

# Efficient Fine-Tuning of LLMs

with Low-Rank Adaption (LoRA)



Presented by

*Greg Loughnane, Founder & CEO  
Chris Alexiuk, Co-Founder & CTO*



# ALIGNING OUR AIM

lide



# BY THE END OF TODAY...

- What is **fine-tuning**, **PEFT**, and **LoRA**
- **How to fine-tune LLMs** with open-source tools from **Hugging Face**



# OVERVIEW

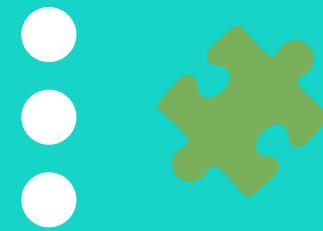
- 🚧 Prototyping
- ⚖️ Fine-Tuning
- ⚡ Parameter-Efficient Fine-Tuning (PEFT)
- 🏅 Low-Rank Adaption (LoRA)
- 👤 FT with Hugging Face
- ❓ Conclusions, QA





# PROTOTYPING





# PROTOTYPING LLM APPS

1. Prompt Engineering
2. Question Answering Systems
3. Fine-Tuning Models



# The optimization flow

Context  
optimization

What the model  
needs to know

RAG

All of the above

Prompt engineering

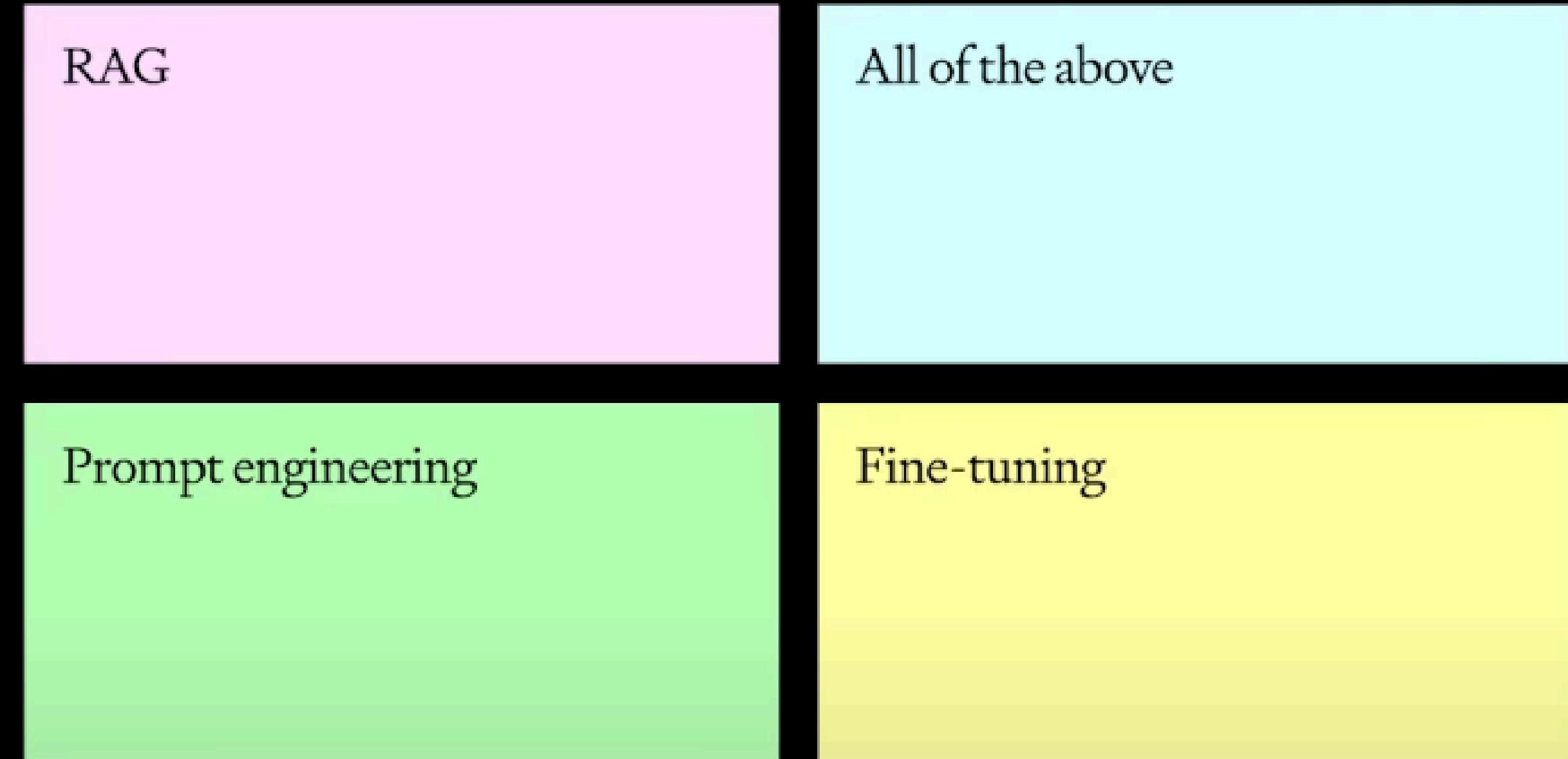
Fine-tuning

LLM optimization  
How the model needs to act



# The optimization flow

Context  
optimization  
  
What the model  
needs to know

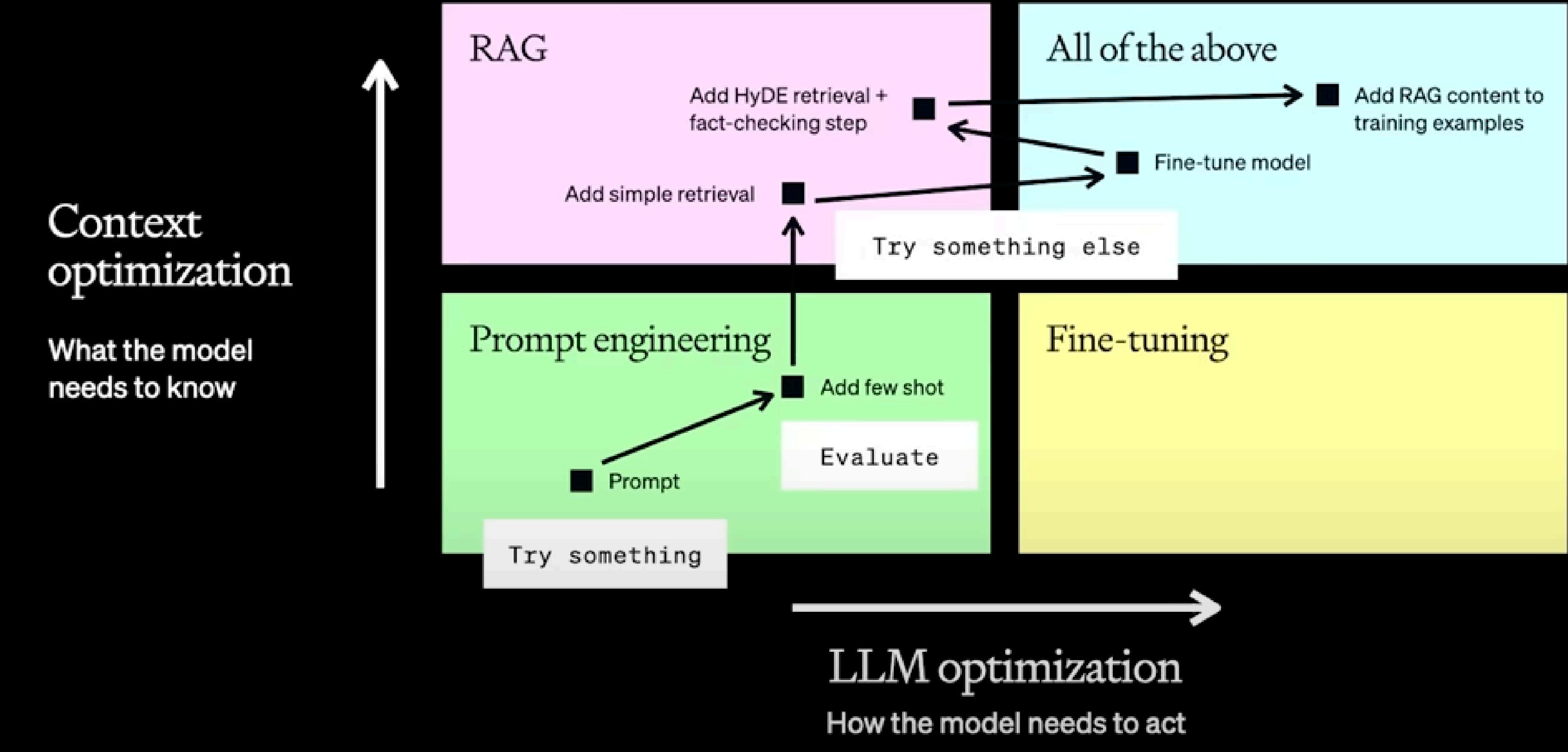


LLM optimization  
How the model needs to act

# The optimization flow

Context optimization

What the model needs to know



# :Small Language Models (SLMs)

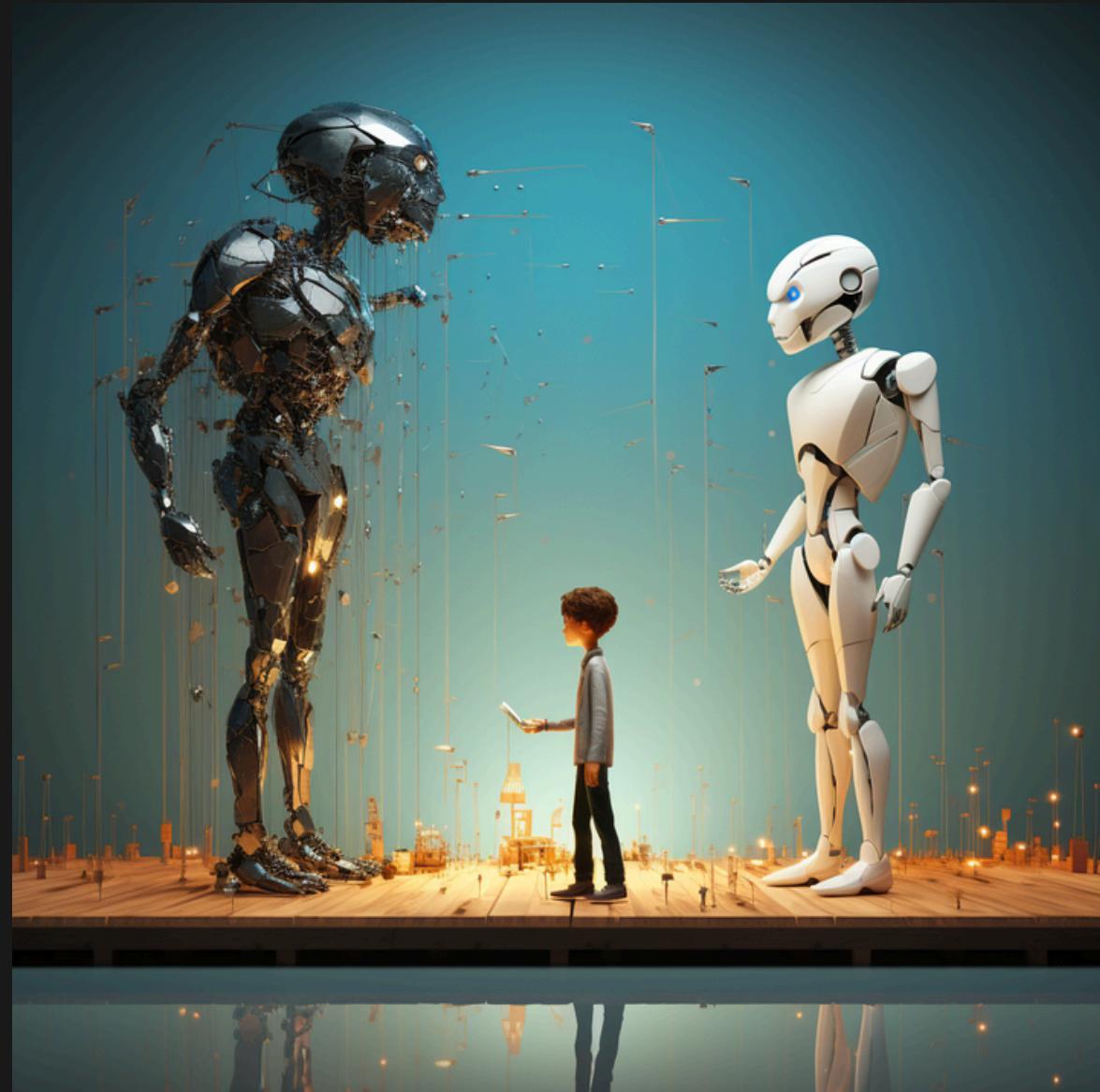
Efficiency, Transparency , Accuracy, Security



**TinyStories: How Small Can Language Models Be and Still Speak...**

Language models (LMs) are powerful tools for natural language processing, but they often struggle to produce coherent and fluent text when they are small. Models with around 125M parameters such...

[arXiv.org](https://arxiv.org)



**Textbooks Are All You Need**

We introduce phi-1, a new large language model for code, with significantly smaller size than competing models: phi-1 is a Transformer-based model with 1.3B parameters, trained for 4 days on 8...

