Understanding Large Language Models

Carsten Eickhoff, Michael Franke and Polina Tsvilodub

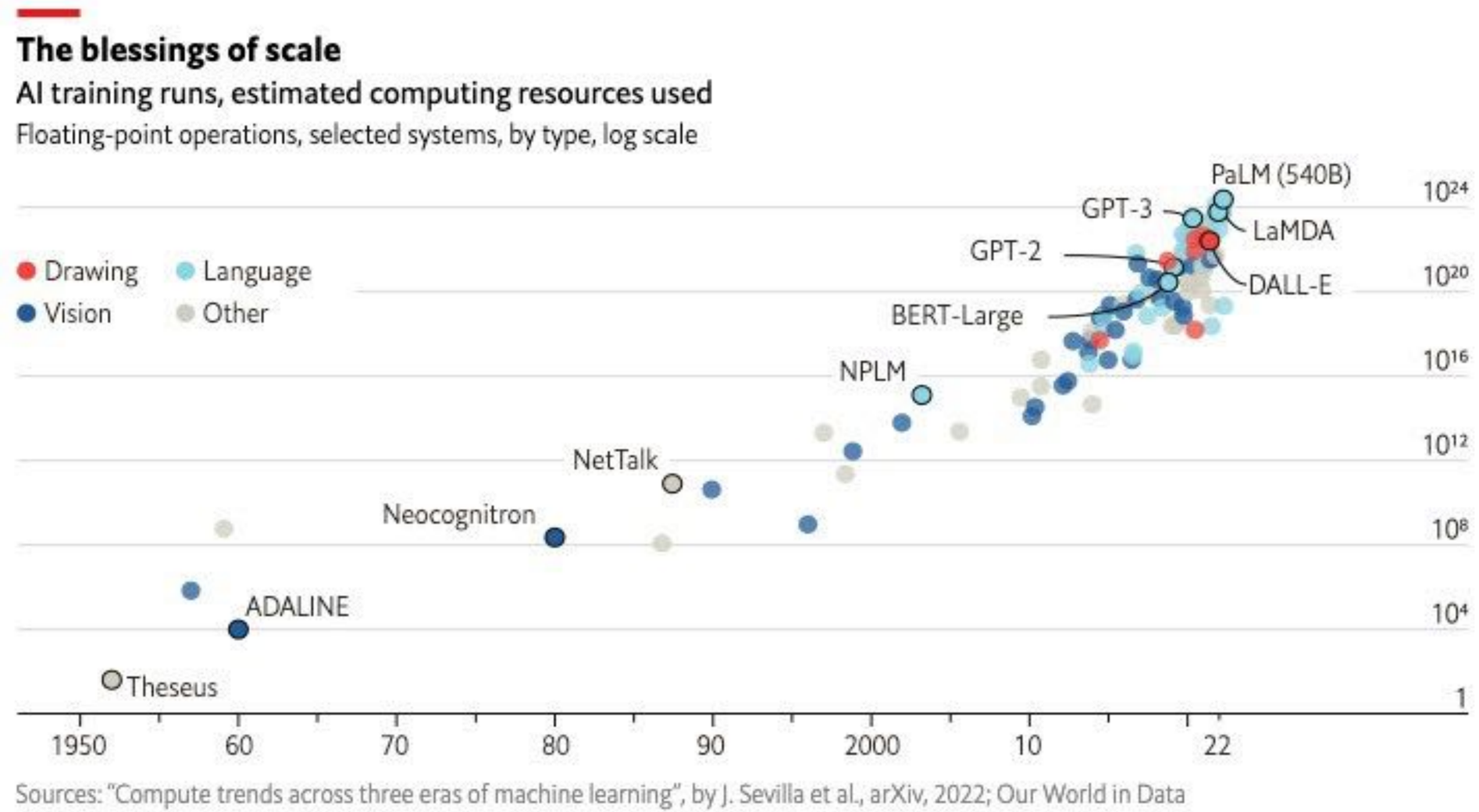
Session 05: Finetuning & RLHF

Main learning goals

1. Recap
   - model sizes, capabilities
2. Supervised Finetuning
   - basic concepts, PEFT, LoRA, QLoRA, prompt tuning
3. Instruction Finetuning
   - basic concepts
4. Recap: RL
   - reinforcement learning
5. Learning from Humans…
   - online vs. offline learning, quality signals
6. RLHF
   - language models as policies, cutting the human out of the loop

