

ABB - Session 5

Software 3.0, Fine-tuning

Shaw Talebi

Today's Session

1. Housekeeping

- 1.1. Announcements
- 1.2. Homework 3

2. Fine-tuning

- 2.1. What is Fine-tuning?
- 2.2. When to Fine-tune
- 2.3. 3 Types of FT
- 2.4. Model Compression


3. Examples


- 3.1. Fine-tuning BERT for Text Classification
- 3.2. Fine-tuning a LinkedIn Post Writer


Announcements

1) Special Topics for Session 6


React with your #1


**Shaw Talebi** Yesterday at 8:01PM
Hello Builders!
Great to see the practical projects being submitted for HW4!
Looking ahead to Session 6, I will have time to cover 1-3 "special topics" (depending on depth) that haven't been covered yet.
What would you want to see?
100 = Multimodal Models
👉 = Guardrails
❤️ = AI Agents
😂 = Evals
👍 = Tool Use
🔥 = Going to prod
Talk soon,
Shaw
P.S. The emojis are unrelated to the topics. These are the only reaction options 😊





Comment the rest

**Pierluigi Chiusolo** Yesterday at 8:16PM
❤️
100
🔥

**Dushyant vishnoi** Yesterday at 9:28PM
👉 = Guardrails
❤️ = AI Agents
😂 = Evals
👍 = Tool Use

Homework 4

Shoutouts 🎉

Recipe Recommender System

Vladimir Belony

Applicant Matching System

Juan Ignacio Beiroa

Bluesky QA Bot

Claudia Ng

Stock Analysis Tool

Sangeeta Bahri

Foundational RAG System

Rod Morrison

Research Paper QA Bot

Serg

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



Fine-tuning

What is Fine-tuning?

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



Pre-trained Model

(Self-supervised)

15T Tokens

(Llama 3)

Fine-tuned Model

(Supervised)

~25B Tokens

(Llama 3.1 Instruct)

What is Fine-tuning?

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



**Fine-tuned
Model**

~25B Tokens

(Llama 3.1 Instruct)



**Additional
Fine-tuning**

~50k-5M Tokens

(Fine-tuning model for style)

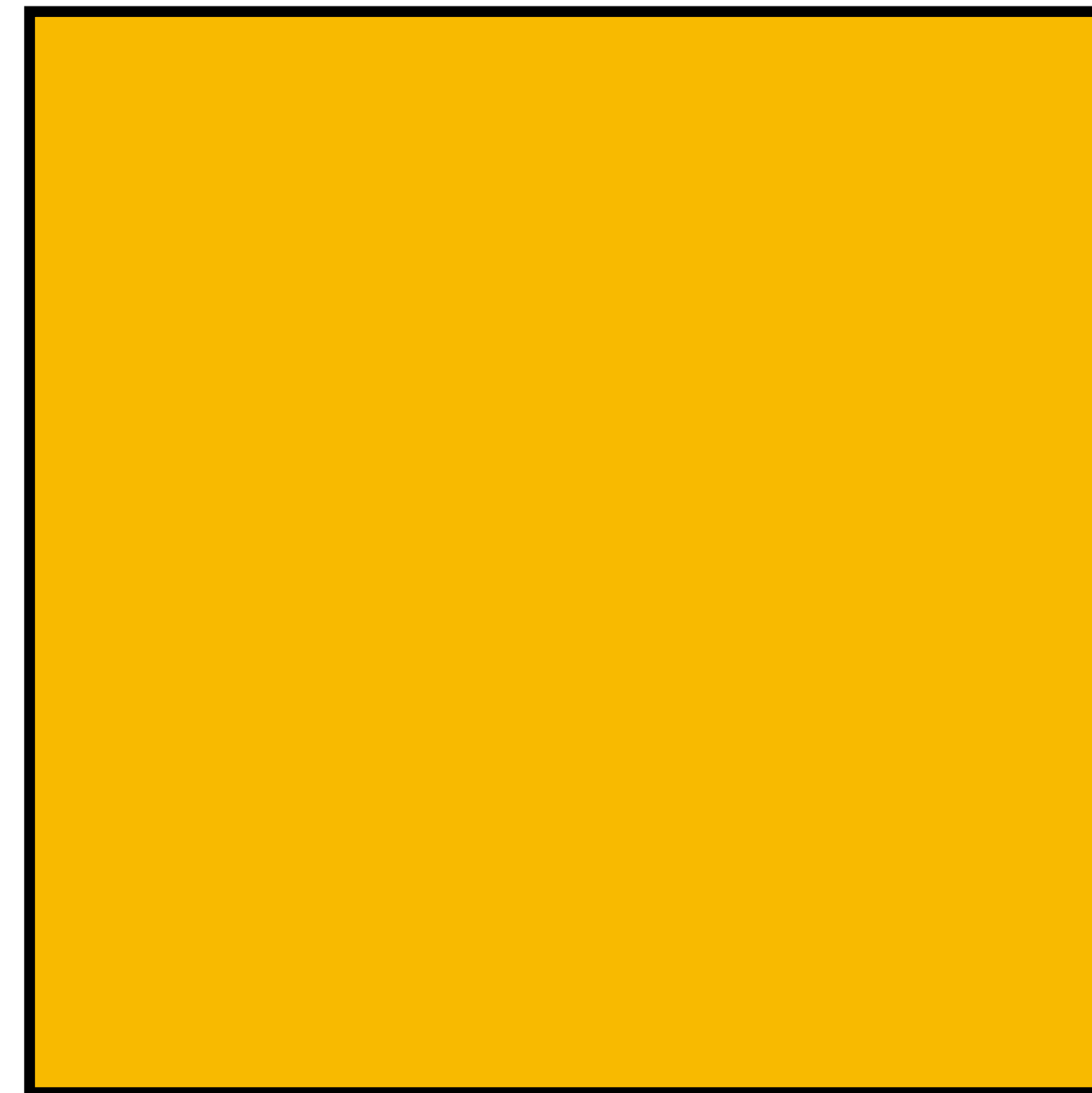
Why Fine-tune?

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



InstructGPT (1.3B)

>



GPT-3 (175B)

Why Fine-tune?

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

Base Model

(davinci (GPT-3))

Prompt: Tell me how to fine-tune a model

Completion:

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?

Fine-tuned Model

(text-davinci-003)

Prompt: Tell me how to fine-tune a model

Completion:

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.