[arXiv.org](https://arxiv.org)





# FINE-TUNING

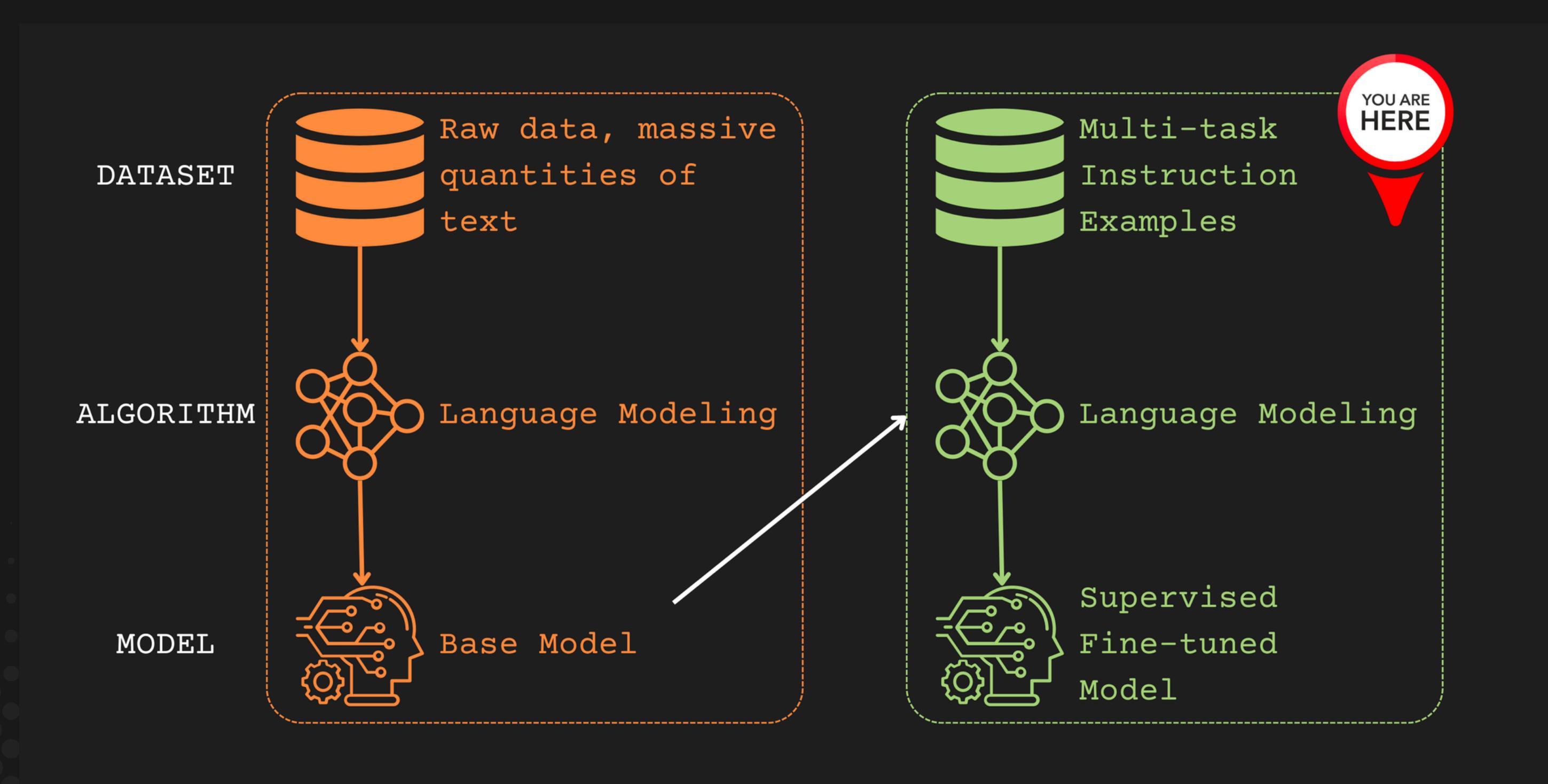
FINE TUNE...



HELP YOU TO, WE SHALL

Henry Rappe

# FINE-TUNING LLMS

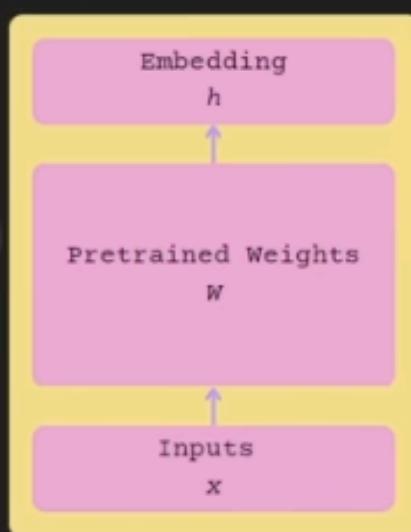




# MECHANICS OF FINE-TUNING

## Fine-tuning Explained

### 1. Forward Pass

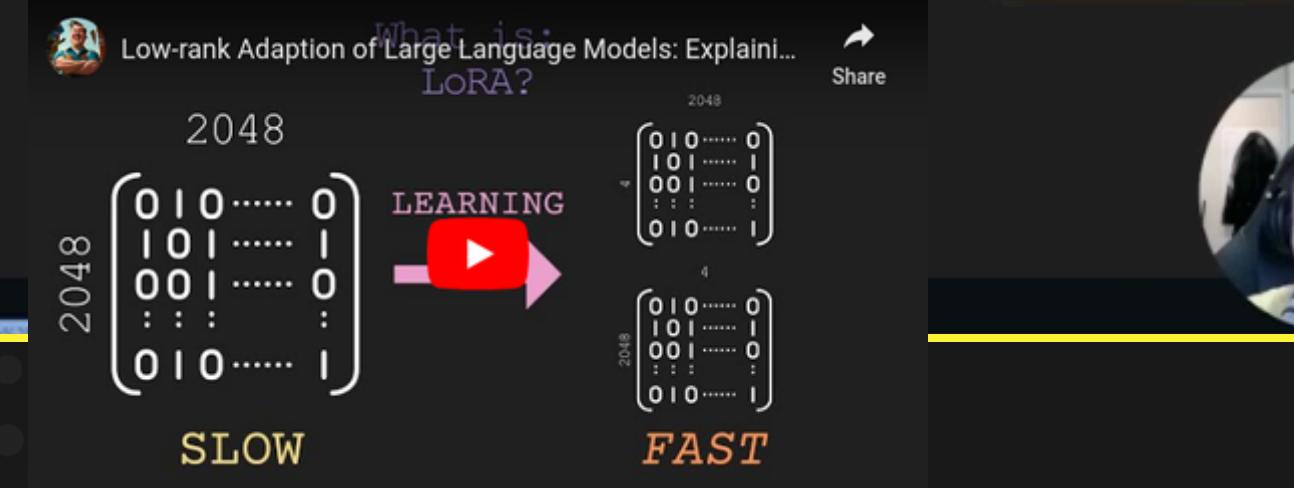


### 2. Backpropagation

$$\Delta W = \alpha(-\nabla L_w)$$

Get  $\Delta W$  via backprop

### 3. Updated Forward Pass



CS.LG] 11 Jan 2024

## Fine-Tuning Language Models with Just Forward Passes

Sadhika Malladi\*

Tianyu Gao\*

Eshaan Nichani

Alex Damian

Jason D. Lee

Danqi Chen

Sanjeev Arora

Princeton University  
{smalladi, tianyug, eshnich, ad27, jasonlee, danqic, arora}@princeton.edu

### Abstract

Fine-tuning language models (LMs) has yielded success on diverse downstream tasks, but as LMs grow in size, backpropagation requires a prohibitively large amount of memory. Zeroth-order (ZO) methods can in principle estimate gradients using only two forward passes but are theorized to be catastrophically slow for optimizing large models. In this work, we propose a memory-efficient zeroth-order optimizer (MeZO), adapting the classical ZO-SGD method to operate in parallel on multiple GPUs. MeZO achieves state-of-the-art performance on several benchmarks while being orders of magnitude faster than backpropagation.

## Paper page - Fine-Tuning Language Models with Just Forward Passes

Join the discussion on this paper page

huggingface



# PEFT



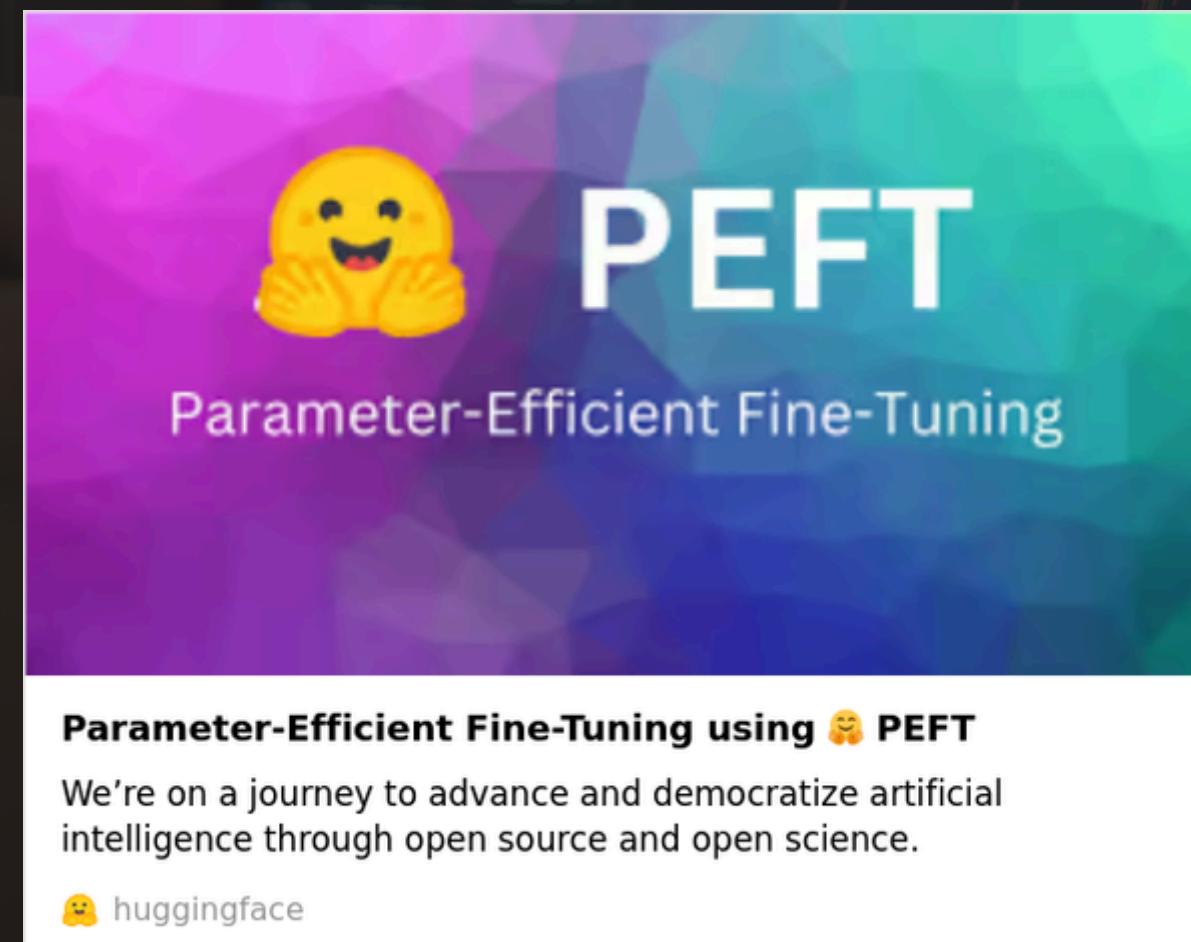
I just spent our entire  
AI budget on fine-  
tuning a model



But at least we have  
a working model  
now, right?



“Fine-tuning [pretrained LLMs] on downstream datasets results in huge performance gains when compared to [off-the-shelf LLMs, zero-shot].”



# WHY PEFT?

LLMs are so big!

7B, 13B, 34B, 70B



imgflip.com



# SCALING LAWS

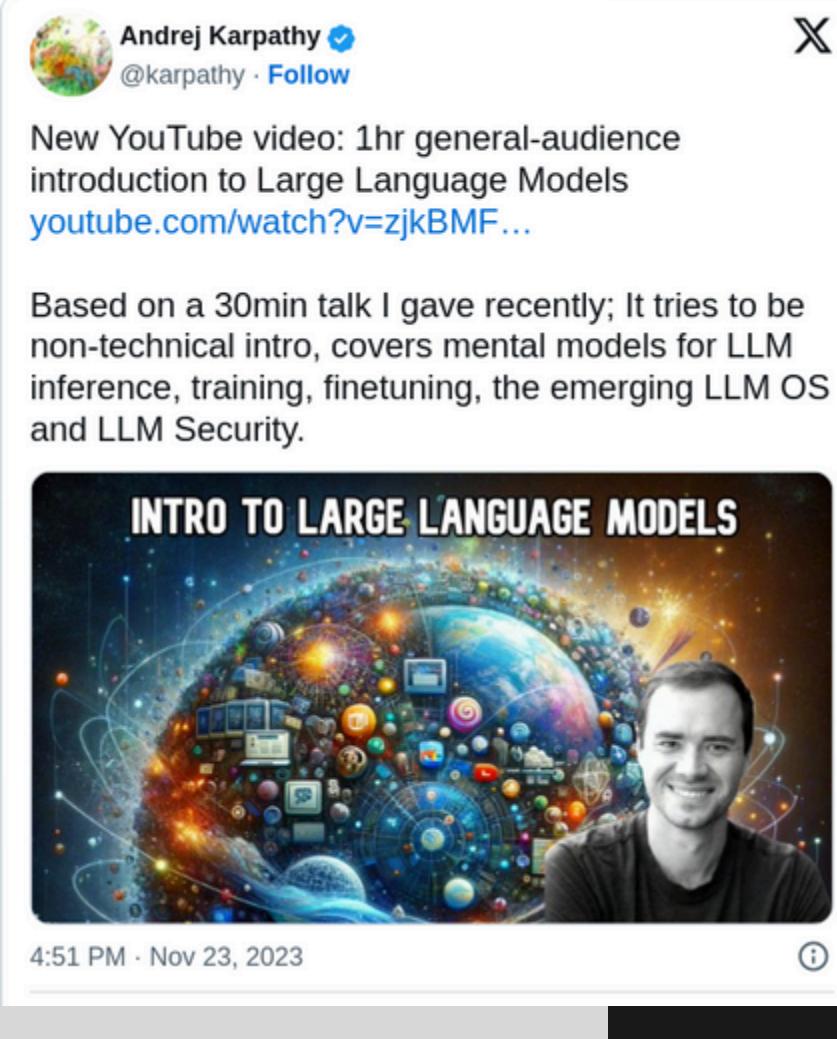
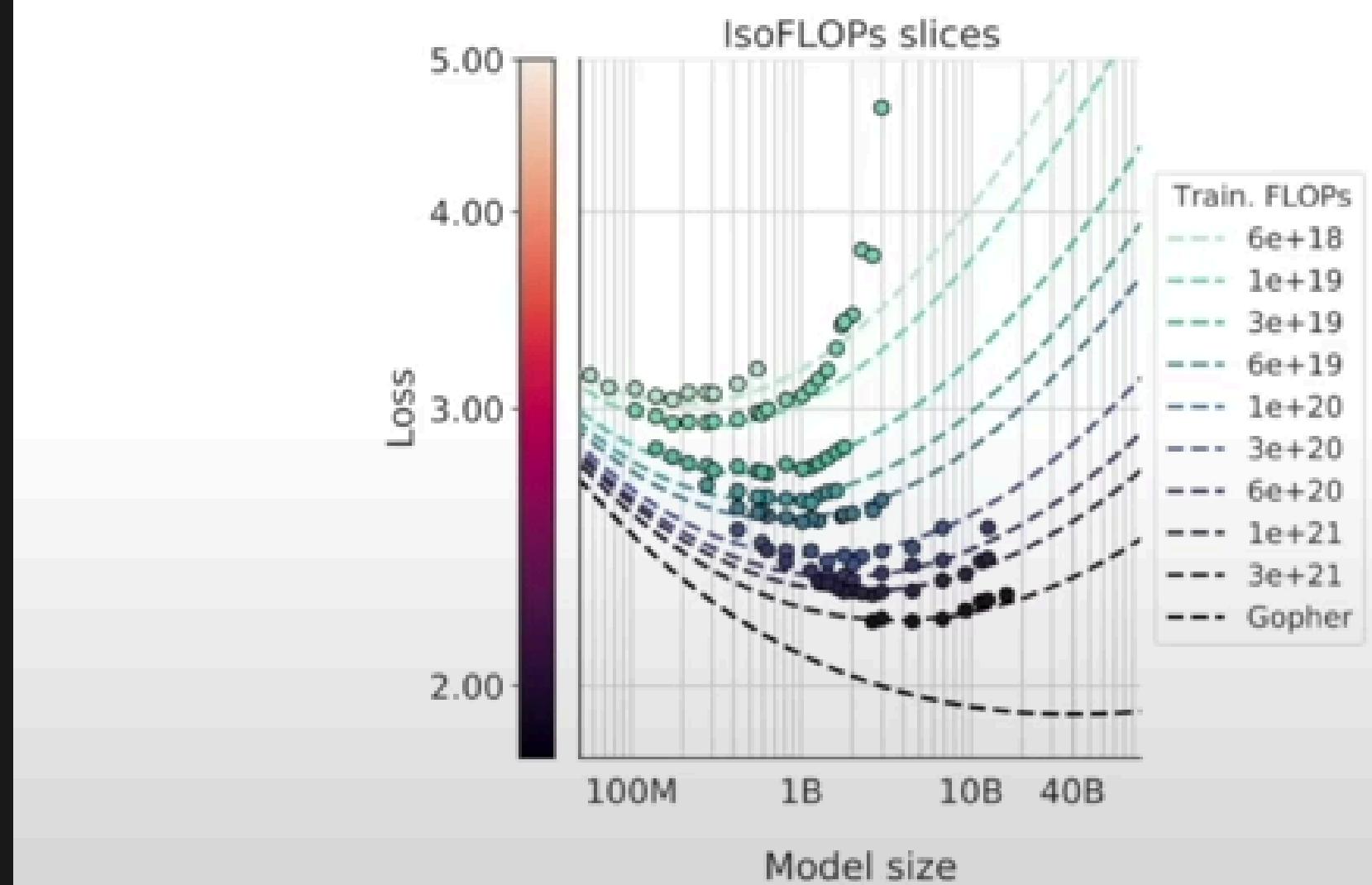
1.  $N$  = number of parameters

