ABB - Session 5

Fine-tuning

Shaw Talebi

ABB #4 - Spring 2025

Today's Session

ABB #4 - Spring 2025

Announcements

Guest Lecture Next Week!

Sangeeta Bahri

ABB2 Alumna

Founder @ EquityResearch

Powered by Advanced AI Technology

Smart Stock Analysis
Powered by AI

Make data-driven investment decisions. It's like having a Wall
Street analyst on speed dial - minus the attitude & expensive suit.

ABB #4 - Spring 2025

Homework
Shoutouts 🎉

(Real Estate) Blog to LI Post

Joe Frausto (HW2)

Lead Scoring App

Joe Frausto (HW3)

2 Levels of LLM Development

How to get LLMs to do what you want…

Level 1

Adapting models via prompts and tools

Prompt Engineering 💬

RAG 🔍

Tool-use 🛠️

Level 2

Adapting models via additional training

Fine-tuning 🪛

Post-training 👨‍💻

ABB #4 - Spring 2025

Fine-tuning

ABB #4 - Spring 2025

What is Fine-tuning?

Adapting a model to a particular task through **additional training**

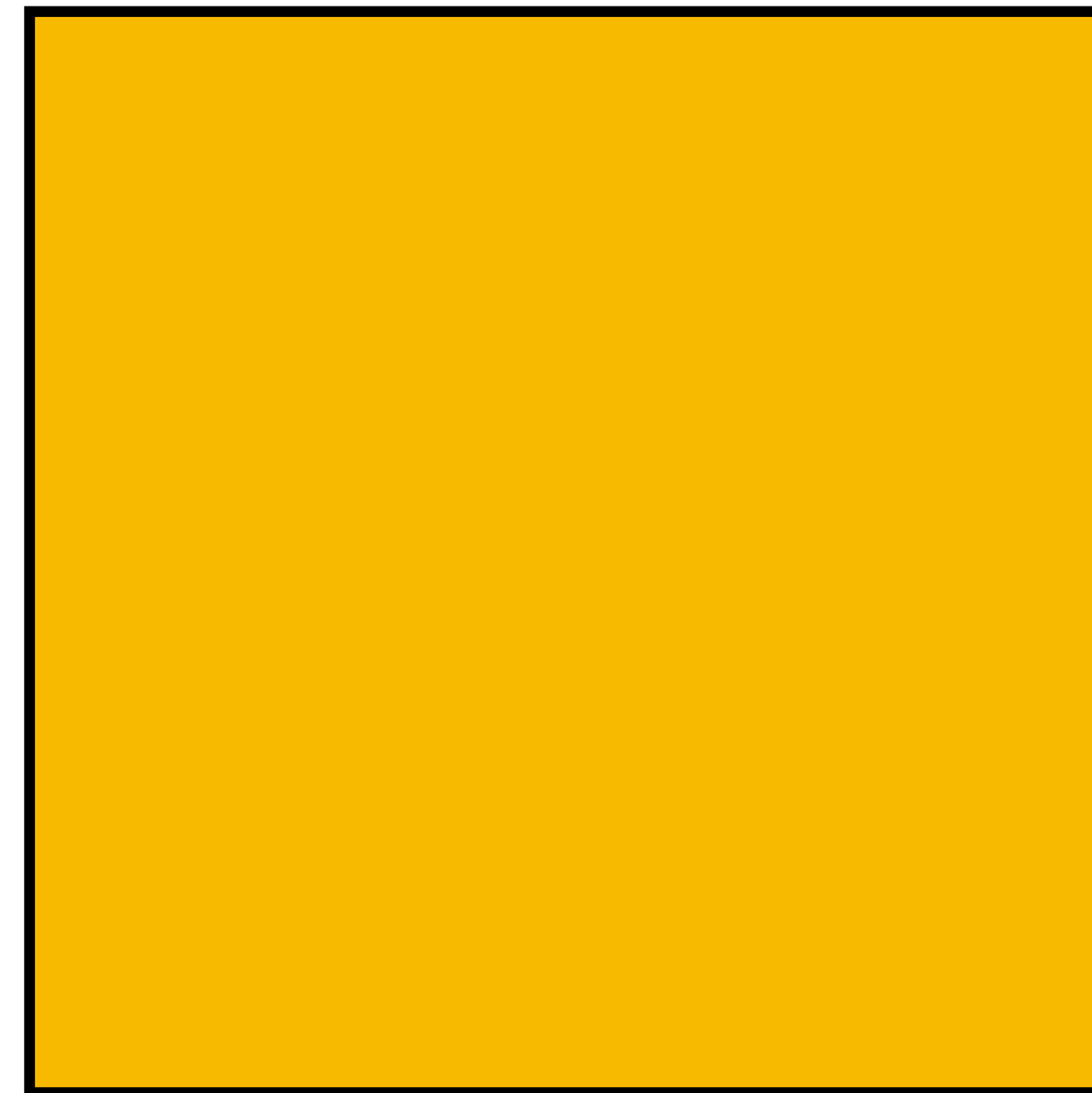| **Pre-trained Model**	**Fine-tuned Model**	**More Fine-tuning**
(Self-supervised)	(Supervised)	(RLHF or DPO)
Internet Document Completer	Instruction Follower	(Aligned) Instruction Follower
~10T Tokens	*~10B Tokens*	*~10B Tokens*

ABB #4 - Spring 2025

Why Fine-tune?

1) Smaller fine-tuned models outperform bigger ones (on specific tasks)

InstructGPT (1.3B) **>** **GPT-3 (175B)**

ABB #4 - Spring 2025

Why Fine-tune?
1) Smaller fine-tuned models outperform bigger ones (on specific tasks)

GPT-3

Tell me how to fine-tune a model.

Response:

How can I control the complexity of a model?
How do I know when my model is done?
How do I test a model?
How do I define a task for my model?
How do I scale a model?
How do I make a probabilistic model?
How do I start using a model once it's complete?
How do I cope with missing data?
How do I make my modeling efforts sustainable?

InstructGPT

Tell me how to fine-tune a model.

Response:

Fine-tuning a model involves adjusting the parameters of a pre-trained model in order to make it better suited for a given task. There are generally three steps to fine-tuning a model:

1. Select a base model - Select a pre-trained deep learning model that has been trained on a large dataset.
2. Adjust parameters - Adjust parameters of the pre-trained model to better suit the desired task. This may include changing the number of layers, adjusting learning rate, adding regularization, or tweaking the optimizer.
3. Train the model - Train the new model on the desired dataset. The amount of data and the amount of training required will depend on the task and the model.

[3][4]   ABB #4 - Spring 2025

Why Fine-tune?
2) Lower compute costs and faster inference (from shorter prompts)

Prompt before fine-tuning

```
You are an expert sentiment analysis tool. Given a text, classify its
sentiment as "Positive," "Negative," or "Neutral." Here are some
examples:

Examples
Text: "I absolutely love the new design of this product! It's user-
friendly
and looks amazing." Sentiment: Positive

Text: "The service was terrible. I had to wait for over an hour, and
the staff
was rude." Sentiment: Negative

Text: "The movie was okay. It wasn't particularly exciting, but it
wasn't bad
either." Sentiment: Neutral

Text: "I'm so happy with the customer support I received. They
resolved my
issue quickly." Sentiment: Positive

Text: "The meal was bland, and I wouldn't recommend it to anyone."
Sentiment:
Negative

Text: "The weather is unpredictable today." Sentiment: Neutral

Now analyze the following text:
Text: "[Your input text here]" Sentiment:
```

Prompt after fine-tuning

```
You are an expert sentiment analysis tool. Given a text, classify its
sentiment as "Positive," "Negative," or "Neutral."

Analyze the following text:
Text: "[Your input text here]" Sentiment:
```

Time & money saved!

