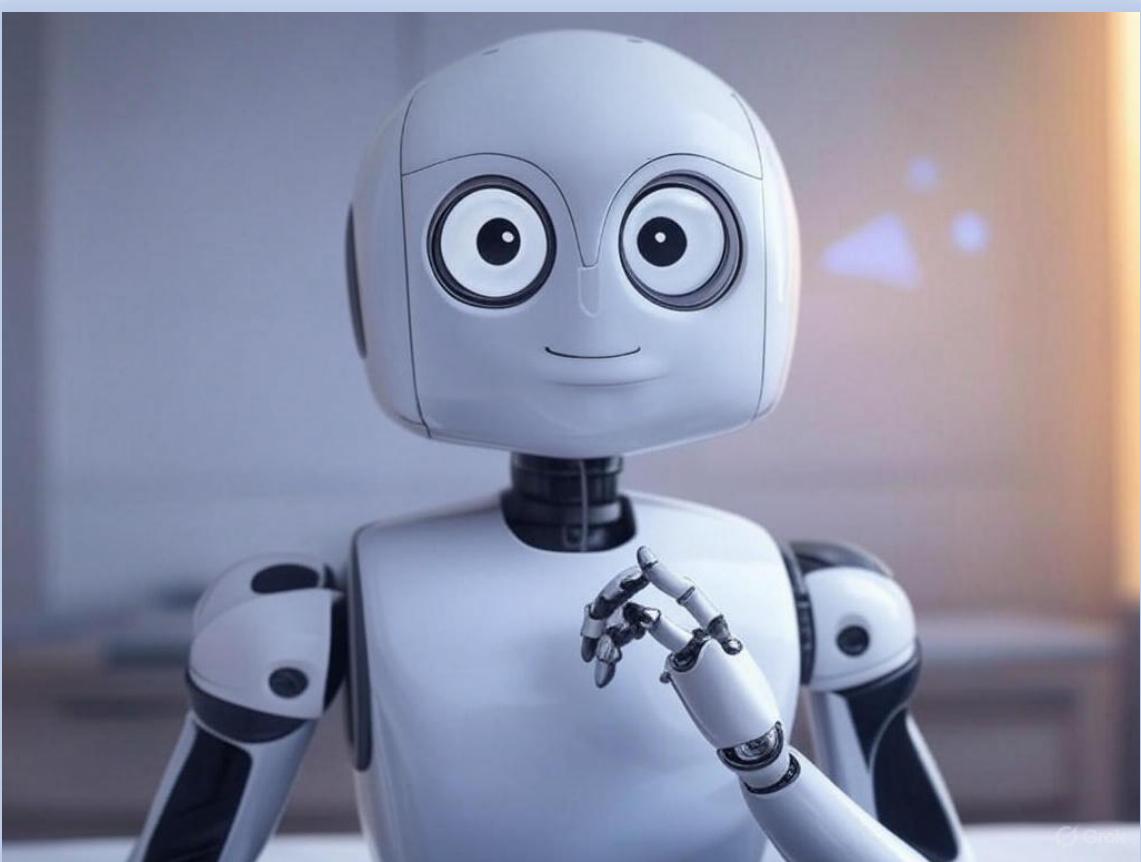
Ai



Exploring DeepSearch

Grok

Hints on Using Grok



- ❖ **Features:** You can turn off "Think" mode or "DeepSearch" whenever you wish. It is not always advantageous to have these modes turned on. For example, if you want Grok to draw images, rather than describing the image it is imagining, turn **off** "Think".
- ❖ **Attachments:** You can attach many different types of documents with the paperclip.
- ❖ **Logo:** Clicking on the Grok logo returns you to the home page and a new chat. Here you will see some default prompts you can modify like Code, Analyze and Research.
- ❖ **Privacy (Ghost) Mode:** You can turn on Temporary chat or search your chat history at the top right.



Activity: DeepSearch

Ai

- ❖ **Research Challenge:** Use Grok's **DeepSearch** to find and summarize recent articles on a specific topic (e.g., Redrawing of Singaporean Electoral Boundaries in 2025)
- ❖ **Fact-Checking Debate:** Employ **DeepSearch** to gather data and opinions from the web and X, using the information to support or refute a given statement. Select any statement you wish, like "Nuclear power is not safe for Singapore"





Lesson 07

Summary of Material Covered



Lesson 07 Review

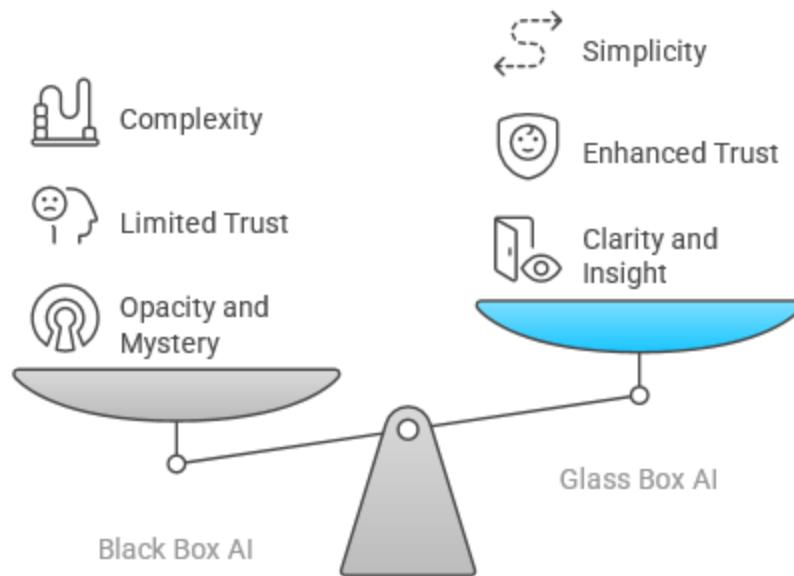
- ❖ **Intro to XAI:** Transforms AI from "**Black Box**" to "**Glass Box**" for transparency and trust.
- ❖ **Why It Matters:** Highlights trust, governance, & fairness in AI.
- ❖ **Real-Life Impact:** XAI in Healthcare, Finance, Law, & Education. Example: Mia AI in cancer detection)
- ❖ **Meet Grok:** A witty AI by X, free & interactive for exploring XAI
- ❖ **Grok's XAI Features:** How Grok supports Explainable AI. Note these features are **not unique to Grok.**
 - **Dead-On Citations:** Links straight to X posts and web sources.
 - **Clear Reasoning:** Shows its logic, step by step.
 - **Open Search:** Reveals how it digs through X and the web.
 - **Visible Sources:** See exactly what Grok uses.
 - **Fresh Explanations:** Always up-to-date and relevant.
 - **Simple Breakdowns:** Makes responses easy to grasp.
 - **Trust Built-In:** Verify claims on the spot.

Ai



Lesson 07 Review

Ai



Visible Sources

Visibility of used sources

Open Search

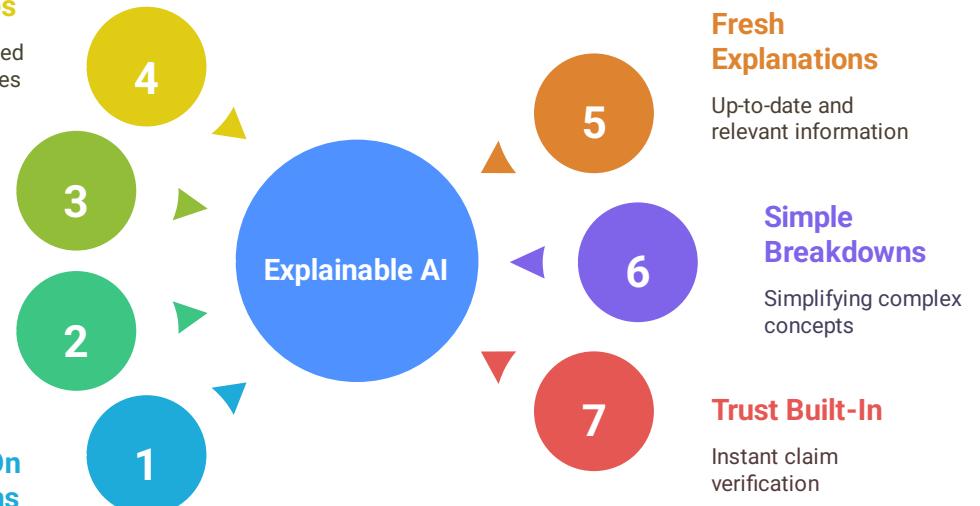
Transparency in search processes

Clear Reasoning

Step-by-step logic presentation

Dead-On Citations

Direct links to posts and sources for verification



Thank you

School of Infocomm

C240 AI Essentials and Innovations

© Republic Polytechnic 2025: All Rights Reserved



**SUPERCHARGE
YOUR RESUME
WITH AI**

Supercharge Your Resume with AI: Land Interviews Faster

Write smarter.
Apply faster.
Get noticed.

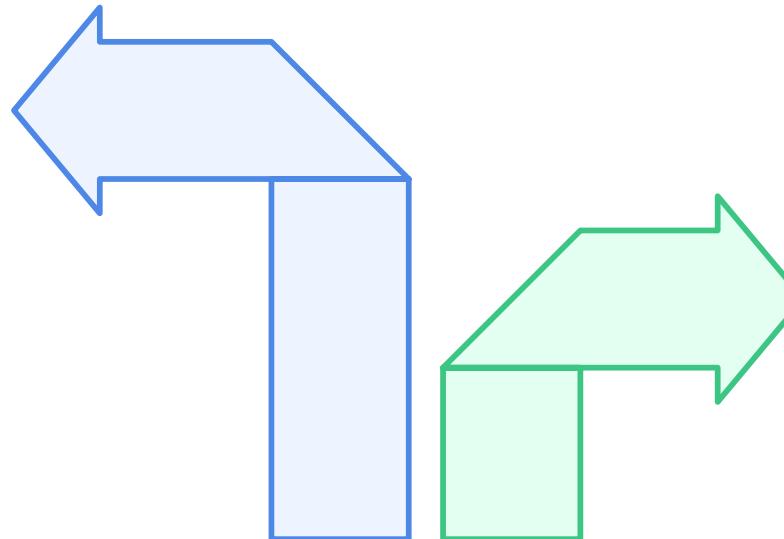


Resume vs. Curriculum Vitae(CV)

Which document should I use for my job application?

CV

Suitable for academic or research roles, providing a comprehensive overview of qualifications.



Resume

Ideal for internships and entry-level positions, focusing on brevity and relevance.

Why Use AI to Build Your Resume?



- ❖ **This is an AI module:** The goal is not just a great resume, it's learning how to use modern AI tools to think, write and revise effectively.
- ❖ **Speed + structure:** AI helps you organise your thoughts quickly and gives you a solid starting draft. No blank page paralysis.
- ❖ **Tailored resumes stand out:** Tools like ChatGPT and Gemini help you align your resume to specific job roles (e.g., OCBC vs Paris Baguette).
- ❖ **Better phrasing, fewer mistakes:** Grammarly and LLMs improve grammar, tone, clarity and style, making your writing more professional.
- ❖ **Simulate the employer's view:** You can ask AI to critique your resume from a hiring manager's perspective before you send it.
- ❖ **Interview-ready content:** If you write it with AI and can explain it clearly, you're already rehearsing for interview questions.
- ❖ **Research is easier:** NotebookLM lets you track and review company and role information, which boosts both resume relevance and interview performance.