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 to Fine-tune

3 common use cases

1. Customizing style or tone 

2. Improving reliability of desired output structure 

3. Performing tasks that are hard to articulate via prompts 

When NOT to Fine-tune

3 things to consider

1. Before improving your prompt



2. Before trying RAG



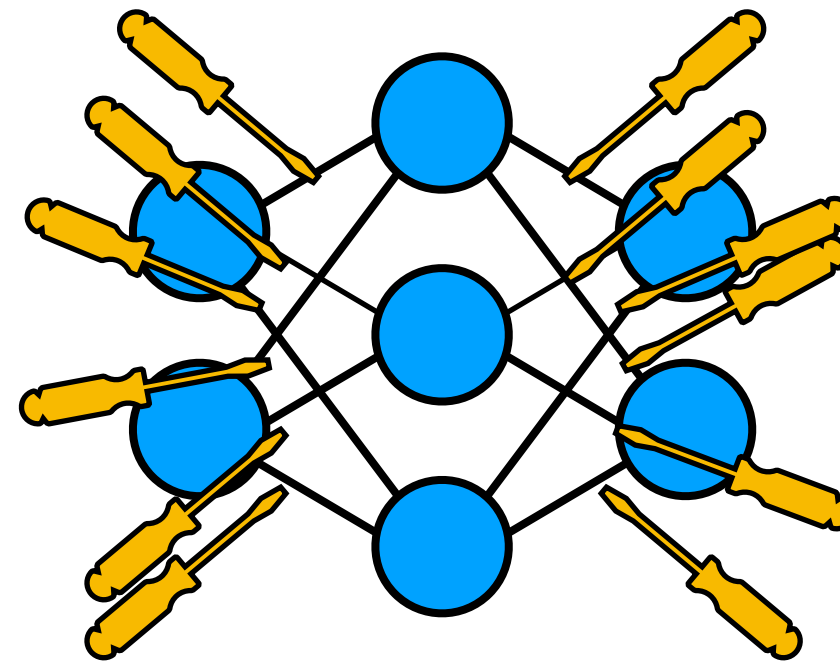
3. Trying to teach model new knowledge



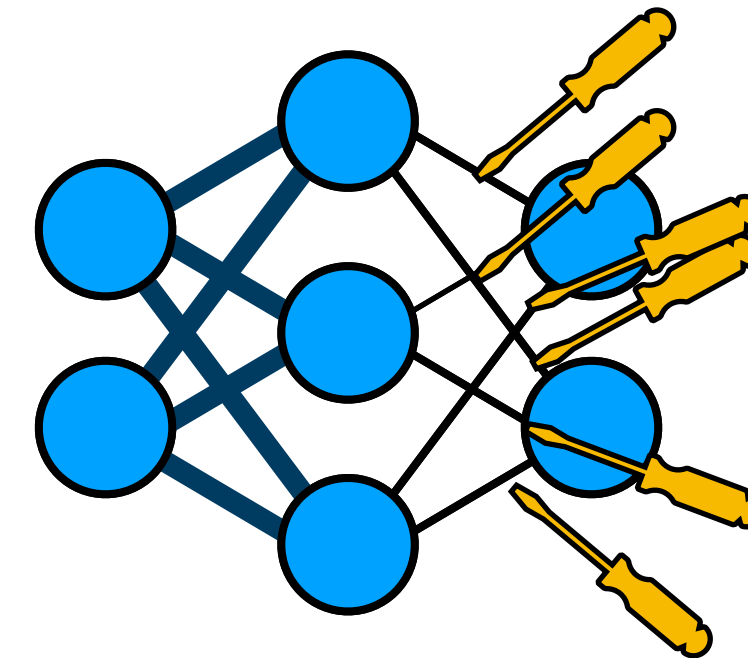
How Do I Fine-tune?

3 Flavors

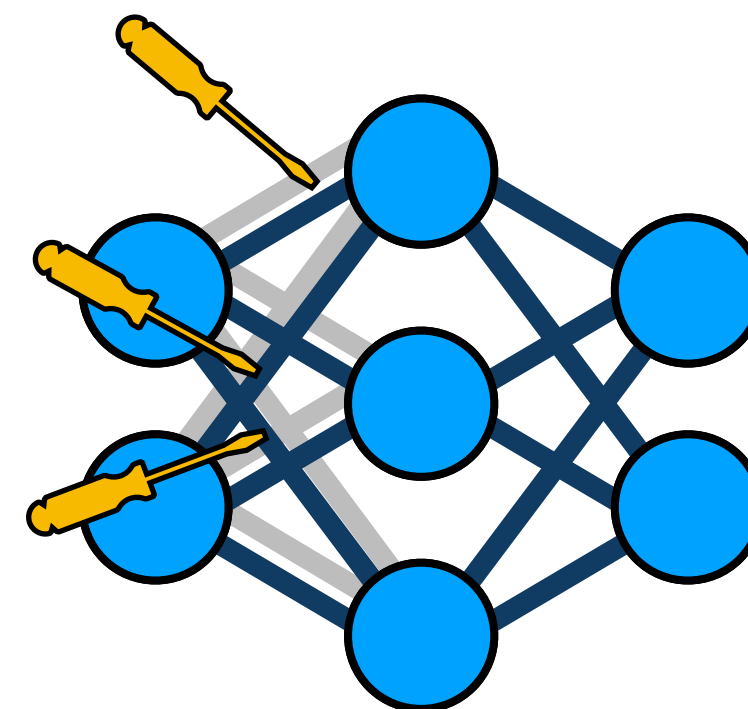
1) Retrain all parameters
aka “Full Fine-tuning”



2) Retrain some parameters
aka “Transfer Learning”

