[COMSE6998-015] Fall 2024 Introduction to Deep Learning and LLM based Generative AI Systems

Lecture 11

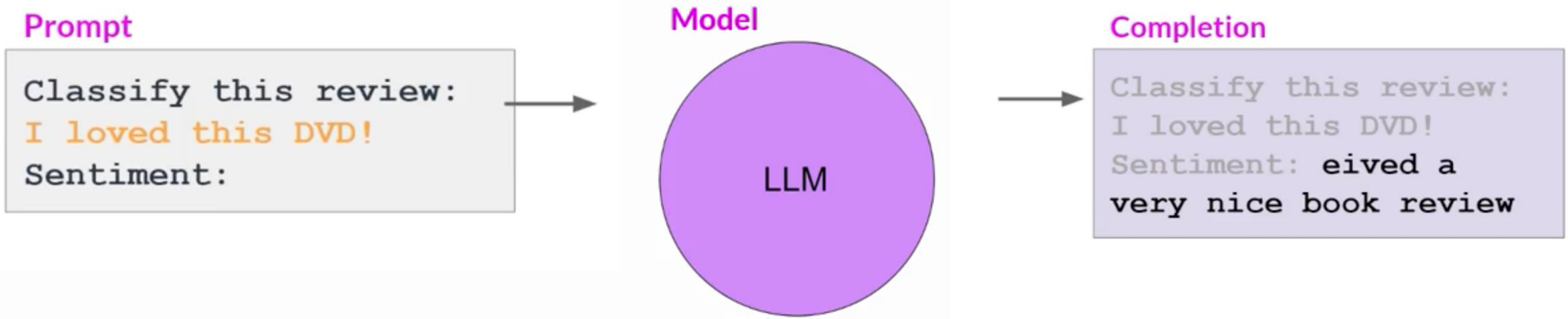
Parijat Dube, Chen Wang

Agenda

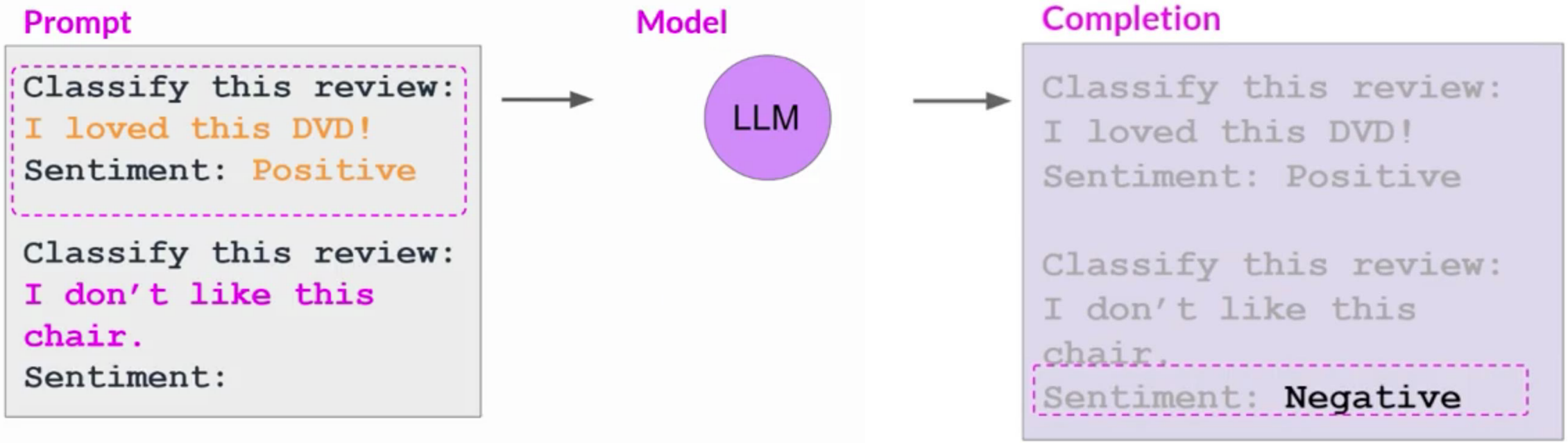
- Instruction Finetuning
- Single Task Fine Tuning
- Multi-task, Instruction Finetuning
- Parameter Efficient Finetuning (PEFT)
- Low-Rank Adaptation of Large Language Models (LoRA)
- Soft Prompt Tuning

Finetuning an LLM with instruction prompts

In-context learning (ICL) - zero shot inference



In-context learning (ICL) - one/few shot inference



One-shot or Few-shot Inference

Limitations of in-context learning

```
Classify this review:
I loved this movie!
Sentiment: Positive

Classify this review:
I don't like this chair.
Sentiment: Negative

Classify this review:
This sofa is so ugly.
Sentiment: Negative

Classify this review:
Who would use this product?
Sentiment:
```

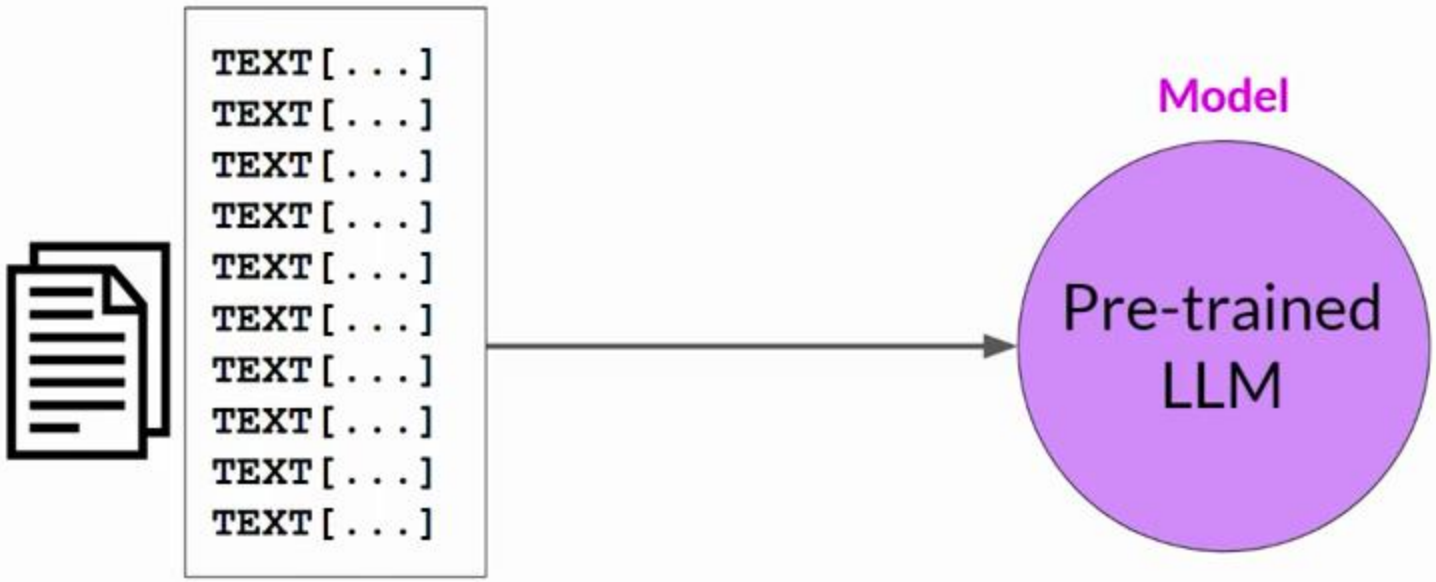
Even with multiple examples

- In-context learning may not work for smaller models LLM
- Examples take up space in the context window

Instead, try **fine-tuning** the model

LLM fine-tuning at a high level

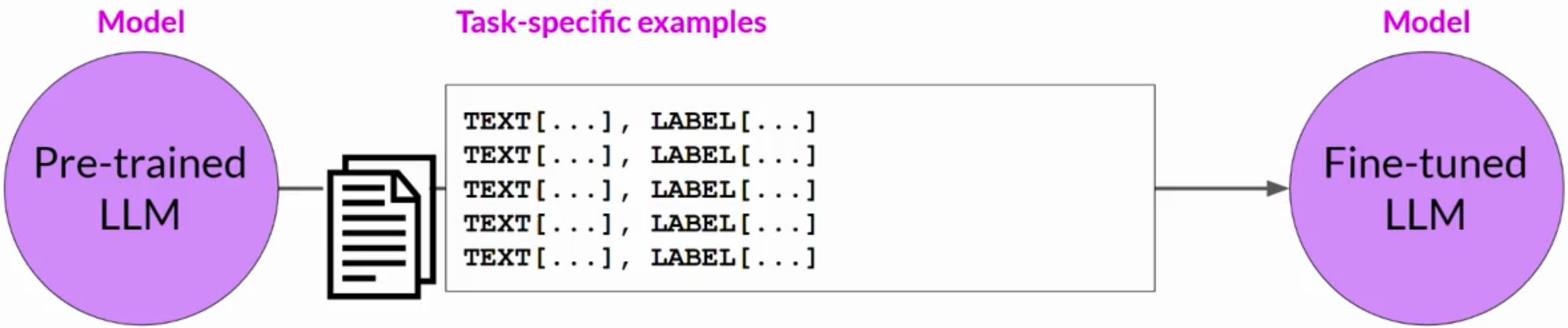
LLM pre-training



GB - TB - PB
of unstructured textual data

LLM fine-tuning at a high level

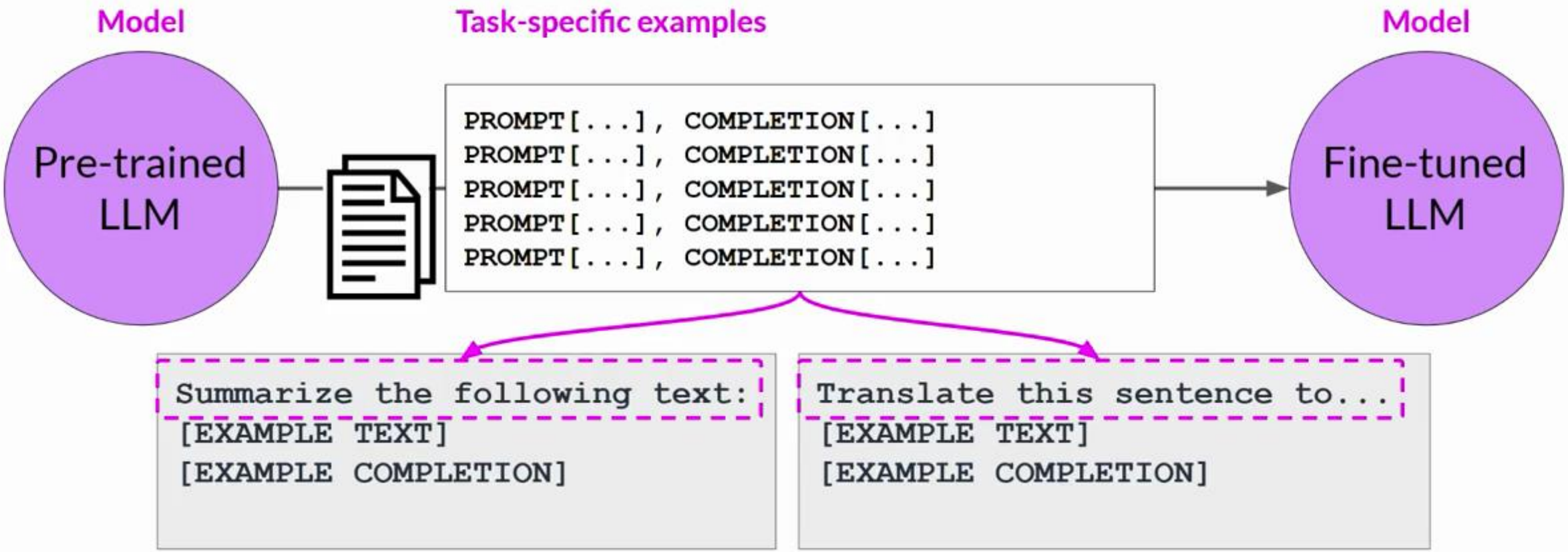
LLM fine-tuning



GB - TB
of labeled examples for a specific task or set of tasks

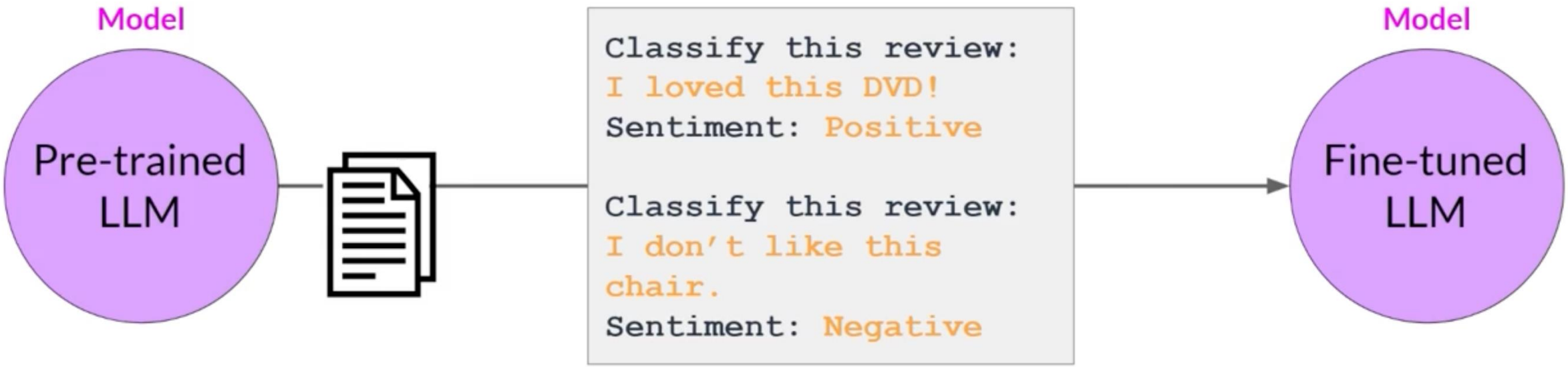
Using prompts to fine-tune LLMs with instruction

LLM fine-tuning



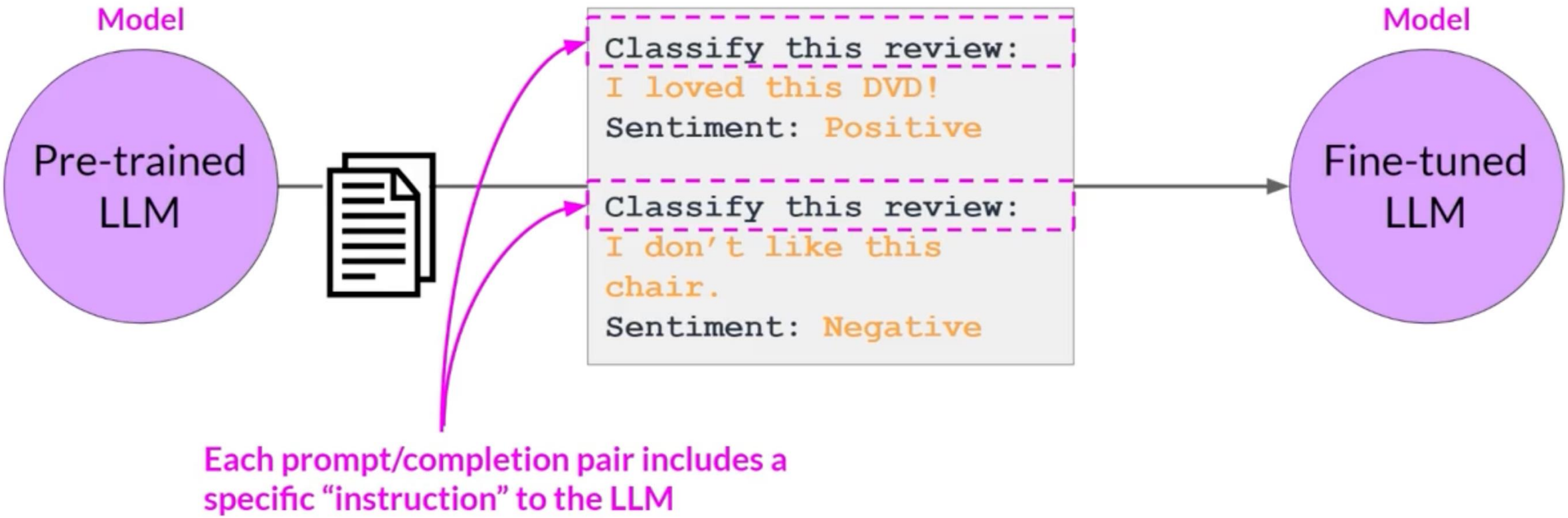
Using prompts to fine-tune LLMs with instruction

LLM fine-tuning



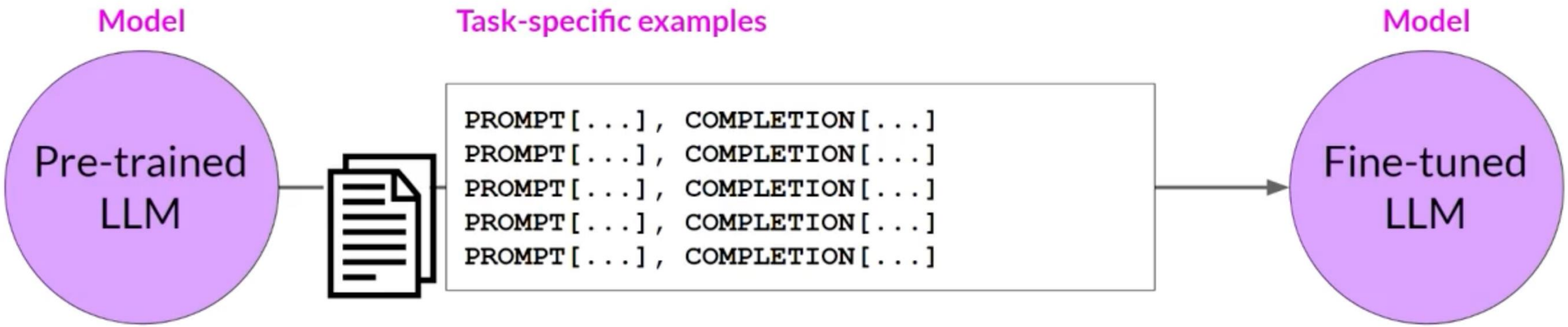
Using prompts to fine-tune LLMs with instruction

LLM fine-tuning



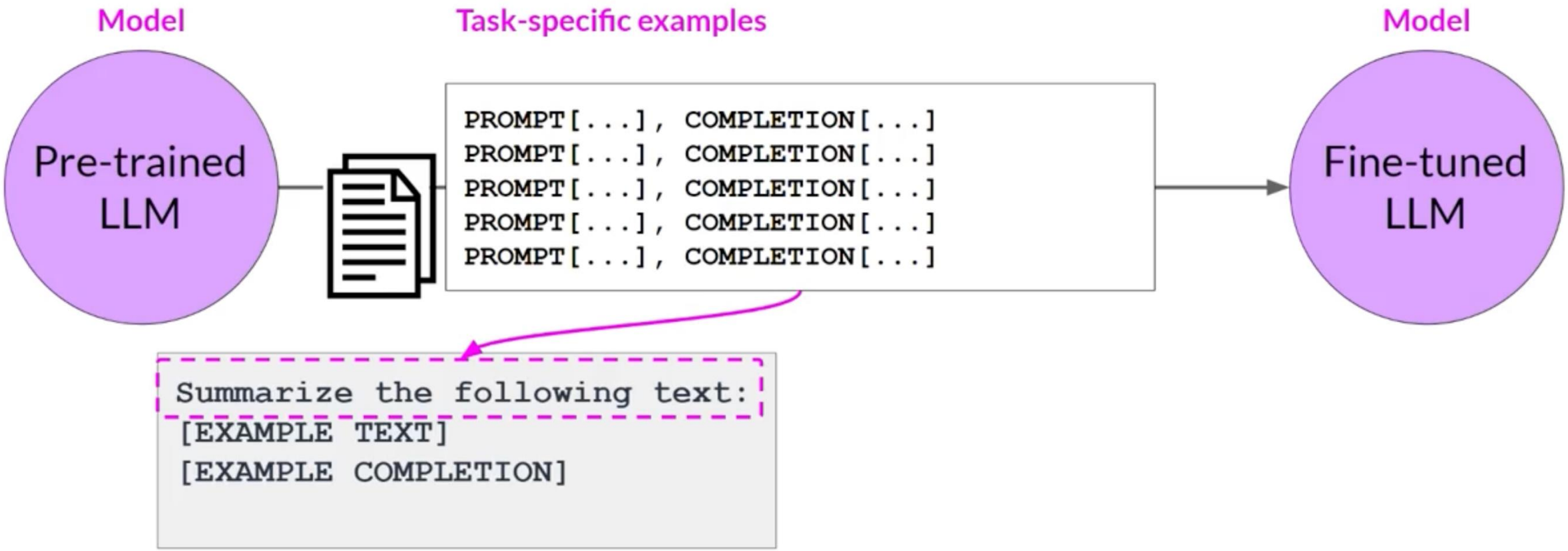
Using prompts to fine-tune LLMs with instruction

LLM fine-tuning



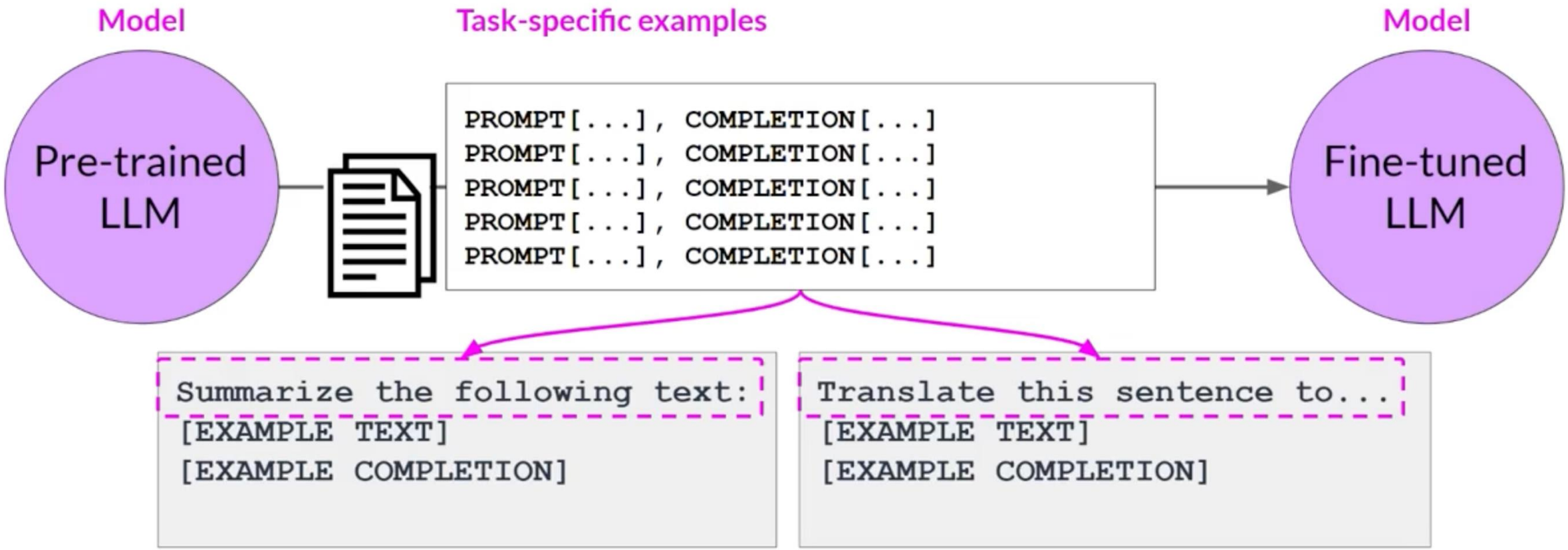
Using prompts to fine-tune LLMs with instruction

LLM fine-tuning



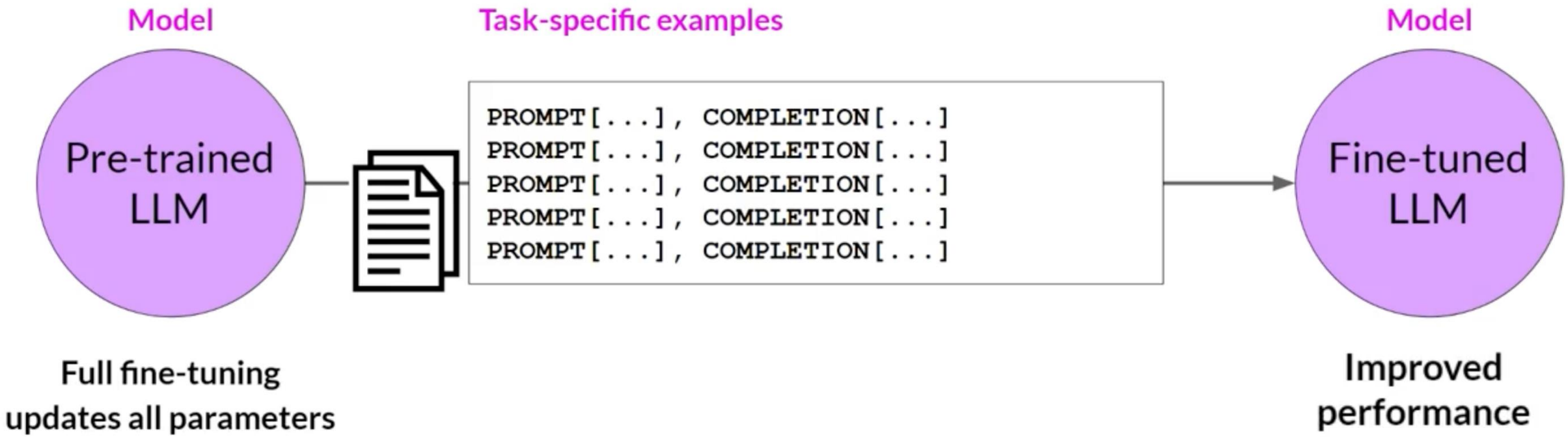
Using prompts to fine-tune LLMs with instruction

LLM fine-tuning



Using prompts to fine-tune LLMs with instruction

LLM fine-tuning



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

LLM fine-tuning process

LLM fine-tuning

Prepared instruction dataset

Training splits

```
PROMPT[...], COMPLETION[...]
PROMPT[...], COMPLETION[...]
PROMPT[...], COMPLETION[...]
PROMPT[...], COMPLETION[...]
PROMPT[...], COMPLETION[...]
```