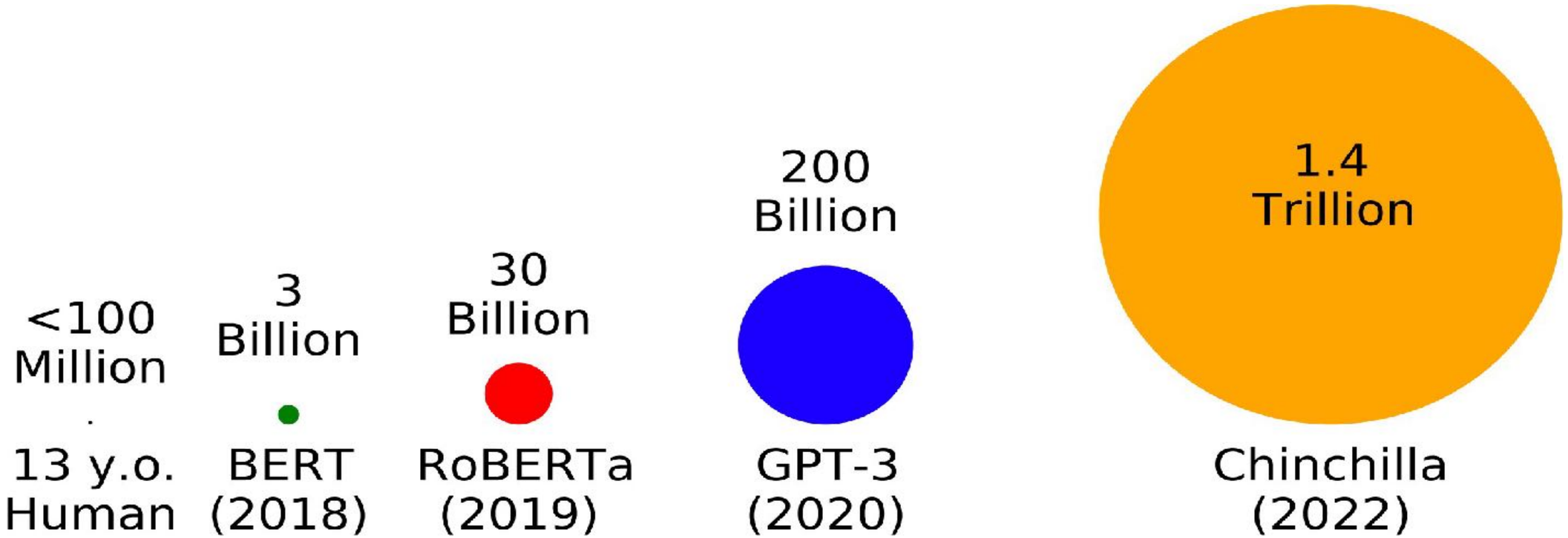
Recap

Larger and larger models



https://www.economist.com/interactive/briefing/2022/06/11/huge-foundation-models-are-turbo-charging-ai-progress

Trained on More and More Data



tokens seen during training

https://babylm.github.io/

What does Pretraining Teach us?

- *Stanford University is located in _____, California.* [Trivia]
- *I put ___ fork down on the table.* [syntax]
- *The woman walked across the street, checking for traffic over ___ shoulder.* [coreference]
- *I went to the ocean to see the fish, turtles, seals, and _____.* [lexical semantics/topic]
- *Overall, the value I got from the two hours watching it was the sum total of the popcorn and the drink. The movie was ___.* [sentiment]
- Iroh went into the kitchen to make some tea. Standing next to Iroh, Zuko pondered his destiny. Zuko left the _____. [some reasoning – this is harder]
- I was thinking about the sequence that goes 1, 1, 2, 3, 5, 8, 13, 21, ____ [some basic arithmetic; they don't learn the Fibonnaci sequence]

