UVA DATA SCIENCE

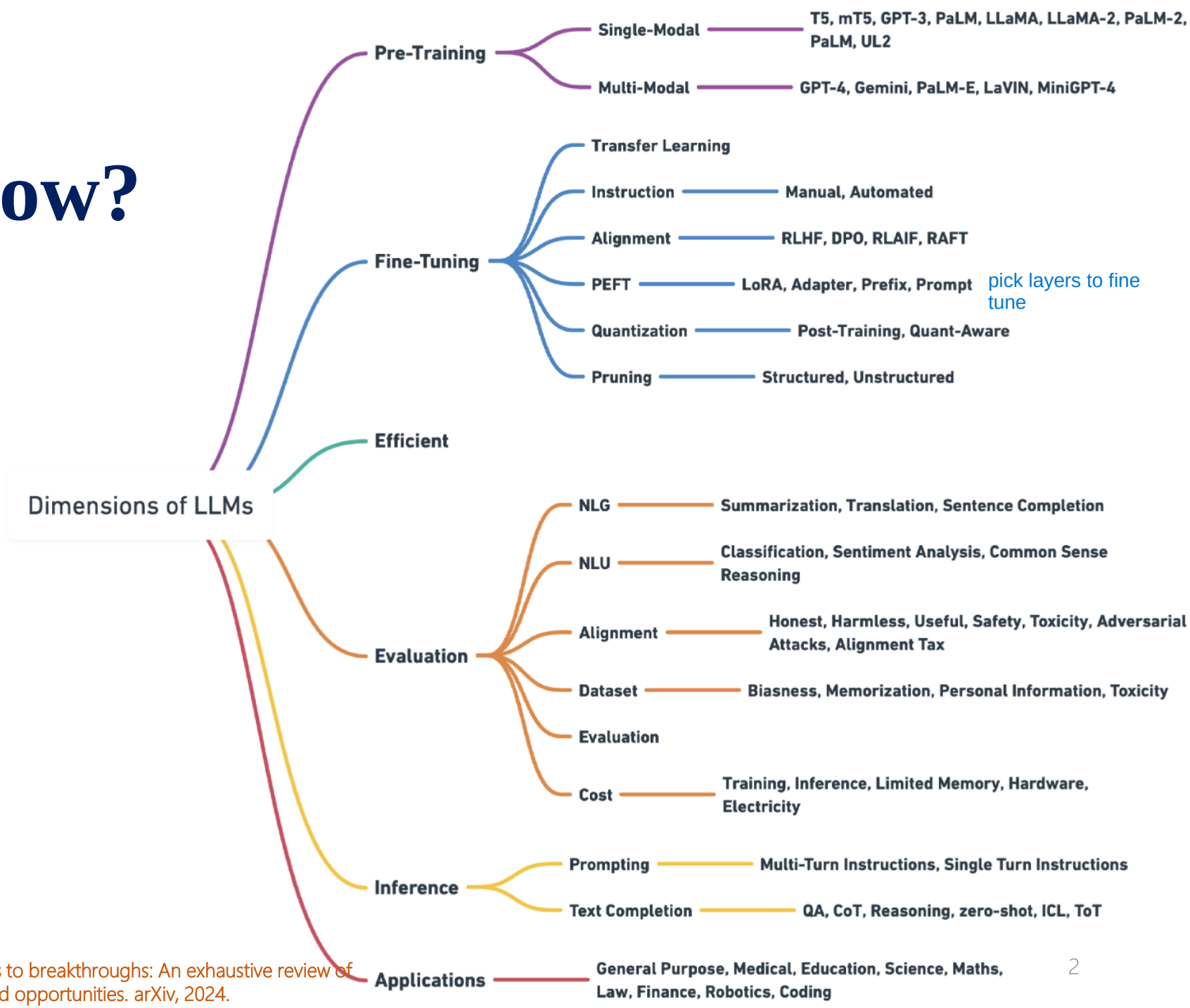
DS 6051 Decoding Large Language Models

Finetuning LLMs

For our project, we want to do this
Take a pretrained llm and finetune it for our task

