Advanced AI Techniques

Prompt Engineering Strategies & Mixture-of-Experts Architecture

What is Prompt Engineering?

The Art of Instruction

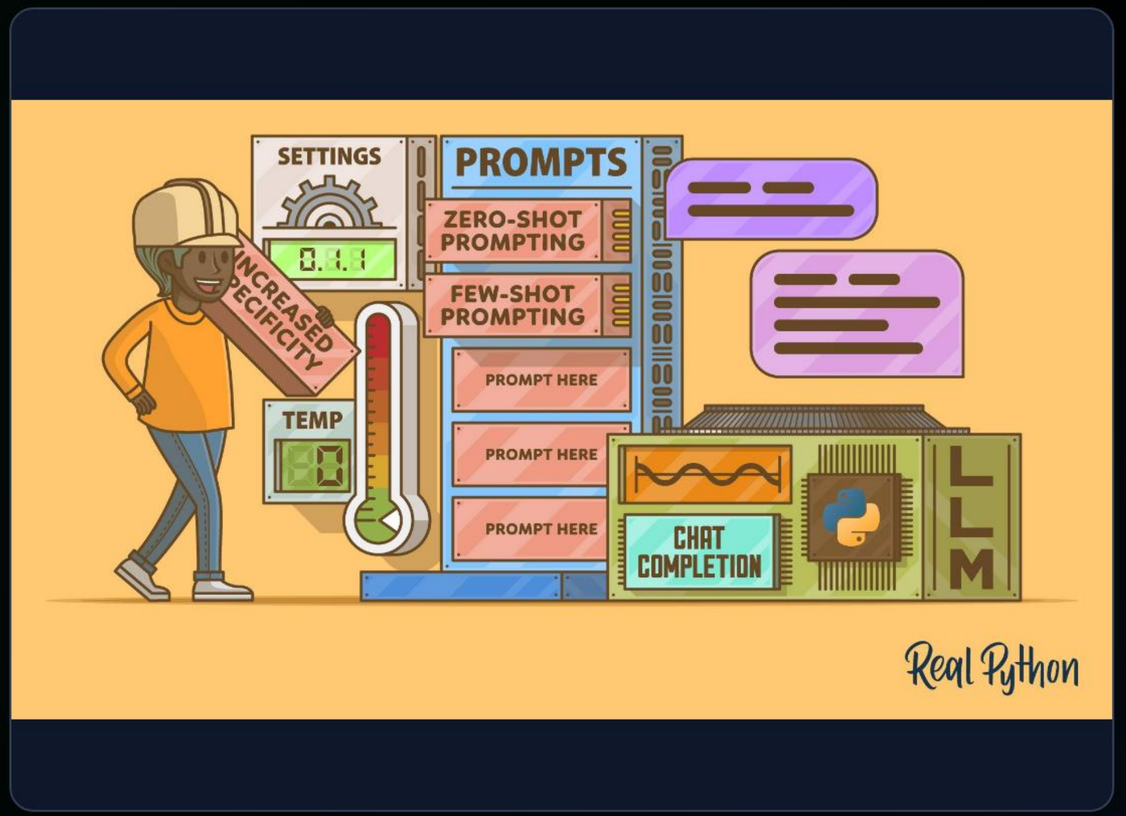
Prompt engineering is the iterative process of designing, refining, and optimizing inputs (prompts) to guide Generative AI models toward producing specific, high-quality outputs.