# When Should I Fine-tune?

No ❌

1. Before improving your prompt (systematically via evals)

2. Trying to teach model new knowledge (RAG is better)

3. Before gathering input-output pairs (training data)

Yes ✅

1. Improving reliability of desired output structure

2. Customizing response style or format

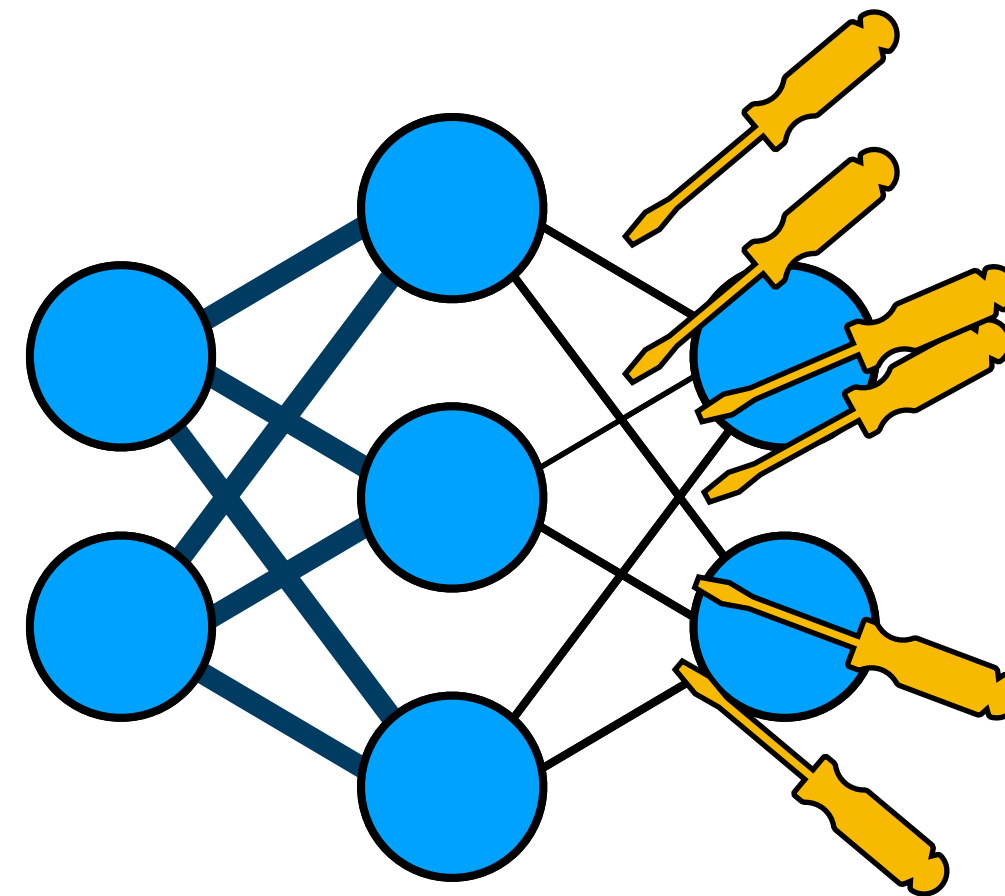3. Performing tasks that are hard to articulate via prompts

ABB #4 - Spring 2025

How Do I Fine-tune?
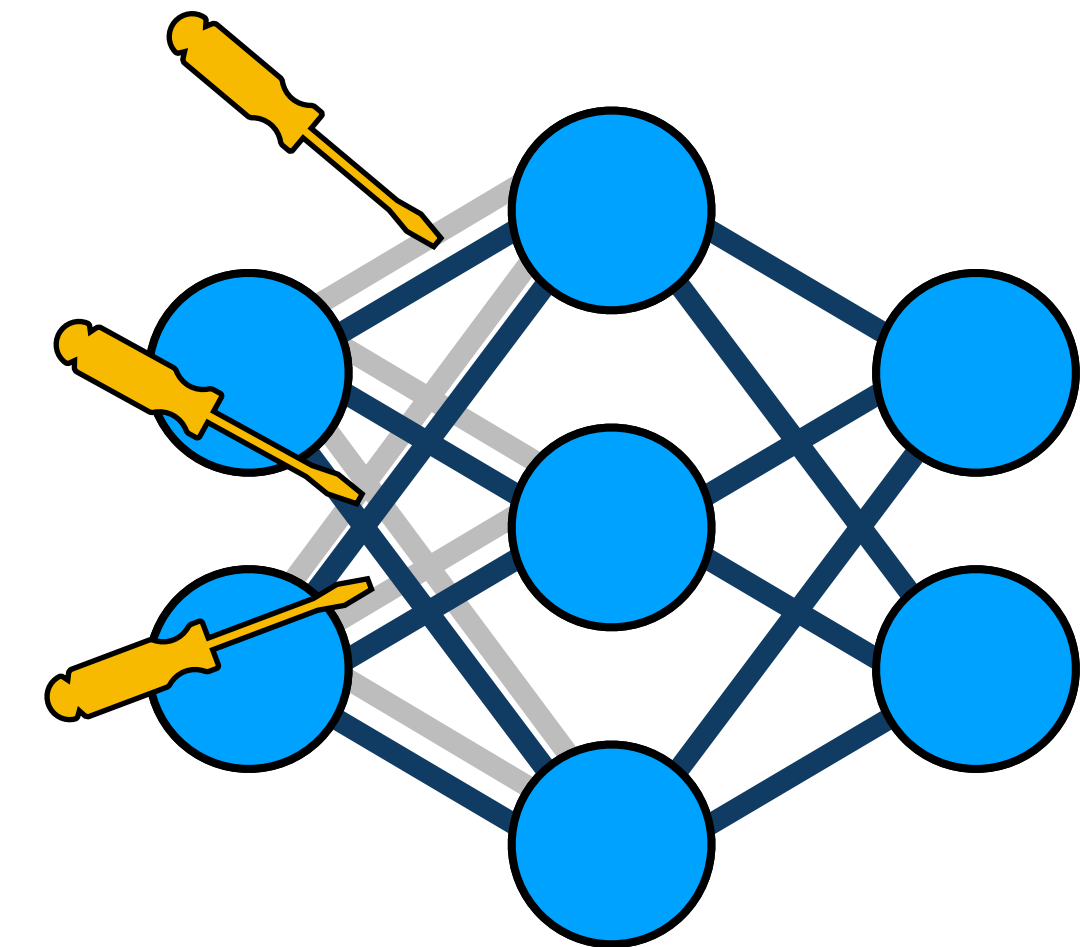## 3 Approaches
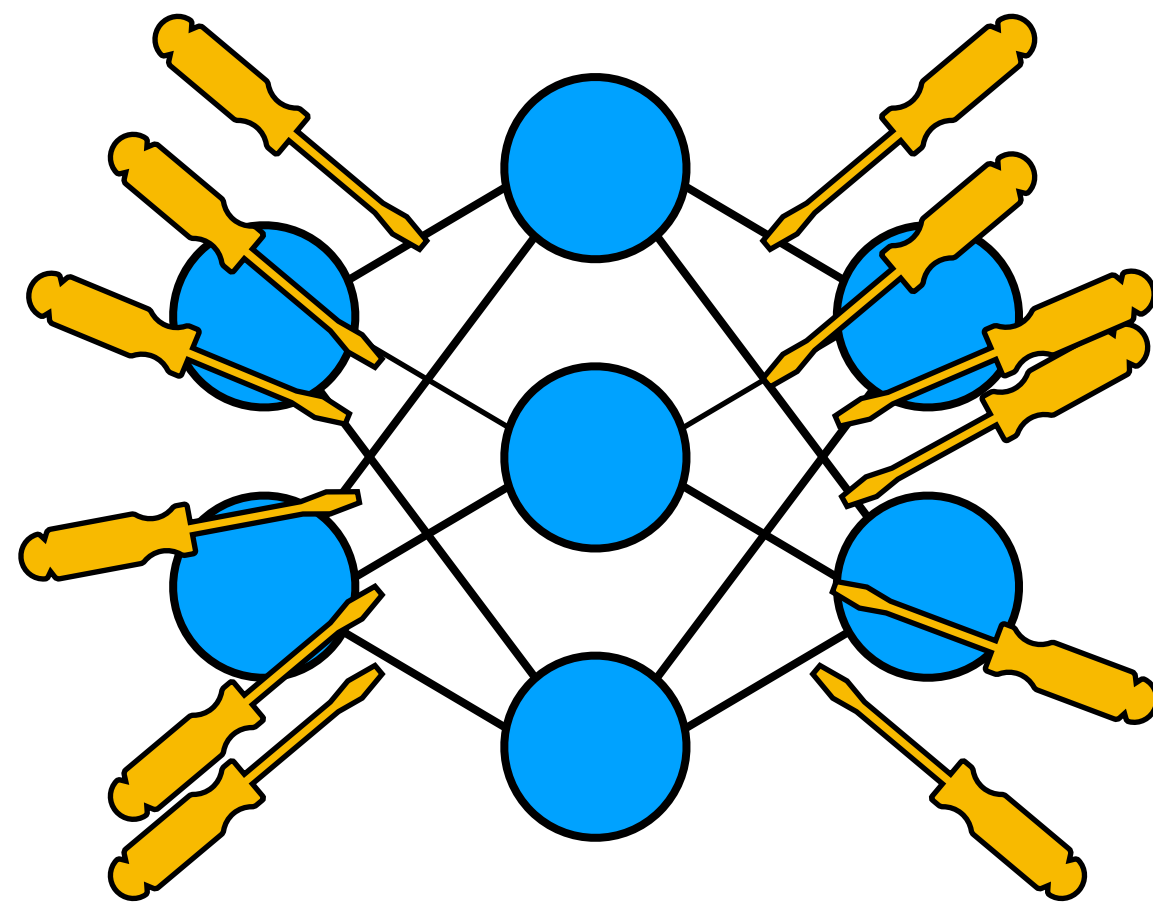
1) All parameters

aka "Full Fine-tuning"

2) Some parameters

aka "Transfer Learning"

3) New parameters

aka Adapters

ABB #4 - Spring 2025

How Do I Fine-tune?

Training Data: Input-output pairs

Example: Fine-tuning Response Style

Inputs: Questions/DMs

> Hi , I have a question, does this woks on ECG signal that are "changing"?

Outputs: Real responses

> That's a good question! The short answer is yes. Ultimately it comes down to what you are after. If it is to find r peaks, like we did here, it shouldn't be an issue. Hope that helps!

