Modern Deep Learning with PyTorch

7. Finetuning LLMs (4:55 - 5:25 pm)

Schedule

1. Introduction to Deep Learning (1:30 - 2:00 pm)
2. Understanding the PyTorch API (2:00 - 2:30 pm)
3. Training Deep Neural Networks (2:30 - 3:00 pm)

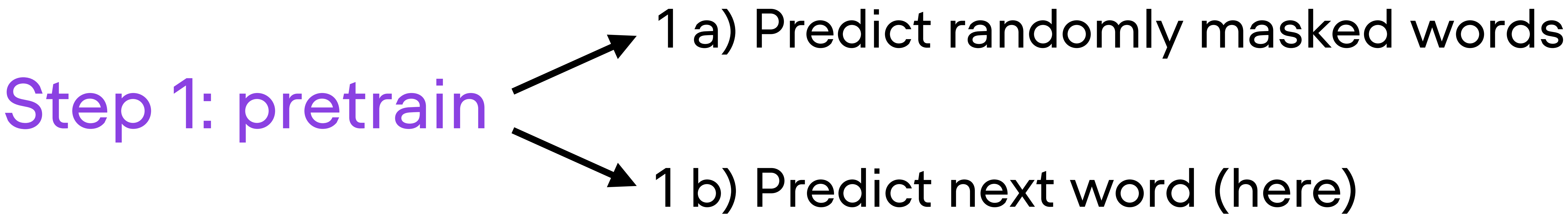
10 Min Break

4. Accelerating PyTorch Model Training (3:10 - 3:45 pm)
5. Organizing PyTorch Code (3:45 - 4:15 pm)
6. More Tips and Techniques (4:15 - 4:45 pm)

10 Min Break

7. Finetuning LLMs (4:55 - 5:25 pm)
8. Wrap Up & Questions (5:25 - 5:30 pm)

Self-supervised pretraining



Step 1: pretrain

1 a) Predict randomly masked words

1 b) Predict next word (here)

Step 2: fine-tune

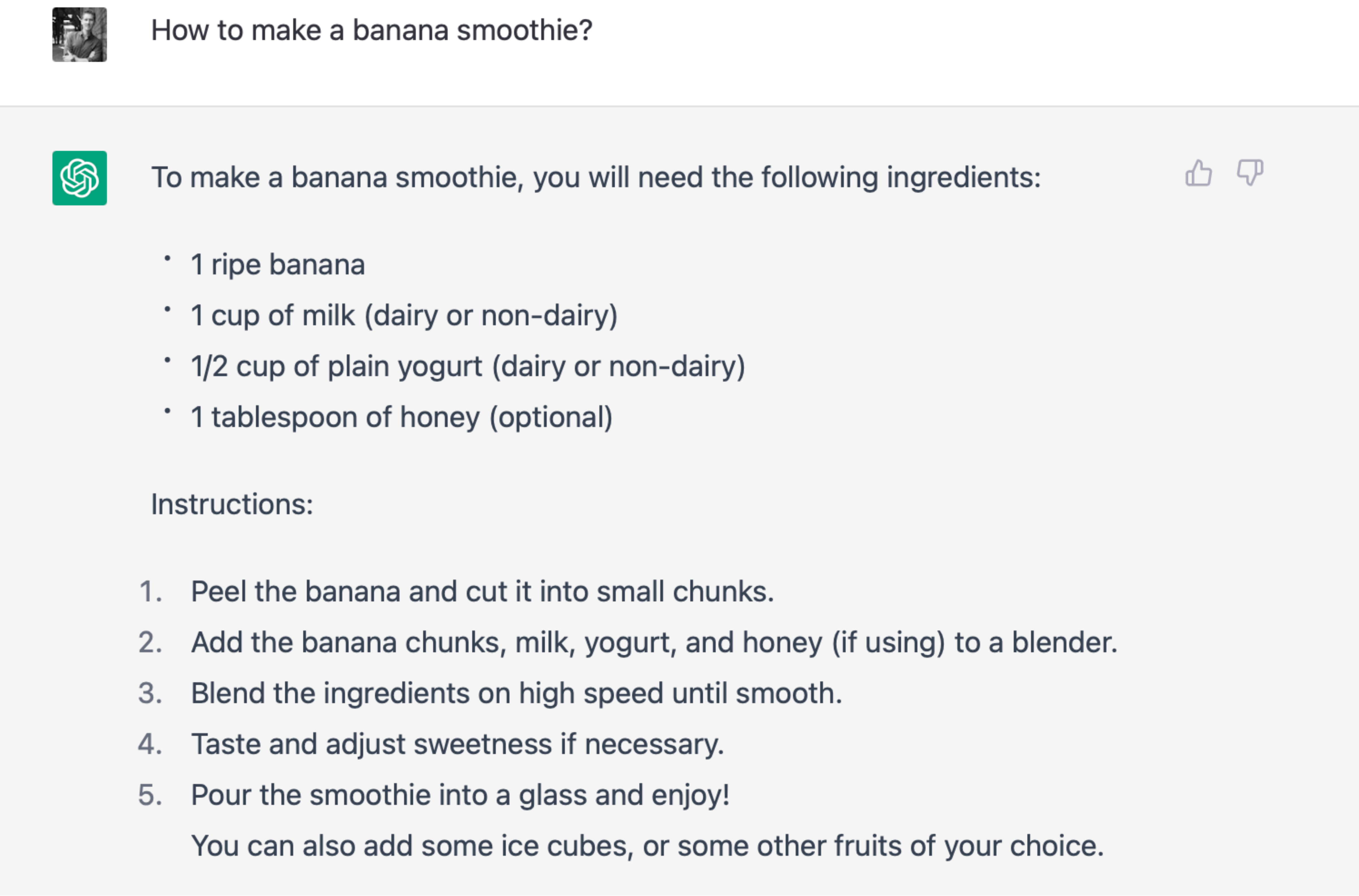
	Date
GPT 1	2018
GPT 2	2019
GPT 3	2020
InstructGPT & ChatGPT	2022

	Date	Size
GPT 1	2018	110 million
GPT 2	2019	1.5 billion
GPT 3	2020	175 billion
InstructGPT & ChatGPT	2022	GPT 3-based

	Date	Size	Paper
GPT 1	2018	110 million	Improving Language Understanding by Generative Pre-Training
GPT 2	2019	1.5 billion	Language Models are Unsupervised Multitask Learners
GPT 3	2020	175 billion	Language Models are Few-Shot Learners
InstructGPT & ChatGPT	2022	GPT 3-based	Aligning Language Models to Follow Instructions

GPT models are generative models (as opposed to classifiers)

GPT models are generative models (as opposed to classifiers)



How to make a banana smoothie?

To make a banana smoothie, you will need the following ingredients:

- 1 ripe banana
- 1 cup of milk (dairy or non-dairy)
- 1/2 cup of plain yogurt (dairy or non-dairy)
- 1 tablespoon of honey (optional)

Instructions:

1. Peel the banana and cut it into small chunks.
2. Add the banana chunks, milk, yogurt, and honey (if using) to a blender.
3. Blend the ingredients on high speed until smooth.
4. Taste and adjust sweetness if necessary.
5. Pour the smoothie into a glass and enjoy!
   You can also add some ice cubes, or some other fruits of your choice.