Rudimentary Models of Agents, Beliefs, and Actions

> ***Pat watches a demonstration of a bowling ball and a leaf being dropped at the same time in a vacuum chamber. Pat, who is a physicist, predicts that*** *the bowling ball and the leaf will fall at the same rate.*

Changing the last sentence of the prompt, we get:

> ***. . . Pat, who has never seen this demonstration before, predicts that*** *the bowling ball will fall to the ground first. This is incorrect. In a vacuum chamber, there is no air*

Language Models as Agent Models [Andreas, 2022]

Learning to Code (Pretty well!)

```
// Determine whether the sentiment of text is positive
// Use a web service
async function isPositive(text: string): Promise<boolean> {
  const response = await fetch(`http://text-processing.com/api/sentiment/`, {
    method: "POST",
    body: `text=${text}`,
    headers: {
      "Content-Type": "application/x-www-form-urlencoded",
    },
  });
  const json = await response.json();
  return json.label === "pos";
}
```

https://github.com/features/copilot

Memorizing Advanced Medical Facts

Rapid and chronic ethanol tolerance are composed of distinct memory-like states in Drosophila

Abstract

Ethanol tolerance is the first type of behavioral plasticity and neural plasticity that is induced by ethanol intake, and yet its molecular and circuit bases remain largely unexplored. Here, we characterize three distinct forms of ethanol tolerance in male Drosophila: rapid, chronic, and repeated. Rapid tolerance is composed of two short-lived memory-like states, one that is labile and one that is consolidated. Chronic tolerance, induced by continuous exposure, lasts for two days, induces ethanol preference, and hinders the development of rapid tolerance through the activity of

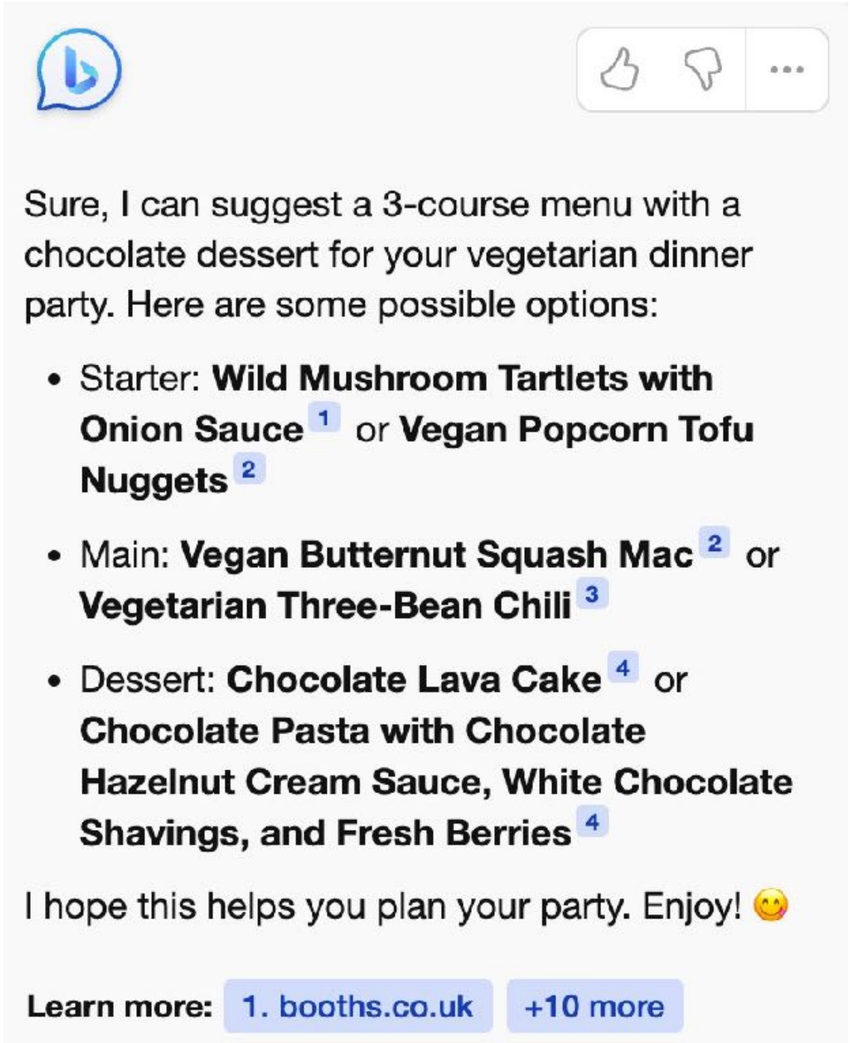
[Larnerd, 2023]