2.  $D$  = amount of text

*"If you train a bigger model on more text, we have a lot of confidence that next word prediction will improve." ~*

**Andrej Karpathy**

=> We can expect more intelligence “for free” by scaling



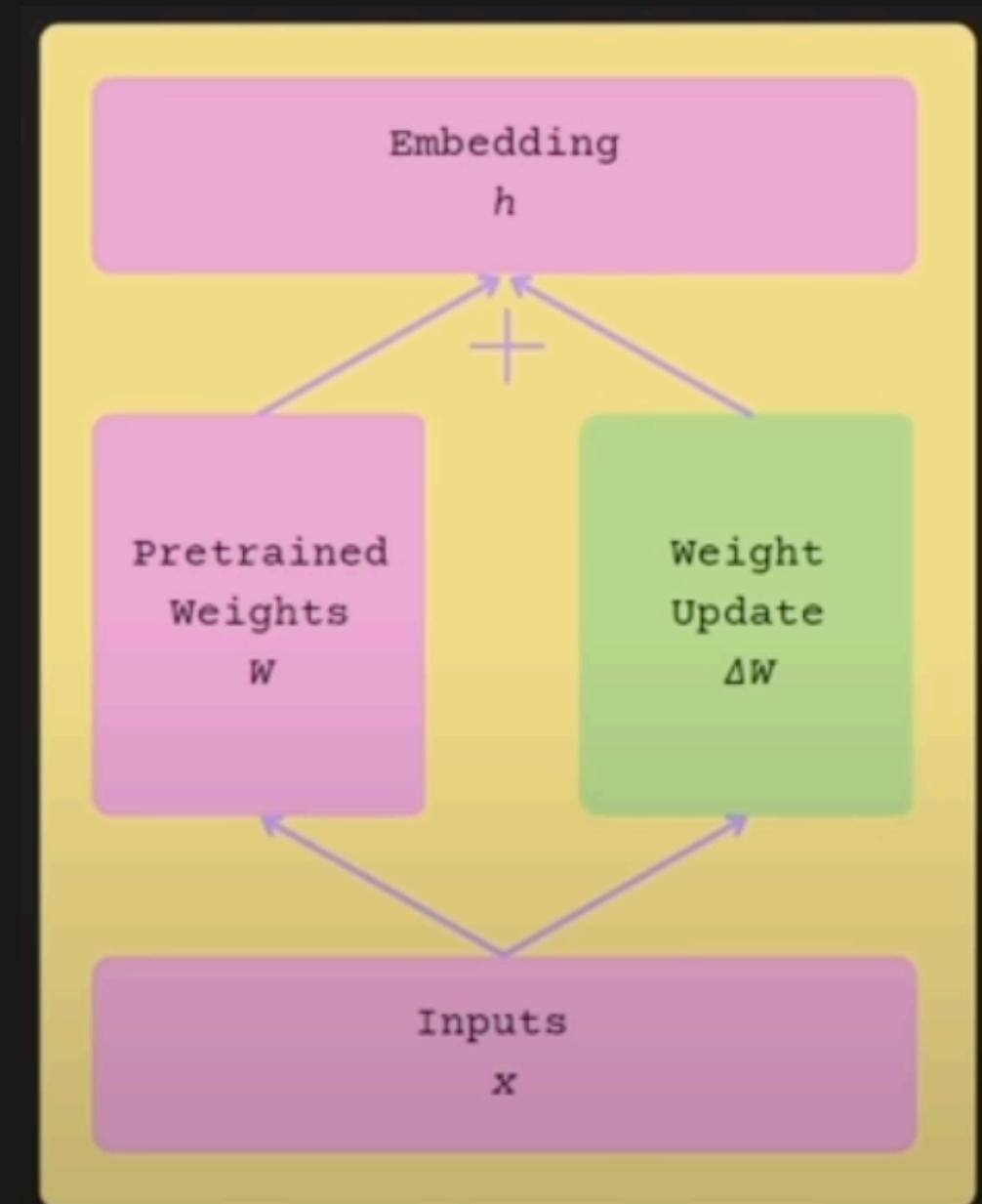
Training Compute-Optimal Large Language Models

We investigate the optimal model size and number of tokens for training a transformer language model under a given compute budget. We find that current large language models are...

[arXiv.org](https://arxiv.org/)

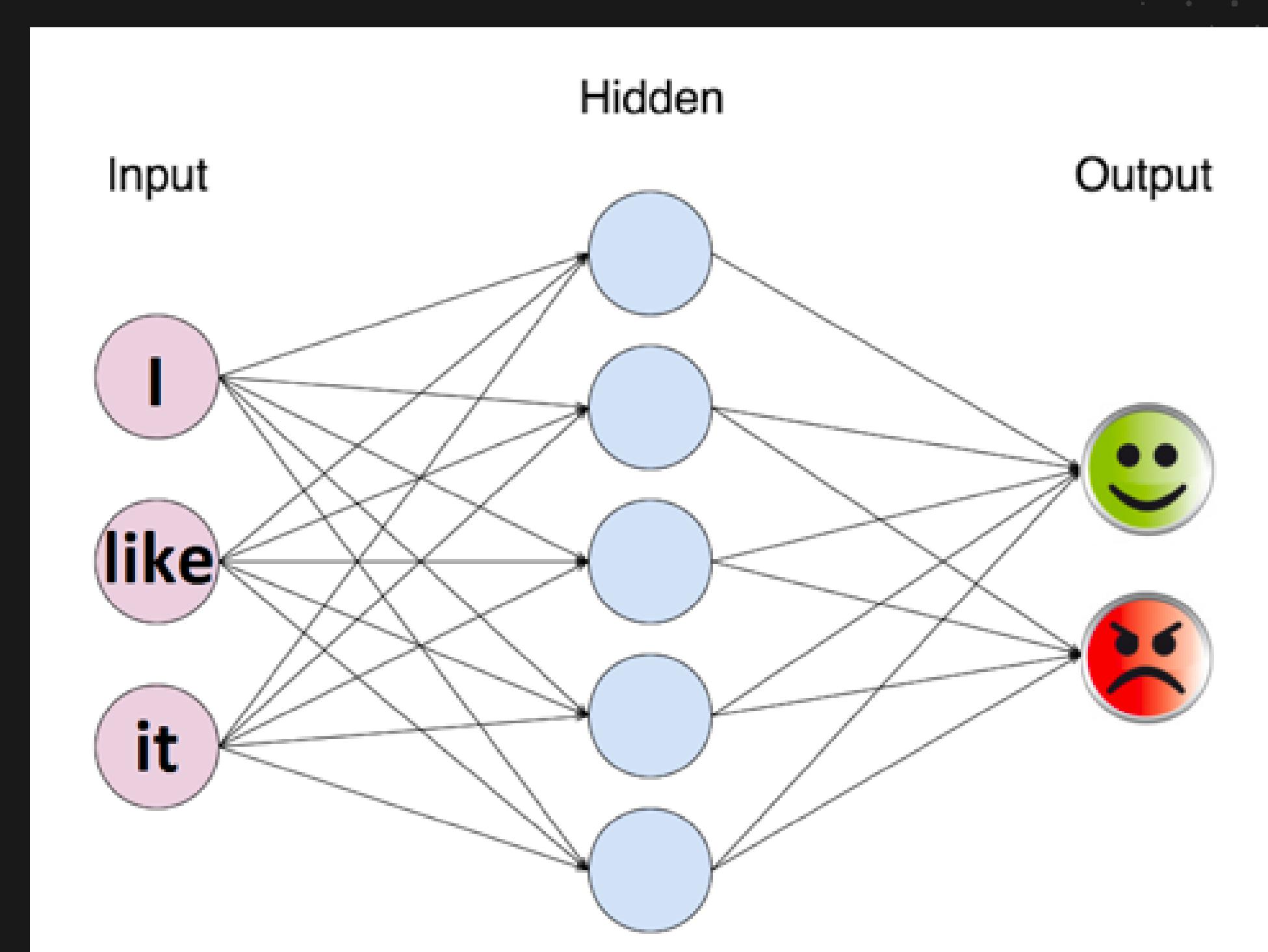
# PARAMETERS

- **Weights = parameters**
- Parameters = *Floating point numbers*
- *Full precision = 32-bits ...*



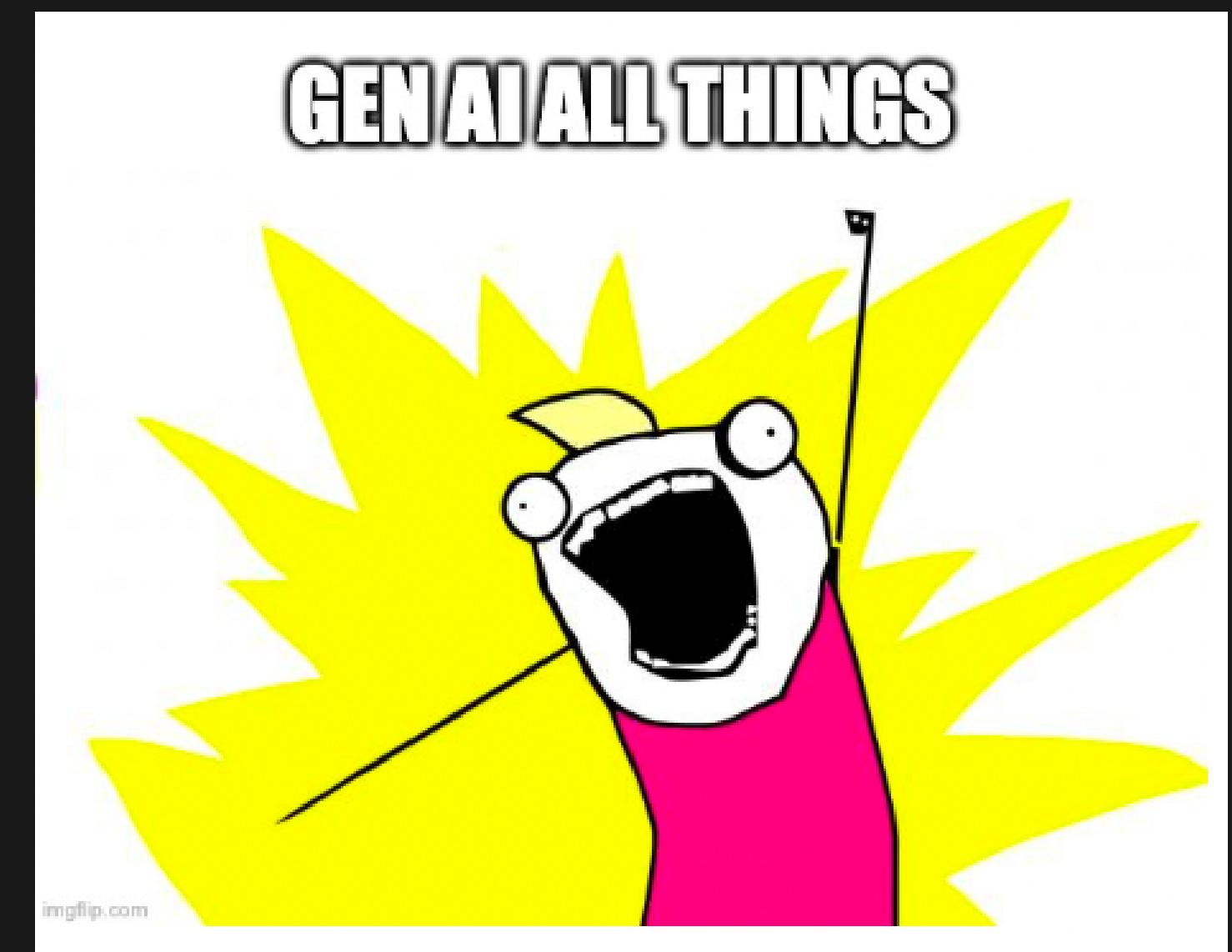
# NEURAL NETWORK TRAINING

Pre-trained = mostly tuned!



# TWO CHALLENGES

1. Full fine-tuning often **infeasible** on consumer hardware.
2. Storing and deploying fine-tuned models independently for each downstream task becomes **expensive**.