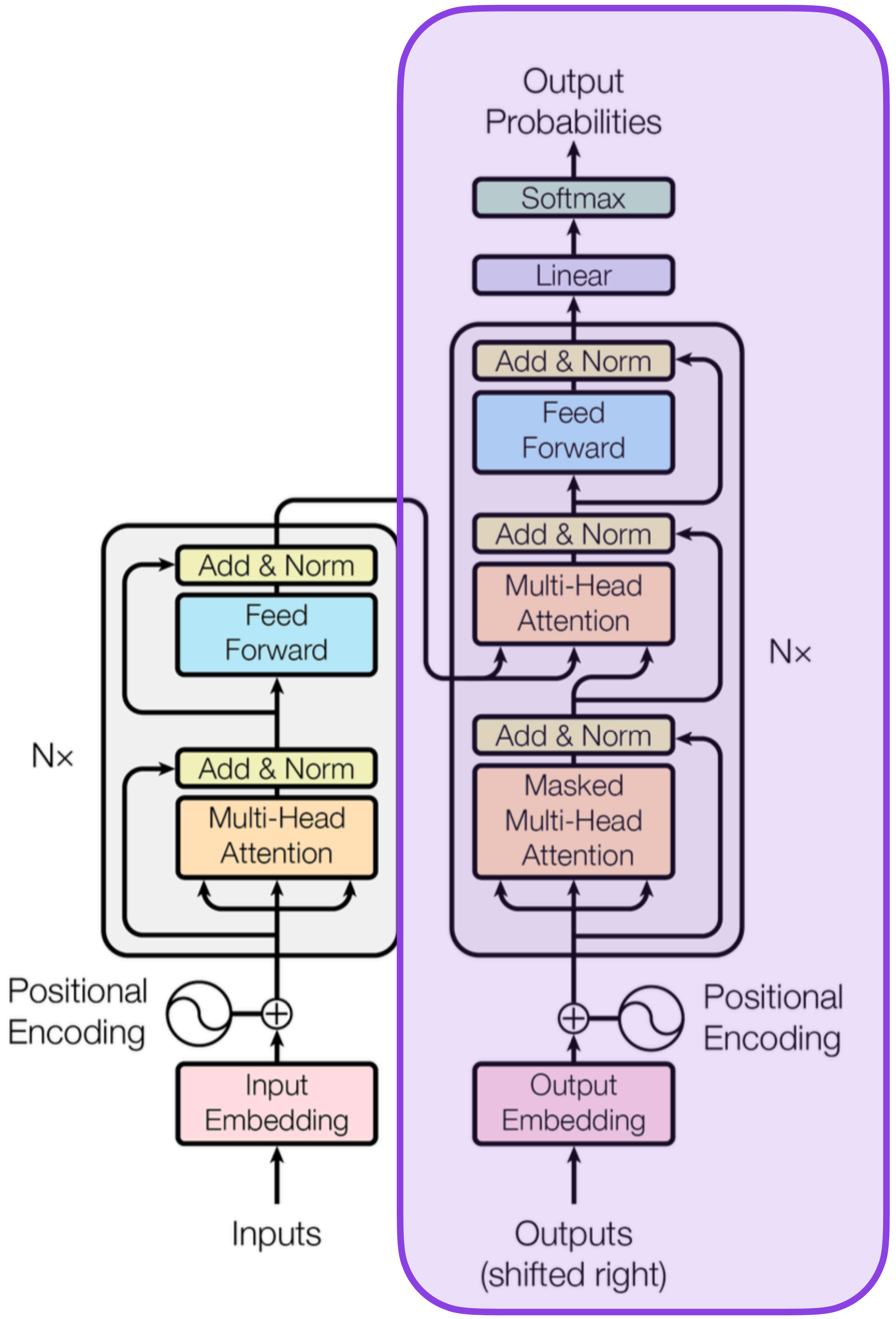
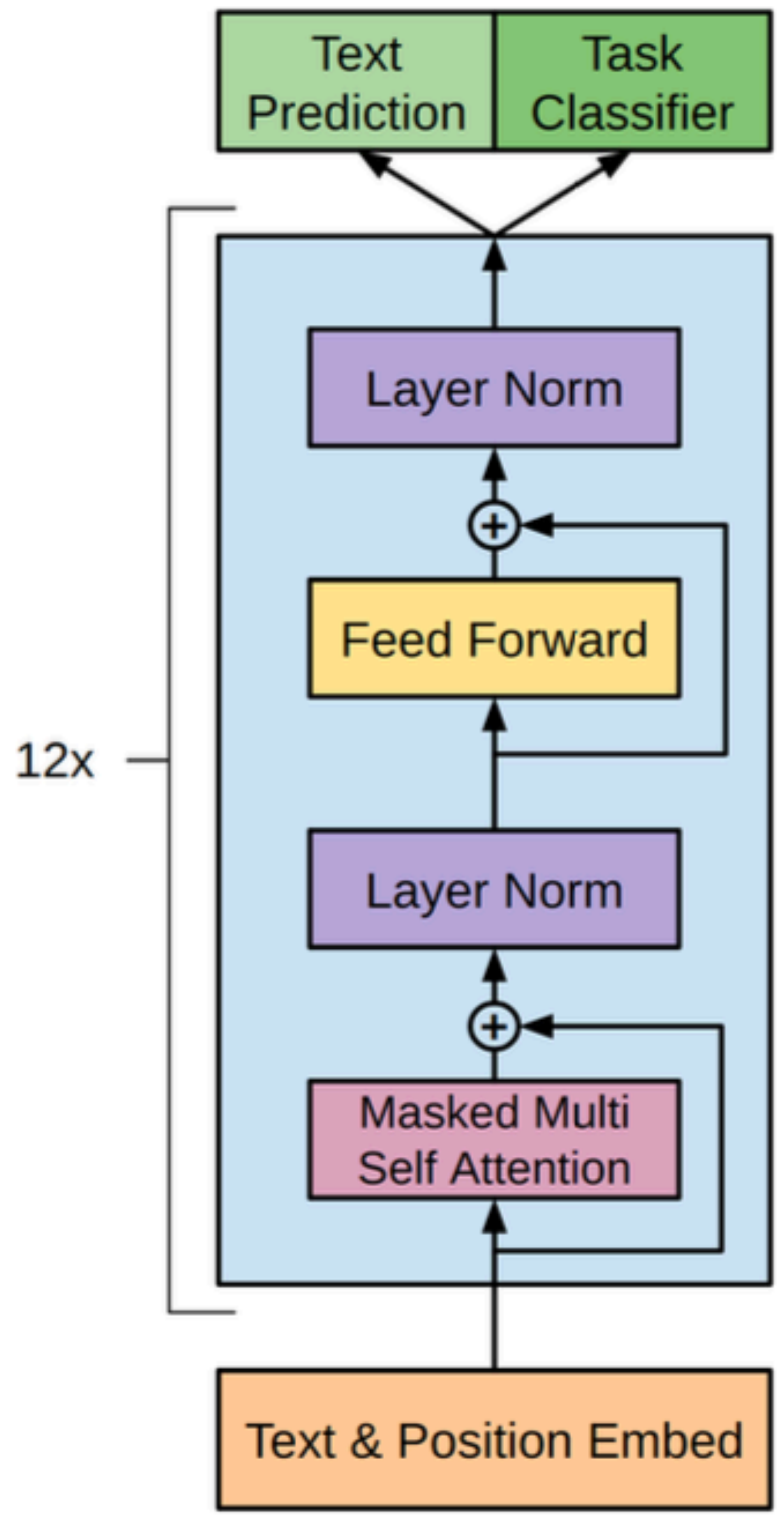
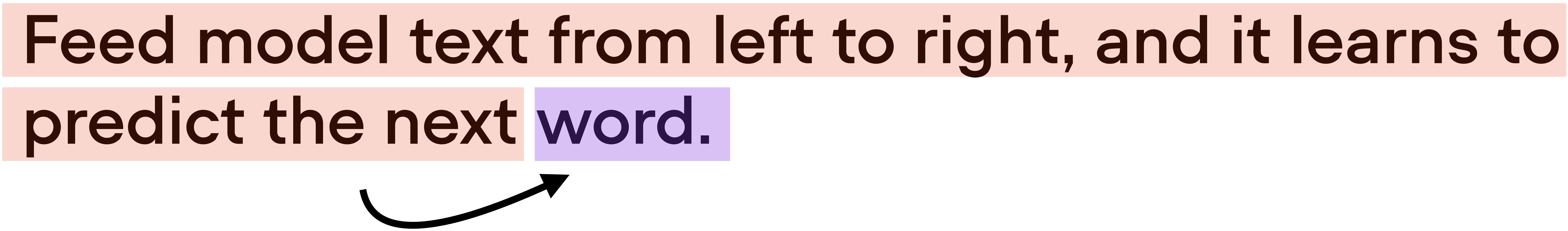


Figure 1: The Transformer - model architecture.

GPT is essentially the decoder part of the original transformer



https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf



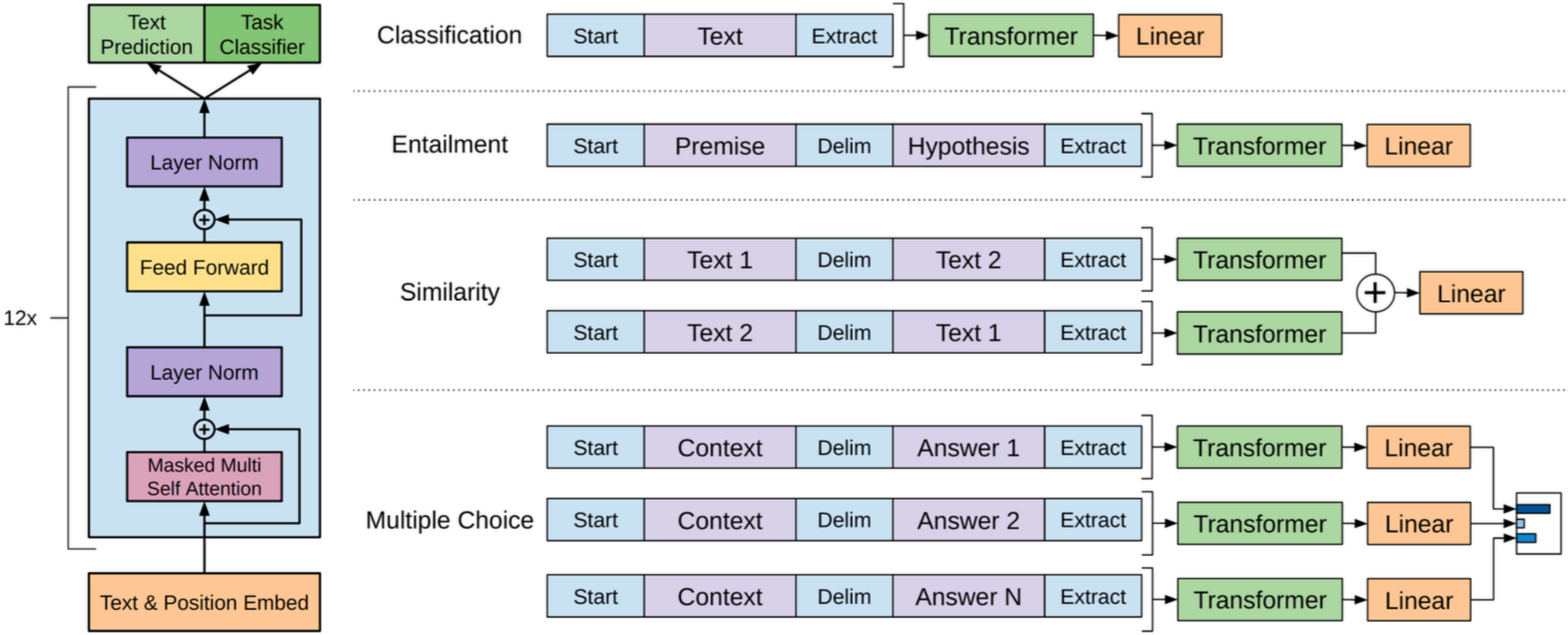
Feed model text from left to right, and it learns to predict the next word.

