MINERVA

AI4S
BSC AI 4 Science Fellowships

Adding Knowledge to LLMs

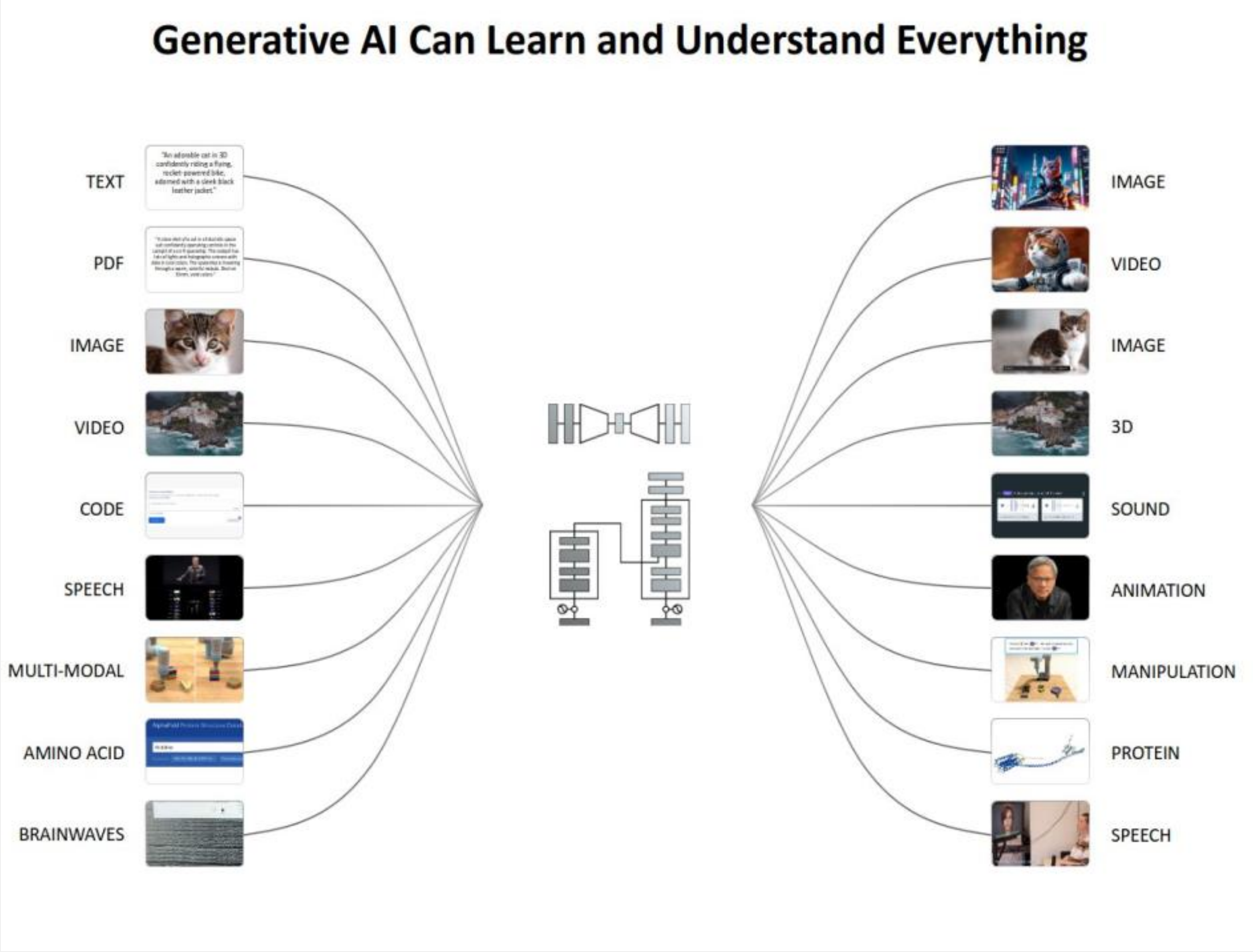
Alexandros Paliouras - AI Engineer
Guillem Cortiada Rovira - AI Engineer
Barcelona Supercomputer Center

Minerva AI Winter School, February 2026

Agenda

- Introduction
- What can we build with LLMs?
- Methods to Add Knowledge to LLMs
  1. Prompt Engineering
  2. Fine-Tuning
  3. Retrieval-Augmented Generation (RAG)
  4. Agents
- System-Level Trade-Offs
- Demo on Fine-Tuning and Agents
- Hands-On Notebooks

Introduction – Generative AI



Introduction – HPC and AI

HPCs

- Advancements to Exascale FLOPs
- Combinational architectures CPUs/GPUs
- Instrumental on science domains
    - Climate Modeling
    - Computational Chemistry
    - Biomedical Research
    - Astrophysical Simulations

EuroHPC Joint Undertaking

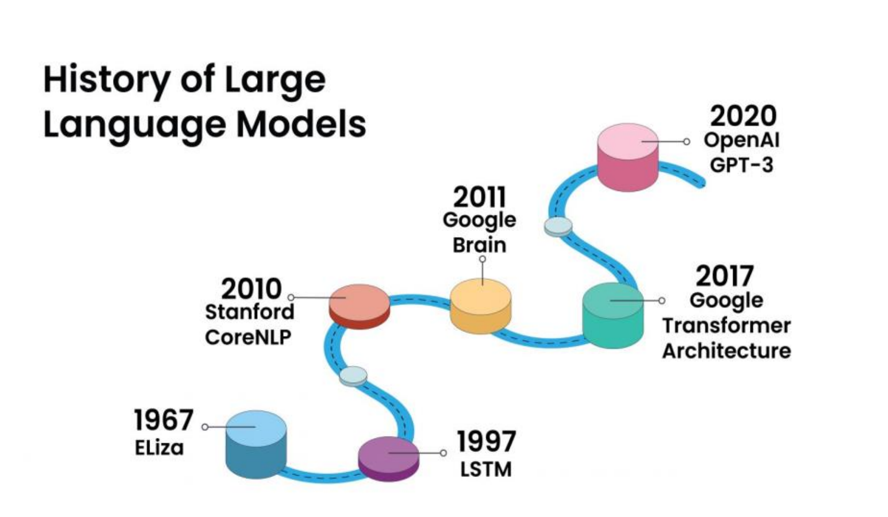
TOP 500 The List.

JUNE 2025	TOP500	Green500
JUPITER	#4	#1(JEDI) #21(Booster)
LUMI	#9	#36
LEONARDO	#10	#70
MARENOSTRUM 5	#14	#44
MELUXINA	#136	#79
KAROLINA	#196	#76
DISCOVERER	#260	#244
DEUCALION	#299	#116
VEGA	#307	#287

EuroHPC JU

LLMs

- Advancements in text & image understanding and generation
- Scale up on parameters (~70-600b params => 0.14-1.2 TB memory)
- Powerful tools can be attached to various domains
- Computational intensive
- Resource & Power greedy



Agenda

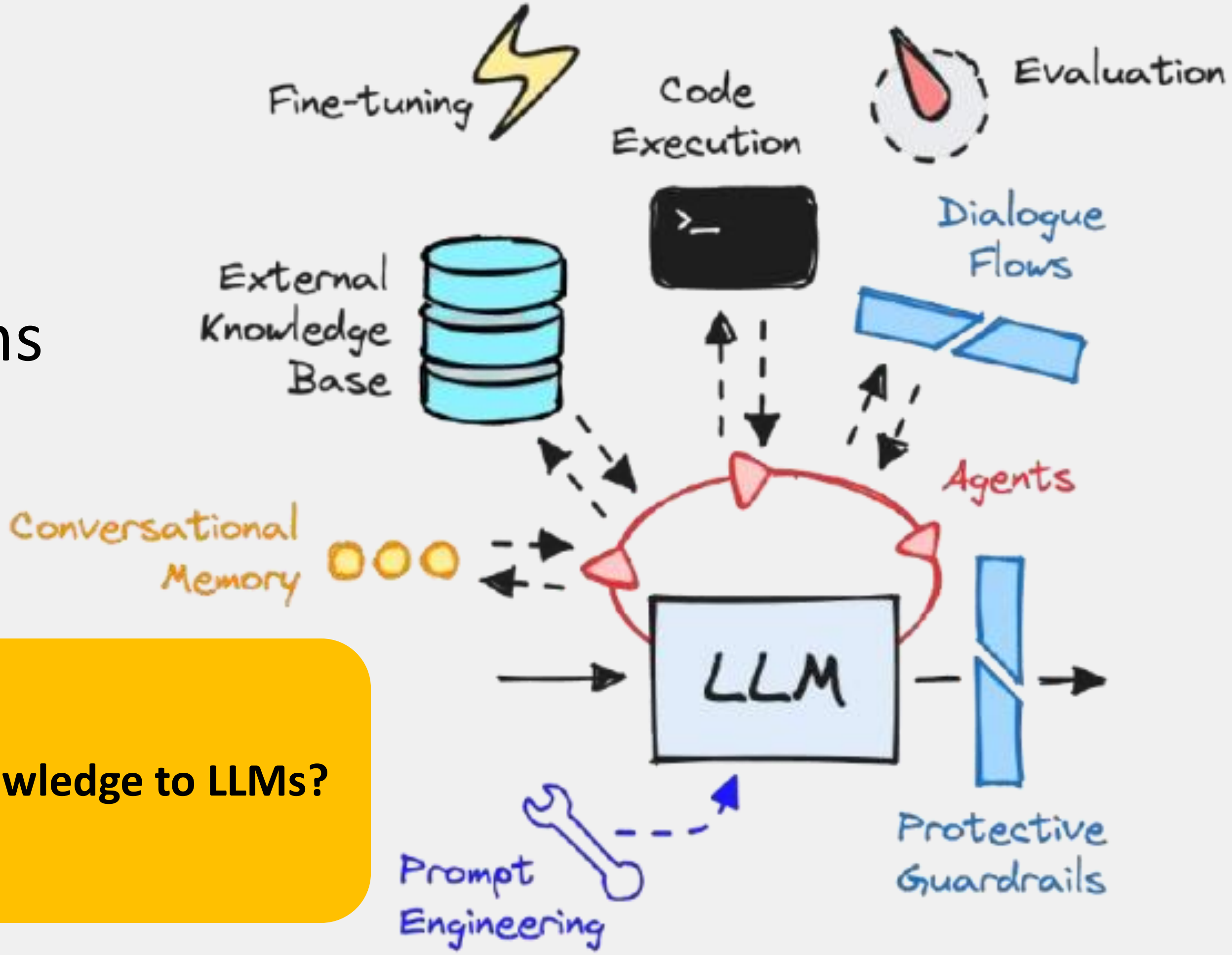
- Introduction
- **What can we build with LLMs?**
- Methods to Add Knowledge to LLMs
  1. Prompt Engineering
  2. Fine-Tuning
  3. Retrieval-Augmented Generation (RAG)
  4. Agents
- System-Level Trade-Offs
- Demo on Fine-Tuning and Agents
- Hand-On Notebooks

What can we build with LLMS?

- Conversational Systems
- Content Generation Tools
- Code & Developer Tools
- Knowledge & Research Systems
- Automation & Agents

What can we build with LLMS?

- Conversational Systems
- Content Generation Tools
- Code & Developer Tools
- Knowledge & Research Systems
- Automation & Agents



How do we add knowledge to LLMs?

Agenda

- Introduction
- What can we build with LLMs?
- **Methods to Add Knowledge to LLMs**
    1. Prompt Engineering
    2. Fine-Tuning
    3. Retrieval-Augmented Generation (RAG)
    4. Agents
- System-Level Trade-Offs
- Demo on Fine-Tuning and Agents
- Hands-On Notebooks

Prompt Engineering Example

Smart Chatbot – Customer Support Assistant

User Question

I can't log into my account. What should I do?

Basic Prompt

Answer the customer's question

You can try resetting your password. If that doesn't work, contact support.

Engineered Prompt

You are a professional and friendly customer support assistant for a SaaS company. Answer clearly and concisely. If the user reports a technical issue, provide step-by-step troubleshooting instructions. If you are unsure, politely suggest contacting support. Maintain a calm and helpful tone.

I'm sorry you're experiencing trouble logging in — I'll help you resolve this.
Please try the following steps:

- Click on "Forgot Password" on the login page.
- Check your email (including spam folder) for the reset link.
- Ensure your Caps Lock is off and your email address is entered correctly.

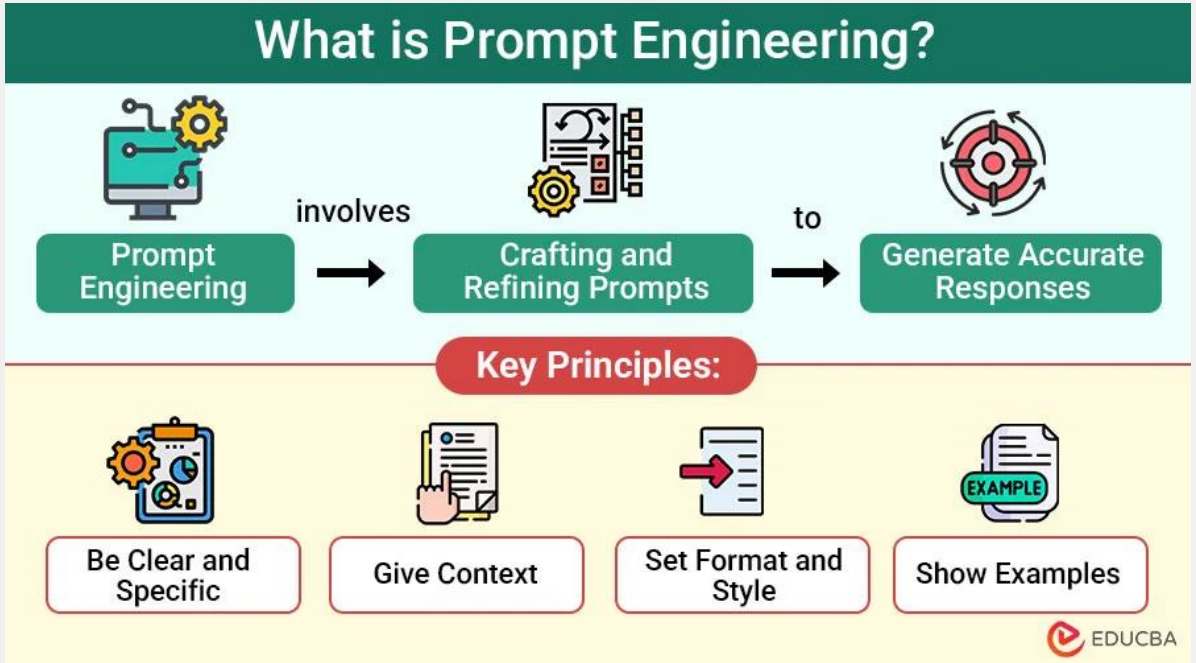
If the issue continues after resetting your password, please contact our support team, and we'll be happy to assist further.

Methods to Add Knowledge to LLMs – Prompt Engineering

- **LLMs are a black box. Prone to hallucination**, and **unable to provide up-to-date information.**
- Unlike traditional programming, where code executes exact instructions, **AI models generate output based on probabilities.**
- They **don't "*understand*" language** like humans.

Prompt Components:

- System Prompt
- Instructions
- Context (External Information)
- One-shot/Few-shot Prompting
- Query
- Answer



Methods to Add Knowledge to LLMs – Prompt Engineering

Full Prompt Example

System Prompt: You are an experienced data scientist.

Instructions: Analyze performance metrics and highlight anomalies.

Context: Use this system performance log: [insert log data].

Few-shot:

Q: Server uptime in Week 1

A: Uptime was 99.9%, no significant downtime detected.

Q: Error rate in Week 1

A: Error rate increased by 3%, primarily due to database timeouts.