# THE SOLUTION

1. Only **fine-tune** a **small number of** (extra) model **parameters**
2. Better at:
  - a.small data
  - b.out-of-domain

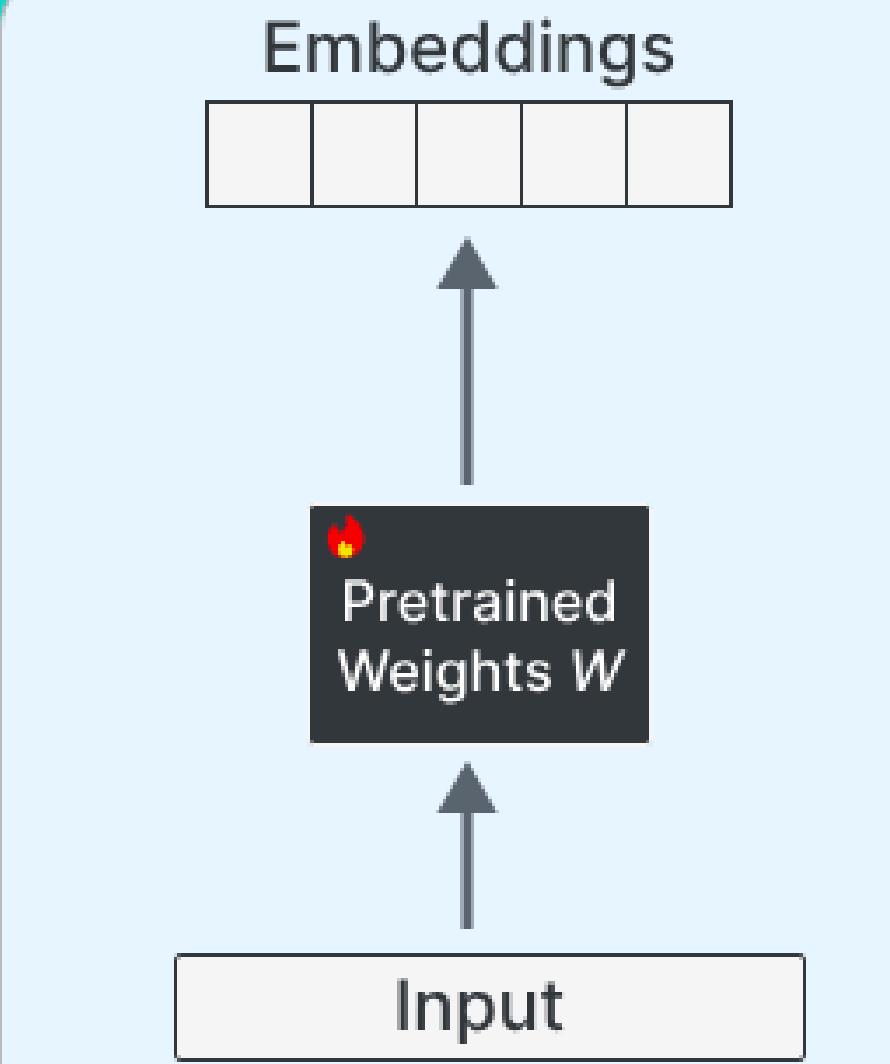




# LoRA

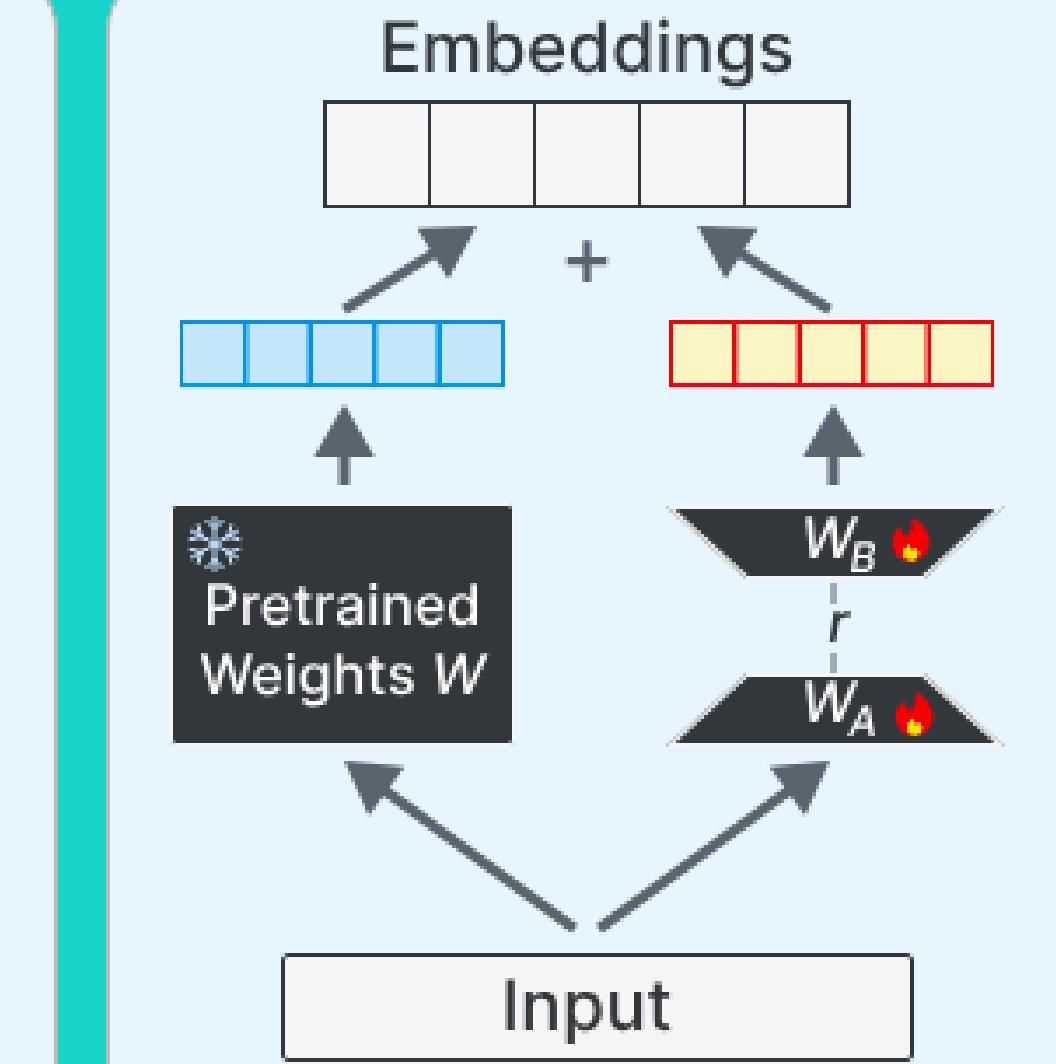
## Regular Fine-Tuning

Update **all** weights

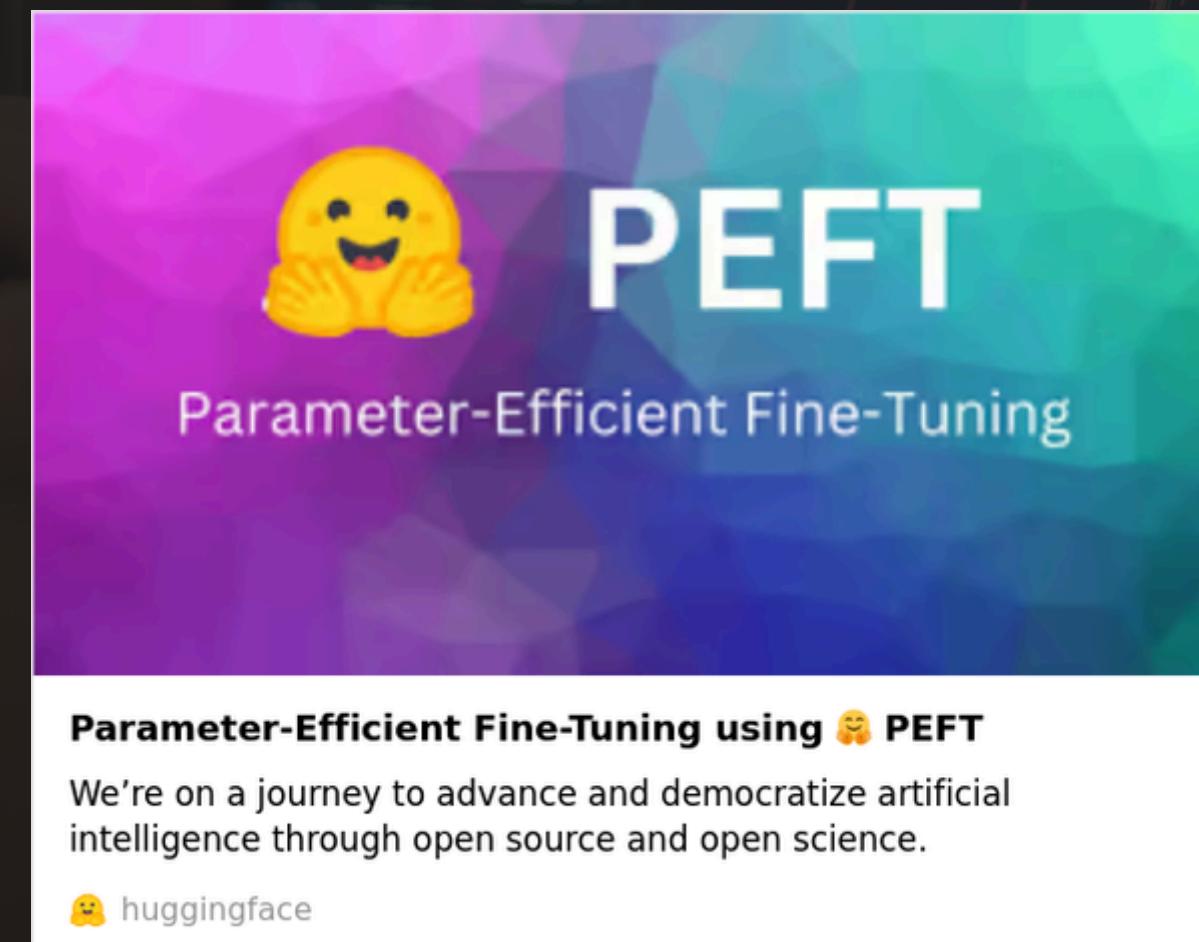


## Low-Rank Adaptation

Update a **small representation** of the weights



LORA is the #1 PEFT Method  
you should know!



# ⋮ FINE TUNING REALLY WORKS!

“Common pre-trained models have very low ‘*intrinsic dimension*’; [there exists a] re**parameter**ization that is as effective for **fine-tuning** as the full parameter space.”



**Intrinsic Dimensionality Explains the Effectiveness of Language...**

Although pretrained language models can be fine-tuned to produce state-of-the-art results for a very wide range of language understanding tasks, the dynamics of this process are not well...

# ⋮ LoRA Really Works!

Hypothesis:

“The updates to the weights also have a low “*intrinsic rank*” during adaption.”



## **LoRA: Low-Rank Adaptation of Large Language Models**

An important paradigm of natural language processing consists of large-scale pre-training on general domain data and adaptation to particular tasks or domains. As we pre-train larger models, full...

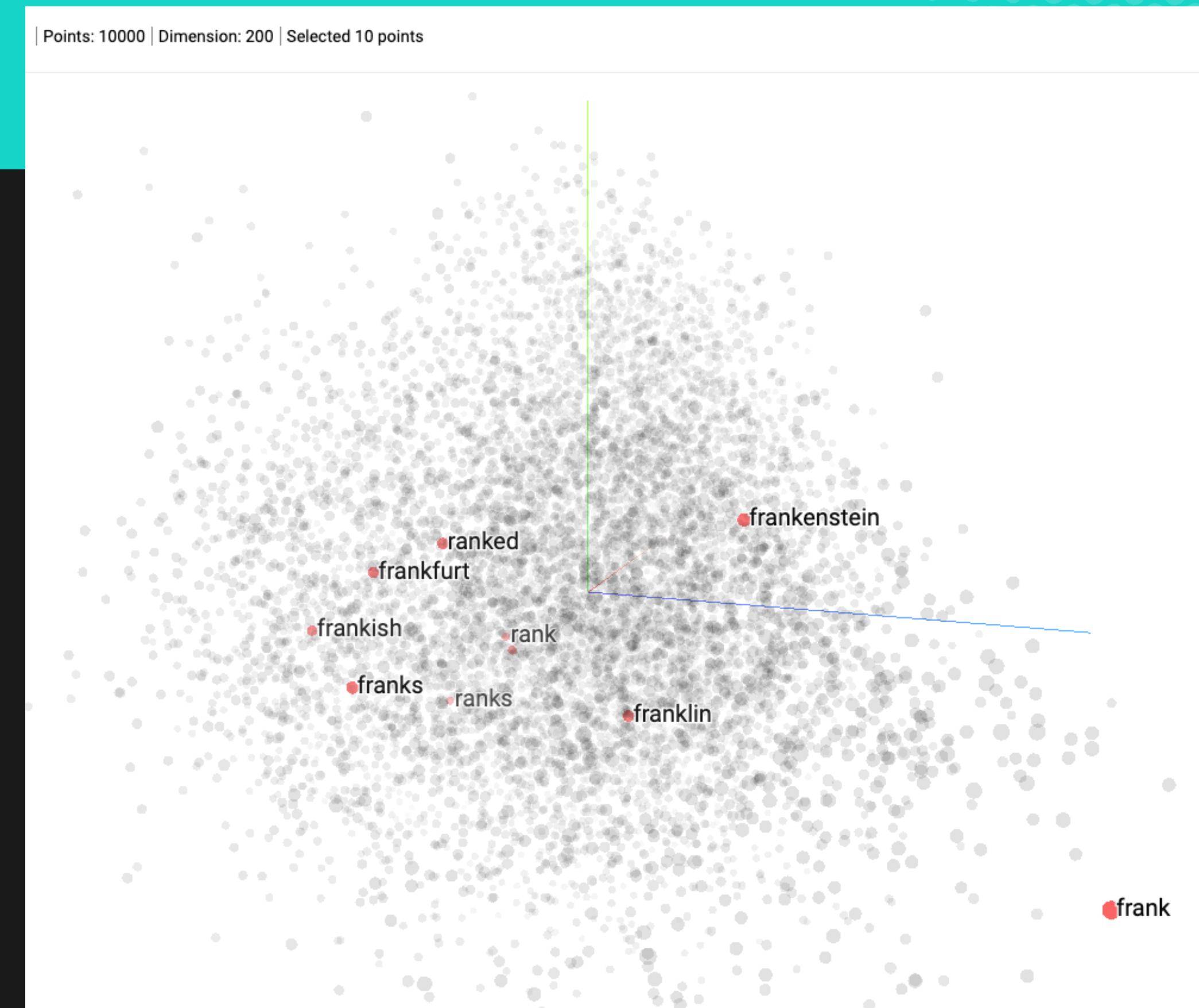


# Matrix Rank

Def:

The **number of linearly independent columns** in the weight/parameter matrix

Proxy for the number of **dimensions with unique information**



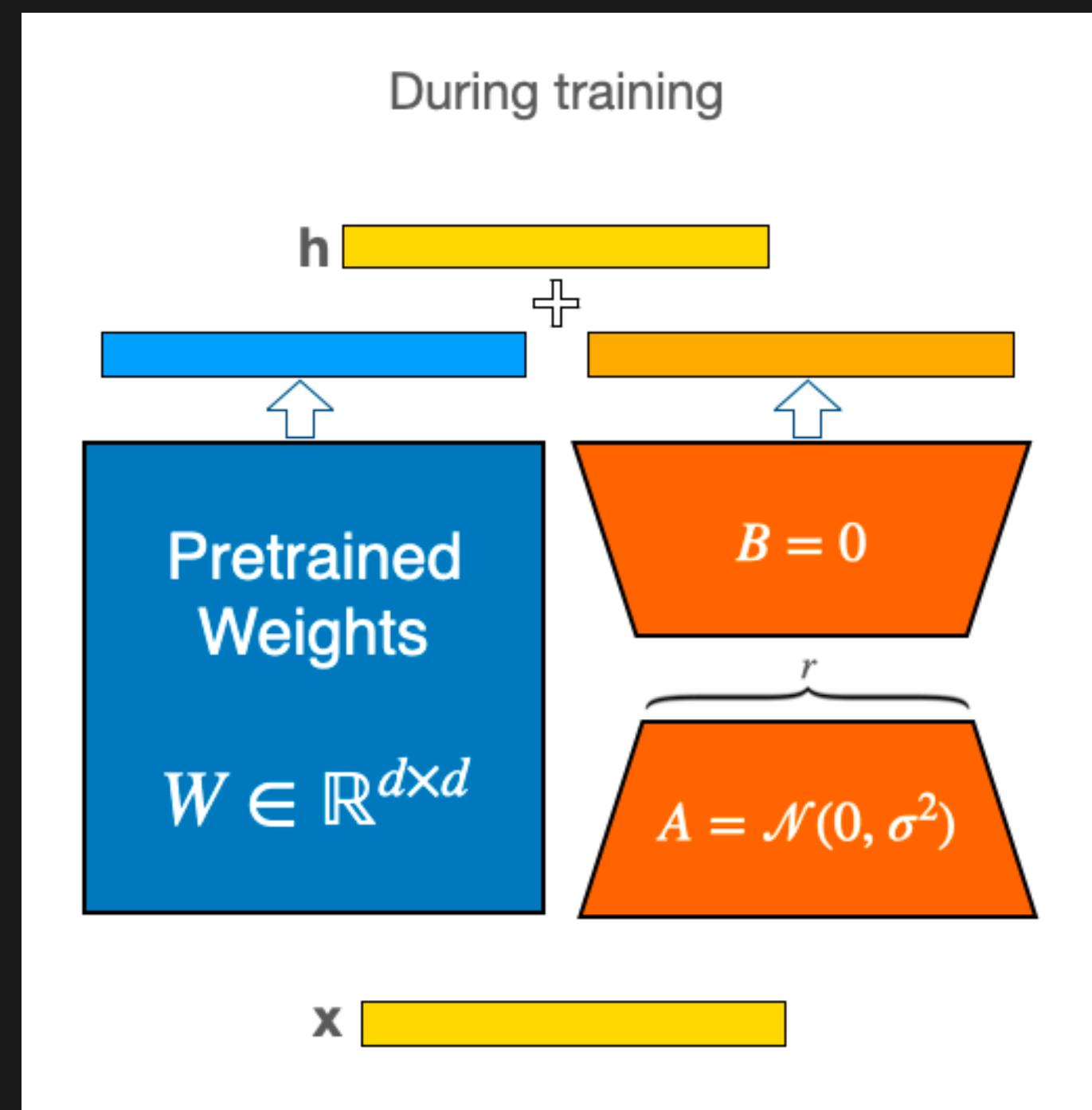
# Matrix Decomposition

Convert a difficult matrix computation problem into several easier tasks.



# How LoRA Works

1. Freezes the **pre-trained** model **weights**
2. Injects **trainable rank decomposition matrices** into each layer of the Transformer architecture



# Matrix Decomposition

Convert a difficult matrix computation problem into several easier tasks.



# LoRA and Transformers



The Lego Blocks of Transformers

Share

AI MAKERSPACE

THE LEGO BLOCKS OF TRANSFORMERS

Wednesday, Nov. 29 5 PM to 8 PM PT

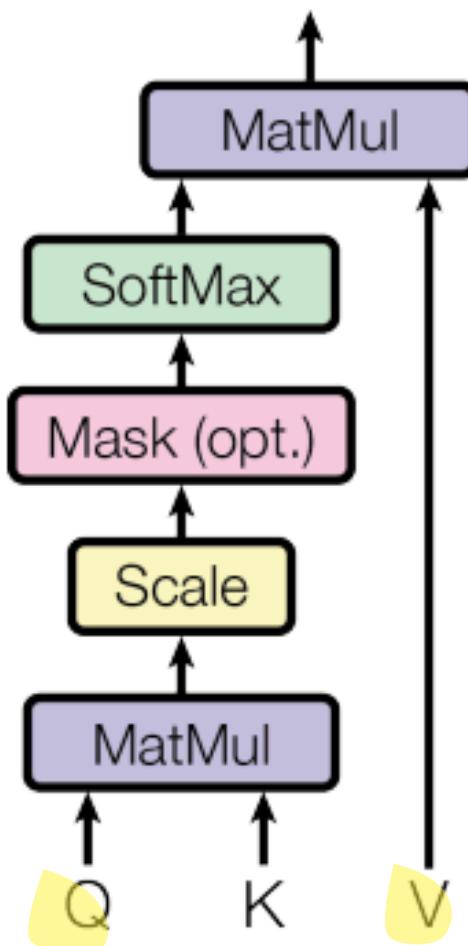
A special event brought to you by the founders of AI Makerspace

Watch on YouTube

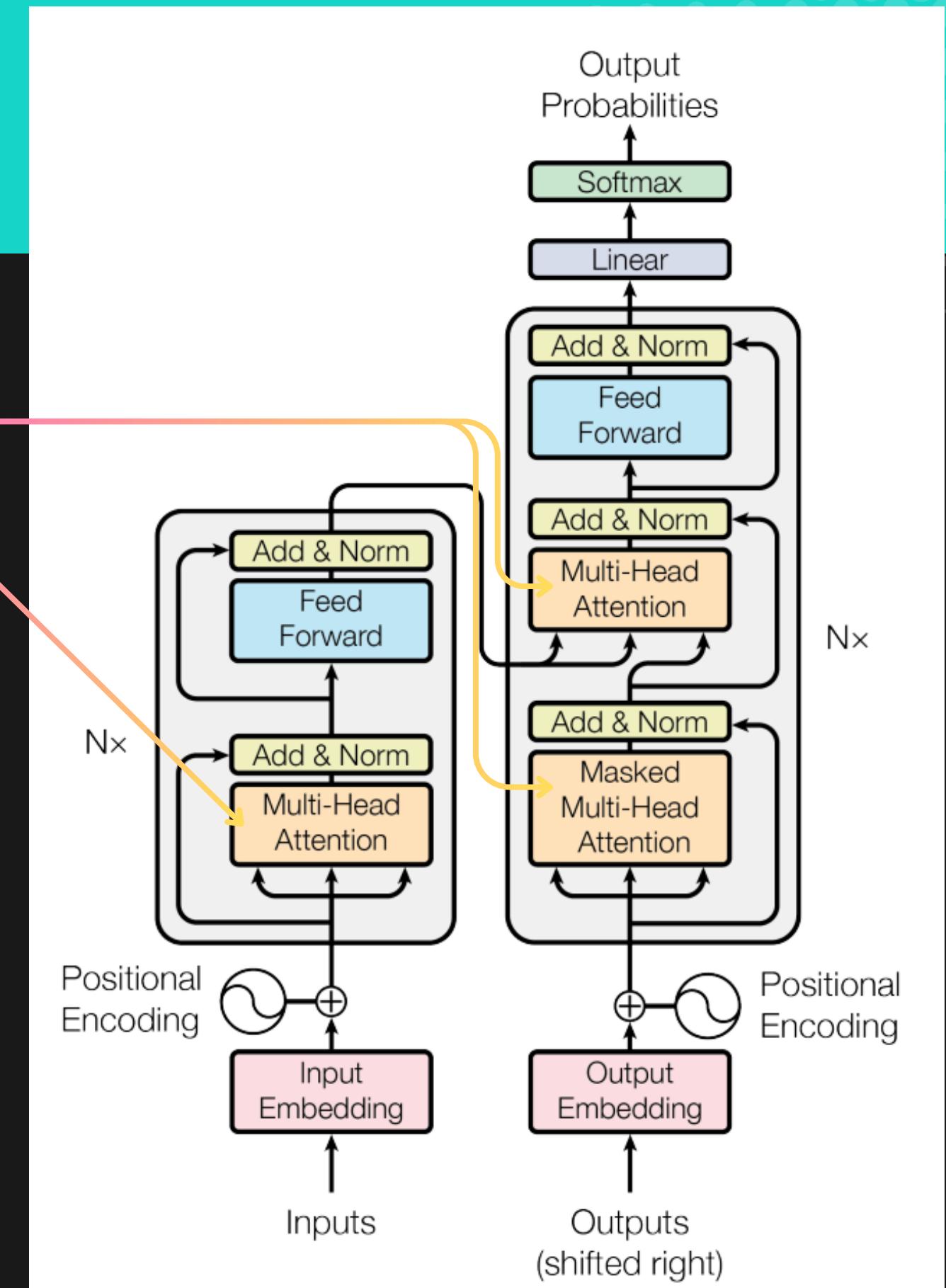
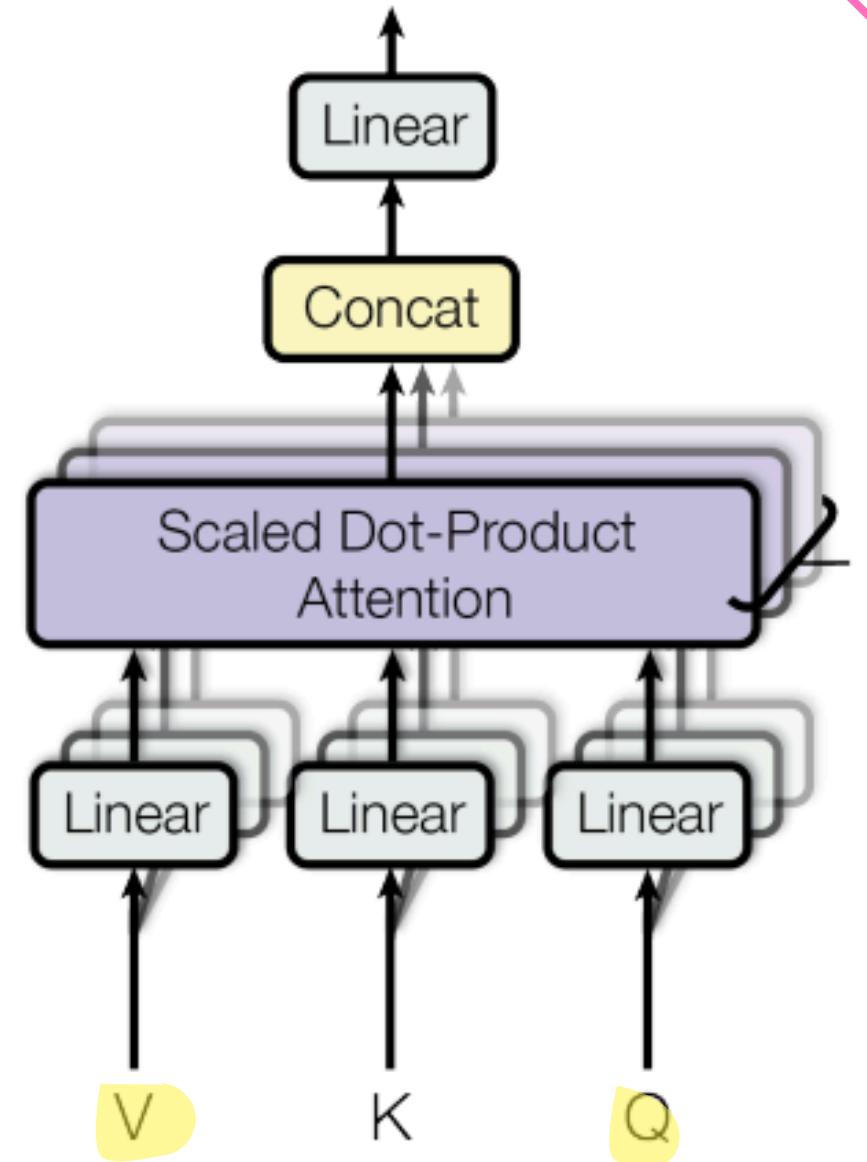
A promotional graphic for a special event. It features the "AI MAKERSPACE" logo at the top left. The main title "THE LEGO BLOCKS OF TRANSFORMERS" is displayed in large, bold, green letters, with a red YouTube play button icon integrated into the letter "O". Below the title, the event details "Wednesday, Nov. 29 5 PM to 8 PM PT" and "A special event brought to you by the founders of AI Makerspace" are written. At the bottom, there is a "Watch on YouTube" button. The background of the graphic is dark with colorful LEGO blocks scattered around.

# Attention

## Scaled Dot-Product Attention

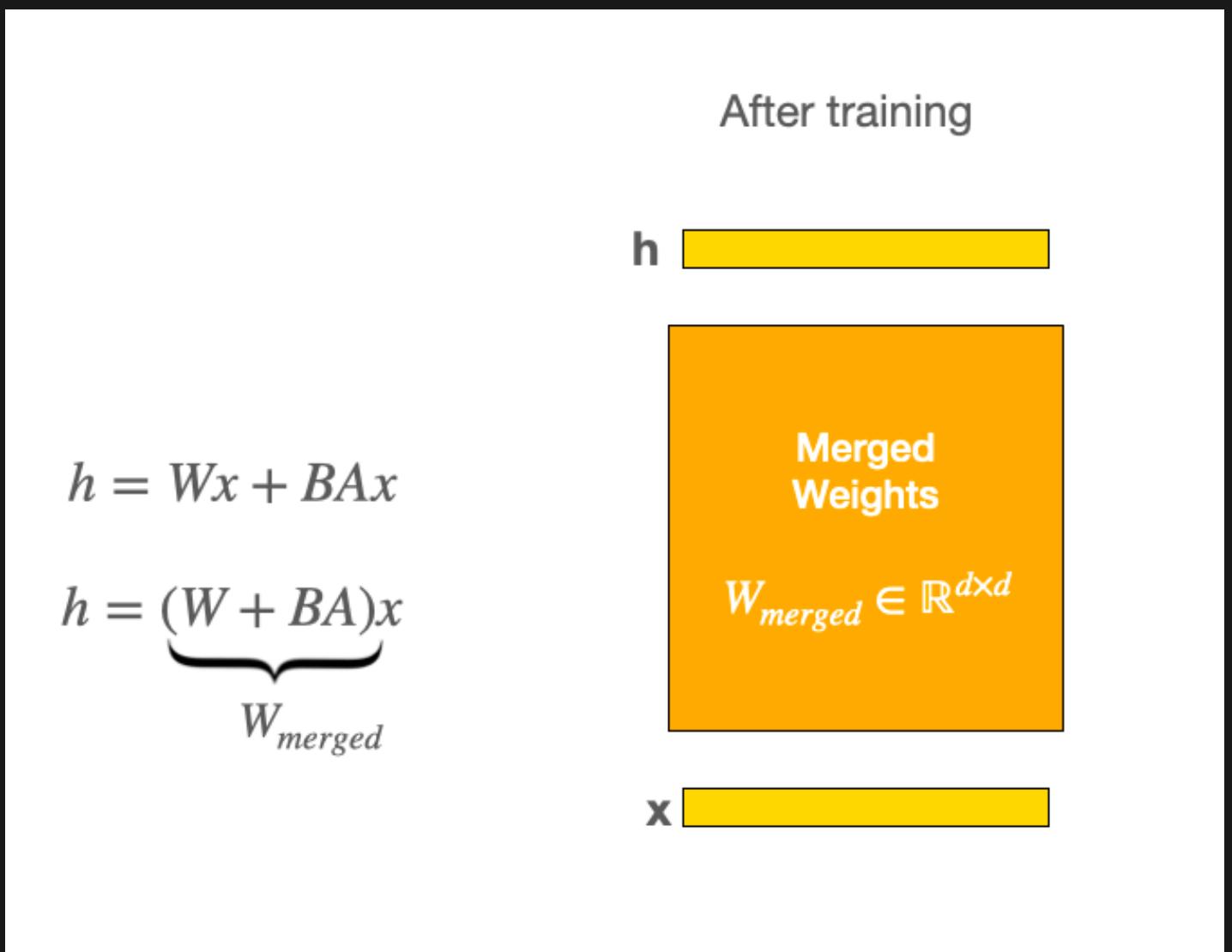


## Multi-Head Attention



# LoRA Advantages

- More **efficient**
- Adapters for **many tasks**
- **Combine** with other PEFT methods
- Comparable to full fine-tuning
- **No** additional inference **latency**