Self-supervised pretraining

Step 1: pretrain → Predict next word

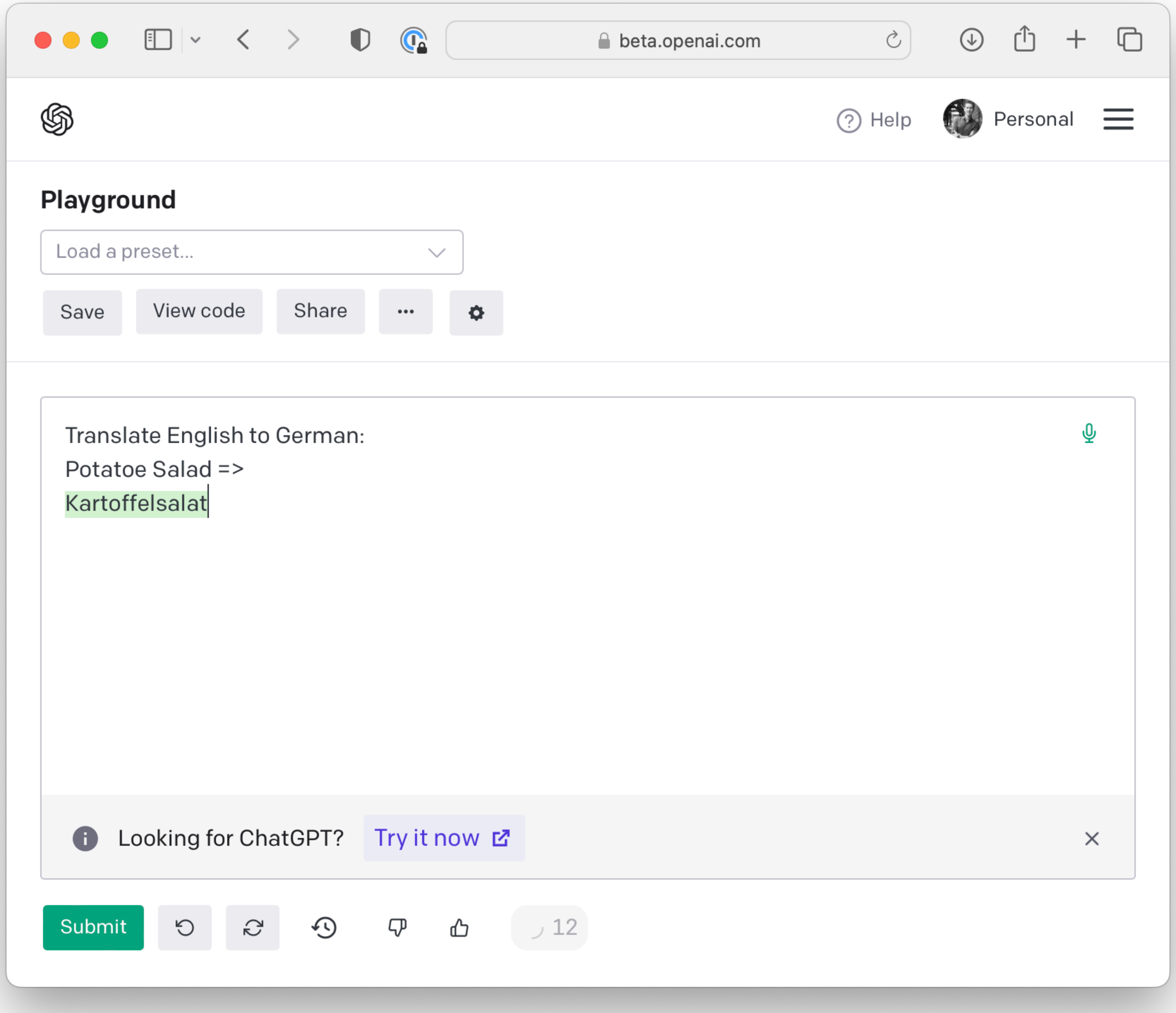
Step 2: fine-tune

Fine-tune for target task



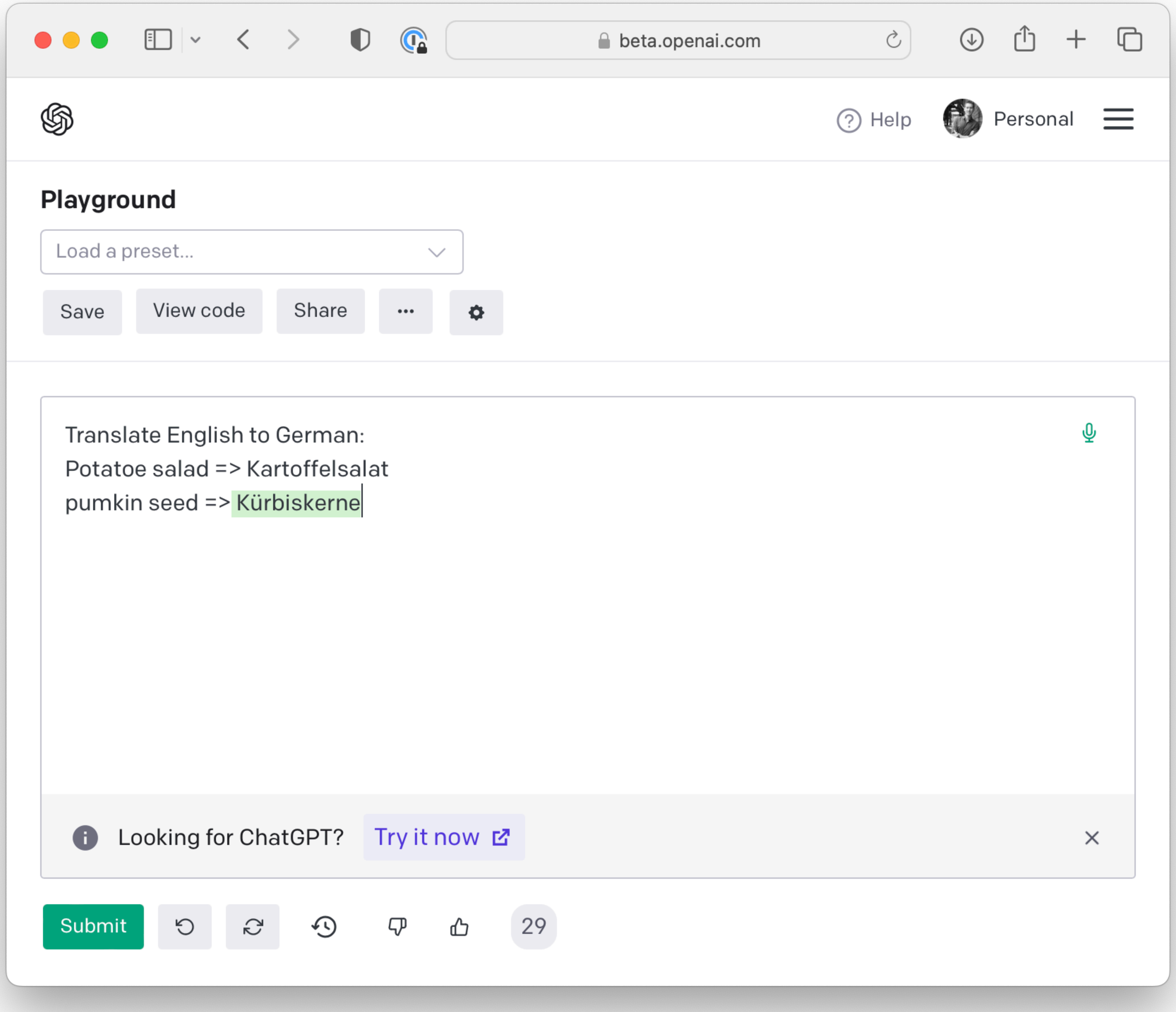
GPT 2 and 3 focused on zero- and few-shot learning via in-context learning

Zero-shot



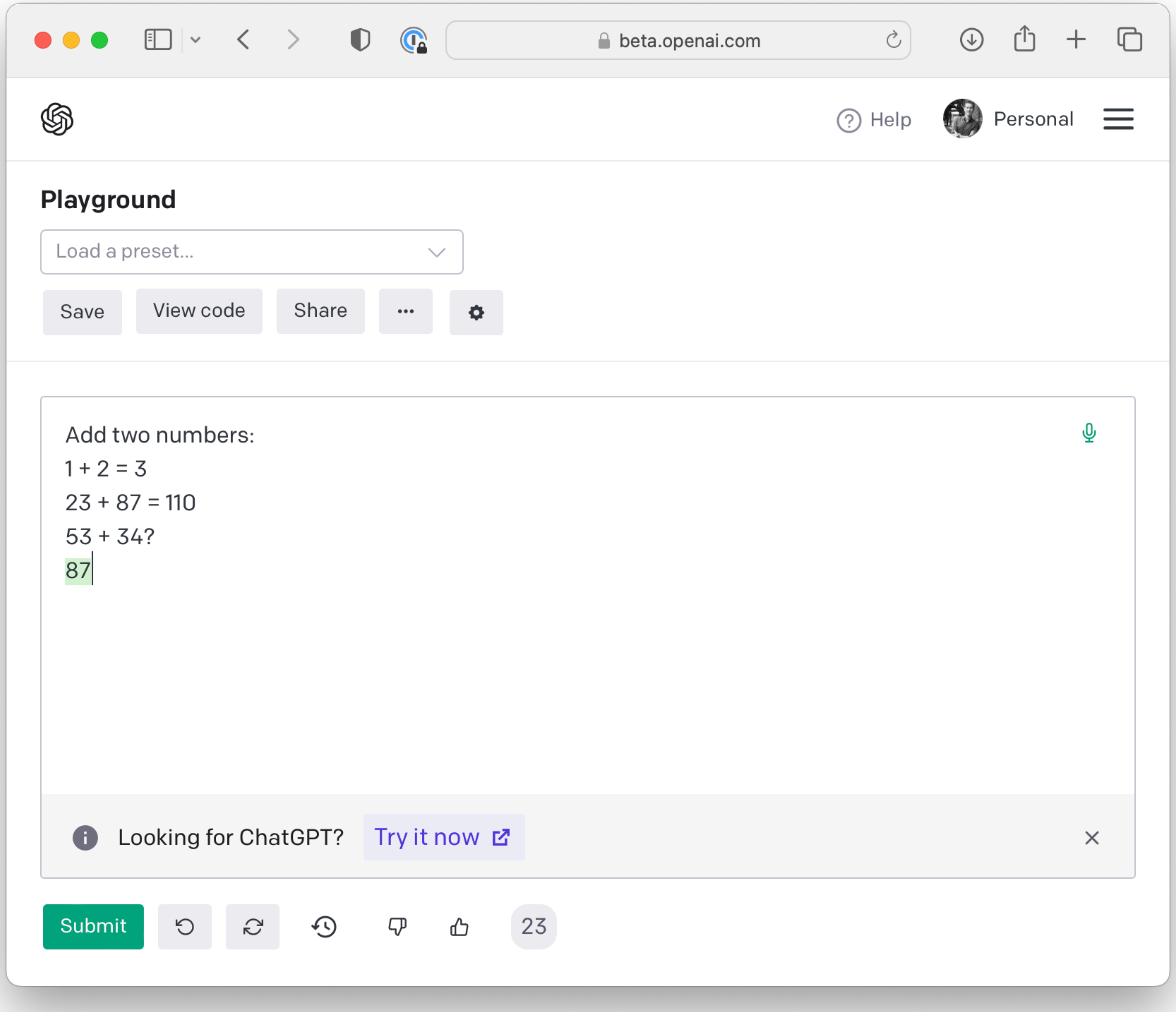
Translate English to German:
Potatoe Salad =>
Kartoffelsalat