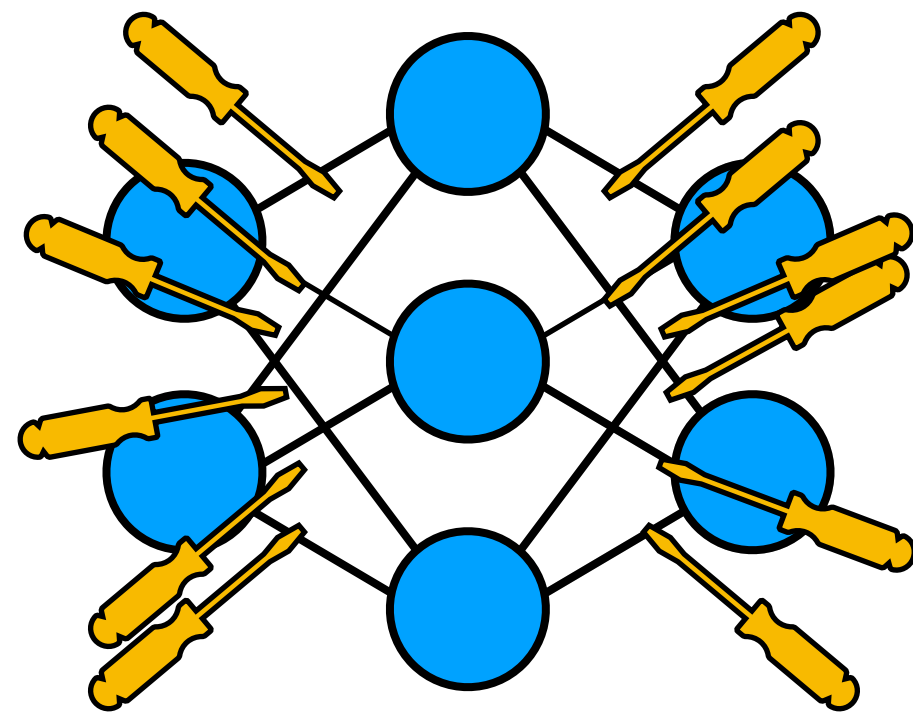


3) Do something more clever
aka “Parameter-Efficient Fine-tuning (PEFT)”

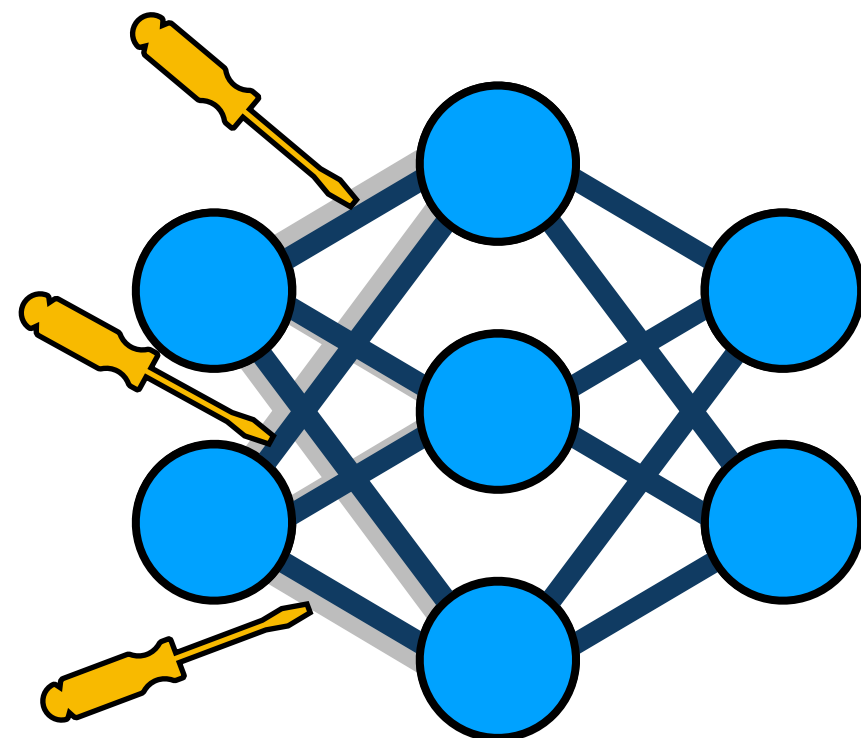


LoRA (Low-Rank Adaptation)

Fine-tunes model by adding **small set** of trainable parameters



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



Full Fine-tuning: $h(x) = W_0 x$

$$\begin{matrix} \boxed{W_0} & x & = & h(x) \\ \text{Trainable} & & & \end{matrix}$$

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

$$\begin{matrix} \left(\boxed{W_0} + \boxed{B \quad A} \right) & x & = & h(x) \\ \text{Frozen} & \text{Trainable} & & \end{matrix}$$

100-1000X savings!

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



Examples

Example 1

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/

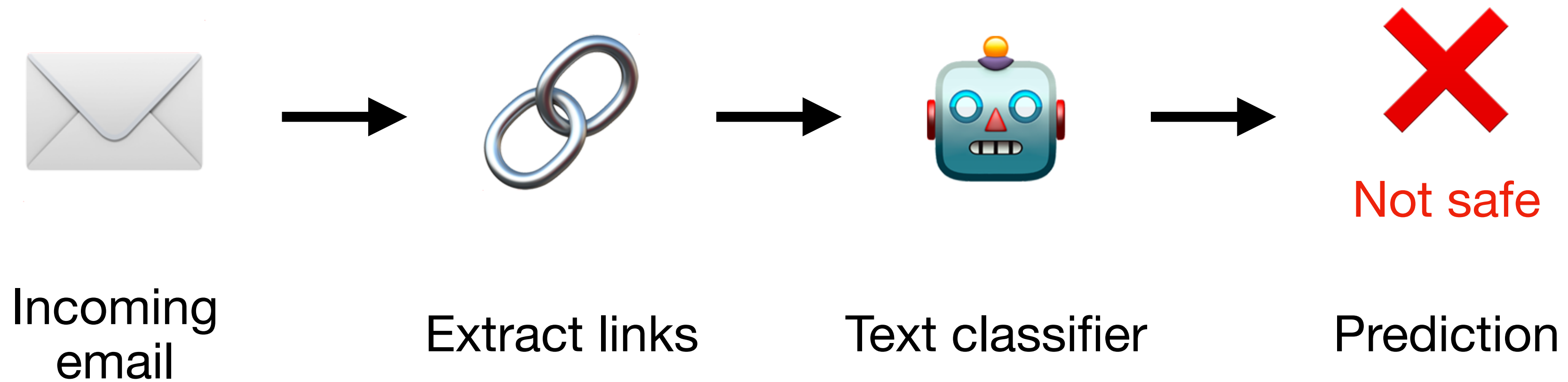
For more details, please check the terms on our website or feel free to reach out with any questions.

We look forward to hearing from you soon!