JOHN DOE
IT INTERN

+1-234-567-8901 • john.doe@email.com • city.tsale.com/in/johndoe

EDUCATION

Bachelor of Science in Computer Science May 2025
Anytown University

- Currsework; Data Structures, Algorithms, Datácase Maç, software engineering

PROJECTS

Portfolic Website

- Personal responsive website developed using HTML, CSS, and Java Scripi using dynamic.content via JavaScript, deployed esit and GitHub Pages

Task Manager Application

- CRUD task management ant app buill with React for front-end for Wo dejs and MongoDB response and RESTful API endpoints

Bookstore Database

- Relational database system reliaed bookstore, a set eparegras repignemettet using SQL, Including queries for management inventory, customers, si

SKILLS

- Programming; Python, Java, JavaScript, SQL
- Web Development: HTML, CSS; React, Node.js
- Tools & Technologies: Git, GitHub, MongoDB
- Languages; English (fluent), Spanish (conversational)

EXPERIENCE

IT Support Assistant, Tech Solutions Inc. Summer 2023
Tech Solutions Inc., Anytown

- Provided technical support to 1000+end users/indugnesing hardware, software, and network (issues; resolving 50+ tickets monthly)
- Assisted in deploying, configuring, and maintaining worksrations, printers
- Documented IT processes and user guides to improve issue resolution efficiency

The AI-Driven Resume Workflow



- ❖ **💡 Tip:** Tailor each resume to the company. OCBC and Paris Baguette may both need IT interns, but their expectations may be very different.
- ❖ **Explore jobs & companies:** Use NotebookLM or Gemini to extract insights from job ads, websites, or PDFs. Start building a company research notebook in NotebookLM. Doing so will help when tailoring the resume and preparing for interviews.
- ❖ **Understand the role & context:** Ask ChatGPT: "What's the difference between working IT at a bank vs a retail chain?" Tailor based on scope, tools, and soft skill focus.
- ❖ **Extract job keywords:** Use an LLM to summarise key skills and expectations from each job description.

The AI-Driven Resume Workflow (cont'd)



- ❖ **Write achievement bullets:** Use LLM prompts to refine your raw experiences with more impact. Ask it to target your achievements to the job description.
- ❖ **Refine & proofread:** Grammarly checks tone, grammar, and clarity.
- ❖ **Review with AI again:** Use prompts like: "Act as a hiring manager. What's missing or overstated?" or "What would make this resume more impressive for this job?"

Mindset Tips for Resume Success



Principle	What It Means for You
Start with the end in mind	Know your ideal company, role or domain. Tailor everything towards that goal. Ignore feelings of inadequacy.
You are not your resume	The resume is just a tool. Focus on how you can help the company, not just listing what you've done. What value do you bring?
Every job is different	Avoid one-size-fits-all resumes. Maybank may need a specialist; HDB might want a flexible generalist.
Do your homework	Use NotebookLM to collect insights about the company and role. Show you care and understand them.
Evidence beats adjectives	Replace "team player" with quantifiable, specific actions and results.
Get feedback before sending	Use AI tools and ask a peer or lecturer for one final check.
Think interview now	Don't add anything you cannot explain in an interview. Use AI to simulate tough questions .

Key Tools



Tool	What It Does
ChatGPT or Gemini Pro	Draft bullets, tailor resumes, generate cover letters, extract job keywords, and rewrite for clarity. Use Gemini via Google AI Studio for free.
NotebookLM	Gather and organise company and role information. Create a notebook with sections like role summary, required skills, your matching experiences, and interview prep notes.
Grammarly	Fix grammar, tone, and consistency. Helpful for final polish.



Explore Jobs & Companies

Activity



Start Your Company Research Notebook



- ❖ **Pick a real company:** Choose one company you might apply to (e.g., OCBC, Grab, Paris Baguette, NCS, Shopee).
- ❖ **Find one resource:** Locate a job ad, company website, or news article about that company. Copy the link or text into NotebookLM.
- ❖ **Ask AI a smart question:** Paste or upload the material into NotebookLM. Then ask:
 - What kind of IT interns would this company likely need?
 - What stands out about the company culture or tech stack?
- ❖ **Create a note:** Title the note: Company: [Company Name]. Add sections for:
 - Role summary
 - Required skills
 - Possible projects
- ❖ **Recall:** Note down your matching experience (brief bullets)



Understand Role & Context

- ❖ **Compare:** Pick two companies from different sectors (e.g., OCBC Bank and Red Cross Singapore).
- ❖ In ChatGPT, ask: "What's the difference between working IT at [Company A] vs [Company B]?"
- ❖ Use the response to compare differences in:
 - Job scope
 - Expected soft skills
 - Tech tools or focus areas
- ❖ Summarise the key distinctions in your notes or NotebookLM under a new section: Role Context Comparison



Extract Job Keywords

1. Visit a local job platform such as [LinkedIn Jobs](#), [MyCareersFuture](#), [Grab](#), or [JobStreet](#).
 - ❖ Search for an IT internship role (preferably in Singapore).
 - ❖ Copy and paste the job description into ChatGPT or Gemini.
 - ❖ Prompt: **Extract the top 10 technical skills and 5 soft skills this job requires. Format them as a checklist.**
 - ❖ Save the results into your NotebookLM under a new note titled: Job Keywords - [Company Name]
 - ❖ Bonus: Compare the keywords across 2 different roles and reflect—Which one suits your current strengths more?

Ai

The background image shows an aerial view of a large, modern office complex. The buildings are white with multiple levels and flat roofs. They are surrounded by a lush green landscape, including trees and a small body of water. The sky is clear and blue.

Alignment



Alignment is Key

You have experience, but it doesn't quite match what is required. This is where LLM can help. You may want to change the temperature a little.

- 💡 **Student's experience:** I built a website for a school project promoting food donation and hosted it on GitHub Pages.
- 🏢 **Job Request:** Website maintenance skills required.
- 💬 **Prompt:** Rewrite this "I built a website for a school project promoting food donation and hosted it on GitHub Pages" as a one-line resume bullet using action + task + result. Make it relevant to a job description asking for **website maintenance skills**.
- ✓ **Result:** Deployed and maintained a static food donation campaign site on GitHub Pages, updating content weekly and ensuring 99% uptime over three months.



Alignment is Key

You have experience, but this time it's a bit of a stretch to see how it could apply to a particular job request.

 **Student's experience:** I created a Python game for a school project that tracks player progress and scores.

 **Job Request:** Data entry and management skills required (e.g., Excel, structured documentation, handling large datasets)..

 **Prompt:** Rewrite my experience as a one-line resume bullet using action + task + result. Make it relevant to a job requiring data entry and management skills.

 **Result:** Designed and coded a Python-based game that tracked and stored player data in structured format, demonstrating attention to data accuracy and organisation.

You may be surprised how good AI is at alignment. Just don't go too far as it will become obvious 😊



Write Your Resume

Activity





Complete your Resume

Complete your Resume either in MS Word or Google Docs.

- ❖ **Request:** Ask your favourite AI to create it for you. All foundational models are good at this task. You can use Google, OpenAI, Perplexity, Grok or a model in LM Studio
- ❖ **Prompt:** "I am a polytechnic student in Singapore studying Information Technology. Please create a **resume** in British English for an IT internship role. Format it using standard Word headings (Education, Projects, Technical Skills, etc.). Focus on clarity, professionalism and brevity. The tone should match Singapore industry expectations (e.g., concise, no photo, no NRIC). Include one academic project, one CCA, and placeholder contact details. Use standard fonts like Calibri or Arial."
- ❖ **Generate:** The AI will now create a template you can use. Depending on your selection of AI it may even create the MS Word/Google Docs file for you. If you prefer a different format, use prompt chaining until you get exactly what you want.



Complete your Resume

- ❖ **Update the Template:** Update the resume with your own details. Use the techniques taught in this lesson.
- ❖ **Review & Critique:** Ask the AI for a final review and critique.
- ❖ **Save:** When you are completely satisfied, save the resume in a location you will not forget.
- ❖ **Remember:** For each job your resume should change slightly. Generic resumes are not as good as custom resumes for landing interviews.
- ❖ **Upload:** When satisfied, upload your completed resume to MST.

- ❖ **Bonus:** If you have time, create a cover letter. Cover letters, like resumes should change slightly for different positions.
- ❖ **Bonus:** Use other tools like Grammarly for that extra polish.

Ai



Lesson 08

Summary of Material Covered



Lesson 08 Review

- ❖ 🧠 This lesson was entirely about **real-world application of AI skills**.
- ❖ ✅ You were expected to **use AI tools** (ChatGPT, Gemini, NotebookLM, Grammarly) to build and refine your resume.
- ❖ 🧠 Good resumes are **tailored to the job** - OCBC and Paris Baguette may expect different things.
- ❖ 🔎 Use AI to help you **analyse the job**, understand the company, and plan your positioning.
- ❖ 🖌 Write strong **achievement bullets** using the action + task + result format.
- ❖ 📄 Extract **job keywords** from local job platforms using ChatGPT or Gemini.
- ❖ 📄 Use NotebookLM to **organise your research**: company, role, skills, alignment.
- ❖ 💬 Your resume is a **living document** -- review, rewrite and polish with AI support.
- ❖ 💬 Always be ready to explain your bullet points in an interview.

Thank you

School of Infocomm

C240 AI Essentials and Innovations

© Republic Polytechnic 2025: All Rights Reserved



botpress

Master the Future: Build Smarter RAG Bots with Botpress!

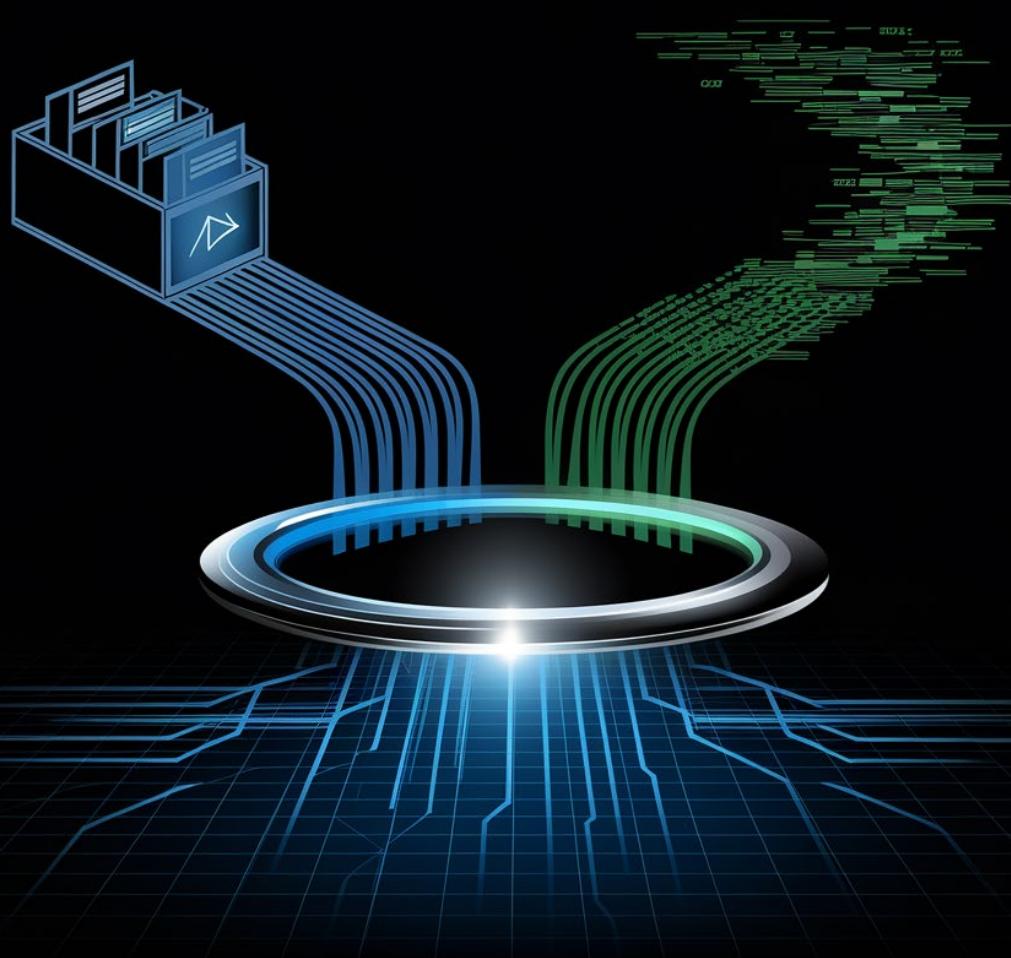
Unleash AI Brilliance, One Bot
at a Time



RAG

Retrieval Augmented Generation

Retrieval-Augmented Generation (RAG)



- ❖ **What It Is:** A hybrid AI approach combining retrieval and generation for smarter, context-aware responses.
- ❖ **Retrieval Step:** Pulls relevant info from a data source or other sources using a query.
- ❖ **Generation Step:** Uses a language model to create a coherent, tailored answer based on retrieved data.
- ❖ **Why It's Powerful:** Boosts accuracy and relevance by grounding AI outputs in real, up-to-date information.
- ❖ **Use Case:** Think chatbots that fetch facts from documents before replying—no more guessing!

