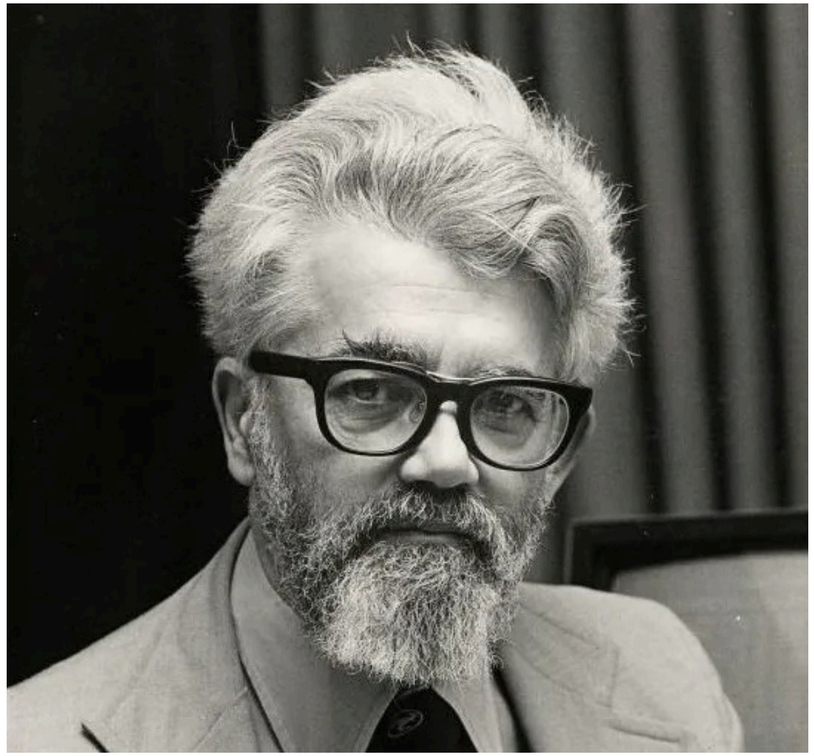
Introduction to Large Language Models (LLMs) Training and Adaptations

- Priyam Ghosh

(Upcoming AI Research Intern @Nvidia)
(Ex JPM @NASA)

HISTORY OF ARTIFICAL INTELLIGENCE

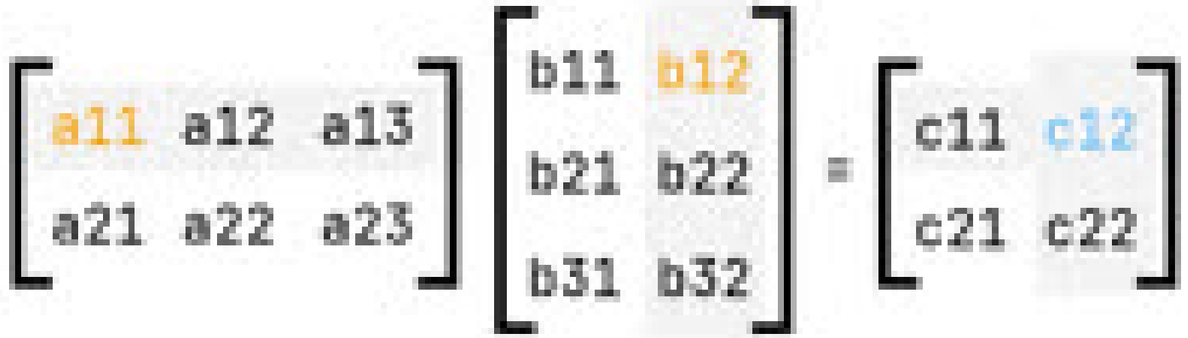
Alan Turing



John McCarthy

How LLMs Learn??

Behind the magic, LLMs are just glorified matrix multiplication machines.



$$c11 = a11 * b11 + a12 * b21 + a13 * b31$$

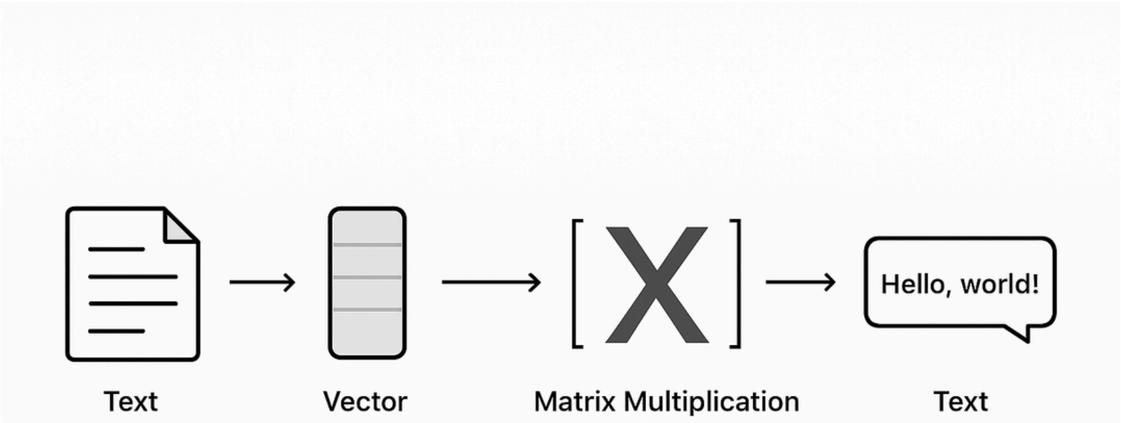
$$c12 = a11 * b12 + a12 * b22 + a13 * b32$$