Training Dataset

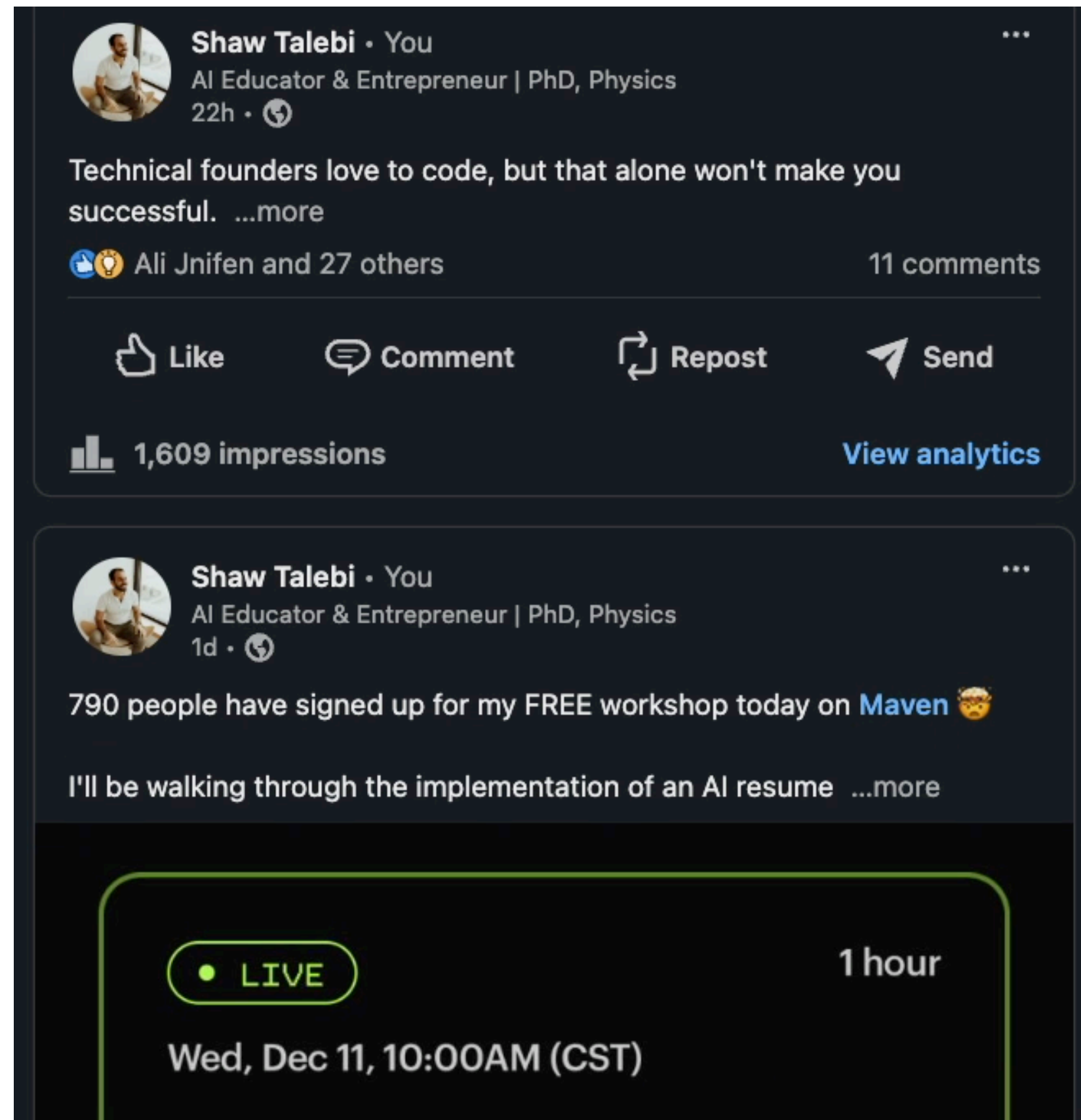| Question	Response

Prompt Template

[INST] ShawGPT, functioning as a virtual data science consultant... (instructions cont.)
{Question} [/INST]

{Response}

Examples

ABB #4 - Spring 2025

Example 1
Fine-tuning a LinkedIn Post Writer (Motivation)

Major marketing channel for my content and offers

... but I do it myself which limits high volume and high quality content

ABB #4 - Spring 2025

Example 1
Fine-tuning a LinkedIn Post Writer (Motivation)

Can't you use ChatGPT?

Doesn't sound like me…

… and lacks taste

> **Python felt impossible… until I learned these 5 things.**
>
> When I first started with Python, it wasn't the syntax that tripped me up—it was *how to think like a programmer*. I'd get stuck in tutorials, wondering how people were able to "just know" what to write.
>
> Turns out, learning Python isn't just about *what* you write—it's about *how* you approach problems. Here are the 5 mindset shifts that changed everything:
>
> **1 Break problems into smaller steps**
> Stop staring at the big picture. Solve one small piece at a time. (If you don't know where to start, just print something—anything!)
>
> **2 Google like it's your superpower**
> The best Python developers aren't human encyclopedias. They're *really good* at searching for solutions and reading Stack Overflow.
>
> **3 Code >>> Theory**
> Learn by doing. Build tiny projects—even silly ones. You'll remember more from 10 lines of messy code than 2 hours of reading.
>
> **4 Start simple. THEN optimize.**
> Write code that *works first*. Worry about making it "pretty" or efficient later. (Perfection is the enemy of progress.)
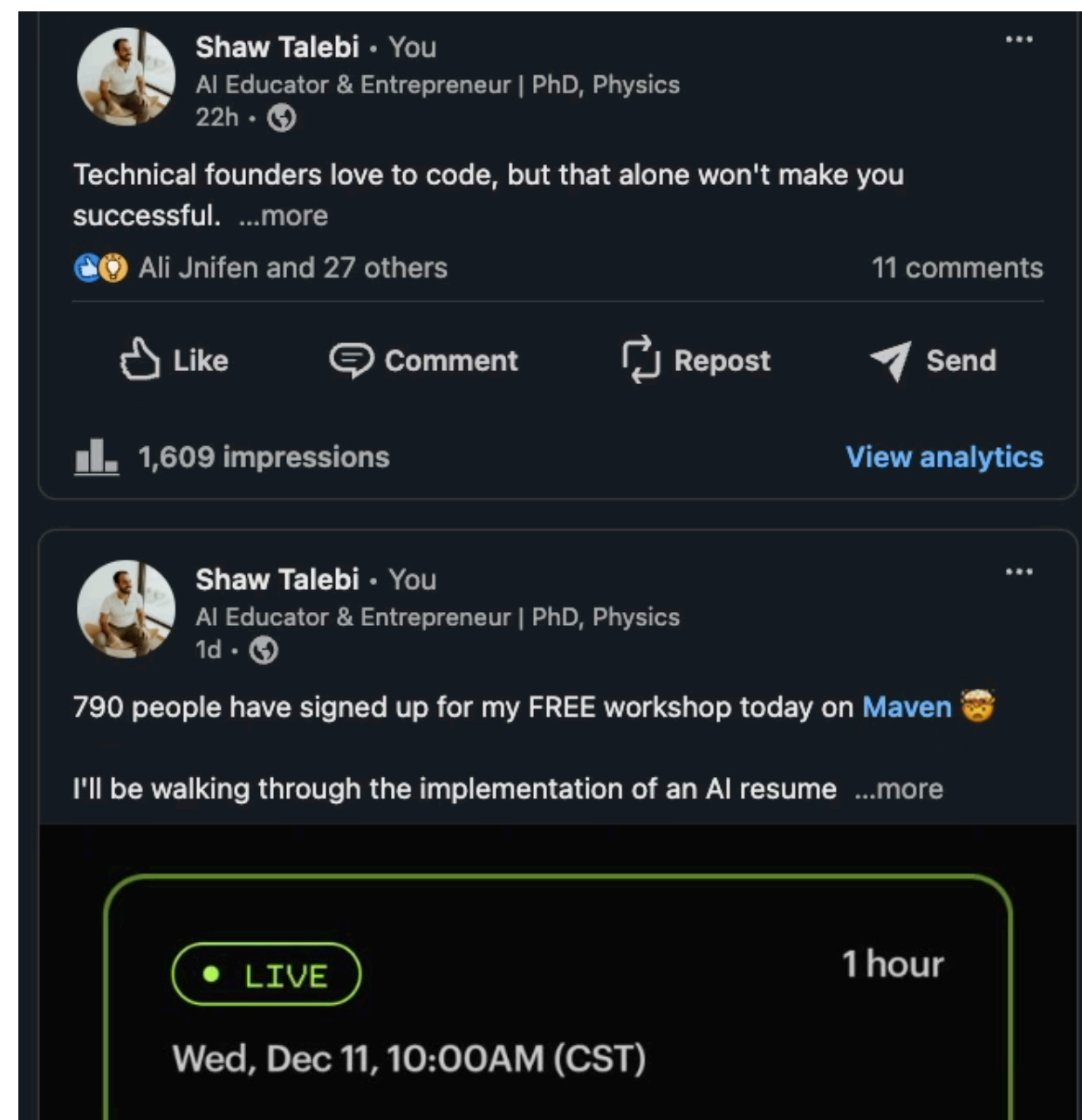>
> **5 Copy and tweak**
> Find code examples and *tinker*. Change a variable. Add a print statement. See what breaks. Play like this until it all clicks.