BASIC PROMPT

```
"Write an email about the project delay."
```

ENGINEERED PROMPT

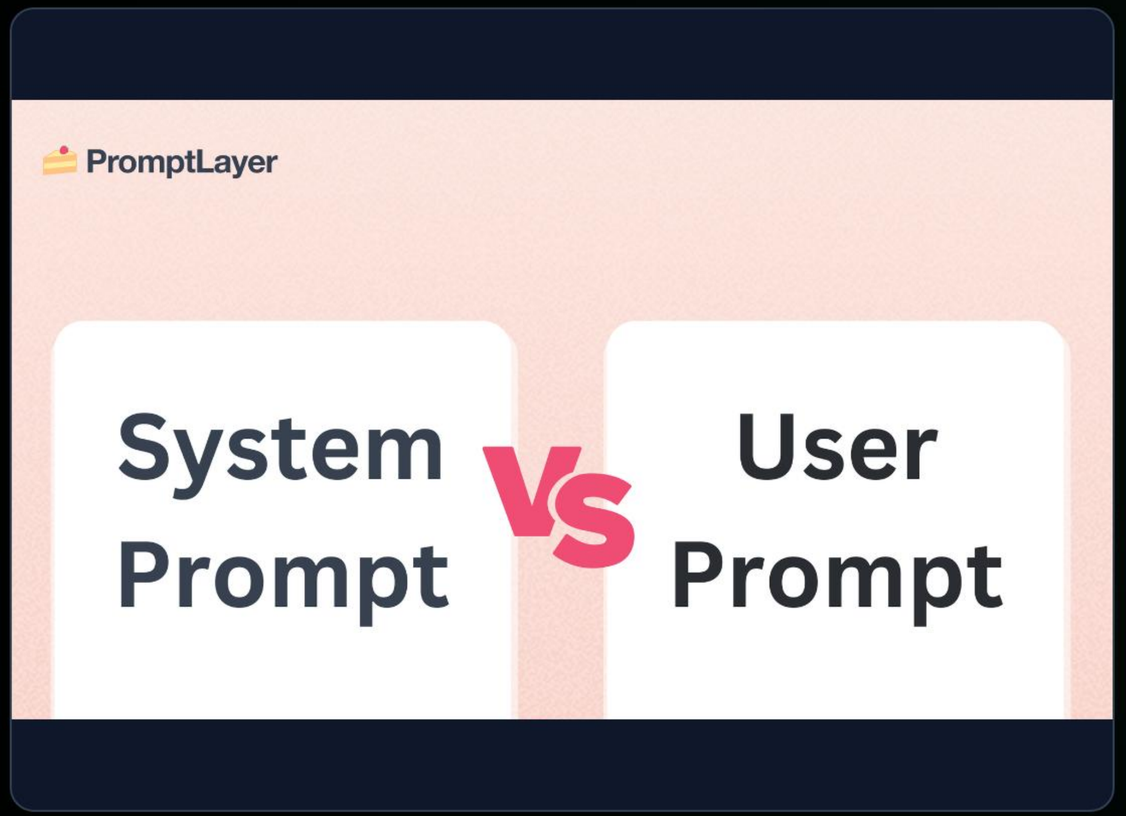
```
"Act as a professional Project Manager. Write a
polite but firm email to stakeholders explaining a
2-week delay due to supply chain issues, outlining
the mitigation plan."
```



Fundamental Prompting

The Building Blocks of LLM Interaction

System Prompting



Setting the "Rules of the Game"

System prompts are instructions provided "under the hood" before the user interaction begins. They define the model's persistent behavior, tone, and constraints.

- **Invisibility:** Often hidden from the end-user.
- **Security:** Used to prevent the model from answering illegal or unethical queries (Guardrails).
- **Consistency:** Ensures the model stays in character across a long conversation.