Retrieval-Augmented Generation (RAG)



YouTube

But it is not all Sunshine and Lollipops



Pros	vs	Cons
Enhances accuracy		Slower process
Reduces hallucinations		Quality reliance
Scalable		Complex implementation
Contextual		Higher resource use
Flexible integration		Risk of irrelevant data

❖ Pros:

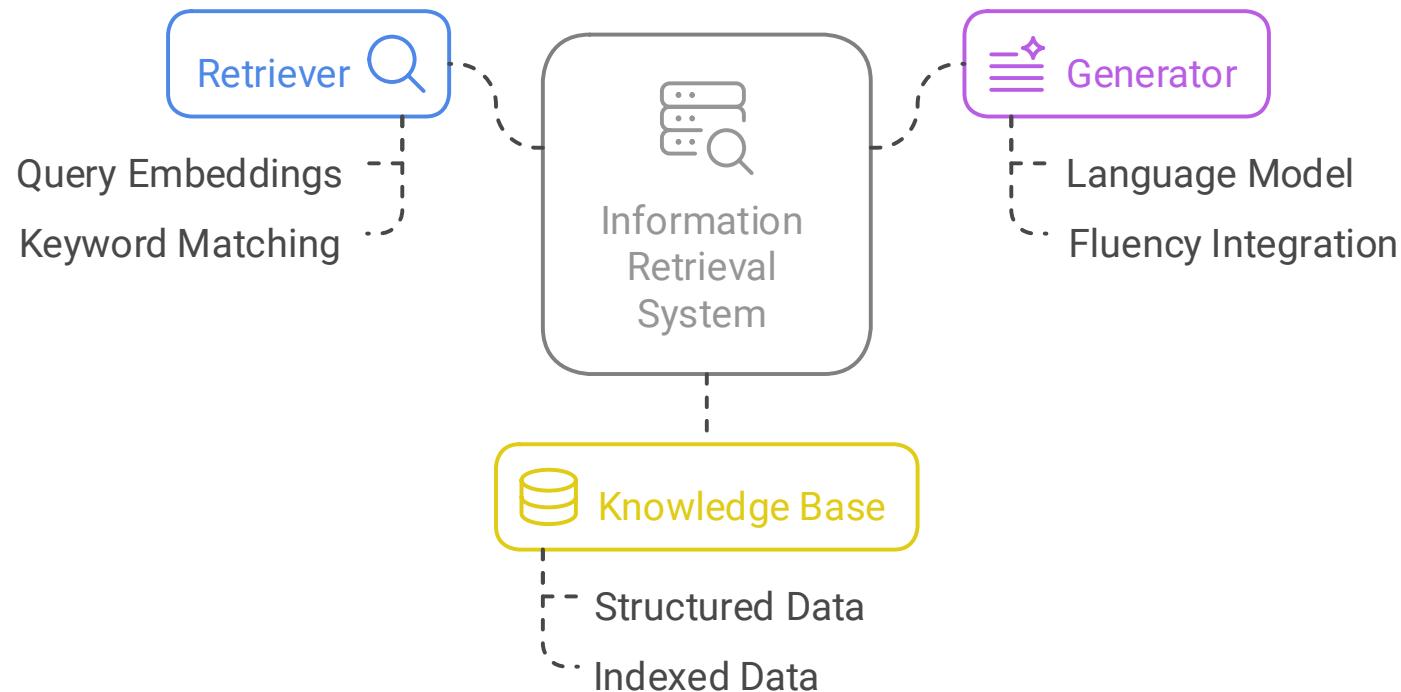
- Enhances accuracy: up-to-date, dynamic information.
- Reduces hallucinations by using real data.
- Scalable: adapts easily to new datasets.
- Contextual: tailors responses to each query.
- Flexible: integrates with existing knowledge bases.

❖ Cons:

- Slower due to the retrieval process.
- Relies on the quality of external data.
- Complex to implement and optimize.
- Higher resource usage (compute, memory).
- Risk of irrelevant or noisy retrieved data.

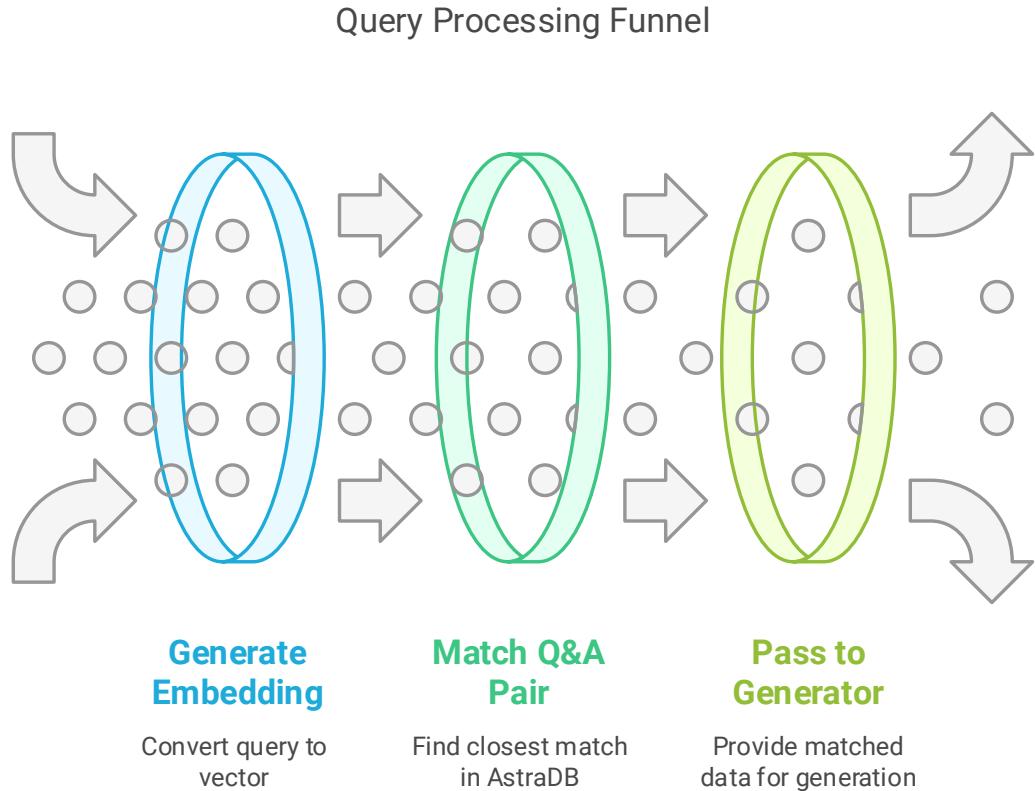


Key Components



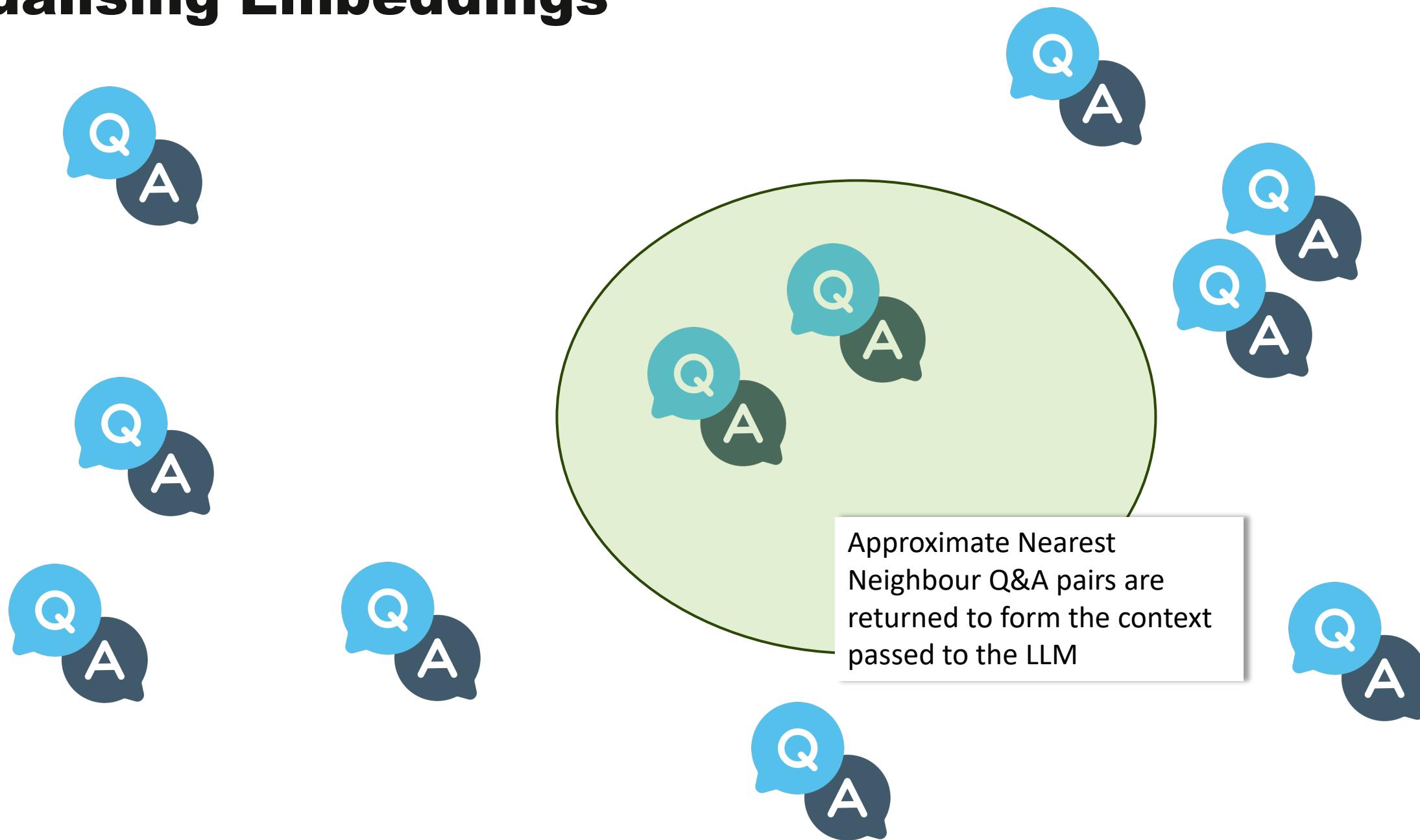
- ❖ **Retriever:** Searches for relevant info using query embeddings or keywords. Matches queries to documents in the knowledge base via vector **similarity** search
- ❖ **Generator:** A language model that takes retrieved documents plus the query to produce a coherent answer. Blends retrieved info with its pre-trained knowledge for fluency.
- ❖ **Knowledge Base:** The external data repository (e.g., documents, databases, web). Must be well-structured and indexed for efficient retrieval, often using embeddings for semantic search.

Vector Database



- ❖ **What They Are:** Databases storing text as vectors (big numbers) for semantic search.
- ❖ **Role in RAG:** Retriever uses them to fetch relevant Q&A pairs from a knowledge base. Closest few answer vectors to the question vector.
- ❖ **Practical Setup:** For a Q&A document, chunk each Q&A pair. Convert each Q&A into embeddings using a model like **SentenceTransformers**. Store in vector DB.
- ❖ **How It Works:** Query → embedding query → match closest Q&A pairs → pass closest QnA pairs to the model as context.
- ❖ **Why It Helps:** Ensures precise retrieval of specific Q&A chunks, improving answer relevance.