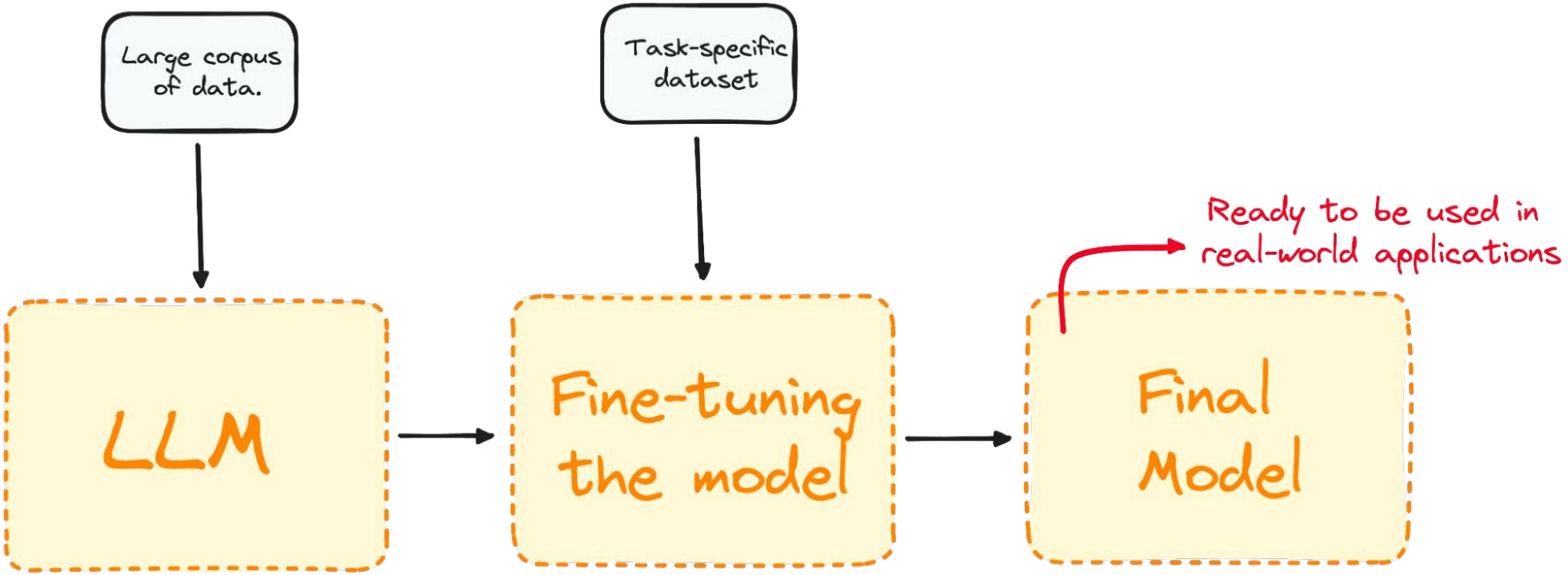
Chirag Agarwal
Assistant Professor
School of Data Science
University of Virginia

What have we learned until now?



Parthasarathy et al. The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities. arXiv, 2024.

What is Fine-tuning, and Why is it Important?



https://www.datacamp.com/tutorial/fine-tuning-large-language-models

Types of Fine-tuning

- Supervised Fine-tuning
- Few-shot (or in-context) learning

providing k short examples of how want model to perform and concating them with actual query and then asking the question

- Transfer learning

use learned weights from another model to give yourself a good first guess
Deepseek did this to chat, it made a giant dataset by promting chat so many times and then used that to fine tune the model (note not the ground truth)