Best Regards,
The Service Center Team

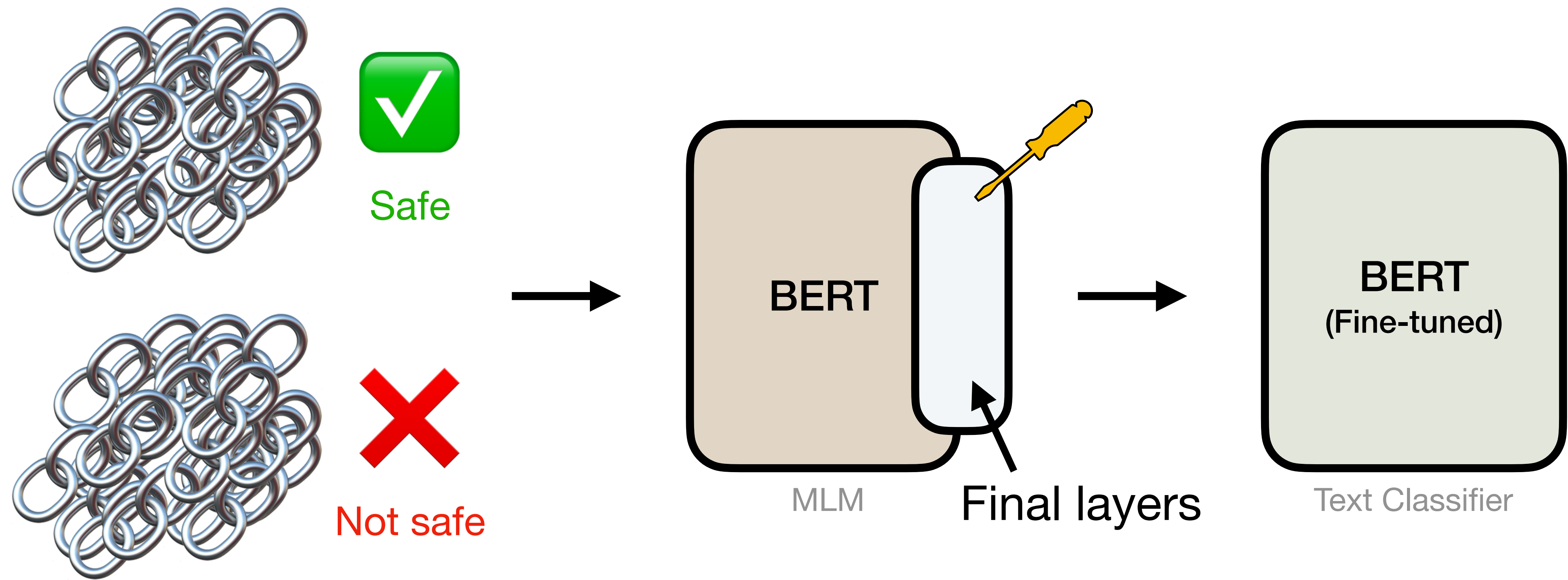
Example 1

Fine-tuning BERT for Text Classification (Overview)



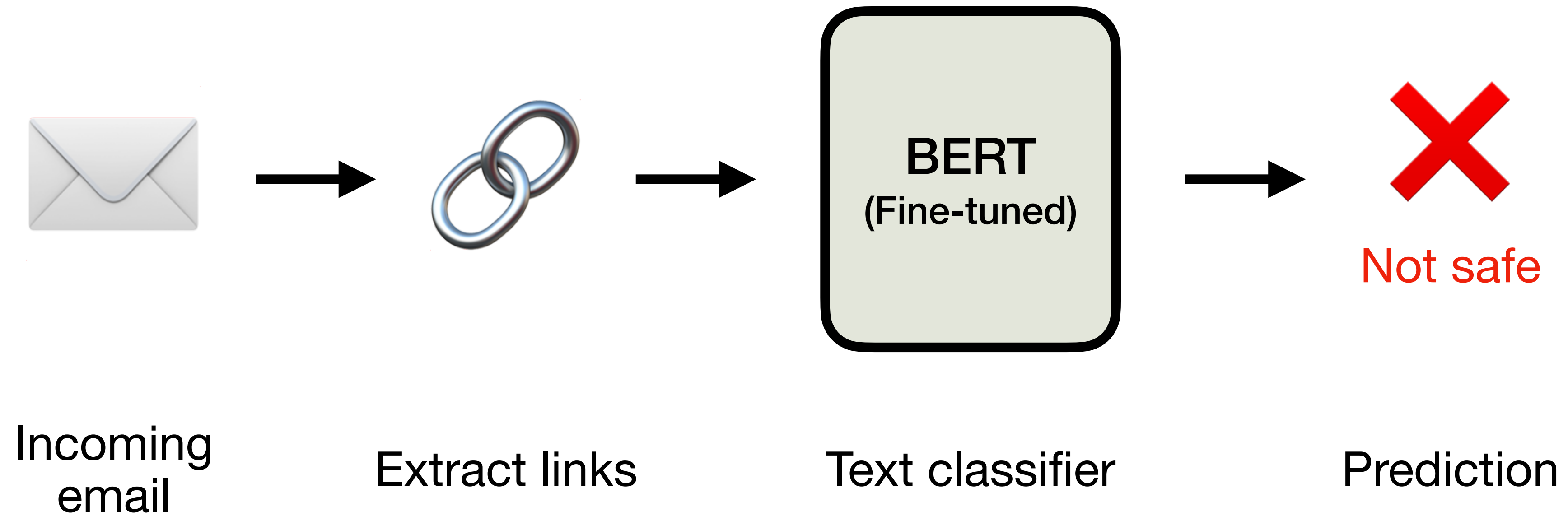
Example 1

Fine-tuning BERT for Text Classification (Overview)