ABB #4 - Spring 2025

Example 1
Fine-tuning a LinkedIn Post Writer (Overview)

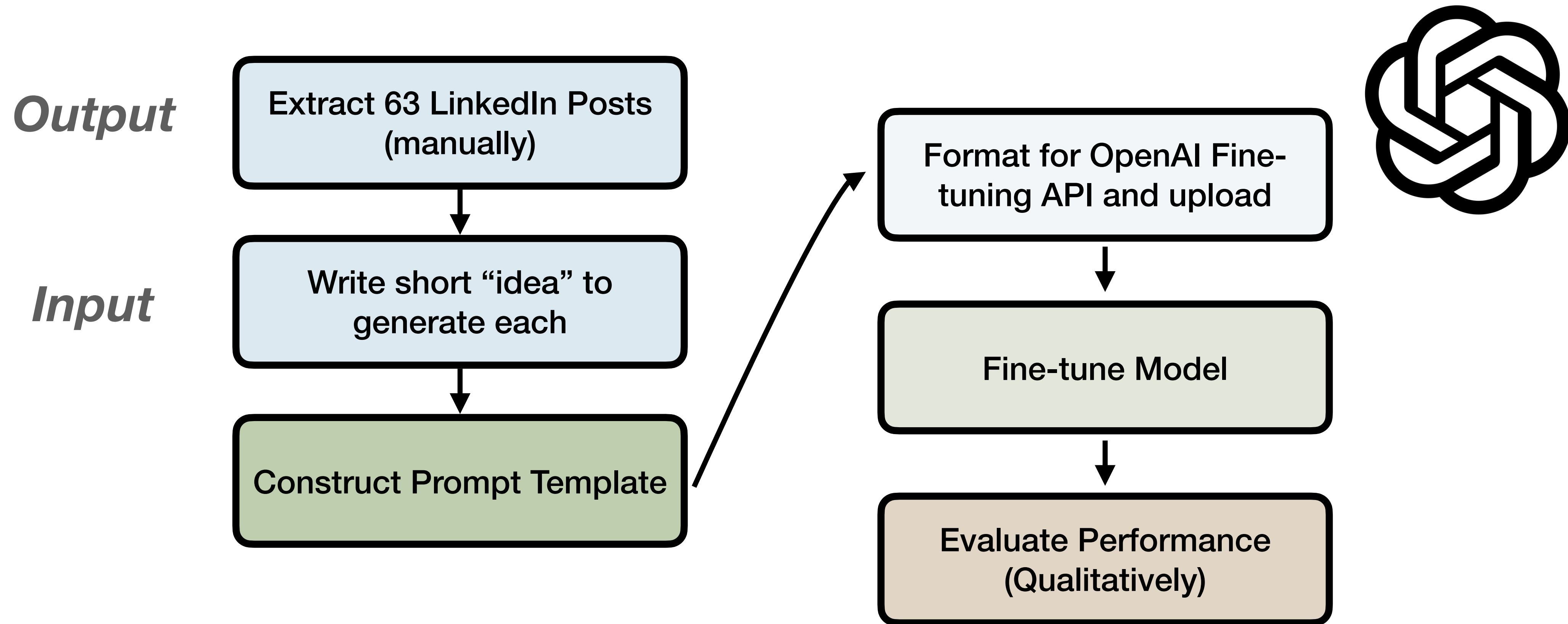Extract LI posts   Fine-tune GPT-4o-mini   LI Post Writer
                                                                    (in my likeness)

ABB #4 - Spring 2025

Example 1

Fine-tuning a LinkedIn Post Writer (Flowchart)

Output

Extract 63 LinkedIn Posts (manually)

Input

Write short "idea" to generate each

Construct Prompt Template

Format for OpenAI Fine-tuning API and upload

Fine-tune Model

Evaluate Performance (Qualitatively)

ABB #4 - Spring 2025

Example 1
Fine-tuning a LinkedIn Post Writer (Code)

ABB #4 - Spring 2025

Example 2

Fine-tuning BERT for Text Classification (Motivation)

Subject: Congratulations on Your New Offer! Special Opportunity Just for You

From: Service Center <rewards@service-center.com>

Hi there!

We're excited to let you know that you've been selected for an exclusive deal. This offer is designed to match your preferences, and it's available for a limited time.

Simply follow our instructions to learn more about how you can take advantage of this unique opportunity.

To proceed, you only need to confirm your email by visiting the link below:

Confirm Email

paypal.com.cgi-bin.login1589114111g859f.sonreir.cl/paypal/2/

For more details, please check the terms on our website or feel free to reach out with any questions.

We look forward to hearing from you soon!

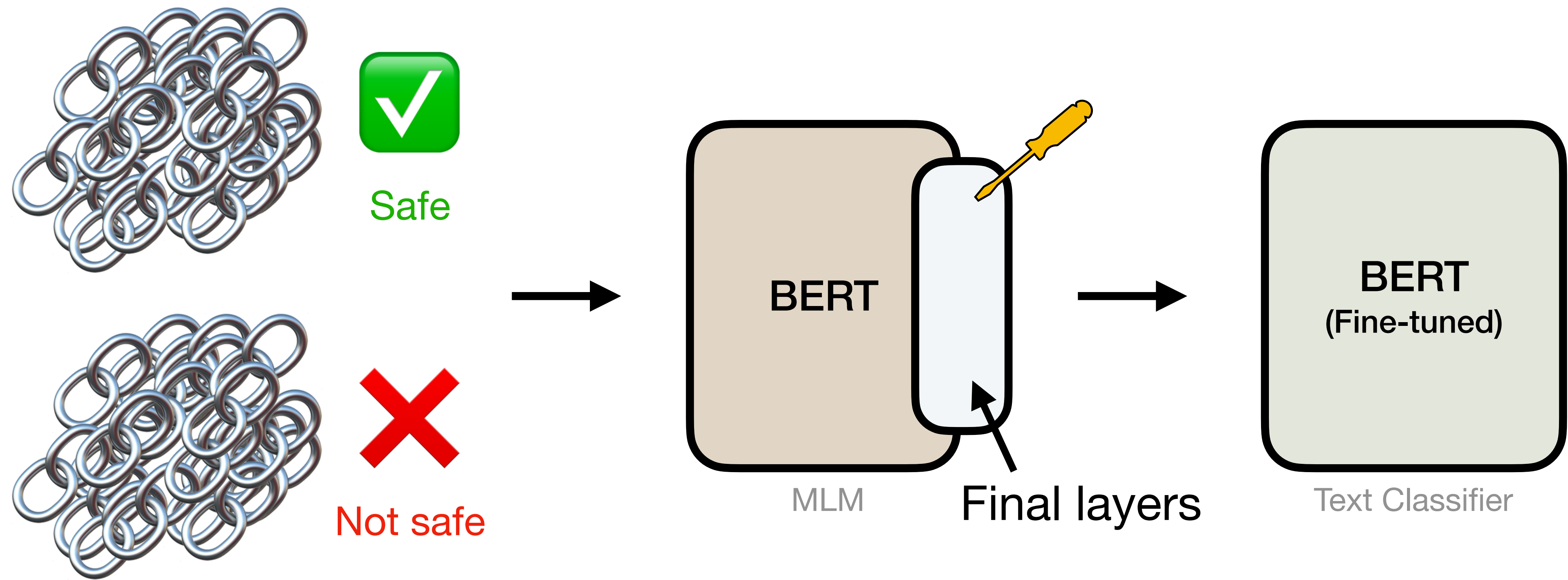Best Regards,
The Service Center Team
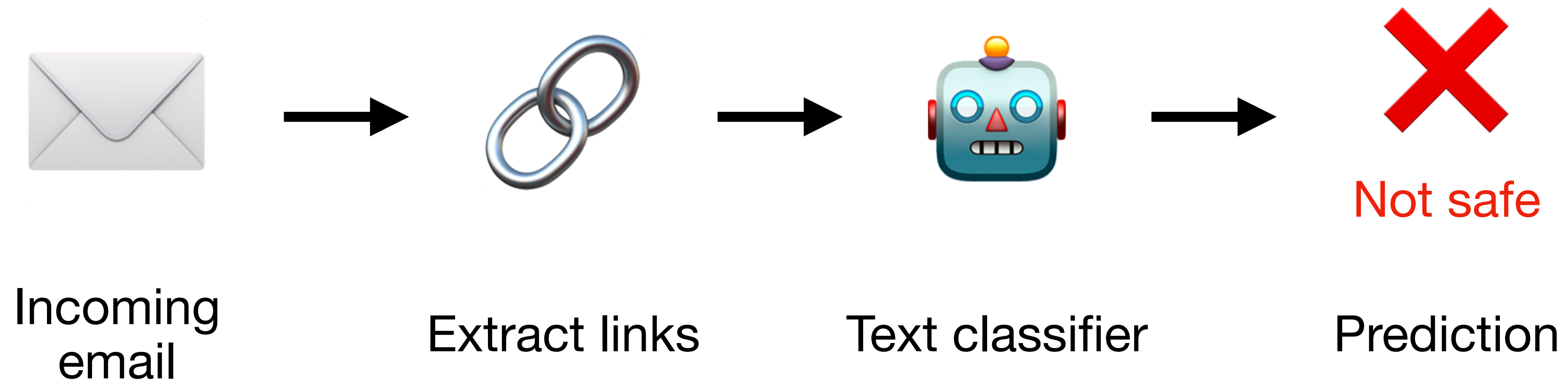
ABB #4 - Spring 2025