One-shot



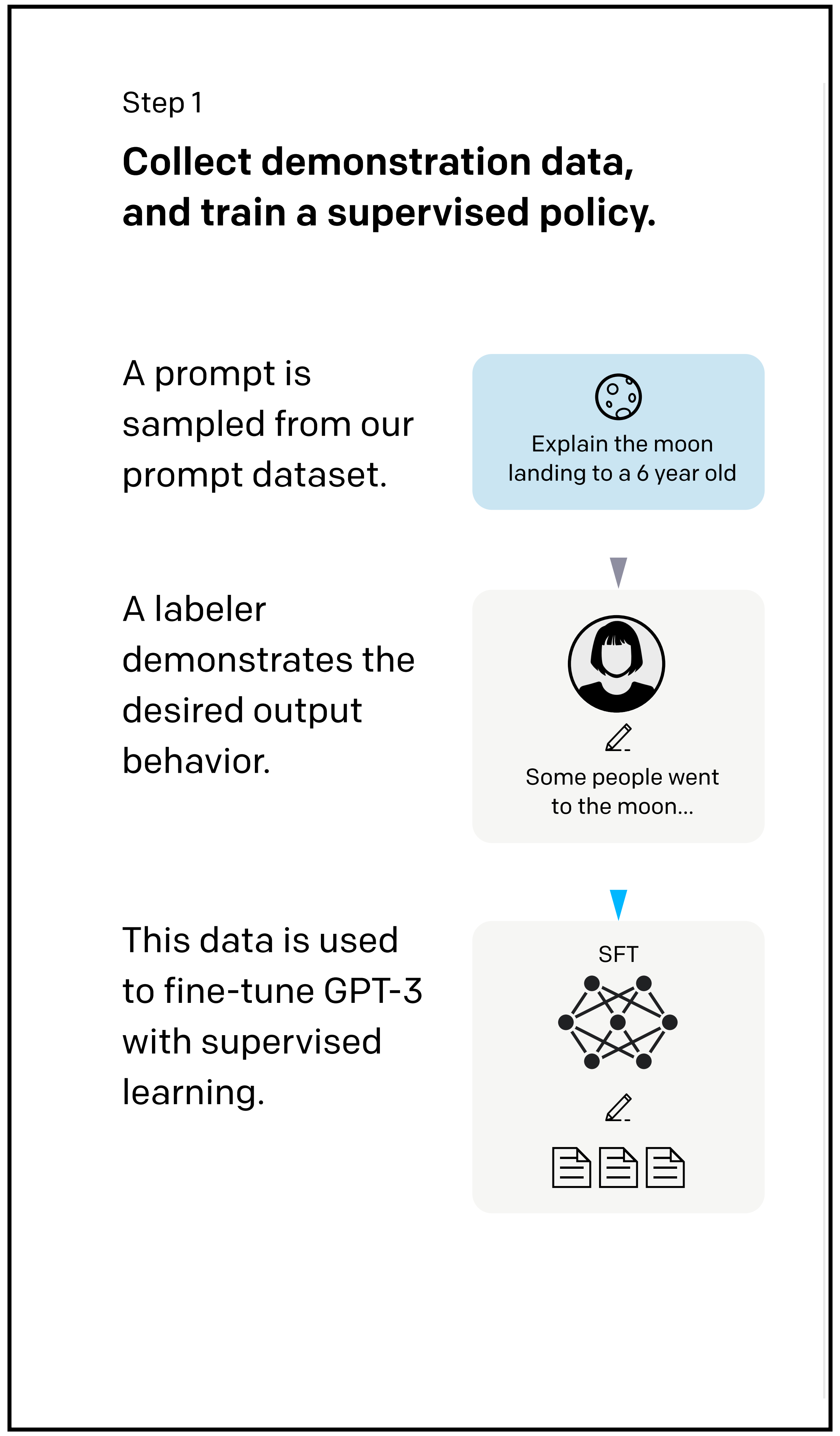
Translate English to German:
Potatoe salad => Kartoffelsalat
pumkin seed => Kürbiskerne

Few-shot



Add two numbers:
1 + 2 = 3
23 + 87 = 110
53 + 34?
87

InstructGPT and ChatGPT are Additionally Trained on Human Feedback



Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

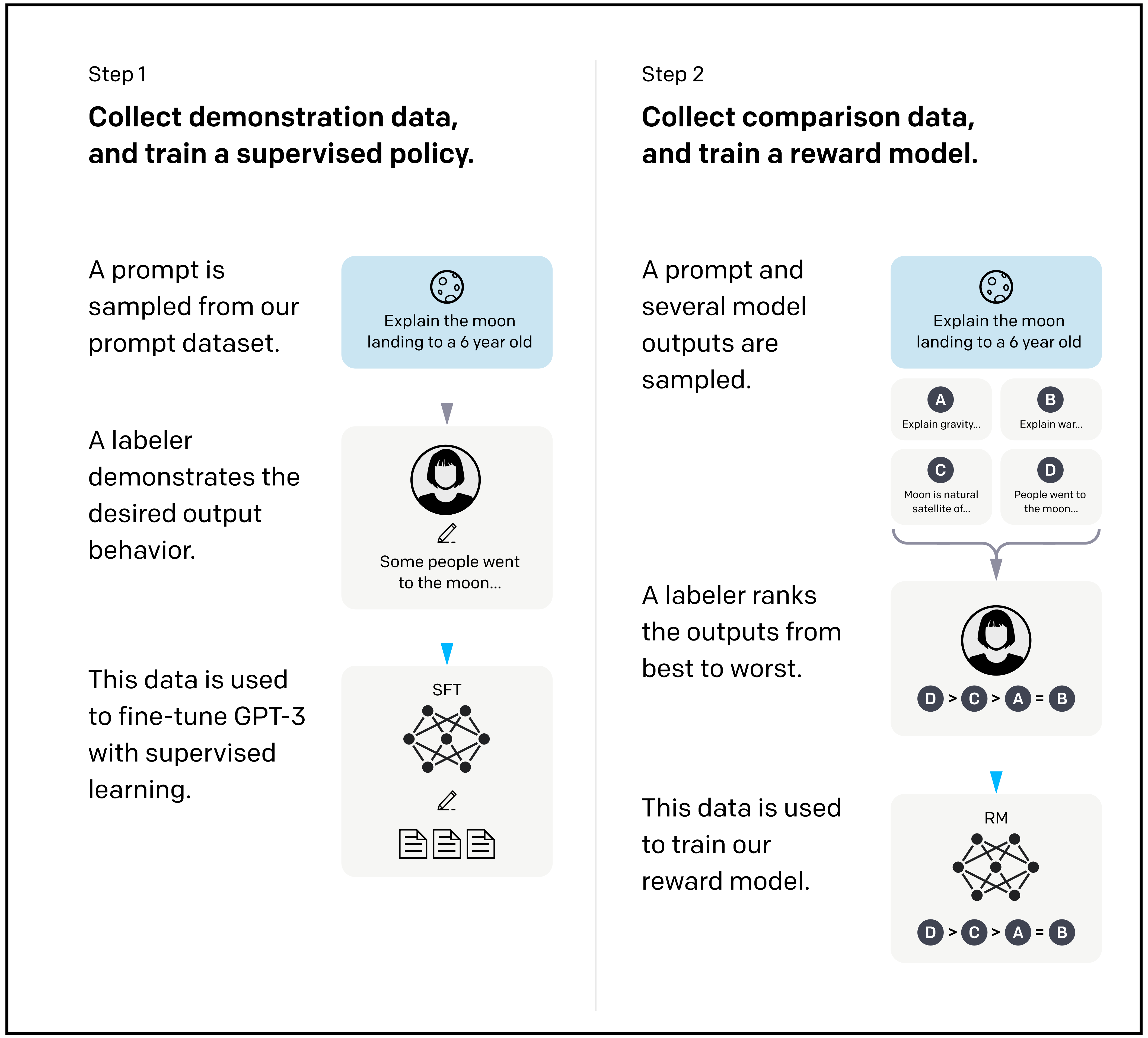
A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

Explain the moon landing to a 6 year old
Some people went to the moon...
SFT

Training language models to follow instructions with human feedback, https://arxiv.org/abs/2203.02155