Query: Server uptime and error rate in Week 2

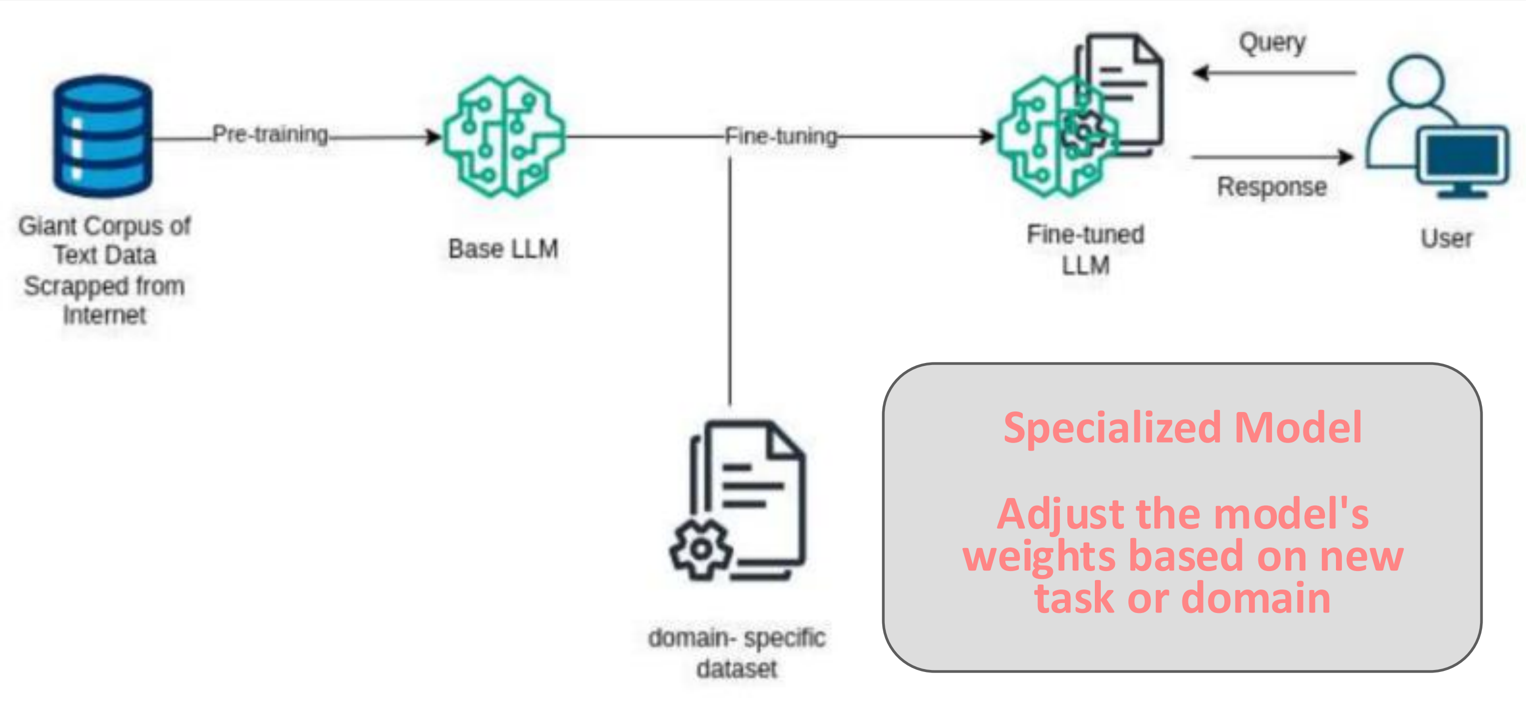
Answer:

Agenda

- Introduction
- What can we build with LLMs?
- **Methods to Add Knowledge to LLMs**
  1. Prompt Engineering
  2. **Fine-Tuning**
  3. Retrieval-Augmented Generation (RAG)
  4. Agents
- System-Level Trade-Offs
- Demo on Fine-Tuning and Agents
- Hands-On Notebooks

Methods to Add Knowledge to LLMs - Fine-Tuning

What is Fine-Tuning?



Pros:

- Better Task-Specific Performance
- Consistent Output
- Less Prompt engineering Needed
- Reduce Hallucinations
- Less Compute than pre-training

Cons:

- Requires High-Quality Data
- Costly and Time-Intensive
- Overfitting
- Reduced Flexibility

Methods to Add Knowledge to LLMs - Fine-Tuning

Pre-trained Model vs Fine-tuned Model

Dimension	Pre-trained Model	Fine-tuned Model
Training Data	Large, diverse internet-scale data	Additional domain/task-specific dataset
Purpose	General intelligence	Specialized performance
Behavior	Broad but generic	Tailored and consistent
Output Format Control	Limited	Strong and consistent
Domain Knowledge	General	Specialized
Cost to Build	Already trained	Requires additional training
Maintenance	None	Retraining may be needed
Flexibility	Very flexible	More narrow but precise
Best For	Chatbots, brainstorming, general Q&A	Classification, extraction, domain assistants

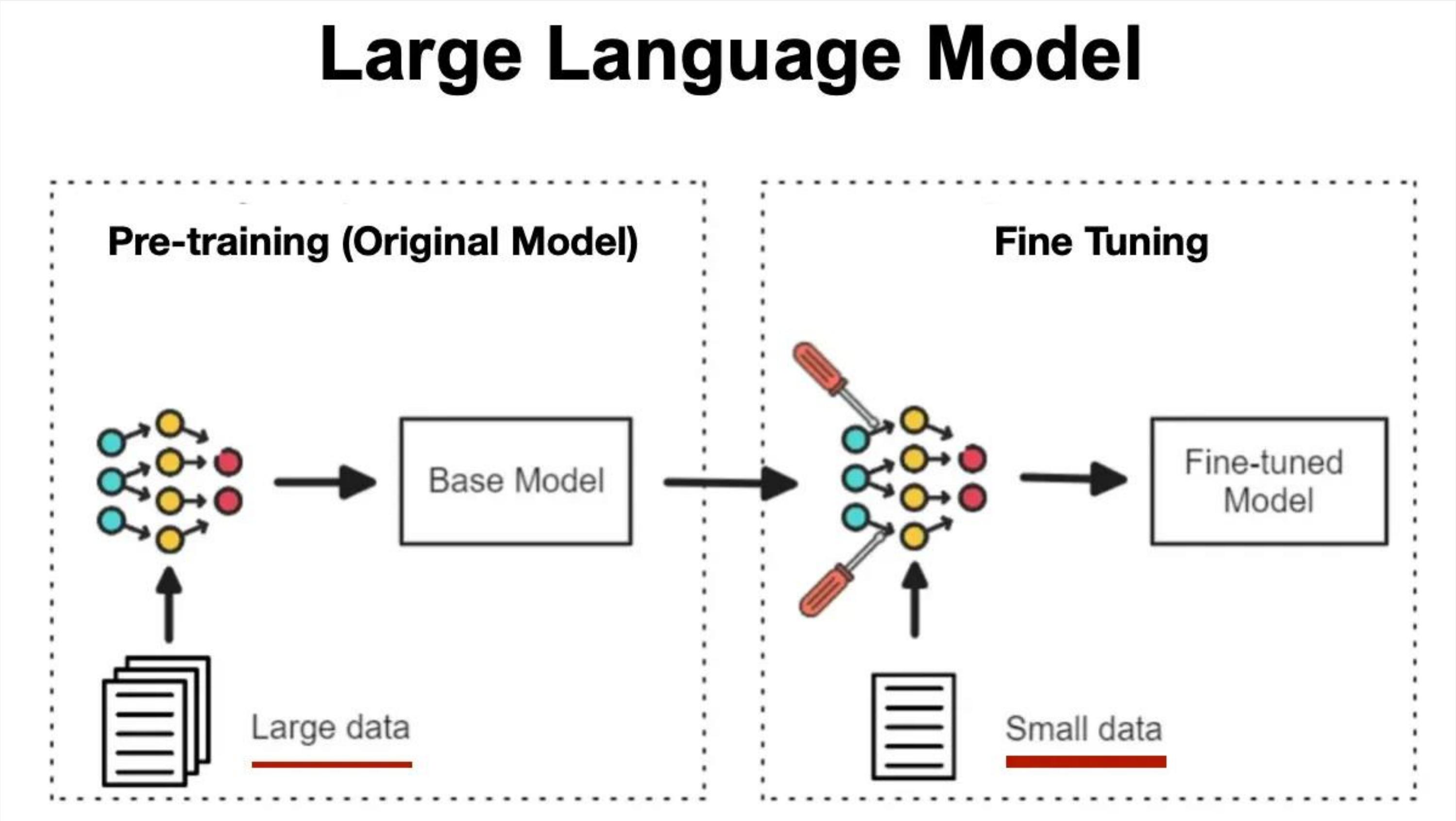
Methods to Add Knowledge to LLMs - Fine-Tuning

Types of Fine-Tuning:

- Full Fine-Tuning
- Parameter-Efficient Fine-Tuning (PEFT)
- Instruction Fine-Tuning
- Task-Specific Fine-Tuning
- Domain-Adaptive Fine-Tuning (DAFT)
- Reinforcement Learning from Human Feedback (RLHF)

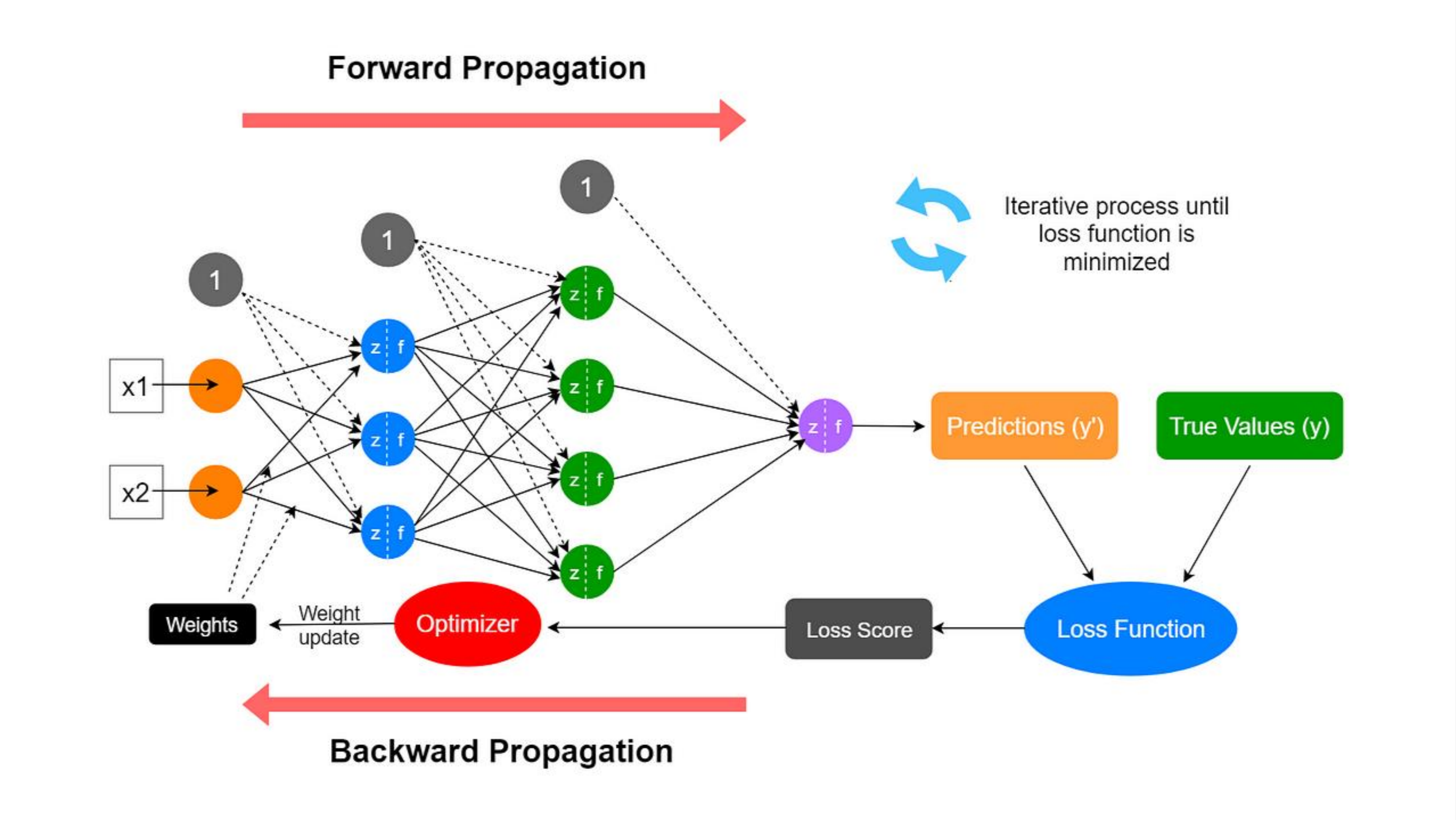
Methods to Add Knowledge to LLMs - Fine-Tuning

Full Fine-Tuning



Methods to Add Knowledge to LLMs - Fine-Tuning

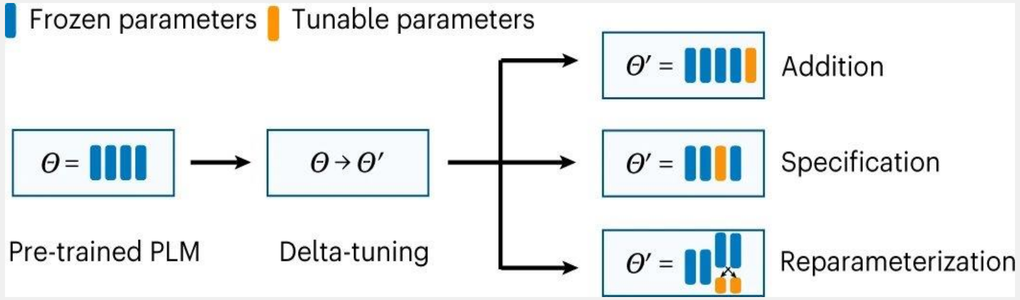
Full Fine-Tuning

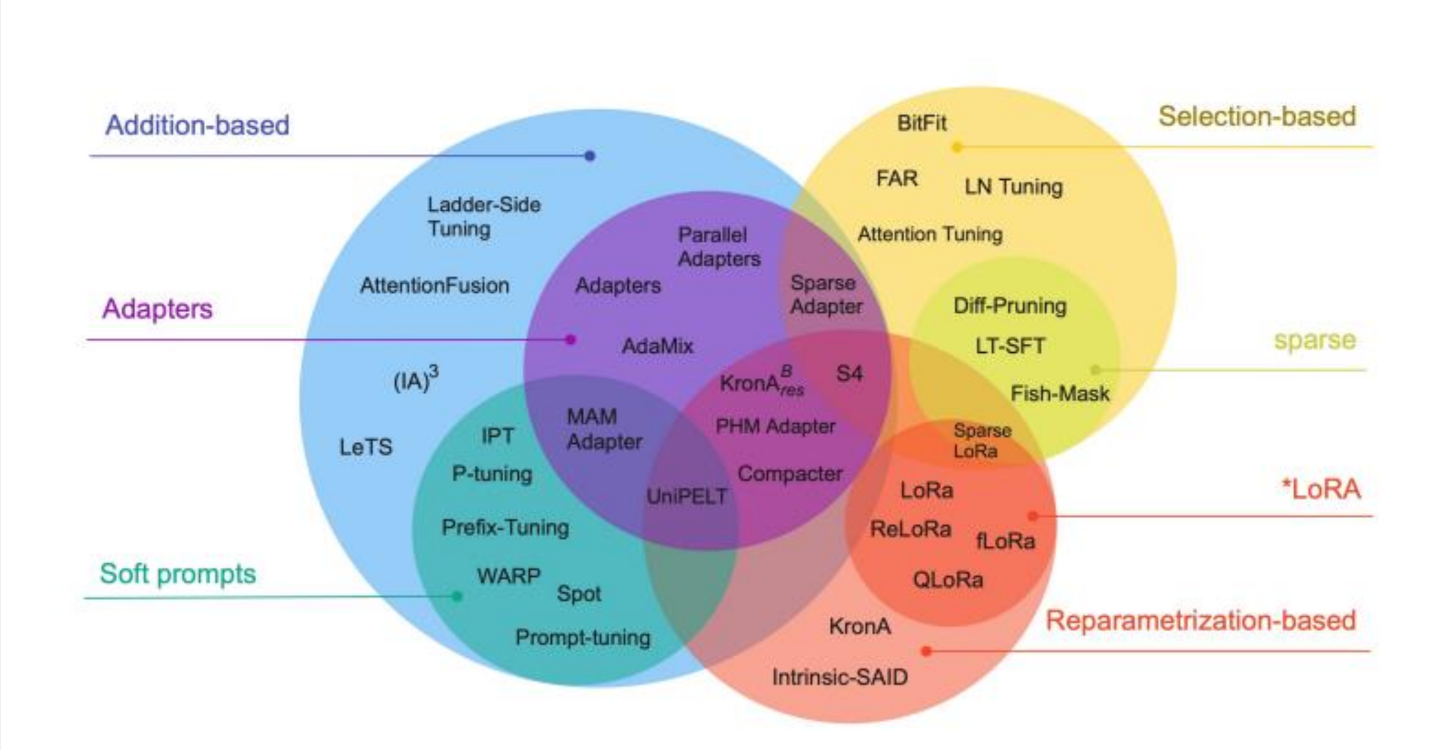


Methods to Add Knowledge to LLMs - Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT)