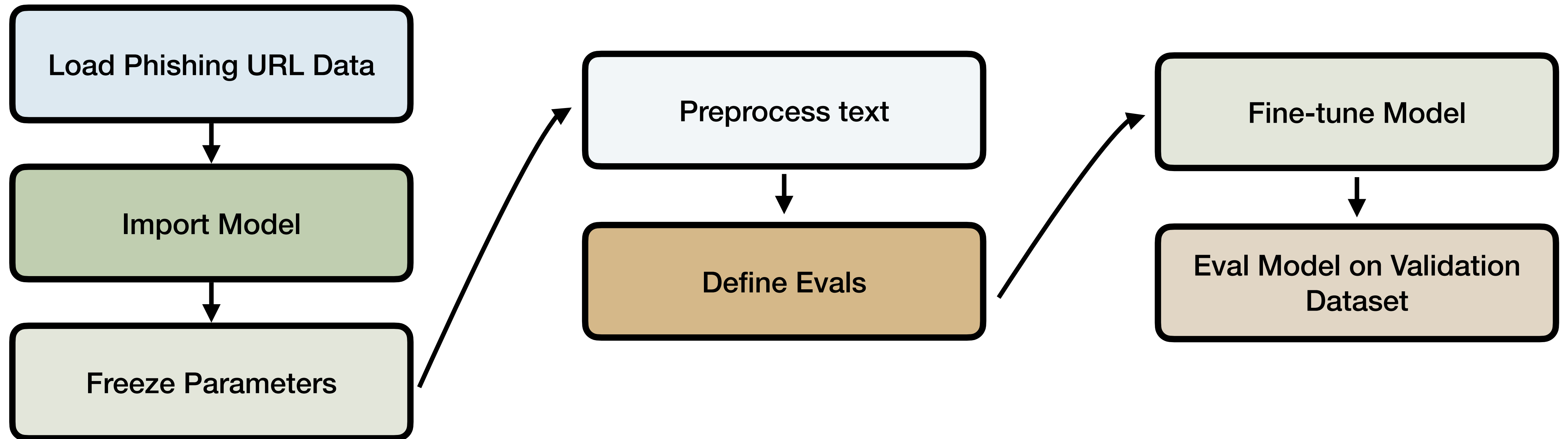
Example 1

Fine-tuning BERT for Text Classification (Overview)



Example 1

Fine-tuning BERT for Text Classification (Flowchart)



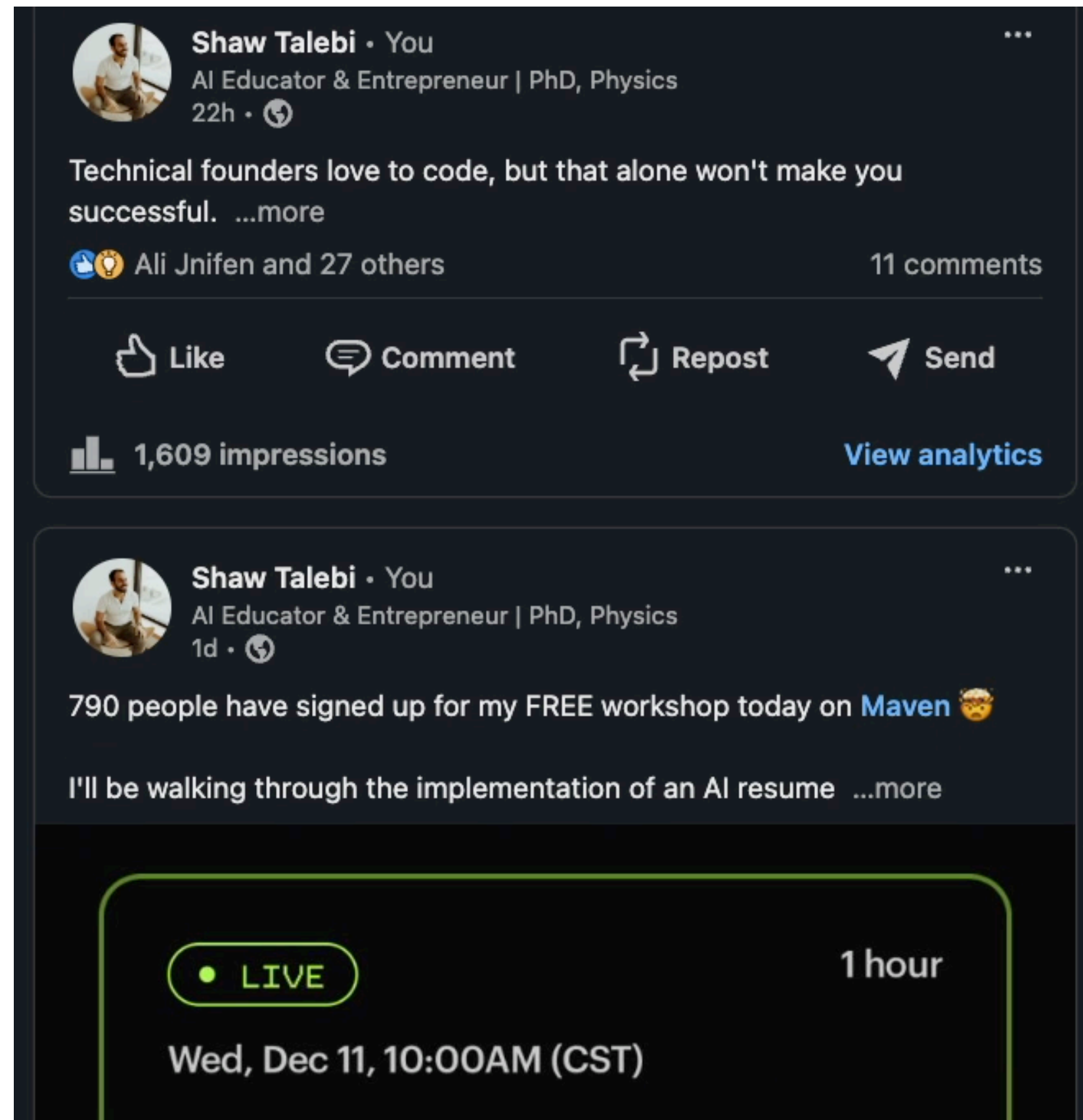
Example 1

Fine-tuning BERT for Text Classification (Code)



Example 2

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

Example 2

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.