$$c21 = a21 * b11 + a22 * b21 + a23 * b31$$

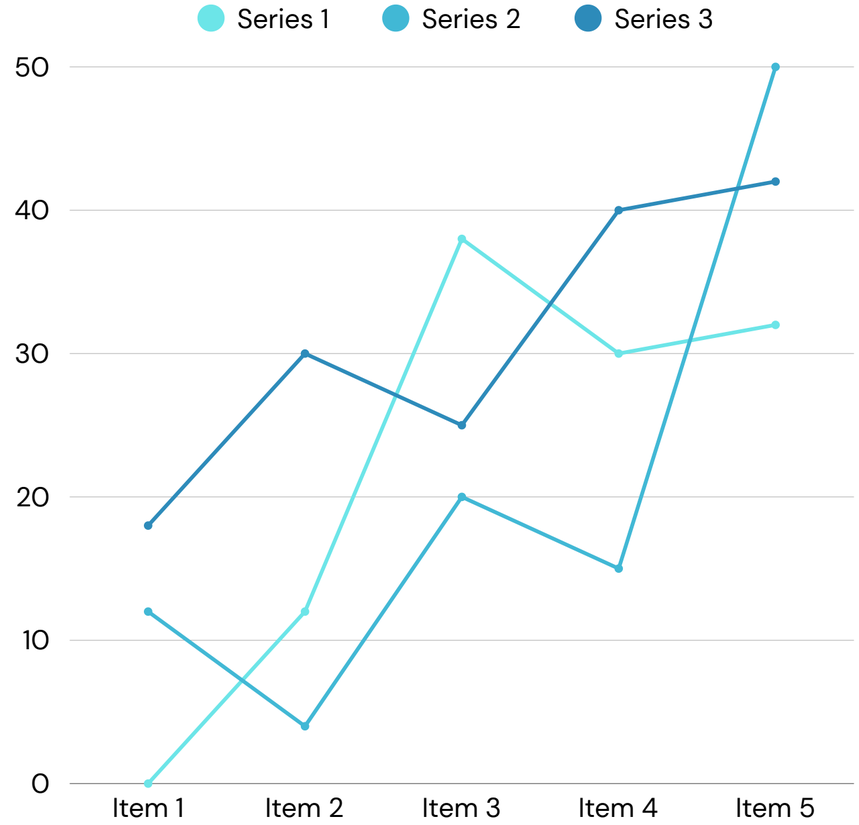
$$c22 = a21 * b12 + a22 * b22 + a23 * b32$$

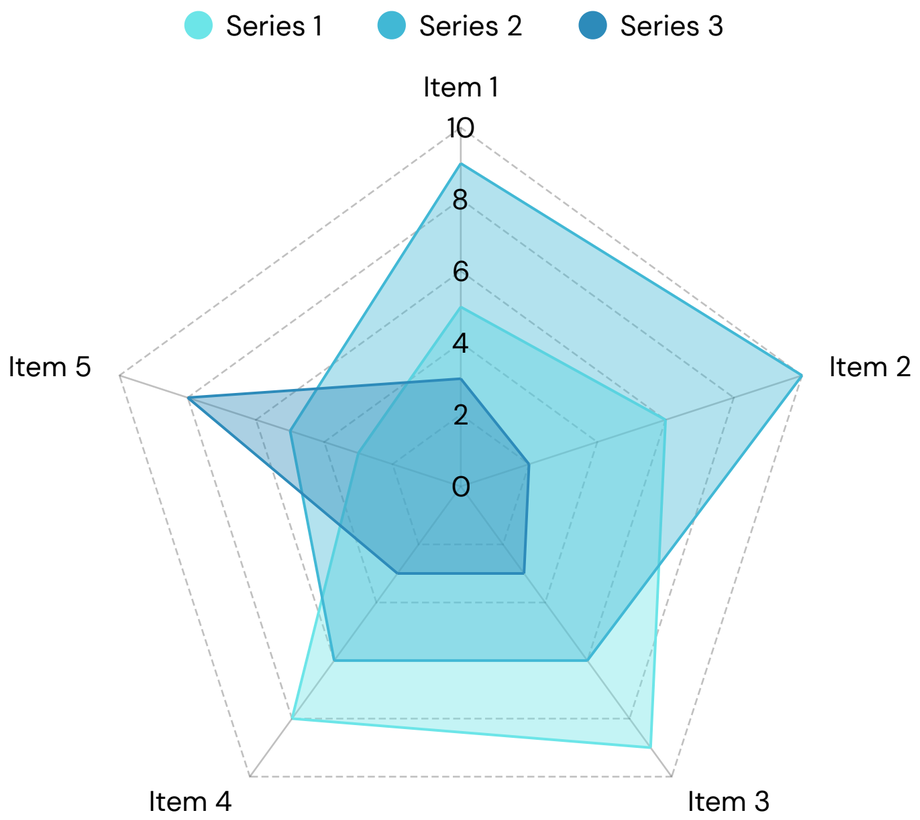
"Input Embedding" ➔ "Weight Matrices" ➔ "Output Embedding"

- Every word, sentence, and meaning is converted into **numbers** (**vectors**).
- These numbers are **multiplied** by massive **matrices** (learned during training).
- The output is new numbers, which are **decoded** back into **words**.
- At their core, LLMs are huge chains of **matrix multiplications — billions** of times!