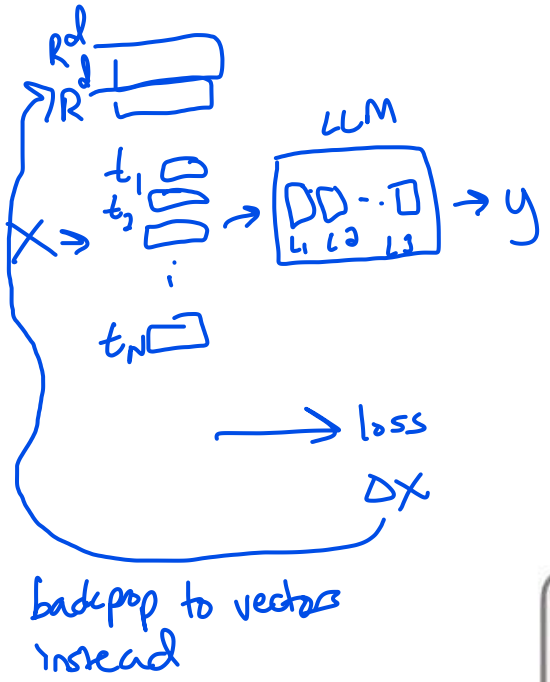
- Prefix Tuning

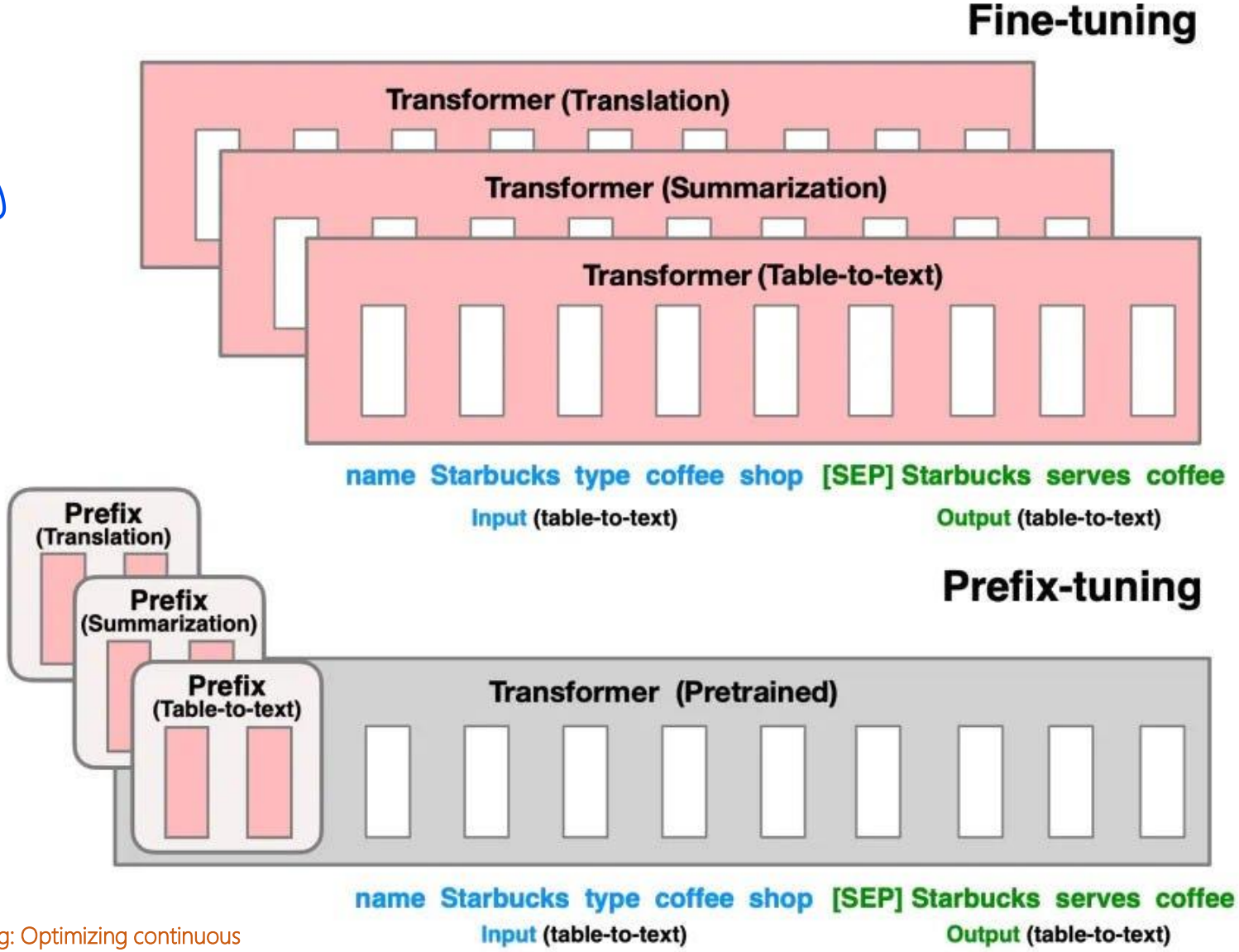
Before AI

- Parameter Efficient Fine-tuning (PEFT)