1. **Preserves** vast majority of model's original weights
2. 3 ways:
    - Additive
    - Selective
    - Reparametrization

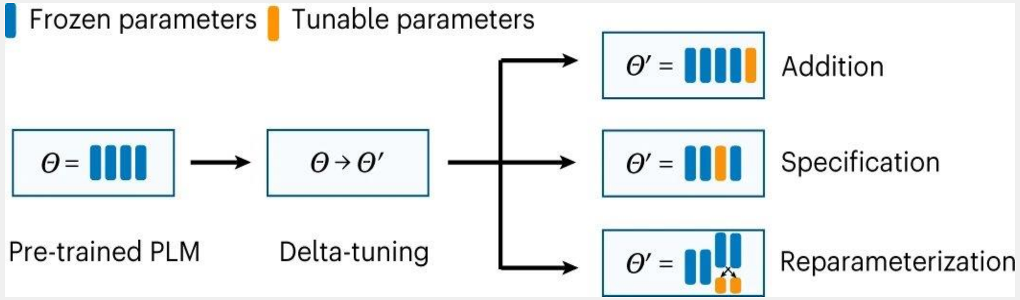


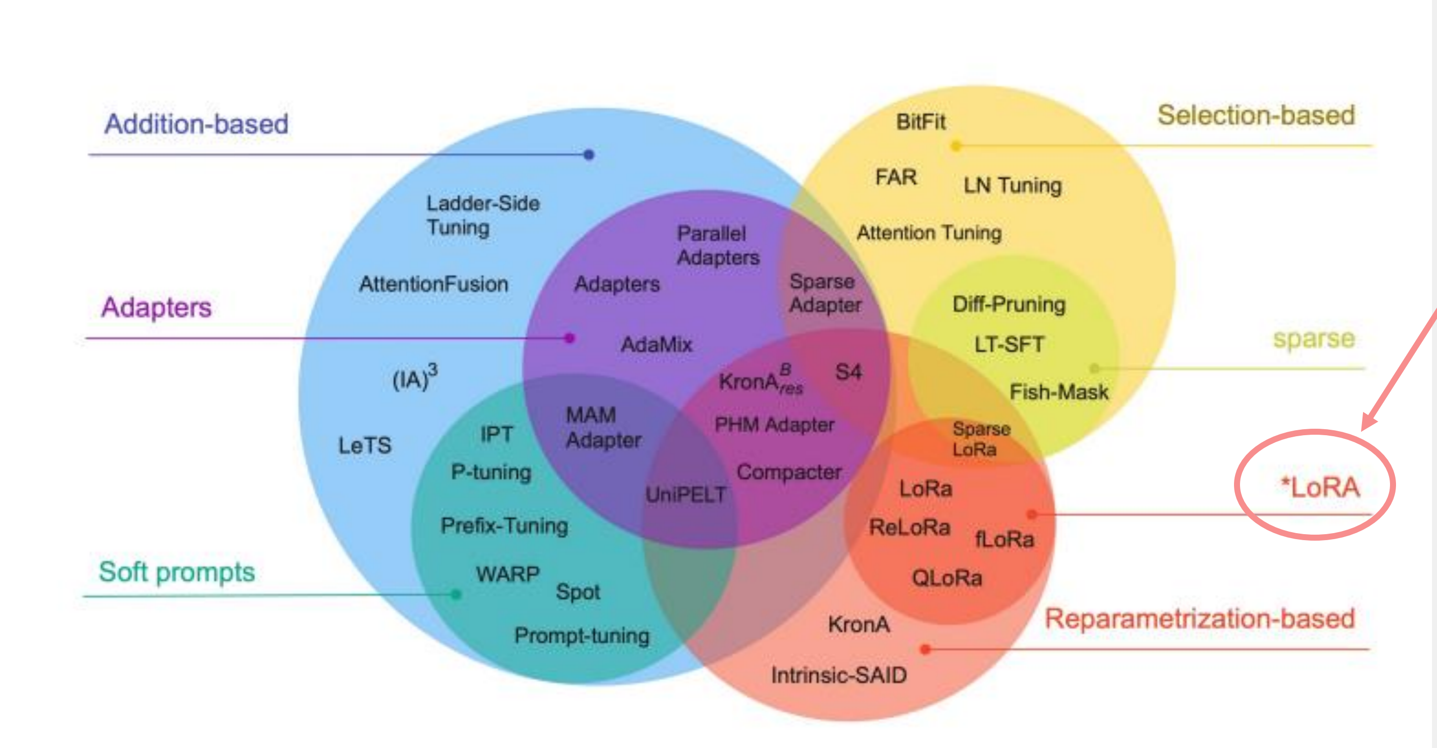


Methods to Add Knowledge to LLMs - Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT)

1. **Preserves** vast majority of model's original weights
2. 3 ways:
   - Additive
   - Selective
   - Reparametrization





Parameter-Efficient Fine-Tuning (PEFT) - LoRA

- Original Weights Frozen
- LoRA introduces the idea of **Matrix Decomposition** into its Low-Rank **Adaptation**
- LoRA injects two small matrices A and B to approximate weight updates.

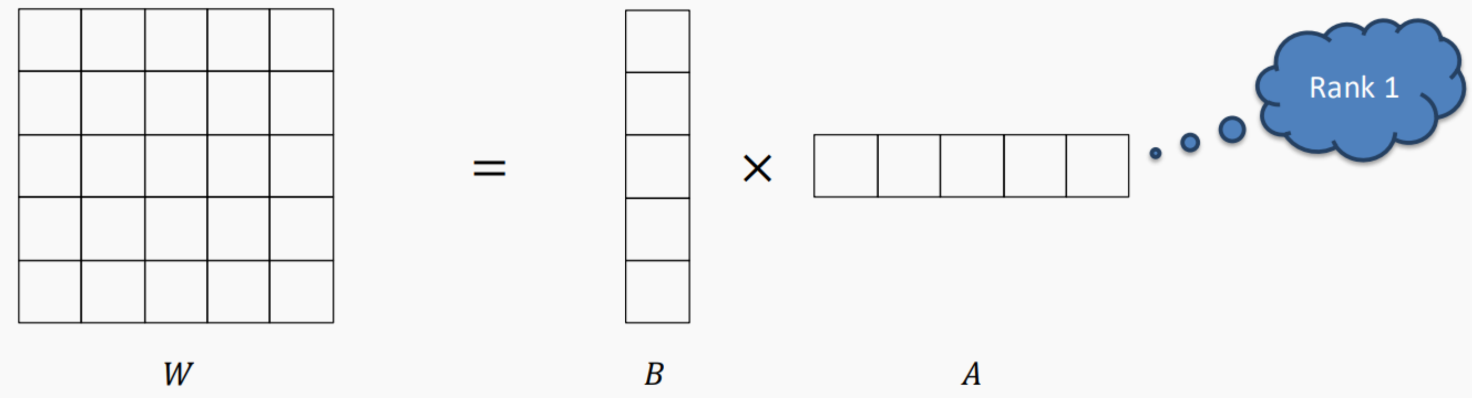
W (weight matrix) $\in R^{m \times n}$

can be decomposed into:

W = BA

where $B \in R^{m \times r}$

and $A \in R^{r \times n}$



Parameter-Efficient Fine-Tuning (PEFT) - LoRA

- Original Weights Frozen
- LoRA introduces the idea of **Matrix Decomposition** into its Low-Rank **Adaptation**
- LoRA injects two small matrices A and B to approximate weight updates.

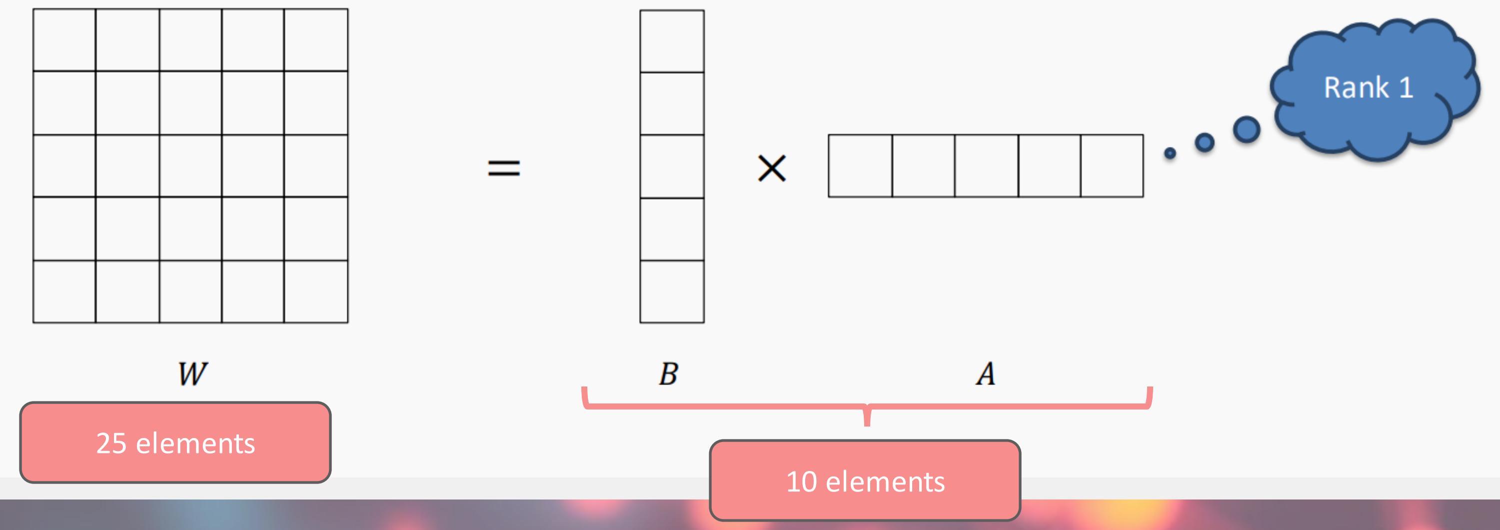
W (weight matrix) $\in R^{m\times n}$

can be decomposed into:

W = BA

where B $\in R^{m\times r}$

and A $\in R^{r\times n}$



Parameter-Efficient Fine-Tuning (PEFT) - LoRA

Update matrix

Scaling parameter

Decomposed matrices

Initial LLM Weights

$$W_0 + \Delta W = W_0 + \frac{\alpha}{r} BA$$

Rank of BA

Rank:

Determines the **size of these matrices.**

- **Low:** Fewer parameters → Faster, but possibly less expressive.
- **High:** More Parameters → Slower, more expressive and more memory.

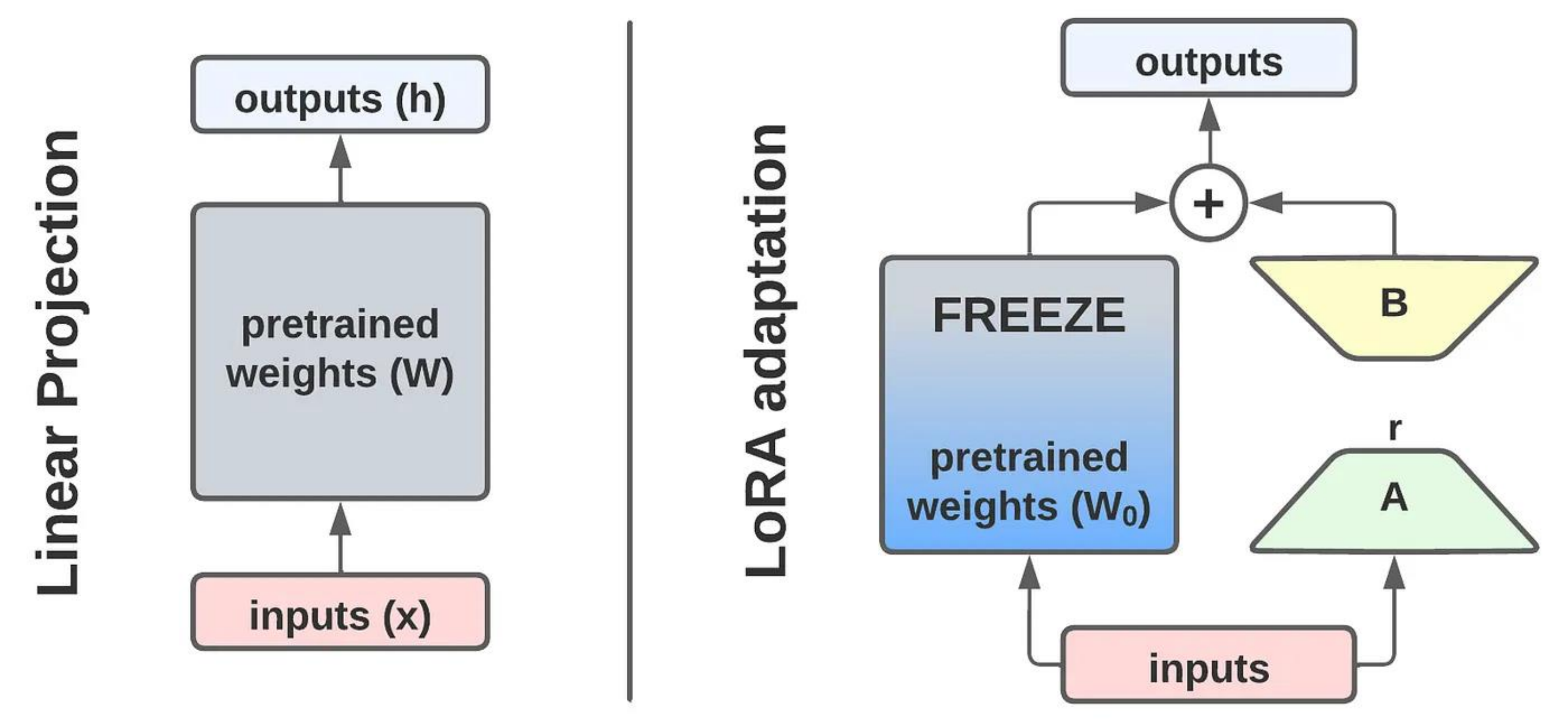
Scaling Parameter:

- Controls the magnitude of the LoRA update relative to the original weights.
- Allow us to **control the strength** of the LoRA contribution.

Methods to Add Knowledge to LLMs - Fine-Tuning

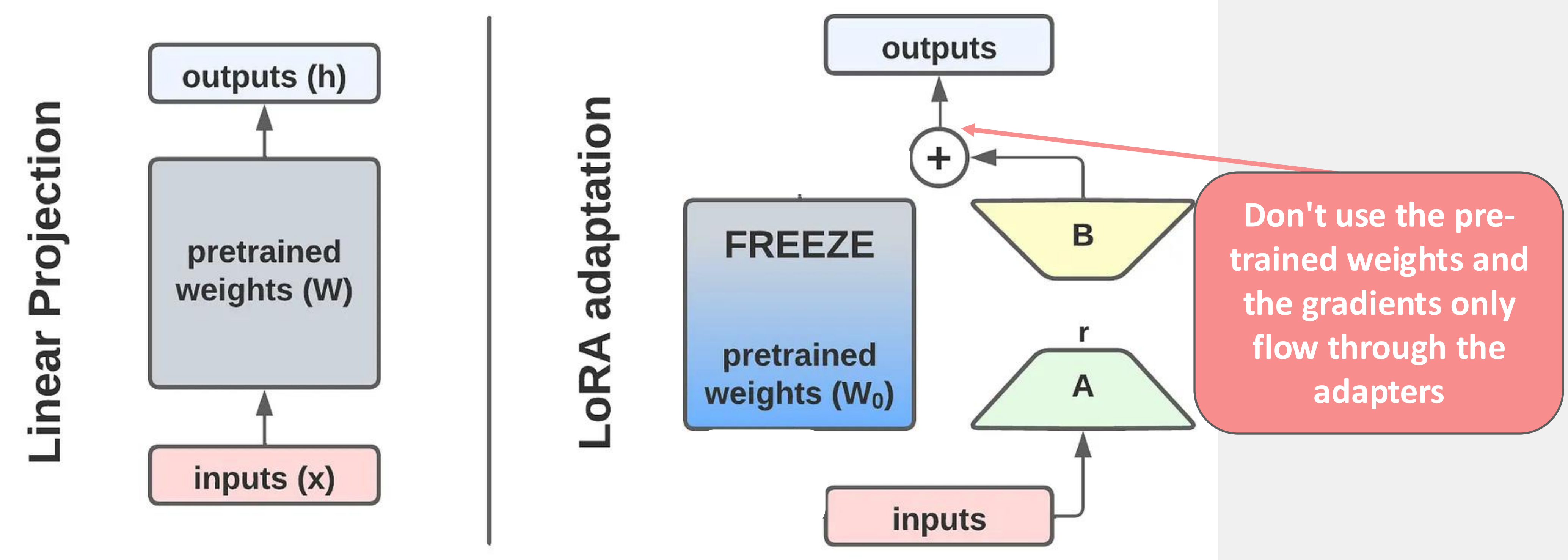
Parameter-Efficient Fine-Tuning (PEFT) - LoRA

Forward Pass



Parameter-Efficient Fine-Tuning (PEFT) - LoRA

Backward Pass and Optimizer Step



Methods to Add Knowledge to LLMs - Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT) - LoRA

Number of Trainable Parameters Table:

TABLE VI: Comparison of the average 5-shot MMLU test accuracy of LLaMA-7B and LLaMA-13B models fine-tuned with Alpaca. The higher the MMLU accuracy, the better. We also report total model parameters (# APs) and the ratio of trainable parameters.