Data: The Best Friend of Large Language Models (LLMs)





Training Process: Predicting the Next Word

1

Core Objective

LLMs learn by predicting the next word in a sequence, such as guessing "jumps" after "The quick brown fox."

2

Input and Output

The model receives a sequence of words as input and outputs probabilities over a large vocabulary representing possible next words.

3

Learning Through Prediction

By continually refining predictions against actual next words during training, the model develops deep contextual understanding of language.

The "Tireless Musician" Analogy

Ujjwal is ____?

1. sleeping
2. dead
3. लौंडियाबाज़

1. **Data Feasting :** student reading the book and understanding stuffs
2. **Self-supervized Predictions** trying to solve a puzzle game with predictions and internal error solving
3. **Model Architecture & Forward Pass** like an assembly line going from one layer to other trying to create the best possible output
4. **Learning Loop** redoing each step and in ever step it learns more and more and repeats it again.
5. **Scale and Repeat**

The Training Loop: Forward Pass, Loss, Backpropagation

Forward Pass

The input data is processed through the model layers to generate predicted output probabilities.

Loss Calculation

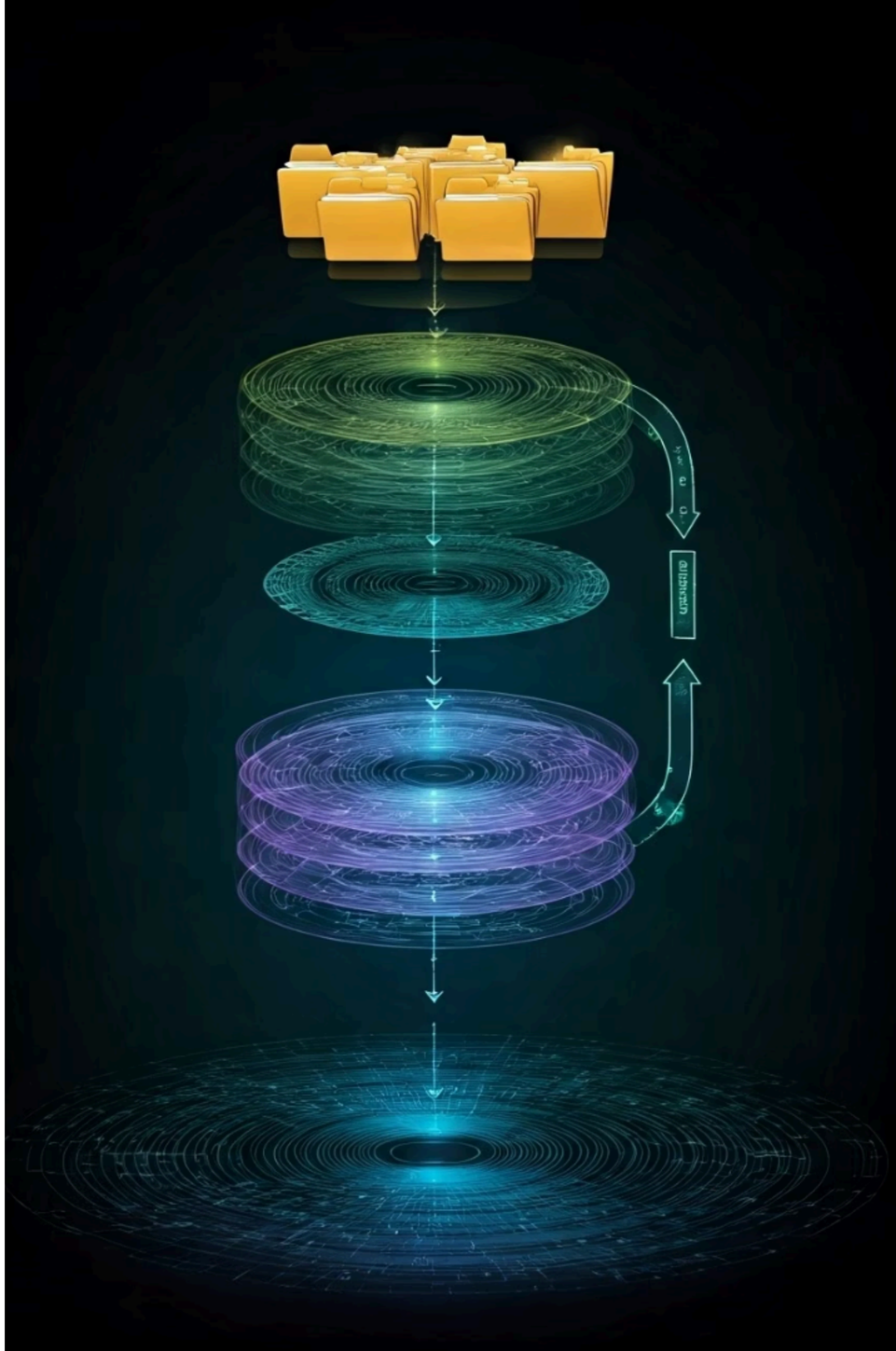
The model’s predictions are compared to the true next word using a loss function to quantify errors.