Training

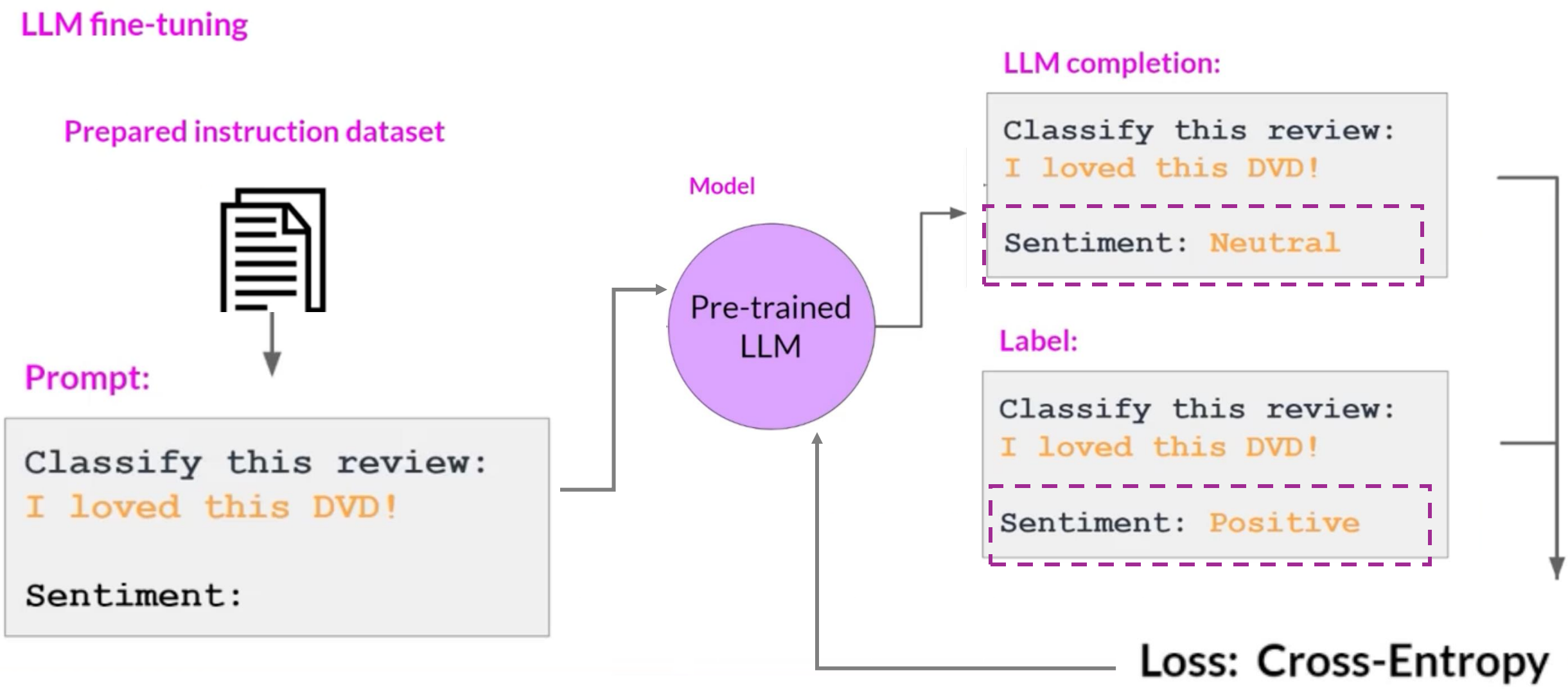
```
PROMPT[...], COMPLETION[...]
...
```

Validation

```
PROMPT[...], COMPLETION[...]
...
```

Test

LLM fine-tuning process



LLM fine-tuning process

LLM fine-tuning

Prepared instruction dataset

Training splits

```
PROMPT[...], COMPLETION[...]
PROMPT[...], COMPLETION[...]
PROMPT[...], COMPLETION[...]
PROMPT[...], COMPLETION[...]
PROMPT[...], COMPLETION[...]
```

Training

```
PROMPT[...], COMPLETION[...]
...
```

Validation

validation_accuracy

```
PROMPT[...], COMPLETION[...]
...
```

Test

LLM fine-tuning process

LLM fine-tuning

Prepared instruction dataset

Training splits

```
PROMPT[...], COMPLETION[...]
PROMPT[...], COMPLETION[...]
PROMPT[...], COMPLETION[...]
PROMPT[...], COMPLETION[...]
PROMPT[...], COMPLETION[...]
```

Training

```
PROMPT[...], COMPLETION[...]
...
```

Validation

```
PROMPT[...], COMPLETION[...]
...
```

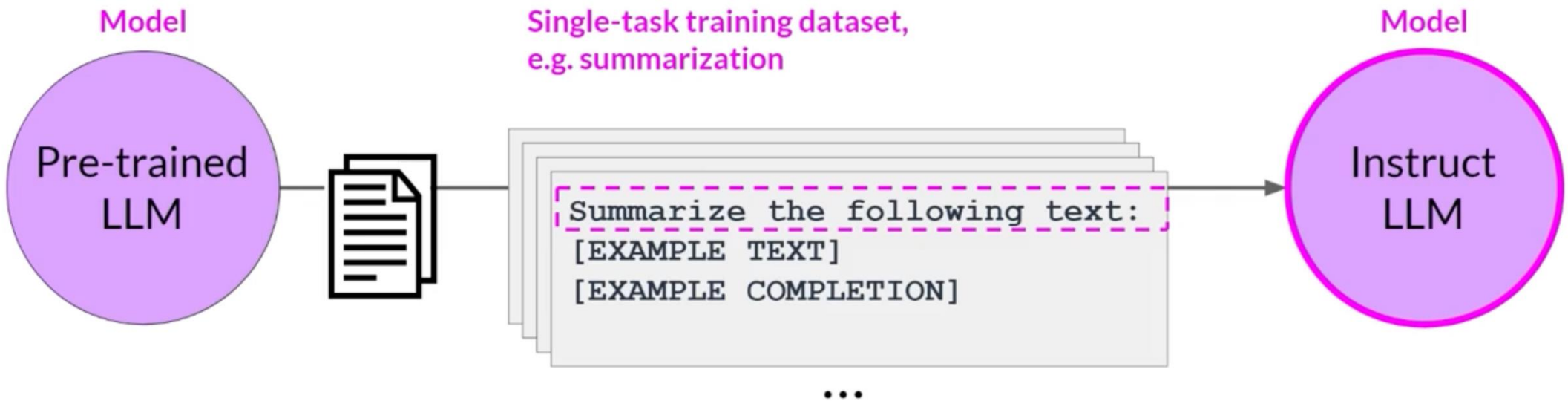
Test

test_accuracy

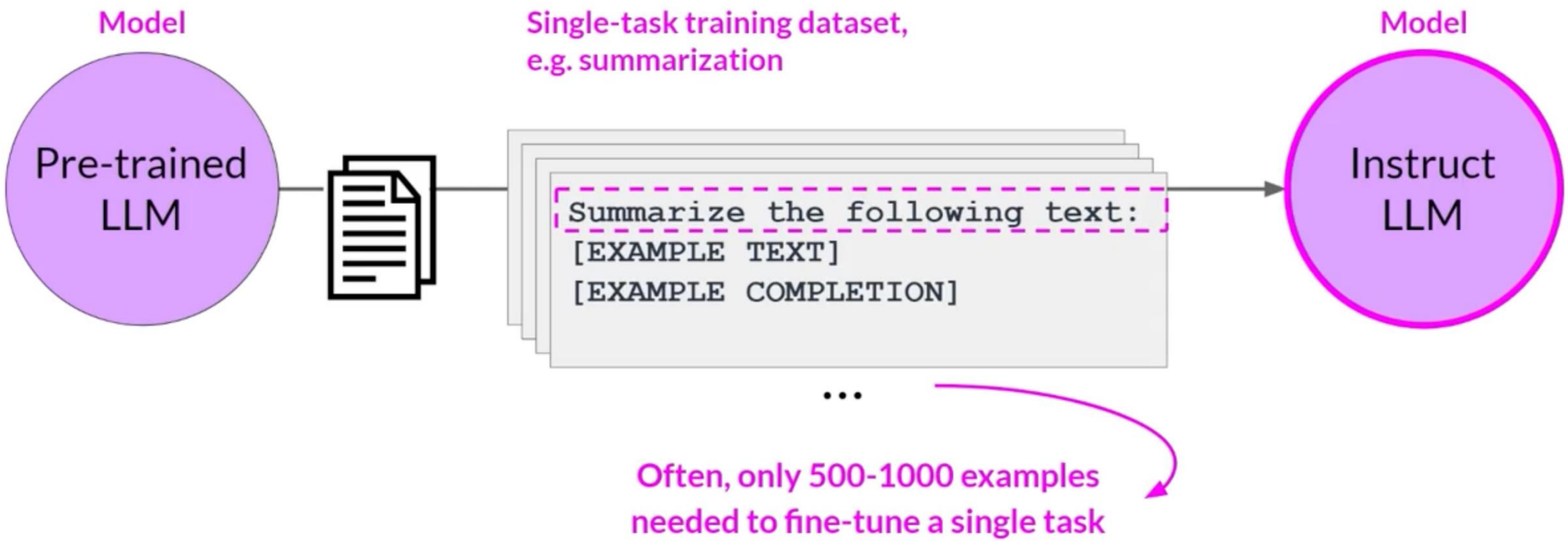
LLM fine-tuning process

LLM fine-tuning process

Fine-tuning on a single task

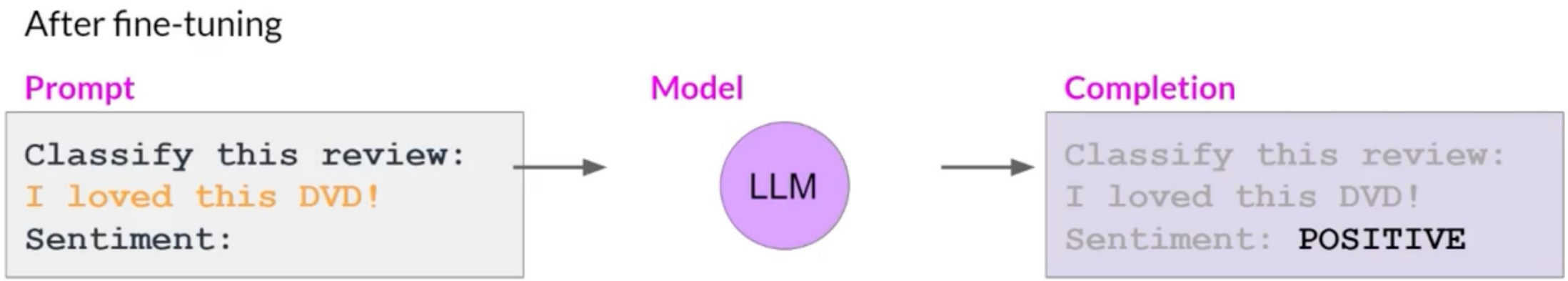


Fine-tuning on a single task



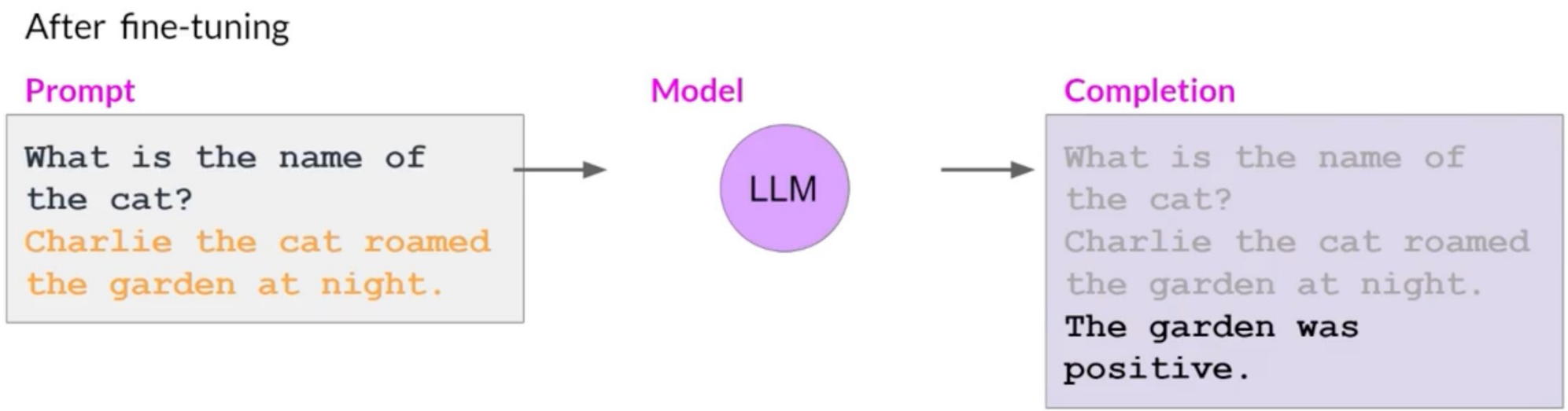
Catastrophic forgetting

- Fine-tuning can significantly increase the performance of a model on a specific task…



Catastrophic forgetting

- …but can lead to reduction in ability on other tasks

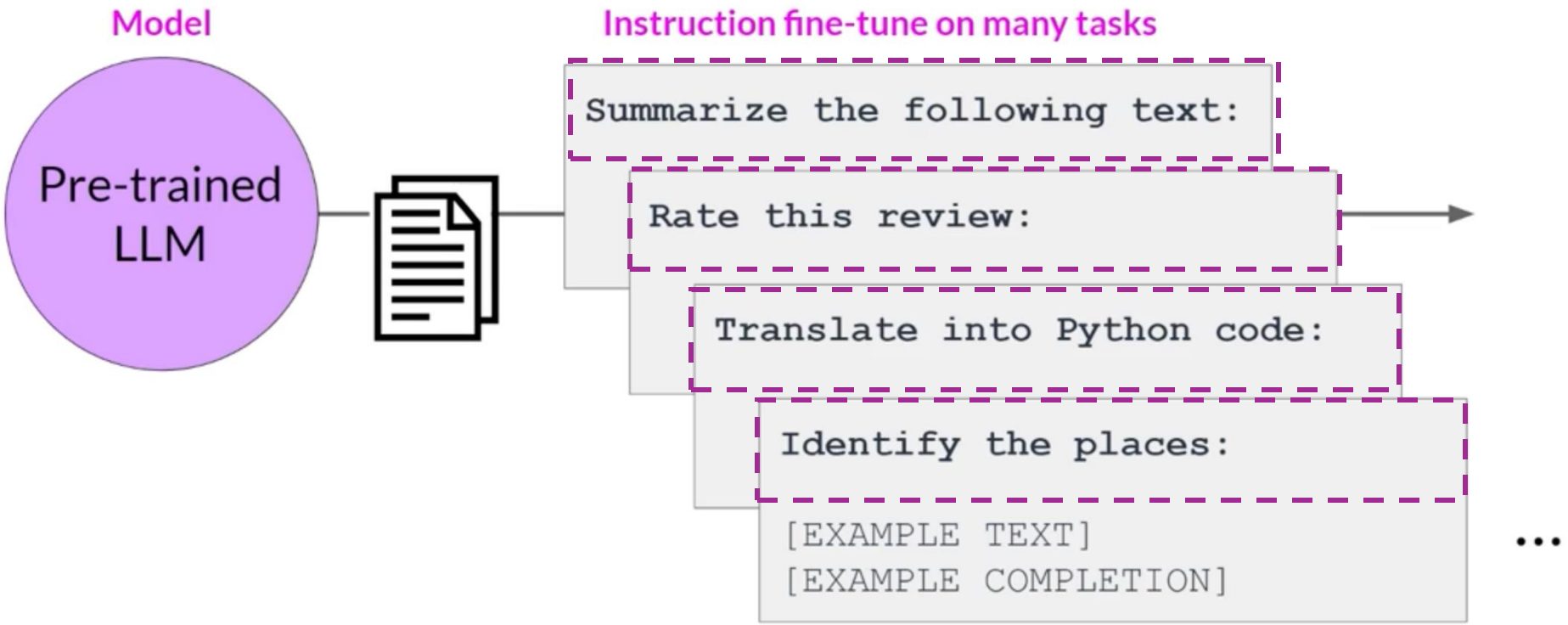


How to avoid Catastrophic forgetting?

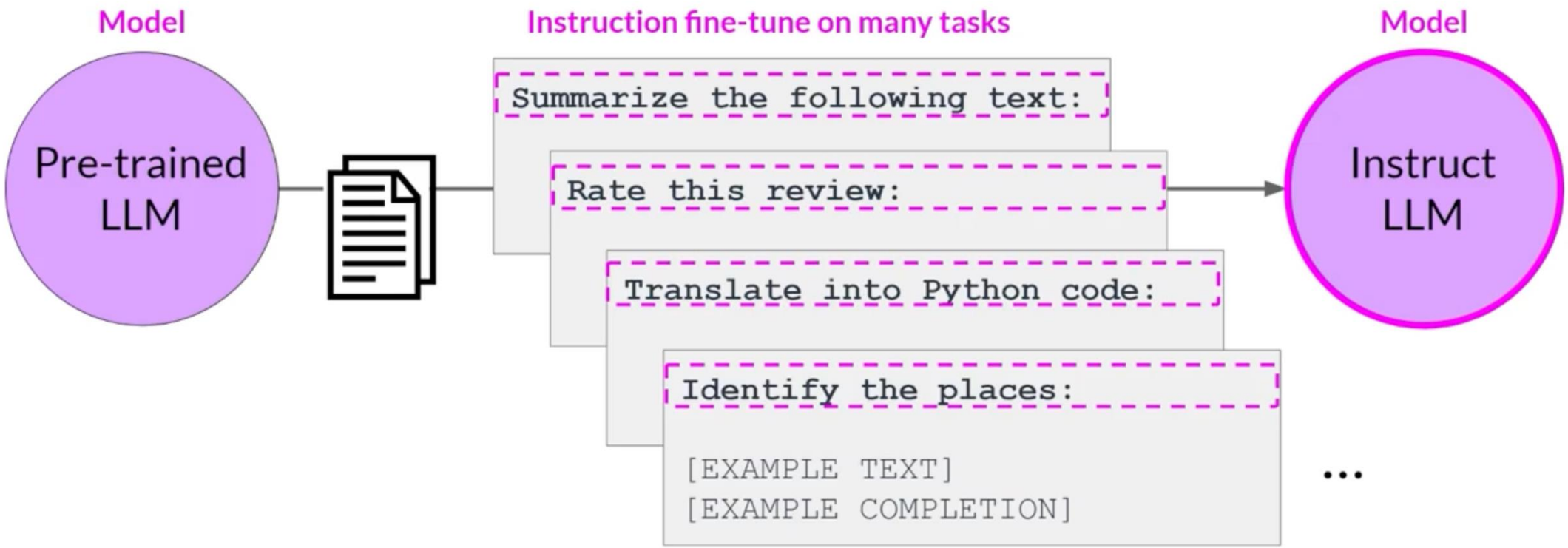
- First note that you may not have to!
- Fine tune on **multiple-tasks** at the same time.
- Consider **Parameter Efficient Fine Tuning** (PEFT)

Multi-task, Instruction Finetuning

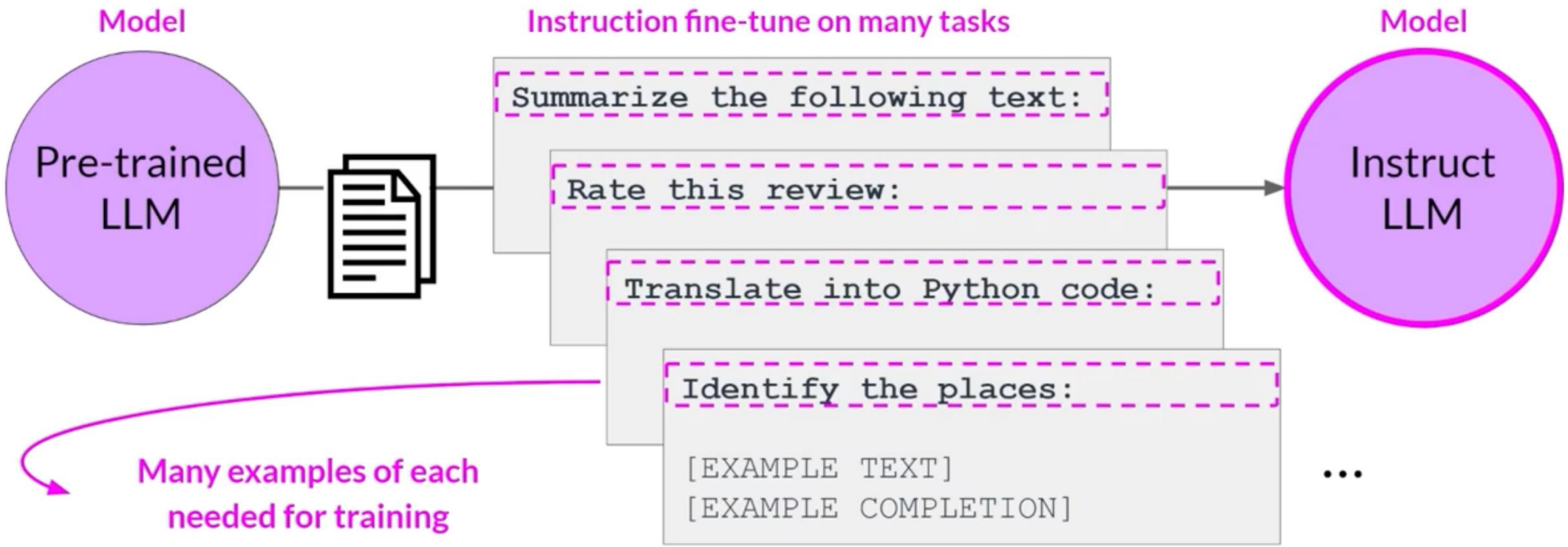
Multi-task, instruction fine-tuning



Multi-task, instruction fine-tuning



Multi-task, instruction fine-tuning



Instruction fine-tuning with FLAN

- FLAN models refer to a specific set of instructions used to perform instruction fine-tuning

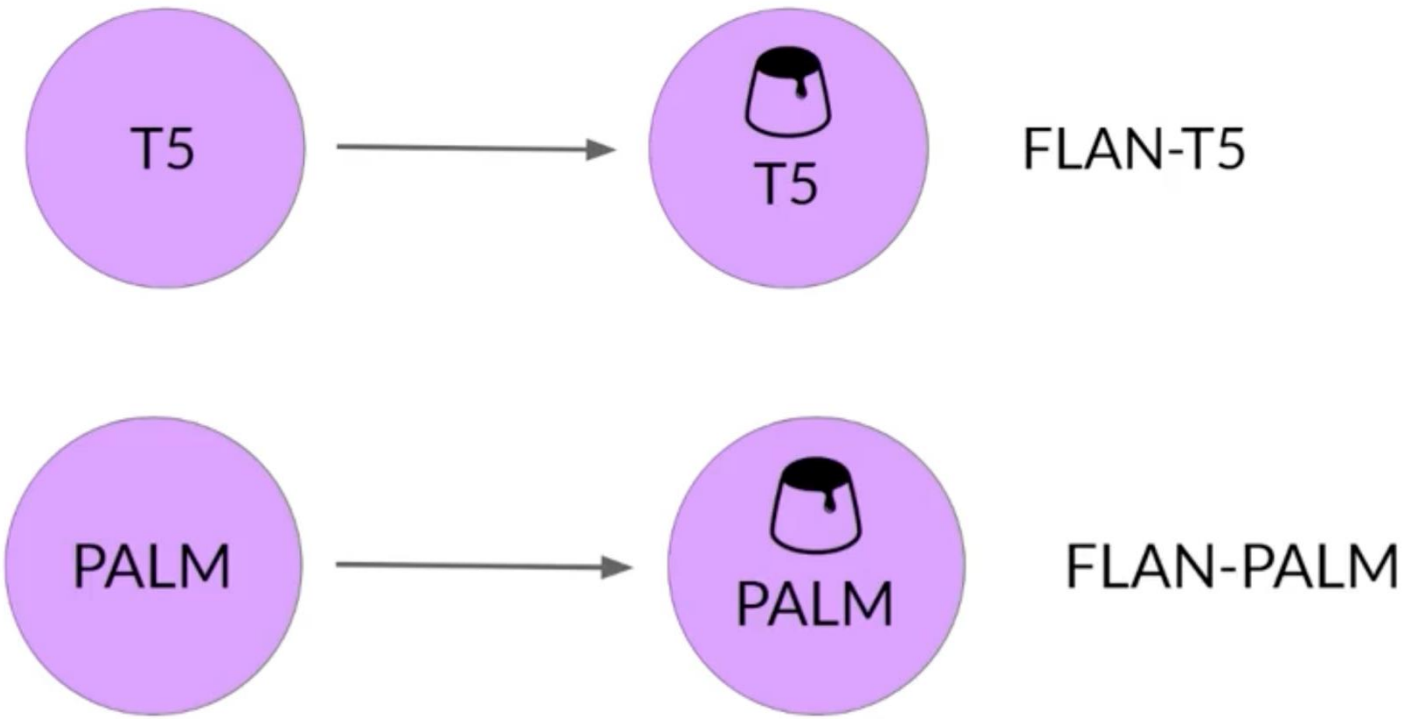
FLAN

"The metaphorical dessert to the main course of pretraining"

Instruction fine-tuning with FLAN

- FLAN models refer to a specific set of instructions used to perform instruction fine-tuning

FLAN
(**F**ine-tuned
LAnguage **N**et)



FLAN-T5: Fine-tuned version of pre-trained T5 model

- FLAN-T5 is a great, general purpose, instruct model

Source: Chung et al. 2022, “Scaling Instruction-Finetuned Language Models”

FLAN-T5: Fine-tuned version of pre-trained T5 model

- FLAN-T5 is a great, general purpose, instruct model

T0-SF

- Commonsense Reasoning,
- Question Generation,
- Closed-book QA,
- Adversarial QA,
- Extractive QA

...

55 Datasets
14 Categories
193 Tasks

Muffin

- Natural language inference,
- Code instruction gen,
- Code repair
- Dialog context generation,
- Summarization (SAMSum)

...

69 Datasets
27 Categories
80 Tasks

CoT (reasoning)

- Arithmetic reasoning,
- Commonsense reasoning
- Explanation generation,
- Sentence composition,
- Implicit reasoning,

...

9 Datasets
1 Category
9 Tasks

Natural Instructions

- Cause effect classification,
- Commonsense reasoning,
- Named Entity Recognition,
- Toxic Language Detection,
- Question answering

...