Visualising Embeddings



Terms



- ❖ **Grounded:** Responses are based on real, retrieved data, not just the model's guesses.
- ❖ **Retrieval:** The process of finding relevant info from a knowledge base using a query.
- ❖ **Generation:** Creating a coherent answer using a language model and retrieved data.
- ❖ **Embedding:** Numerical vector representing text's meaning for similarity search.
- ❖ **Vector Database:** Stores embeddings for fast, semantic retrieval (e.g., **Weaviate**, AstraDB, Pinecone).
- ❖ **Knowledge Base:** External data source (e.g., documents, Q&A) from which RAG retrieves.
- ❖ **Retriever:** Component that searches the knowledge base for relevant info.
- ❖ **Generator:** Language model that produces the final answer from the context and the user's question.
- ❖ **Semantic Search:** Finding data based on **meaning**, not just keywords, using embeddings.
- ❖ **Hallucination:** When a model makes up incorrect info. RAG reduces hallucinations.



botpress

Streamline AI development with
Botpress's powerful, low-code
conversational AI platform.

Introducing Botpress



❖ What is Botpress?

- Open-source conversational AI platform.
- Purpose-built for creating chatbots and virtual agents.

❖ Key Features:

- Visual drag-and-drop flow editor.
- Built-in knowledge bases with retrieval-augmented generation (RAG).
- Connects easily to APIs, databases, and tools.
- Fast prototyping and cloud-hosted deployment options.

❖ Why Use Botpress?

- Simpler and faster to build chatbots compared to heavier frameworks.
- Ideal for AI applications needing real, document-grounded responses.

Botpress Knowledge Base & Vector Search



❖ Built-in Knowledge Base:

- Upload documents, text, or URLs to create a searchable repository.
- Enables bots to provide accurate, document-grounded responses.

❖ Semantic Search with Vector Embeddings:

- Transforms content into vector embeddings to capture semantic meaning.
- Facilitates retrieval of relevant information based on user queries.

❖ Vector Database Integration:

- Utilizes **Weaviate**, an open-source vector database, to store and manage embeddings.
- Supports efficient semantic search and retrieval-augmented generation (RAG).

❖ Scalable and Efficient:

- Handles large datasets with optimized search capabilities.
- Enhances the bot's ability to provide precise and contextually relevant answers.





Botpress Studio & Visual Flow Editor

Slide 3: Botpress Studio & Visual Flow Editor

❖ Botpress Studio:

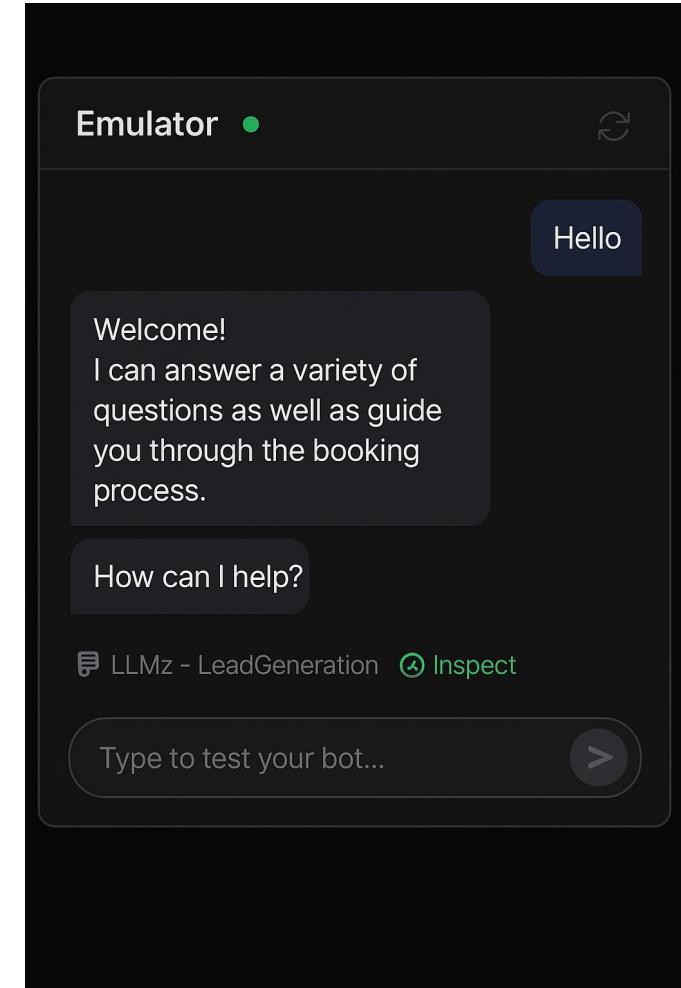
- Centralized environment for building, testing, and deploying AI agents.
- Integrates tools for managing knowledge bases, flows, and integrations.

❖ Visual Flow Editor:

- Drag-and-drop interface for designing conversation flows.
- Utilize nodes and cards to define dialogue logic and actions.
- Supports modular workflows for complex conversation structures.

❖ Built-in Emulator:

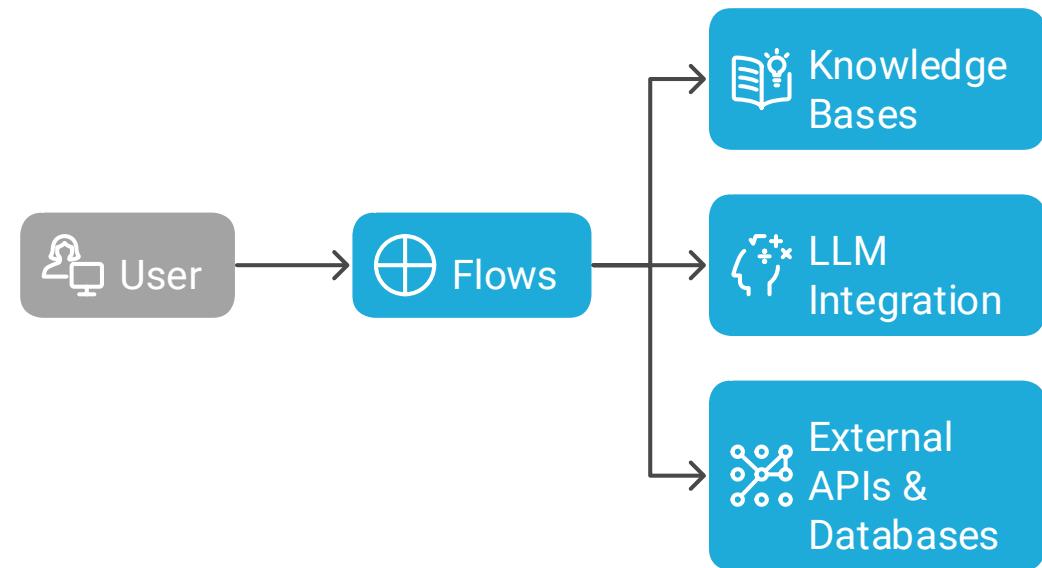
- Test and debug conversations in real-time within the Studio.
- Inspect LLM decisions and iterations for accurate responses.





Botpress: How it all connects!

- ❖ **User:** Interacts with the chatbot via messaging interface.
- ❖ **Flows:** Handle dialogue structure, control conversation logic, context switching, and actions.
- ❖ **Knowledge Bases:** Power Retrieval-Augmented Generation (RAG) respond using real documents and data.
- ❖ **LLM Integration:** Under the hood, Botpress can call large language models (e.g., OpenAI) to enhance responses.
- ❖ **External APIs & Databases:** Optional connections to pull live data or trigger business workflows.



Ai

Activity



Kahoot!



Botpress

Activity





Activity: Create a RAG Chatbot

- ❖ Open the document called: **Building a RAG System in Botpress**
- ❖ Individually, follow the timing given in the document
- ❖ Post your results to MST.



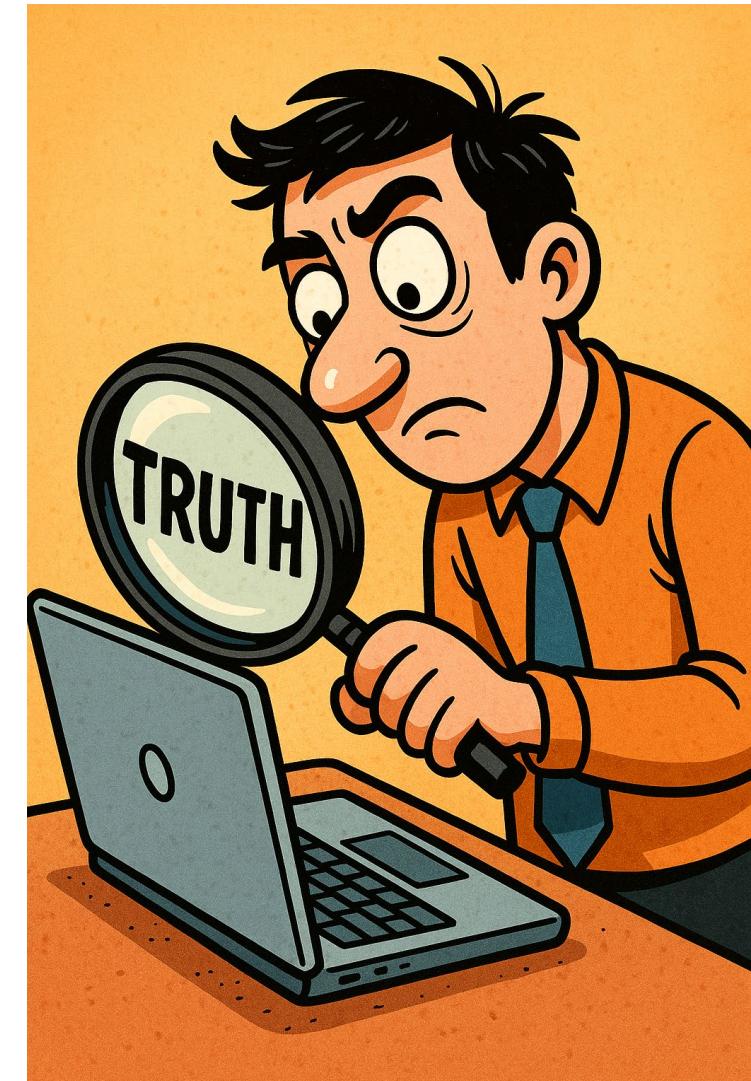
Lesson 09

Summary of Material Covered

Lesson 09 Review



- ❖ **Retrieval-Augmented Generation (RAG):** Combines retrieval of real-world data with AI generation for accurate, grounded responses.
- ❖ **Key Components:** Retriever, Generator, Knowledge Base, Vector Database (e.g., Weaviate).
- ❖ **RAG Pros & Cons:** Boosts accuracy and context but adds complexity and resource demands.
- ❖ **Botpress Introduction:** Open-source, low-code platform for creating RAG-powered conversational AI.
- ❖ **Botpress Features:** Built-in knowledge bases, semantic search with vector embeddings, drag-and-drop visual flow editor, LLM integration.