Multi-Step Problem Solving

I need to throw a dinner party for 6 people who are vegetarian. Can you suggest a 3-course menu with a chocolate dessert?

[Microsoft Bing]

(Also see OpenAI's ChatGPT, Google's Bard, Anthropic's Claude)



Sure, I can suggest a 3-course menu with a chocolate dessert for your vegetarian dinner party. Here are some possible options:

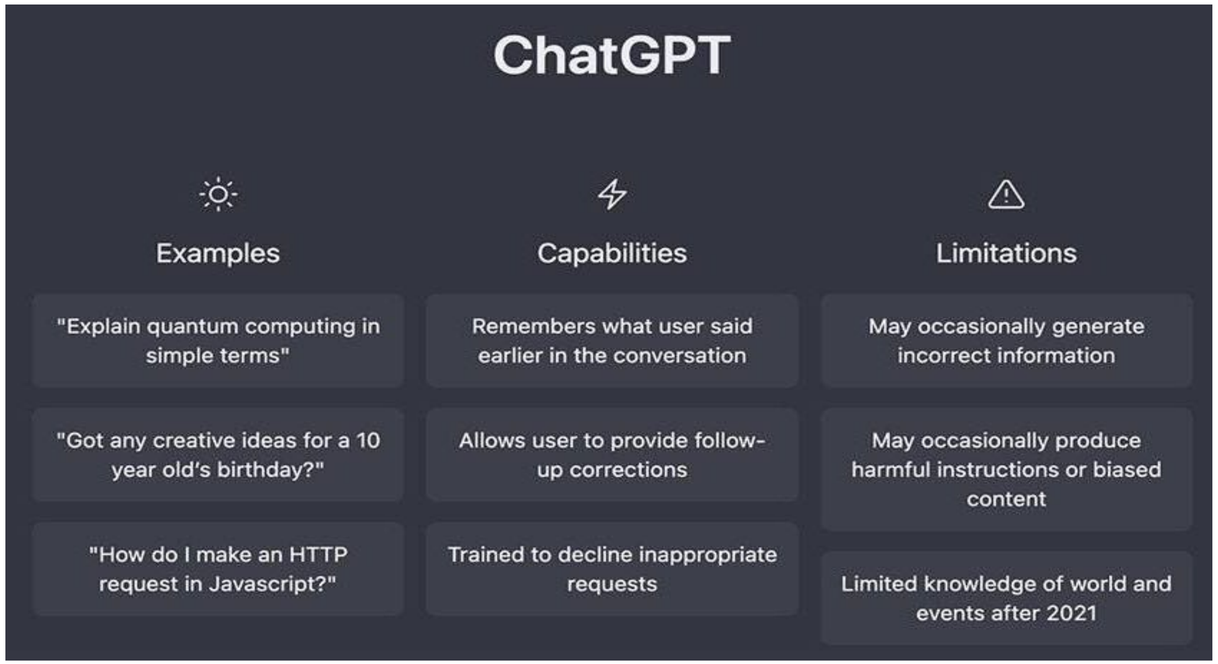
- Starter: **Wild Mushroom Tartlets with Onion Sauce** [1] or **Vegan Popcorn Tofu Nuggets** [2]
- Main: **Vegan Butternut Squash Mac** [2] or **Vegetarian Three-Bean Chili** [3]
- Dessert: **Chocolate Lava Cake** [4] or **Chocolate Pasta with Chocolate Hazelnut Cream Sauce, White Chocolate Shavings, and Fresh Berries** [4]

I hope this helps you plan your party. Enjoy!