372 Datasets
108 Categories
1554 Tasks

Source: Chung et al. 2022, "Scaling Instruction-Finetuned Language Models"

SAMSum: A dialogue dataset

Sample prompt training dataset (**samsum**) to fine-tune FLAN-T5 from pretrained T5

Datasets: **samsum** | Tasks: Summarization | Languages: English

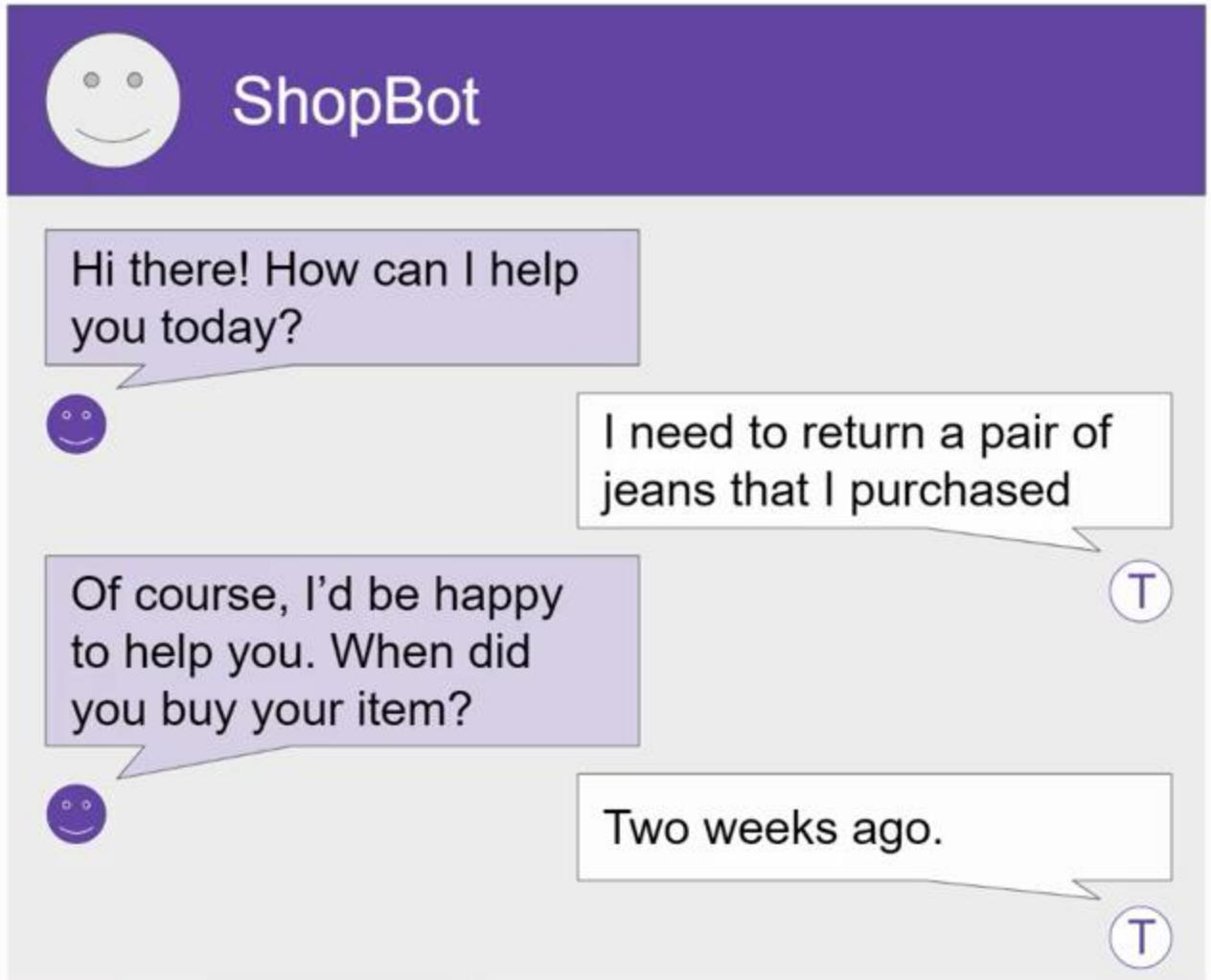
dialogue (string)	summary (string)
"Amanda: I baked cookies. Do you want some? Jerry: Sure! Amanda: I'll bring you tomorrow :-)"	"Amanda baked cookies and will bring Jerry some tomorrow."
"Olivia: Who are you voting for in this election? Oliver: Liberals as always. Olivia: Me too!! Oliver: Great"	"Olivia and Olivier are voting for liberals in this election. "
"Tim: Hi, what's up? Kim: Bad mood tbh, I was going to do lots of stuff but ended up procrastinating Tim: What did…	"Kim may try the pomodoro technique recommended by Tim to get more stuff done."

Source: https://huggingface.co/datasets/samsum, https://github.com/google-research/FLAN/blob/2c79a31/flan/v2/templates.py#L3285

Sample FLAN-T5 prompt templates

```
"samsum": [
      ("{dialogue}\n\Briefly summarize that dialogue.", "{summary}"),
       ("Here is a dialogue:\n{dialogue}\n\nWrite a short summary!",
        "{summary}"),
       ("Dialogue:\n{dialogue}\n\nWhat is a summary of this dialogue?",
        "{summary}"),
       ("{dialogue}\n\nWhat was that dialogue about, in two sentences or less?",
        "{summary}"),
       ("Here is a dialogue:\n{dialogue}\n\nWhat were they talking about?",
        "{summary}"),
       ("Dialogue:\n{dialogue}\nWhat were the main points in that "
        "conversation?", "{summary}"),
       ("Dialogue:\n{dialogue}\nWhat was going on in that conversation?",
        "{summary}"),
]
```

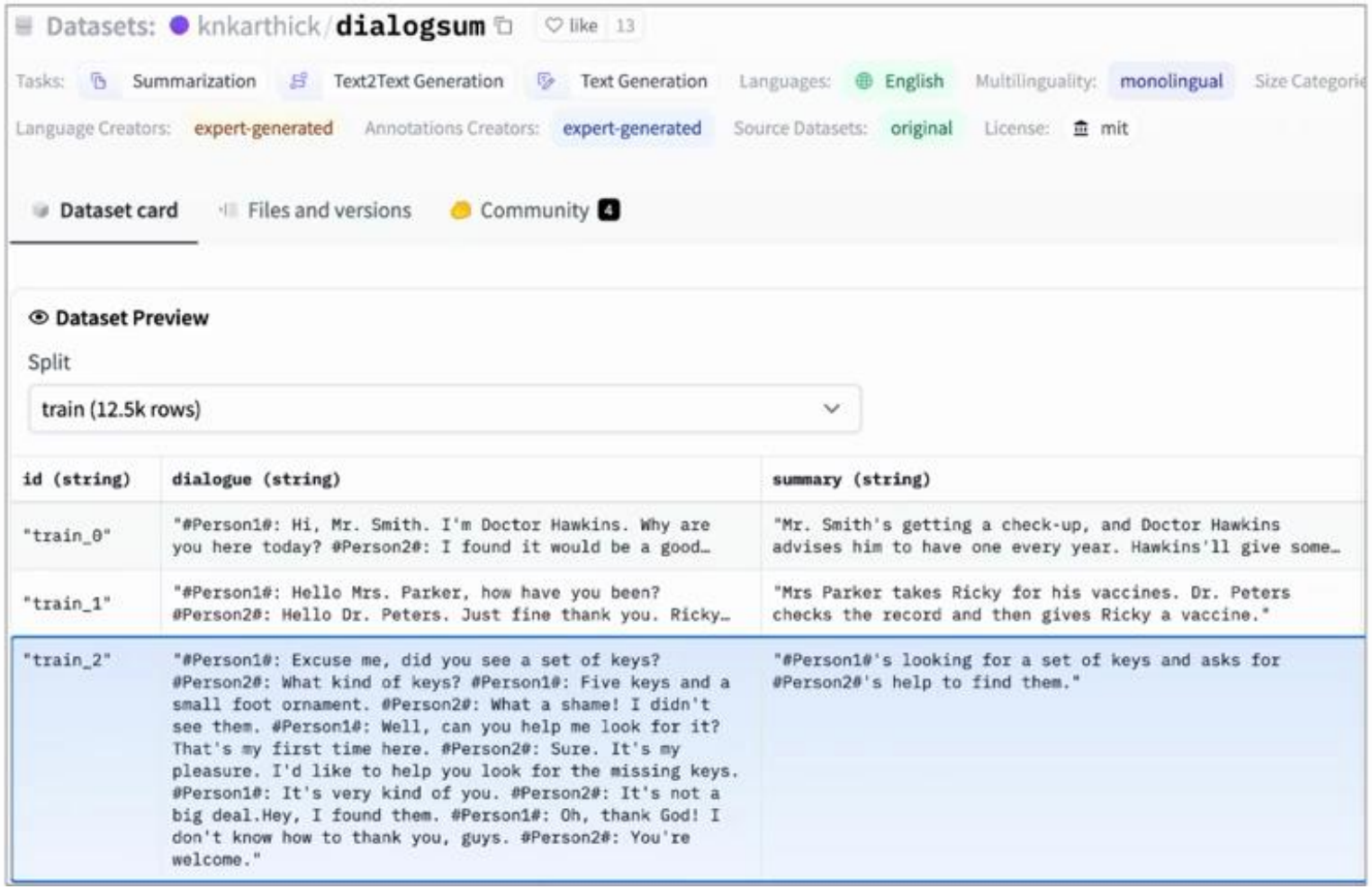
Improving FLAN-T5's summarization capabilities



Goal: Summarize conversations to identify actions to take

Improving FLAN-T5's summarization capabilities

Further fine-tune FLAN-T5 with a domain-specific instruction dataset (**dialogsum**)



Example support-dialog summarization

Prompt (created from template)

```
Summarize the following conversation.
Tommy: Hello. My name is Tommy Sandals, I have a reservati
Mike: May I see some identification, sir, please?
Tommy: Sure. Here you go.
Mike: Thank you so much. Have you got a credit card, Mr.
Sandals?
Tommy: I sure do.
Mike: Thank you, sir. You'll be in room 507, nonsmoking,
queen bed.
Tommy: That's great, thank you!
Mike: Enjoy your stay!
```

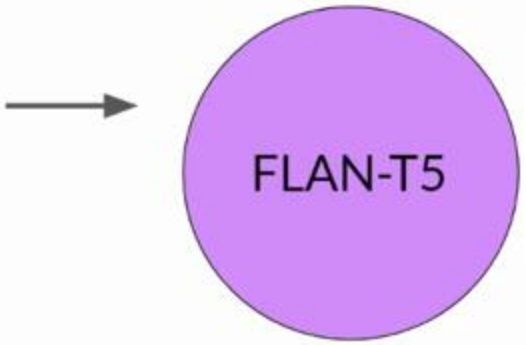
Source: https://huggingface.co/datasets/knkarthick/dialogsum/viewer/knkarthick--dialo

Summary **before** fine-tuning FLAN-T5 with our dataset

Prompt (created from template)

```
Summarize the following
conversation.
Tommy: Hello. My name is
Tommy Sandals, I have a
reservation.
Mike: May I see some
...
...
...
Tommy: That's great, thank
you!
Mike: Enjoy your stay!
```

Model



Completion (Summary)

```
Tommy Sandals has a reservation
for a room at the Venetian
Hotel in Las Vegas.
```

Adequate completion, but does not match human baseline.

Human baseline summary:

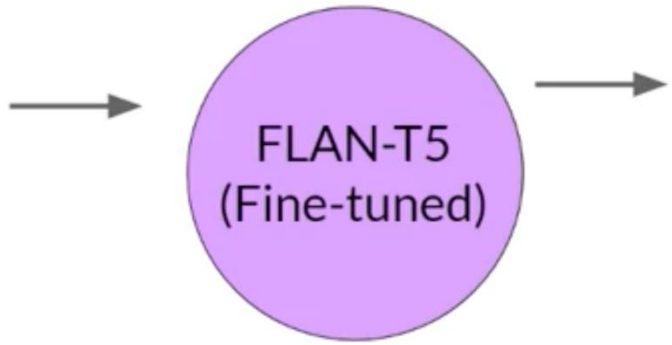
```
Tommy Sandals has got a
reservation. Mike asks for his
identification and credit card
and helps his check-in.
```

Summary **after** fine-tuning FLAN-T5 with our dataset

Prompt (created from template)

```
Summarize the following
conversation.
Tommy: Hello. My name is
Tommy Sandals, I have a
reservation.
Mike: May I see some
...
...
...
Tommy: That's great, thank
you!
Mike: Enjoy your stay!
```

Model

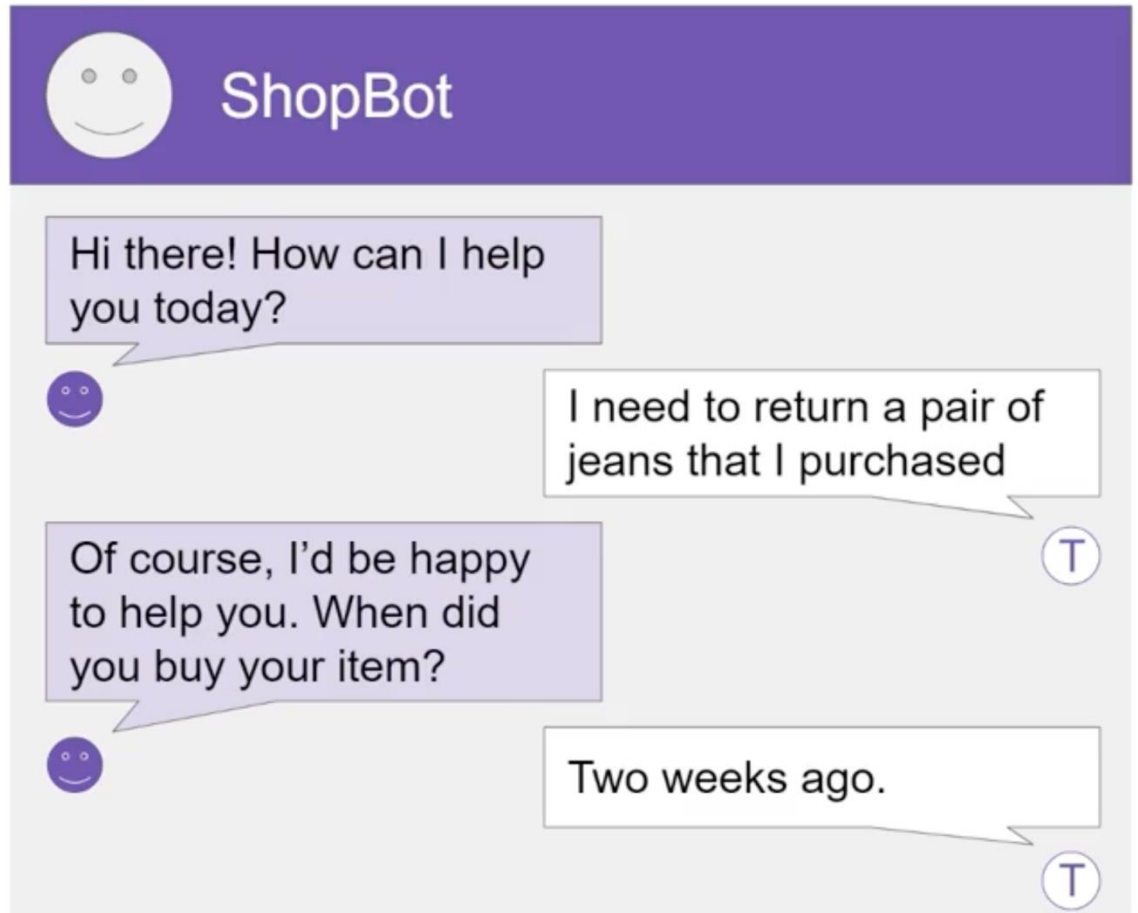


Completion (Summary)

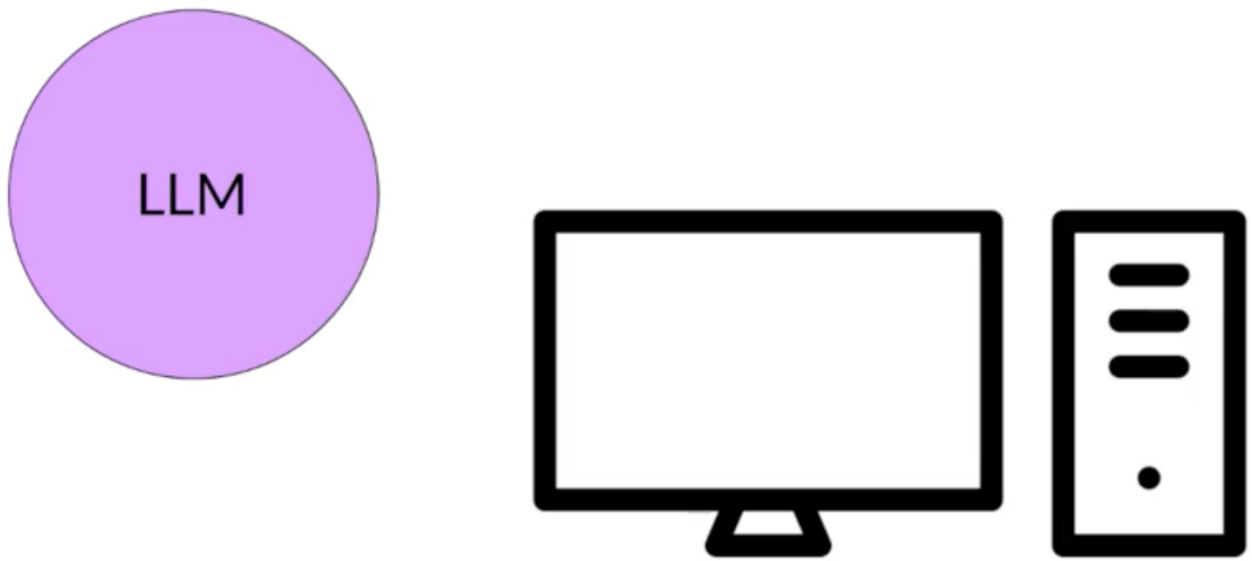
```
Tommy Sandals has a
reservation and checks in
showing his ID and credit
card. Mike helps him to
check in and approves his
reservation.
```

Better summary, more-closely matches human baseline.

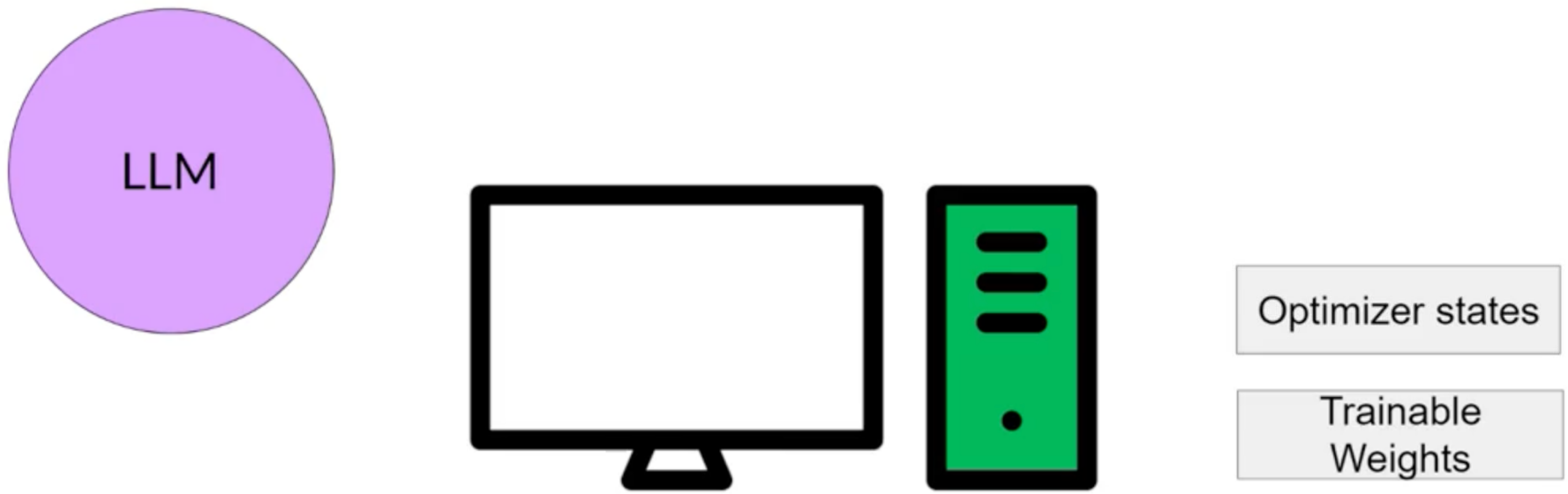
Fine-tuning with your own data



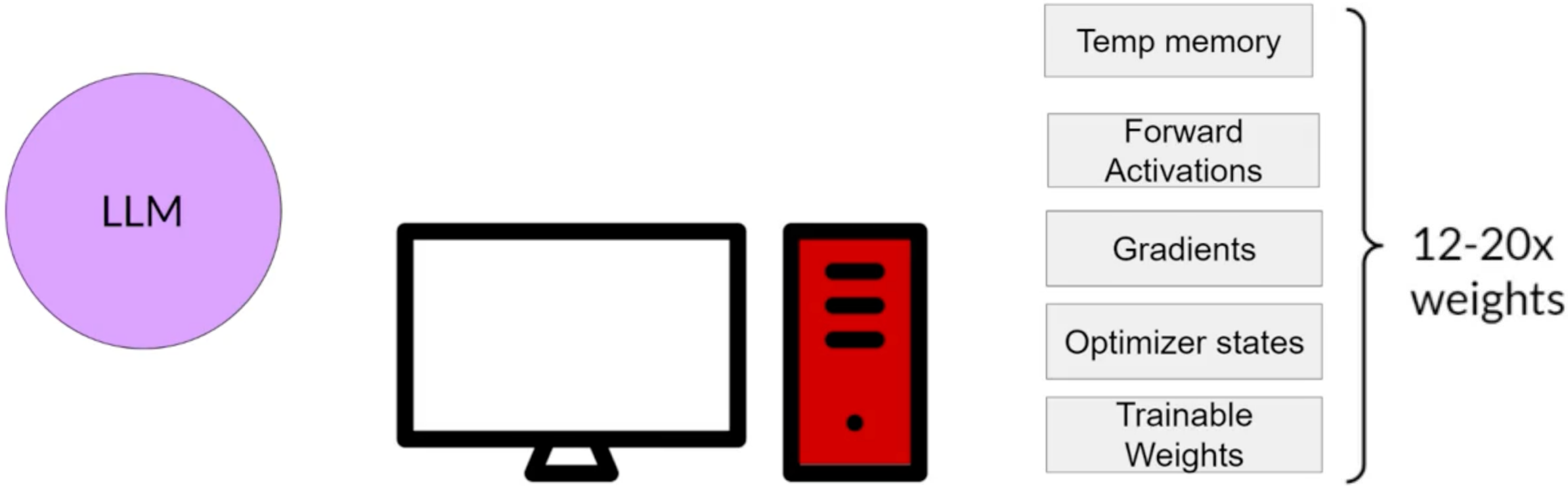
Full fine-tuning of large LLMs is challenging



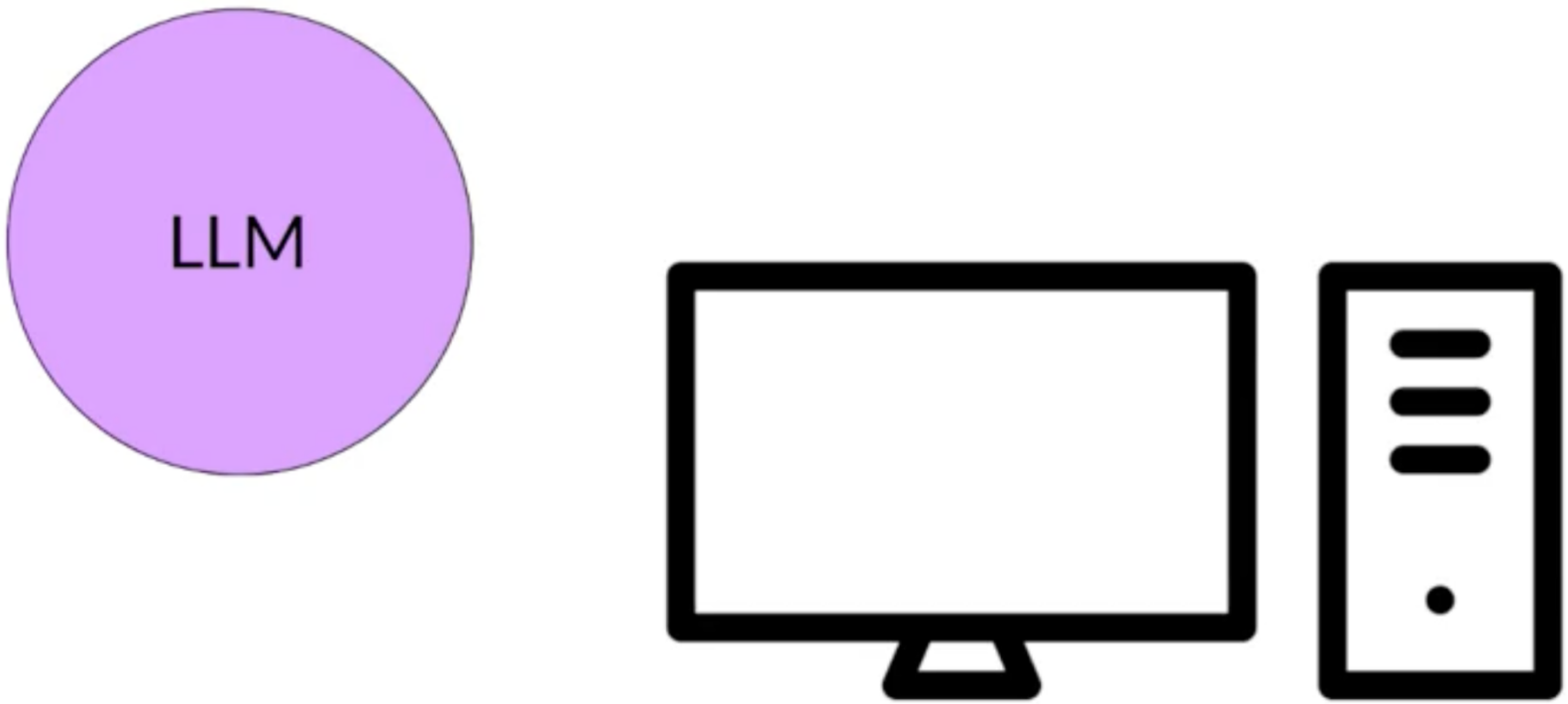
Full fine-tuning of large LLMs is challenging