InstructGPT and ChatGPT are Additionally Trained on Human Feedback



Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

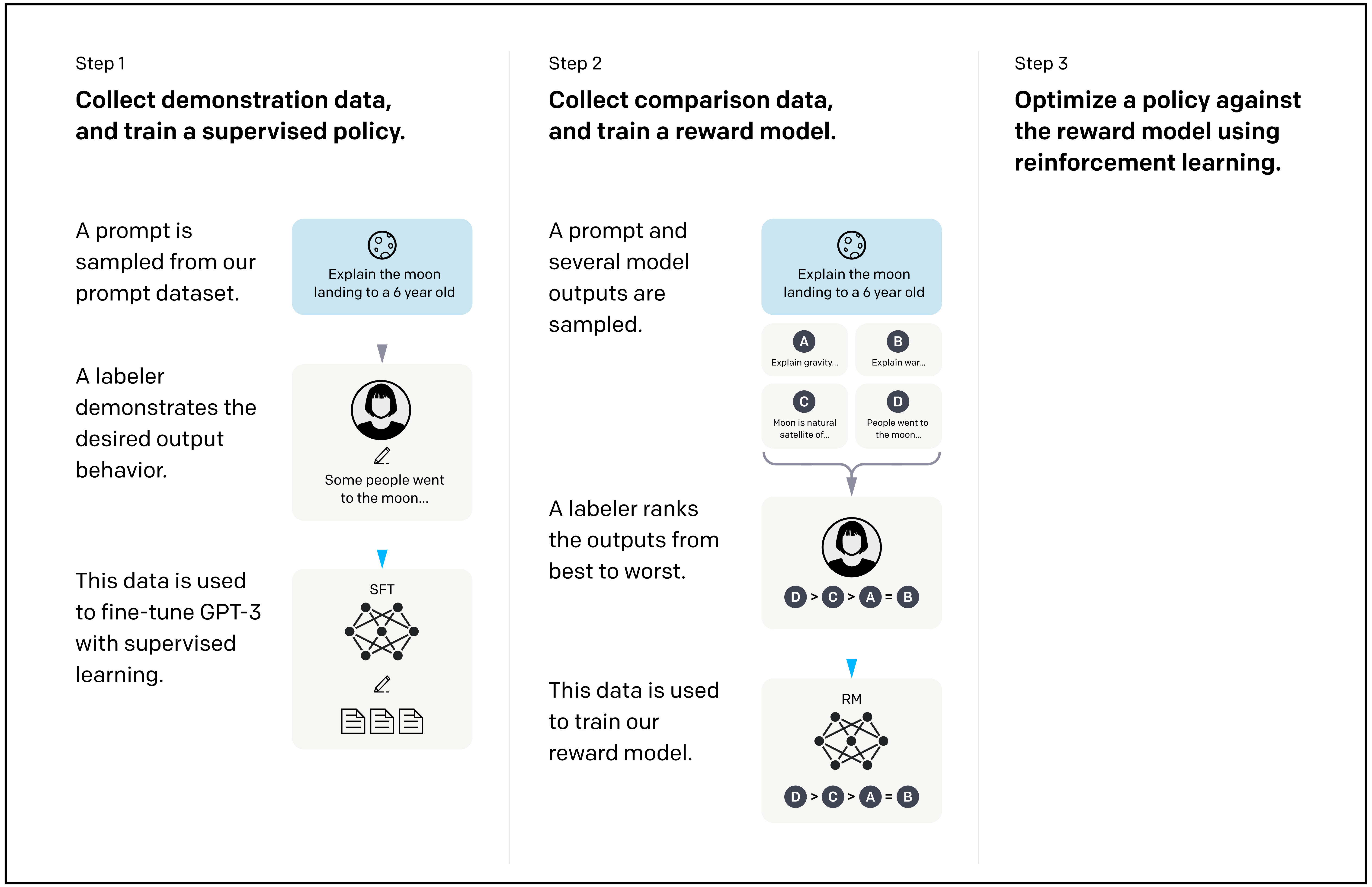
A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Explain the moon landing to a 6 year old
Some people went to the moon...
SFT
Explain the moon landing to a 6 year old
A
Explain gravity...
B
Explain war...
C
Moon is natural satellite of...
D
People went to the moon...
D > C > A = B
RM
D > C > A = B

Training language models to follow instructions with human feedback, https://arxiv.org/abs/2203.02155

InstructGPT and ChatGPT are Additionally Trained on Human Feedback



Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Step 3

Optimize a policy against the reward model using reinforcement learning.

Explain the moon landing to a 6 year old
Some people went to the moon...
SFT
Explain the moon landing to a 6 year old
A Explain gravity...
B Explain war...
C Moon is natural satellite of...
D People went to the moon...
D > C > A = B
RM
D > C > A = B

Training language models to follow instructions with human feedback, https://arxiv.org/abs/2203.02155

Today, transformers (large language models) are also used for …

- Classification (e.g., BERT)
- Various text summarization and generation tasks (e.g., GPT)
- Conversational chatbots (e.g., ChatGPT)
- Protein structure prediction from sequence data (e.g., AlphaFold 2)

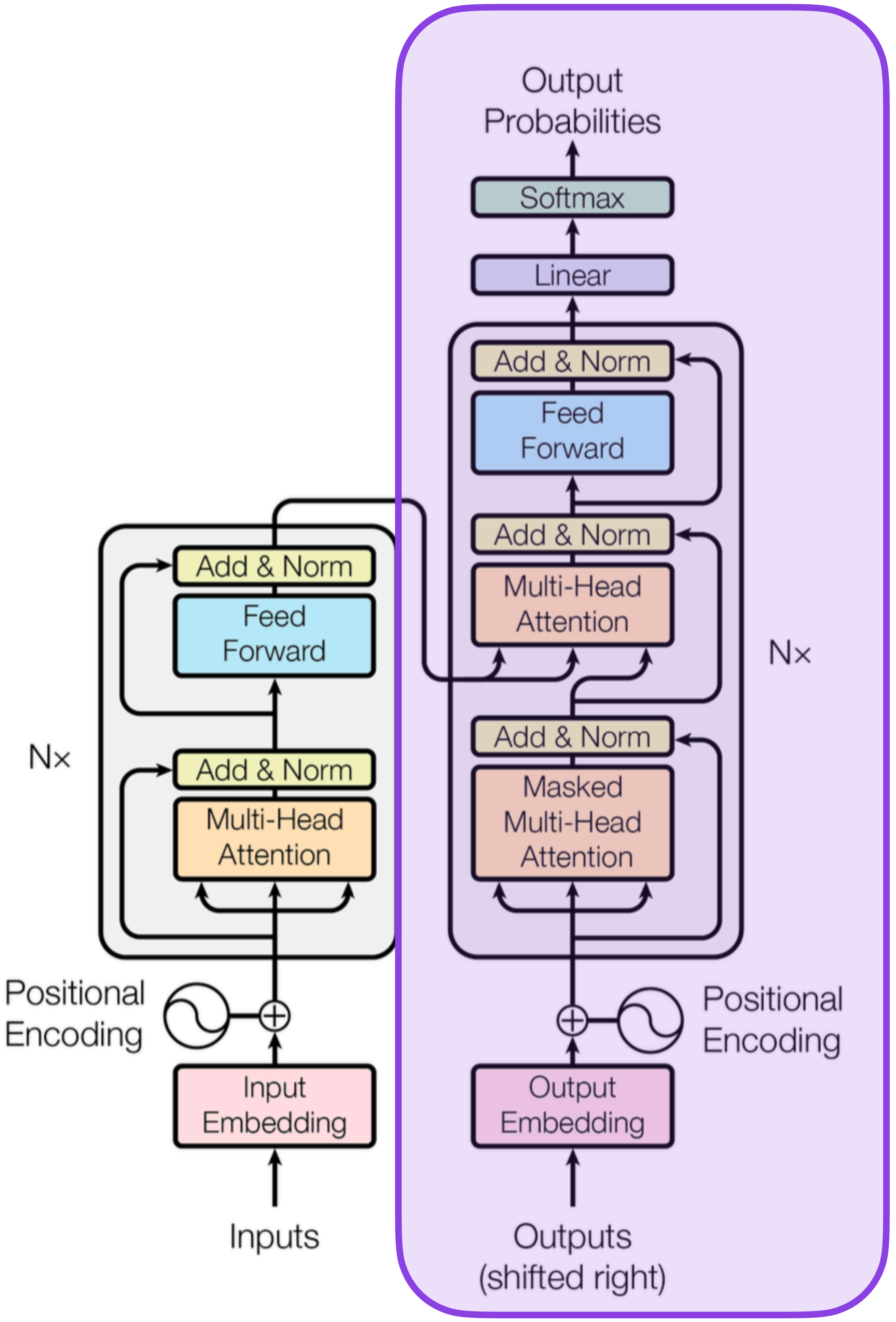
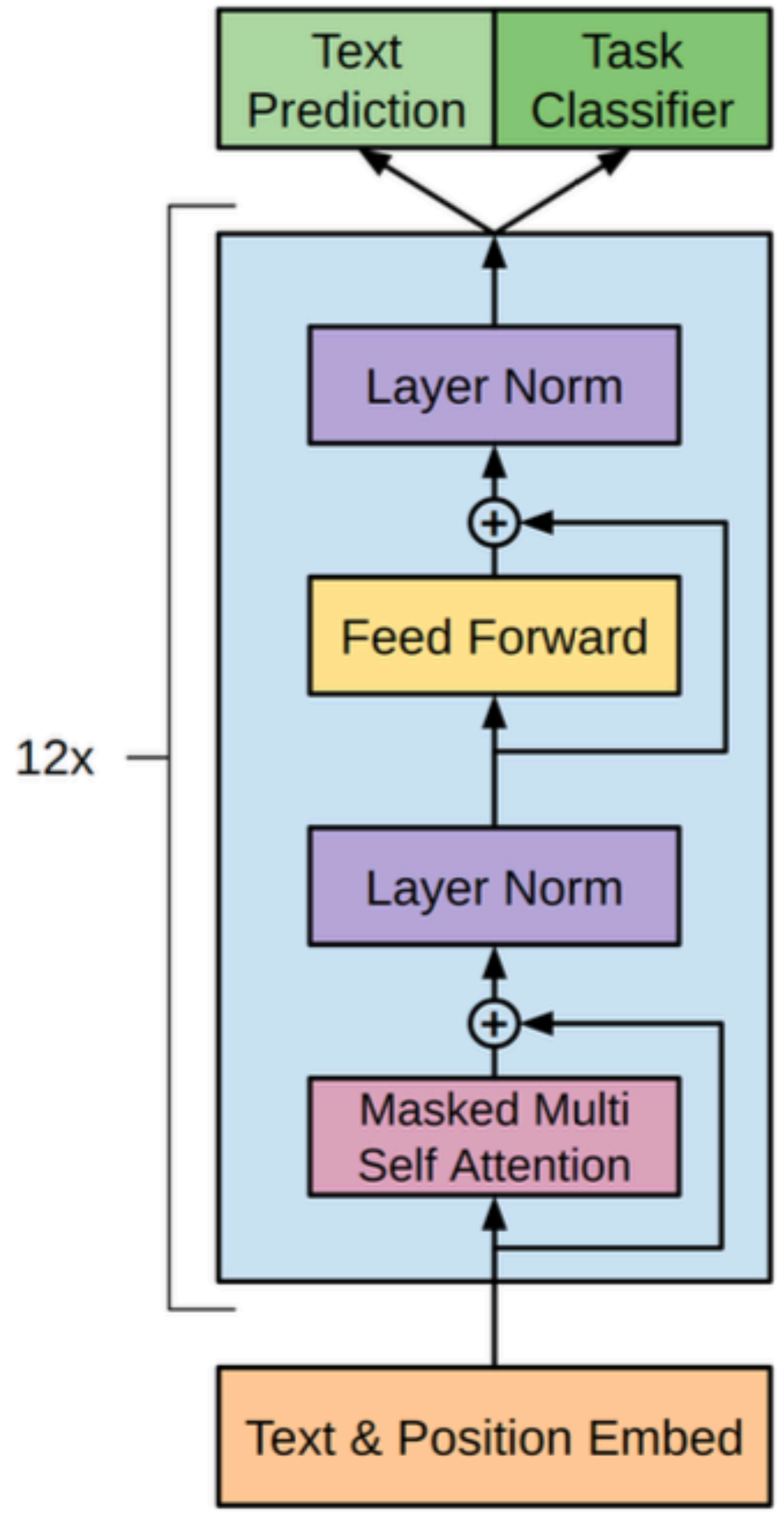