Model	PEFT Method	# TPs	# APs	% Params	5-shot MMLU Accuracy
LLaMA-7B-Alpaca	FT	6738.4M	6738.4M	100	**41.79**
	$(IA)^3$	1.58M	6740.0M	0.02	37.88
	LoRA	159.9M	6898.3M	2.32	40.67
	QLoRA	79.9M	3660.3M	2.18	39.96
LLaMA-13B-Alpaca	FT	13015.9M	13015.9M	100	**49.60**
	$(IA)^3$	2.48M	13018.3M	0.02	47.42
	LoRA	250.3M	13266.2M	1.88	47.49
	QLoRA	125.2M	6922.3M	1.81	47.29

Source: Xu, L., Xie, H., Qin, S. J., Tao, X., & Wang, F. L. (2026). Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Methods to Add Knowledge to LLMs - Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT) - LoRA

Number of Trainable Parameters Table:

TABLE VI: Comparison of the average 5-shot MMLU test accuracy of LLaMA-7B and LLaMA-13B models fine-tuned with Alpaca. The higher the MMLU accuracy, the better. We also report total model parameters (# APs) and the ratio of trainable parameters.

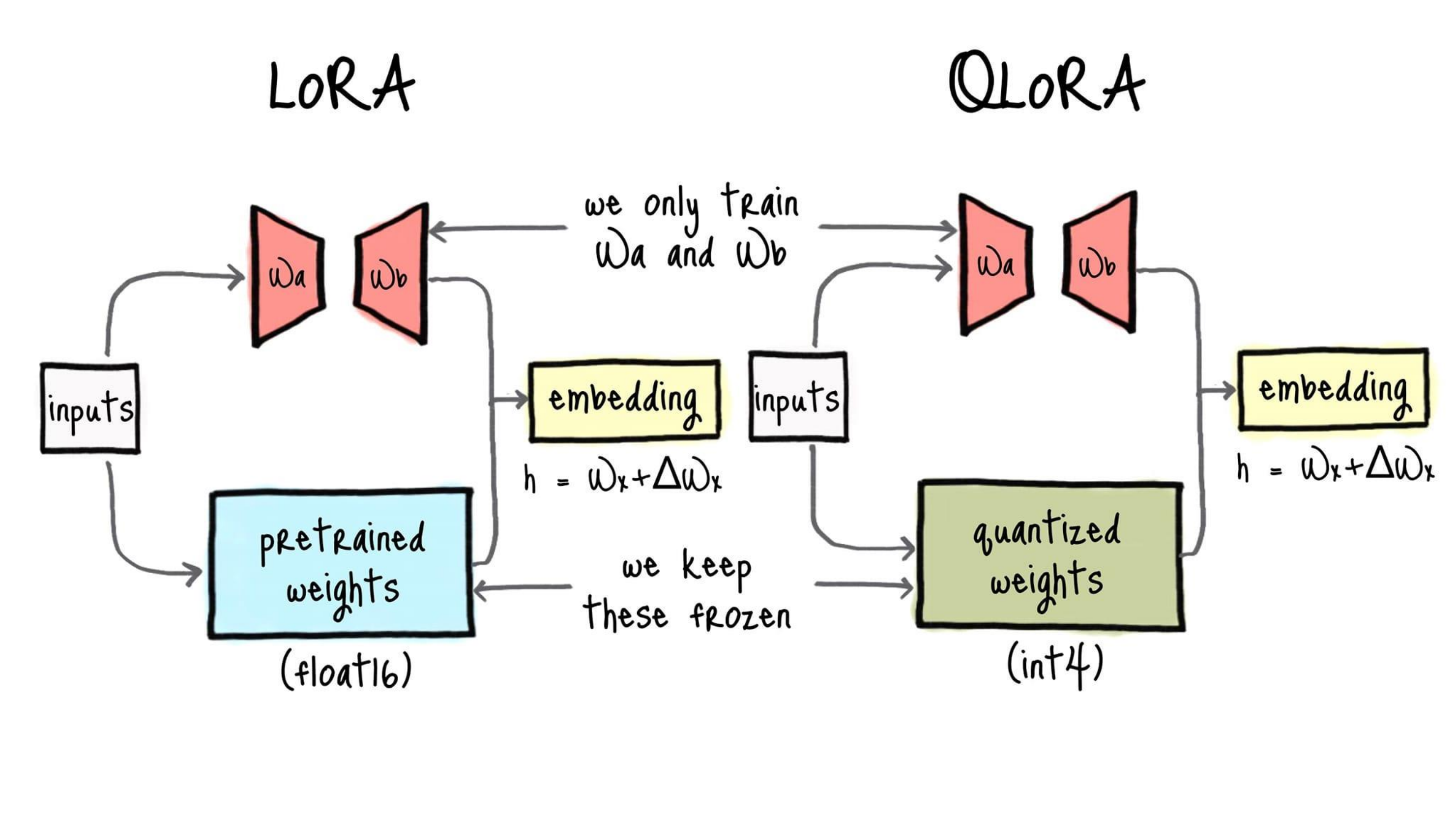
Model	PEFT Method	# TPs	# APs	% Params	5-shot MMLU Accuracy
LLaMA-7B-Alpaca	FT	6738.4M	6738.4M	100	**41.79**
	$(IA)^3$	1.58M	6740.0M	0.02	37.88
	LoRA	159.9M	6898.3M	2.32	40.67
	QLoRA	79.9M	3660.3M	2.18	39.96
LLaMA-13B-Alpaca	FT	13015.9M	13015.9M	100	**49.60**
	$(IA)^3$	2.48M	13018.3M	0.02	47.42
	LoRA	250.3M	13266.2M	1.88	47.49
	QLoRA	125.2M	6922.3M	1.81	47.29

Source: Xu, L., Xie, H., Qin, S. J., Tao, X., & Wang, F. L. (2026). Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Parameter-Efficient Fine-Tuning (PEFT) - LoRA

Aspect	PEFT	Full Fine-Tuning
Parameters updated	Small subset	All
GPU memory	Low	Very high
Training speed	Faster	Slower
Performance	Very strong	Maximum possible
Catastrophic Forgetting	Less prone	More prone

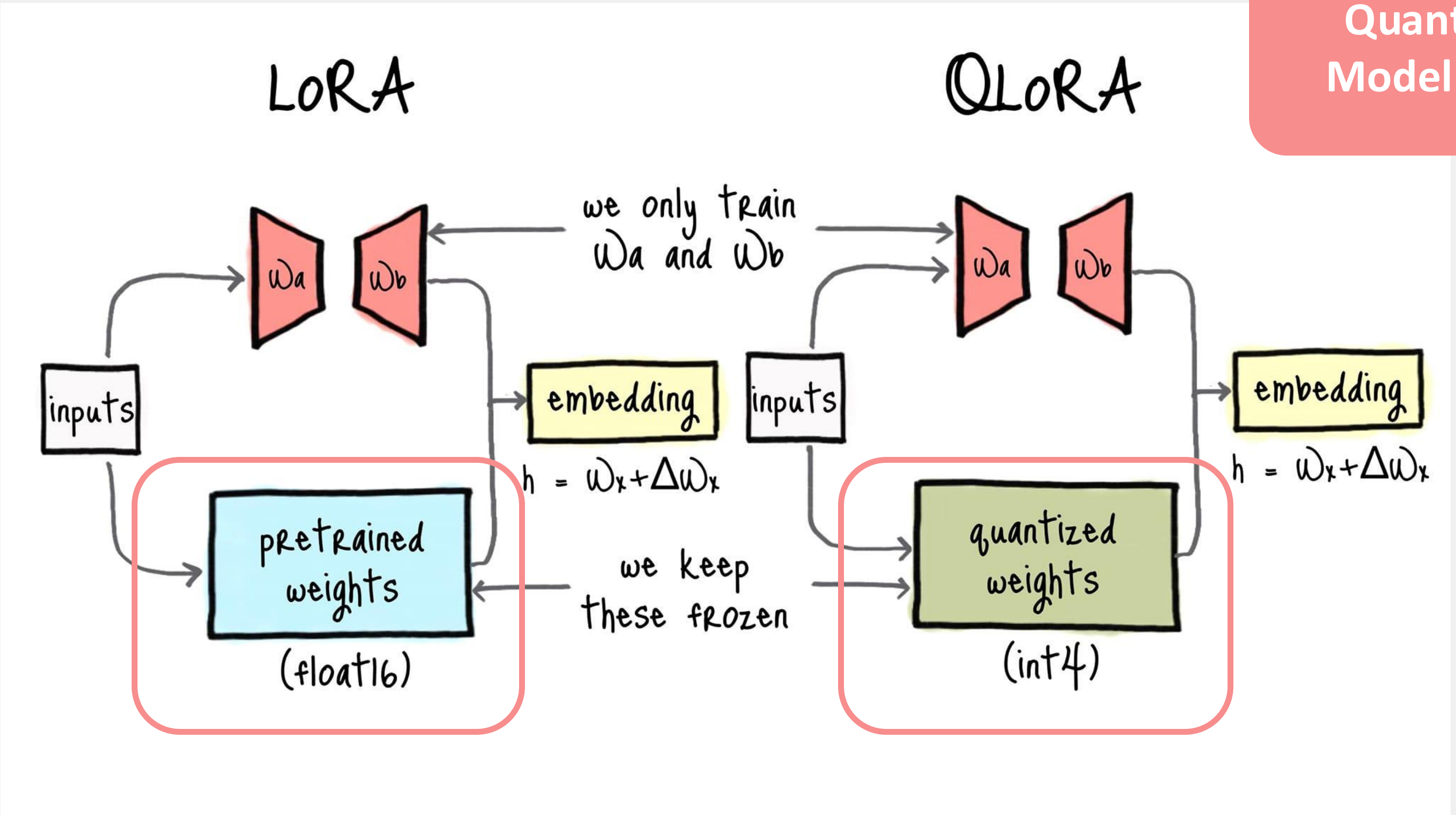
Parameter-Efficient Fine-Tuning (PEFT) - QLoRA



Methods to Add Knowledge to LLMs - Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT) - QLoRA

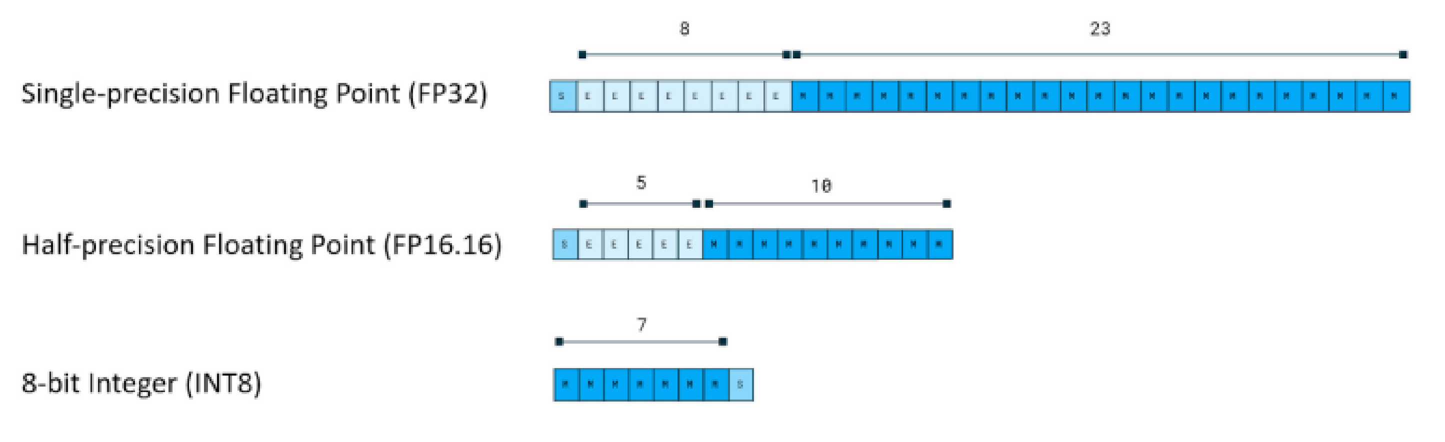
Quantization the Base Model to 4-bit precision



Methods to Add Knowledge to LLMs - Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT) - QLoRA - Quantization

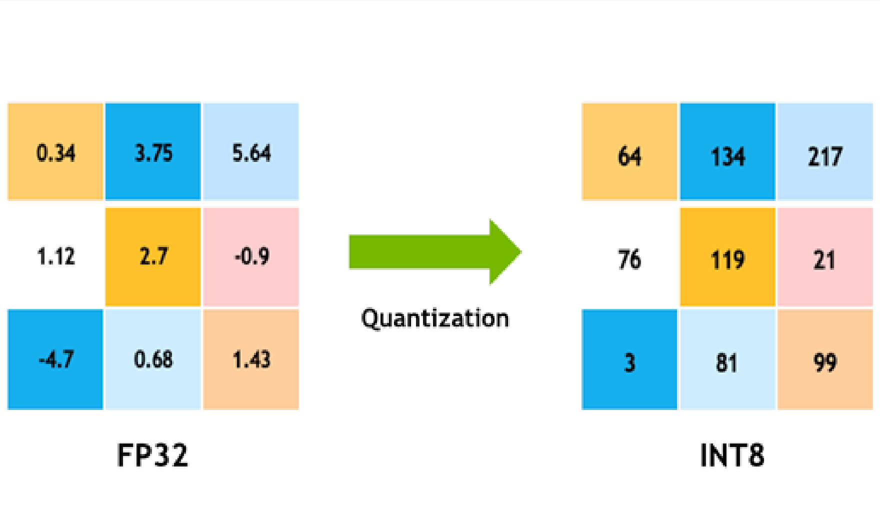
- **Reducing the precision** of numbers to make something smaller and faster
  - **Benefits:**
    - Memory Reduction
    - Faster Computation
    - Lower Power Consumption

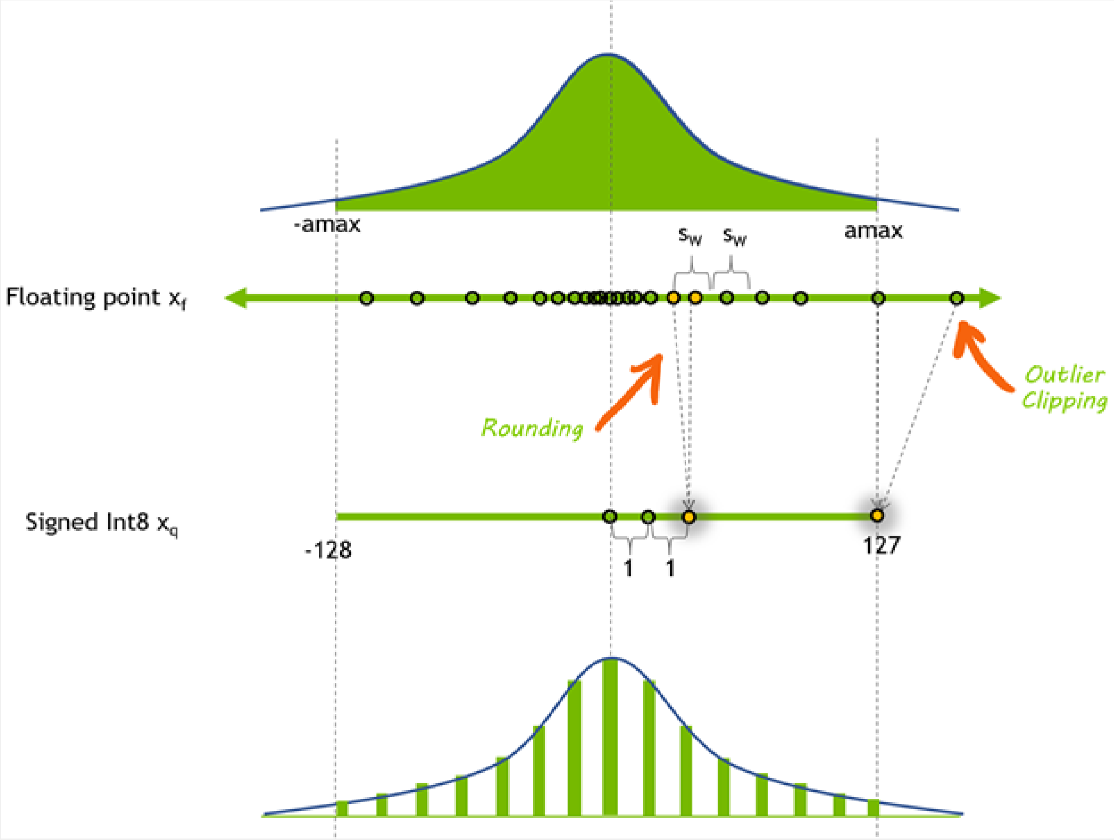


  - **Easiest Example:**
    - *High precision:* 0.73482937
    - *Lower precision:* 0.73