Backpropagation

Gradients of the loss are computed and propagated backwards to update the model’s parameters.

Optimization

Algorithms like Adam or SGD adjust weights to minimize error, iteratively improving model accuracy.

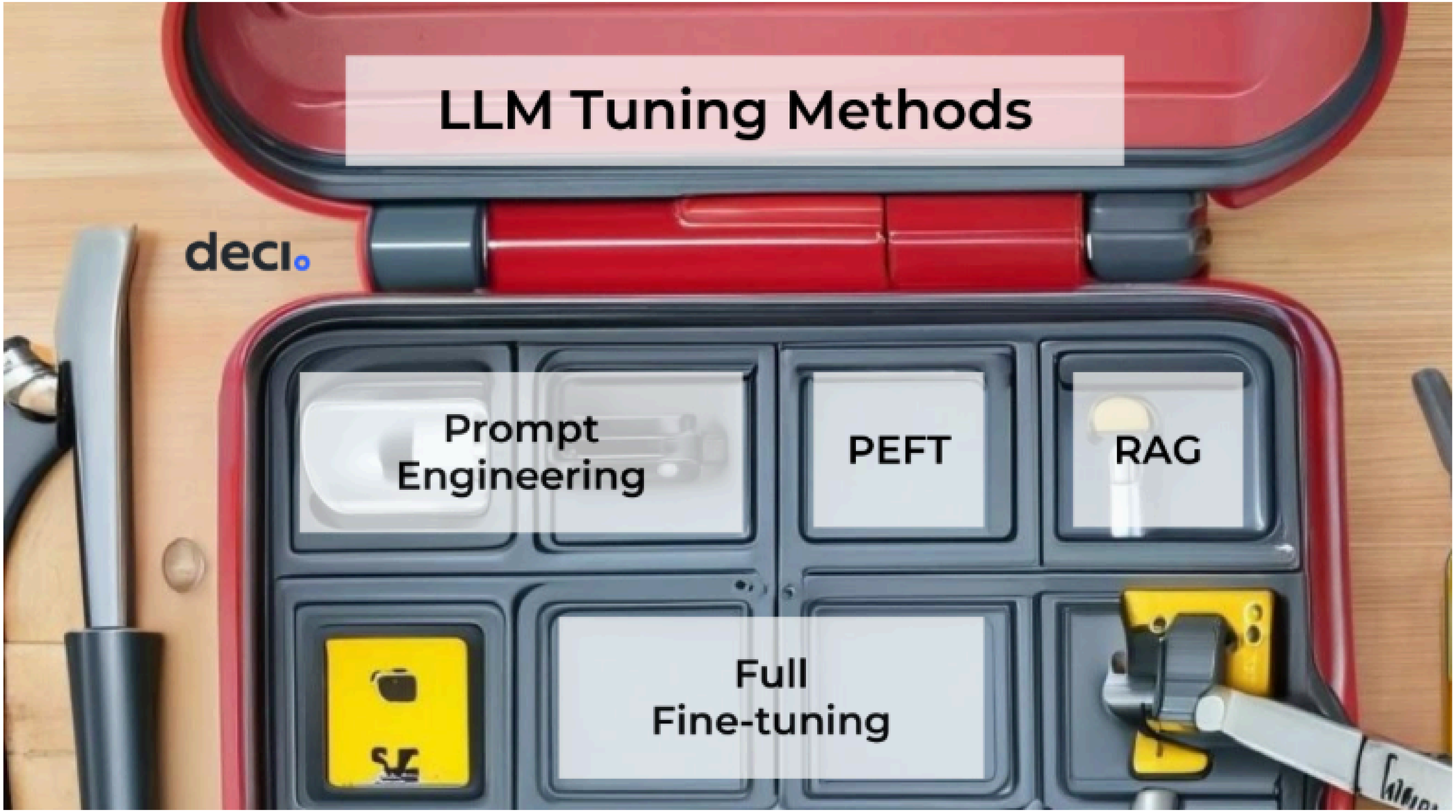


Fine-Tuning LLMs: What It Is and Why It’s Needed

Pre trained Model

Domain Specific Fine tuning

- Updated Accuracy
- Customization
- Data Privacy
- Handling Rare Scenarios


LLM Tuning Methods
deci
Prompt Engineering
PEFT
RAG
Full Fine-tuning

Fine-tuning and Adaptation

Full Fine-Tuning

Complete retraining of all model parameters on specific task data to tailor performance.