Role Prompting

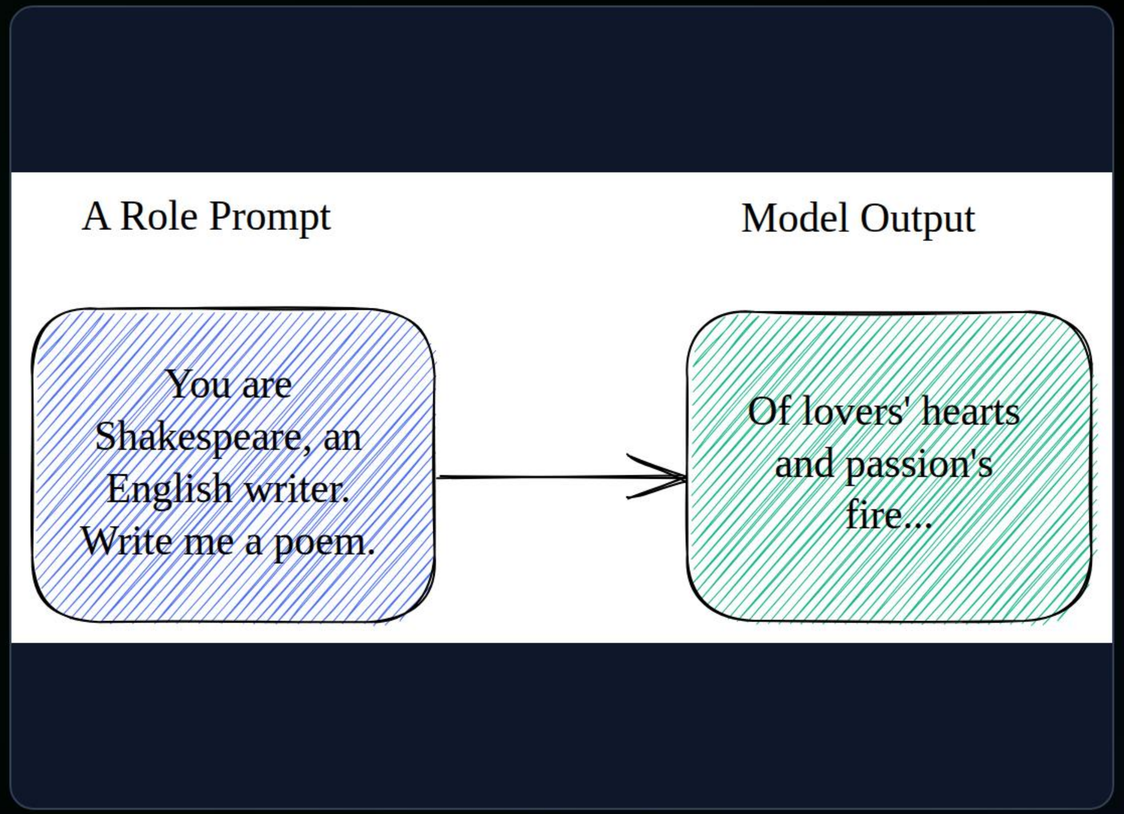
Assigning a Persona

Explicitly telling the AI "who" to be primes the model to access specific subsets of its training data associated with that role.

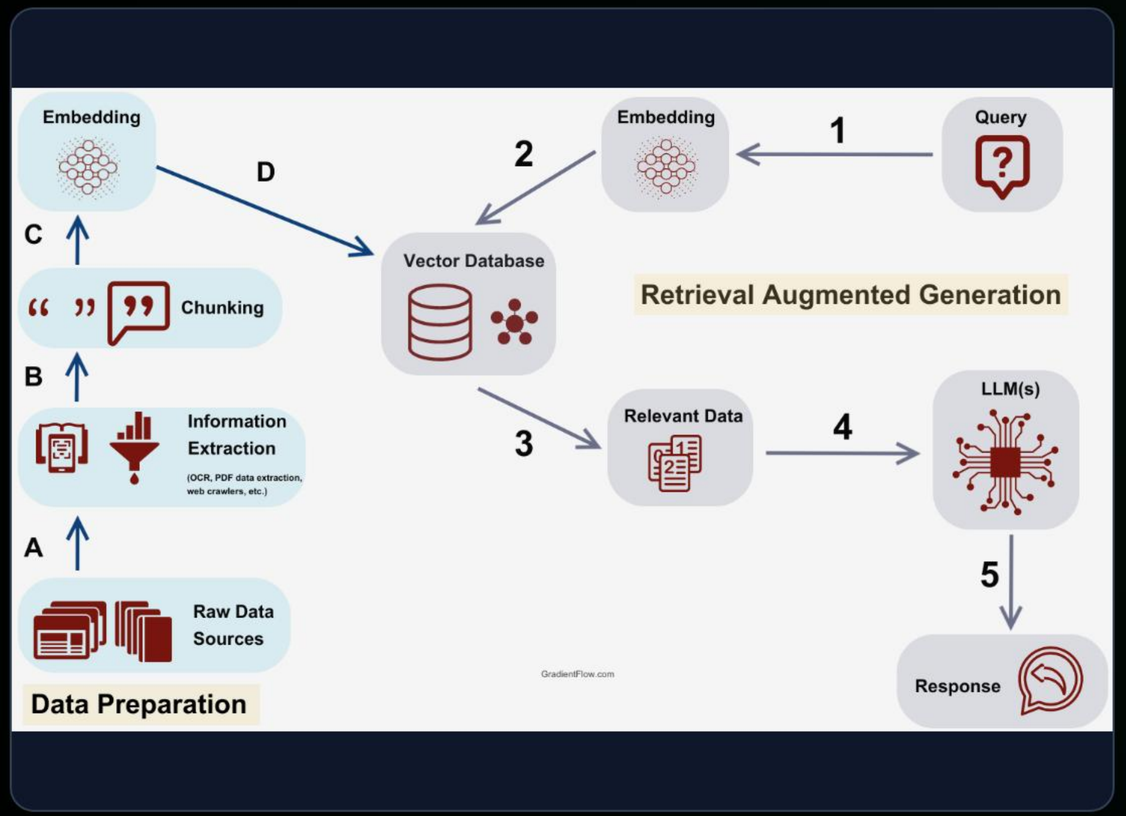
EXAMPLE

```
"You are a senior Javascript Developer with 10 years of experience. Explain asynchronous programming to a junior developer using a cooking analogy."
```