Thank you

School of Infocomm

C240 AI Essentials and Innovations

© Republic Polytechnic 2025: All Rights Reserved



Agents Unleashed: Crafting AI Agents with Flowise

Forge Smart Agents, Flow by Flow

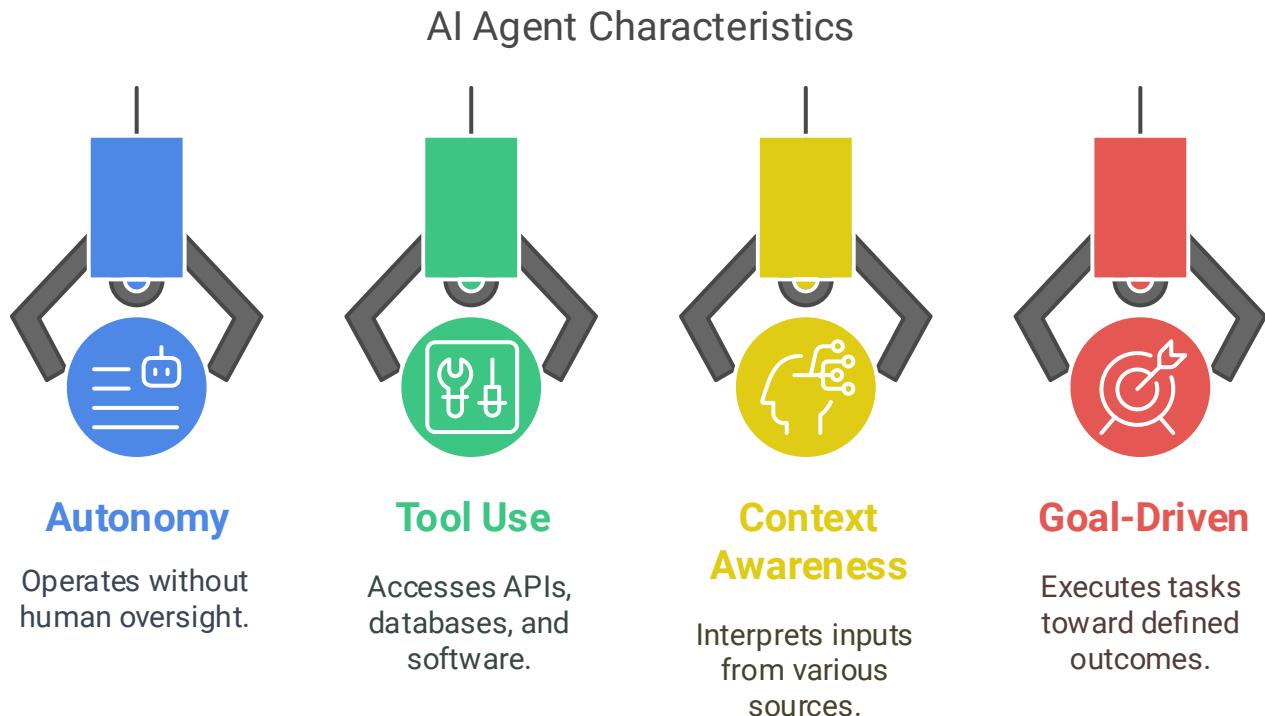


AI Agents

Your AI ally, effortlessly
supportive.



Definition of an AI Agent



An AI agent is a **digital entity** capable of autonomously performing tasks by understanding its environment, following instructions, using tools, and adapting its actions to achieve specific goals.

AI Agent Components



Core Thinking

Central processing unit. AI Brain. A Large Language Model that can think and make decisions.



Instructions *

The prompts that direct the AI's actions and responses.



Memory

The memory system that keeps track of previous interactions for continuity.



Knowledge *

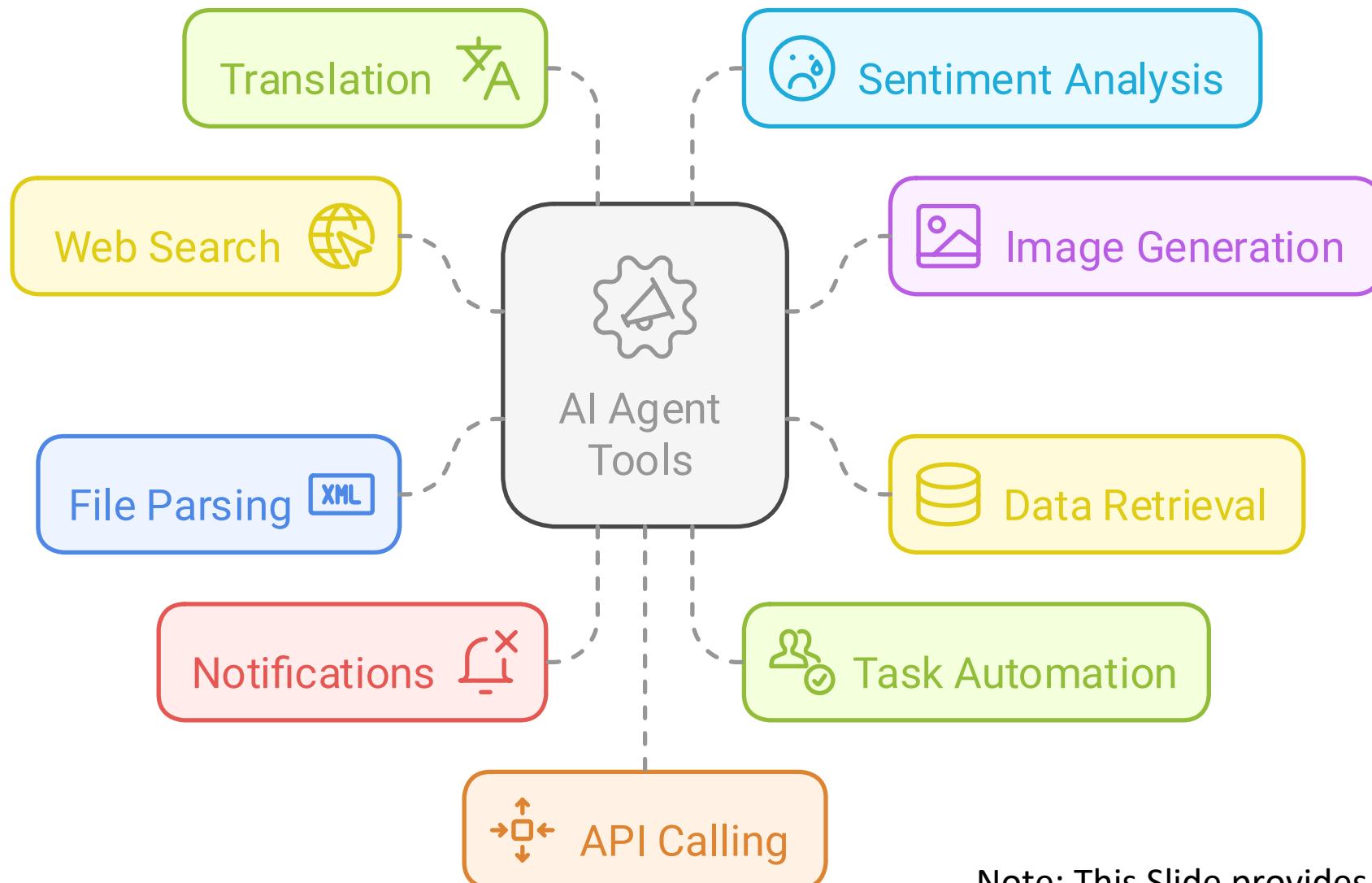
Optional additional resources that provide the AI with extra information. (e.g., A document)



Tools *

Tools that enhance the AI's functionality with specialized tasks.

AI Agent Tools



Note: This Slide provides examples of AI tools. The list is not exhaustive.

AI Agent Applications



1. Research Agents

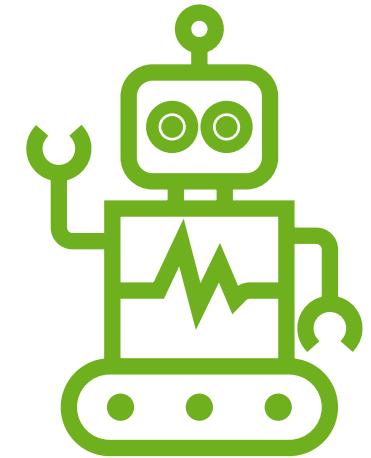
- *Examples:* Perplexity AI, Consensus, Scite
- *Functions:* Web scraping, data aggregation, summarising findings

2. Creative Agents

- *Examples:* Canva Magic Studio, Midjourney, Runway, Adobe Firefly
- *Functions:* Image and video generation, content creation, style transfer

3. Personal Assistants

- *Examples:* Google Assistant, Apple Siri, Amazon Alexa
- *Functions:* Scheduling, reminders, answering queries





AI Agent Applications (cont'd)

4. Copilots (Role-Specific Business Assistants)

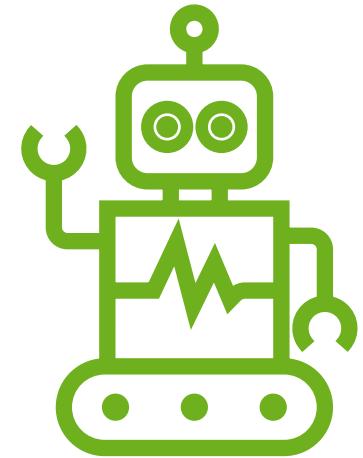
- *Examples:* GitHub Copilot, Microsoft Copilot
- *Functions:* Code assistance, document drafting, task automation

5. Customer Support Agents

- *Examples:* Zendesk AI, Octocom, Ada
- *Functions:* Code assistance, document drafting, task automation

6. Lead Generation & Appointment Setting Agents

- *Examples:* Saleshandy, Apollo.io, Clay
- *Functions:* Capturing leads, booking appointments, qualifying prospects



AI Agent vs. AI Chatbot

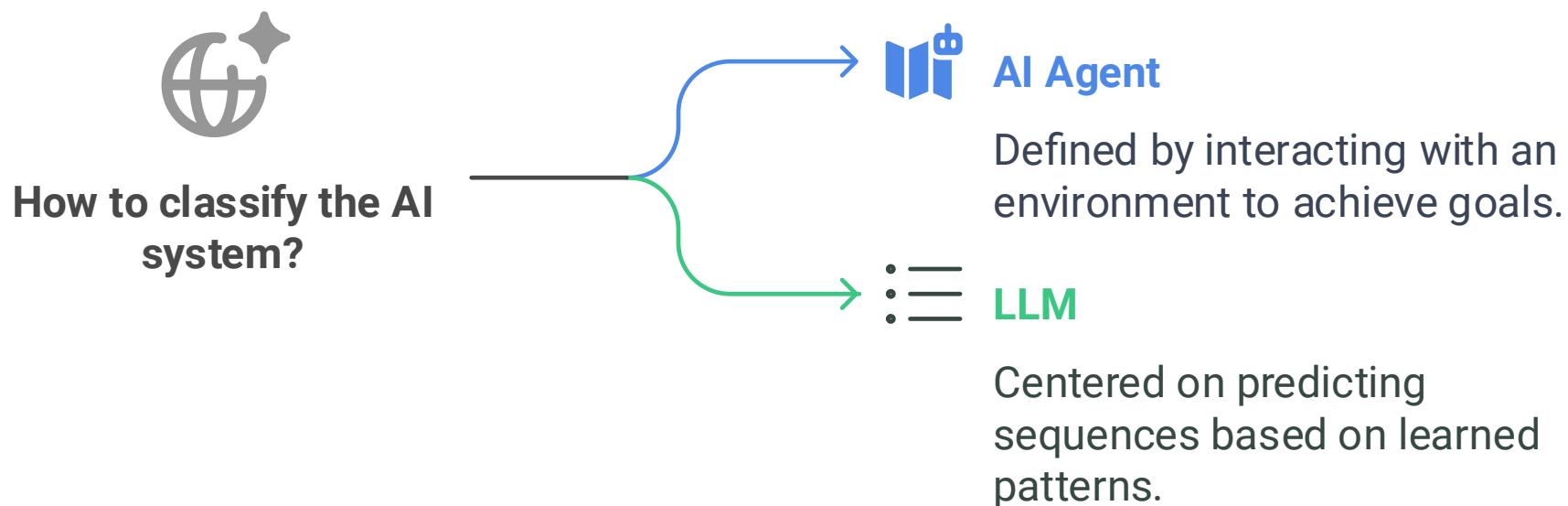


Aspect	AI Agent	AI Chatbot (ChatGPT, Gemini, Grok)
Definition	System that autonomously achieves goals	Model focused on conversational output.
Autonomy	AI Agent: High autonomy, initiates actions, uses tools.	Low autonomy, reacts to prompts, conversation-driven.
Scope	Full system (perception, decision, action)	Primarily language processing.
Functionality	Executes multi-step tasks.	Generates responses, follows scripts.
Example	Drift bot capturing leads, then booking appointments.	Zendesk bot answering support tickets



The Fundamental Difference.

- ❖ **Ai Chatbot:** Fundamentally defined by its ability to predict sequences (usually text) based on learned patterns, primarily conversational. While its generation process involves steps resembling sense-decide-act, its core architecture and purpose are centered on statistical language modeling, not direct environmental interaction in the traditional agent sense.
- ❖ **AI Agent:** Fundamentally defined by interacting with an environment via a **sense-decide-act** cycle. The focus is on achieving goals within that environment.