# RECAP



# Fine-Tuning

## Modifying LLM behavior by updating weights



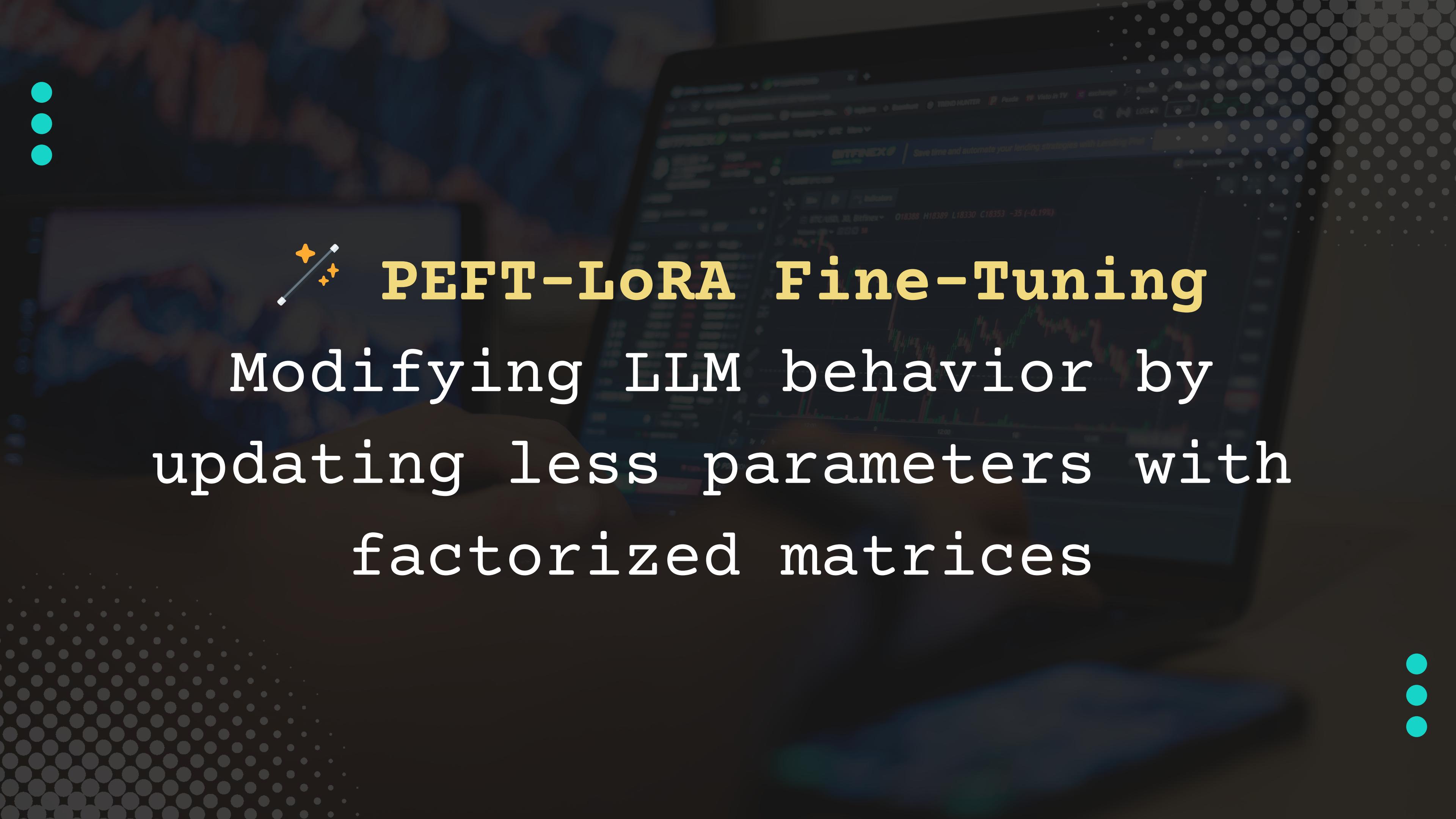
PEFT

# Fine-tuning w/ less parameters



# Low Rank Adaption

## Fine-tuning w/ factorized matrices



# PEFT-LoRA Fine-Tuning

Modifying LLM behavior by  
updating less parameters with  
factorized matrices

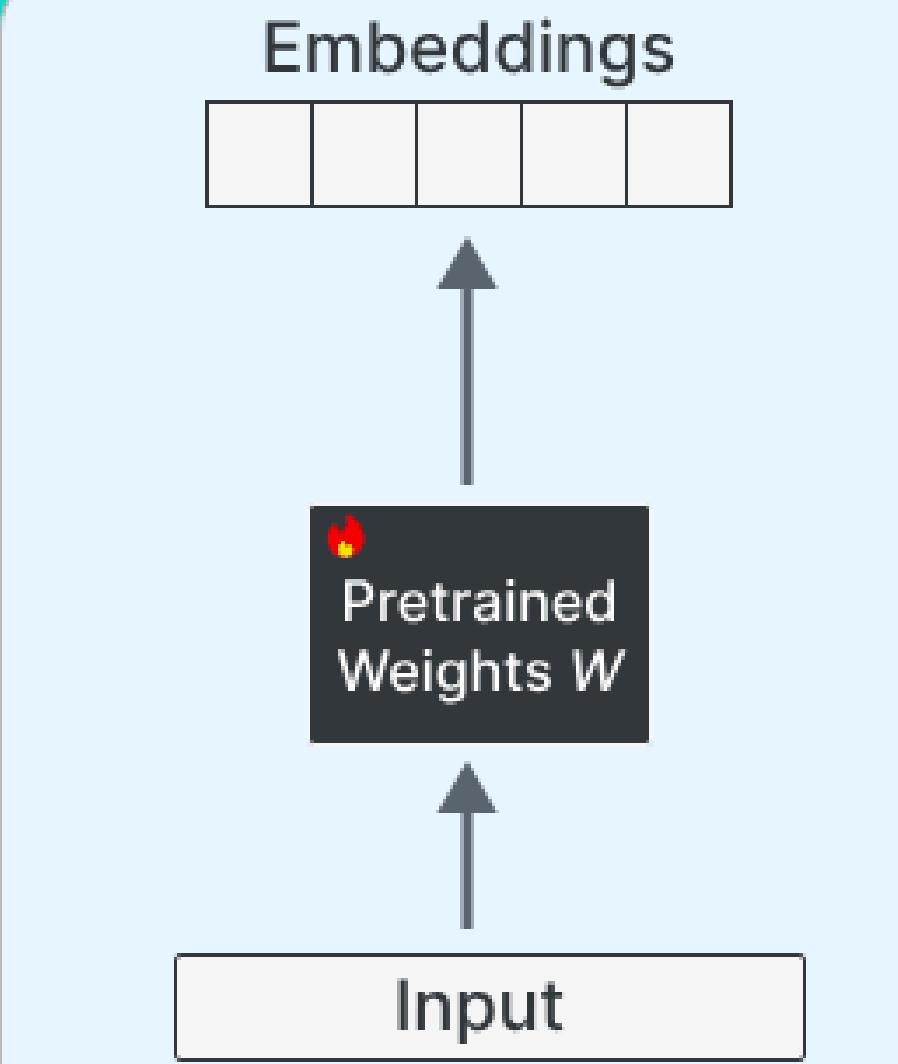




# PEFT-LoRA Fine-Tuning

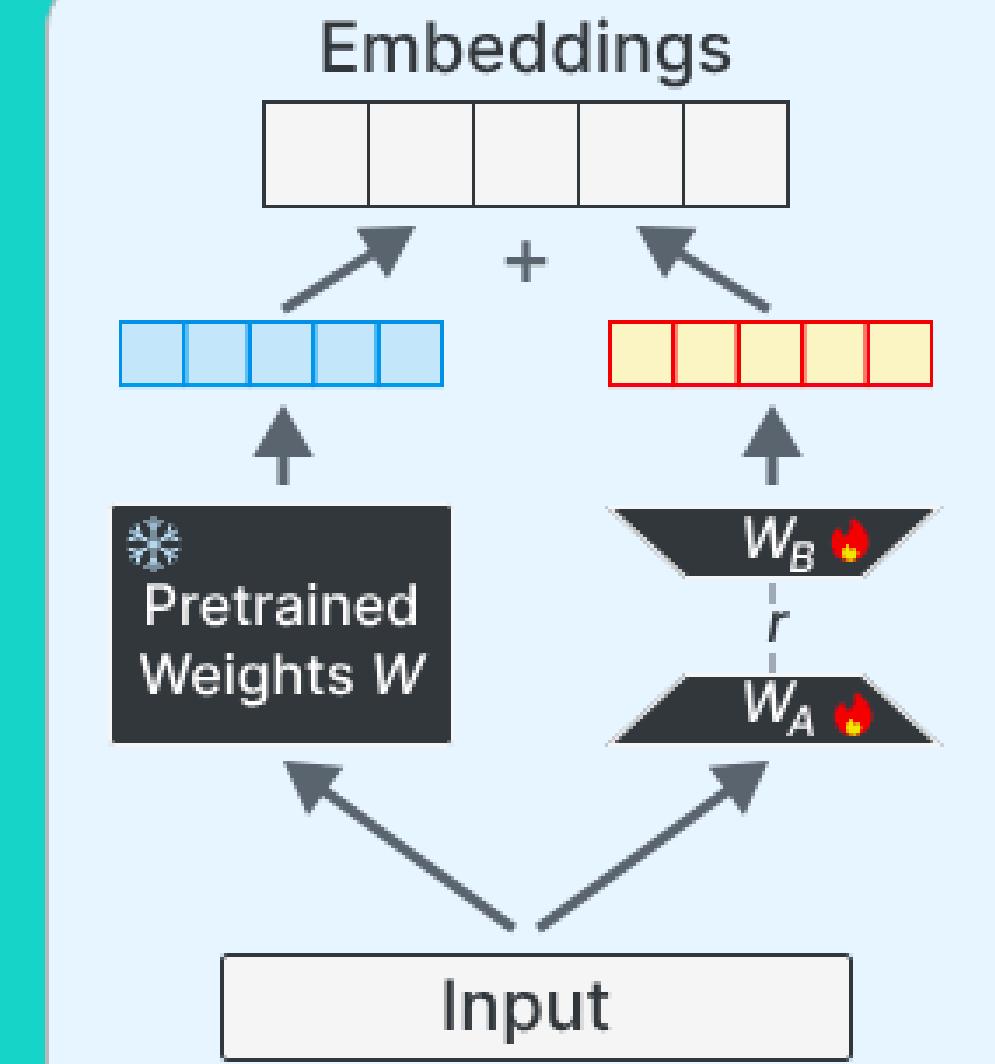
## Regular Fine-Tuning

Update **all** weights



## Low-Rank Adaptation

Update a **small representation** of the weights





# TODAY'S BUILD

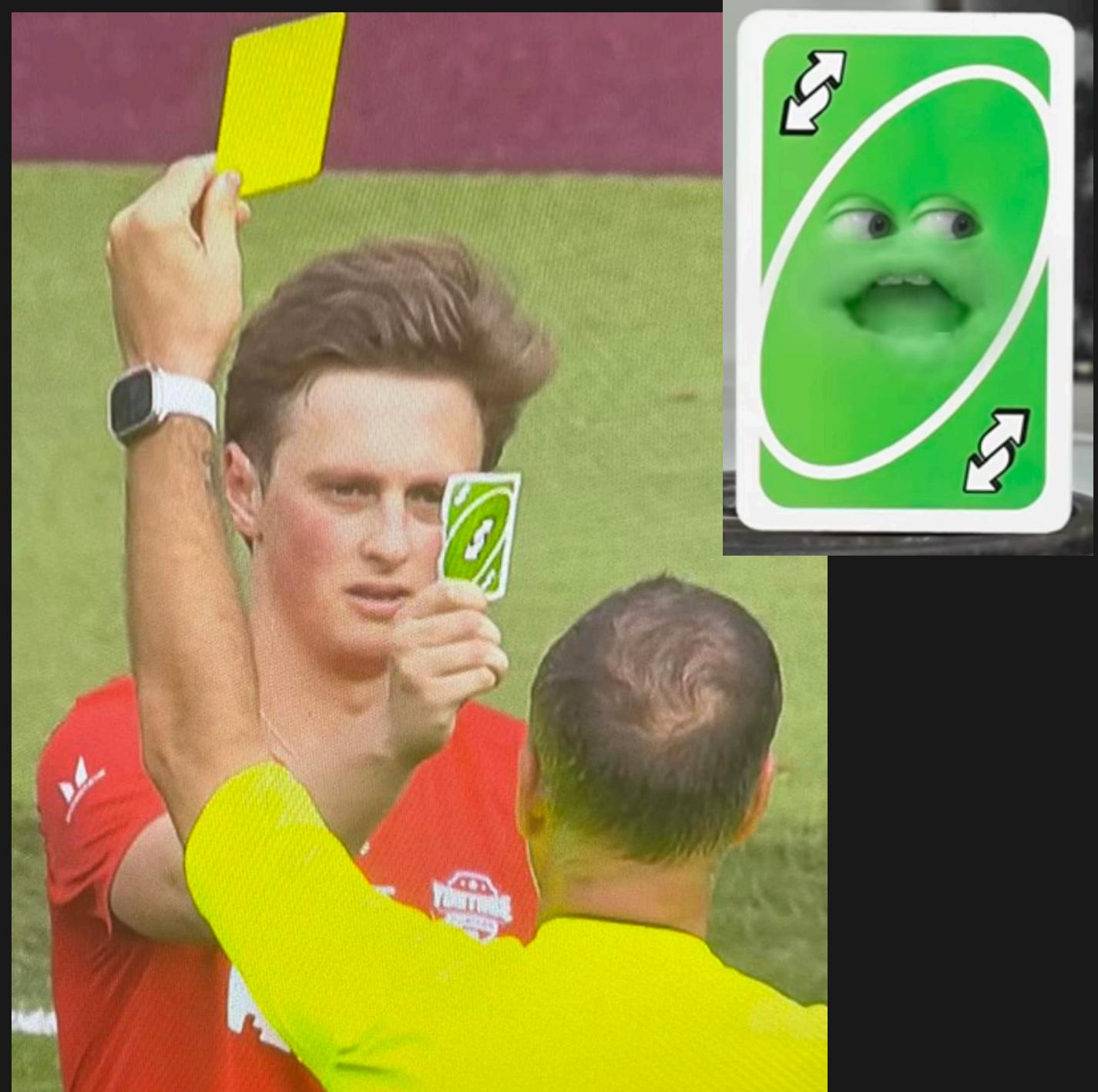
# • AN UNO REVERSE CARD APPLICATION

**Given:**

- **Response** (LLM Output)

