GENERATIVE AI MEET UP

08th July, 2023

Zephania Reuben

AI/ML Specialist

www.nsoma.me | @nsomazr

Bridging the Gap

Adapting LLMs for Local Use Cases

Common LLMs Use Cases and Tasks

Summarisation

Translation

Chatbots

Information Retrieval

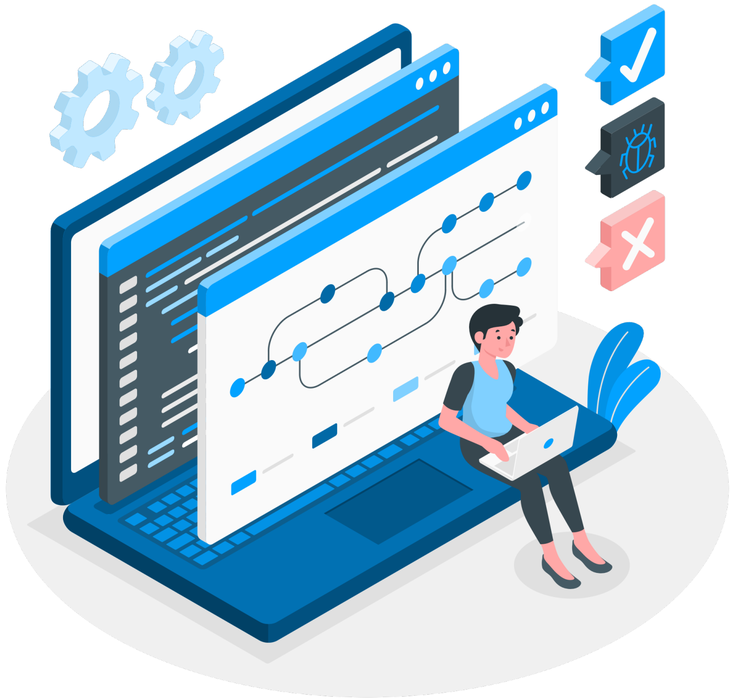
Code Generation

Questions Answering

How Large are LLMs?

What it Takes to Pre-Train LLMs?

#roughly



People(Talents/Labelers)

Data

Computational

Computational Challenge

1 Parameter ~4 Bytes [32 bit float]

1B Parameters ~4 GigaBytes

Only for Model Parameters

175B Parameters ~14000 GigaBytes

(Model parameters, gradients, optimizer states, and temp memory)

Computational Challenge

Single NVIDIA A100 GPU has 80GB Memory Size

Computational Challenge

What is the cost for pre-training GPT-4 with about

1.76 Trillion Parameters *

* source: the-decoder.com

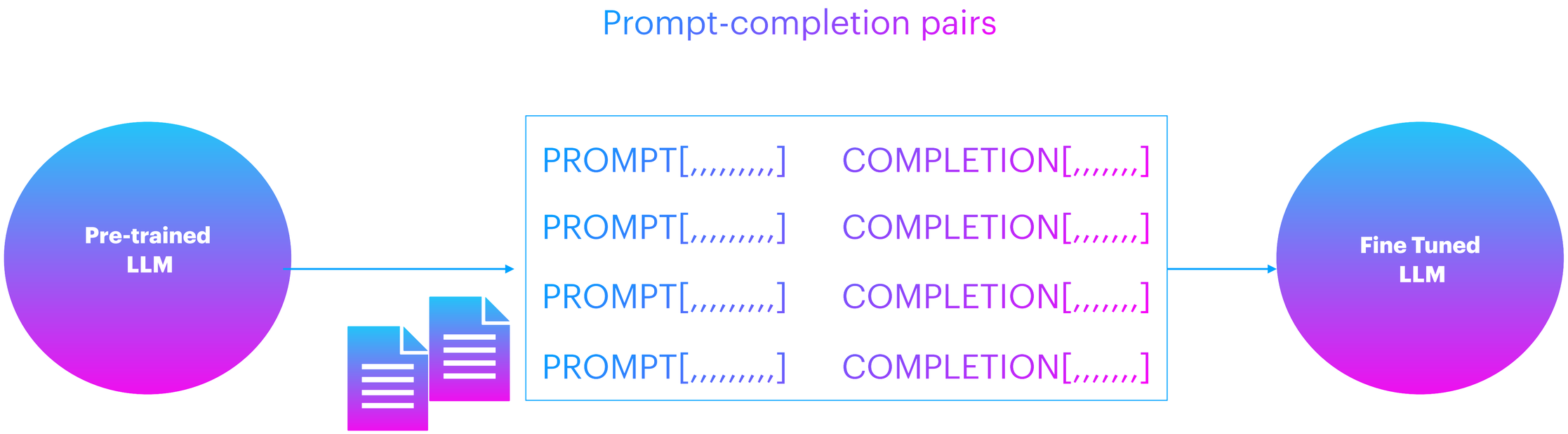
How do we adapt LLMs locally?