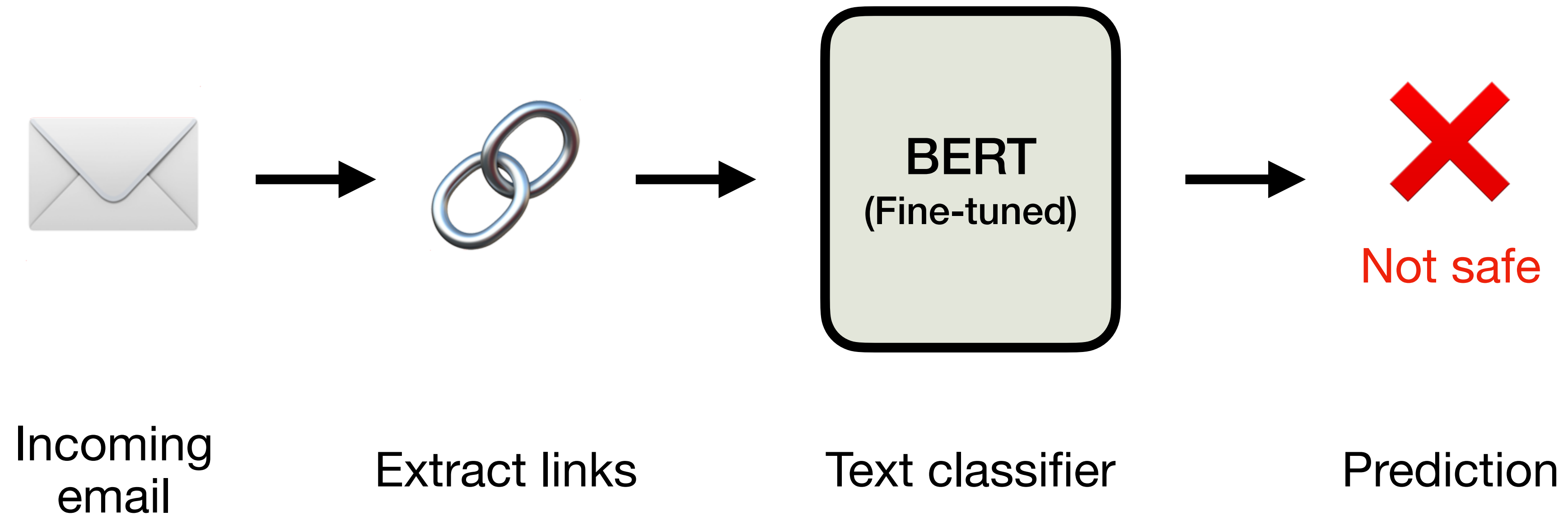# Example 2
Fine-tuning BERT for Text Classification (Overview)

| Incoming email | Extract links | Text classifier | Prediction |

Not safe

ABB #4 - Spring 2025

Example 2
Fine-tuning BERT for Text Classification (Overview)

✅ Safe

❌ Not safe

BERT

MLM

Final layers

BERT (Fine-tuned)

Text Classifier

ABB #4 - Spring 2025

Example 2
Fine-tuning BERT for Text Classification (Overview)

Incoming email → Extract links → Text classifier (BERT (Fine-tuned)) → Prediction: Not safe

ABB #4 - Spring 2025

Example 2
Fine-tuning BERT for Text Classification (Flowchart)

Load Phishing URL Data

Import Model

Freeze Parameters

Preprocess text

Define Evals

Fine-tune Model

Eval Model on Validation Dataset

ABB #4 - Spring 2025
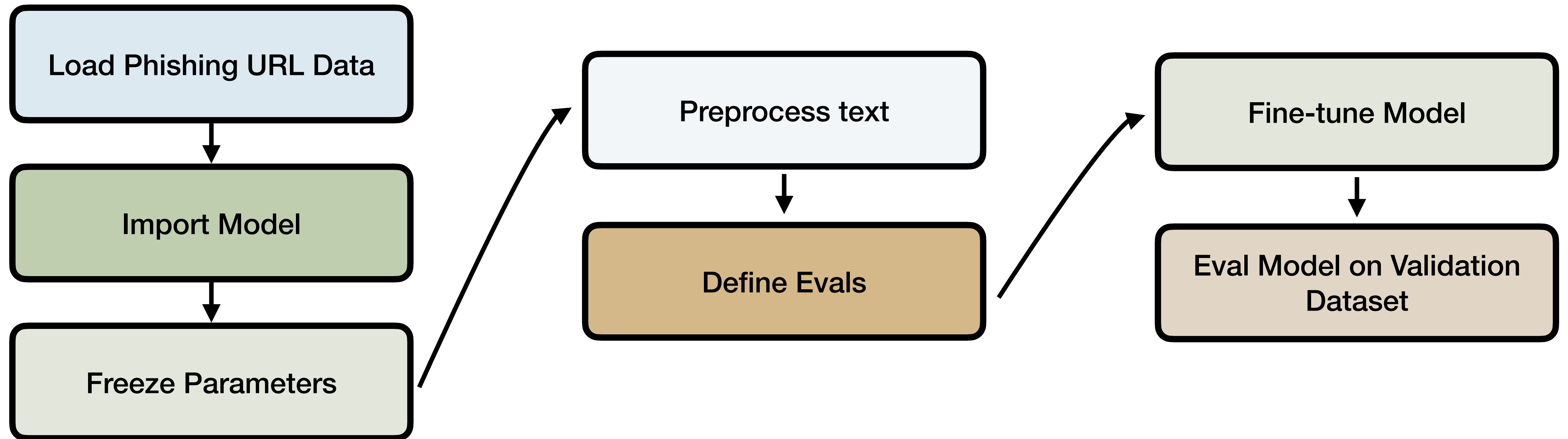
Example 2
Fine-tuning BERT for Text Classification (Code)

ABB #4 - Spring 2025

More on Fine-tuning/Post-training

1. Efficient Fine-tuning with QLoRA

2. Fine-tuning Text Embeddings

3. Fine-tuning Multimodal Embeddings

4. Local Fine-tuning on Mac

5. Fine-tuning FLUX.1

6. Distilling LLMs

7. RLHF + DPO (Preference Tuning)

ABB #4 - Spring 2025

Homework 5

Project 💻

Fine-tune a Model **(More examples provided)**

Pre-work ✍️

Session 6: AI Project Management

Session 6: Project Discovery Questions

ABB #4 - Spring 2025

⭐ = more FT examples

References

[1] Fine-Tuning BERT for Text Classification ⭐

[2] arXiv:2407.21783 [cs.AI]

[3] arXiv:2203.02155 [cs.CL]

[4] Fine-tuning Large Language Models (LLMs)

[5] OpenAI Fine-tuning Documentation

[6] arXiv:2312.05934 [cs.AI]

[7] QLoRA—How to Fine-tune an LLM on a Single GPU ⭐

[8] arXiv:2106.09685 [cs.CL]

[9] Compressing Large Language Models ⭐

ABB #4 - Spring 2025

LoRA (Low-Rank Adaptation)

Fine-tunes model by adding **small set** of trainable parameters
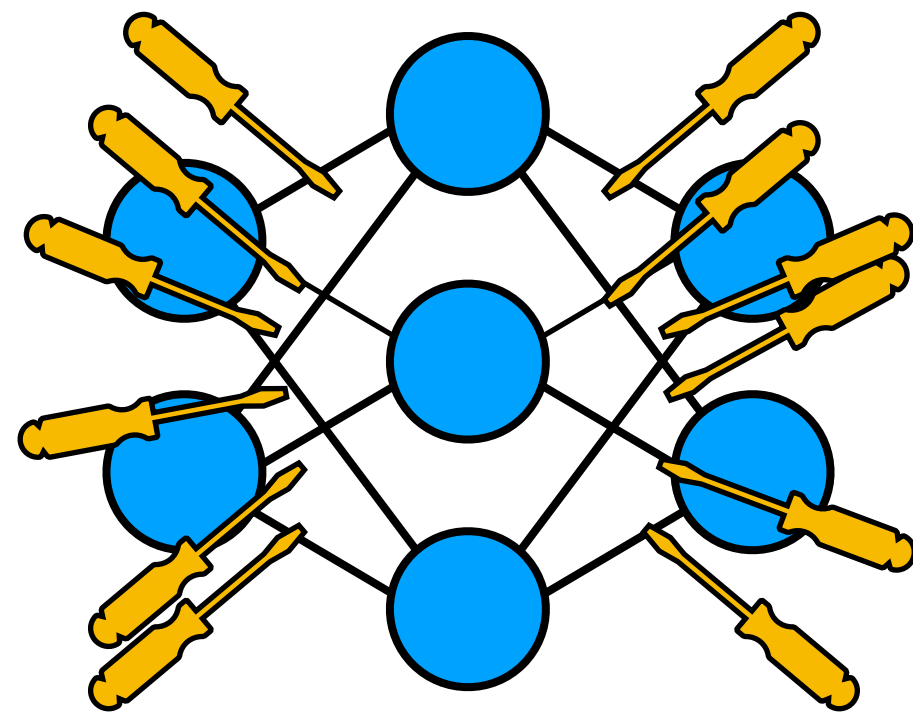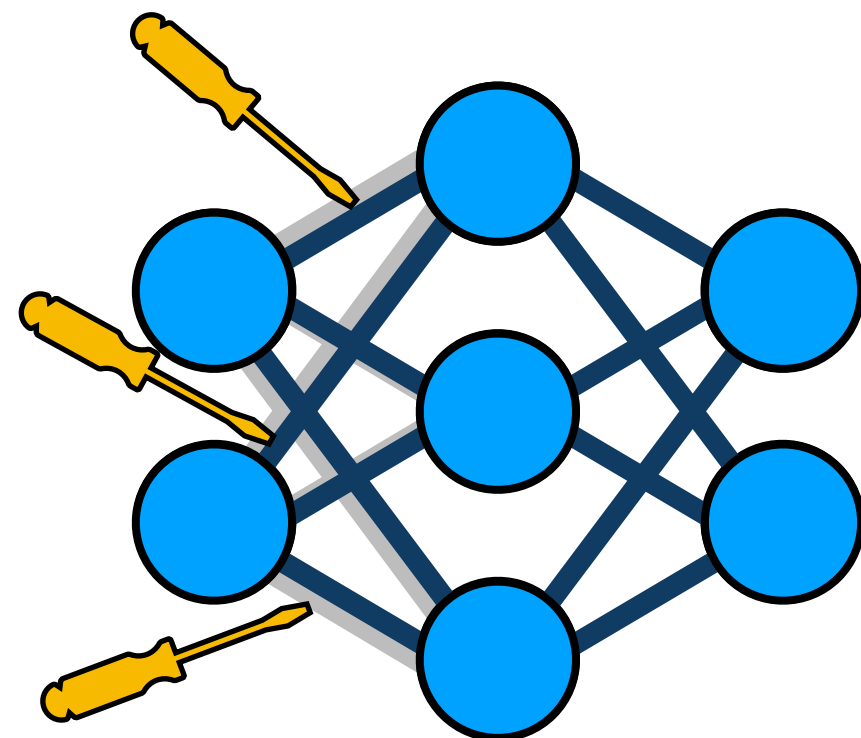
Full Fine-tuning: $h(x) = W_0 x$