Prompting / Few-Shot Learning

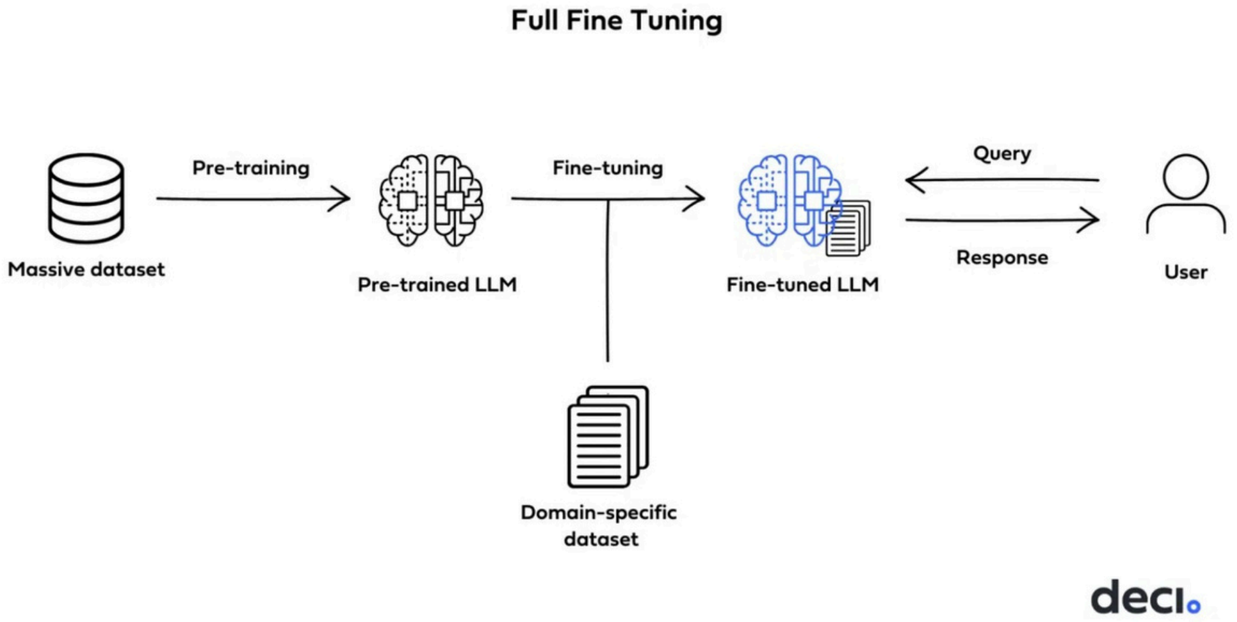
Using clever input prompts with few examples to steer model outputs without retraining.

Parameter-Efficient Fine-Tuning (LoRA)

Updating only a small subset of parameters to adapt models efficiently with lower resource requirements.

Full Fine tuning

- **Updates all model parameters for deep adaptation**
- **Needs large datasets and high computational resources**
- **Delivers maximum accuracy for specialized tasks**
- **Risk of overfitting and losing general knowledge**
- **Best for mission-critical, highly customized applications**



Model parameters are the internal variables of a machine learning model that are learned from data during training.