Quantization

Multi-GPU Compute Strategies

LLMs Fine Tuning

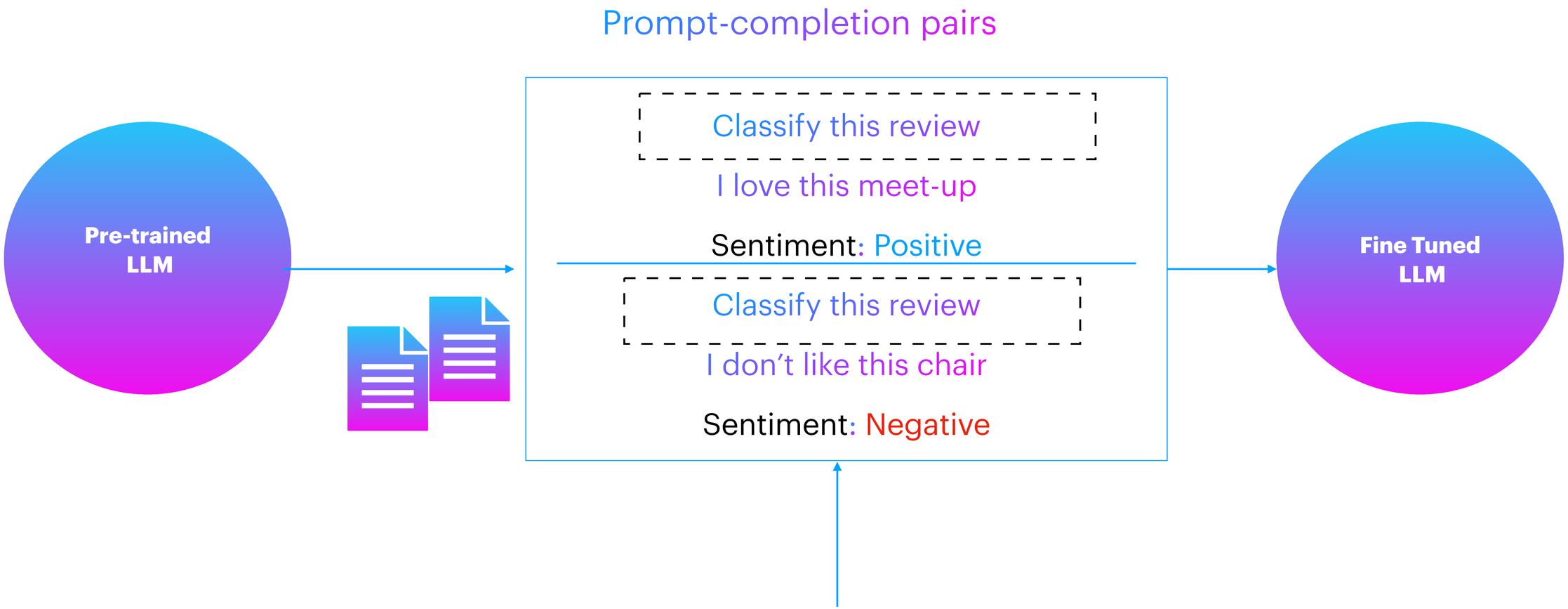
Fine Tuning at High Level



LLM Fine Tuning

Fine Tuning is Supervised Learning Process.

Using Prompts to Fine Tune LLM with Instructions



Sample prompt instruction templates

Classification / sentiment analysis

```
jinja: "Given the following review:\n{{review_body}}\npredict the associated rating\
  \ from the following choices (1 being lowest and 5 being highest)\n- {{ answer_choices\
  \ | join('\\n- ') }} \n|||\n{{answer_choices[star_rating-1]}}"
```

Text generation

```
jinja: Generate a {{star_rating}}-star review (1 being lowest and 5 being highest)
  about this product {{product_title}}.  |||  {{review_body}}
```

Text summarization

```
jinja: "Give a short sentence describing the following product review:\n{{review_body}}\
  \ \n|||\n{{review_headline}}"
```

Source: https://github.com/bigscience-workshop/promptsource/blob/main/promptsource/templates/amazon_polarity/templates.yaml

Using Prompts to Fine Tune LLM with Instructions

* Full fine-tuning updates all parameters

Fine-tuning on a single task

LLM Catastrophic forgetting

How to avoid catastrophic forgetting?