Methods to Add Knowledge to LLMs - Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT) - QLoRA - Quantization





Methods to Add Knowledge to LLMs - Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT) - QLoRA - Quantization

$$W_{FP32} = \begin{bmatrix} 0.12 & -0.87 & 0.45 \\ 0.33 & -0.21 & 0.78 \end{bmatrix}$$

- **Min-Max Values**

$$x_{min} = -0.87, \quad x_{max} = 0.78$$

- **Scale to INT8** Range [-128, 127]

$$\text{scale} = \frac{x_{max} - x_{min}}{255} = \frac{0.78 - (-0.87)}{255} = \frac{1.65}{255} \approx 0.00647$$

- **Zero Point:** -128

$$\text{zero\_point} = \text{round}(-128 - (-0.87)/0.00647) = \text{round}(-128 + 134.42) = \text{round}(6.42) \approx 6$$

- **Quantize Each Weight**

$$x_{int8} = \text{round}\left(\frac{x_{FP32}}{\text{scale}} + \text{zero\_point}\right)$$

$$W_{INT8} = \begin{bmatrix} 25 & -128 & 76 \\ 57 & -26 & 127 \end{bmatrix}$$

Methods to Add Knowledge to LLMs - Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT) - QLoRA - Quantization

$$W_{FP32} = \begin{bmatrix} 0.12 & -0.87 & 0.45 \\ 0.33 & -0.21 & 0.78 \end{bmatrix}$$

Memory (fp32) = 6 x 4bytes = 24 bytes

- **Min-Max Values**

$$x_{min} = -0.87, \quad x_{max} = 0.78$$

- **Scale to INT8** Range [-128, 127]

$$\text{scale} = \frac{x_{max} - x_{min}}{255} = \frac{0.78 - (-0.87)}{255} = \frac{1.65}{255} \approx 0.00647$$

- **Zero Point:** -128

$$\text{zero\_point} = \text{round}(-128 - (-0.87)/0.00647) = \text{round}(-128 + 134.42) = \text{round}(6.42) \approx 6$$

Memory (int8) = 6 bytes

- **Quantize Each Weight**

$$x_{int8} = \text{round}\left(\frac{x_{FP32}}{\text{scale}} + \text{zero\_point}\right)$$

$$W_{INT8} = \begin{bmatrix} 25 & -128 & 76 \\ 57 & -26 & 127 \end{bmatrix}$$

Methods to Add Knowledge to LLMs - Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT) - QLoRA - Quantization

$$W_{FP32} = \begin{bmatrix} 0.12 & -0.87 & 0.45 \\ 0.33 & -0.21 & 0.78 \end{bmatrix}$$

Memory (fp32) = 6 x 4bytes = 24 bytes

Extra Memory Used with Scale and Zero-Point!

Scale Factor (FP32) --> 4 bytes
Zero-Point (INT8) --> 1 bytes

- **Min-Max Values**

$$x_{min} = -0.87, \quad x_{max} = 0.78$$

- **Scale to INT8** Range [-128, 127]

$$\text{scale} = \frac{x_{max} - x_{min}}{255} = \frac{0.78 - (-0.87)}{255} = \frac{1.65}{255} \approx 0.00647$$

- **Zero Point:** -128

$$\text{zero\_point} = \text{round}(-128 - (-0.87)/0.00647) = \text{round}(-128 + 134.42) = \text{round}(6.42) \approx 6$$

Memory (int8) = 6 bytes

- **Quantize Each Weight**

$$x_{int8} = \text{round}\left(\frac{x_{FP32}}{\text{scale}} + \text{zero\_point}\right)$$

$$W_{INT8} = \begin{bmatrix} 25 & -128 & 76 \\ 57 & -26 & 127 \end{bmatrix}$$

Methods to Add Knowledge to LLMs - Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT) - QLoRA

Number of Trainable Parameters Table:

TABLE VI: Comparison of the average 5-shot MMLU test accuracy of LLaMA-7B and LLaMA-13B models fine-tuned with Alpaca. The higher the MMLU accuracy, the better. We also report total model parameters (# APs) and the ratio of trainable parameters.

Model	PEFT Method	# TPs	# APs	% Params	5-shot MMLU Accuracy
LLaMA-7B-Alpaca	FT	6738.4M	6738.4M	100	**41.79**
	$(IA)^3$	1.58M	6740.0M	0.02	37.88
	LoRA	159.9M	6898.3M	2.32	40.67
	QLoRA	79.9M	3660.3M	2.18	39.96
LLaMA-13B-Alpaca	FT	13015.9M	13015.9M	100	**49.60**
	$(IA)^3$	2.48M	13018.3M	0.02	47.42
	LoRA	250.3M	13266.2M	1.88	47.49
	QLoRA	125.2M	6922.3M	1.81	47.29

Source: Xu, L., Xie, H., Qin, S. J., Tao, X., & Wang, F. L. (2026). Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Methods to Add Knowledge to LLMs - Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT) - QLoRA

Number of Trainable Parameters Table:

TABLE VI: Comparison of the average 5-shot MMLU test accuracy of LLaMA-7B and LLaMA-13B models fine-tuned with Alpaca. The higher the MMLU accuracy, the better. We also report total model parameters (# APs) and the ratio of trainable parameters.

Model	PEFT Method	# TPs	# APs	% Params	5-shot MMLU Accuracy
LLaMA-7B-Alpaca	FT	6738.4M	6738.4M	100	**41.79**
	$(IA)^3$	1.58M	6740.0M	0.02	37.88
	LoRA	159.9M	6898.3M	2.32	40.67
	QLoRA	79.9M	3660.3M	2.18	39.96
LLaMA-13B-Alpaca	FT	13015.9M	13015.9M	100	**49.60**
	$(IA)^3$	2.48M	13018.3M	0.02	47.42
	LoRA	250.3M	13266.2M	1.88	47.49
	QLoRA	125.2M	6922.3M	1.81	47.29

Source: Xu, L., Xie, H., Qin, S. J., Tao, X., & Wang, F. L. (2026). Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Parameter-Efficient Fine-Tuning (PEFT) - QLoRA

Feature	LoRA	QLoRA
Base model precision	FP16/BF16	4-bit quantized
LoRA adapters	Yes	Yes
Trainable params	Small	Small
GPU Memory Usage	Reduced	Massively reduced
Peak GPU Memory Usage	~20 Gb (7B~13B model)	~8 Gb (7B~13B model)
Can fine-tune 65B model on 48GB GPU?	Usually no	Yes

Instruction Fine-Tuning

- Training process where a model is trained to **follow human instructions** correctly.
- Instead of just predicting the next word, the model learns:
  - **When a user gives an instruction, produce the appropiate response.**

Instruction	Input	Output
Suggest a good restaurant	Los Angeles, CA	In Los Angeles, CA, I suggest Rossoblu Italian Restaurant
Rewrite the sentence with mode descriptive words	The game is fun	The game is exhilarating and ejoyable
Calculate the area of the triangle	Base: 5cm; Height: 6cm	The area of the triangle is 15cm2

Methods to Add Knowledge to LLMs - Fine-Tuning

Instruction Fine-Tuning

- Training process where a model is trained to **follow human instructions** correctly.
- Instead of just predicting the next word, the model learns:
  - **When a user gives an instruction, produce the appropiate response.**

Instruction	Input	Output
Suggest a good restaurant	Los Angeles, CA	In Los Angeles, CA, I suggest Rossoblu Italian Restaurant
Rewrite the sentence with mode descriptive words	The game is fun	The game is exhilarating and ejoyable
Calculate the area of the triangle	Base: 5cm; Height: 6cm	The area of the triangle is 15cm2

Behaviour-Focused

Methods to Add Knowledge to LLMs - Fine-Tuning

Task-Specific Fine-Tuning

- Involves training the model on a **smaller, task-specific dataset.**
- For example:
  - Summarize this, translate that, Code Generation, Sentiment classification, etc.

❗ **Important:** Not necessarily instruction-style!

Task-Specific	Input	Output
Sentiment Classification	The customer service was extremely helpful and friendly.	Positive
Translate English to Italian	Where is the nearest hospital?	Dov'è l'ospedale più vicino?

Methods to Add Knowledge to LLMs - Fine-Tuning

Task-Specific Fine-Tuning

- Involves training the model on a **smaller, task-specific dataset.**
- For example:
  - Summarize this, translate that, Code Generation, Sentiment classification, etc.

❗ **Important:** Not necessarily instruction-style!

Task-Specific	Input	Output
Sentiment Classification	The customer service was extremely helpful and friendly.	Positive
Translate English to Italian	Where is the nearest hospital?	Dov'è l'osp

Objective-Focused

Methods to Add Knowledge to LLMs - Fine-Tuning

Domain-Adaptive Fine-Tuning

- The model learns about a **specialized domain deeply** with domain-specific data, so, it better understands specialized language, terminology, and style.
- Examples:
  - Legal contracts, Medical journals, Financial reports, etc

Domain-Adaptive	Input	Output
Classify patient's symptom description into a category	Patient shows elevated blood sugar levels and reports Patient has a headache and mild fever.	Flu
Predict market sentiment from financial news headlines	TechCorp reports record quarterly profits, beating analyst expectations.	Positive

Methods to Add Knowledge to LLMs - Fine-Tuning

Domain-Adaptive Fine-Tuning

- The model learns about a **specialized domain deeply** with domain-specific data, so, it better understands specialized language, terminology, and style.
- Examples:
  - Legal contracts, Medical journals, Financial reports, etc

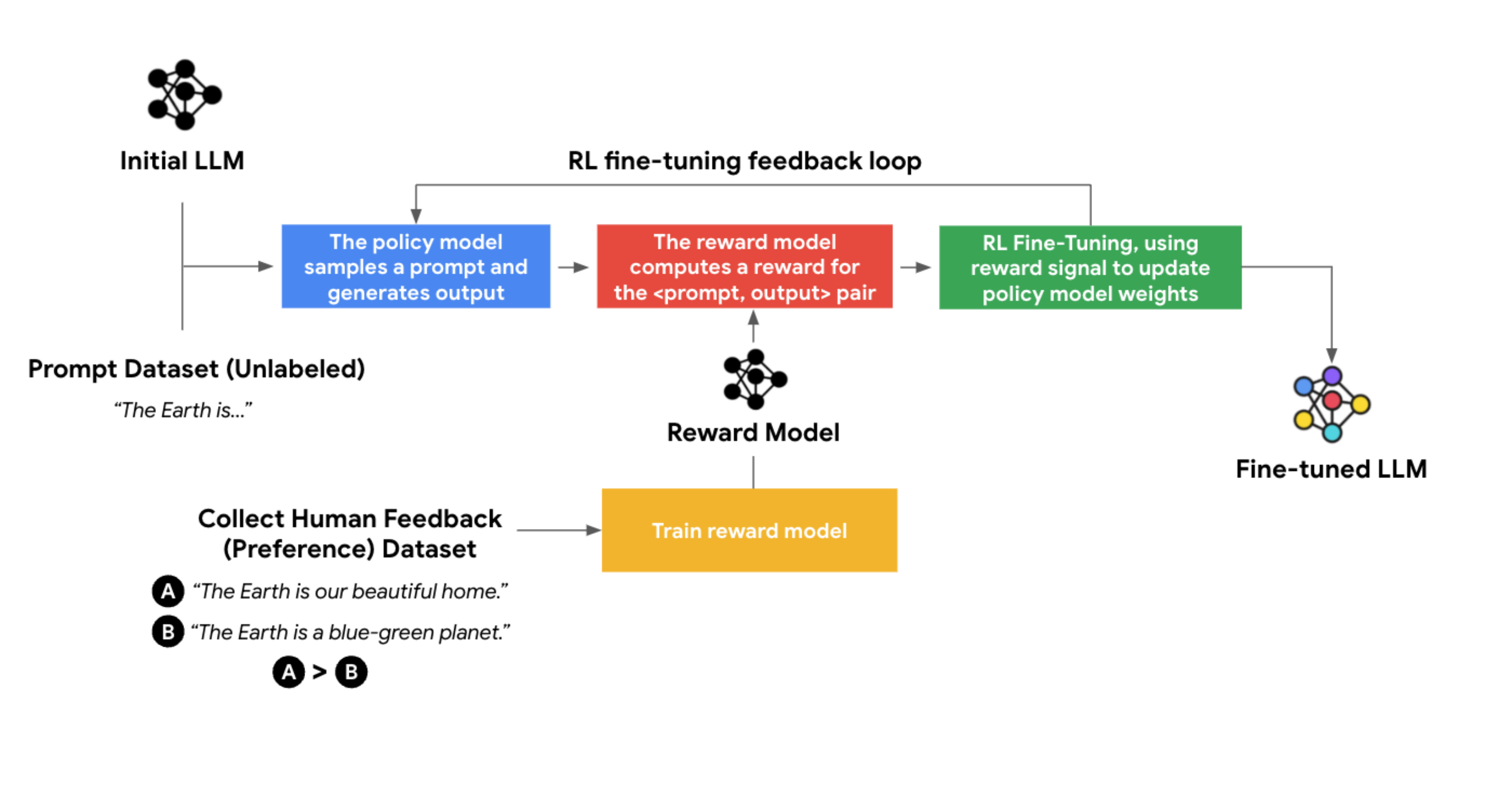
Domain-Adaptive	Input	Output
Classify patient's symptom description into a category	Patient shows elevated blood sugar levels and reports Patient has a headache and mild fever.	Flu
Predict market sentiment from financial news headlines	TechCorp reports record quarterly profits, beating analyst expectations.	Positive

Knowledge-Focused

Methods to Add Knowledge to LLMs - Fine-Tuning

Reinforcement-Learning from Human Feedback (RLHF)

- Technique used to align LLMs with Human Preferences. The model learns to produce **answers humans find helpful, safe, or aligned with a goal.**



What we need:

- *Base Language Model* --> generate text
- *Reward model* --> predict human preference score for any generated output