$$W_0 \quad x \quad = \quad h(x)$$

Trainable

$x \quad h(x) \quad y$

LoRA: $h(x) = W_0 x + \Delta W x = W_0 x + BAx$

$$\left(W_0 + \boxed{B \quad A} \right) x = h(x)$$

Frozen   **Trainable**

100-1000X savings!

ABB #4 - Spring 2025

QLoRA (Quantized LoRA)
Combining LoRA and Quantization

Full Finetuning

LoRA

QLoRA

CPU

Optimizer State (32 bit)

Adapters (16 bit)

Base Model

Parameter Updates

Gradient Flow

Paging Flow

FP16

FP16

4-bit NormalFloat

10B $\Longrightarrow$ 160GB

10B $\Longrightarrow$ ~40GB

10B $\Longrightarrow$ ~12GB

ABB #4 - Spring 2025
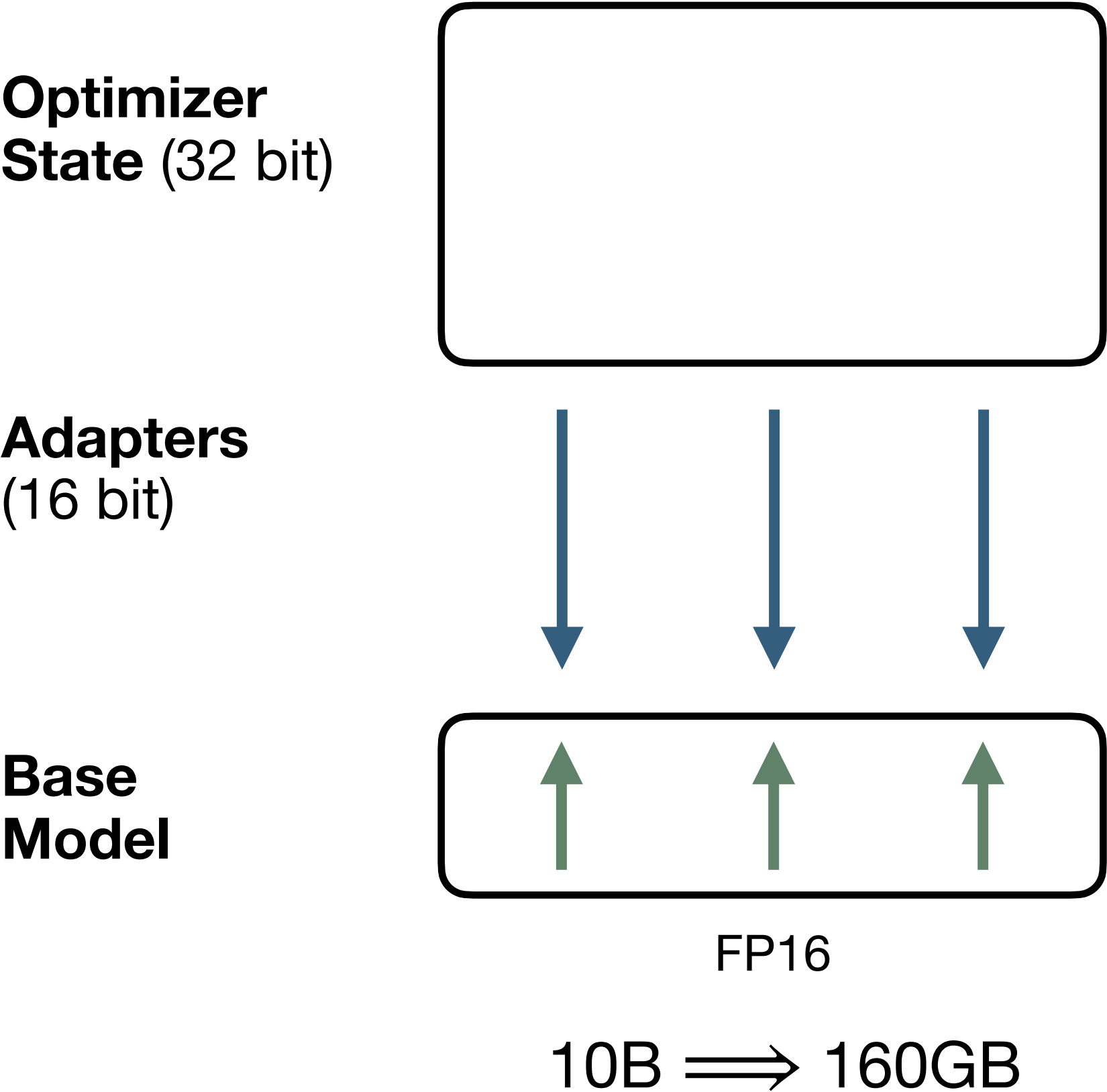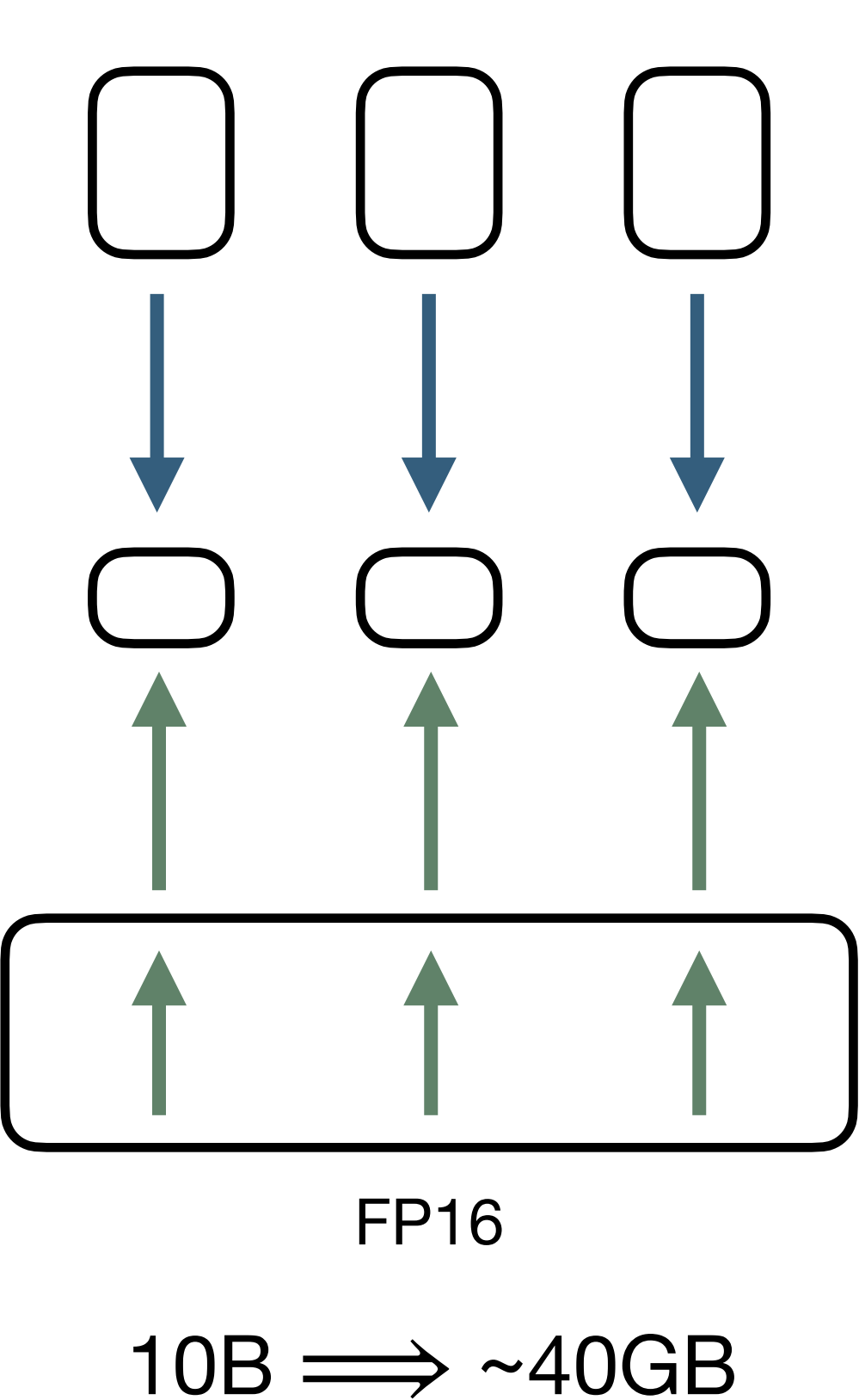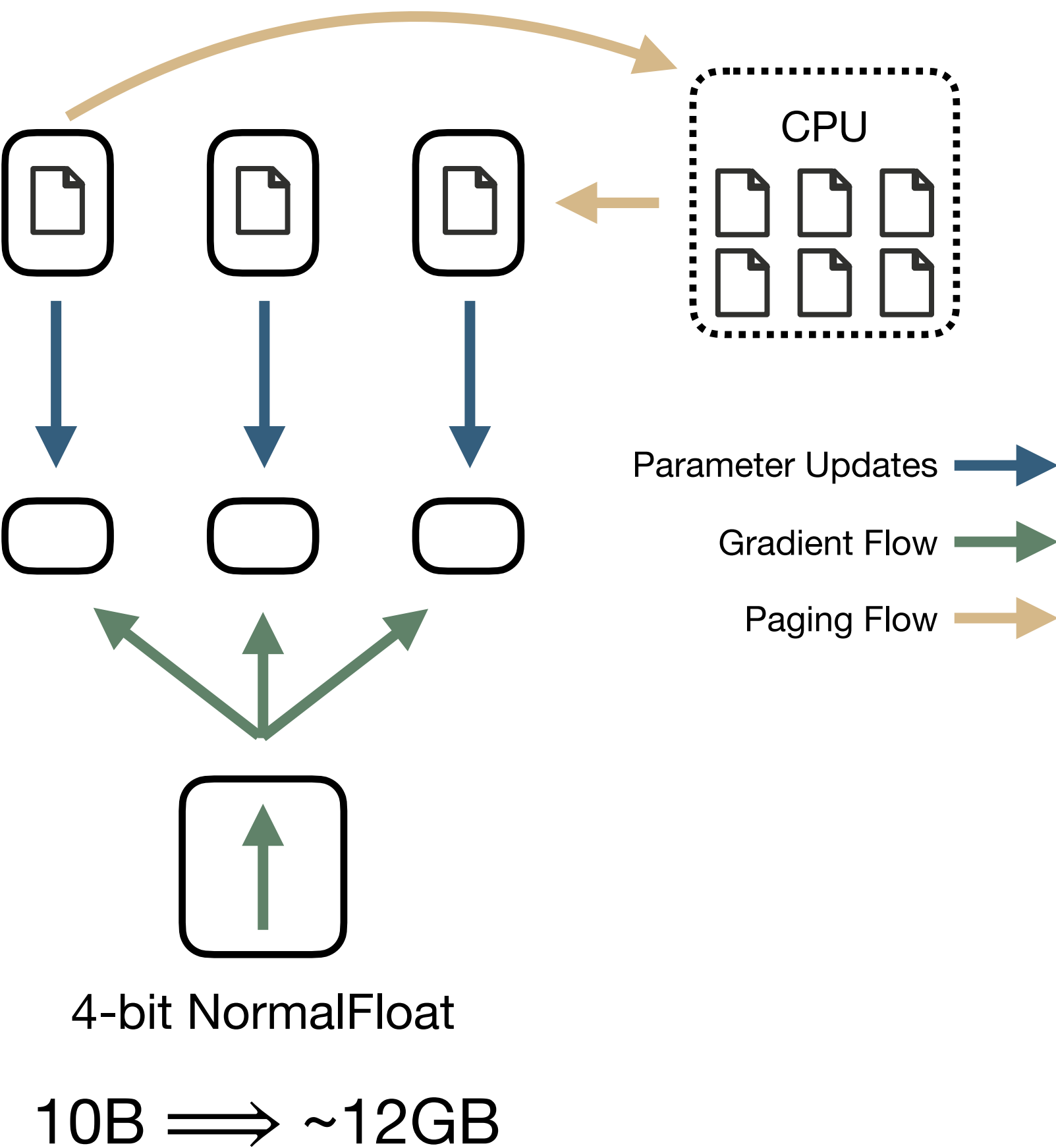
Model Compression