Figure 1: The Transformer - model architecture.

GPT Recap

GPT is essentially the decoder part of the original transformer



https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

GPT Recap

Self-supervised pretraining

Step 1: pretrain → Predict next word (unidirectional self-attention)

Step 2: fine-tune

BERT

Self-supervised pretraining

Step 1: pretrain → ~~Predict next word (unidirectional self-attention)~~

a) Predict randomly masked words (bidirectional / nondirectional)

b) Sentence-order prediction

Step 2: fine-tune

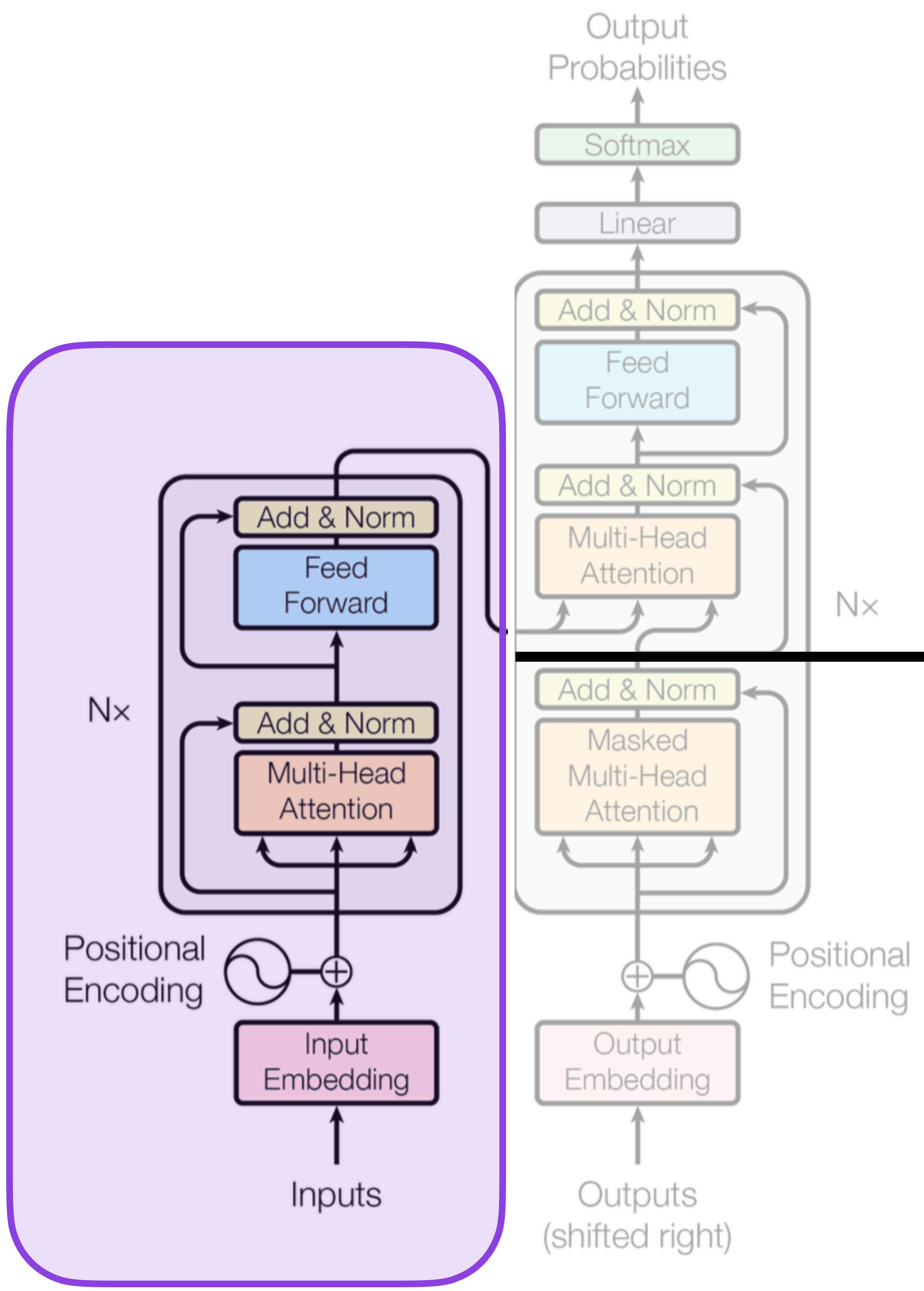
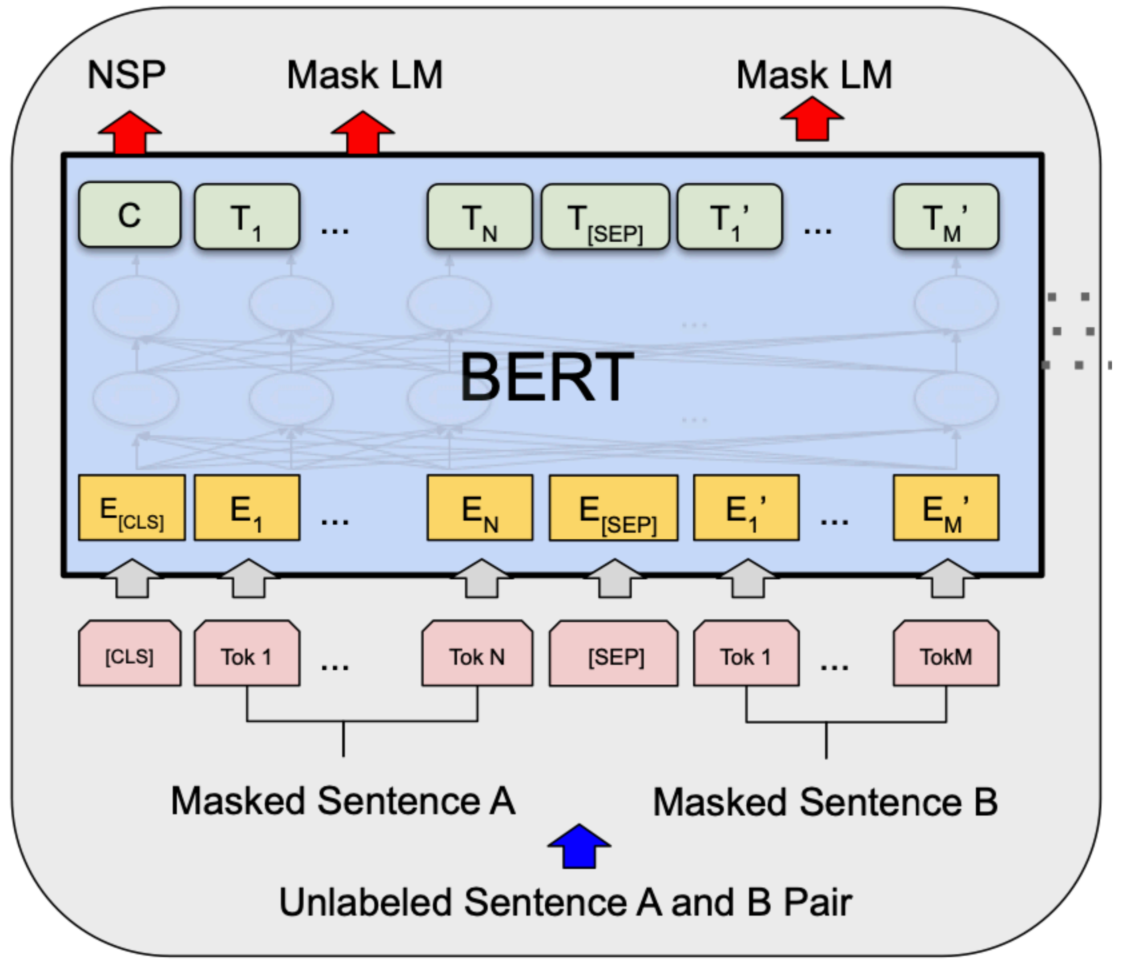


Figure 1: The Transformer - model architecture.