This technique significantly improves the relevance, tone, and depth of the response compared to a generic query.



Contextual Prompting



Grounding the Model

Contextual prompting involves injecting relevant information—such as documents, past conversation history, or data snippets—directly into the prompt.

- **Accuracy:** Forces the model to use provided "ground truth" rather than relying solely on training memory.
- **RAG (Retrieval-Augmented Generation):** The foundation of modern AI search, where relevant context is retrieved and added to the prompt dynamically.
- **Reduced Hallucination:** Minimizes made-up answers.

Zero-Shot vs. Few-Shot Learning

Zero-Shot Prompting

Relies entirely on the model's pre-trained knowledge without explicit examples.

- **Pros:** Fast, requires minimal token usage.
- **Cons:** Less reliable for complex formatting or niche tasks.
- **Example:** "Classify the sentiment of this text."

Few-Shot Prompting

Provides a set of input-output examples (shots) to guide the model's behavior.

- **Pros:** Significantly improves accuracy and adherence to specific formats.
- **Cons:** Consumes more context window (tokens).
- **Example:** Providing 3 examples of "Text -> Sentiment" before the query.

Advanced Strategies

Beyond Linear Reasoning

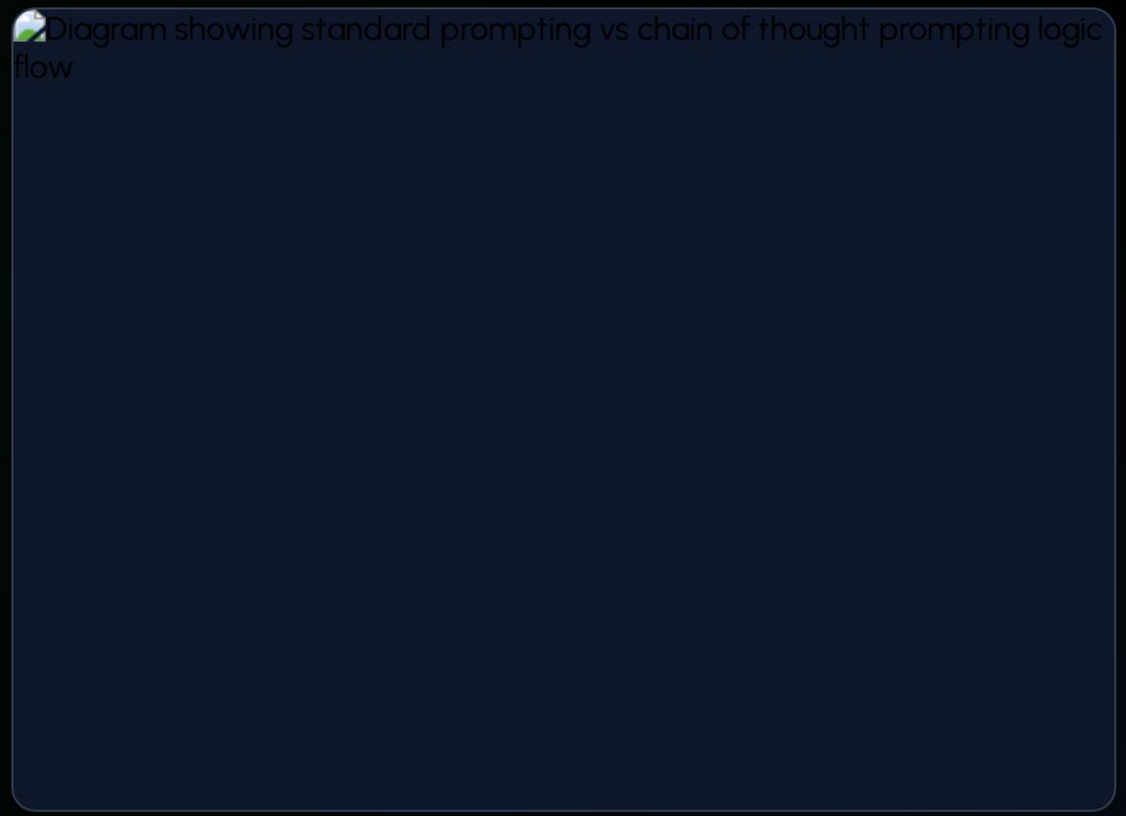
Chain-of-Thought (CoT)