Prompt Tuning

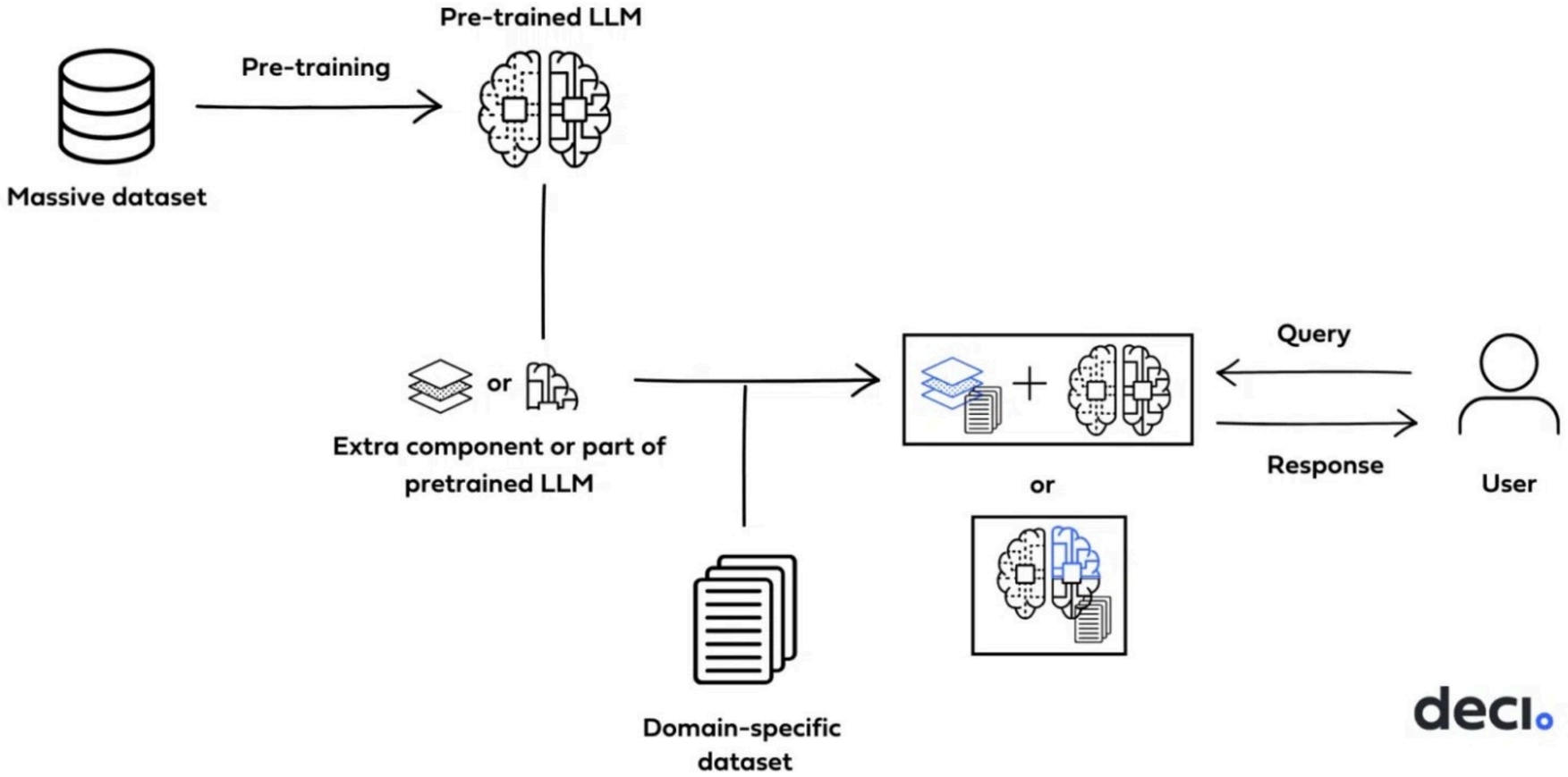
- **Trains small, learnable prompt embeddings**
- **Model weights stay frozen**
- **Persistent effect (task-based)**
- **Efficient, reusable task adaptation**

Soft Prompts: These are artificial, learnable tokens (not human-readable) added to the beginning of the input sequence. They are optimized during training to guide the model toward better performance on a specific task

Analogy: Intelligence Machine with Passcode

Parameter Efficient Fine Tuning (PEFT)
Low Rank Adaptation (LoRA)

Parameter-Efficient Fine-Tuning

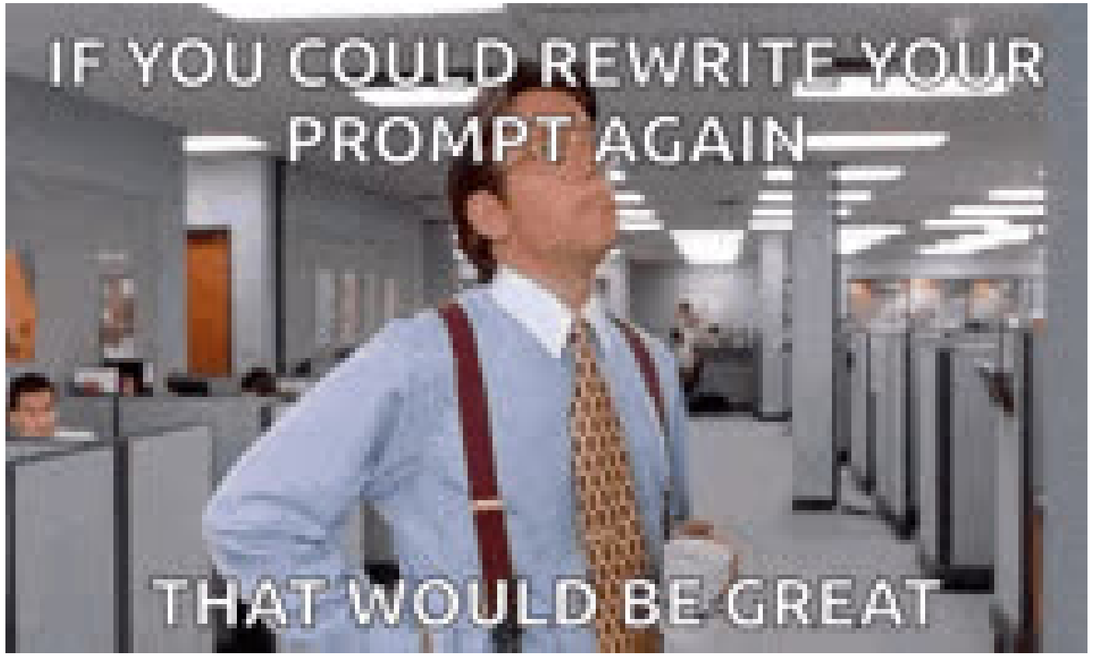


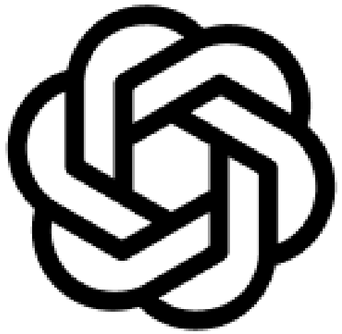
deci

- Adds small trainable matrices to the model (adapters).
- Keeps original model weights frozen.
- Dramatically reduces memory and compute needs.
- Achieves performance close to full fine-tuning.
- Allows quick, cheap adaptation to new tasks.
- Easy to swap between different task adapters.
- No extra inference cost after merging adapters.
- Great for customizing large models efficiently.