BERT is essentially the encoder part of the original transformer



BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,
https://arxiv.org/abs/1810.04805

Step 1: pretrain on large unlabeled dataset (learn a general language model)

a) Predict input sentence given randomly masked words

Input sentence: *The curious kitten deftly climbed the bookshelf*

↓

Pick 15% of the words randomly

The curious kitten deftly climbed the bookshelf

Input sentence: *The curious kitten deftly climbed the bookshelf*

↓

Pick 15% of the words randomly

The curious kitten deftly climbed the bookshelf

↓

- 80% of the time, replace with [MASK] token
- 10% of the time, replace with random token (e.g. ate)
- 10% of the time, keep unchanged

Step 1: pretrain on large unlabeled dataset (learn a general language model)

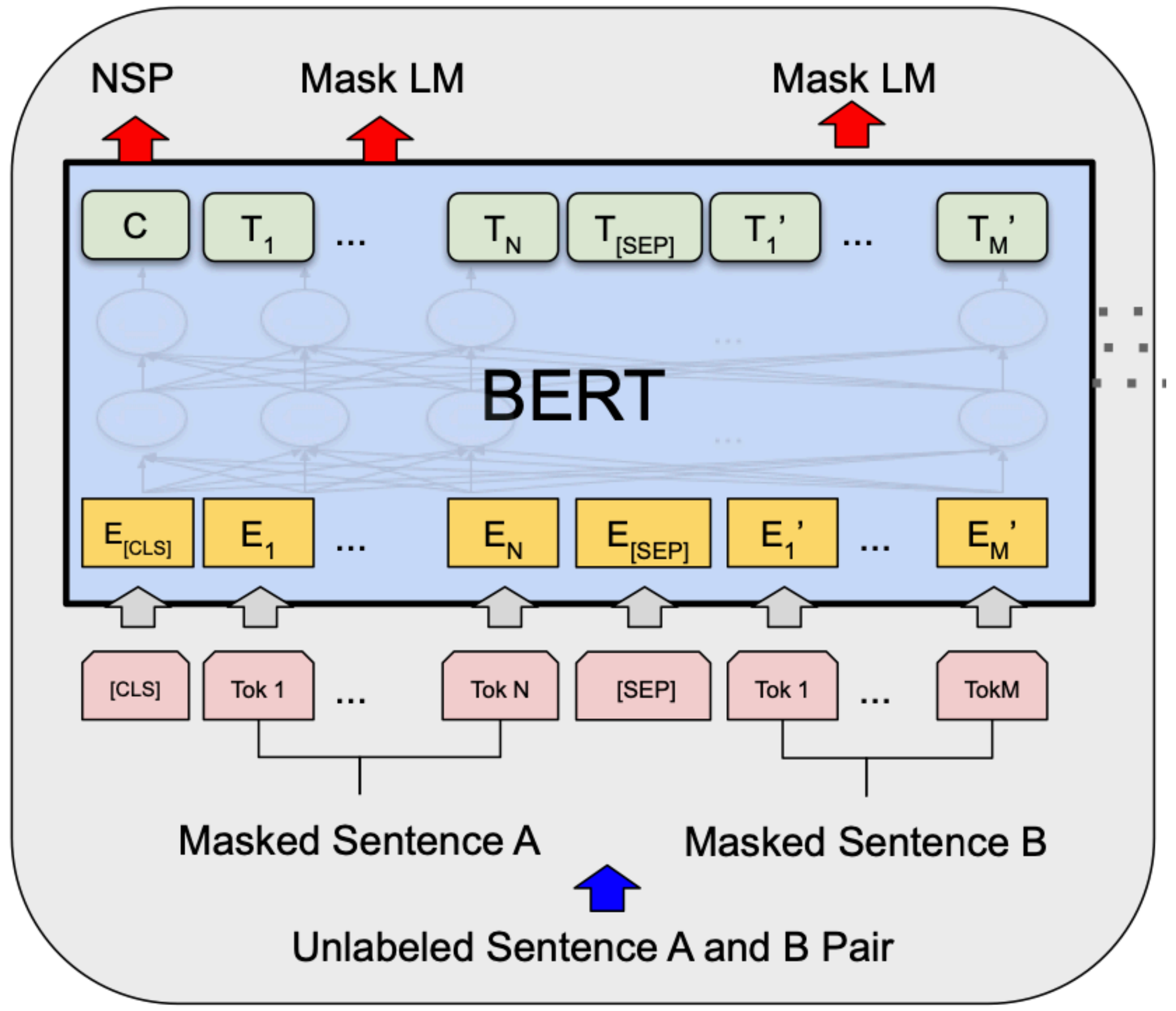
a) Predict input sentence given randomly masked words

b) Predict sentence order

b) Predict sentence order

[CLS] Sentence A [SEP] Sentence B

Placeholder for the IsNext=True / False label in the decoder output



b) Predict sentence order

[CLS] Toast is a simple yet delicious food [SEP] It’s often served with butter, jam, or honey.

IsNext = True

[CLS] It’s often served with butter, jam, or honey. [SEP] Toast is a simple yet delicious food.