Unlocking Reasoning

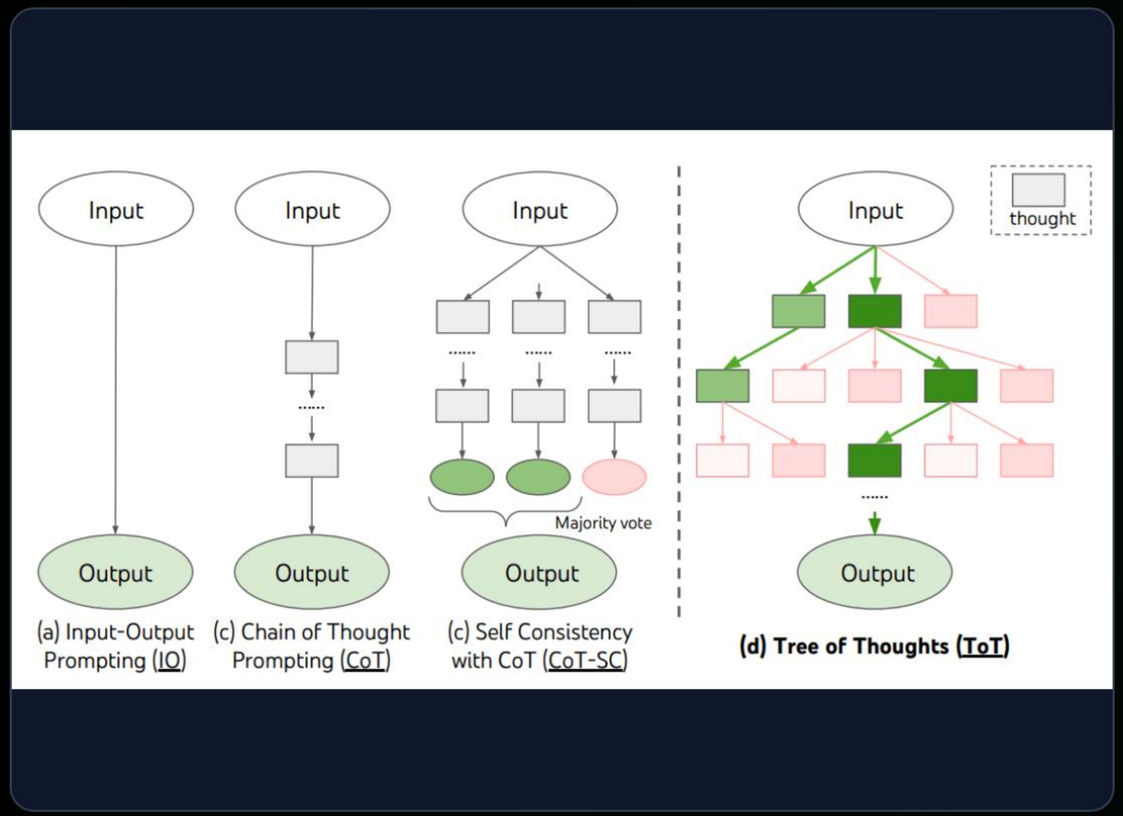
Standard prompting often fails at multi-step reasoning problems because the model tries to jump straight to the answer.