Prefix Tuning

In 2021 did not have generate models



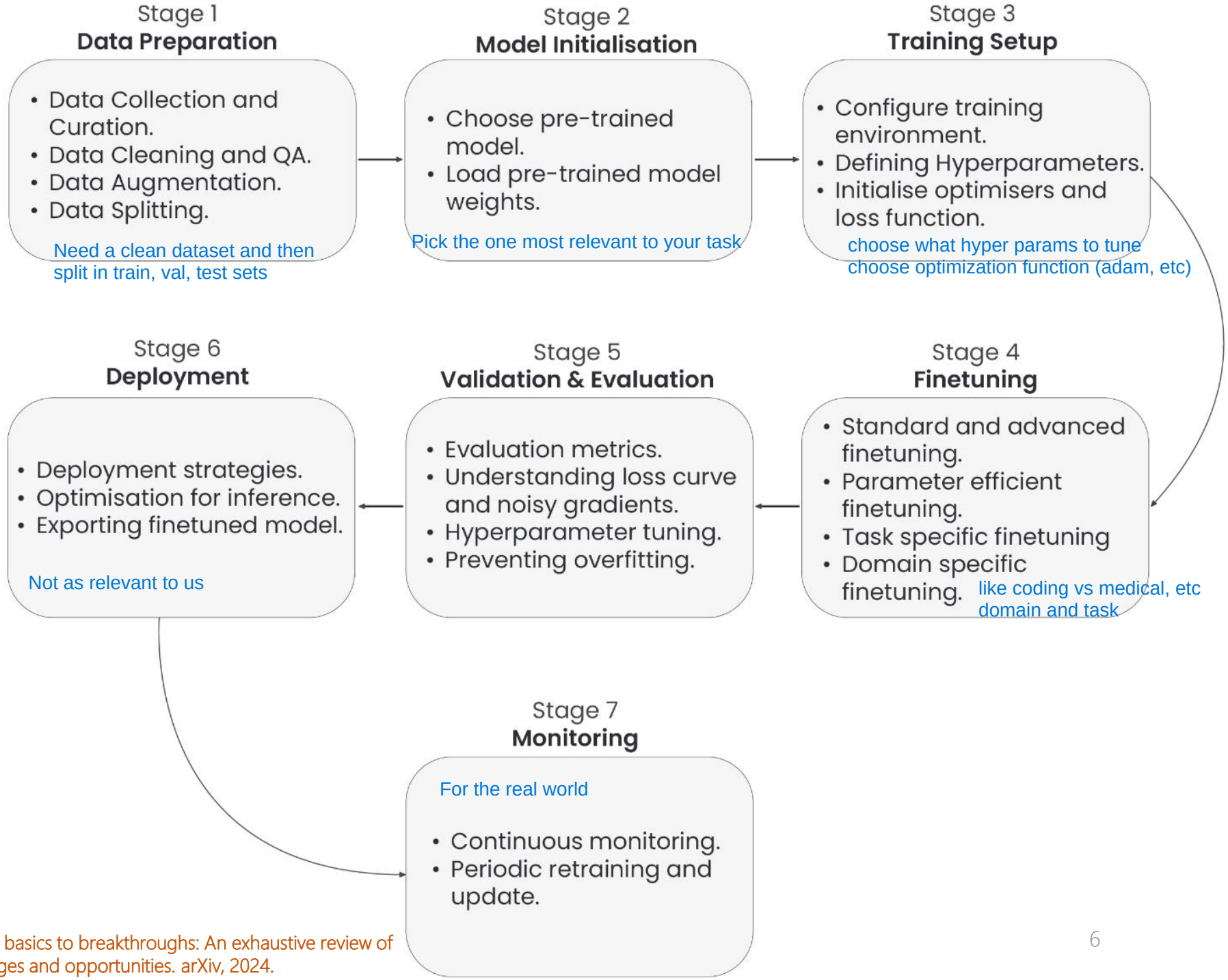


change weights of entire model in very small way

They fixed the weights of the transformer and added new token embeddings and randomly init, then rather than fine tuning the weights, they fine tuned the new tokens

Li, Xiang Lisa, and Percy Liang. "Prefix-tuning: Optimizing continuous prompts for generation." arXiv, 2021

Seven Stage Fine-Tuning Pipeline for LLMs



Parthasarathy et al. The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities. arXiv, 2024.

Fine-tuning Best Practices

- Data Quality and Quantity
- Hyperparameter tuning
- Regular evaluation
- Computational requirements

Fine-tuning Pitfalls

- Overfitting — Training loss converges but val does not
- Underfitting
- Catastrophic forgetting — if you don't ask about older tasks often, they forget
  like if you learned how to play soccer but then do not play for a while, when you are asked to play again you are not as good
  Don't want it to learn new skills and then forget older skills

What Parameters to Fine-tune??

- Optimizers sgd, adam, etc
- Learning Rates usually around 2e-4 or 1e-4
- Learning rate Schedule when init an optimizer, use a learning rate to start, how to auto change learning rate as number of epochs changes
- Memory Optimization Parameters (starting to be importatnt)
    - Gradient Checkpointing when to checkpt gradient values
    - Gradient Accumulation rather than backprop for every batch, it accumulates for several batches, it aggregates and then backprops
- Quantization next week have a guest lecture on this
- Regularization Techniques
    - Batch Size
    - Noise Embeddings
    - Label Smoothing

https://medium.com/@rakeshsharma.pr/how-to-get-started-with-finetuning-llms-84affc9e3cbe

Parameter-Efficient Fine-Tuning Techniques

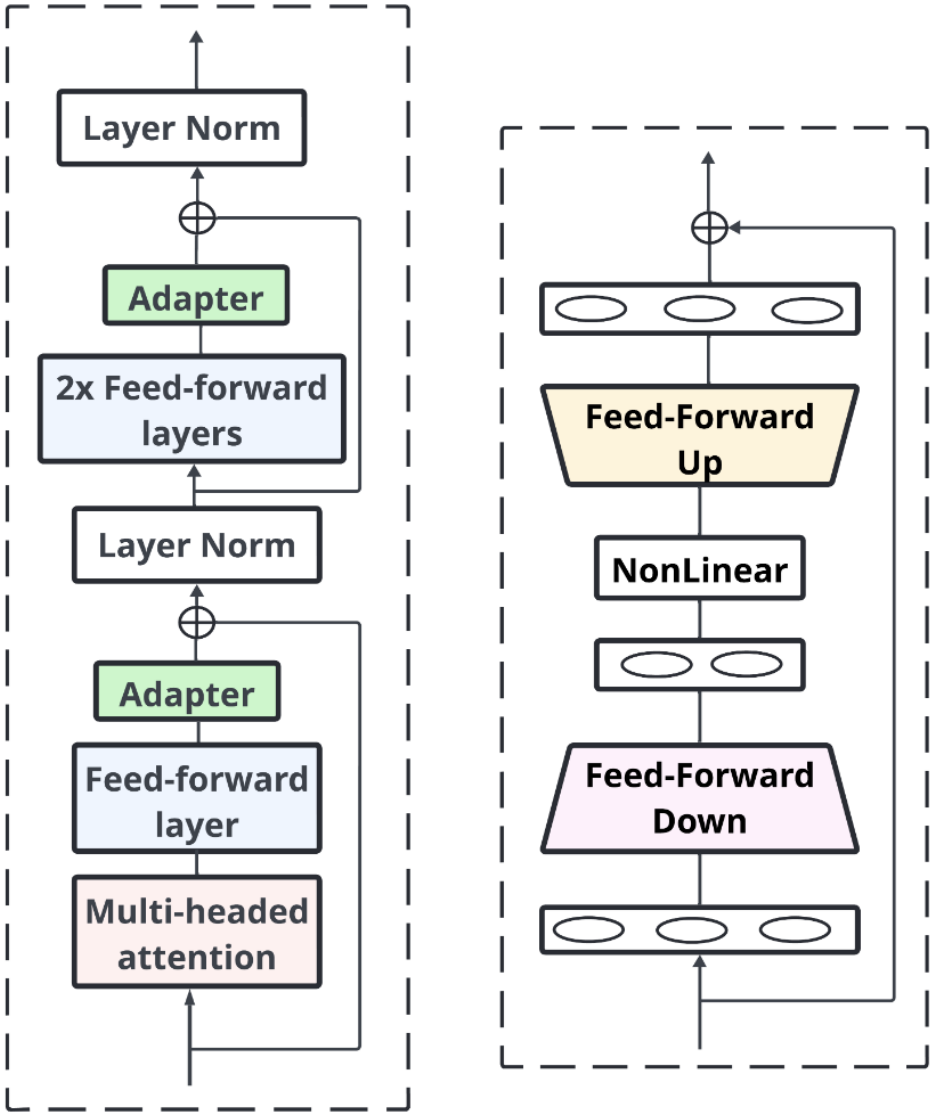
PEFT^

- Adapters
- Low-Rank Adaptation (LoRA)
- QLoRA Quantization LoRA
- Weight-Decomposed Low-Rank Adaptation (DoRA)
- Fine-Tuning with Multiple Adapters

https://medium.com/@rakeshsharma.pr/how-to-get-started-with-finetuning-llms-84affc9e3cbe