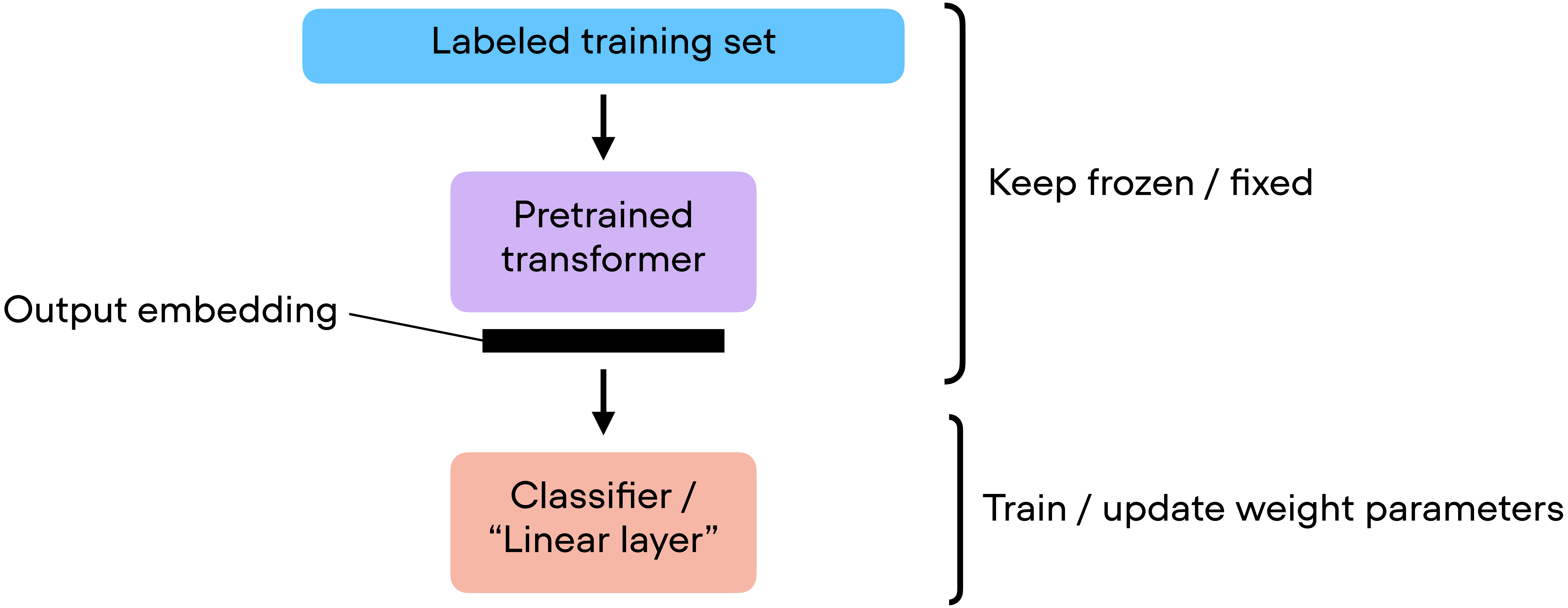
IsNext = False

2 ways of adopting a pretrained transformer for classification

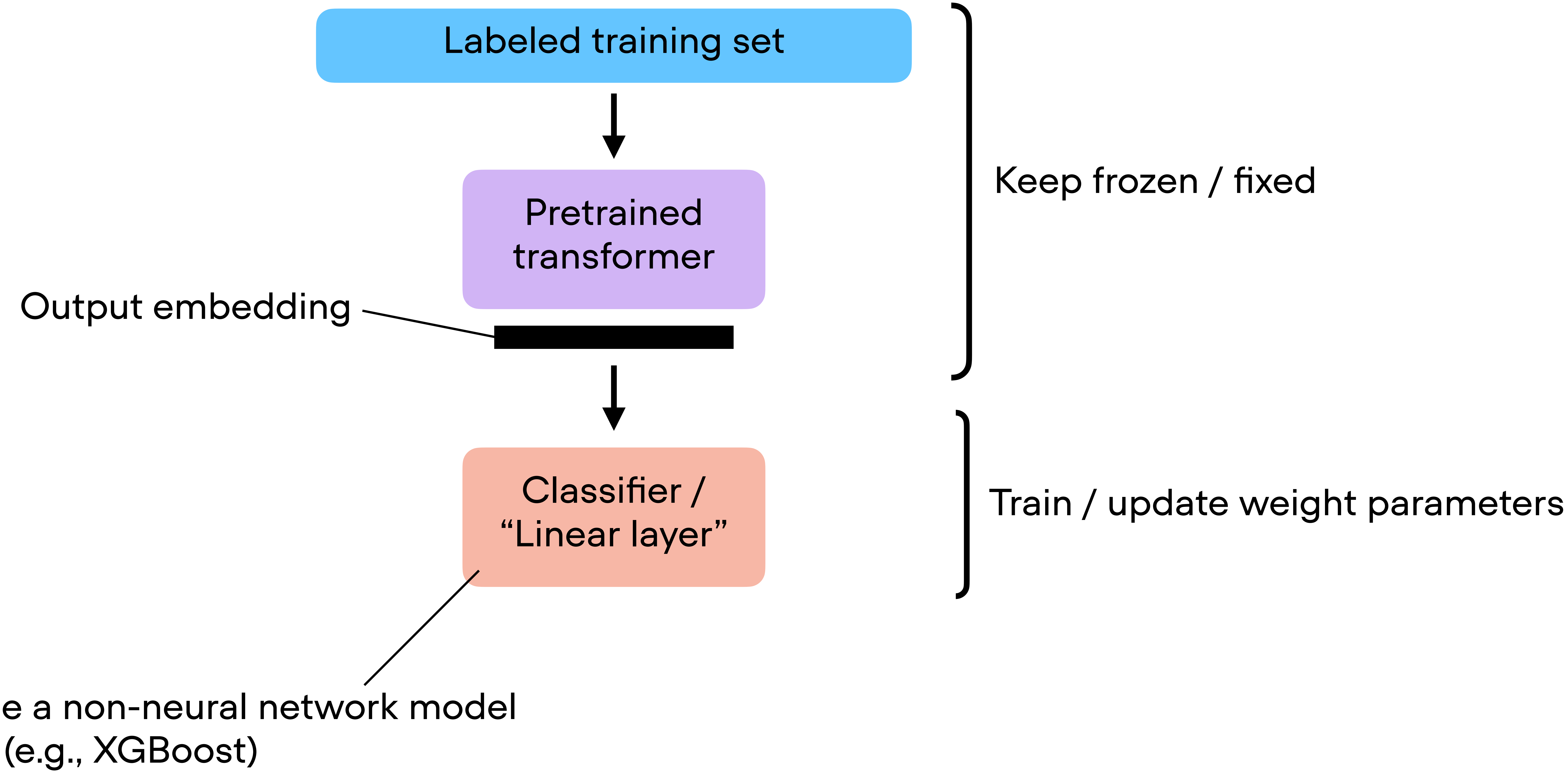
1) Feature-based approach

2) Fine-tuning approach

1) Feature-based approach

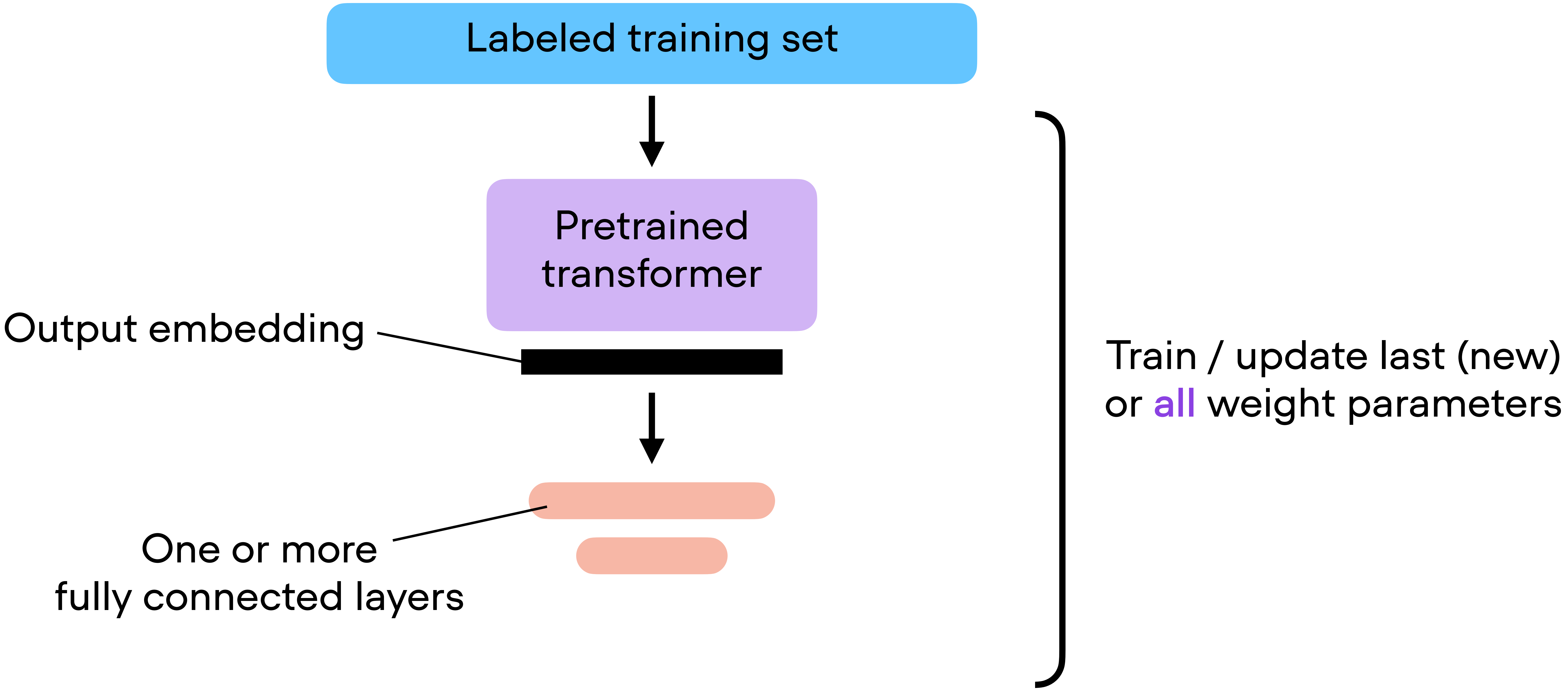


1) Feature-based approach



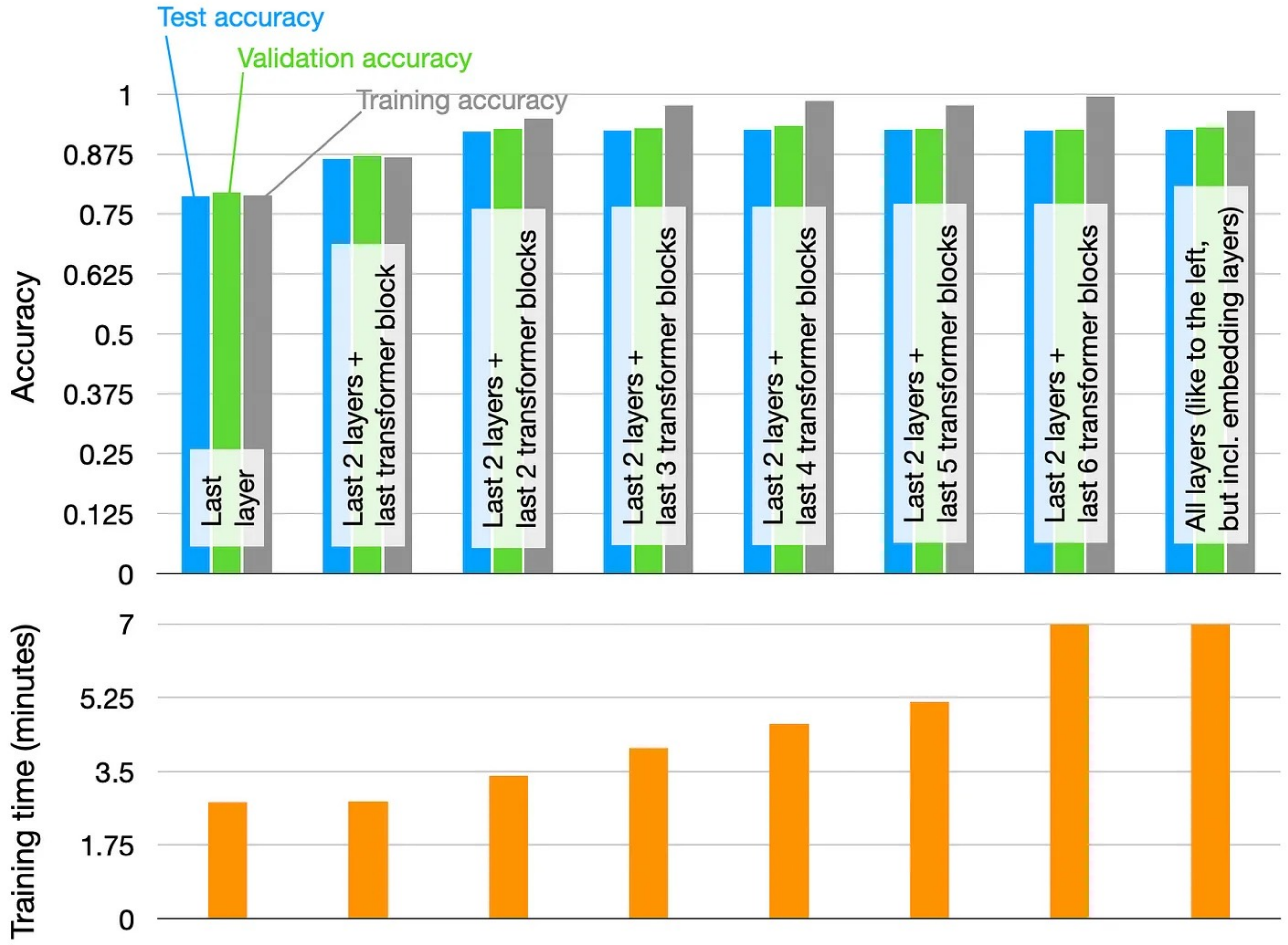
This can also be a non-neural network model (e.g., XGBoost)

1) Fine-tuning approach


Labeled training set
Pretrained transformer
Output embedding
One or more fully connected layers
Train / update last (new) or all weight parameters

Exercise

Toggle between full finetuning and finetuning individual layers



(On 1 GPU)