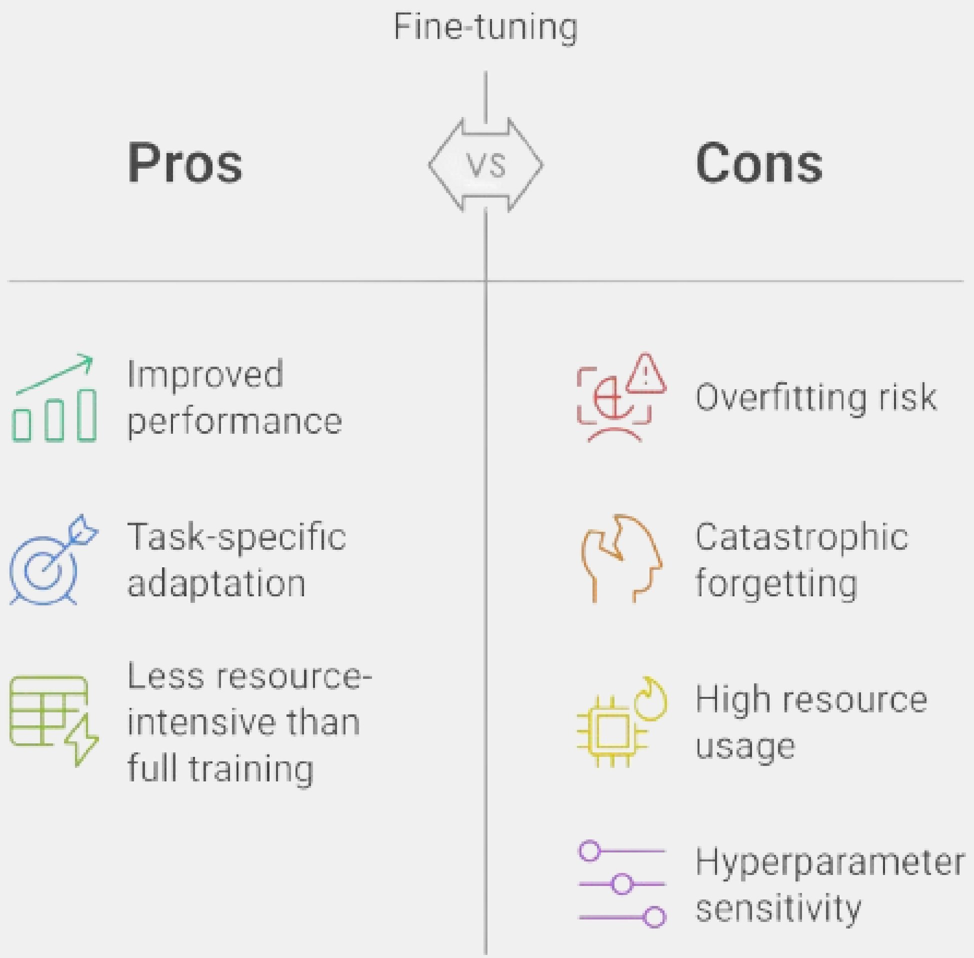
RL Steps:

1. Collect Human Feedback
   - Pairwise Comparisons
2. Reward Model:
   - Transform Model Outputs into a Score based on human preferences
3. Policy Update:
   - RL algorithm (PPO) updates the language model's parameters to increase high-reward outputs.
4. Repeat:
   - Generate new outputs with updated model
   - Score with Reward Model
   - Update again

Methods to Add Knowledge to LLMs - Fine-Tuning

Several Risks:

- Overfitting
- Catastrophic Forgetting
- Data Quality Issues
- Bias Amplification
- Training Instability



Source: https://www.mygreatlearning.com/blog/what-is-fine-tuning/

Agenda

- Introduction
- What can we build with LLMs?
- **Methods to Add Knowledge to LLMs**
    1. Prompt Engineering
    2. Fine-Tuning
    3. **Retrieval-Augmented Generation (RAG)**
    4. Agents
- System-Level Trade-Offs
- Demo on Fine-Tuning and Agents
- Hands-On Notebooks

Methods to Add Knowledge to LLMs - RAG

What is RAG?

- Technique where a **Language Model uses external documents/knowledge base** to help generate answers.
  - Instead of relying only on what the model **memorized during pretraining**, it **retrieves relevant information** at runtime.
  - Model reads the retrived documents and **generates answers grounded in facts.**

Why Do We Need RAG?

- Memory Limitation of LLMs (models forget facts or are outdated)
- Dynamic/Up-to-Date Knowledge (allows use current data)
- Accurate Responses (models can hallucinate or invent facts)
- Context Relevance (enrich context and improve coherence and relevance)
- Domain-Specific Knowledge
- Cost and Efficiency

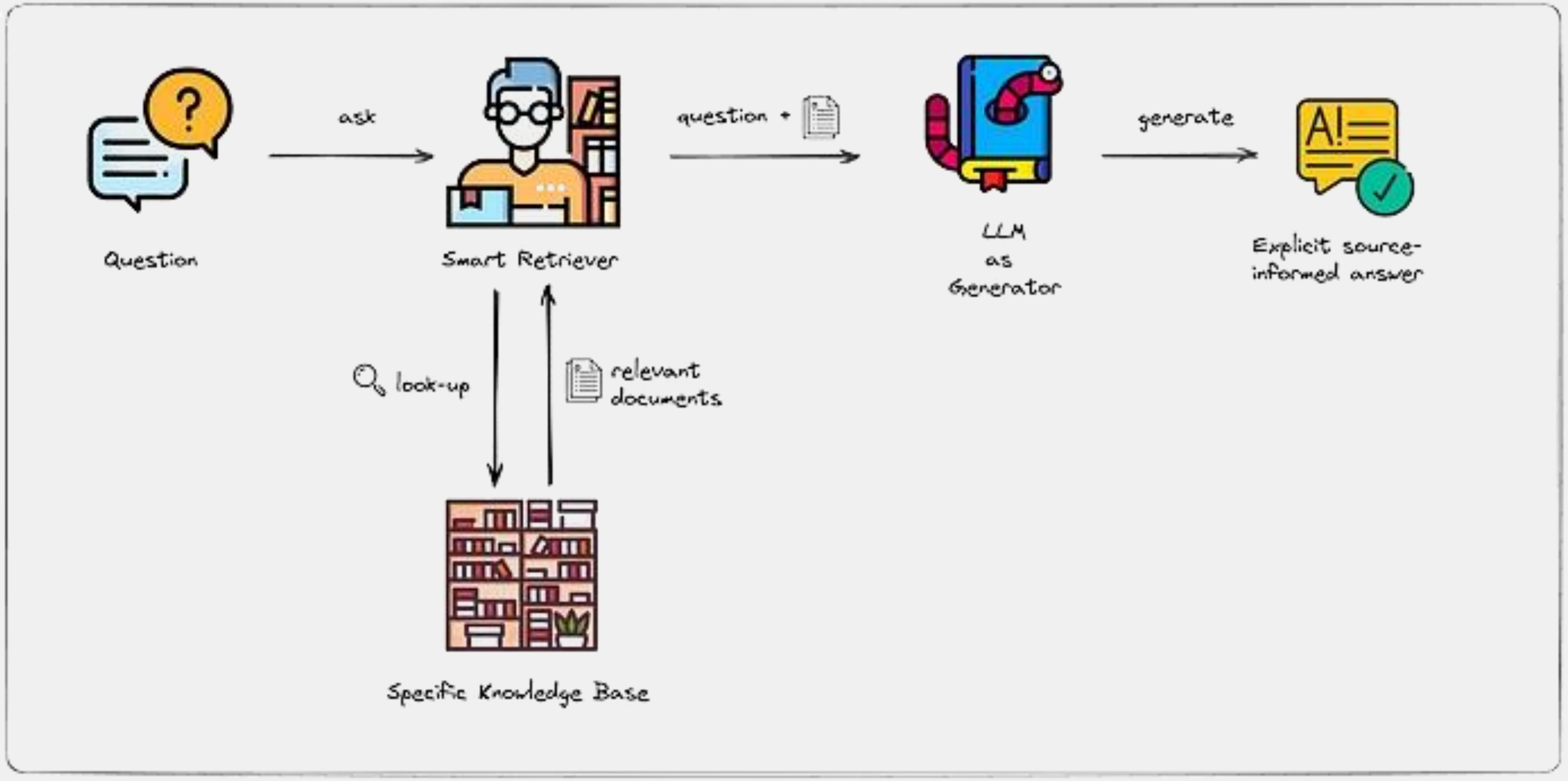
Methods to Add Knowledge to LLMs - RAG

Which to use?

Aspect	Fine-Tuning	RAG
Knowledge	Stored in model weights	Retrieved from external documents
Use case	**Stable**, repetitive tasks	**Dynamic**, up-to-date info
Cost	Expensive to update model	Update knowledge base easily
Accuracy	High on trained domain	Grounded in real documents, less hallucination
When to use	Structured data	Unstructured data
Practical Decision Rule	Problem about **Behaviour** and Task Performance? • The model outputs wrong format	Problem about **Knowledge?** • The model doesn't know my documents

Methods to Add Knowledge to LLMs - RAG

How RAG works?



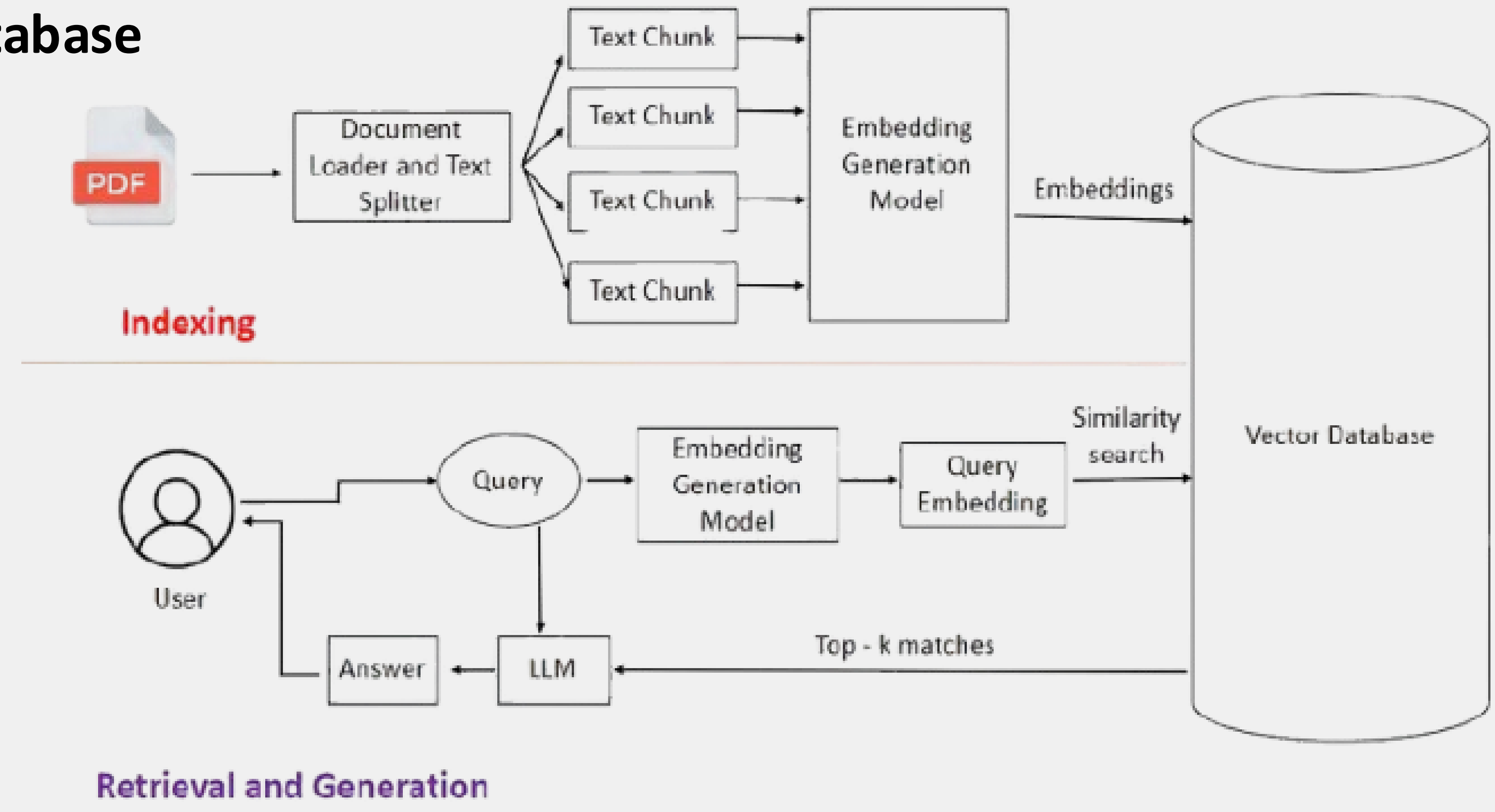
Source: https://www.ml6.eu/en/blog/leveraging-llms-on-your-domain-specific-knowledge-base

What we need:

1. LLM
2. Retrieval Model
3. Documents to construct the "Specific Knowledge Base"
4. Vector DB

Methods to Add Knowledge to LLMs - RAG

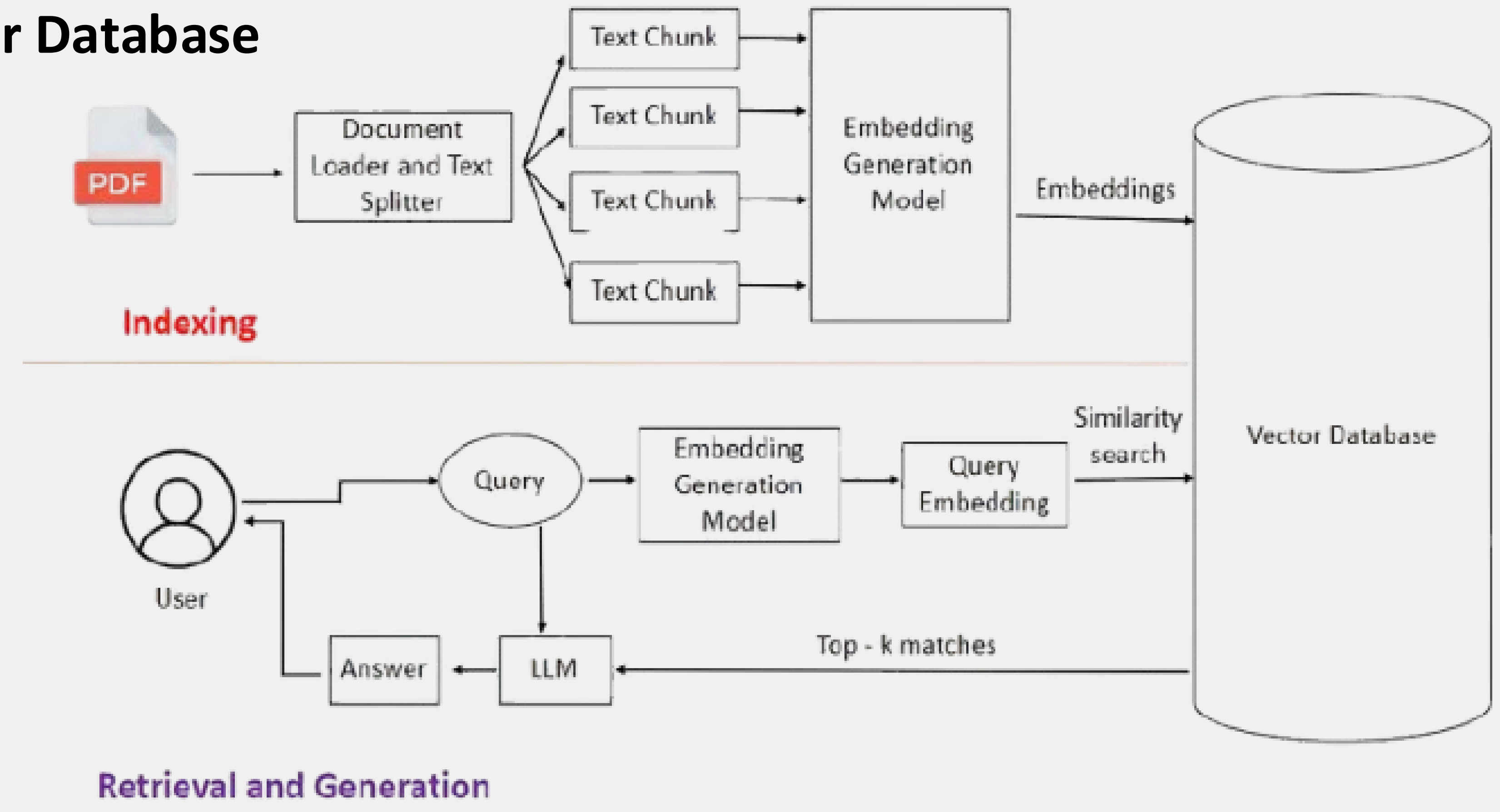
Vector Database



Source: https://medium.com/@pragyashukla2580/rag-retrieval-augmented-generation-for-your-own-documents-5e024267140e