An AI Agent Analogy

- ❖ **LLM Chatbot:** You type into Grok, "Tell me about ecosystems." It responds with a clear explanation: "Ecosystems are communities of living organisms interacting with their environment, like forests or oceans." It stops there, waiting for you to ask more, such as "What's an example?" or "How do they work?" It's like a smart tutor who answers exactly what you ask and nothing extra unless you keep prodding.

Task Completion



AI Agent

100%

Proactive,
completes the
project

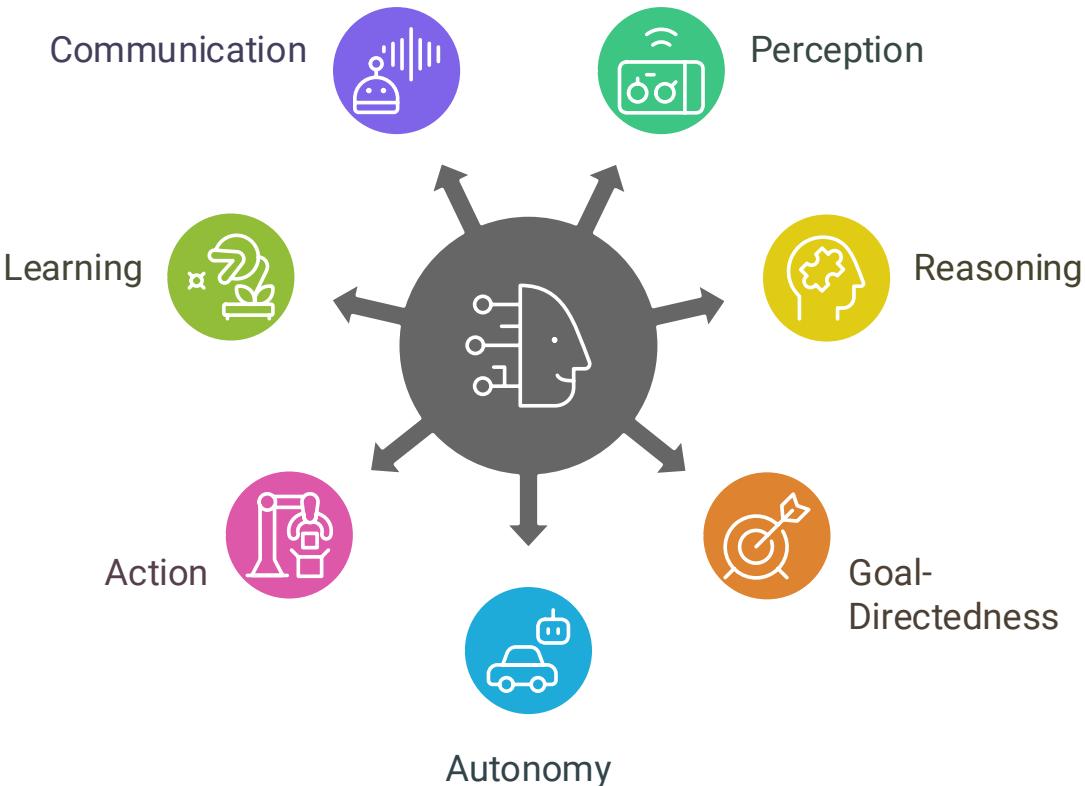
LLM Chatbot

25%

Reactive, answers
only direct
questions

- ❖ **AI Agent:** You send a message to your study assistant app saying, "I need help with my biology project on ecosystems." The app reads your request, searches the web for recent articles on ecosystems, pulls data from a science database, compiles a short report with examples like rainforests and deserts, and then creates a simple chart comparing them. It emails you the report and chart, all without you asking for each piece. It's like a classmate who takes initiative to cover all bases for you.

AI Agents: Core Attributes



1. **Perception:** Senses and interprets its environment using sensors or data inputs
2. **Reasoning and Decision-Making:** Analyzes data, plans, and selects actions to achieve its goals, often using logic, algorithms, or learned models
3. **Goal-Directedness:** Acts to achieve explicit or inferred objectives, often optimizing for a reward or utility function
4. **Autonomy:** Operates independently, making decisions and taking actions without constant human intervention
5. **Action:** Executes chosen actions via actuators or software outputs, thereby influencing the environment
6. **Learning and Adaptation:** Improves its behavior and decision-making over time through feedback and experience
7. **Communication (optional):** May interact and collaborate with humans or other agents



Flowise

Streamline AI development with
Flowise's intuitive, low-code
platform.



- Open-source**, NodeJS-based platform
- Built on LangChain**, supporting complex LLM orchestration
- Drag-and-drop UI** for building agent flows and RAG apps
- Modular building blocks** (LLMs, vector stores, tools, prompts, memory, I/O)
- Visual workflow design** for chaining components into custom pipelines
- Integrates** with OpenAI, Hugging Face, Pinecone, Chroma, and more
- Rapid prototyping** and scalable deployment, local or cloud
- Extensive documentation** and developer resources
- Self-hosting supported**, ideal for secure environments
- AgentFlows** enable autonomous AI agents with memory and tool use
- Embed widgets and APIs** into your apps for easy deployment

Using Flowise



You can use Flowise in three main ways:

1. FlowiseAI Cloud (<https://cloud.flowiseai.com>)

1. Easiest option. No setup needed. Ideal for beginners or quick prototyping.
2. Recommended for: Students, demos, and short-term projects.

2. Local Installation

1. Install via Node.js or Docker. Full control of environment.
2. Good for: Developers, tinkering, offline use.

3. Self-Hosted Deployment

1. Run on your own server or cloud (e.g., Hugging Face).
2. Best for: Production environments.

Free

\$0/month

- ② 2 Flows & Assistants
- ② 100 Predictions / month
- ② 5MB Storage ②
- ② Evaluations & Metrics
- ② Custom Embedded Chatbot Branding
- ② Community Support

Get Started

Recommended Approach for Using Flowise



🧪 Start with Flowise AI Cloud

Use <https://cloud.flowiseai.com> to explore Flowise and build your first agent easily. No setup needed.

☁️ Deploy to Hugging Face

Follow the installation guide in your lesson materials to set up Flowise on Hugging Face **after class**. Read the guide carefully.



Research Agent



Agent Name



</>

+

Buffer Window Memory

Inputs

Size * 20

Additional Parameters

Output

BufferWindowMemory

Custom Tool

Inputs

Select Tool * todays_date_time

Return Direct

Output

CustomTool

Serp API

Inputs

Connect Credential * Serpapi Key

Output

SerpAPI

Tools

Memory

ChatGoogleGenerativeAI

Inputs

Cache

Context Cache

Connect Credential * Google Generative AI

Model Name * gemini-2.0-flash

Temperature 0.5

Allow Image Uploads

Additional Parameters

Output

ChatGoogleGenerativeAI

LLM

Agent

Tool Agent

Inputs

Tools *

Memory *

Tool Calling Chat Model * Chat Prompt Template

Input Moderation

Additional Parameters

Output

AgentExecutor

Hi there! How can I help?

What's the latest news in AI today, and how many days are left until the end of the year?

Here's a summary of the latest AI news:

- Google is using AI to combat scams.
- Google launched 'implicit caching' to make accessing its latest AI models cheaper.
- AI hallucinations are getting worse.
- Pope Leo XIV identifies AI as a main challenge for humanity.

Today is Sunday, 11 May 2025. There are 234 days left until the end of the year.

What's the cube root of 125?

The cube root of 125 is 5.

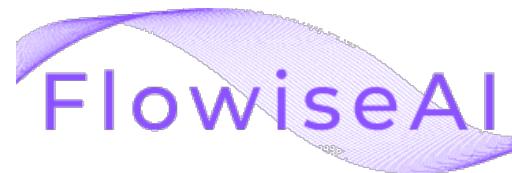
Type your question...

+ - ⌂ 🔍



Activity: Flowise

- ❖ Open the accompanying Flowise activity document
- ❖ Follow the detailed instructions included in the activity document.
- ❖ Don't forget to export your workflow to JSON.
- ❖ Don't forget to post to MST.



Ai



Lesson 10

Summary of Material Covered



Lesson 10 Review

- ❖ An AI agent is a **digital entity** capable of autonomously performing tasks by understanding its environment, following instructions, using tools, and adapting its actions to achieve specific goals.
- ❖ The **core attributes** of an AI Agent may include Perception, Reasoning and Decision-Making, Goal-Directedness, Autonomy, Action, Learning and Adaptation and optionally Communication.
- ❖ **Flowise** is an open-source, drag-and-drop UI. It is a modular, visual workflow tool integrating multiple LLMs, databases and other tools.



HuggingFace Spaces

Ai

IMPORTANT

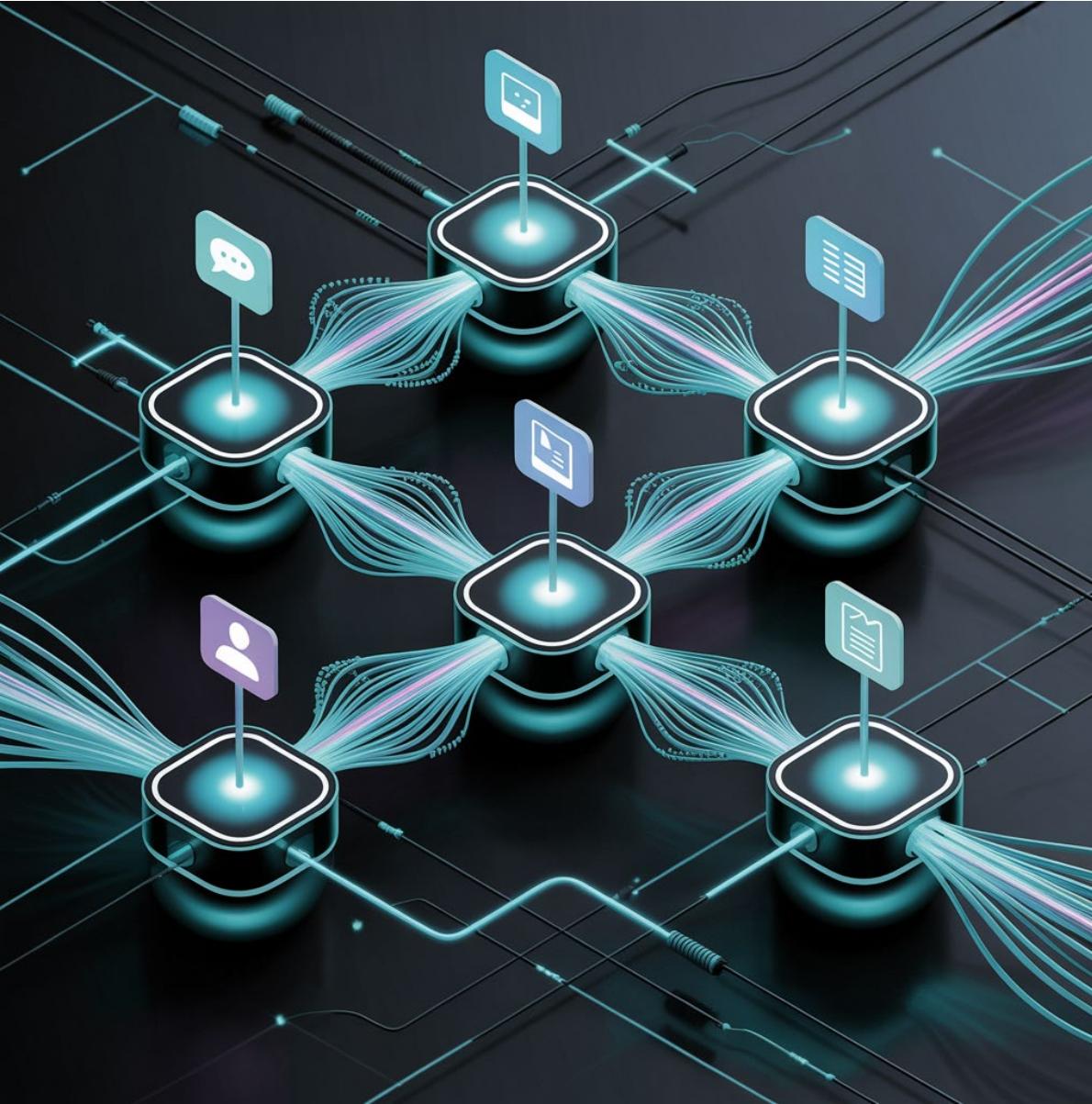
- ❖ **Hugging Face Install:** As you develop your team project, you are likely to encounter inference limits on FlowiseAI.on [FlowiseAI](#).
- ❖ To avoid issues, **you need to set up an additional install on Hugging Face Spaces**. A short, beginner-friendly guide is provided in the lesson materials.
- ❖ Other deployment alternatives exist but will not be supported by your lecturers. You can investigate here: [Flowise Alternative Deployment](#)
- ❖ If you have no alternative to FlowiseAI and run into inference problems in your project or during your presentation, it will not be considered a valid and reasonable excuse. Your team grade will be **adversely** affected.