Full fine-tuning of large LLMs is challenging



Parameter efficient fine-tuning (PEFT)

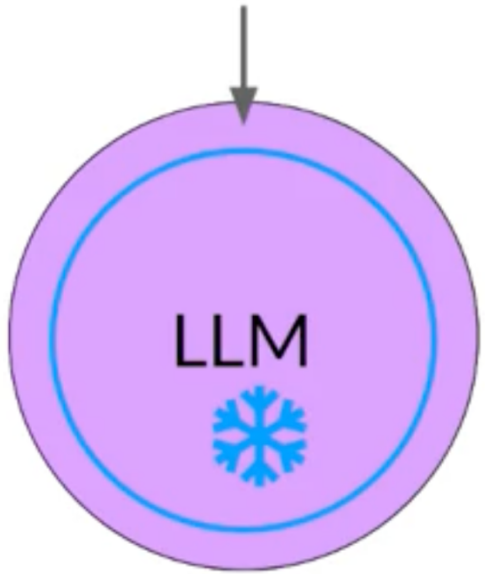


Parameter efficient fine-tuning (PEFT)

LLM with most layers frozen

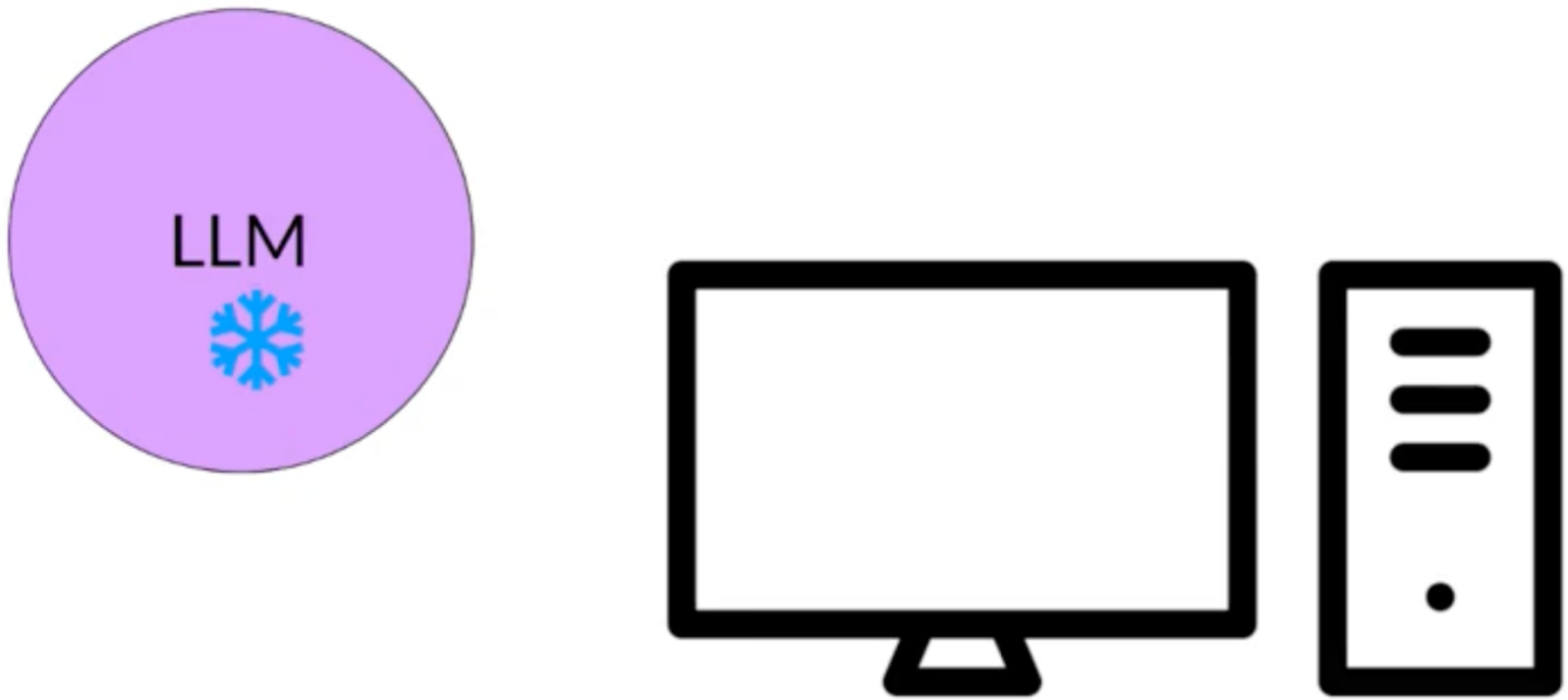
Parameter efficient fine-tuning (PEFT)

Small number of trainable layers



LLM with most layers frozen

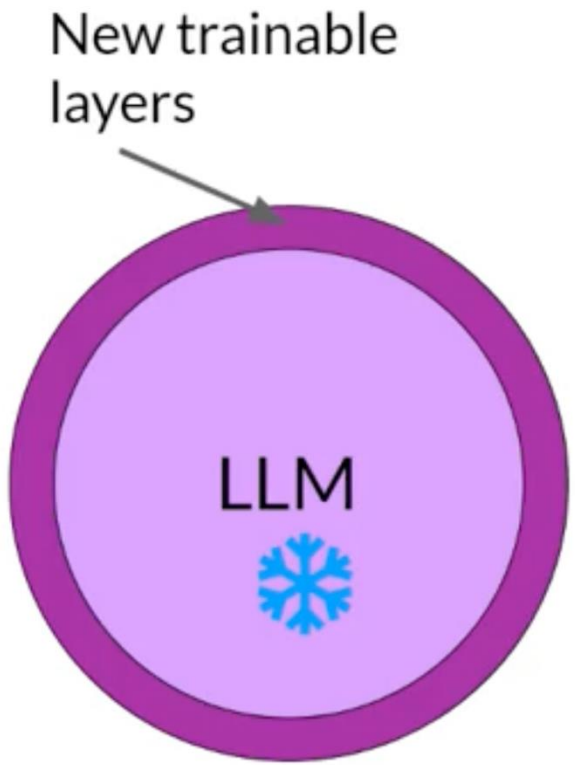
Parameter efficient fine-tuning (PEFT)



Parameter efficient fine-tuning (PEFT)

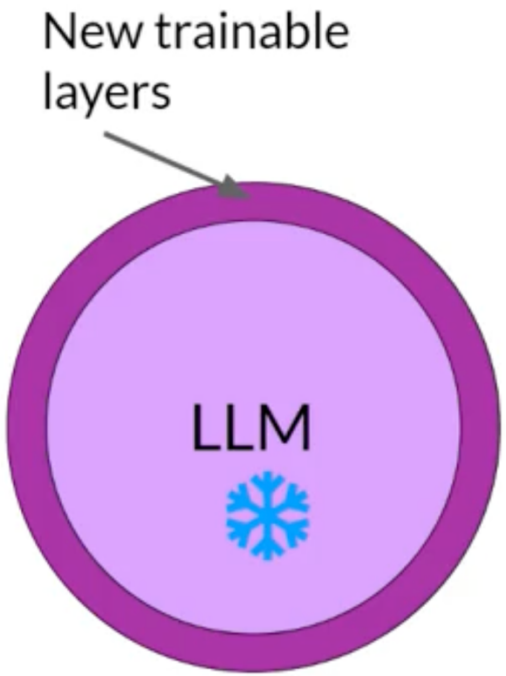
LLM with additional layers for PEFT

Parameter efficient fine-tuning (PEFT)

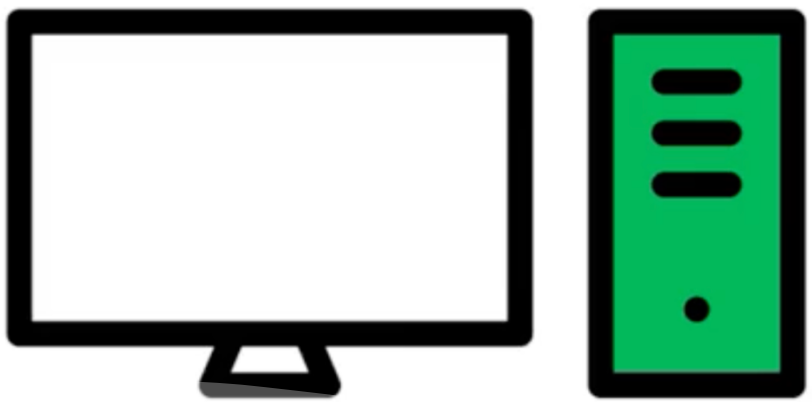


LLM with additional
layers for PEFT

Parameter efficient fine-tuning (PEFT)

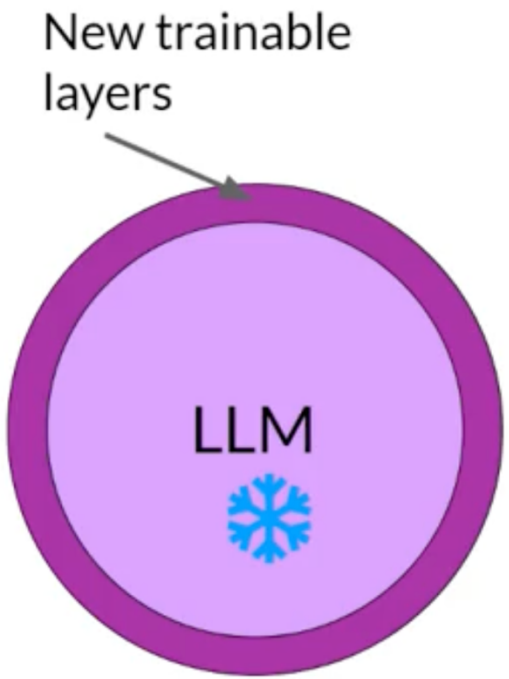


LLM with additional
layers for PEFT

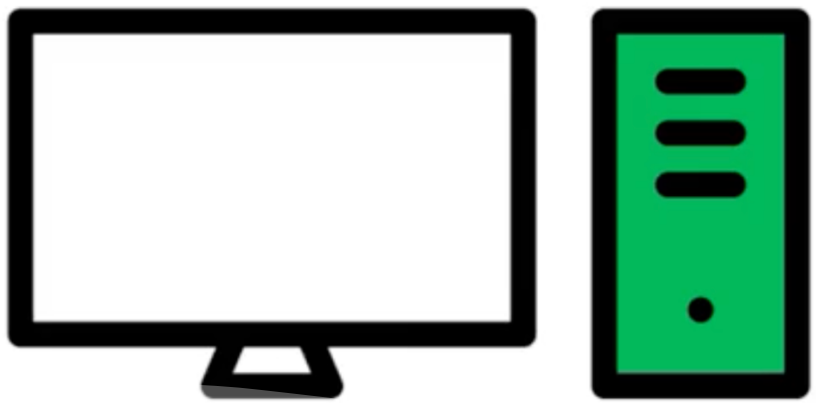


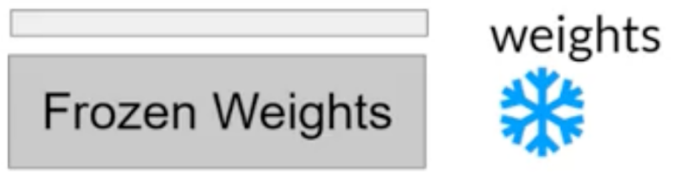
Frozen Weights

Parameter efficient fine-tuning (PEFT)

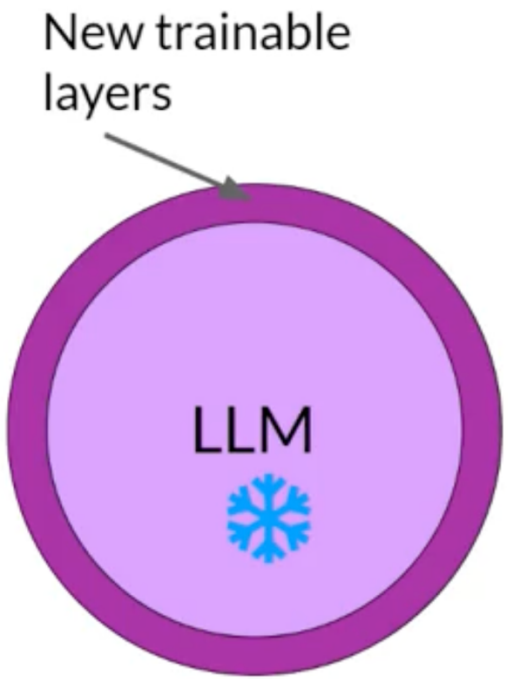


LLM with additional
layers for PEFT

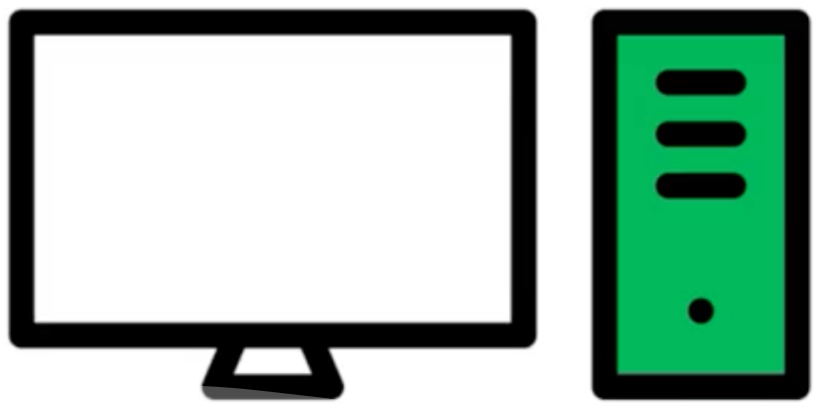


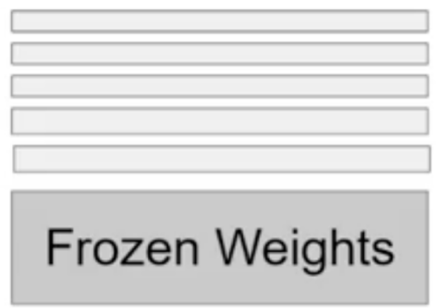


Parameter efficient fine-tuning (PEFT)



LLM with additional
layers for PEFT

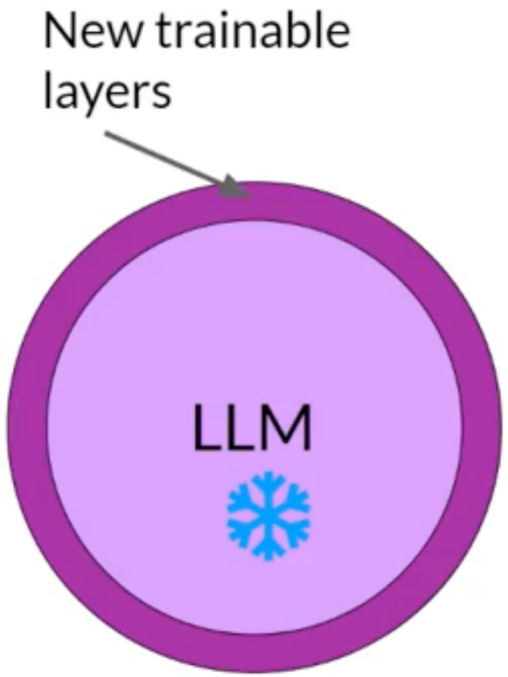




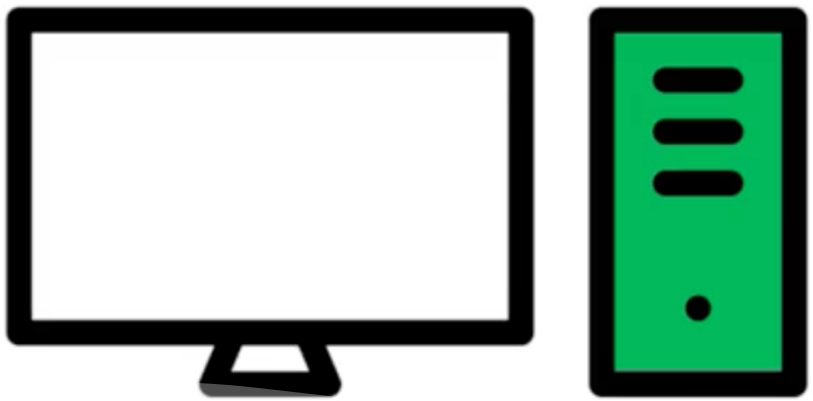
Other
components

Trainable
weights

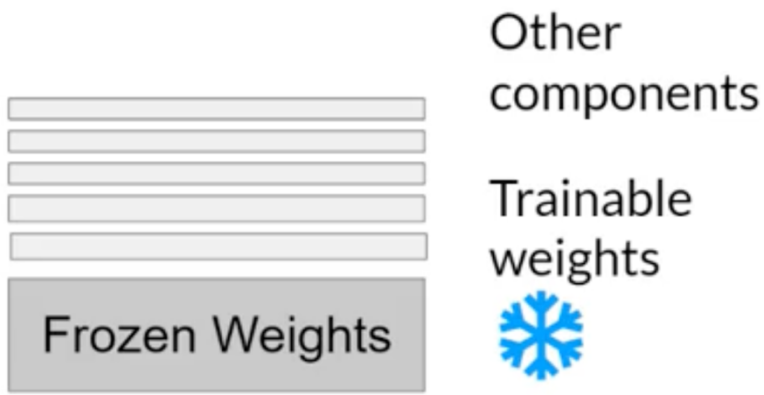
Parameter efficient fine-tuning (PEFT)



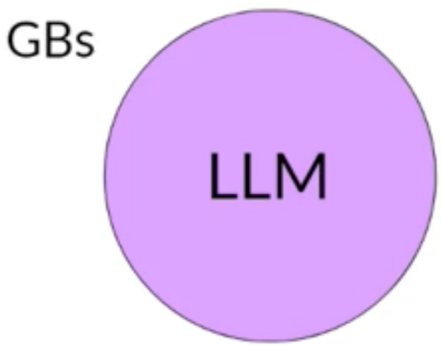
LLM with additional
layers for PEFT



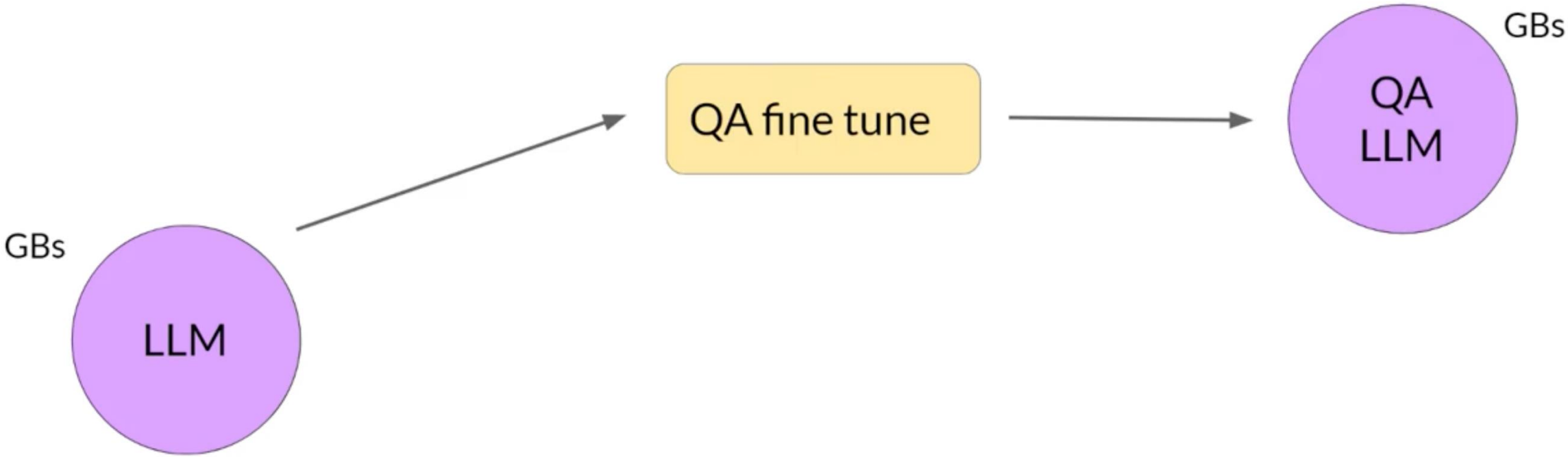
Less prone to
catastrophic forgetting



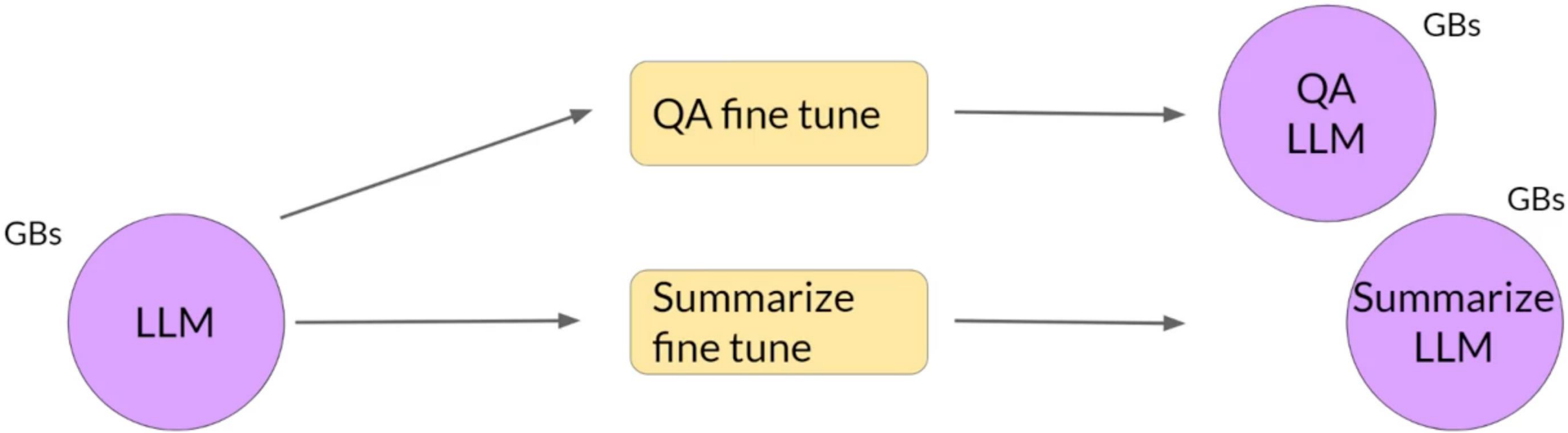
Full fine-tuning creates full copy of original LLM per task



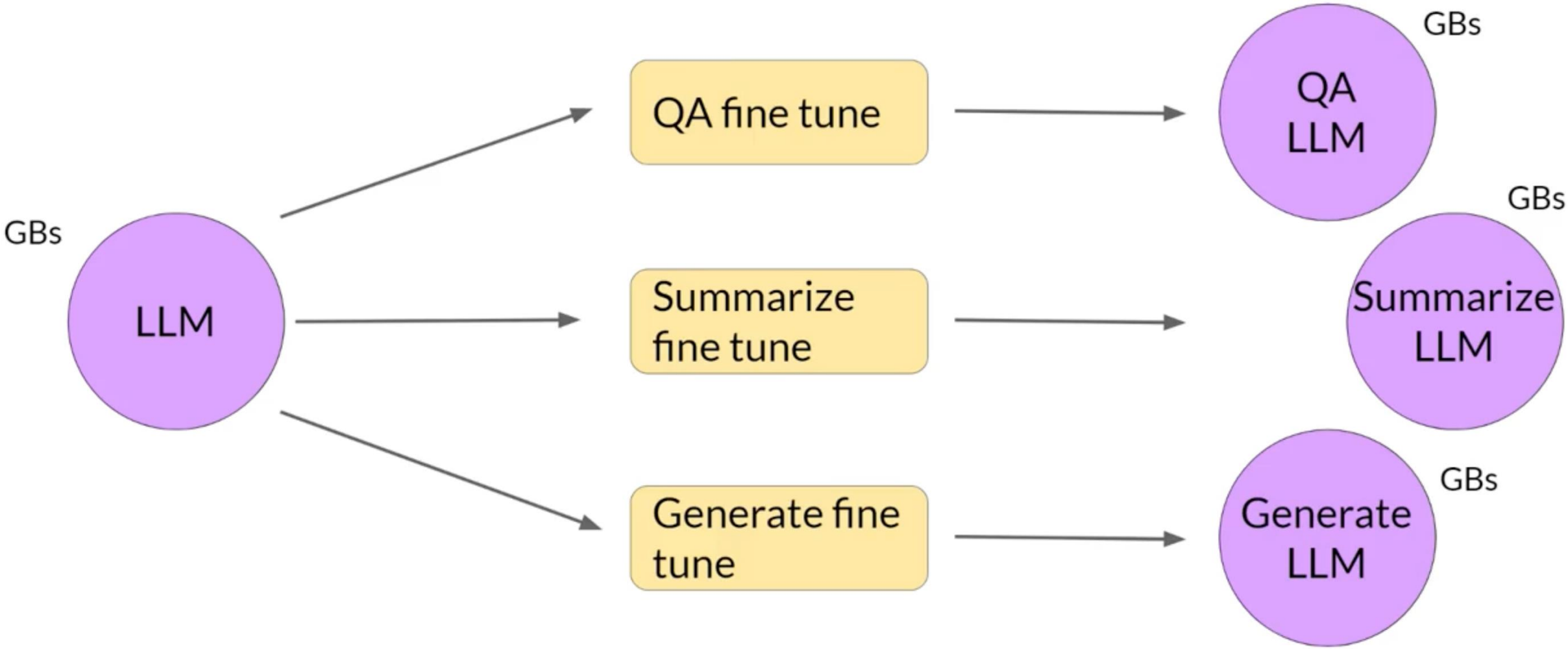
Full fine-tuning creates full copy of original LLM per task



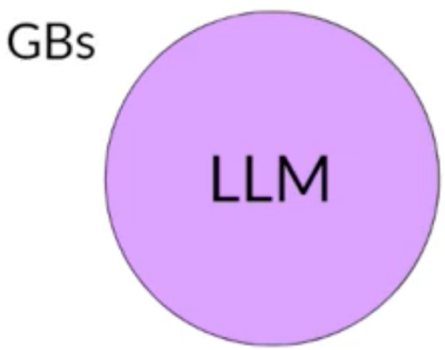
Full fine-tuning creates full copy of original LLM per task