Reduce ML model size without sacrificing performance

1) Quantization
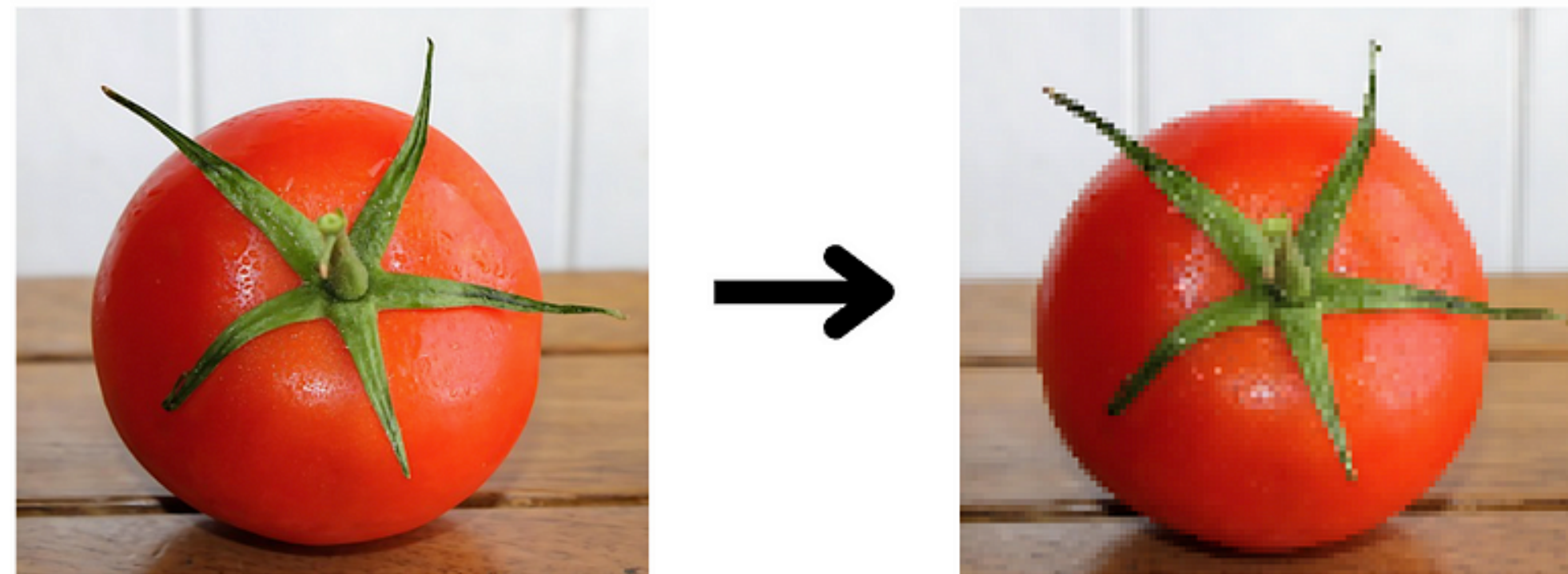
QLoRA = combines this and LoRA enabling LLMs to be fine-tuned on a single GPU!