Adapters

-rather than me fine tuning entire llm, learn adaptors, focus on a few layers



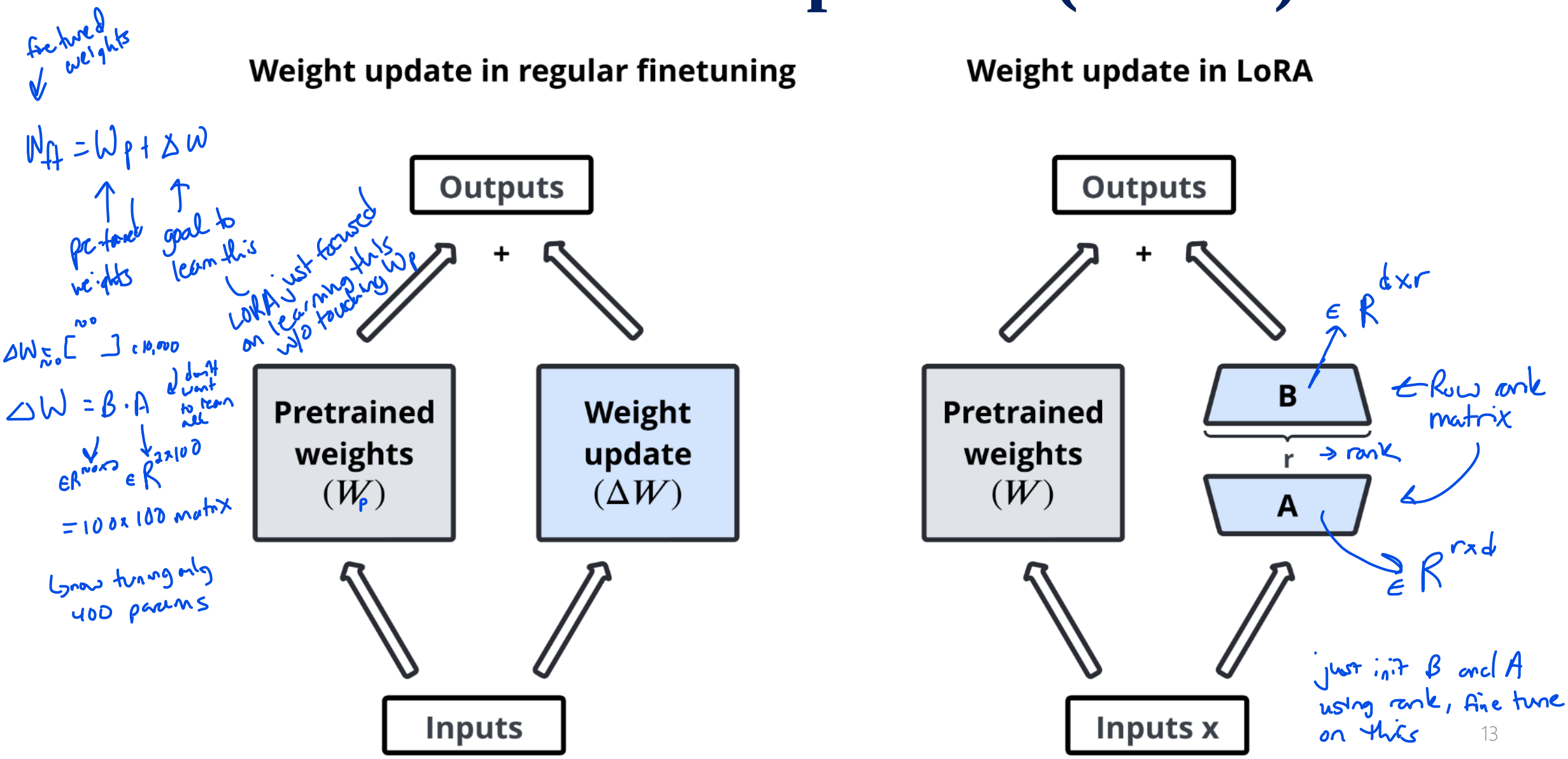
Low-Rank Adaptation (LoRA)

a version that got a lot of attention

- Designed for fine-tuning LLMs, which modifies the fine-tuning process by freezing the original model weights and applying changes to a separate set of weights, added to the original parameters

- LoRA transforms the model parameters into a lowerrank dimension, reducing the number of trainable parameters, speeding up the process, and lowering costs

Low-Rank Adaptation (LoRA)