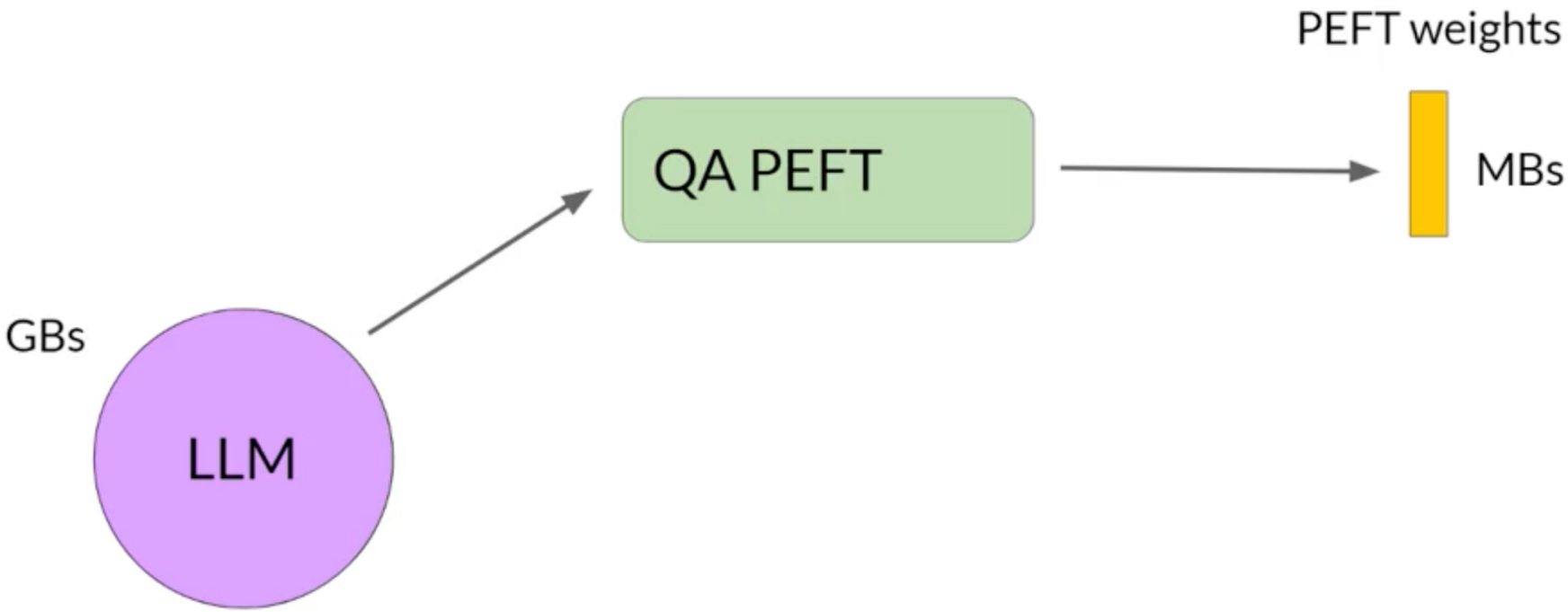
Full fine-tuning creates full copy of original LLM per task



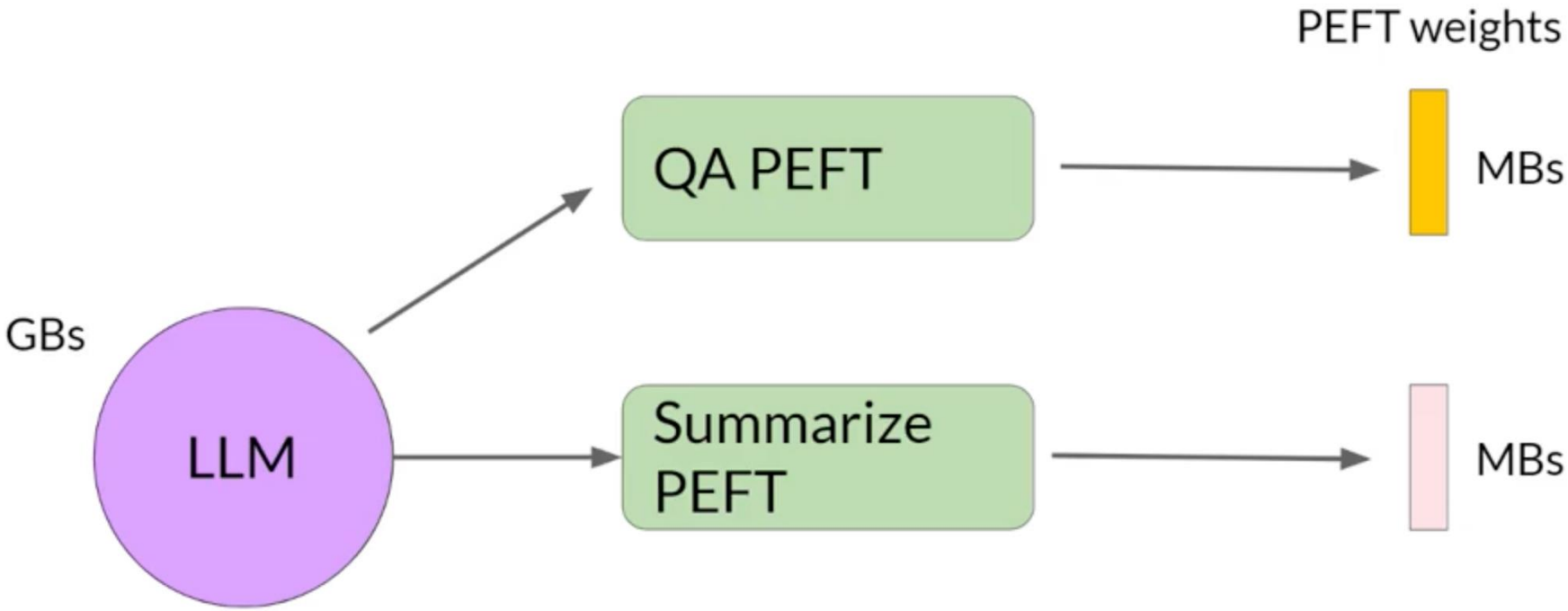
PEFT fine-tuning saves space and is flexible



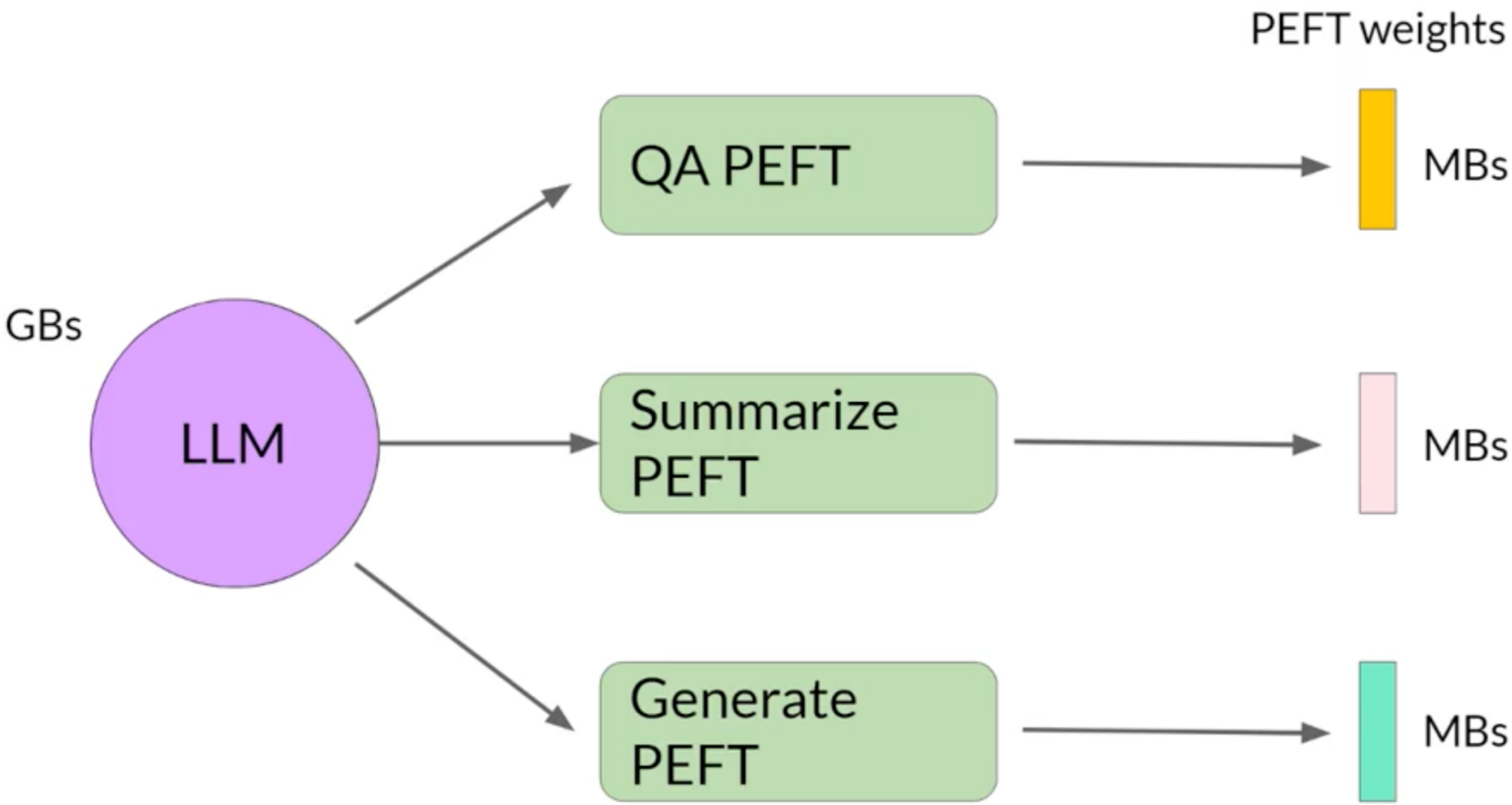
PEFT fine-tuning saves space and is flexible



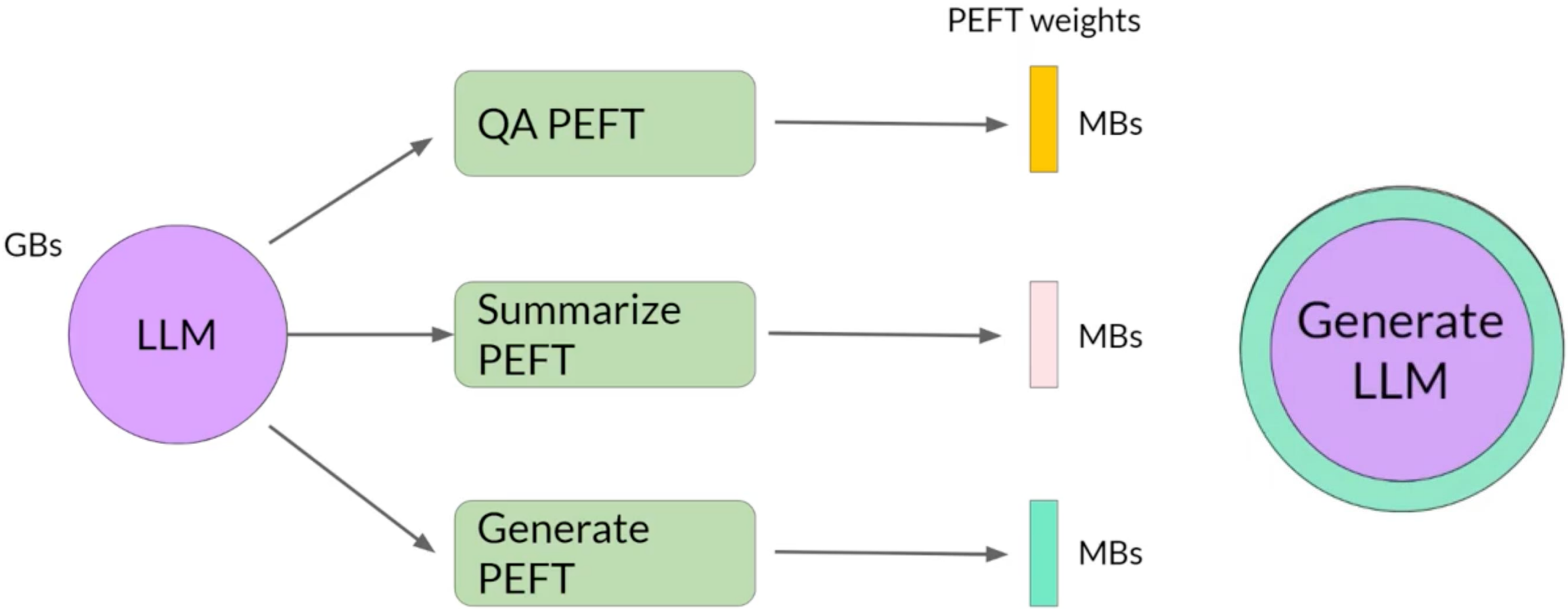
PEFT fine-tuning saves space and is flexible



PEFT fine-tuning saves space and is flexible



PEFT fine-tuning saves space and is flexible



PEFT Trade-offs

Parameter Efficiency

Memory Efficiency

Training Speed

Model Performance

Inference Costs

PEFT methods

Source: Lialin et al. 2023, “Scaling Down to Scale Up: A Guide to Parameter-Efficient Fine-Tuning”,

PEFT methods

Selective

Select subset of initial LLM parameters to fine-tune

Source: Lialin et al. 2023, “Scaling Down to Scale Up: A Guide to Parameter-Efficient Fine-Tuning”,

PEFT methods

Selective	Reparameterization
Select subset of initial LLM parameters to fine-tune	**Reparameterize** model weights using a low-rank representation LoRA

Source: Lialin et al. 2023, "Scaling Down to Scale Up: A Guide to Parameter-Efficient Fine-Tuning",

PEFT methods

Selective	Reparameterization	Additive
Select subset of initial LLM parameters to fine-tune	**Reparameterize** model weights using a low-rank representation **LoRA**	**Add** trainable layers or parameters to model

Source: Lialin et al. 2023, “Scaling Down to Scale Up: A Guide to Parameter-Efficient Fine-Tuning”,

PEFT methods

Selective	Reparameterization	Additive
Select subset of initial LLM parameters to fine-tune	**Reparameterize** model weights using a low-rank representation **LoRA**	**Add** trainable layers or parameters to model Adapters

Source: Lialin et al. 2023, "Scaling Down to Scale Up: A Guide to Parameter-Efficient Fine-Tuning",

PEFT methods

Selective

Select subset of initial LLM parameters to fine-tune

Reparameterization

Reparameterize model weights using a low-rank representation

LoRA

Additive

Add trainable layers or parameters to model

Adapters

Soft Prompts

Source: Lialin et al. 2023, “Scaling Down to Scale Up: A Guide to Parameter-Efficient Fine-Tuning”,

PEFT methods

Selective

Select subset of initial LLM parameters to fine-tune

Reparameterization

Reparameterize model weights using a low-rank representation

LoRA

Additive

Add trainable layers or parameters to model

Adapters

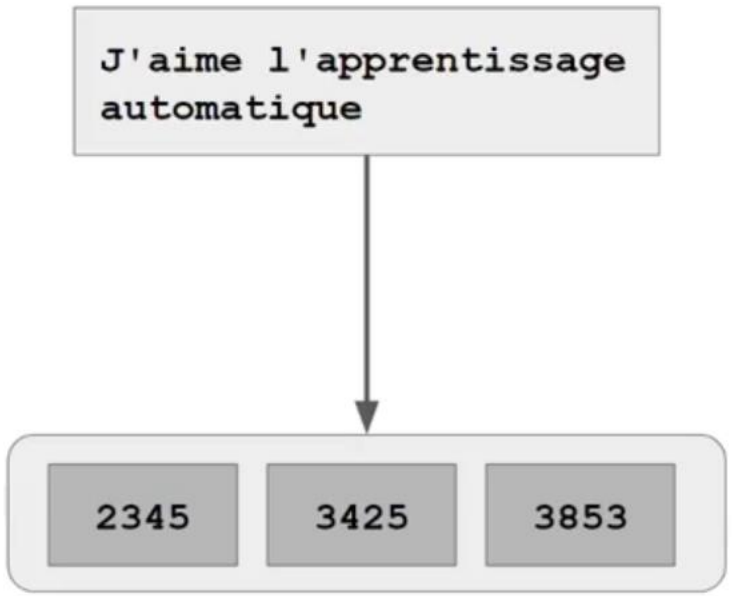
Soft Prompts

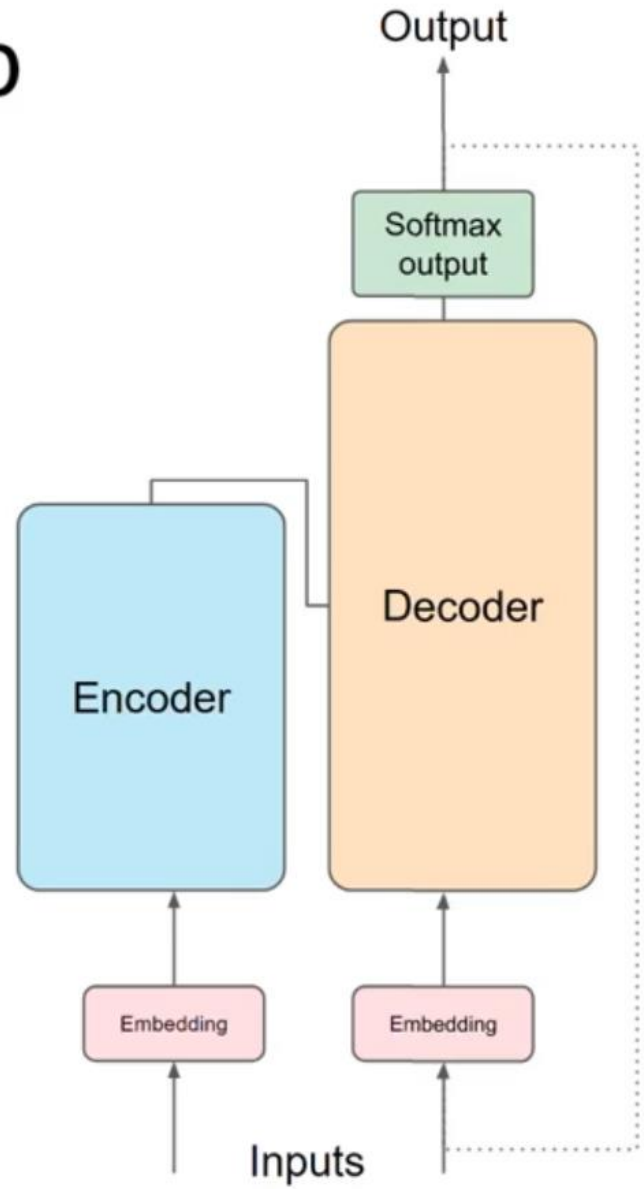
Prompt Tuning

Source: Lialin et al. 2023, “Scaling Down to Scale Up: A Guide to Parameter-Efficient Fine-Tuning”,

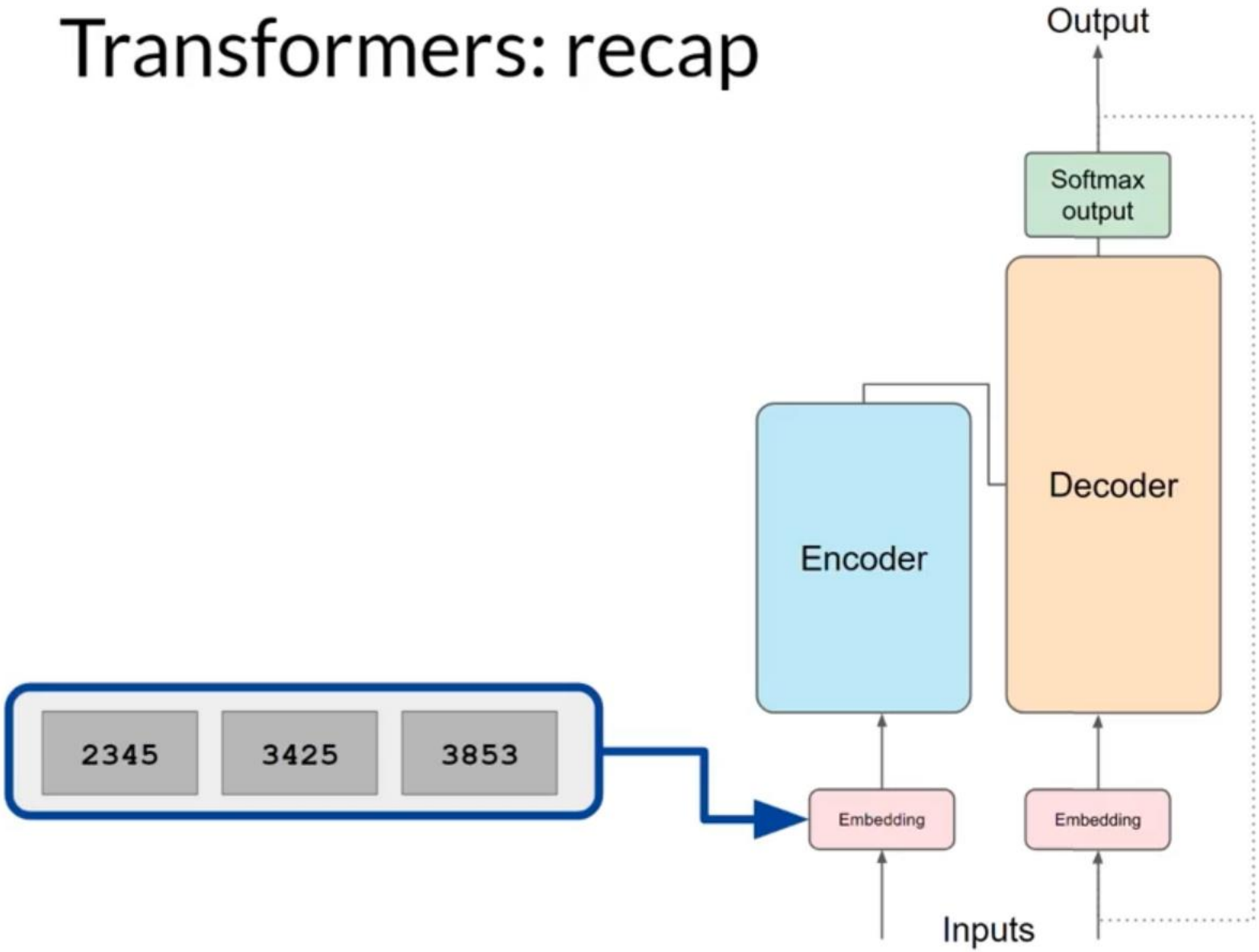
Low-Rank Adaptation of Large Language Models (LoRA)