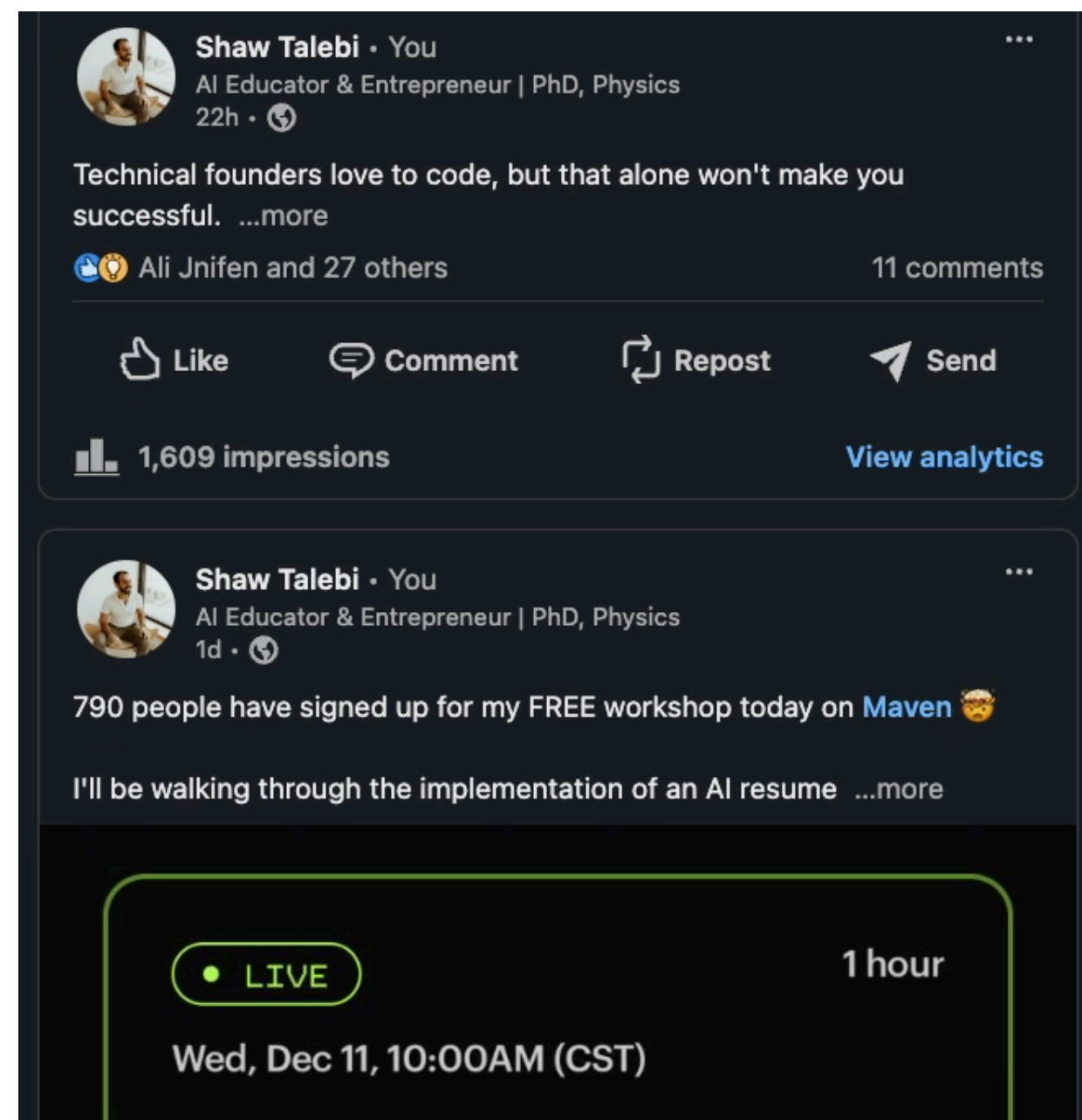
Write code that *works first*. Worry about making it "pretty" or efficient later. (Perfection is the enemy of progress.)

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.

Example 2

Fine-tuning a LinkedIn Post Writer (Overview)



Extract LI posts

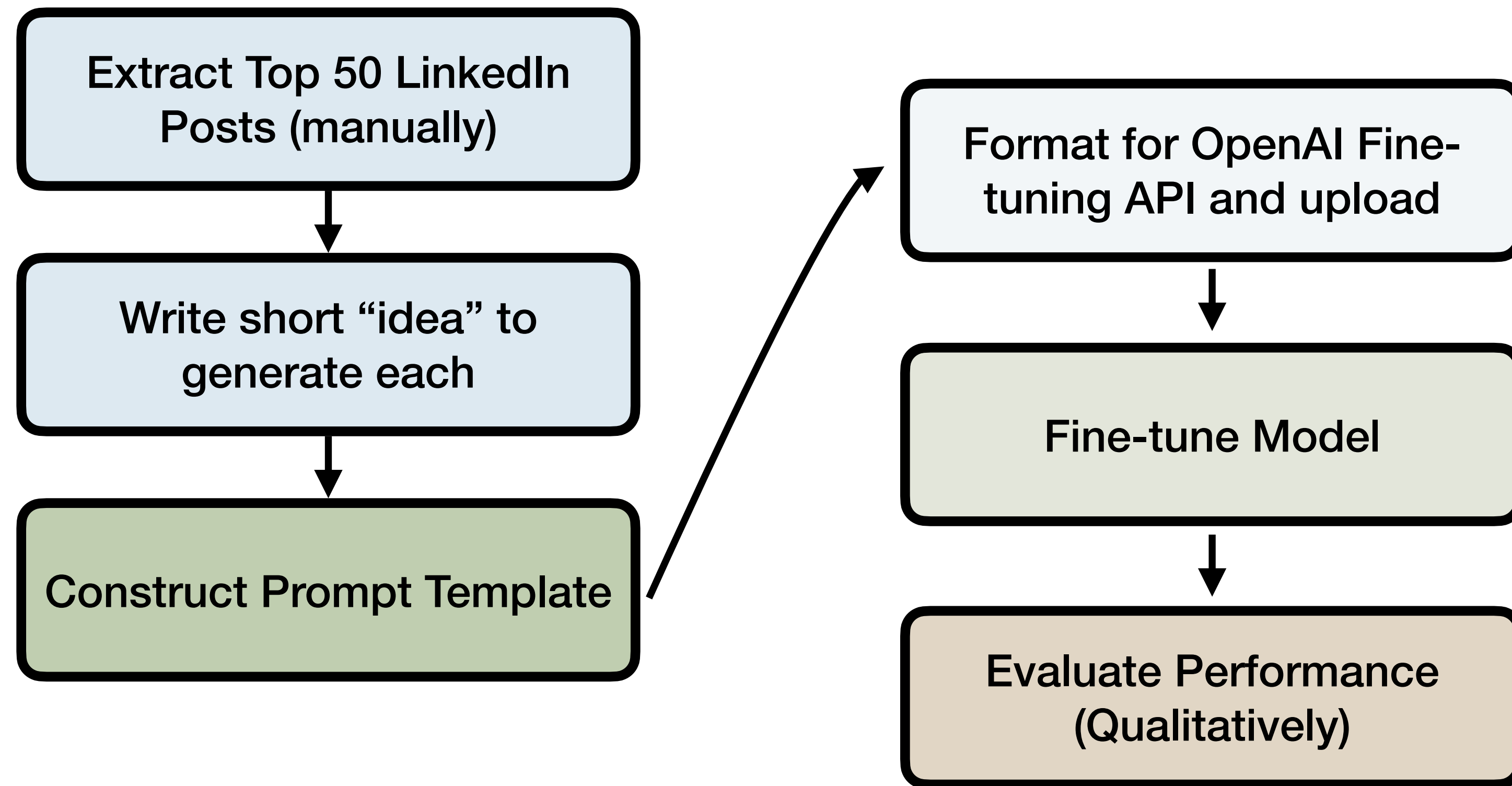


Fine-tune GPT-4o-mini

LI Post Writer
(in my likeness)