You might not have to!

Fine-tune on **multiple tasks** at the same time

Parameter Efficient Fine-tuning (PEFT)

Parameter Efficient Fine-tuning (PEFT)

Full fine-tuning of large LLMs is challenging

Parameter Efficient Fine-tuning (PEFT)

* Full fine-tuning updates all parameters

Parameter Efficient Fine-tuning (PEFT)

New trainable layer

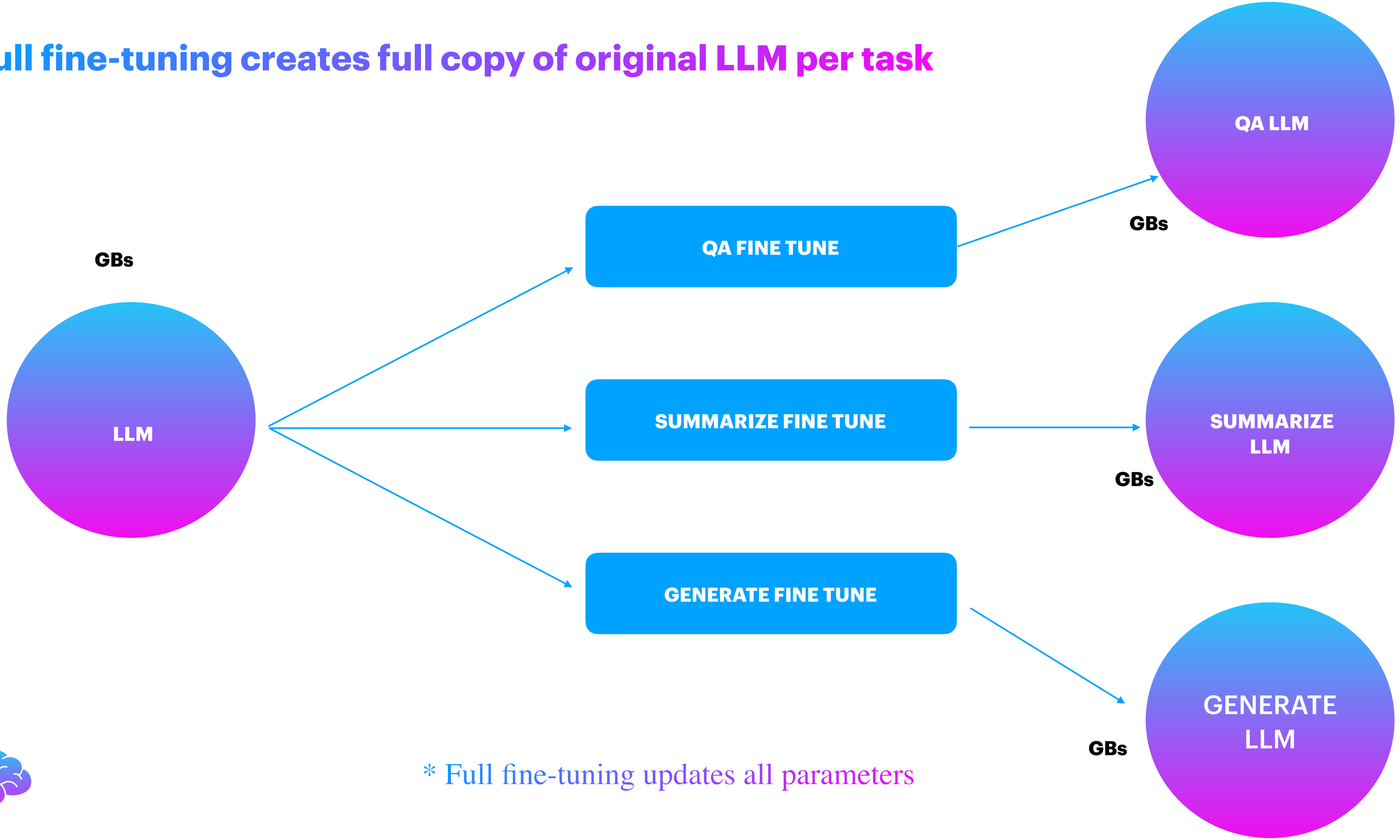
LLM with additional layers for PEFT

Other components

Trainable weights

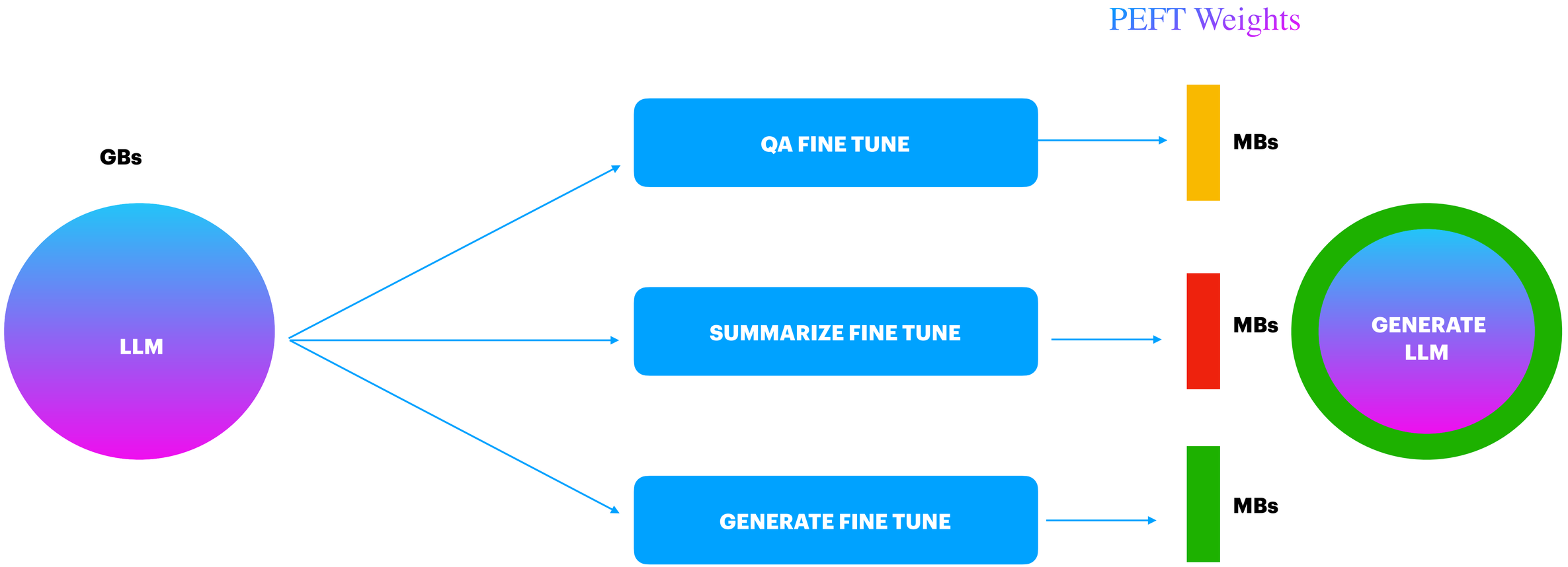
Less prone to catastrophic forgetting

Full fine-tuning creates full copy of original LLM per task



* Full fine-tuning updates all parameters

Using PEFT fine-tuning saves space and is flexible



PEFT Methods

Selective

Additive

Re-parameterization

Re-parameterization with Low Rank Representation

Low-Rank Adaptation of Large Language Models (LoRA)

LoRA: Low Rank Adaption of LLMs

Freeze most of the original LLM weights.

Inject 2 rank decomposition matrices

Train the weights of the smaller matrices

LoRA: Low Rank Adaption of LLMs

Train different rank decomposition matrices for different tasks

Update weights before inference

Inference Weight: Original Weights + LoRA Weights

Hands-on

Fine Tuning Falcon LLM with LoRA

https://lightning.ai/pages/community/finetuning-falcon-efficiently/

What About Human Values?

*Solving Alignment Problem with RLHF

* Out of this presentation scope

Falcon, Alpaca, Vicuna, Llama, and all the variants: quantised, mixed precision, half precision, etc

How to Run LLMs Locally

https://wandb.ai/capecape/LLMs/reports/How-to-Run-LLMs-Locally--Vmlldzo0Njg5NzMx

What are LLMs Local Use Cases?

Local Use Cases of LLMs

Question Answering

What are the limitations?

Thank You