Thank you

School of Infocomm

C240 AI Essentials and Innovations

© Republic Polytechnic 2025: All Rights Reserved



Flowise Fusion: Building AI Crews

Build Smart Teams, Flow by Flow



Multiple AI Agents

What's better than one agent?
Multiple agents!

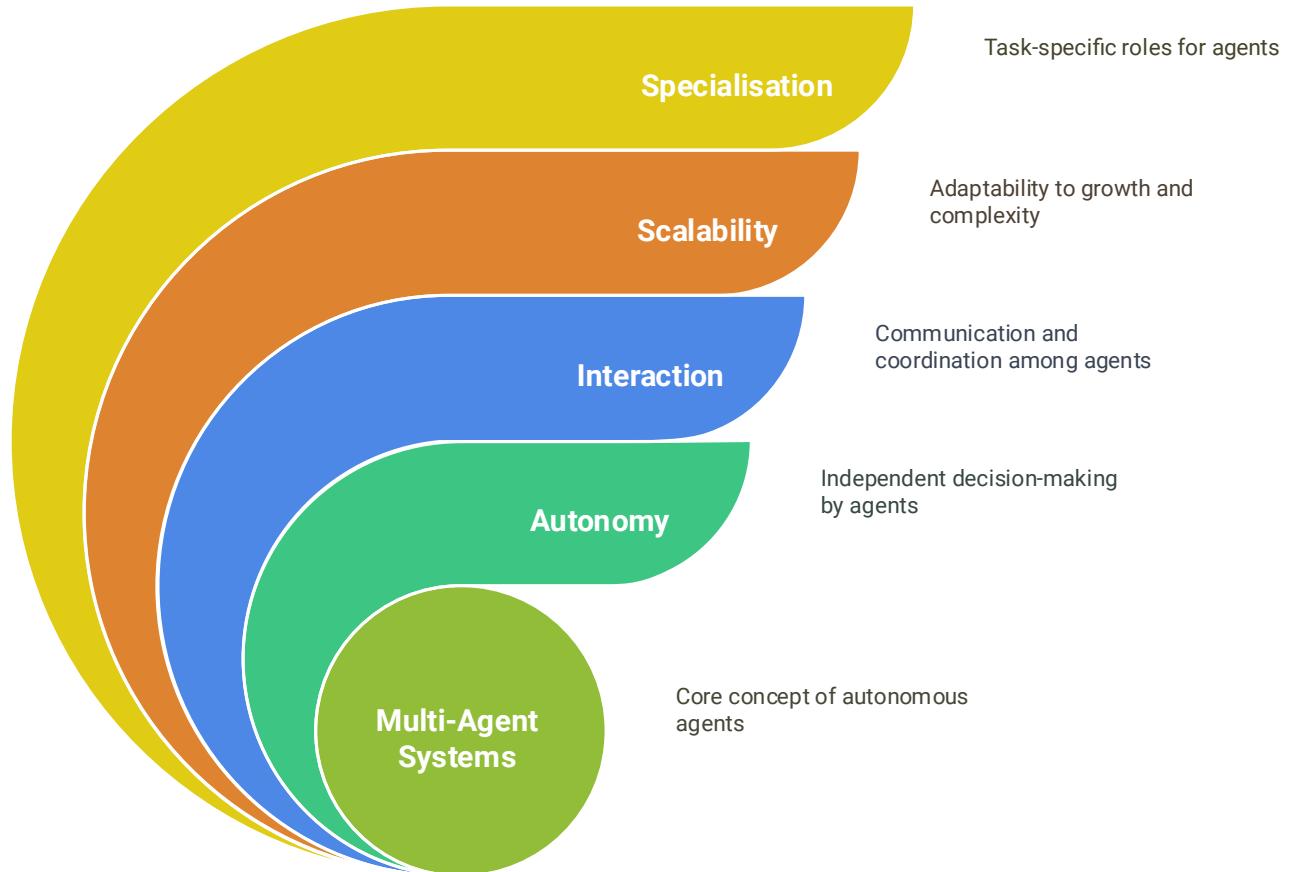


What are Multi-Agent Systems?



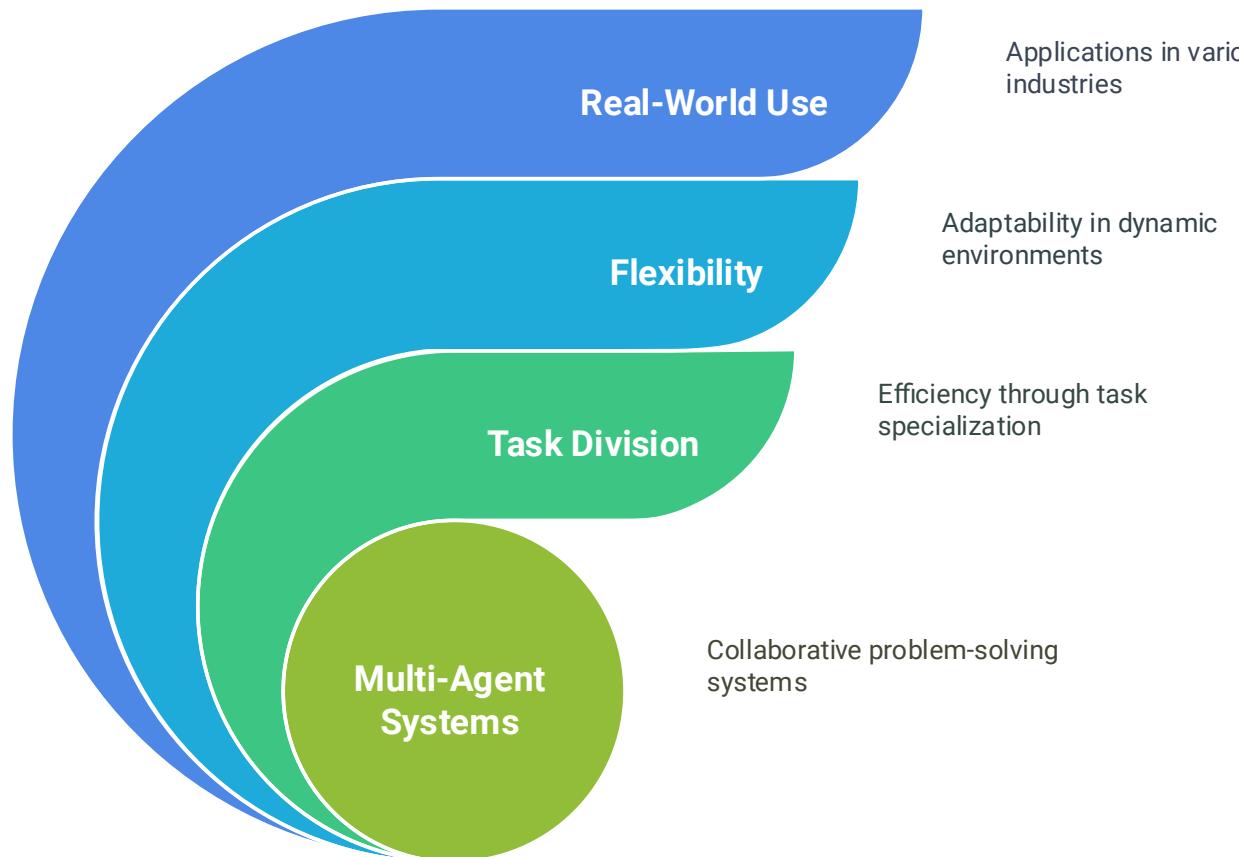
- ❖ **Collection of AI Agents Working Together**
Multi-agent systems consist of autonomous AI agents interacting within a **shared environment**.
- ❖ **Solve Complex Problems via Collaboration**
Agents communicate, share tasks, and combine insights to address challenges that may often **exceed the capability of any single agent**.
- ❖ **Each Agent Has a Specialised Role**
Roles may include sensing, planning, decision-making and execution, each tailored to the agent's strengths.
- ❖ **Example: Teams of Drones Coordinating Flight**
One drone maps terrain, another identifies obstacles, while others manage delivery route together forming a cohesive, intelligent swarm.

Features of Multi-Agent Systems



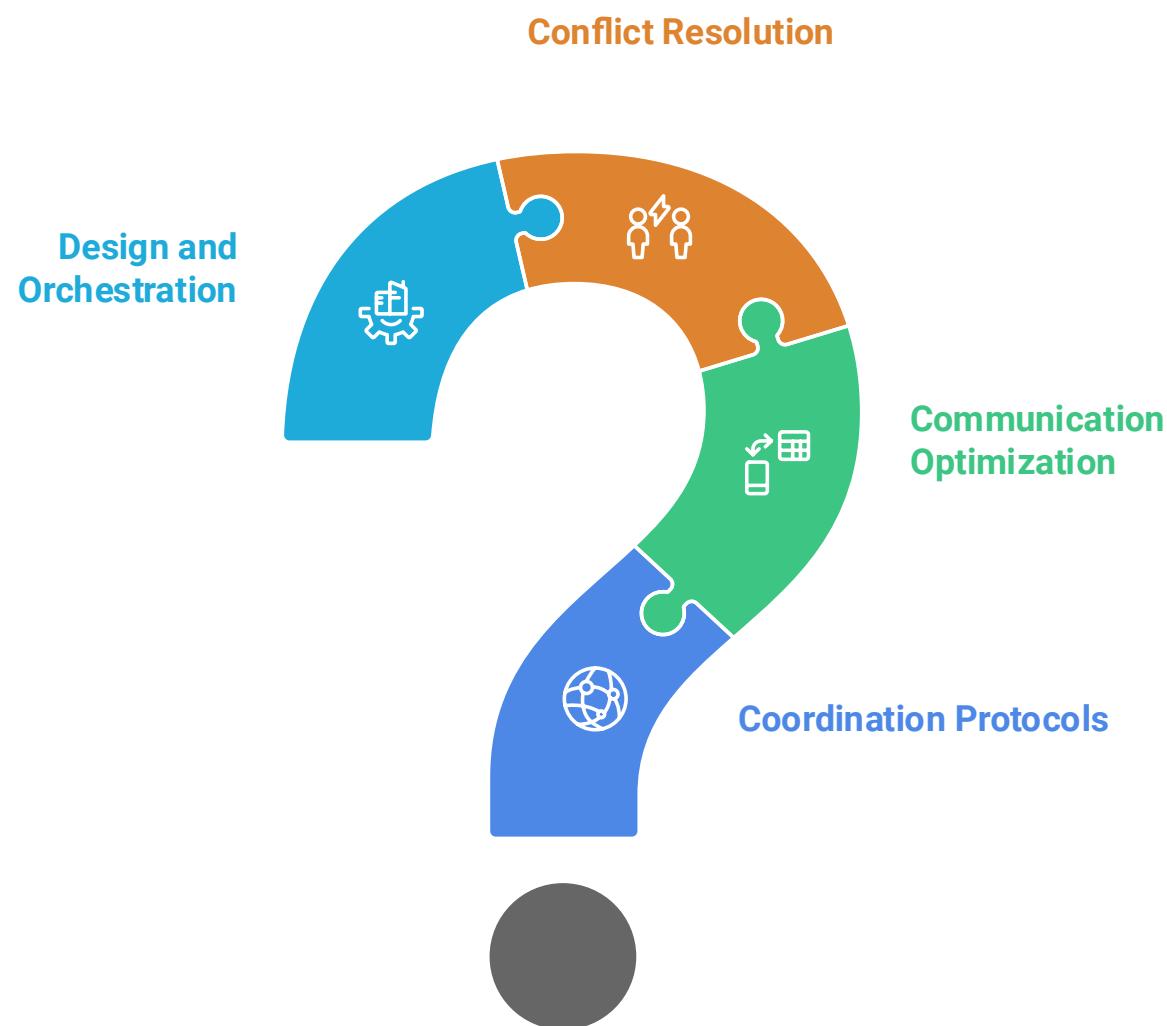
- ❖ **Specialisation:** Agents are designed for specific roles (e.g., data filtering, routing, monitoring), enabling efficient task distribution and parallel processing.
- ❖ **Scalability:** Systems can grow to include more agents, adapting to increased tasks or environments without central bottlenecks.
- ❖ **Interaction:** Agents exchange information, negotiate, and synchronise actions to achieve shared or individual objectives.
- ❖ **Autonomy:** Each agent operates without direct human control, making its own decisions based on goals and inputs.