Transformers: recap

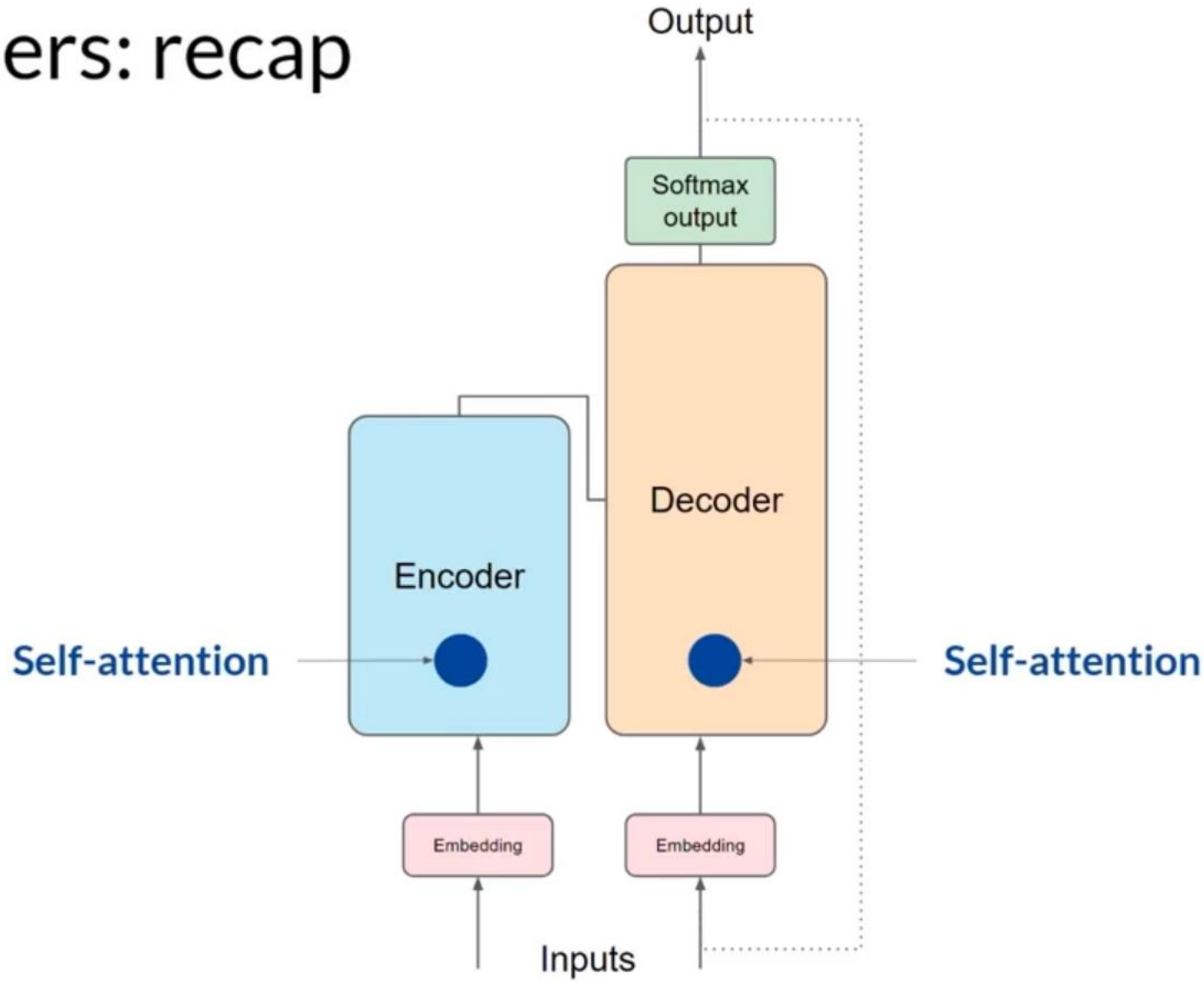




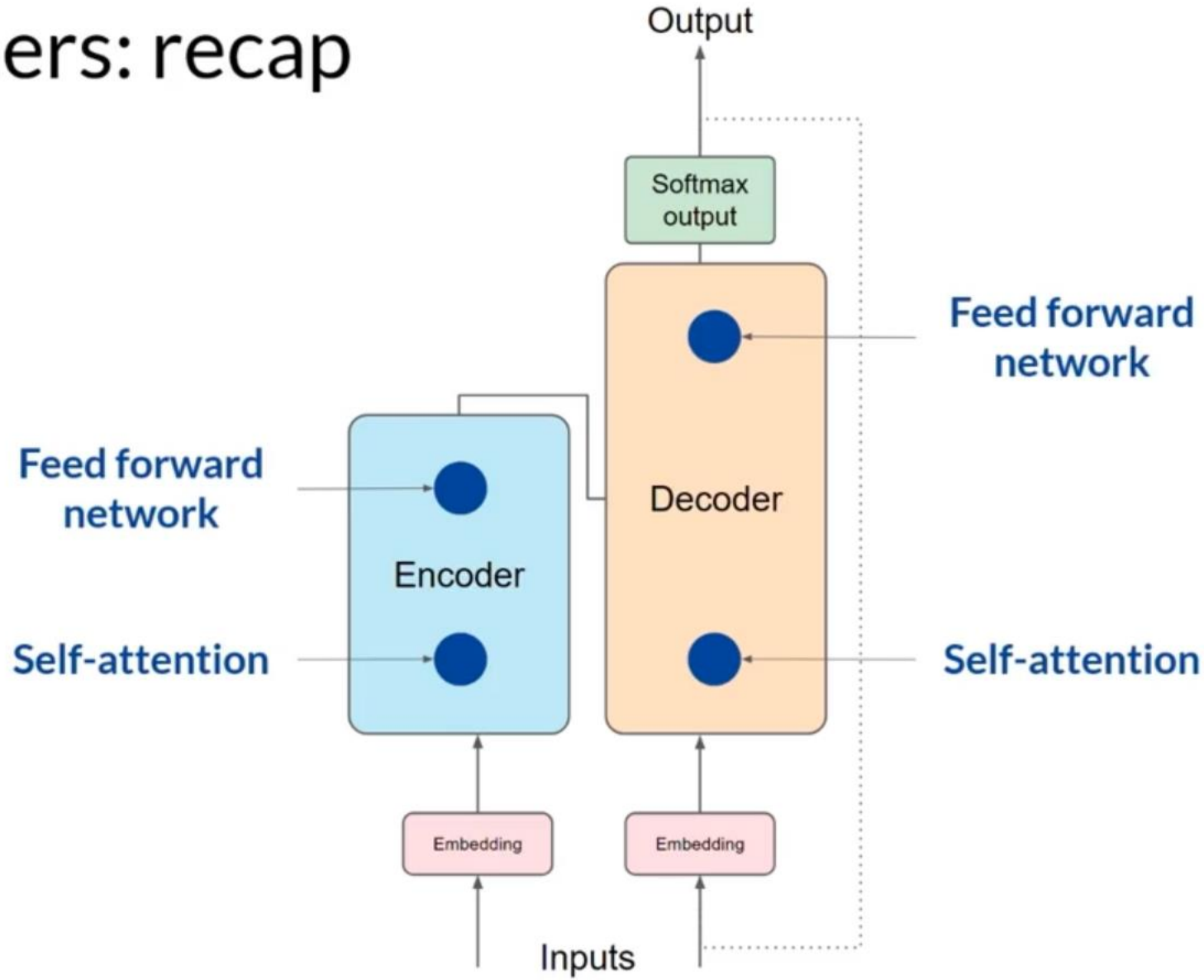
Transformers: recap



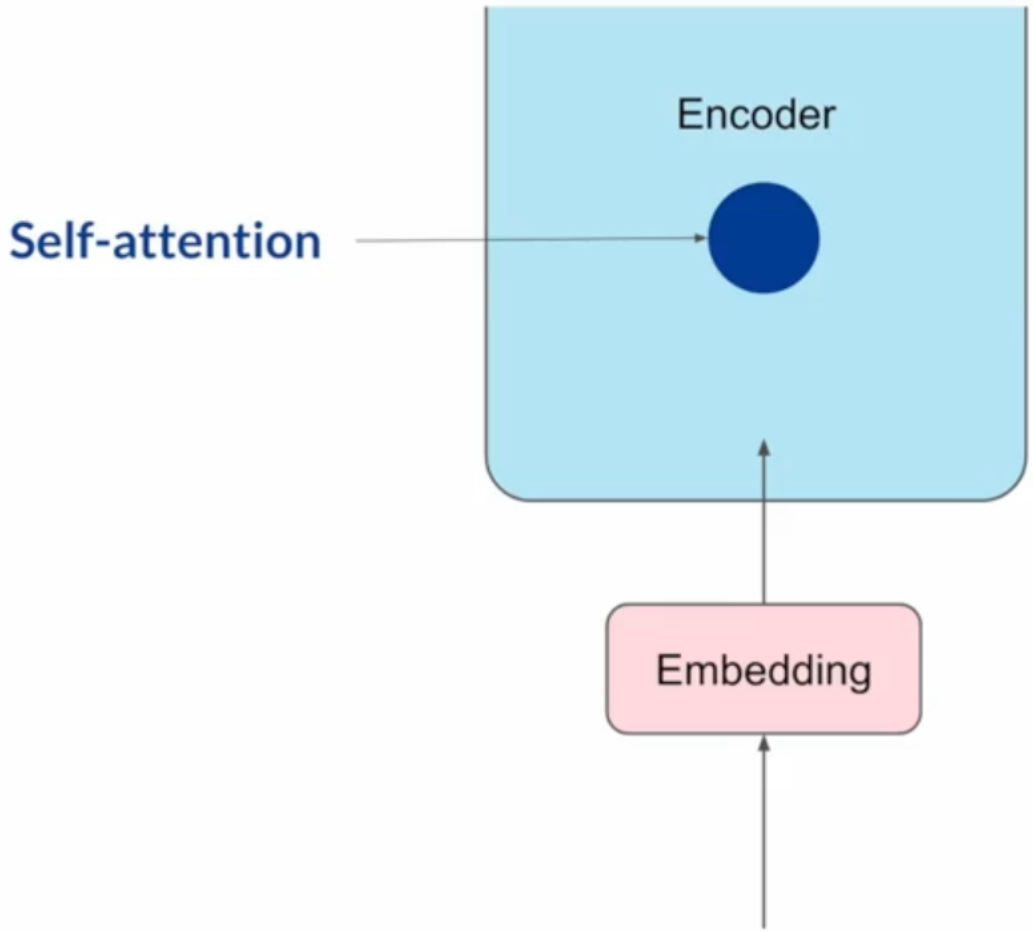
Transformers: recap



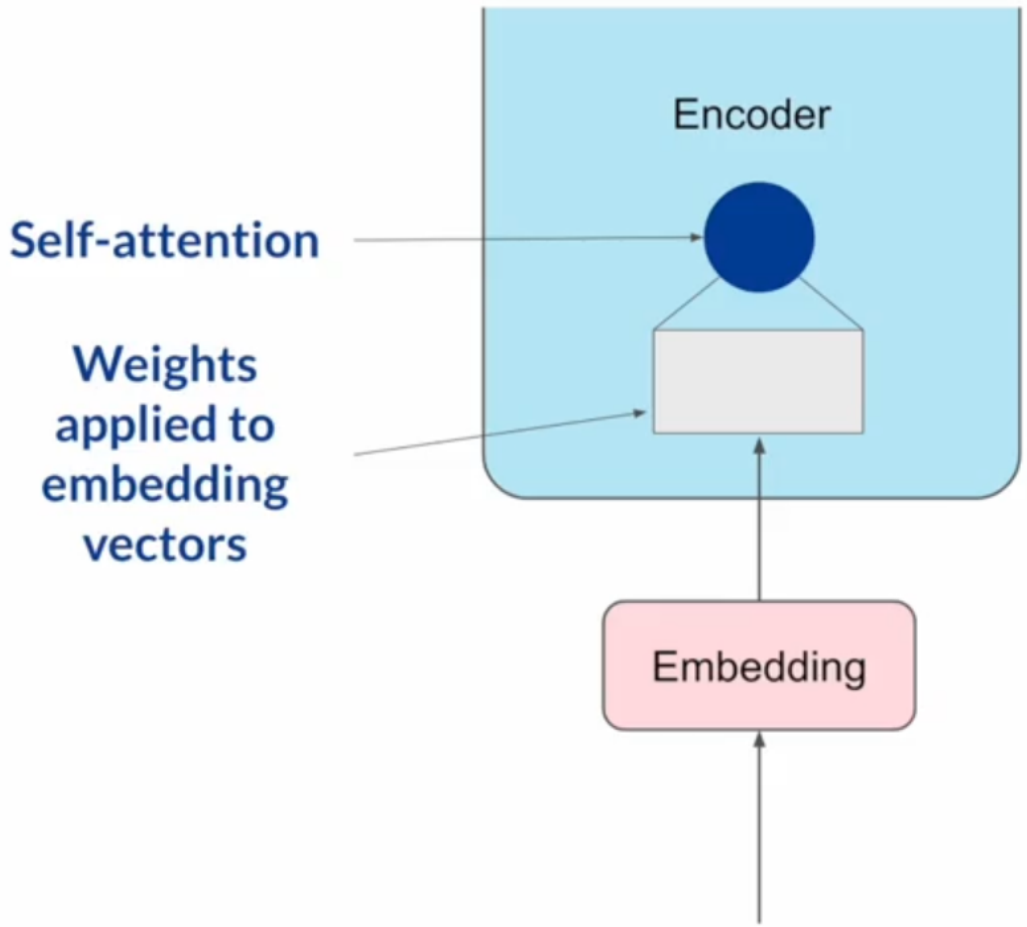
Transformers: recap



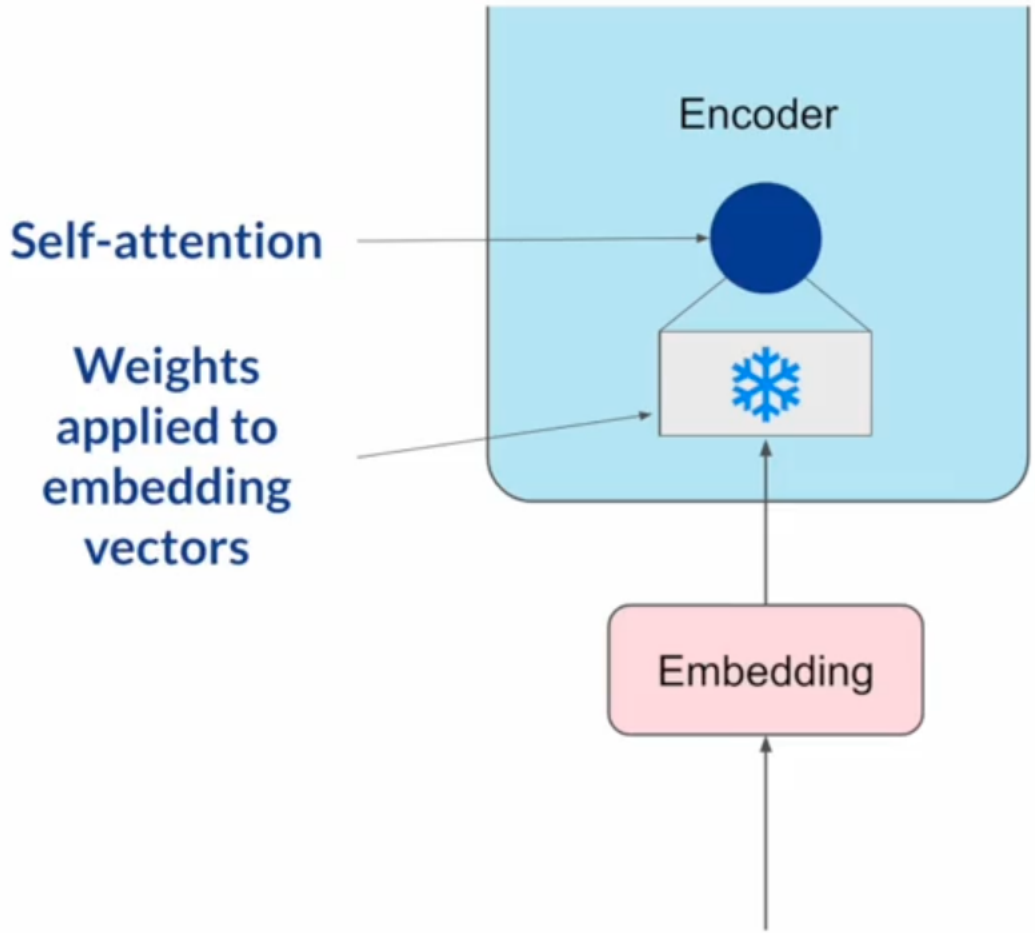
LoRA: Low Rank Adaption of LLMs



LoRA: Low Rank Adaption of LLMs

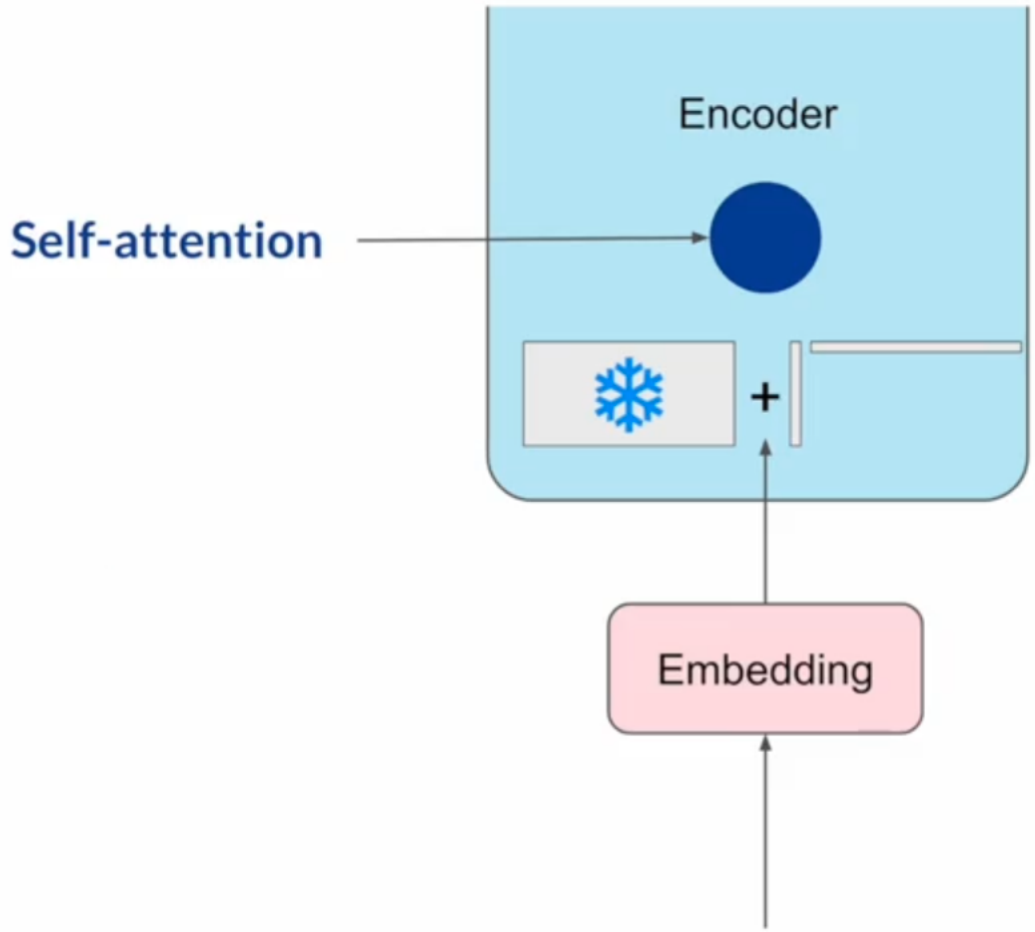


LoRA: Low Rank Adaption of LLMs



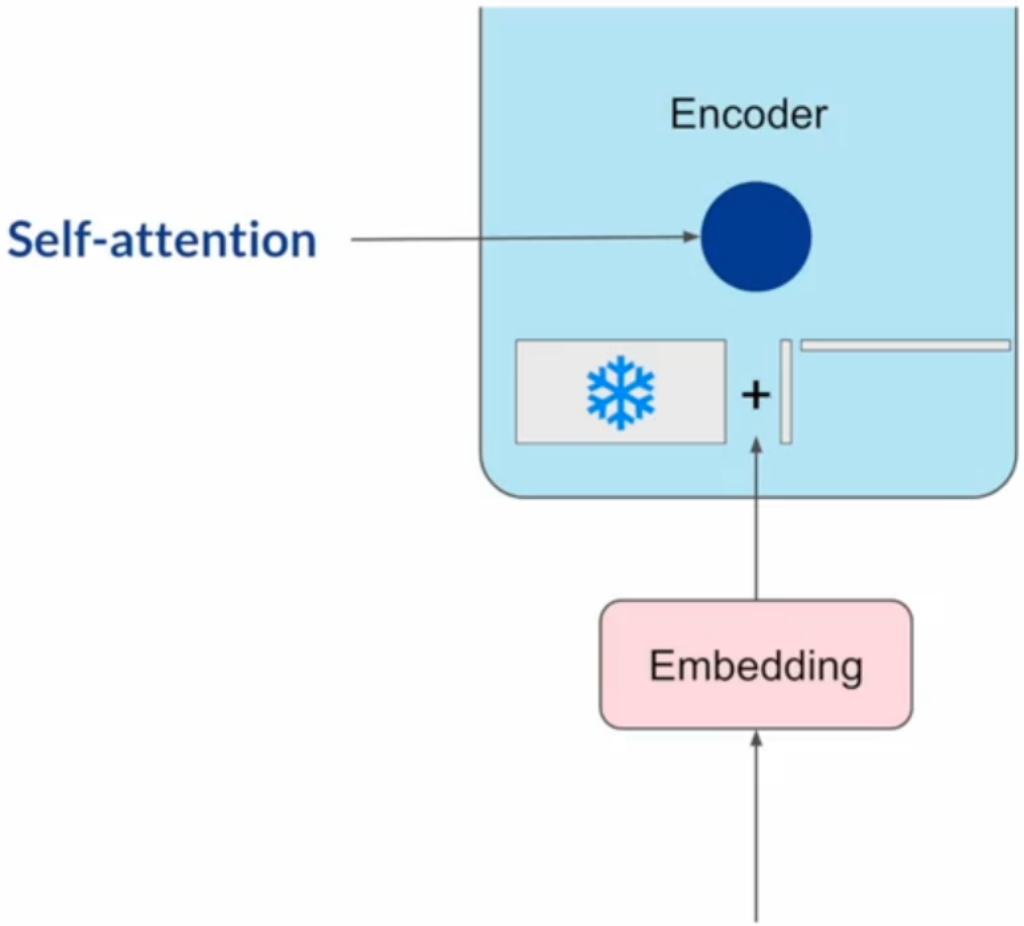
1. Freeze most of the original LLM weights.

LoRA: Low Rank Adaption of LLMs



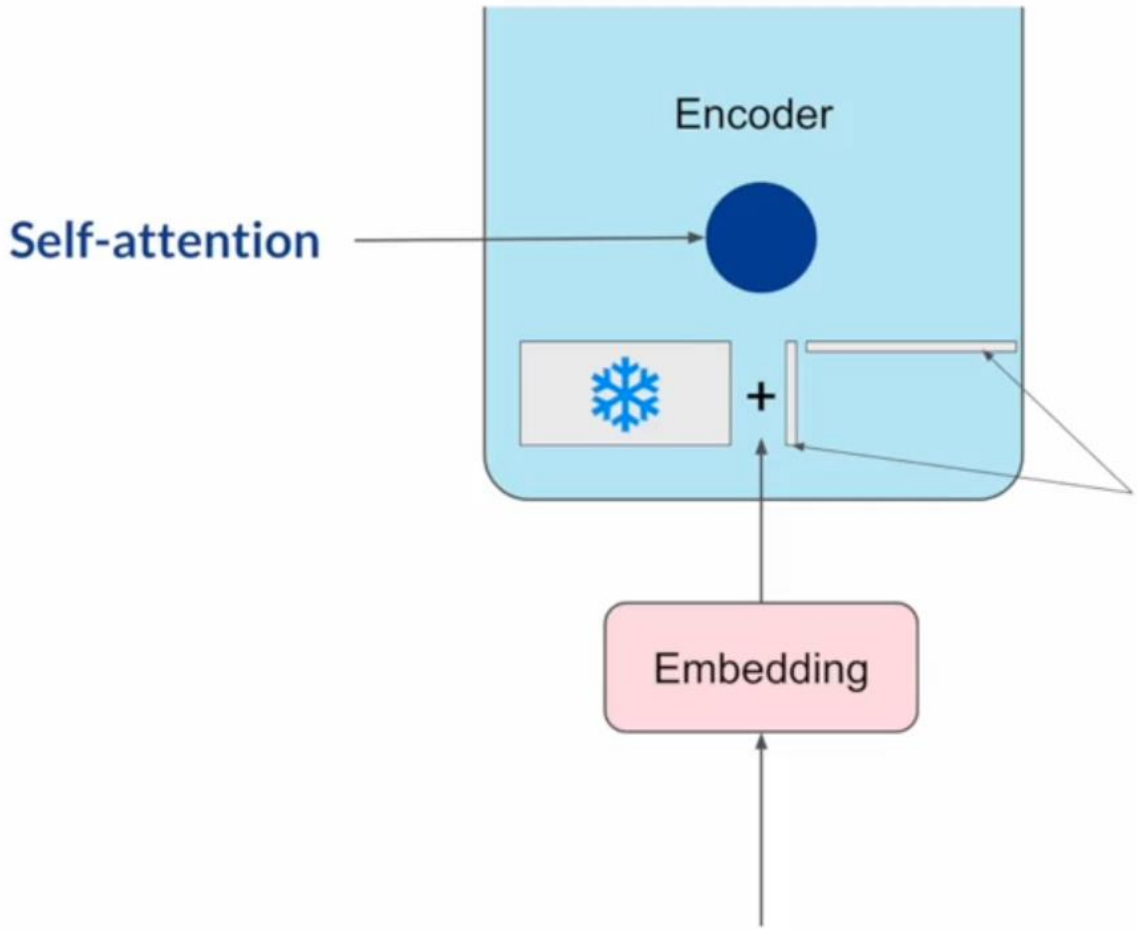
1. Freeze most of the original LLM weights.
2. Inject 2 **rank decomposition matrices**

LoRA: Low Rank Adaption of LLMs



1. Freeze most of the original LLM weights.
2. Inject 2 **rank decomposition matrices**
3. Train the weights of the smaller matrices

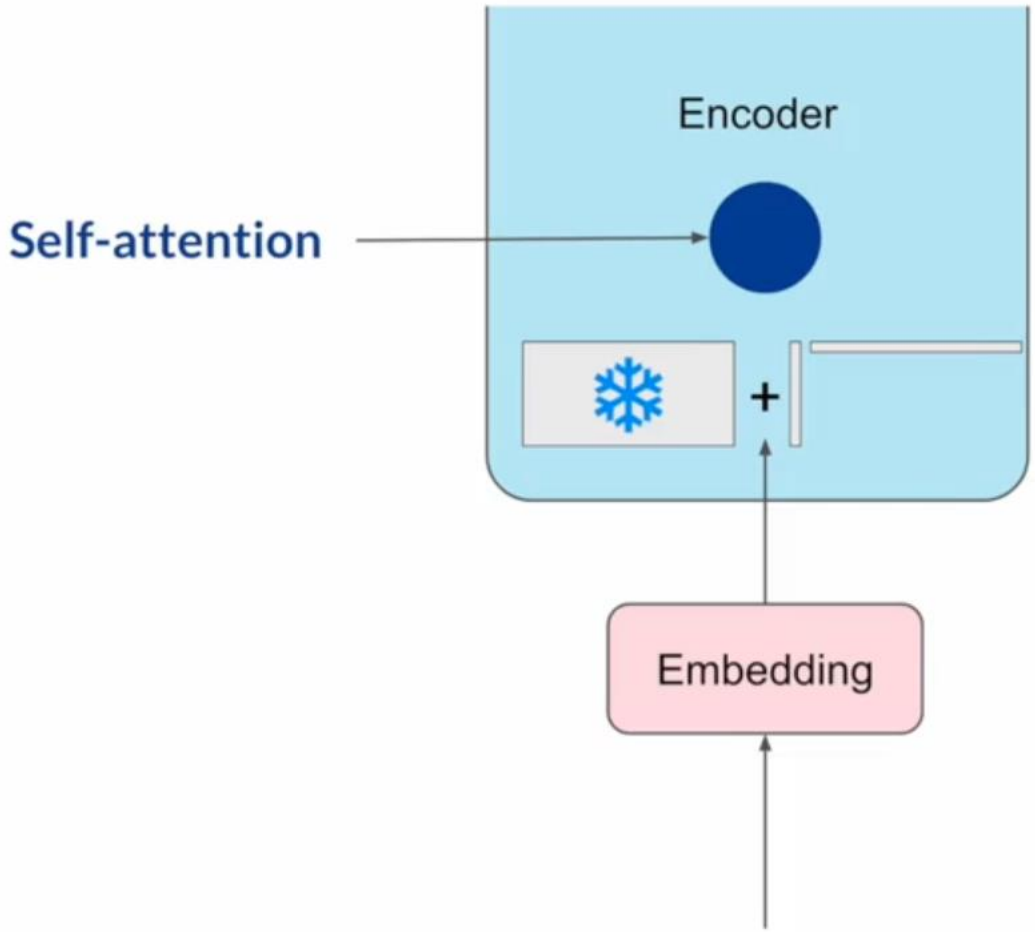
LoRA: Low Rank Adaption of LLMs



1. Freeze most of the original LLM weights.
2. Inject 2 **rank decomposition matrices**
3. Train the weights of the smaller matrices