Example 2

Fine-tuning a LinkedIn Post Writer (Flowchart)



Example 2

Fine-tuning a LinkedIn Post Writer (Code)



Homework 5

Project

Fine-tune a Model (More examples provided)

Pre-work

Session 6: AI Project Management

Session 6: Project Discovery Questions

★ = more FT examples

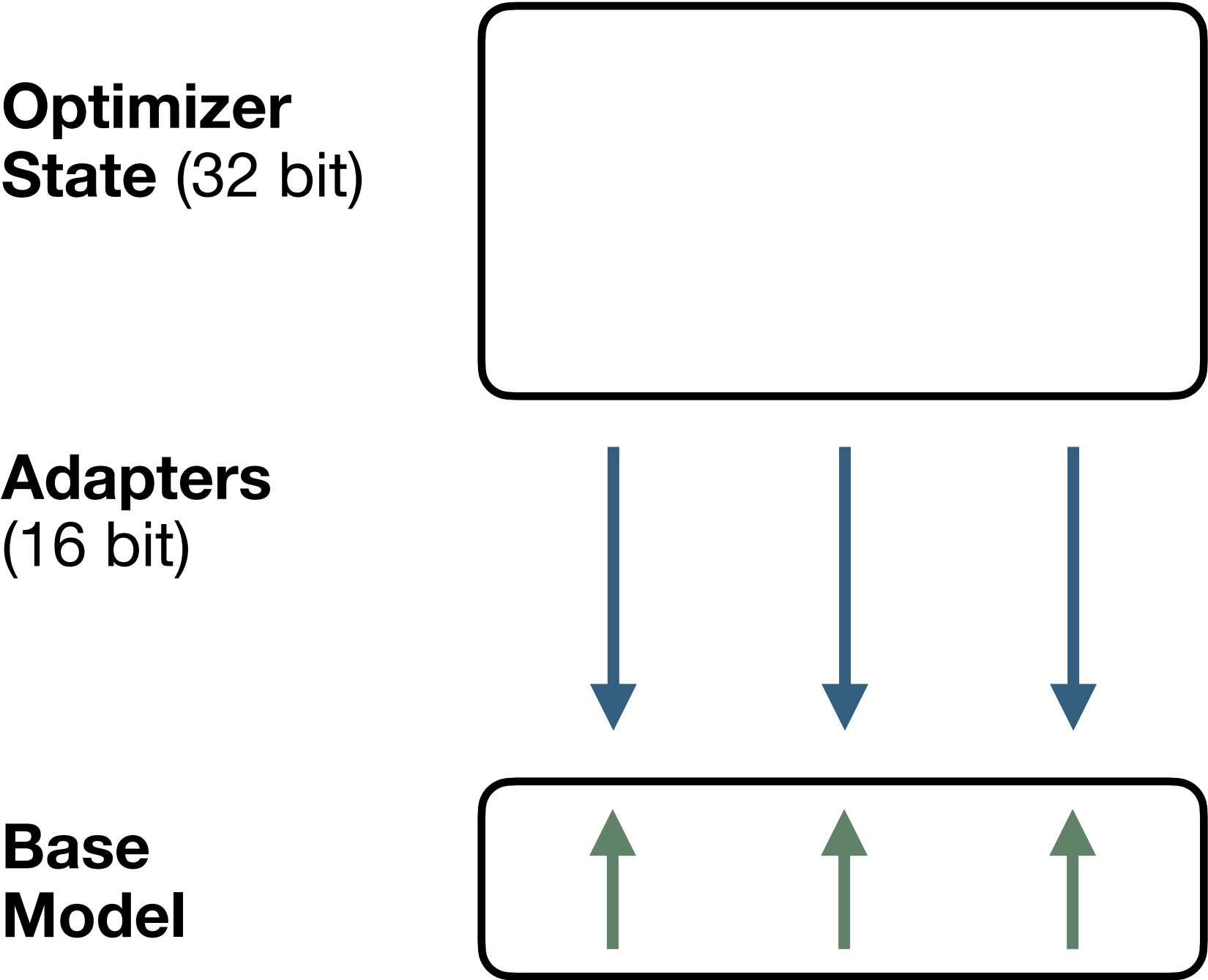
References

- [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 ★

QLoRA (Quantized LoRA)