Benefits of Multi-Agent Systems



- ❖ **Real-World Use:** Applied in robotics (e.g., swarm drones), gaming (NPC coordination), finance (market simulations), and energy grids (load balancing and optimisation).
- ❖ **Flexibility in Dynamic Environments:** Agents can adapt to changes, recover from failures, or reassign roles in real-time, making the system robust and resilient.
- ❖ **Improved Efficiency Through Task Division:** Specialised agents handle different parts of a workflow simultaneously, speeding up processing and reducing redundancy.
- ❖ **Tackle Problems Too Big for Single Agents:** Complex, large-scale tasks—such as disaster response or traffic management—can be broken down and solved collaboratively.

Challenges of Multi-Agent Systems



- ❖ **Coordination Can Be Chaotic**
Without clear protocols, agents may duplicate efforts, miss dependencies, or act at cross-purposes.
- ❖ **Communication Overhead**
Constant data exchange can strain bandwidth, slow down decisions, or introduce latency in fast-moving environments.
- ❖ **Potential for Conflicts Between Agents**
Agents may pursue conflicting goals or compete for limited resources, leading to inefficiencies or deadlocks.
- ❖ **Requires Careful Design and Orchestration**
Successful systems need well-defined roles, interaction rules, and fallback strategies to ensure stability and scalability.

Flowise AgentFlow V2



- ❖ **Modular Workflow Design:** AgentFlow V2 uses standalone, specialised nodes connected visually to define logic, data flow, and execution paths.
- ❖ **Advanced Orchestration:** Supports complex patterns like loops, branching, and human-in-the-loop interactions through a node-dependency and execution queue system.
- ❖ **Agent Collaboration:** Enables agent-to-agent task delegation and communication, with shared access to full conversation history for better context and coordination.
- ❖ **Human-in-the-Loop + Checkpoints:** Execution can pause for human input and resume later, supporting long-running, stateful workflows.
- ❖ **Shared State & Streaming:** Agents may exchange data across steps via Flow State and support real-time streaming of responses using Server-Sent Events (SSE).



Flowise AgentFlow V2 Patterns

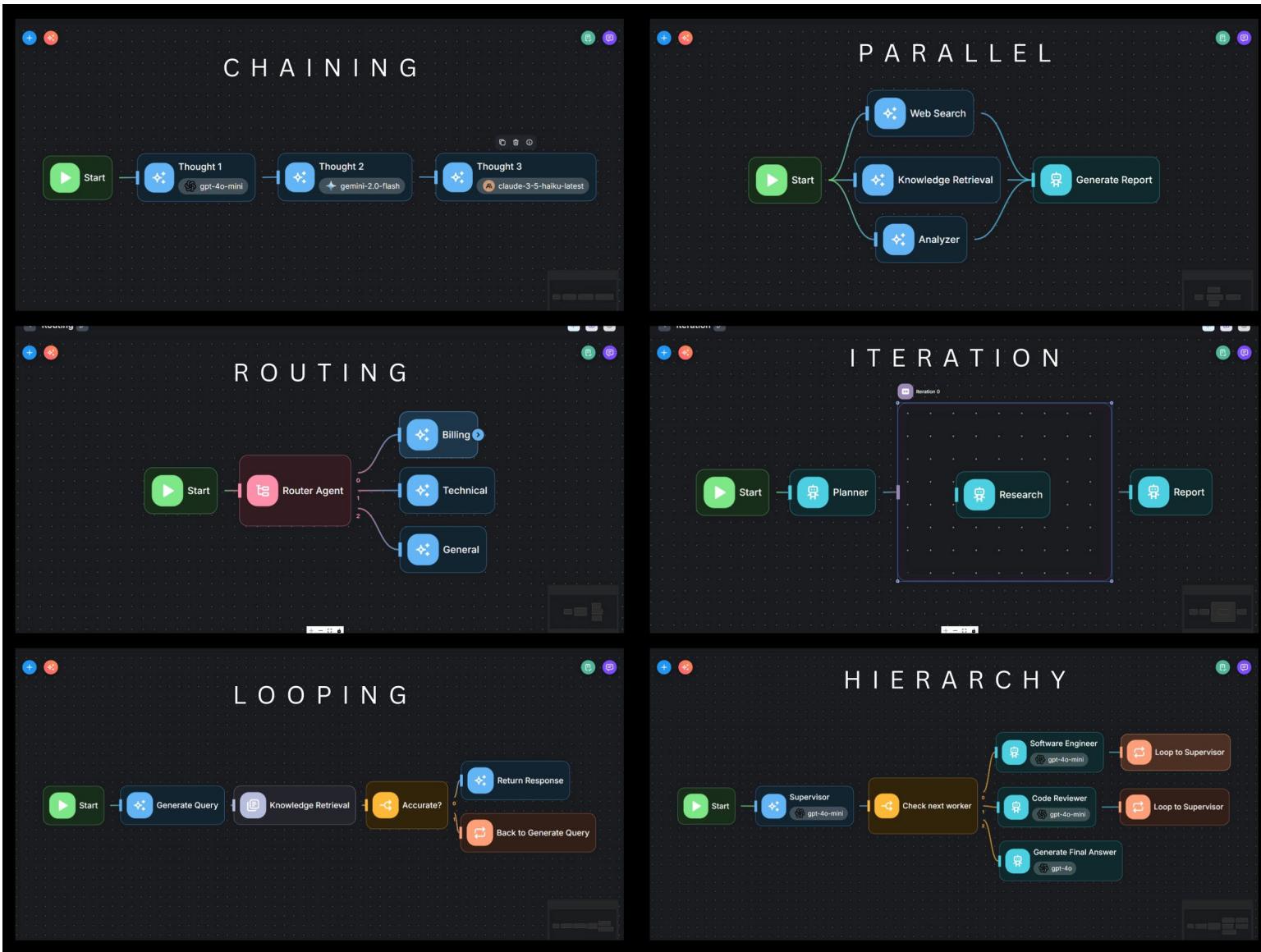


Image from <https://docs.flowiseai.com/using-flowise/agentflow2>



Flowise AgentFlow V2 Nodes

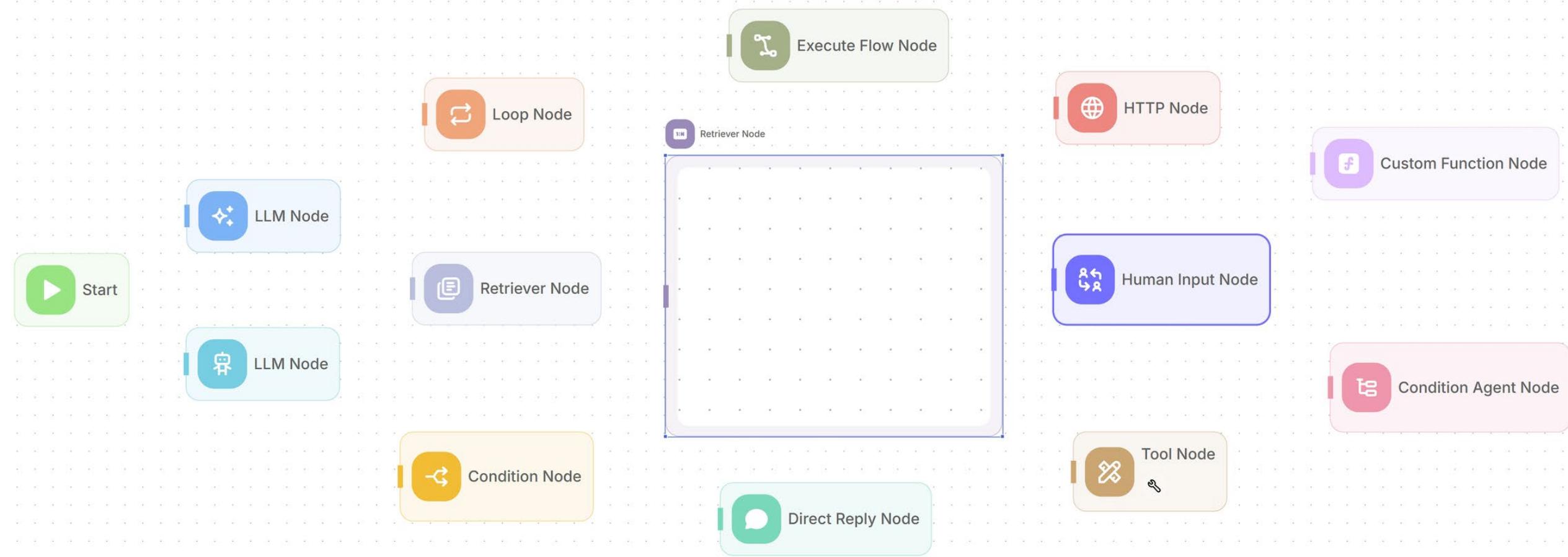


Image from <https://docs.flowiseai.com/using-flowise/agentflowv2>

Reference: <https://docs.flowiseai.com/using-flowise/agentflowv2>



Flowise

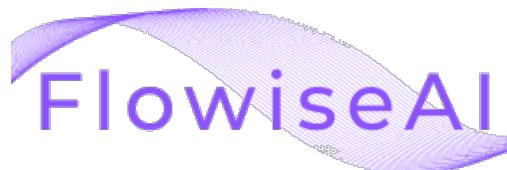
Activity





Activity: Flowise

- ❖ Two different methods to build a RAG system in Flowise covered today:
 - Custom Assistant (20 mins): Follow the detailed instructions included in the activity document entitled:
Flowise Assistant on Hugging Face. (Activity 1).docx
 - Multi-Agent Flow: Follow the detailed instructions included in the activity document entitled:
Flowise Multi-Agent on Hugging Face. (Activity 2).docx
- ❖ Don't forget to post to MST. Posting details are given in the activity documents.



Ai



Lesson 11

Summary of Material Covered



Lesson 11 Review

- ❖ **Multi-Agent Systems:** AI agents collaborate in shared environments to solve complex problems more effectively than individual agents, using specialisation, autonomy, and interaction.
- ❖ **Benefits & Challenges:** Multi-agent systems offer scalability, efficiency, and adaptability, but require careful coordination to avoid conflicts, communication overhead, and design complexity.
- ❖ **Flowise AgentFlow V2:** A modular architecture that enables visual design of sophisticated AI workflows, supporting agent collaboration, human input checkpoints, and real-time data exchange.
- ❖ **Workflow Patterns & Nodes:** Flowise supports patterns like chaining, parallelism, and loops, using specialised nodes (e.g., LLM Node, Human Input Node, Retriever Node) for flexible orchestration.
- ❖ **Practical Application:** Learners apply these concepts by building RAG systems in Flowise using either a Custom Assistant or Multi-Agent Flow, with structured activity guides provided.

Thank you

School of Infocomm

C240 AI Essentials and Innovations

© Republic Polytechnic 2025: All Rights Reserved