**Predict:**

- **Instruction** (Prompt/LLM Input)





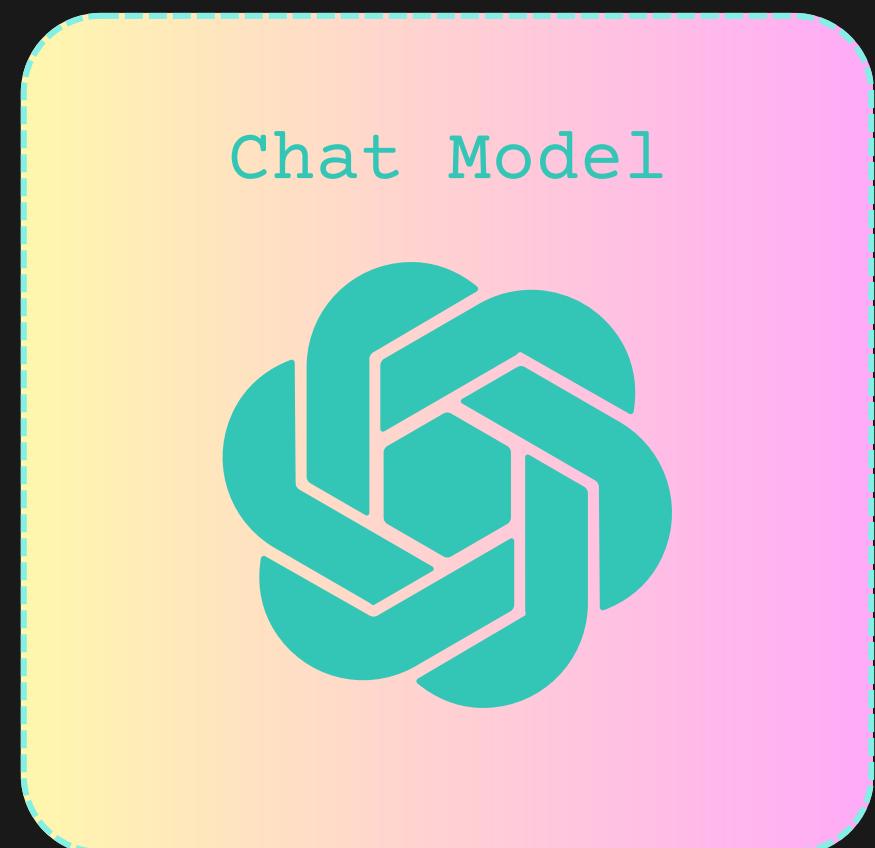
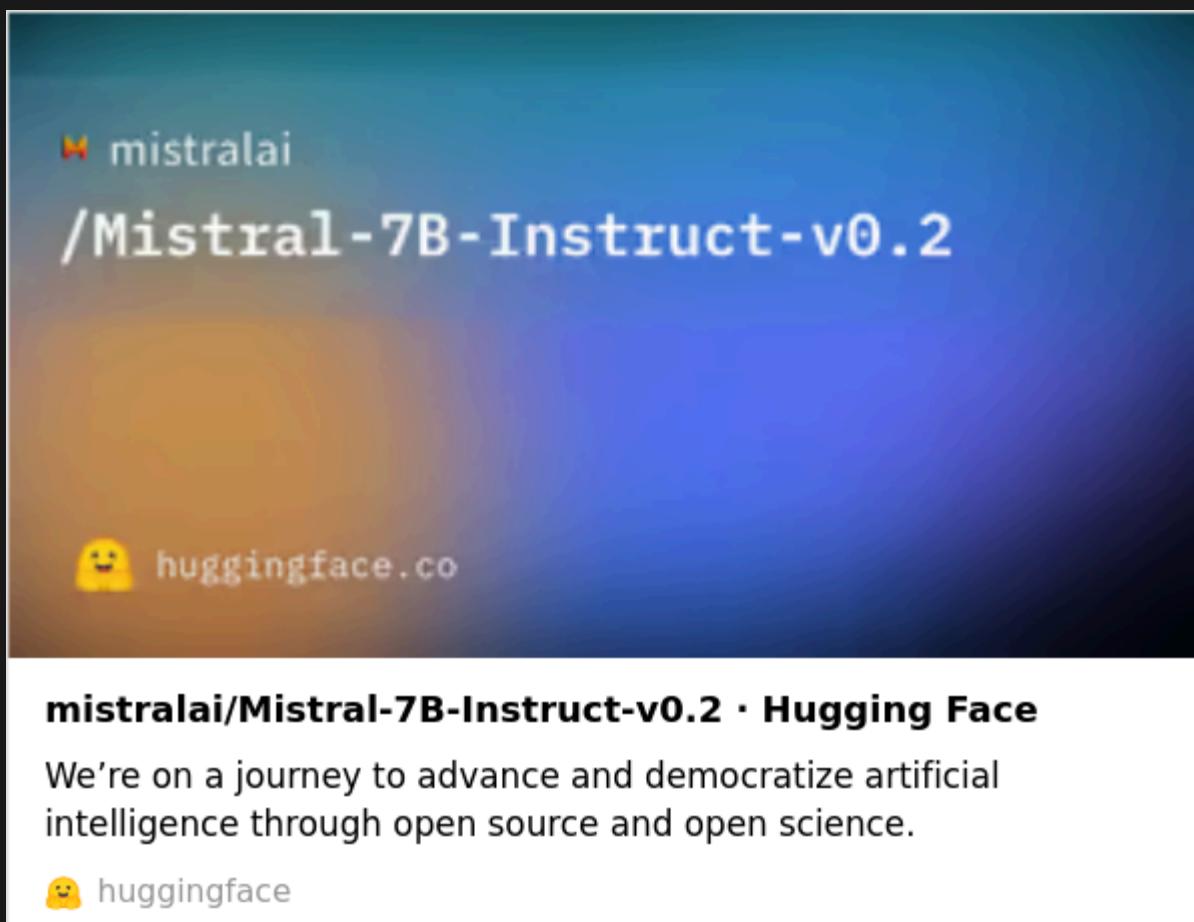
# OUR MODEL



# CHAT (LLM) MODEL

## Chat Model (e.g., LLM)

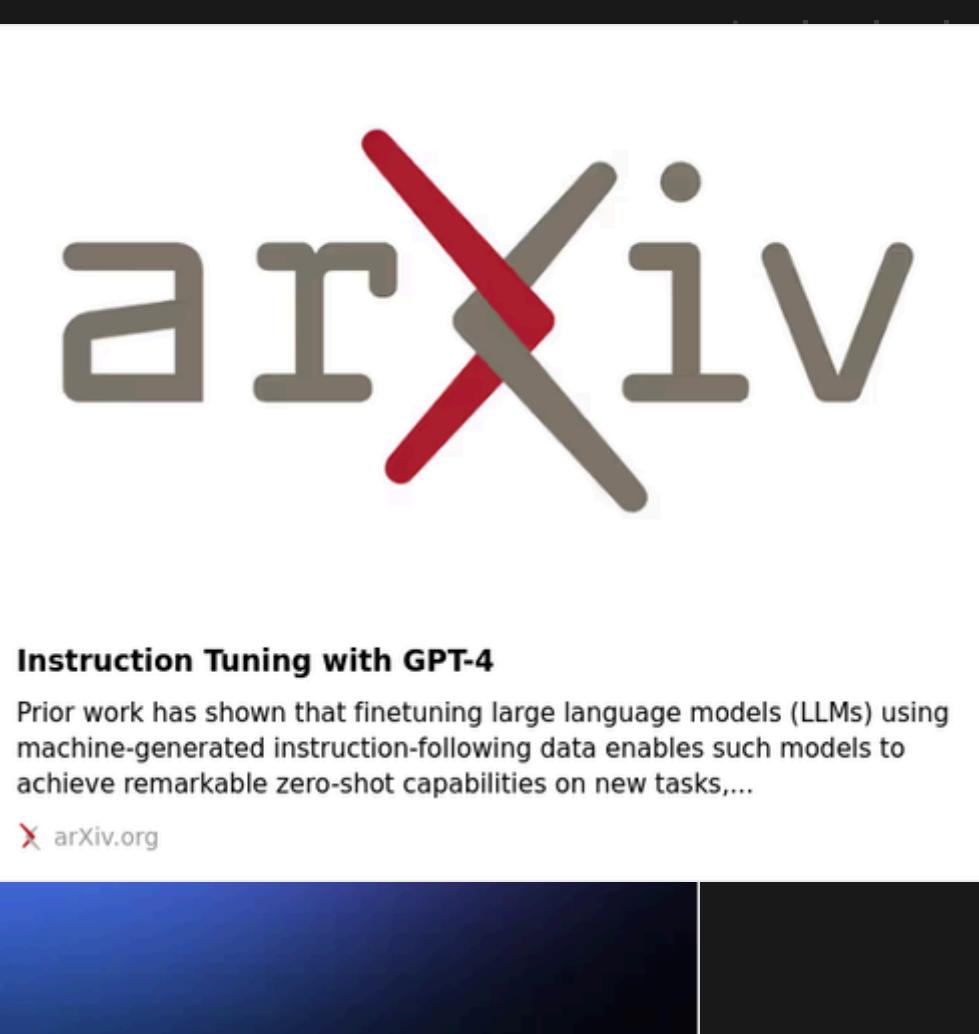
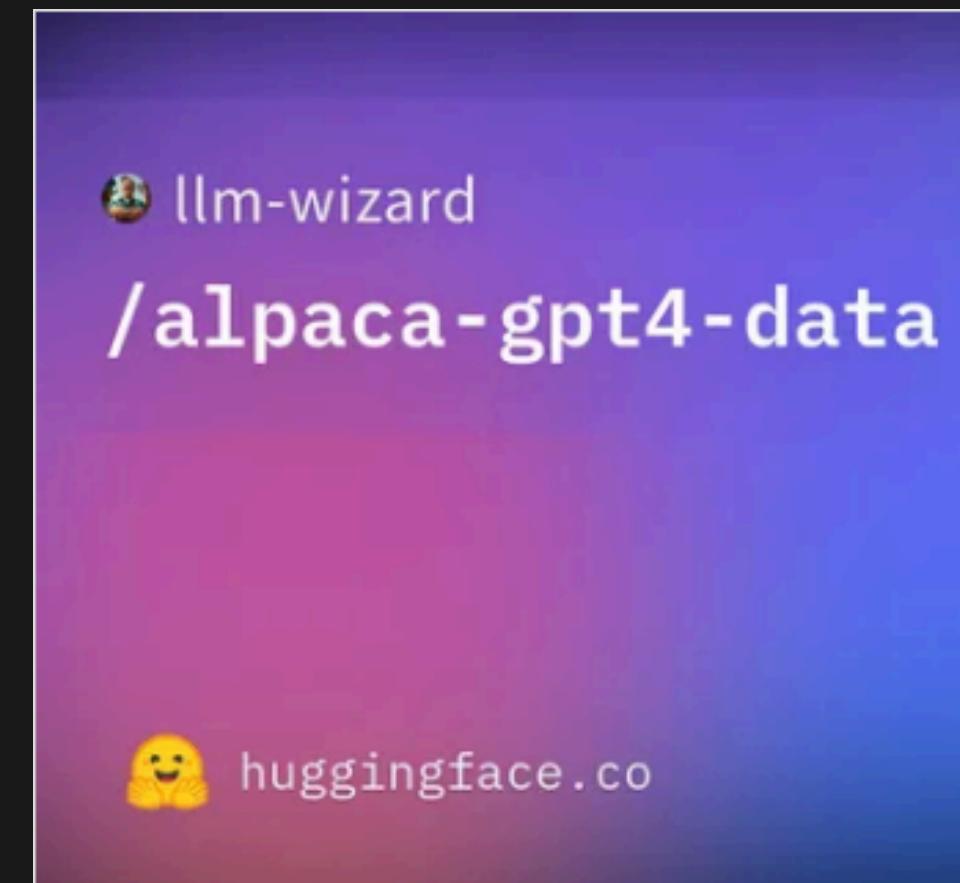
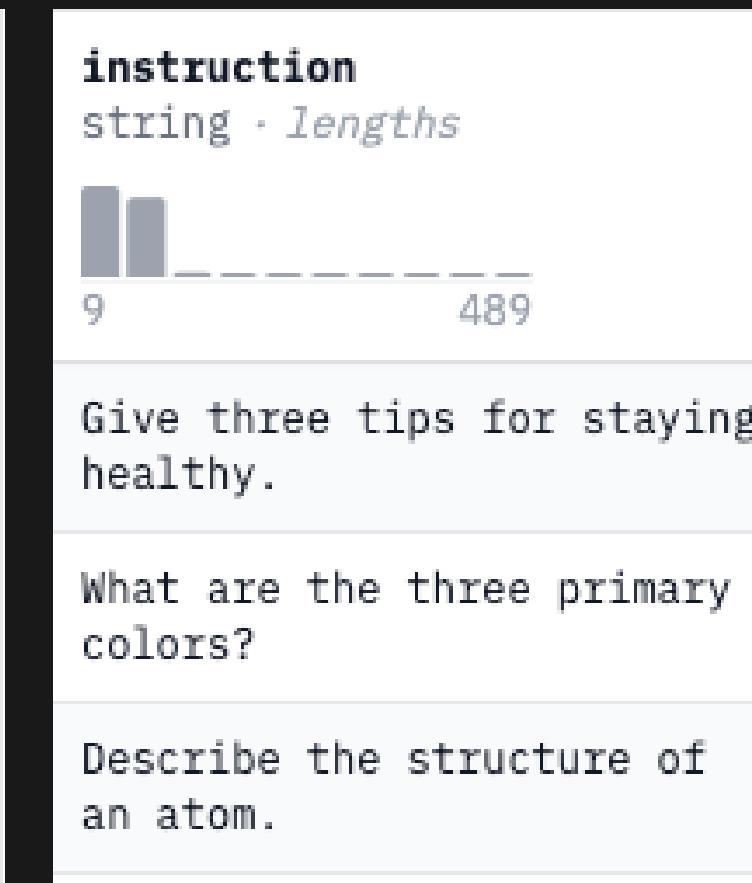
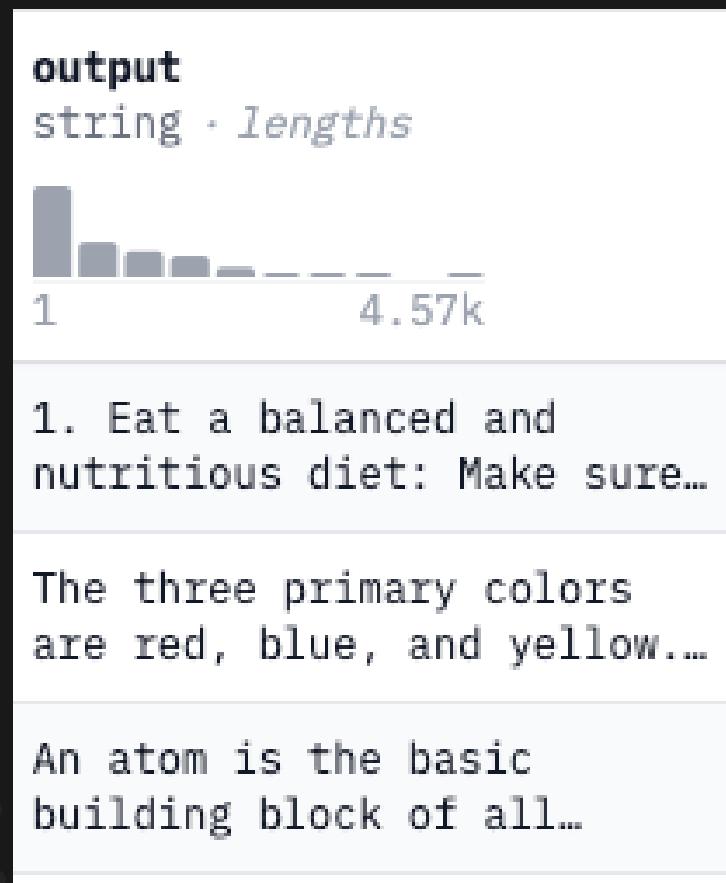
- Mistral 7B