Rank *r* is small dimension, typically 4, 8 … 64

LoRA: Low Rank Adaption of LLMs



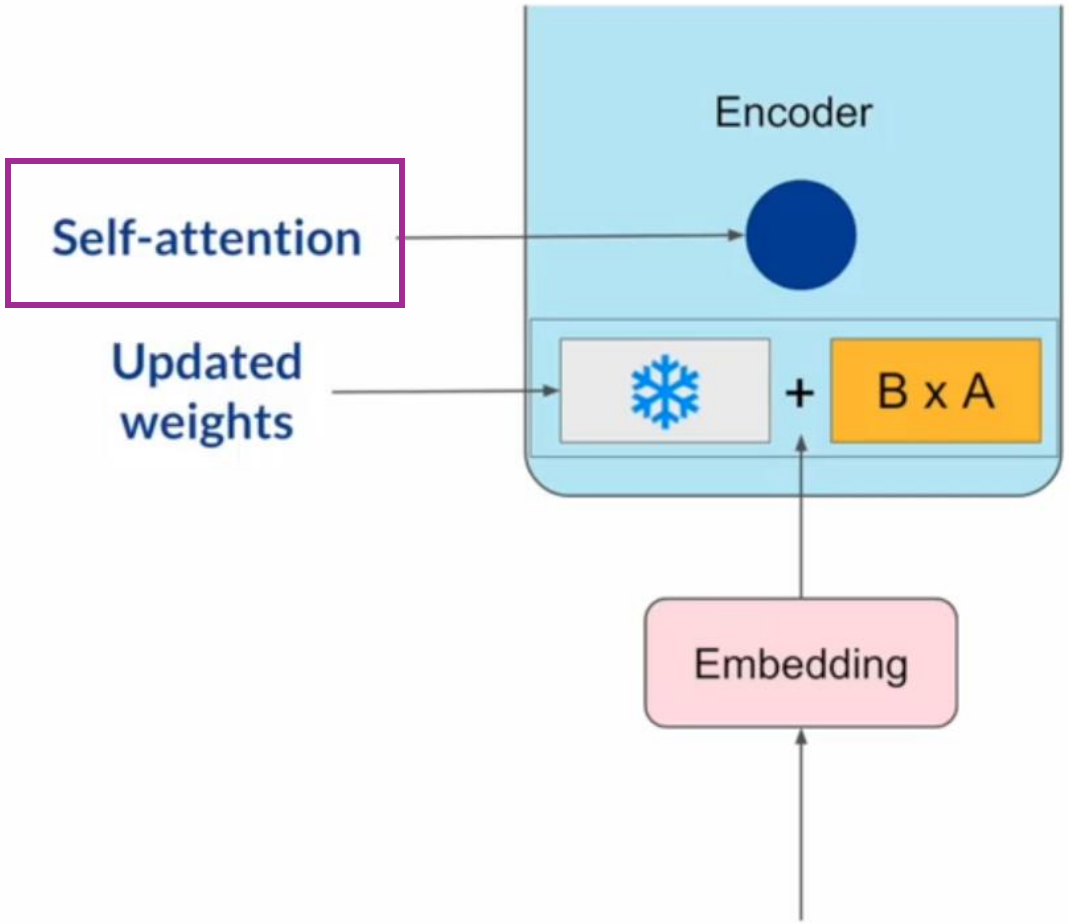
1. Freeze most of the original LLM weights.
2. Inject 2 **rank decomposition matrices**
3. Train the weights of the smaller matrices

Steps to update model for inference

1. Matrix multiply the low rank matrices

2. Add to original weights

LoRA: Low Rank Adaption of LLMs



1. Freeze most of the original LLM weights.
2. Inject 2 **rank decomposition matrices**
3. Train the weights of the smaller matrices

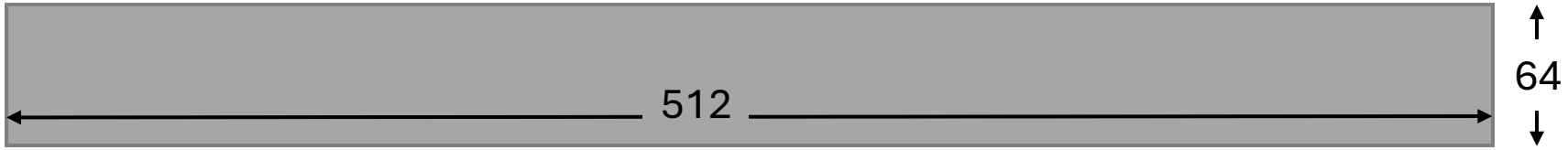
Steps to update model for inference:

1. Matrix multiply the low rank matrices

2. Add to original weights

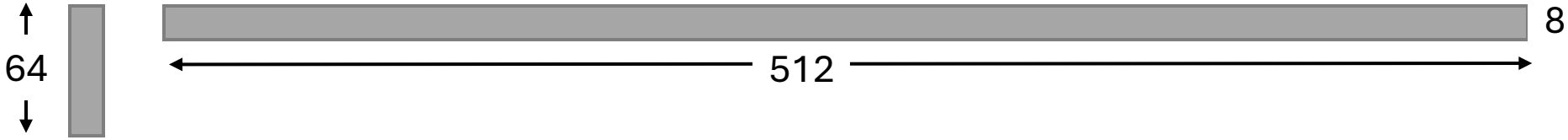
Concrete example using base Transformer as reference

- Use the base Transformer model presented by Vaswani et al. 2017:
  - Transformer weights have dimensions $d \times k = 512 \times 64$
  - So $512 \times 64 = 32{,}768$ trainable parameters

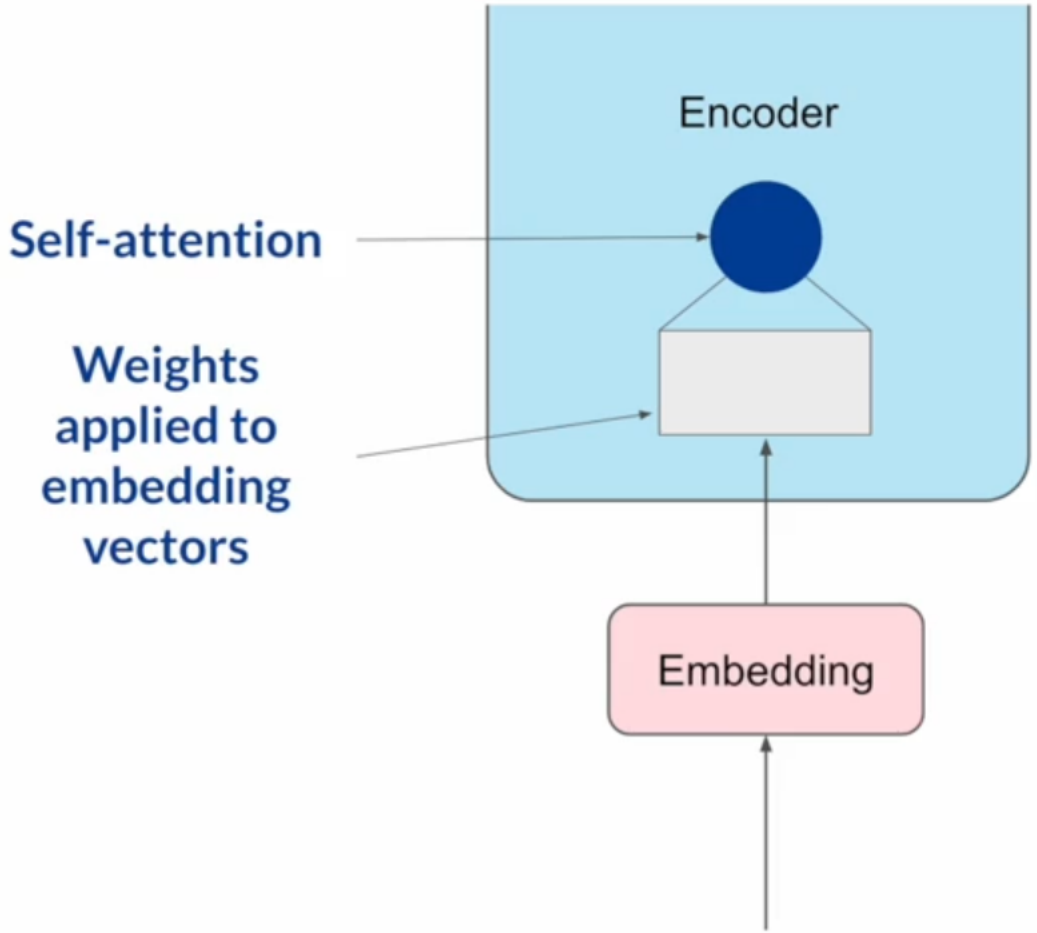


In LoRA with rank $r = 8$:

- A has dimensions $r \times k = 8 \times 64 = 512$ parameters
- B has dimensions $d \times r = 512 \times 8 = 4{,}096$ parameters
- **86% reduction in parameters to train!**

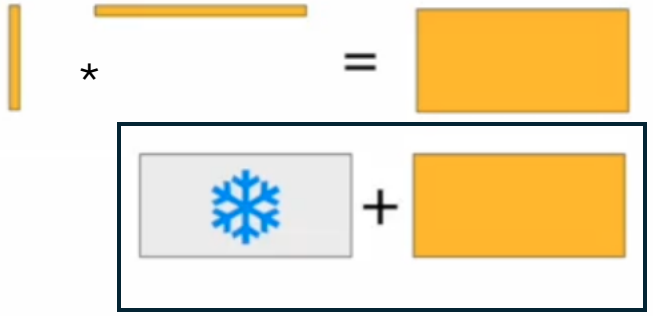


LoRA: Low Rank Adaption of LLMs

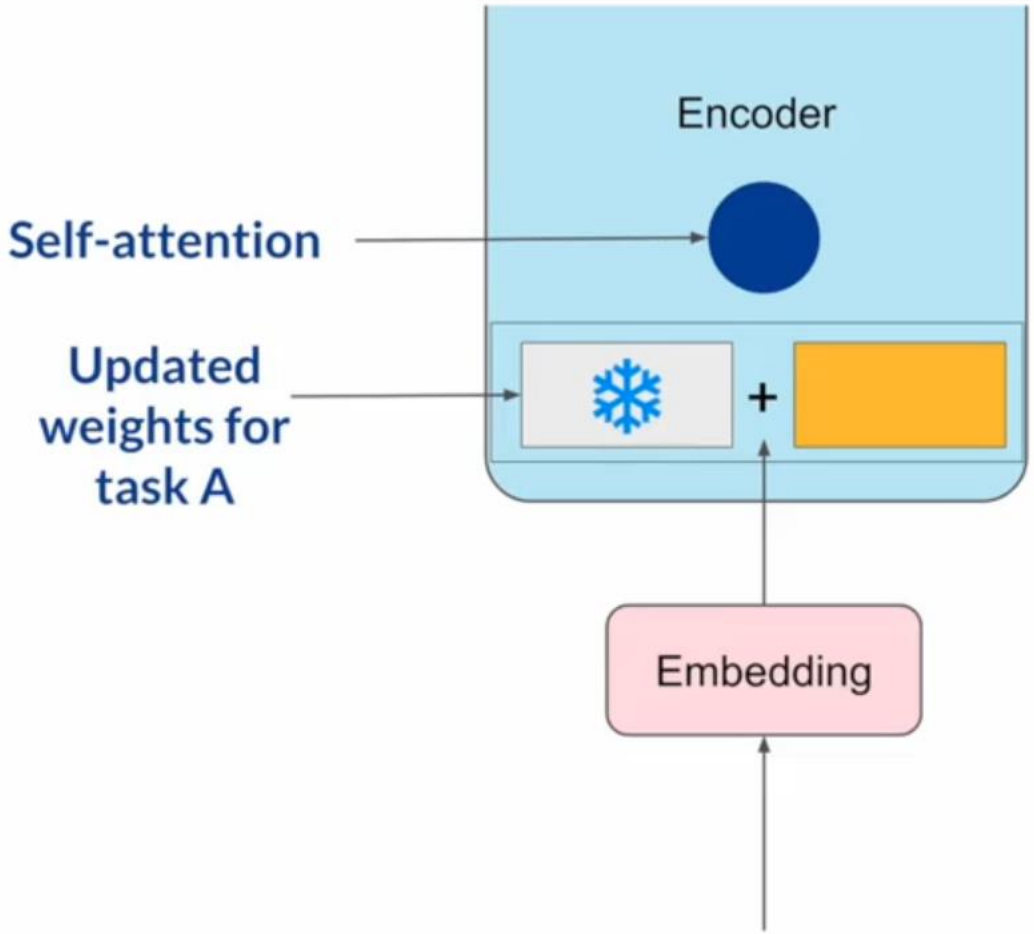


1. Train different rank decomposition matrices for different tasks
2. Update weights before inference

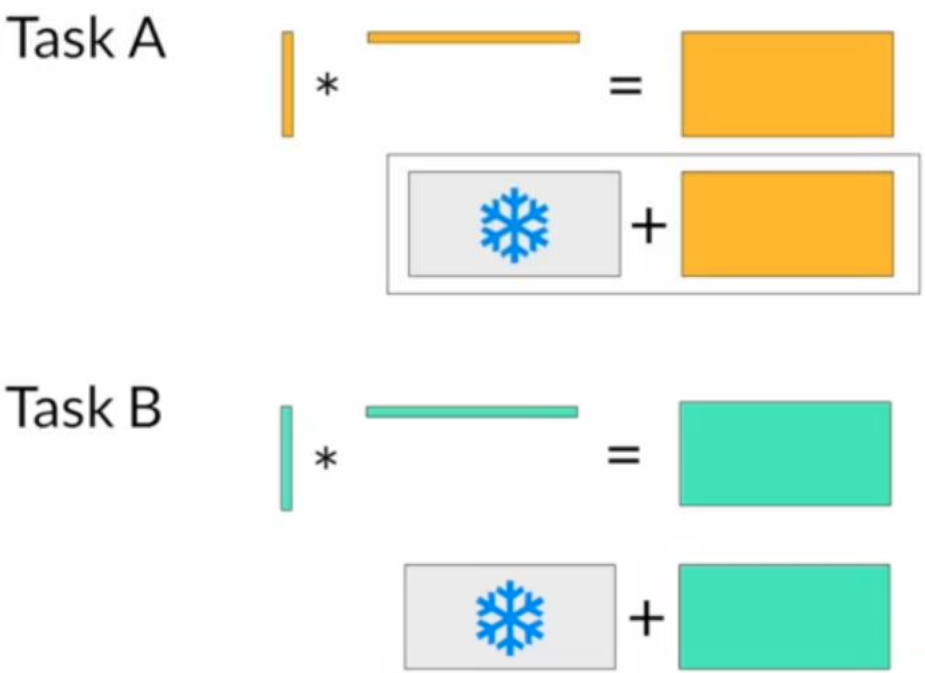
Task A



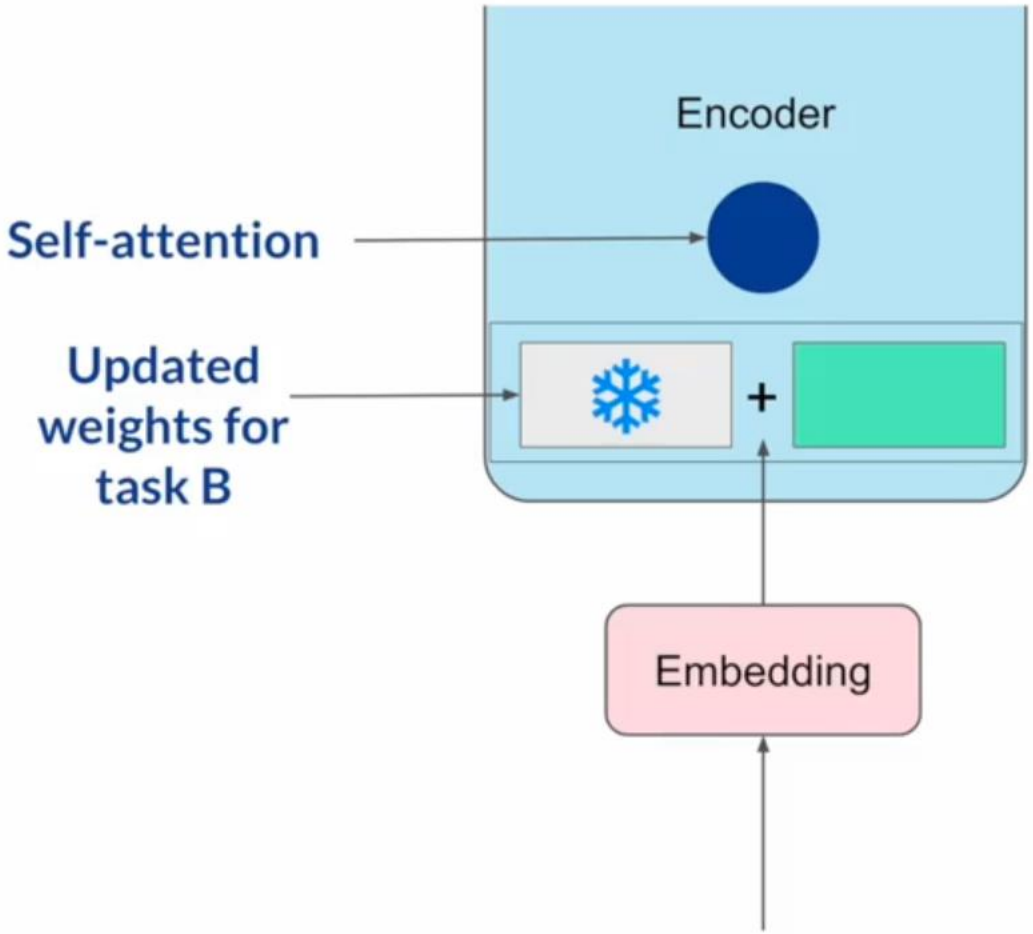
LoRA: Low Rank Adaption of LLMs



1. Train different rank decomposition matrices for different tasks
2. Update weights before inference

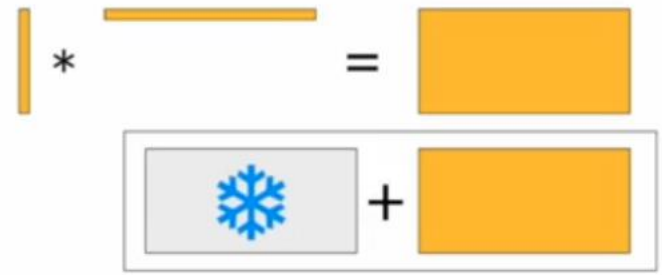


LoRA: Low Rank Adaption of LLMs

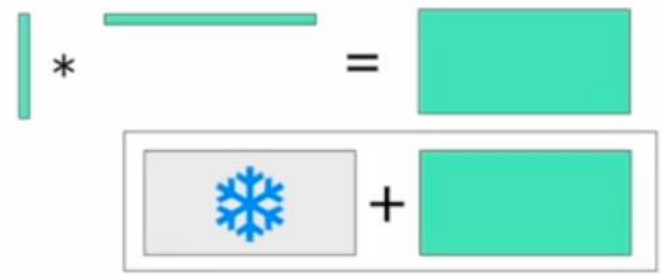


1. Train different rank decomposition matrices for different tasks
2. Update weights before inference

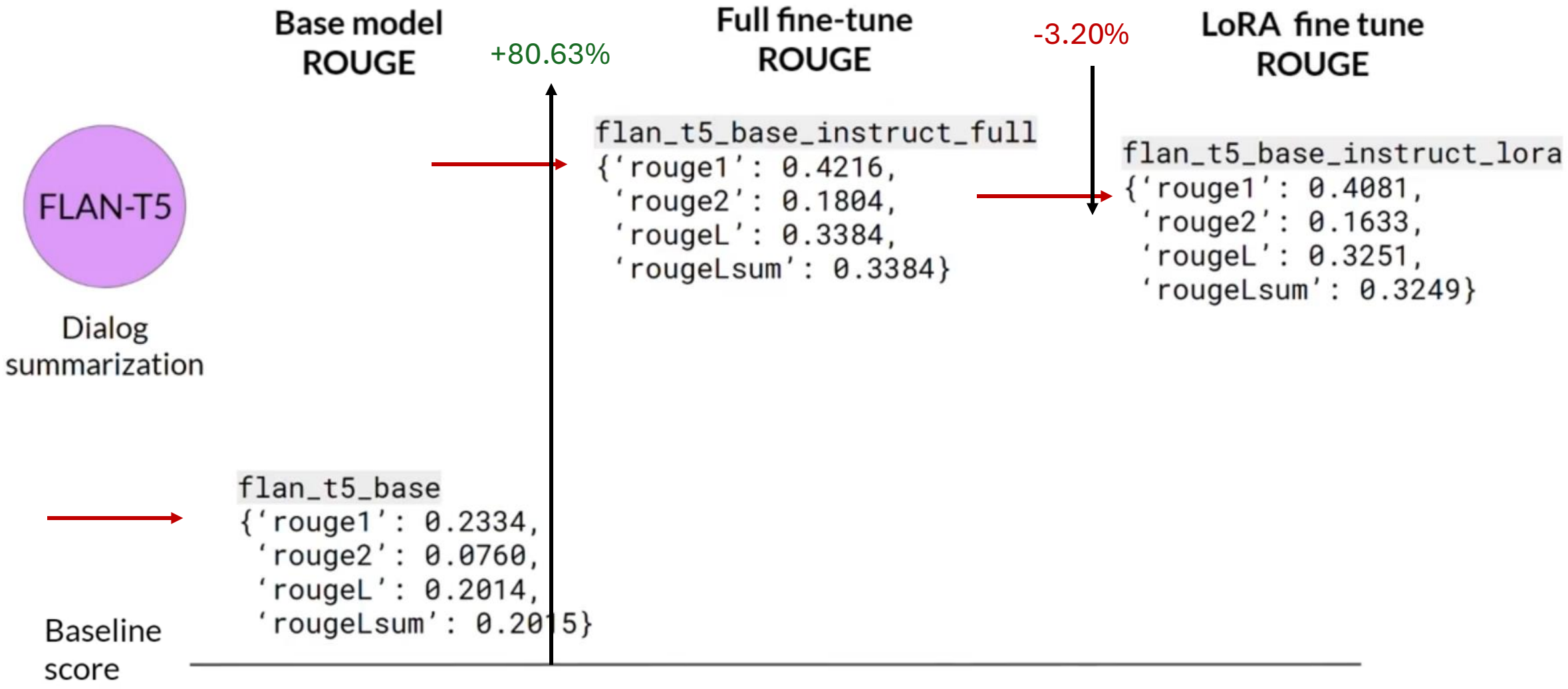
Task A



Task B



Sample ROUGE metrics for full vs. LoRA fine-tuning



Choosing the LoRA rank

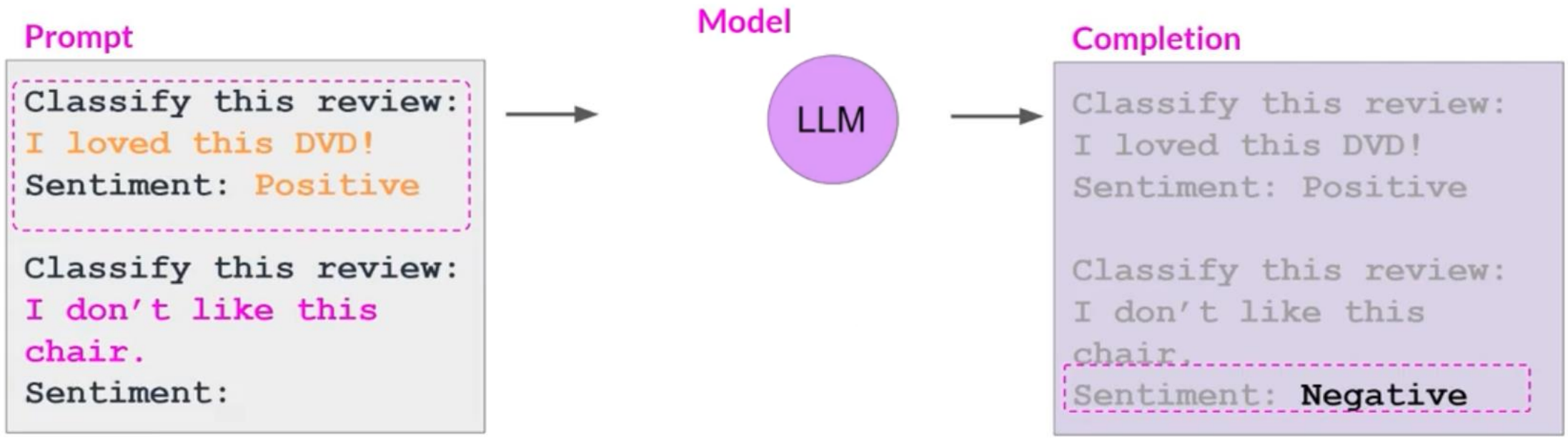
Rank r	val_loss	BLEU	NIST	METEOR	ROUGE_L	CIDEr
1	1.23	68.72	8.7215	0.4565	0.7052	2.4329
2	1.21	69.17	8.7413	0.4590	0.7052	2.4639
4	1.18	**70.38**	**8.8439**	**0.4689**	0.7186	**2.5349**
8	1.17	69.57	8.7457	0.4636	**0.7196**	2.5196
16	**1.16**	69.61	8.7483	0.4629	0.7177	2.4985
32	**1.16**	69.33	8.7736	0.4642	0.7105	2.5255
64	**1.16**	69.24	8.7174	0.4651	0.7180	2.5070
128	**1.16**	68.73	8.6718	0.4628	0.7127	2.5030
256	**1.16**	68.92	8.6982	0.4629	0.7128	2.5012
512	**1.16**	68.78	8.6857	0.4637	0.7128	2.5025
1024	1.17	69.37	8.7495	0.4659	0.7149	2.5090

- Effectiveness of higher rank appears to plateau
- Relationship between rank and dataset size needs more empirical data

Source: Hu et al. 2021, "LoRA: Low-Rank Adaptation of Large Language Models"

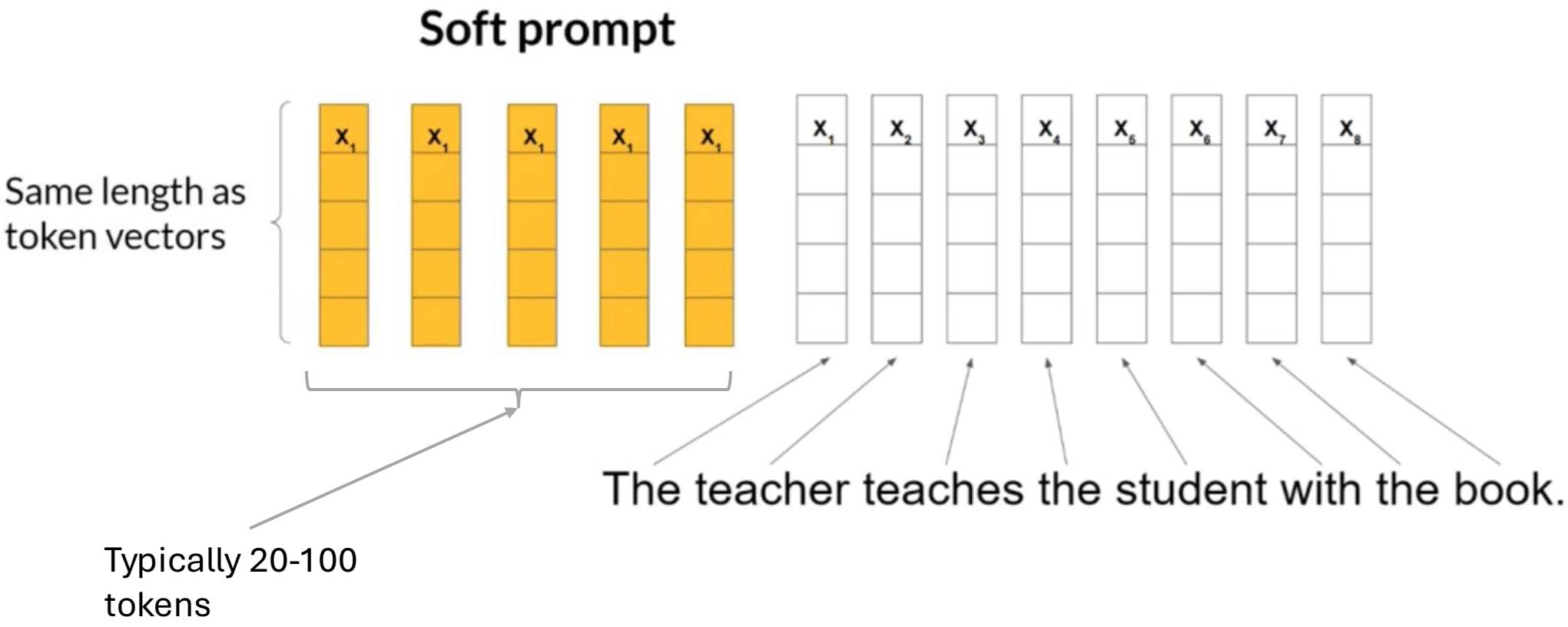
Prompt tuning with soft prompts

Prompt tuning is **not** prompt engineering!