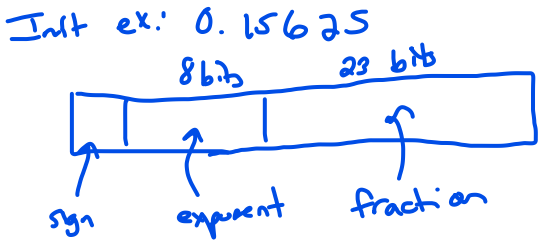


Low-Rank Adaptation (LoRA)

- Benefits:
  - Parameter Efficiency
  - Efficient Storage
  - Lower Memory Footprint
  - Task-Specific Adaptation
  - Comparable Results

- Limitations
  - Fine-tuning Scope
  - Hyperparameter Optimization

weights & params of the model



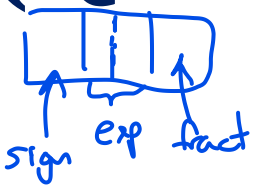
= 32 bits c this for every weight/param

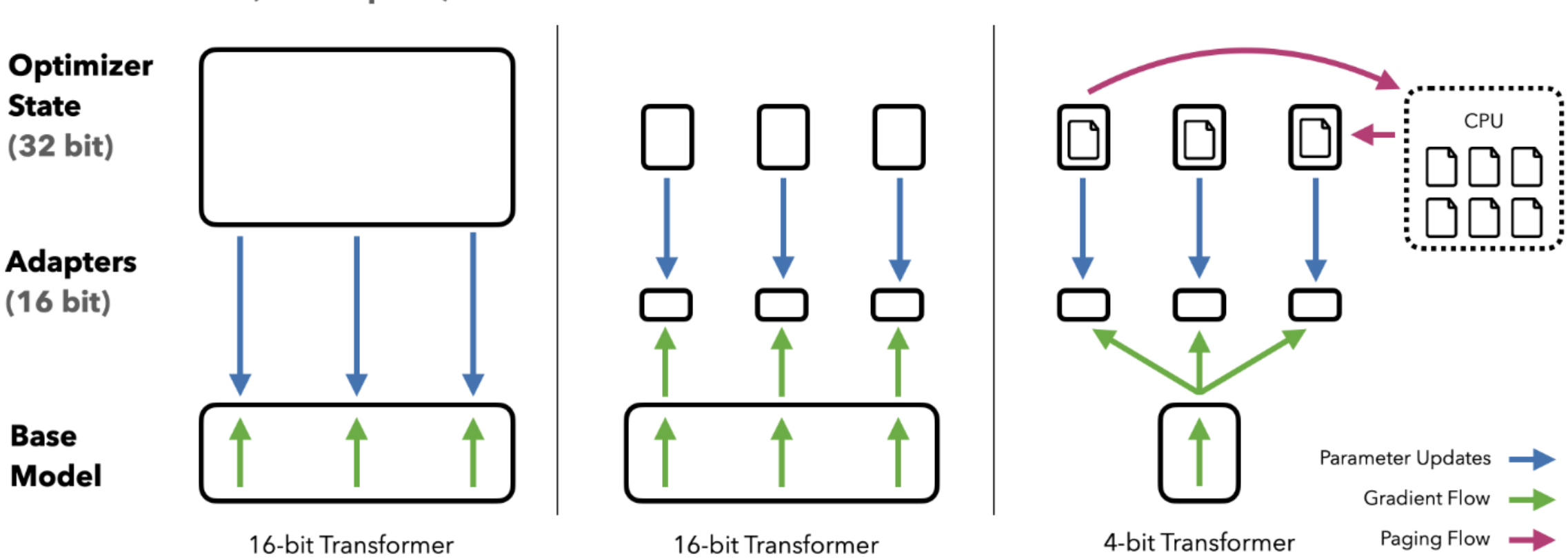
Quantized LoRA (Q-LoRA)

– Brief before guest Tim speaks next week

Q: Do we need this specific → (Does take some performance hit)

4 bit LLM:

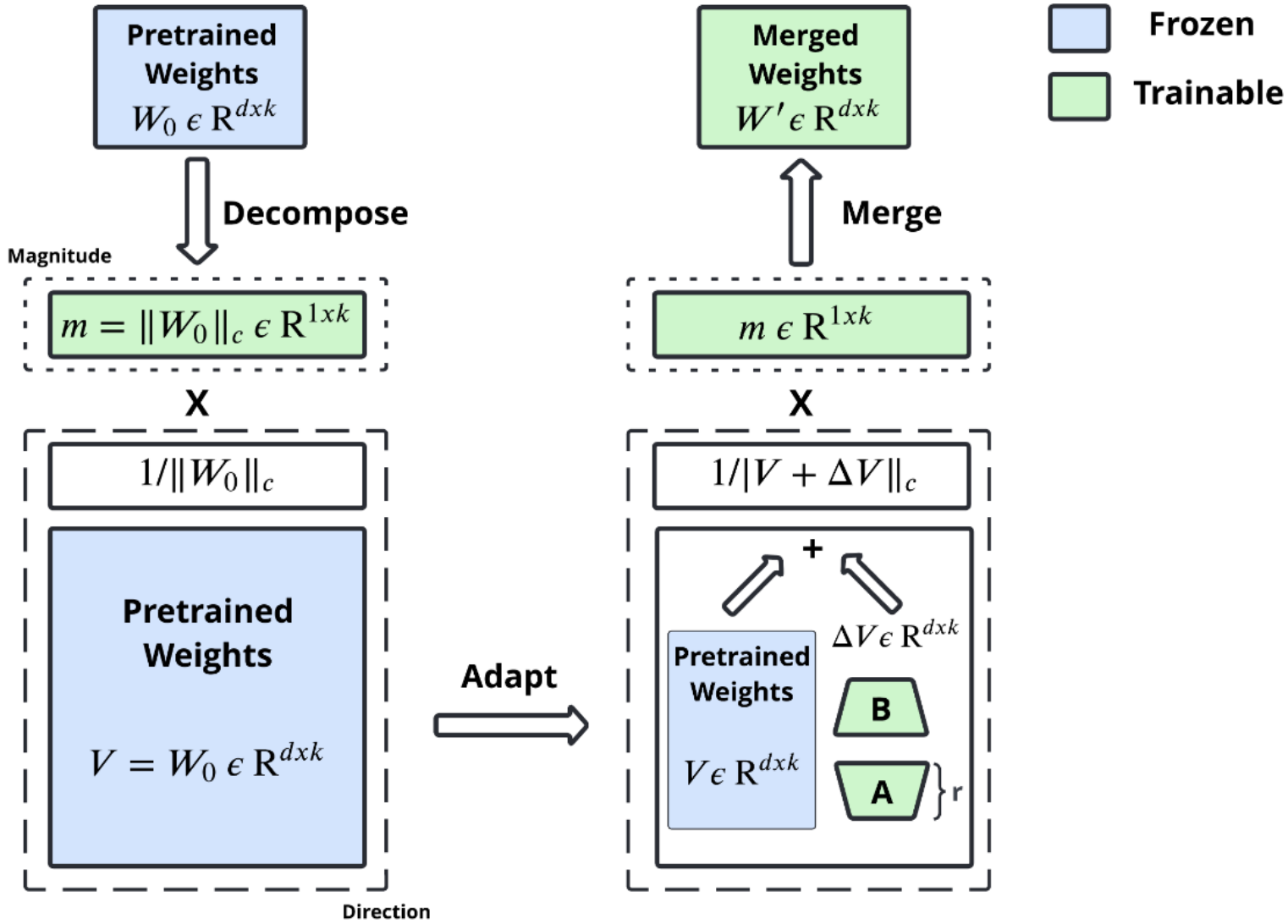




(Another way to Do PEFT

Weight-Decomposed Low-Rank Adaptation (DoRA)