# OUR FINE-TUNING DATA

# Fine-Tuning Dataset



**llm-wizard/alpaca-gpt4-data · Datasets at Hugging Face**

We're on a journey to advance and democratize artificial intelligence through open source and open science.

huggingface





# Fine-Tuning with PEFT- LoRA!

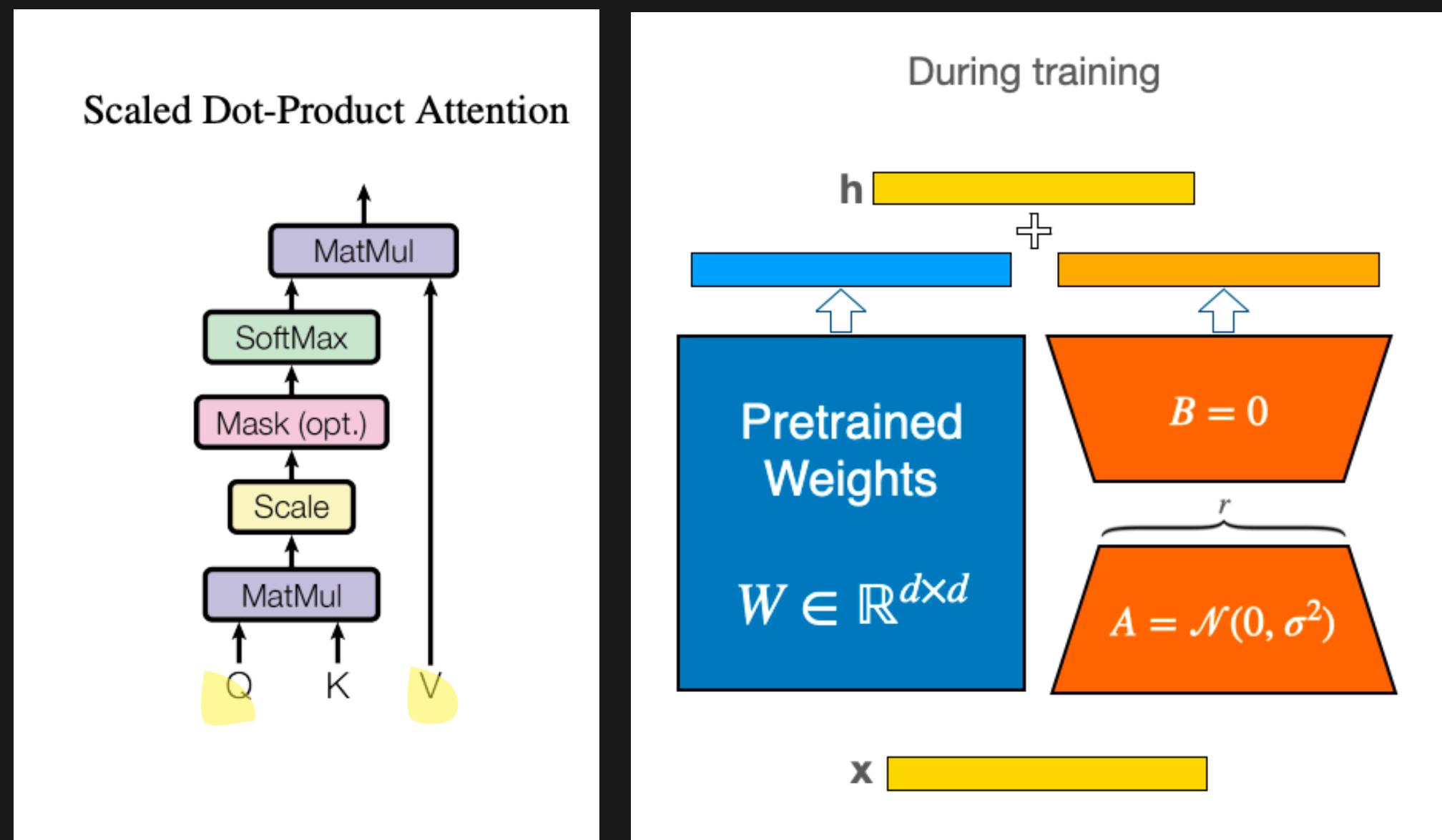
Presented by  
Chris Alexiuk, LLM Wizard ✨

# WHAT WE TRAINED

- All parameters = 3,8B
- Trainable parameters = 2,7M

**Attention layers =  $4096 \times 4096$**

- For  $Q = 4096 \times 4096$ 
  - **LoRA A** =  $4096 \times 64$
  - **LoRA B** =  $64 \times 4096$
- For  $V = 4096 \times 1024$ 
  - **LoRA A** =  $4096 \times 64$
  - **LoRA B** =  $64 \times 1024$

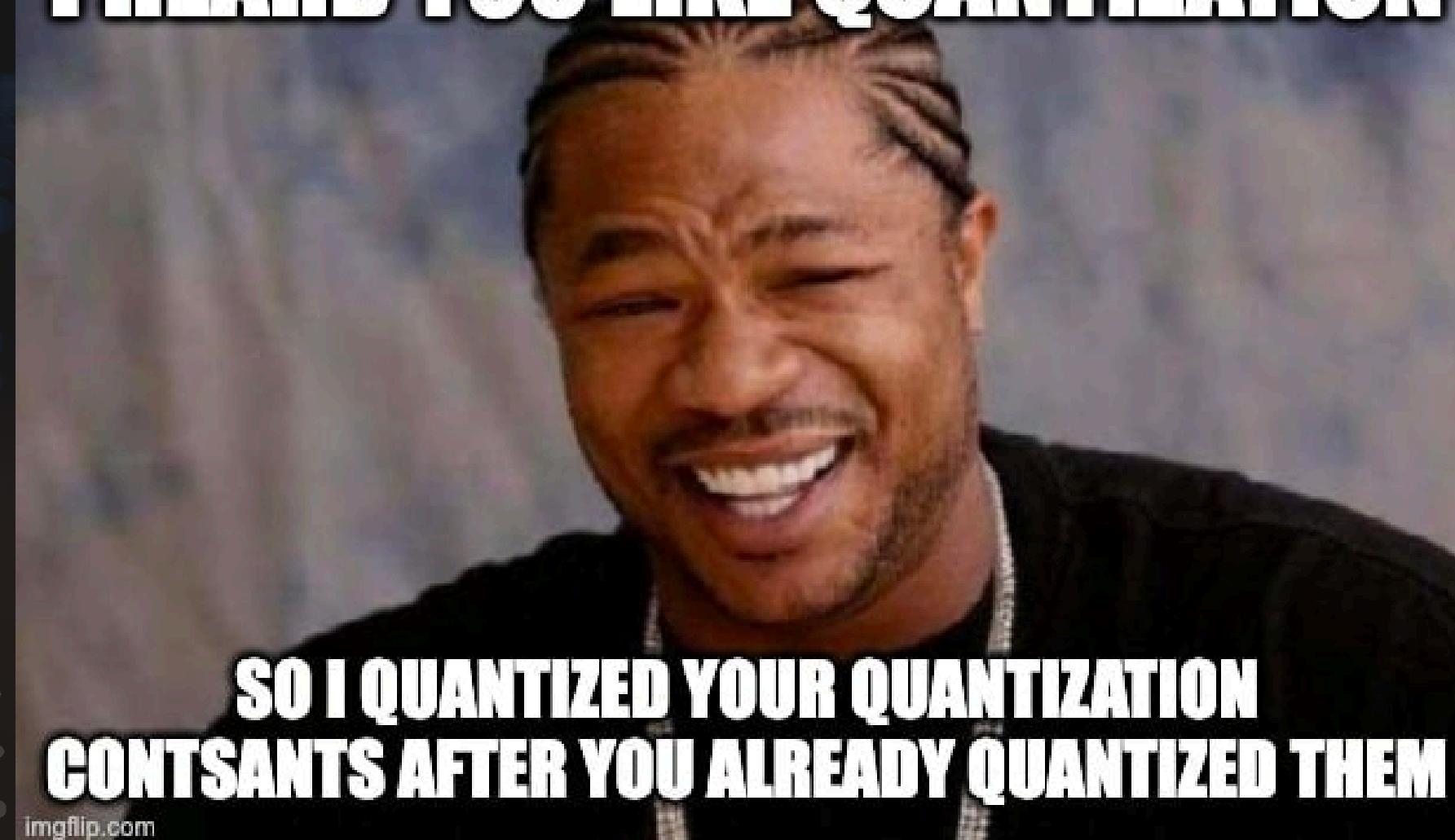


# CONCLUSIONS

-  PEFT-LoRA Fine-Tuning
  - *Modifying LLM behavior by updating less parameters with factorized matrices*
- Fine-Tuning is an important skill for 2024!
  - Helps us leverage the **essential dimensions** of our downstream task!
  - **Small Language Models** (SLMs) are only becoming more popular!
- Next stop ... Quantization and QLoRA!



I HEARD YOU LIKE QUANTIZATION



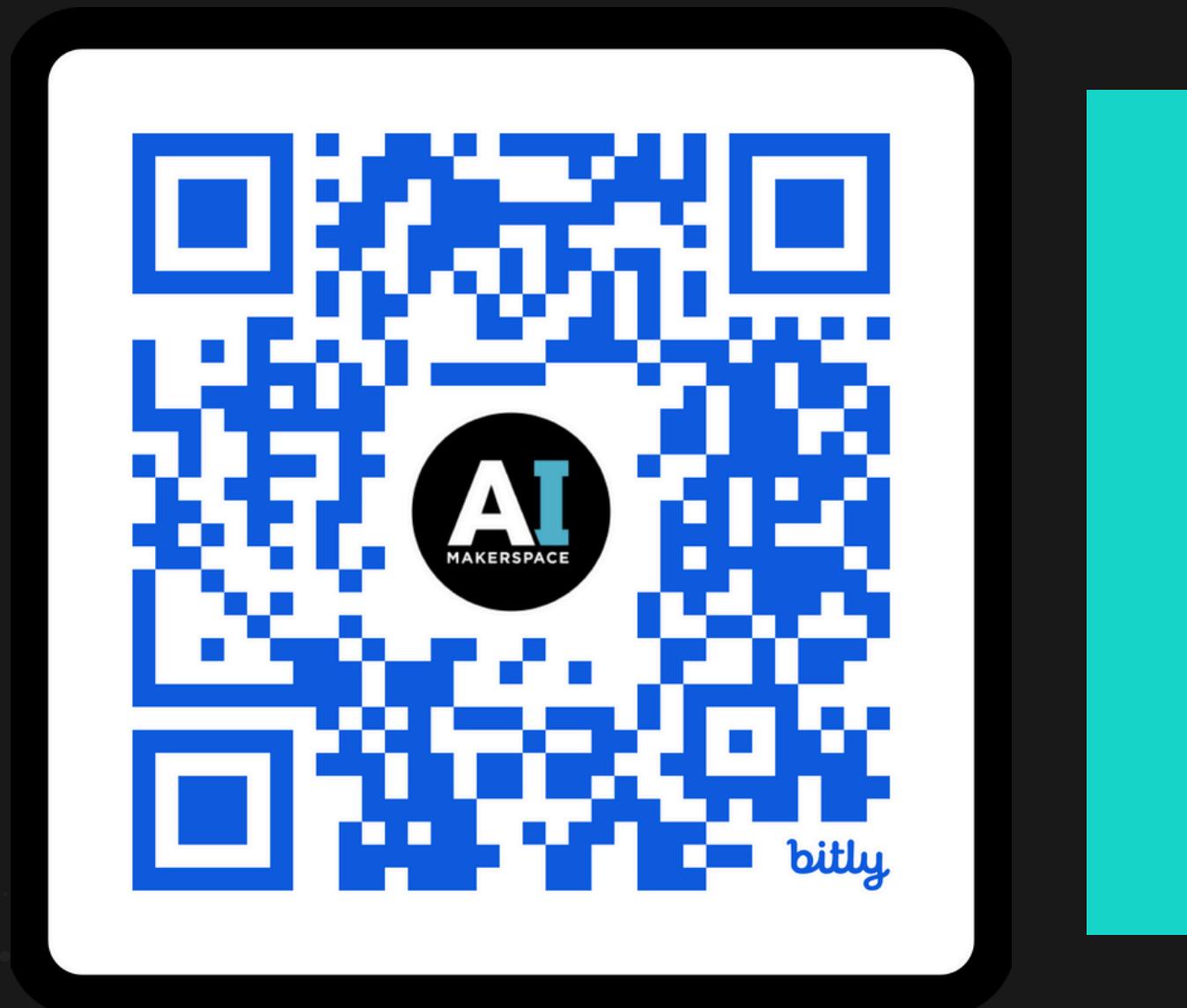
### QLoRA: Efficient Finetuning of Quantized LLMs

We present QLoRA, an efficient finetuning approach that reduces memory usage enough to finetune a 65B parameter model on a single 48GB GPU while preserving full 16-bit finetuning task performance....

X arXiv.org



# QUESTIONS?



Thank you!