Aspect	Full Fine-Tuning	Prompt/Few-Shot Learning	LoRA Fine-Tuning
What is it?	Retrain all model parameters on new data	Guide the model with examples in the prompt	Update a small subset of parameters using adapter layers
Training Needed?	Yes	No	Yes
Resource Usage	Very high (needs lots of compute & time)	None (just prompt crafting)	Low (efficient, less compute needed)
Accuracy Potential	Highest (best for deep customization)	Limited by model's existing knowledge	High (close to full fine-tuning)
Risk of Forgetting	Yes (may lose general knowledge)	None	Low
Speed to Implement	Weeks	Minutes	Days
Best For	Critical, highly specialized tasks	Quick tests, simple or generic tasks	Most domain-specific applications
Example Use Case	Medical diagnostics, legal analysis	Email formatting, sentiment analysis	Customer support bots, industry chatbots

Method	Analogy	Resource Cost	Flexibility	Best For
Full Tuning	Relearning everything	Very High	Low (task-specific)	Critical, high-accuracy tasks
LoRA	Targeted cheat sheets	Low	High (swap adapters)	Efficient multi-task learning
Prompt Tuning	Secret exam hints	Minimal	Moderate (prompt design)	Quick, low-resource tasks





GPT and Modern LLMs

- Large language model that generates **human-like text**
- **Pre-trained** on massive datasets (books, websites, articles)
- **Transformer architecture** uses self-attention for deep context understanding
- Works by predicting the next word in a **sequence**, one at a time
- Can be fine-tuned for specific tasks (e.g., medical, legal, coding)
- Versatile: Used for chatbots, content creation, translation, summarization, and more
- Produces natural, relevant, and coherent text
- Scalable: **Handles huge amounts of data quickly**

Limitations and Challenges of LLMs

Bias and Hallucination

LLMs inherit societal biases from data and sometimes generate false or misleading information, which is a critical challenge.

Context Limitations

LLMs struggle with maintaining coherence over very long text due to fixed context window sizes.