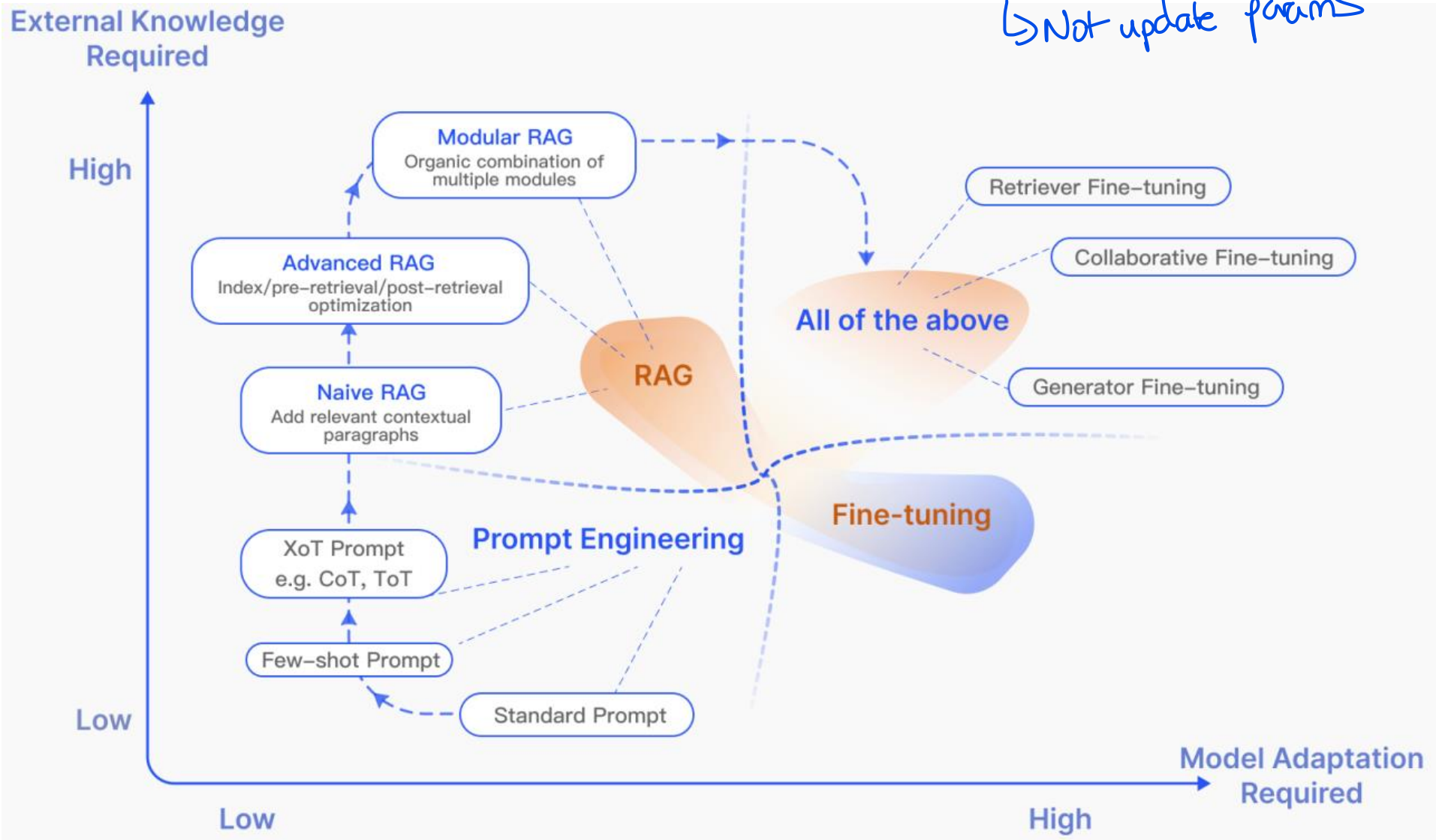
↳ Break ΔW into magnitude and direction



Fine-Tuning with Multiple Adapters

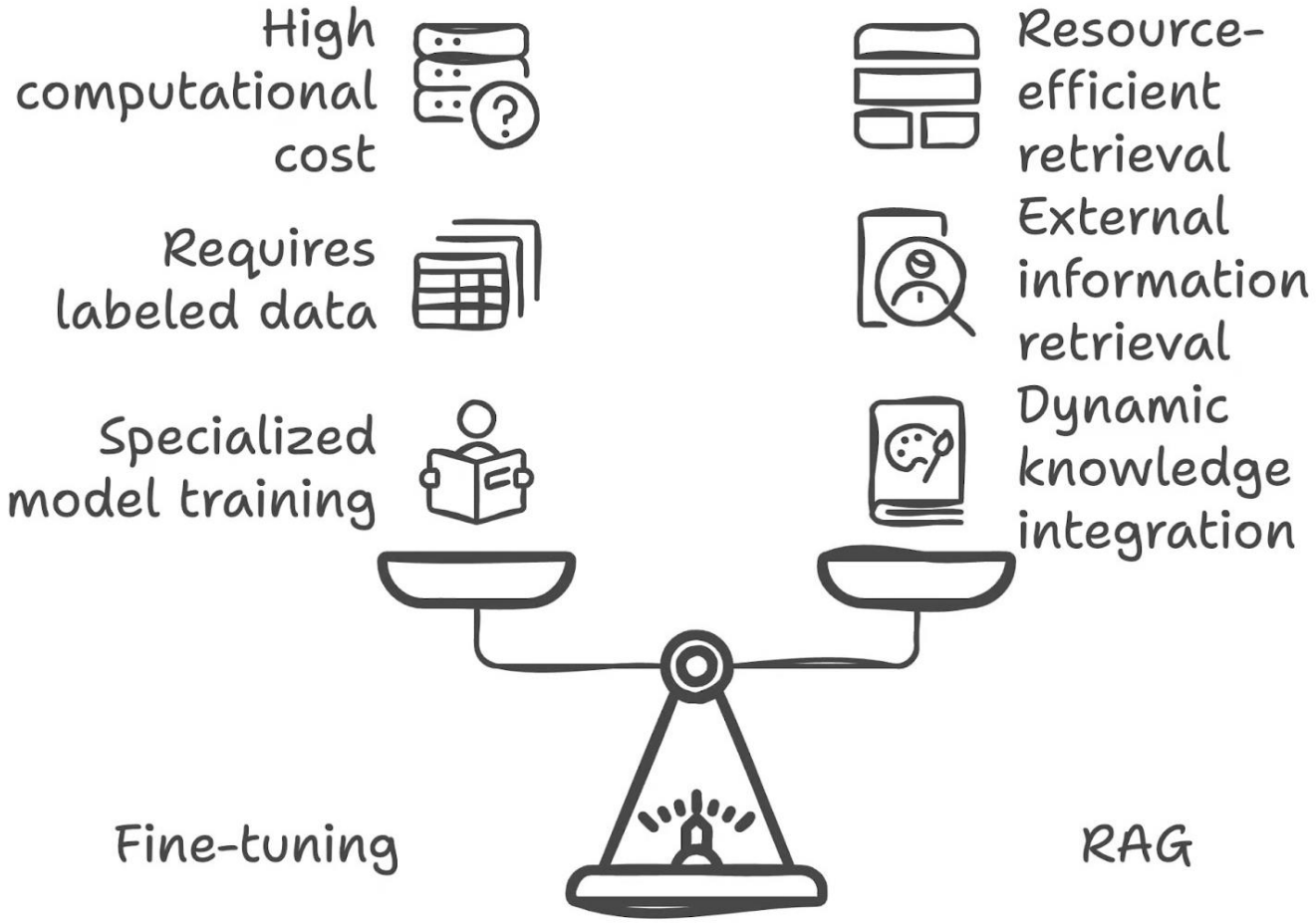
Fine-tuning ≠ vs. RAG

↳ Not update params



Gao et al. Retrieval-augmented generation for large language models: A survey. arXiv, 2023

Fine-tuning vs. RAG



Choose the right approach for your AI task.

https://www.datacamp.com/tutorial/fine-tuning-large-language-models

go thru this, use for projects

Let's Finetune an LLM!!!

https://colab.research.google.com/drive/1IGw93J_PG-uF1yARIW4vylThVby8ReTW?usp=sharing

Next Week!!

- Lecture on the (Un)reliability of reasoning in LLMs