Methods to Add Knowledge to LLMs - RAG

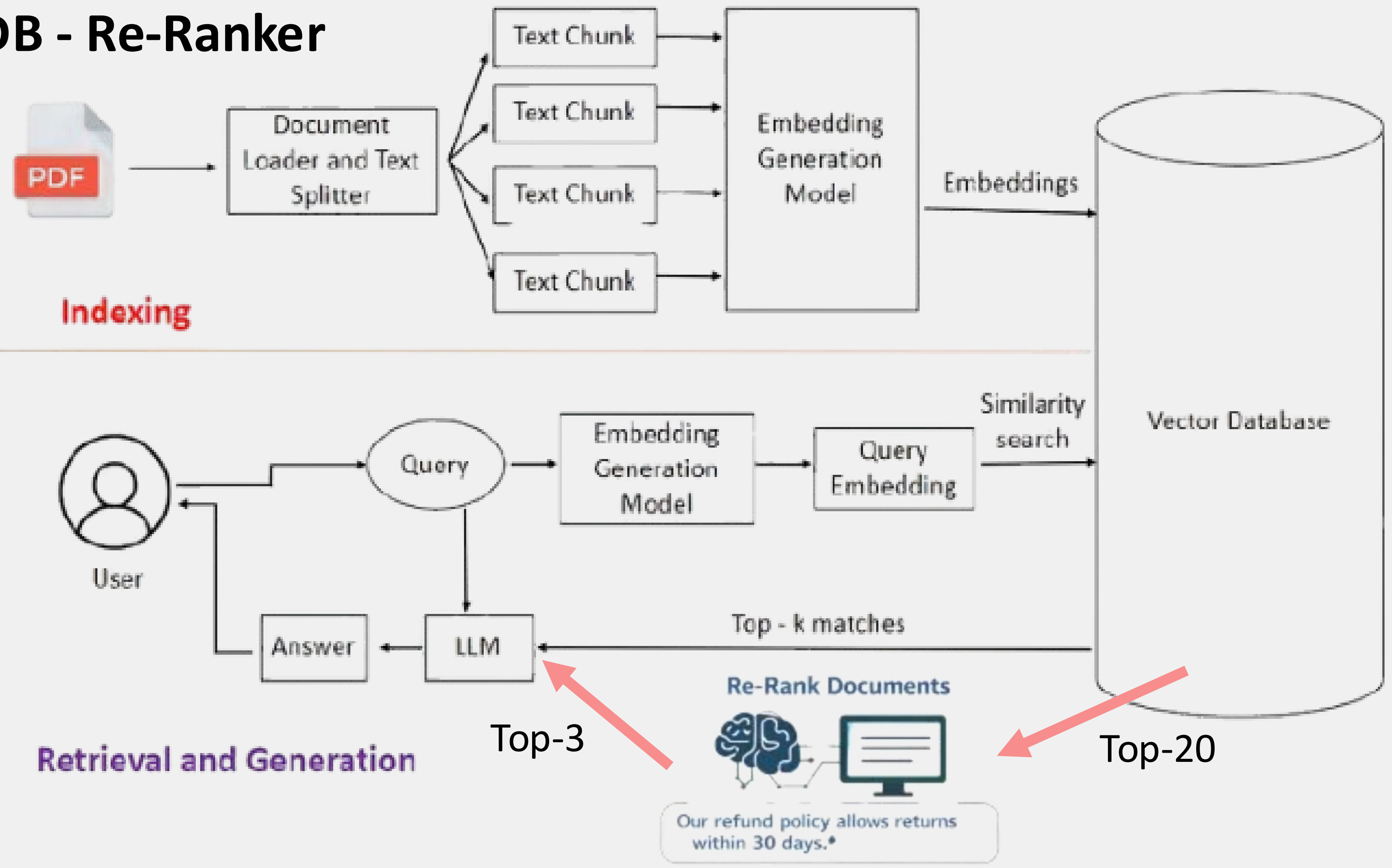
Vector Database



Source: https://medium.com/@pragyashukla2580/rag-retrieval-augmented-generation-for-your-own-documents-5e024267140e

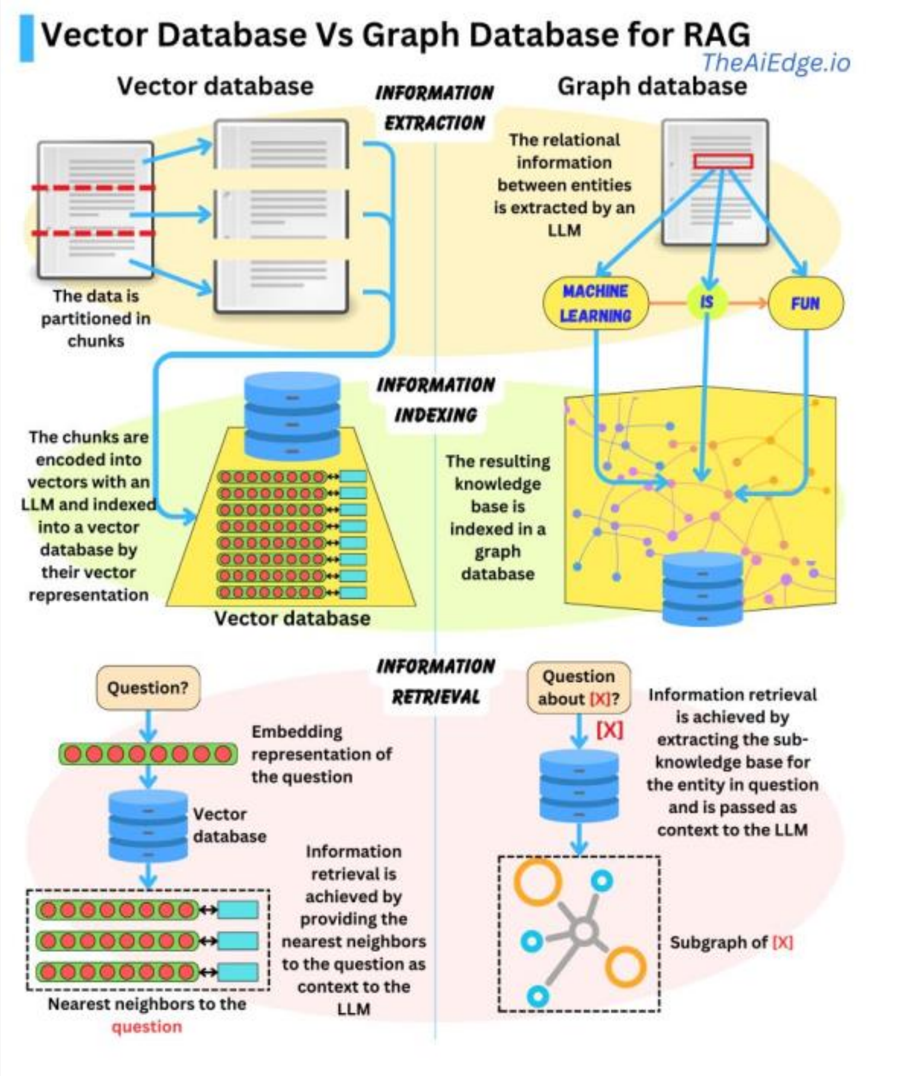
Methods to Add Knowledge to LLMs - RAG

Vector DB - Re-Ranker



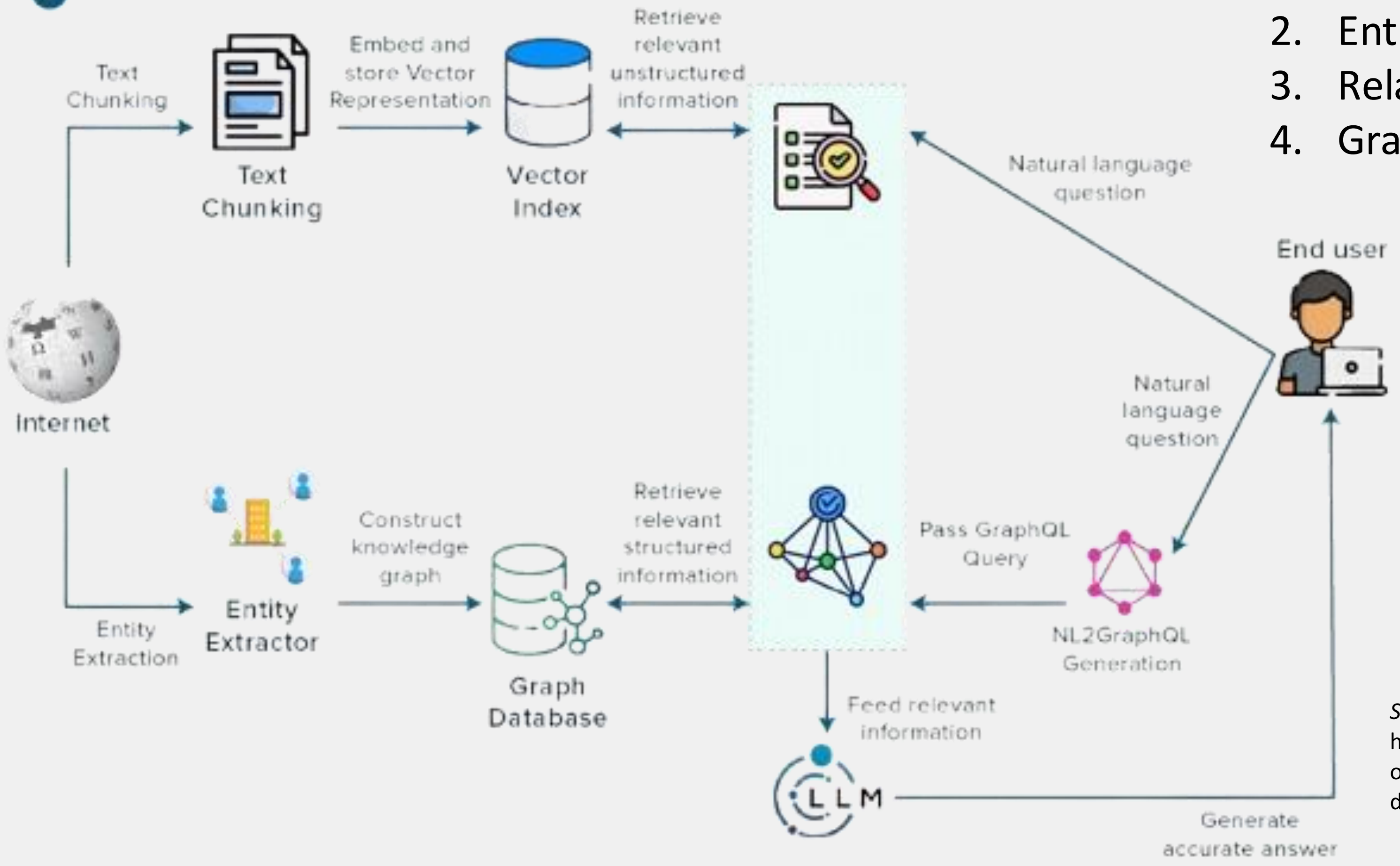
Methods to Add Knowledge to LLMs - RAG

Graph Database



Methods to Add Knowledge to LLMs - RAG

Graph Database



What we need:

1. LLM
2. Entities (Nodes)
3. Relationships (Edges)
4. Graph DB Engine

Source: https://www.tenupsoft.com/blog/boosting-ai-with-graph-and-vector-databases-in-rag-system.html

Agenda

- Introduction
- What can we build with LLMs?
- **Methods to Add Knowledge to LLMs**
    1. Prompt Engineering
    2. Fine-Tuning
    3. Retrieval-Augmented Generation (RAG)
    4. **Agents**
- System-Level Trade-Offs
- Demo on Fine-Tuning and Agents
- Hand-On Notebooks

Methods to Add Knowledge to LLMs - Agents

Agents

- AI system that can autonomously **perform tasks by making decisions, planning actions, and interating with tools or environments.**
  1. "Smart assistant" that can **decide what to do next** rather than just respoding passively.
  2. Powered by **LLMs**, combined with tools usage, reasoning, and memory.

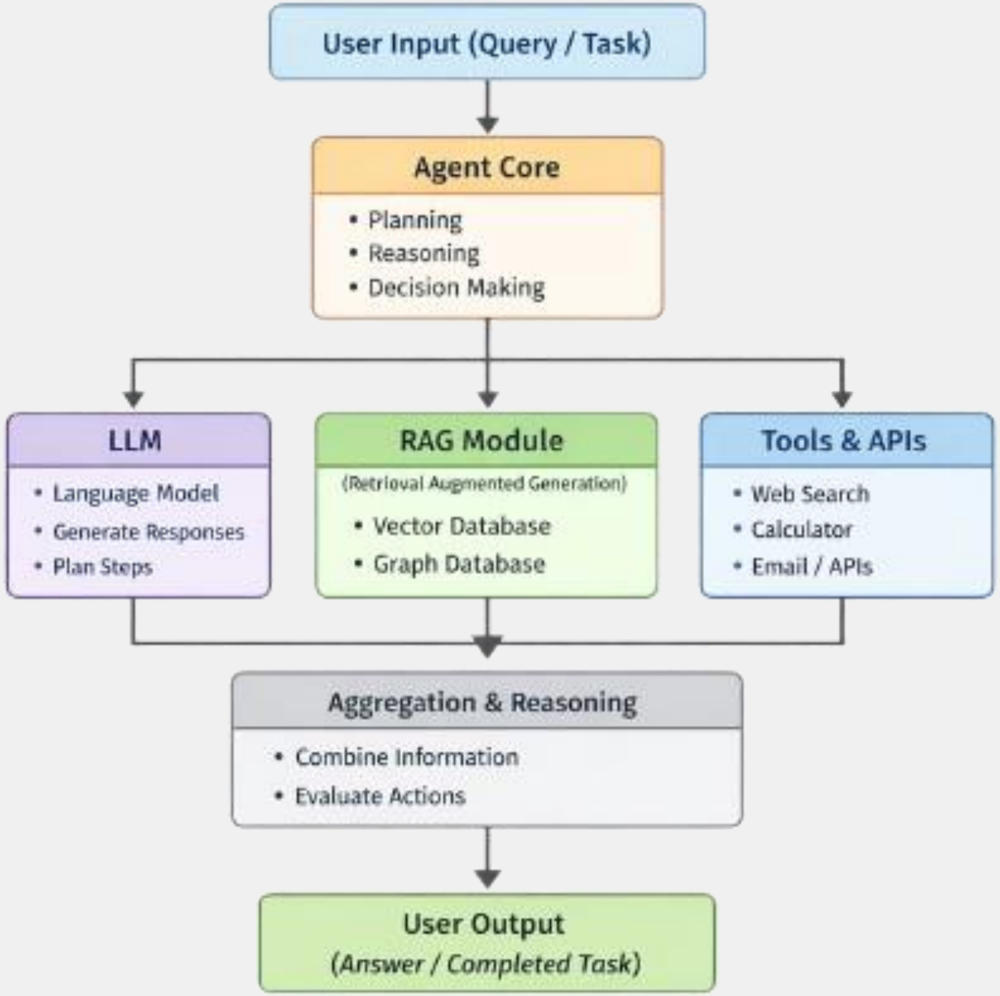
Why are Needed?