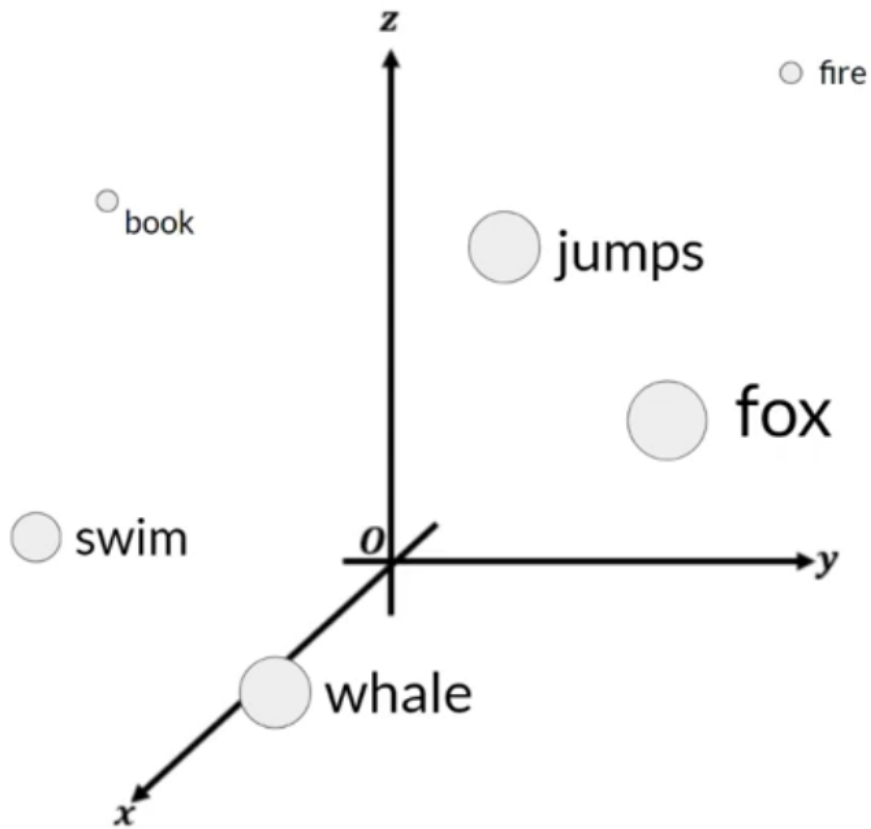


One-shot or Few-shot Inference

Prompt tuning adds trainable “soft prompt” to inputs

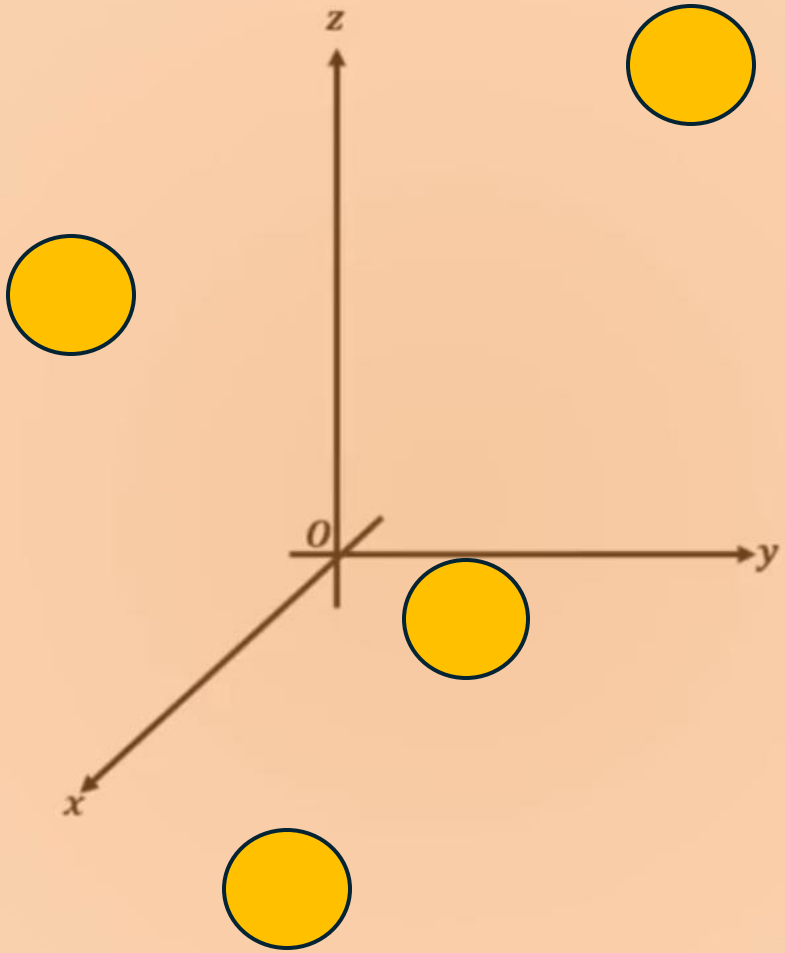


Soft prompts



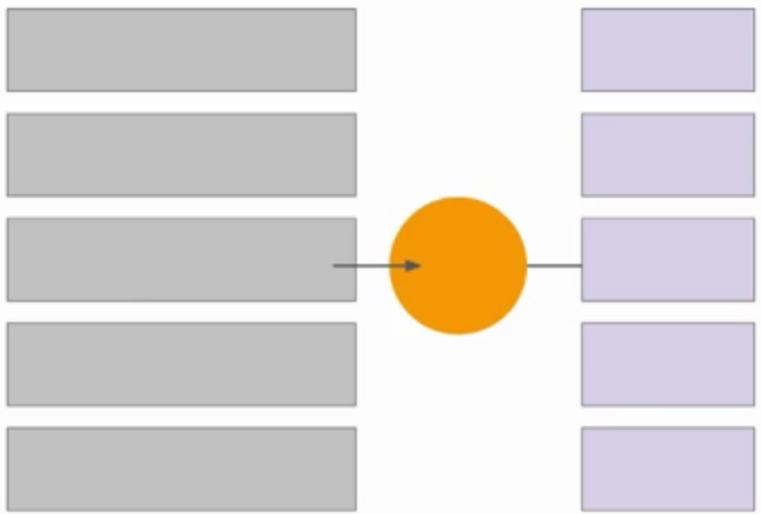
Embeddings of each token exist at unique point in multi-dimensional space

Soft prompts



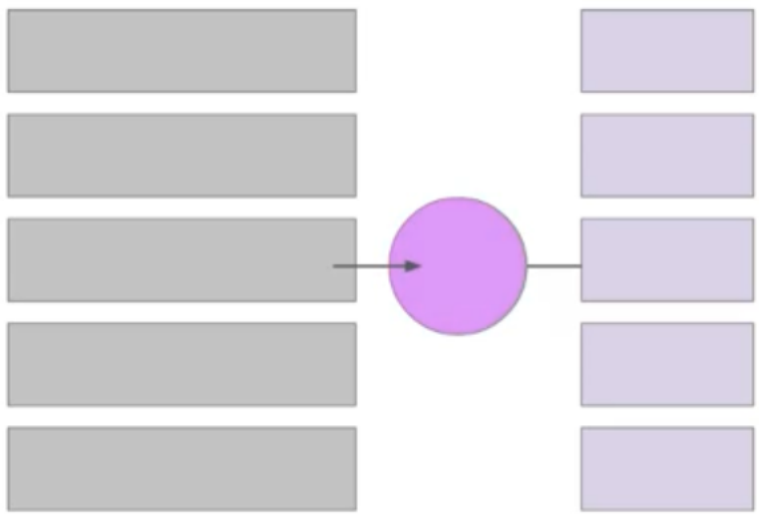
Full Fine-tuning vs prompt tuning

Weights of model updated during training



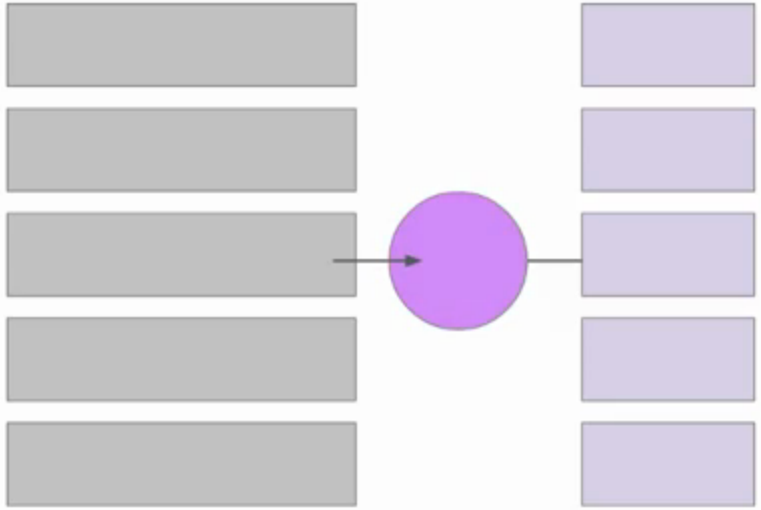
Full Fine-tuning vs prompt tuning

Weights of model updated during training



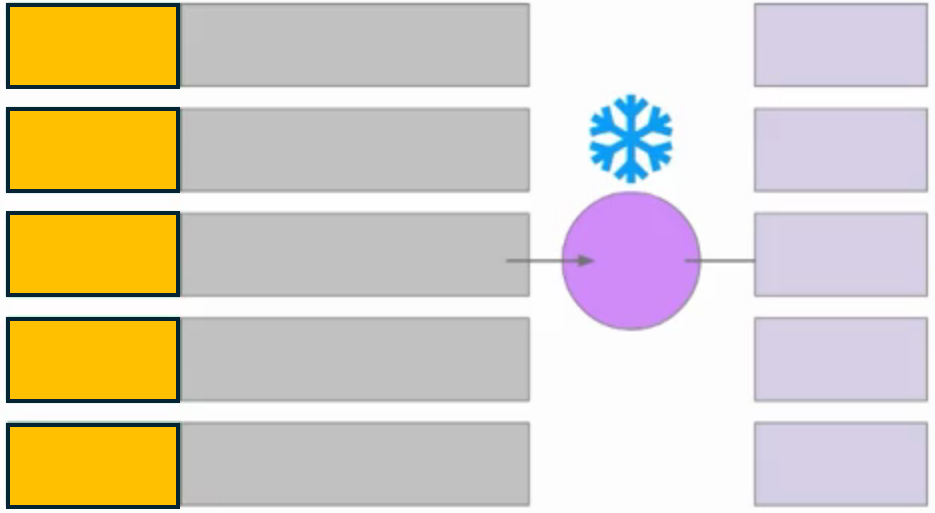
Full Fine-tuning vs prompt tuning

Weights of model updated during training



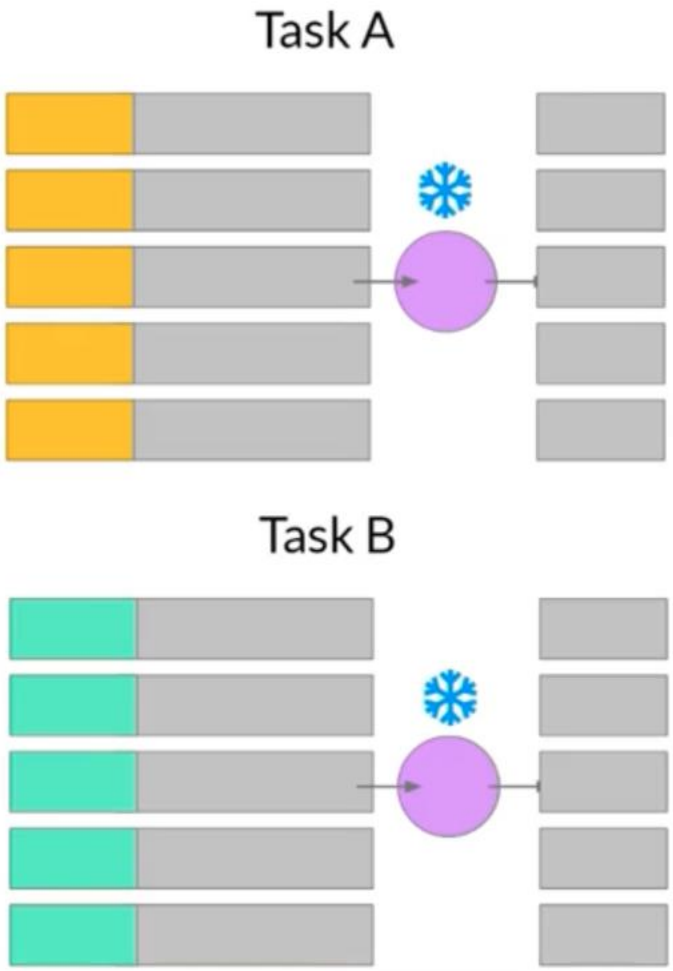
Millions to Billions of parameter updated

Weights of model frozen and soft prompt trained

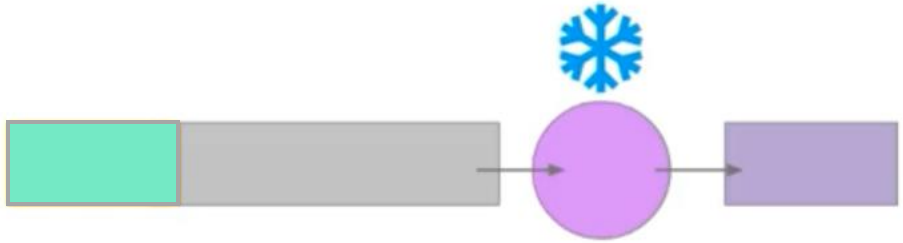


10K – 100K of parameters updated

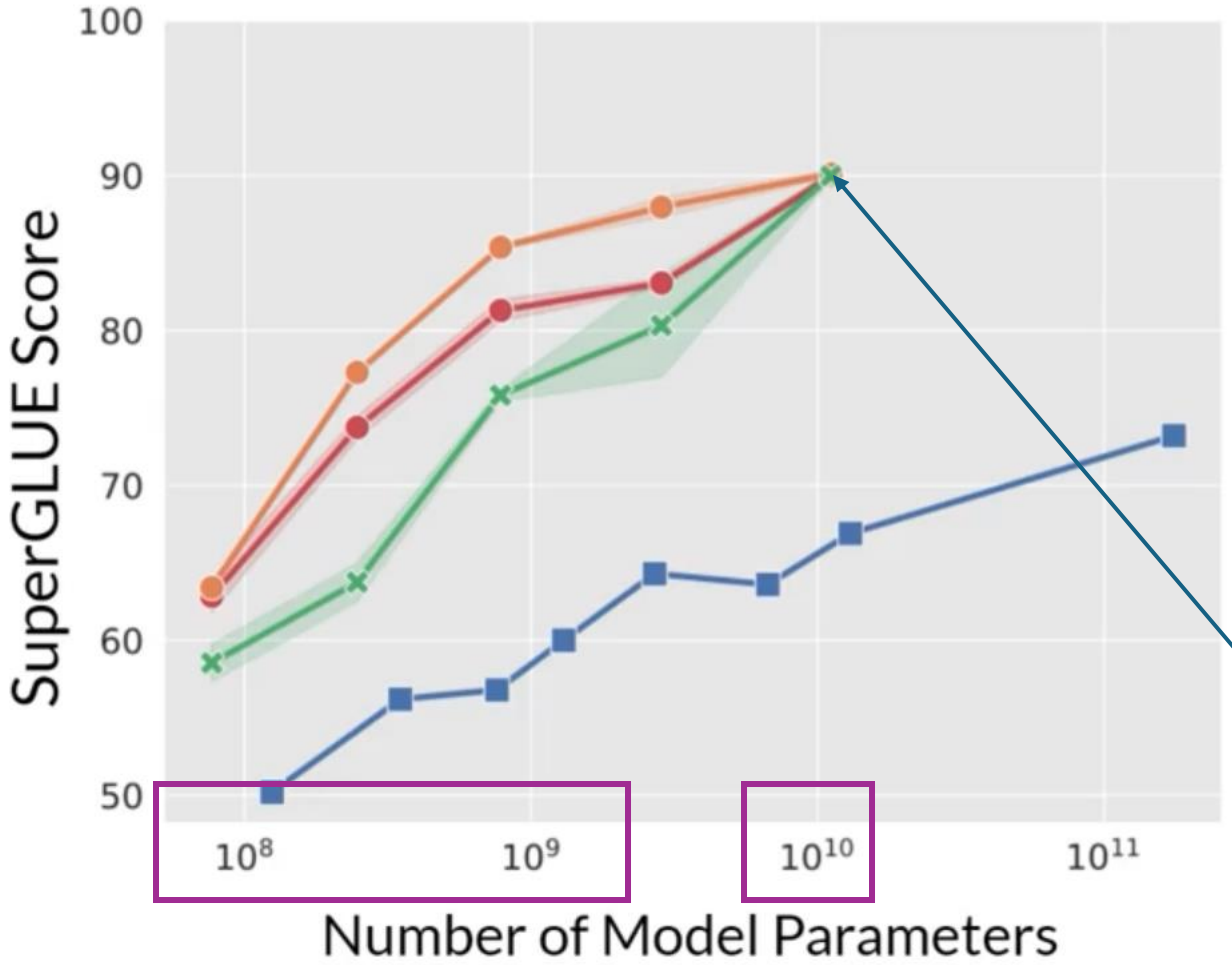
Prompt tuning for multiple tasks



Switch out soft prompt at inference time to change task!



Performance of prompt tuning

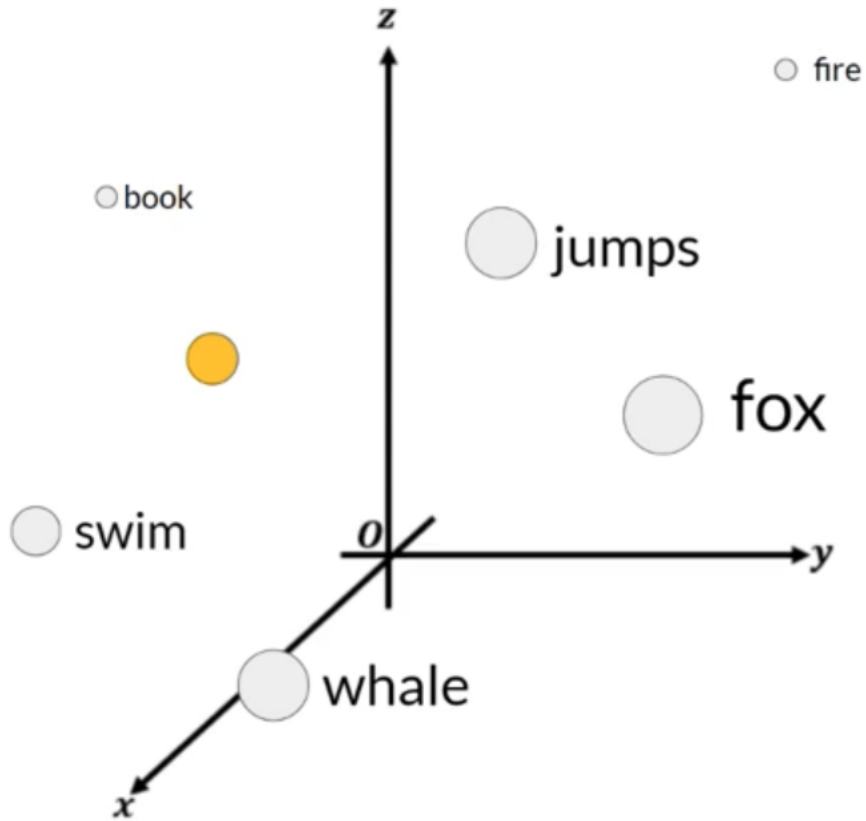


- Full Fine-tuning
- Multi-task Fine-tuning
- Prompt tuning
- Prompt engineering

Prompt tuning can be as effective as full Fine-tuning for larger models!

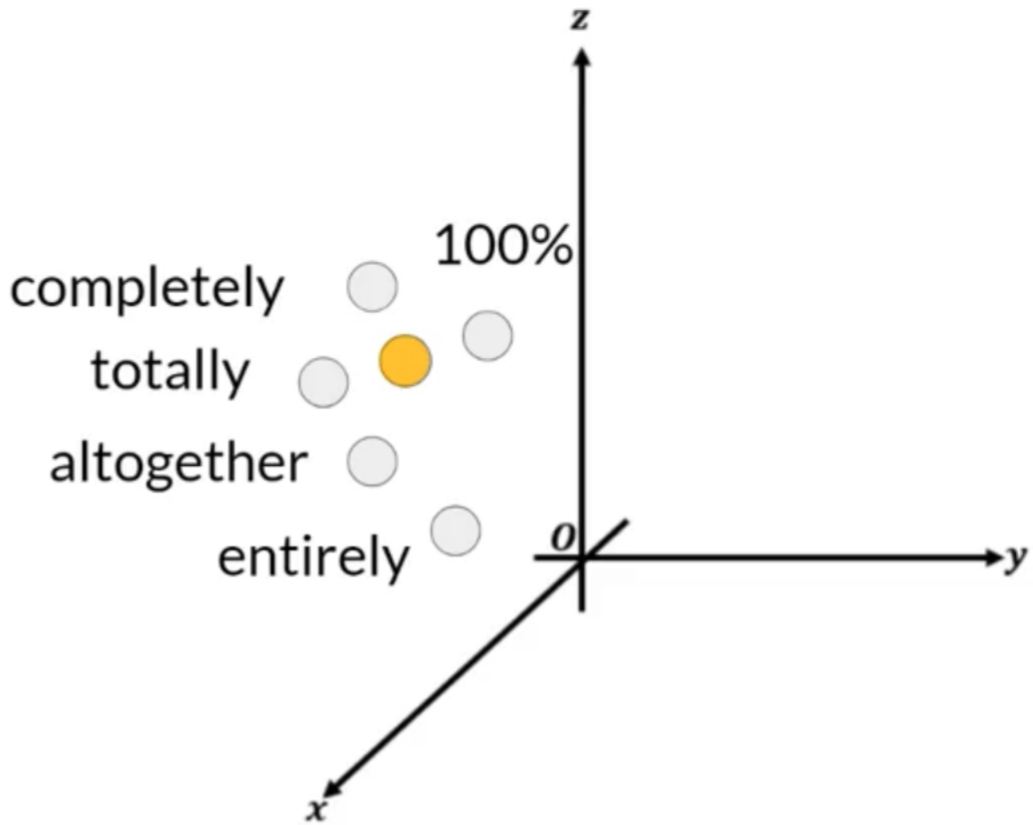
Source: Lester et al. 2021, "The Power of Scale for Parameter-Efficient Prompt Tuning"

Interpretability of soft prompts



Trained soft-prompt embedding does not correspond to a known token…

Interpretability of soft prompts



…but nearest neighbors form a semantic group with similar meanings.

PEFT methods summary

Selective

Select subset of initial LLM parameters to fine-tune

Reparameterization

Reparameterize model weights using a low-rank representation

LoRA

Additive

Add trainable layers or parameters to model

Adapters

Soft Prompts

Prompt Tuning

Key takeaways

- Instruction fine-tuning
  - Prompt templates and data sets for FLAN-T5
- PEFT
  - LoRA
  - Prompt Tuning
- Quantization + LoRA = QLoRA

ROUGE-L-sum

The **ROUGE-L-SUM** score is a variant of the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metric,

- which is widely used to evaluate the quality of generated text,
- especially in tasks like summarization,
- where the goal is to measure how well the generated summary matches a reference summary.

Key Components

- **ROUGE-L** focuses on the **Longest Common Subsequence (LCS)**:
  - Instead of comparing n-grams like ROUGE-N, ROUGE-L evaluates the longest subsequence of words that appears in the same order in both the generated and reference texts.
  - LCS captures the sentence-level structure, rewarding fluency and coherence.
- **SUM (Summarization-specific variant)**:
  - ROUGE-L-SUM is tailored for summarization tasks, ensuring the metric emphasizes overlap in content that is crucial for summaries, such as main ideas and key details.
  - It typically considers **sentence-level LCS** rather than a document-level LCS.

Key Components

- **Components Evaluated**:
  - **Precision**: How much of the LCS in the generated text aligns with the reference text.

$$P = \frac{LCS}{\text{Length of generated summary}}$$

  - **Recall**: How much of the reference text's content is captured in the LCS.

$$R = \frac{LCS}{\text{Length of reference summary}}$$

  - **F1 Score**: The harmonic mean of precision and recall, offering a balanced evaluation.

$$F1 = \frac{2 \times P \times R}{P + R}$$

Applications

- Used in research papers and benchmarks for summarization models (e.g., abstractive or extractive summarization models).
- A good ROUGE-L-SUM score indicates that the generated summary captures important content in a structured and coherent way.