"Let's think step by step"

CoT forces the model to generate intermediate reasoning steps before the final answer. This decomposition allows the model to "debug" its own logic as it generates, leading to significantly higher performance on math and logic tasks.



Tree of Thoughts (ToT)



Branching Logic

While CoT is linear, Tree of Thoughts allows the model to explore multiple possible reasoning paths simultaneously.

- **Exploration:** Generates multiple "thoughts" at each step.
- **Evaluation:** Self-evaluates the promise of each branch.
- **Backtracking:** Can discard poor paths and return to a previous node, similar to how humans solve complex puzzles.

ReAct & Verification

ReAct (Reason + Act)

Interleaves reasoning traces with actions. The model "thinks" about what to do, performs an "action" (like a search), observes the result, and continues thinking.

Chain-of-Verification

A method to reduce hallucinations. The model generates a baseline response, then plans verification questions to check its own work, finally revising the answer.

Agentic Workflow

These strategies move models from passive text generators to active problem solvers capable of interacting with external environments.

Mixture-of-Experts

Scaling Efficiency with Sparse Architectures

The Router & The Experts

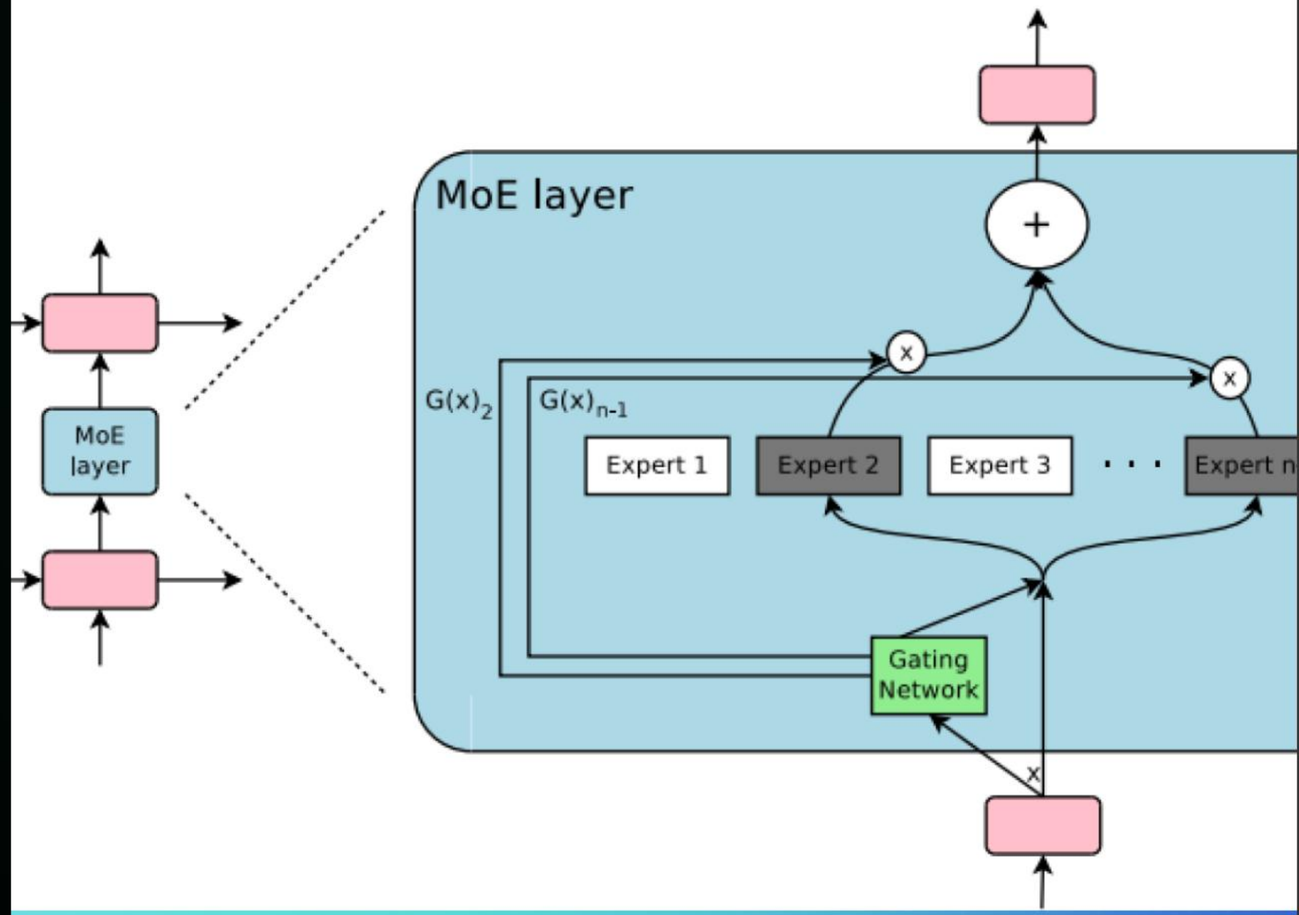
Conditional Computation

In a standard "dense" model, every parameter is used for every input. In a Mixture-of-Experts (MoE) model, a **Gating Network (Router)** determines which specific "experts" (sub-networks) are needed for a given token.

This means only a fraction of the total parameters are active at any one time, allowing for massive model capacity without proportional computational cost.

Hugging Face explains

re of Experts



Why Use Mixture-of-Experts?

Inference Speed

Because only a few experts are active per token, the model runs faster (lower latency) than a dense model of equivalent total size.

Scalability