Combining LoRA and Quantization

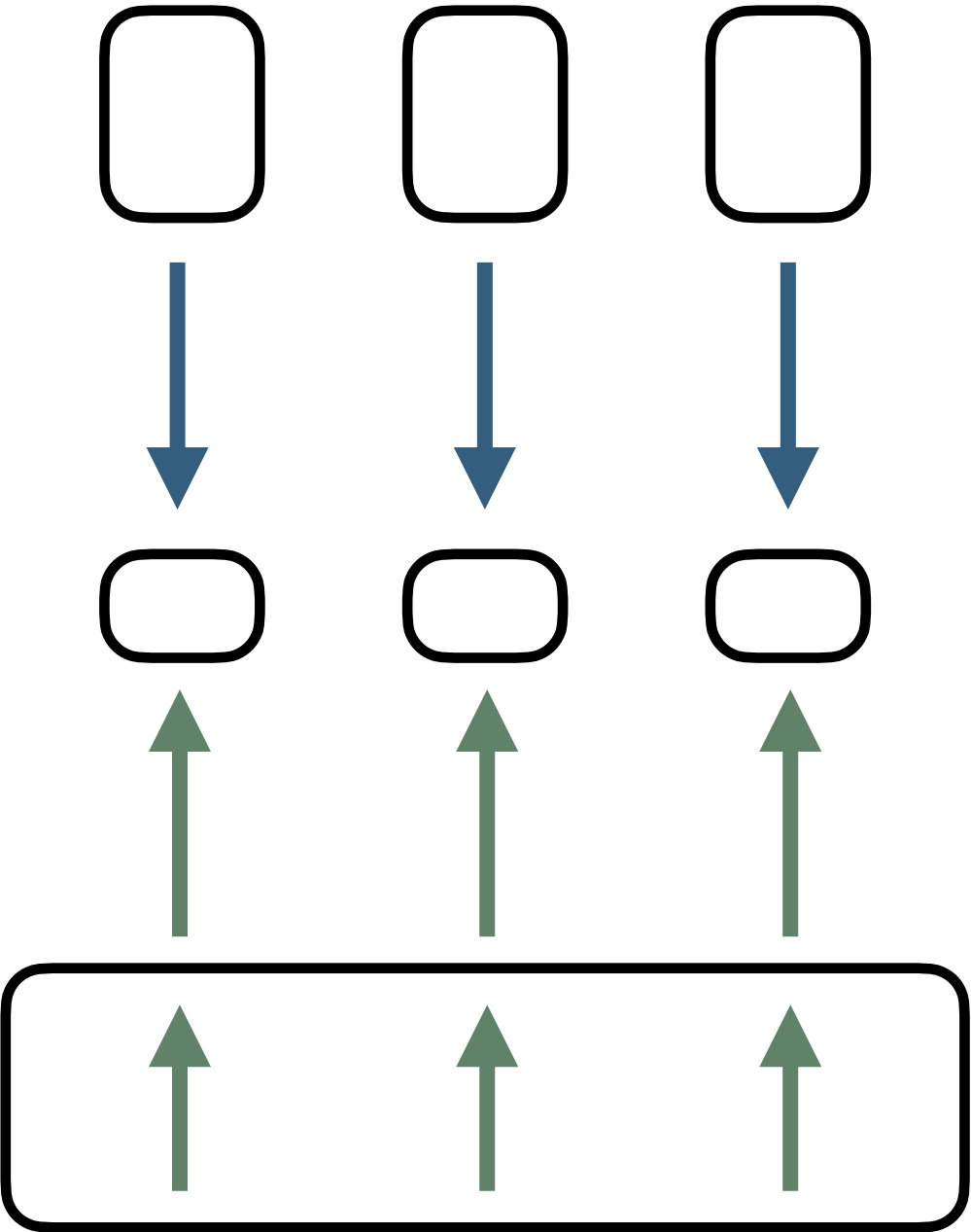
Full Finetuning



FP16

10B \Rightarrow 160GB

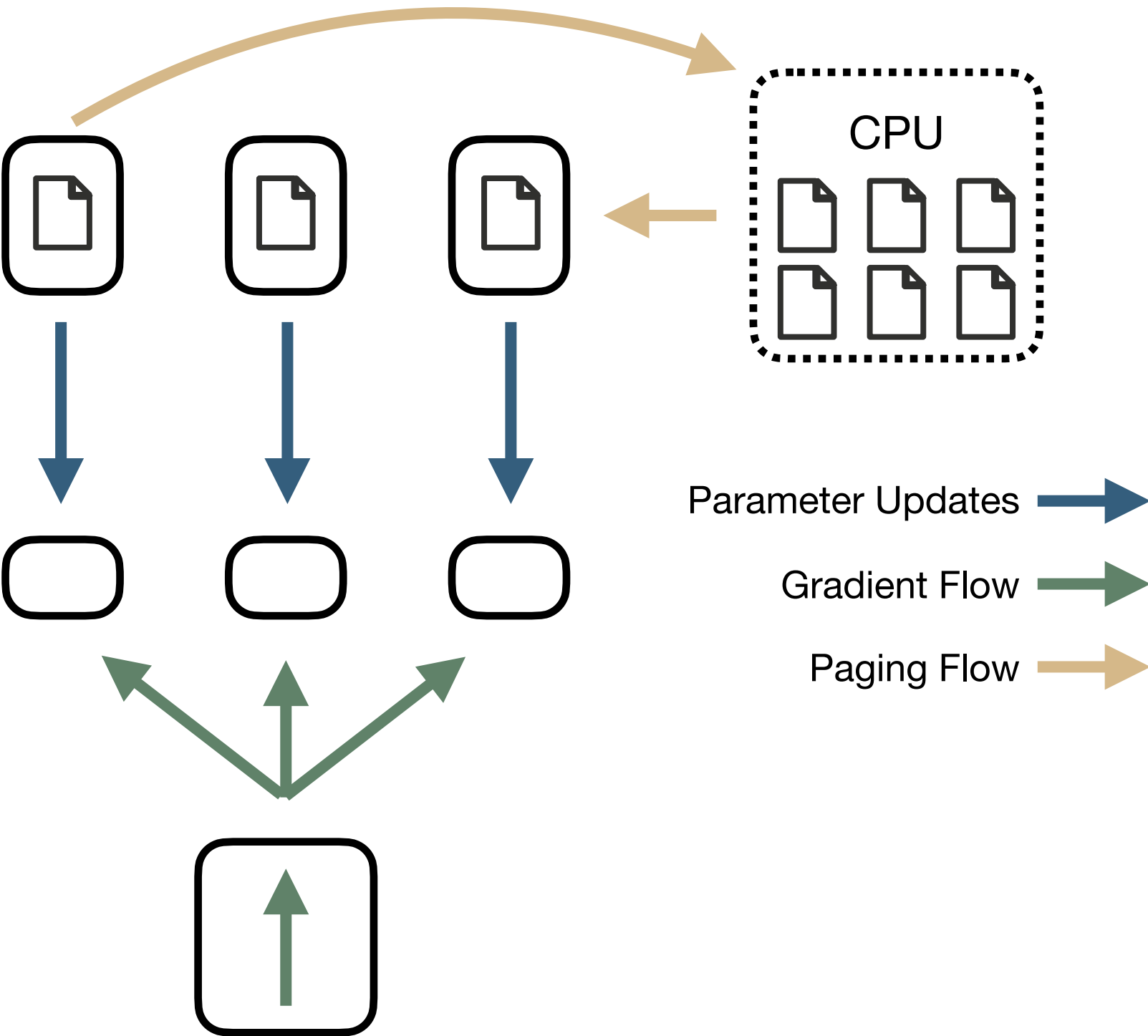
LoRA



FP16

10B \Rightarrow ~40GB

QLoRA



4-bit NormalFloat

10B \Rightarrow ~12GB