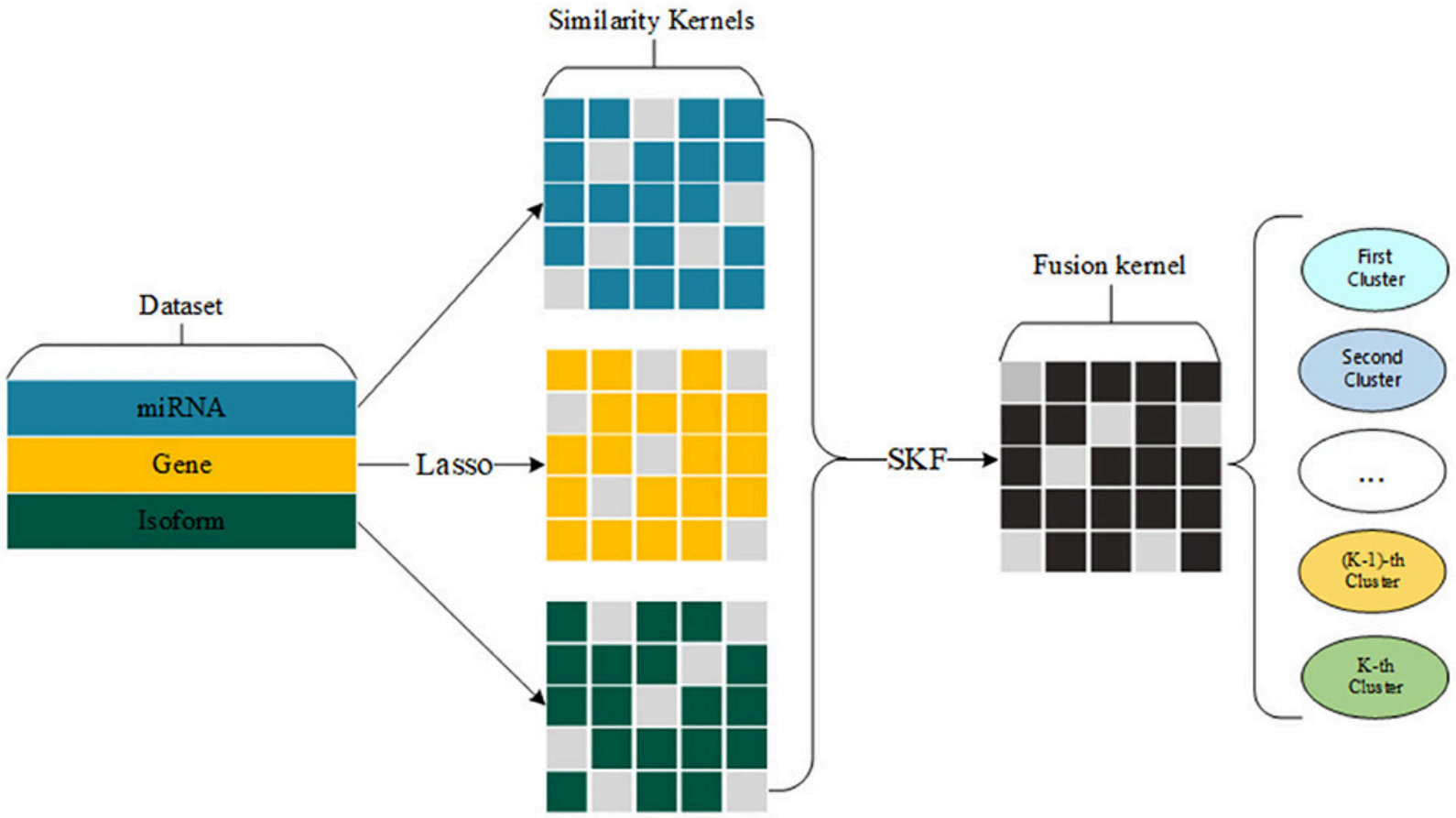
High Resource Costs

Training and running LLMs demand massive computational power and memory, posing environmental and accessibility concerns.

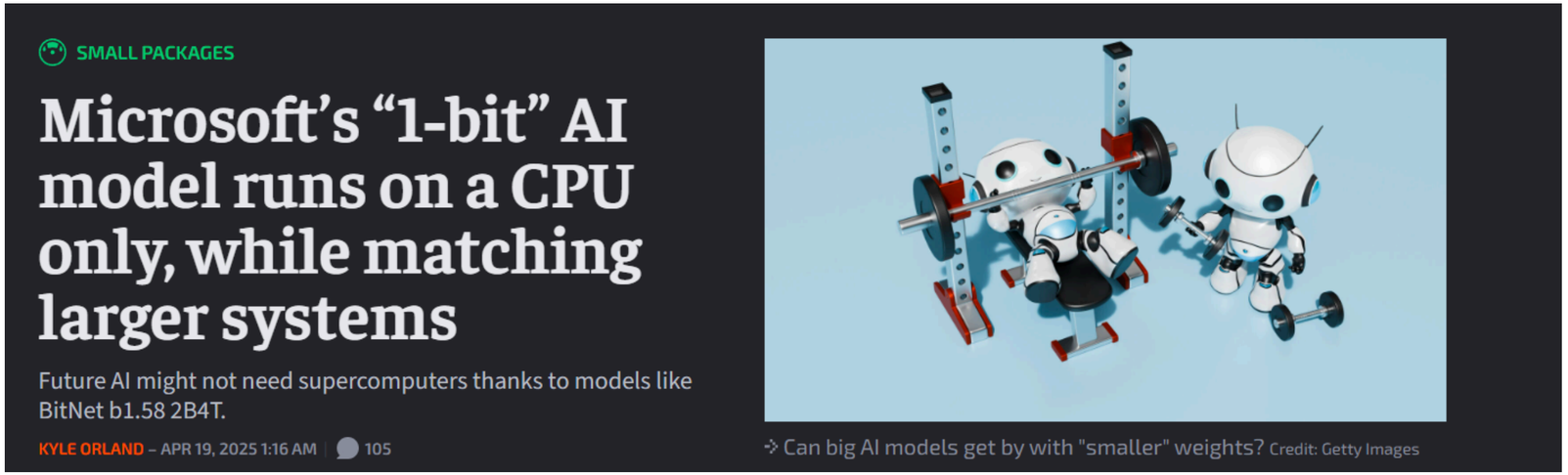
Lack of Real Understanding

These models generate plausible text without true comprehension or real-world awareness, limiting reasoning.

The Shift of LLMs from GPU to CPU-Based Inference



- Normally, each GPU kernel (chef) does its job and stores intermediate results in slow global memory (the shelf)
- With kernel fusion, multiple operations are combined into one kernel (both chefs at the counter), so intermediate results are passed directly using fast registers or shared memory (no shelf)
- This reduces memory traffic, speeds up computation, and makes the whole system more efficient



Real-World Applications of

Content Generation

LLMs create articles, marketing copy, and creative writing efficiently, supporting media and advertising industries.

Chatbots and Assistants

They power conversational agents for customer support and personal virtual assistants improving user interactions.

Translation and Coding

LLMs offer accurate language translation services and assist programmers with code completion and bug detection.

Conclusion: The Future of LLMs

Ongoing Innovation

Research continues to address current limitations, improving accuracy, efficiency, and understanding in LLMs.

Transformative Potential

LLMs are poised to impact industries from education to healthcare, amplifying human capabilities through AI.

Ethics and Responsibility

Responsible development and awareness of ethical implications remain crucial to harnessing the power of LLMs safely.

Dhanyavad