Allows training of models with trillions of parameters. You can add more experts to increase knowledge capacity without exploding compute requirements.

Specialization

Different experts can learn to specialize in different domains (e.g., coding, creative writing, math), leading to better overall performance.

Case Study: Mixtral 8x7B

Sparse Mixture-of-Experts

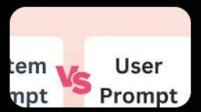
Mistral AI's **Mixtral 8x7B** is a prime example of high-performance MoE.

- **Structure:** It has 8 distinct expert networks.
- **Active Parameters:** For every token, the router selects only the top 2 experts.
- **Efficiency:** While it has 47B total parameters, it uses only ~13B parameters per token during inference.
- **Result:** Matches or beats Llama 2 70B (a dense model) while being 6x faster.

Image Sources

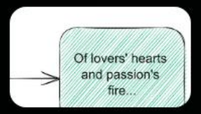
https://files.realpython.com/media/Prompt-Engineering-A-Practical-Example_Watermarked.7106fe3647aa.jpg

Source: realpython.com



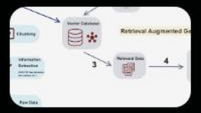
https://blog.promptlayer.com/content/images/2024/12/How-a-Prompt-Engineering-Tool-Improves-AI-Model-Performance--34-.png

Source: blog.promptlayer.com



https://learnprompting.org/docs/assets/basics/role_prompt.svg

Source: learnprompting.org



https://gradientflow.com/wp-content/uploads/2023/10/newsletter87-RAG-simple.png

Source: substack.com

https://cdn.prod.website-files.com/646e63db3a42c618e0a9935c/66d9fbfc84c87919aab55df6_66d9f915654c0c6d5e339110_Multiple%2520Chain%2520of%2520Thought%2520Examples.png

Source: www.prompthub.us

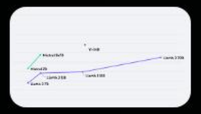
https://www.promptingguide.ai/_next/image?url=%2F_next%2Fstatic%2Fmedia%2FTOT.3b13bc5e.png&w=3840&q=75

Source: www.promptingguide.ai

Image Sources

https://huggingface.co/blog/assets/moe/thumbnail.png

Source: huggingface.co



https://assets.arkinvest.com/media-8e522a83-1b23-4d58-a202-792712f8d2d3/1ccd1b96-59c7-4779-9b5f-075038b03d56/ARK%20Invest_121823_Newsletter_Graph_Open%20Source%20Models.png

Source: www.ark-invest.com