- Automate multi-step tasks
  1. Booking a flight requires checking flights, comparing prices, and completing purchase.
- Connect LLMs with tools and data
  1. Interact with APIs, databases, calculators or web searches.
- Handles dynamic problem-solving
- Improves reliability and safety

Methods to Add Knowledge to LLMs - Agents

How an Agent Works

- User Input
- Agent Core
    1. Decision-making engine
    2. Breaks tasks into sub-tasks and chooses which tools to call
- LLM
- Tools/API Access
    1. External tools for calculations, web search, file handling, etc
- Aggregation & Reasoning Layer
- Output



Methods to Add Knowledge to LLMs – Example Reasoning

```
==============================
QUESTION: I have sore throat for a couple of days and a slight fever


> Entering new AgentExecutor chain...
Thought: I should check if the symptoms are a red flag and if the patient needs immediate attention.
Action: Symptoms Checker
Action Input: I have sore throat for a couple of days and a slight fever
Observation: ```json
{
 "triage_level": "urgent",
 "summary": "Sore throat and a slight fever can be caused by various factors, including viral infections. It's important to monitor your symptoms and seek medical attention if they worsen or you experience other concerning symptoms.",
 "practical_next_step_advice": "You can try over-the-counter pain relievers like acetaminophen or ibuprofen to manage your fever and discomfort. Stay hydrated by drinking plenty of fluids. Consider using a throat lozenge or gargling with salt water to soothe your throat.",
 "red_flags": [
  "Difficulty breathing or shortness of breath",
  "High fever (over 103°F or 39.4°C)",
  "Severe pain or swelling in the throat",
  "Rash or other skin changes",
  "Confusion or disorientation",
  "Chest pain",
  "Persistent vomiting or diarrhea"
 ],
 "follow_up_questions": [
  "Have you experienced any other symptoms, such as cough, runny nose, or body aches?"
 ]
}
```
```
```

Methods to Add Knowledge to LLMs – Example Reasoning

```
Thought:Thought: I should consult a doctor for guidance on my sore throat and fever.
Action: Doctor RAG Assistant
Action Input: I have sore throat for a couple of days and a slight fever. I'm not sure what to do.
Observation: It's smart you're paying attention to your body and reaching out. Sore throats and mild fever can stem from a few different things, some viral, some requiring a bit more attention. Given that you haven't provided specific details about existing symptoms (like other bodily sensations or duration), here's a general approach:

**1. First things first: Self-care!**

 * **Rest:** Give your body a chance to recover. Keep yourself hydrated with plenty of fluids, like water or broth.
 * **Saltwater gargle:** This can ease the throat irritation. Mix 1/4 to 1/2 teaspoon salt in a cup of warm water and gargle for 30-60 seconds, then spit it out and repeat several times a day.
 * **Over-the-counter pain relief:** Take over-the-counter pain relievers like ibuprofen or acetaminophen if you need temporary relief from the fever and aches.
 * **Hydrate:** Drinking fluids is crucial for fighting off whatever's causing your symptoms. Tea, warm water, or broth can all be good options.

**2. When to Seek Professional Care:**

 * **Symptoms worsen:** If your sore throat becomes severe, you experience difficulty swallowing, or your fever persists.
 * **Contagious cough:** A lot of people with similar symptoms often cough, and that increased coughing means that you might spread it to someone at home especially your family.
 * **Persisting symptoms:** If the over-the-counter relief doesn't help, or if you're experiencing more significant pain or discomfort.
 * **High fever:** A fever of 103F (39.4C) or higher or if you are under 6 years old.

**3. Consult a Medical Professional:**

 This is always the best course if you want more accurate information. A healthcare provider can:

 * **Confirm the cause:** Determine if your sore throat is indeed viral, or a symptom of something more serious.
 * **Rule out other conditions:** There can be some more serious issues, and depending on the antibiotics used before your diagnostic blood test is done, there's a chance you have an antibiotic-resistant strain.
 * **Provide personalized advice:** A physician can suggest specific therapies like nasal sprays or antiviral medications if needed.


**Remember:** I'm an AI chatbot, and I can't diagnose medical conditions or prescribe medication. The information here is meant to guide your self-care and provide some initial steps to take.

Thought:Thought: I should consult a doctor for guidance on my sore throat and fever.
Final Answer: Based on your symptoms, it's best to consult a doctor for personalized advice. They can determine the cause of your sore throat and fever and recommend the best course of action.


> Finished chain.
RESPONSE: Based on your symptoms, it's best to consult a doctor for personalized advice. They can determine the cause of your sore throat and fever and recommend the best course of action.
```

Agenda

- Introduction
- What can we build with LLMs?
- Methods to Add Knowledge to LLMs
    1. Prompt Engineering
    2. Fine-Tuning
    3. Retrieval-Augmented Generation (RAG)
    4. Agents
- **System-Level Trade-Offs**
- Demo on Fine-Tuning and Agents
- Hand-On Notebooks

System-Level Trade-Offs

Dimension	Prompt Engineering	Fine-Tuning	RAG	Agents
What it modifies	Prompt only	Model weights	Adds retrieval layer	Adds planning + tools + memory
Changes model behavior?	⚠️ Slightly	✅ Yes (strongly)	❌ No (adds knowledge, not behavior)	✅ Yes (via orchestration)
Adds new knowledge?	❌ No	⚠️ Only if in training data	✅ Yes (external docs)	✅ Yes (via RAG/tools)
Handles changing data?	❌ Poorly	❌ Requires re-training	✅ Excellent	✅ Excellent
Infrastructure complexity	⭐ Very Low	⭐⭐ Medium	⭐⭐⭐ Medium-High	⭐⭐⭐⭐ High
Initial Cost	Very low	High (training cost)	Medium	High

System-Level Trade-Offs

Dimension	Prompt Engineering	Fine-Tuning	RAG	Agents
Maintenance Cost	Low	Medium-High	Low-Medium	Medium-High
Latency impact	None	None	+ Retrieval step	+ Planning + Tool calls
Control over output format	Limited	Strong	Limited	Strong
Multi-step reasoning	Weak	Same as base model	Same as base model	Strong
Tool usage	❌ No	❌ No	❌ No	✅ Yes
Best for	Quick improvements	Behavior/style/task optimization	Factual grounding	Automation & complex workflows

System-Level Trade-Offs – When to use

Prompt Engineering

- You want fast, cheap improvements
- No infrastructure changes
- Behavior adjustments are minor
- Prototyping stage

Best for: experimentation, internal tools, early-stage development

RAG (Retrieval-Augmented Generation)

- Knowledge changes frequently
- You need factual grounding
- Documents are large
- You want to avoid retraining

Best for: enterprise QA, internal documentation, legal/medical knowledge bases

Fine-Tuning

- You need consistent structured output
- The model must adopt a specific tone or policy
- You want task specialization (classification, extraction, domain tasks)
- Prompts alone are not enough

Best for: production systems requiring stable formatting and domain behavior

Agents

- The task requires multi-step reasoning
- The system must interact with APIs or databases
- Automation is required
- You need dynamic decision-making

Best for: workflow automation, research assistants, task execution systems

System-Level Trade-Offs – When to use

Prompt Engineering

- You w
- No in
- Behav
- Proto

Better Instructions

Best for: experi ... age development

RAG (Retrieval-Augmented Generation)

- Know
- You n
- Docu
- You w

Better Knowledge

Best for: enterp ... legal/medical knowledge bases

Fine-Tuning

- You need consistent structured output
- The m ... policy
- You w ... n, extraction, doma
- Promp

Better Behaviour

Best for: production systems requiring stable formatting and domain behavior

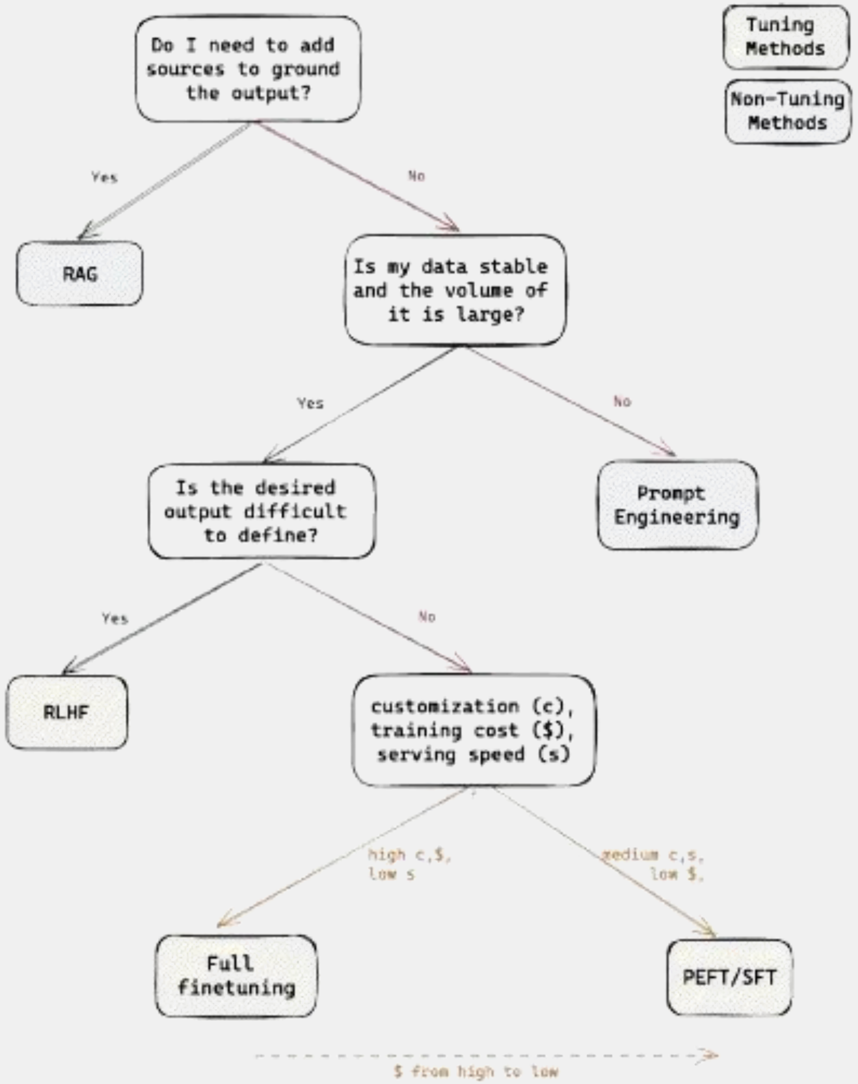
Agents

- The task requires multi-step reasoning
- The s ... tabases
- Autor
- You n

Better Autonomy

Best for: work ... s, task execution systems

System-Level Trade-Offs



Agenda

- Introduction
- What can we build with LLMs?
- Methods to Add Knowledge to LLMs
  1. Prompt Engineering
  2. Fine-Tuning
  3. Retrieval-Augmented Generation (RAG)
  4. Agents
- System-Level Trade-Offs
- **Demo on Fine-Tuning and Agents**
- Hand-On Notebooks

Demo on Fine-Tuning and Agents

Demo on RAG and Agents

Agenda

- Introduction
- What can we build with LLMs?
- Methods to Add Knowledge to LLMs
  1. Prompt Engineering
  2. Fine-Tuning
  3. Retrieval-Augmented Generation (RAG)
  4. Agents
- System-Level Trade-Offs
- Demo on Fine-Tuning and Agents
- **Hands-On Notebooks**

Hands-On Notebooks

1. Download notebooks and job scripts from **GitHub repository**
2. **Start working on notebooks:**
    1. Full Fine-Tuning
    2. LoRA Fine-Tuning
    3. QLoRA Fine-Tuning
    4. RAG
    5. Agents

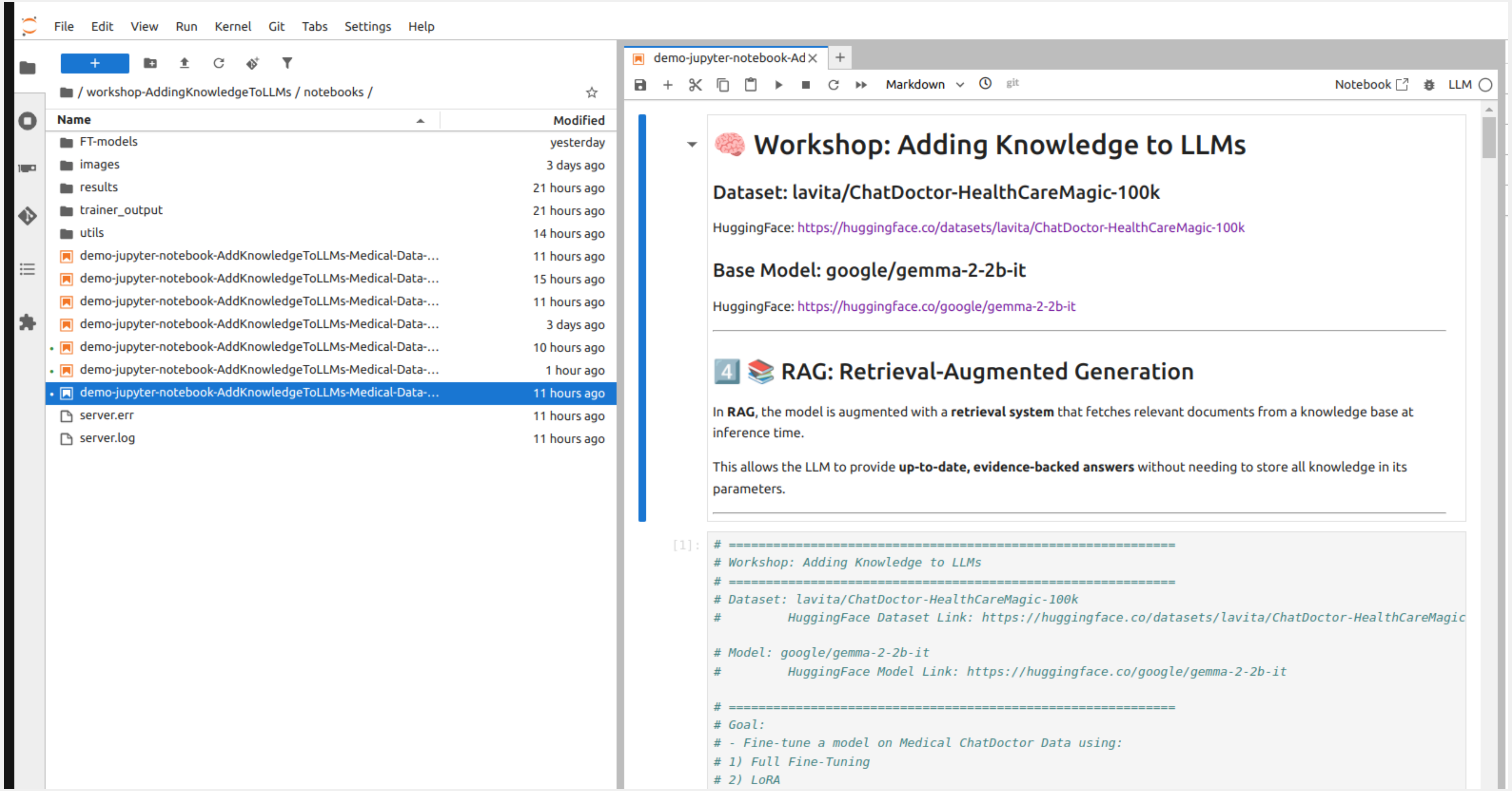
Hands-On Notebooks

1. **Connect to Leonardo Login Nodes**
2. **Go to workshop-AddingKnowledgeToLLMs**
3. **Go to your $HOME**
   - cd $HOME
4. **Copy data (ONLY notebooks and jobscripts) from common repo to your HOME:**
   - cp -r /leonardo_work/tra26_minwinsc/workshop-AddingKnowledgeToLLMs/jobscripts/ $HOME/workshop-AddingKnowledgeToLLMs/
   - cp -r /leonardo_work/tra26_minwinsc/workshop-AddingKnowledgeToLLMs/notebooks/ $HOME/workshop-AddingKnowledgeToLLMs/
5. **Run chappyner** script in your local.
   - (if chappyner is not working) --> Run SLURM scripts '.sh' inside 'jobscripts' folder
     - Add '#SBATCH –reservation=s_tra_minwinsc' for accessing the reservation

Hands-On Notebooks

```
gcortiad@BSC-8488186158:~/Descargas/presentations/AI Winter School - CINECA$ bash chappyner_jupy-original.sh
Enter your username on Cineca clusters: gcortiad

version 0.3.1

├─ Running startup script to ease access on the chosen cluster
├─ Testing connection to cluster
└─┬─ Submitting job for jupyter-train
  ├── Job ID: 34929149
  ├── Waiting for job 34929149 to be running (this might take a while)
  └── Job running!

              Welcome to Leonardo!

 You can find the User Guide here:
    https://docs.hpc.cineca.it/hpc/leonardo.html
 Please stay updated via this website for any news:
    https://www.hpc.cineca.it/hpc-center-news/

──────────────────── Job details ────────────────────
 Your job 34929149 is running on Leonardo using a single node with 4
 tasks. Job is using 4 gpus.
 You are using tra26_minwinsc account on boost_usr_prod partition and
 boost_qos_dbg qos.
 If you need other details please run 'scontrol show job 34929149' on
 the cluster for a full overview of the job.

──────────────────── Tool details ────────────────────
 You can connect to Jupyter Lab using the following url:

 http://127.0.0.1:55221/lab?token=wtt37Hzg7qQyR2XLtvphHJHi1TdVCdra

 until the job expires (at 10:16 CET).To quit the session click on
 "File -> Shut down".
──────────── When you want to leave, press Enter ────────────
```

Hands-On Notebooks



Hands-On Notebooks

1. **LoRA notebook:**
    1. Include more/less Data to Train
    2. Change hyperparameters on Fine-Tuning:
        - Learning Rate, Batch Size, Number of Epochs
    3. Adapter Size Variation
    4. Test it with new prompts
2. **RAG notebook:**
    1. Change top-k retrieved contexts
    2. Understand the impact of retrieval quality
    3. Retrieval with Noisy Queries: Introduce typos or ambiguous phrases to check if the model still retrieves relevant context
3. **Agents notebook:**
    1. Create a new prompt that requires at least 2 reasoning steps or tool calls
    2. Add a new tool and force it to use it

Reach out to us @ *info@minerva4ai.eu*

Thank you

Co-funded by the European Union

EuroHPC Joint Undertaking

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101182737. The JU receives support from the Digital Europe Programme.

AI4S
BSC AI 4 Science Fellowships