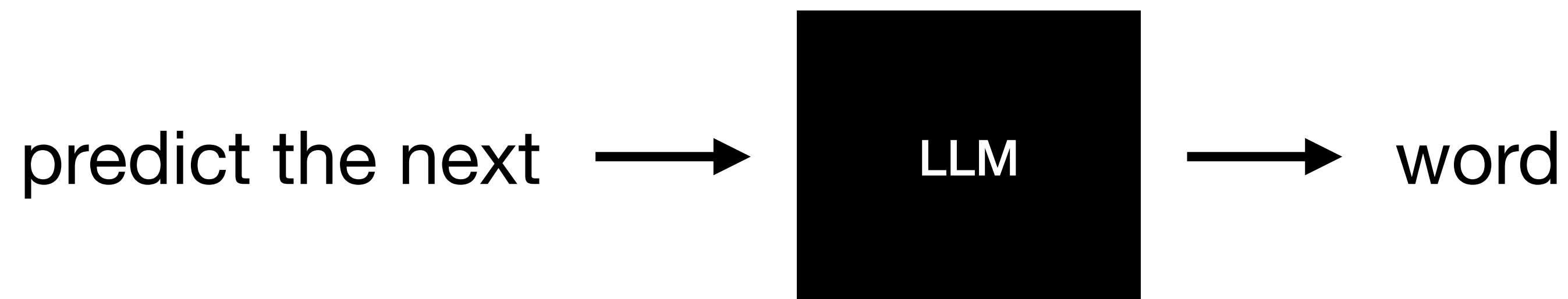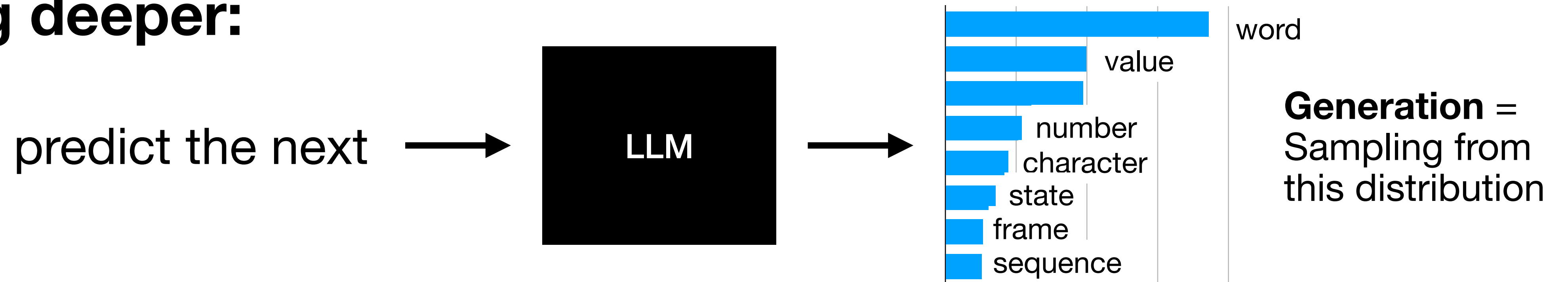2) Knowledge Distillation
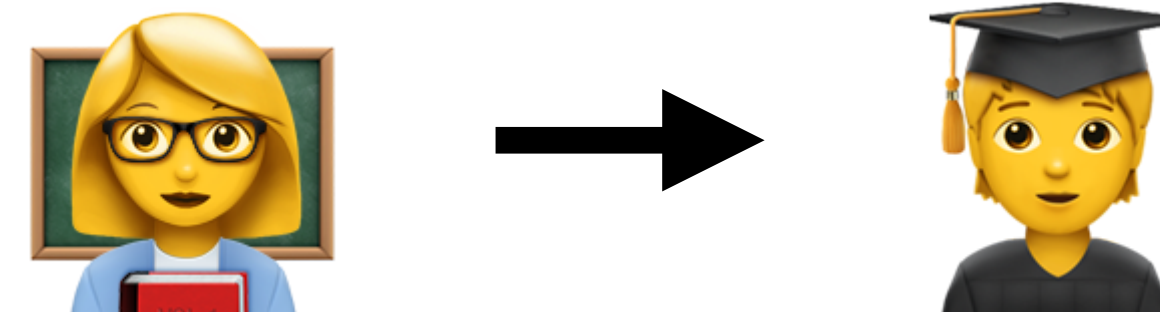
LLM Loss Function
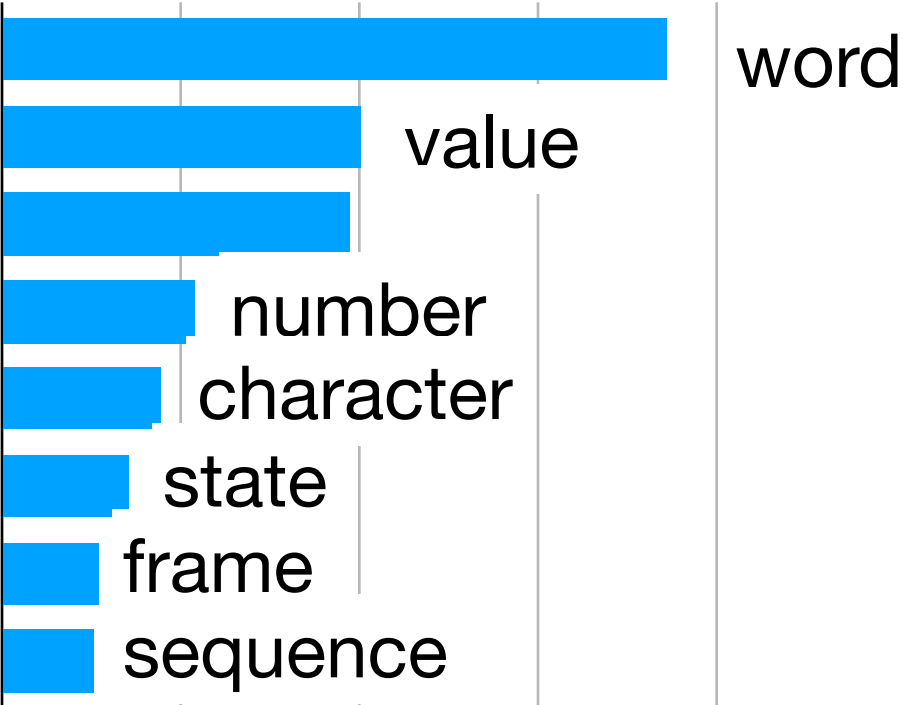
The math behind next-token prediction

Inference:

predict the next → [LLM] → word

Going deeper:

predict the next → [LLM] →

word
value

number
character
state
frame
sequence

Generation =
Sampling from
this distribution

ABB #4 - Spring 2025

LLM Loss Function

The math behind next-token prediction

Training:

predict the next $\longrightarrow$ | LLM | $\longrightarrow$

word
value
number
character
state
frame
sequence

Model prediction

word
value
number
character
state
frame
sequence

Ground truth

token

$$\mathscr{L} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

y_i = ground truth label for i^{th} token
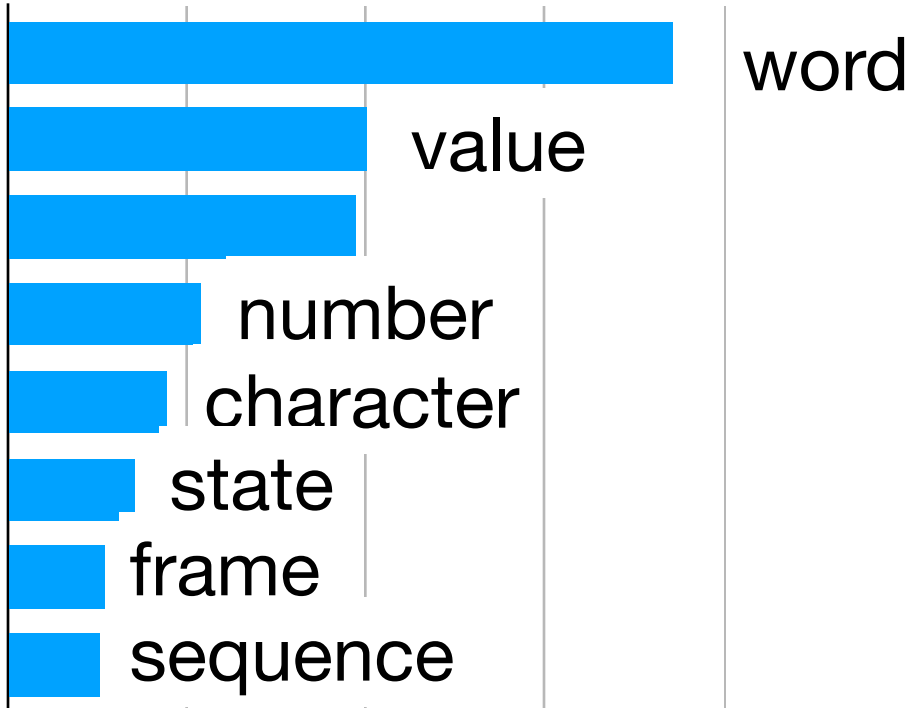
$\hat{y}_i$ = predicted logit for i^{th} token
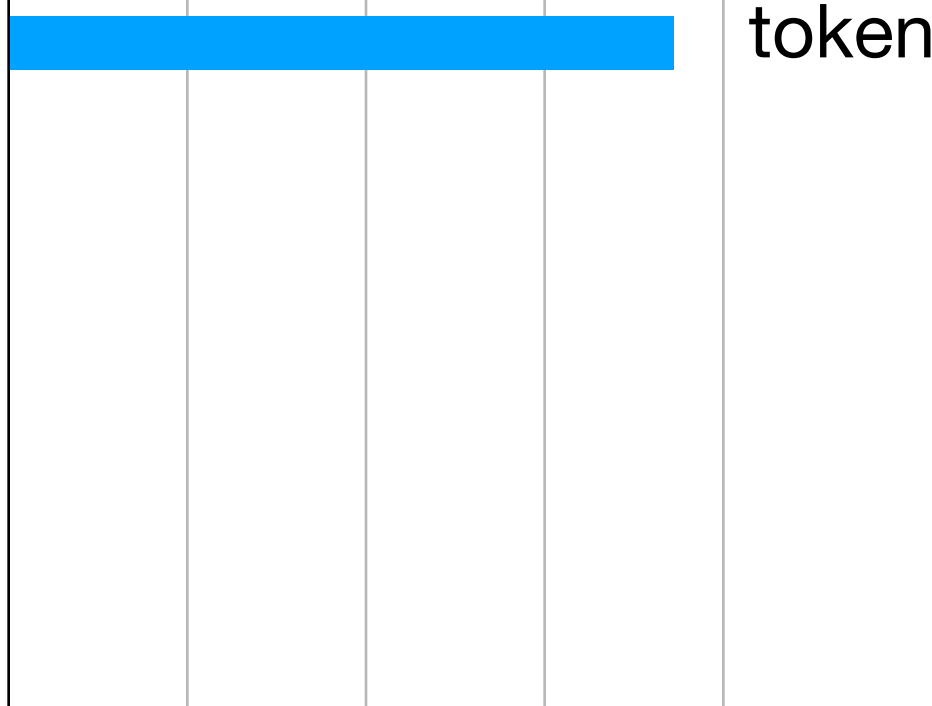
N = num tokens in vocabulary

For hard labels $\implies$ $\mathscr{L} = -\log(\hat{y}_t)$

Where, t = index of ground truth token

ABB #4 - Spring 2025