Learn more: 1. booths.co.uk +10 more

Multitask Assistants

- How do we get from *this*
  - *Stanford University is located in* _____
- to *this*?



Language Modeling ≠ Assisting Users

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION GPT-3

```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

Language models are not *aligned* with user intent [Ouyang et al., 2022].

Language Modeling ≠ Assisting Users

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION **Human**

A giant rocket ship blasted off from Earth carrying astronauts to the moon. The astronauts landed their spaceship on the moon and walked around exploring the lunar surface. Then they returned safely back to Earth, bringing home moon rocks to show everyone.

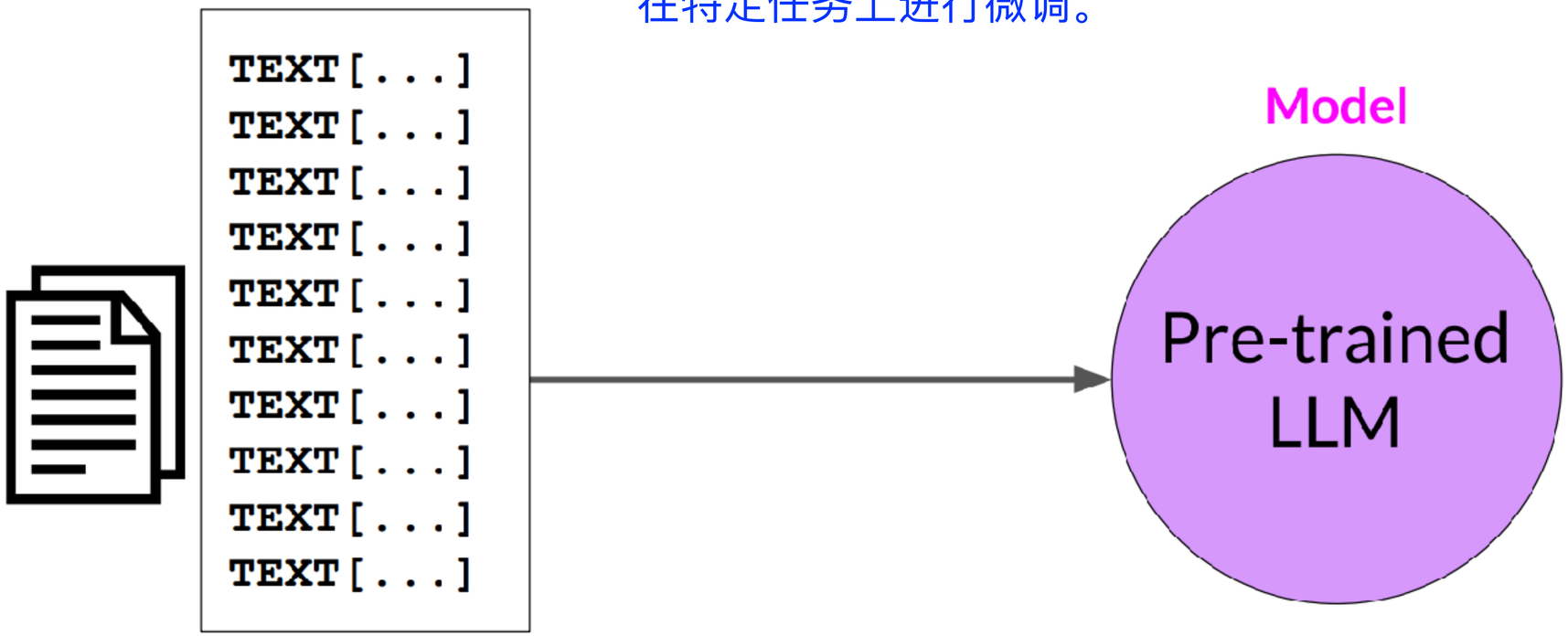
Language models are not *aligned* with user intent [Ouyang et al., 2022].
Finetuning to the rescue!

Supervised Finetuning

Pretraining

@GPT:

Pretrained model是指在大量数据上预先训练过的模型，然后在特定任务上进行微调。

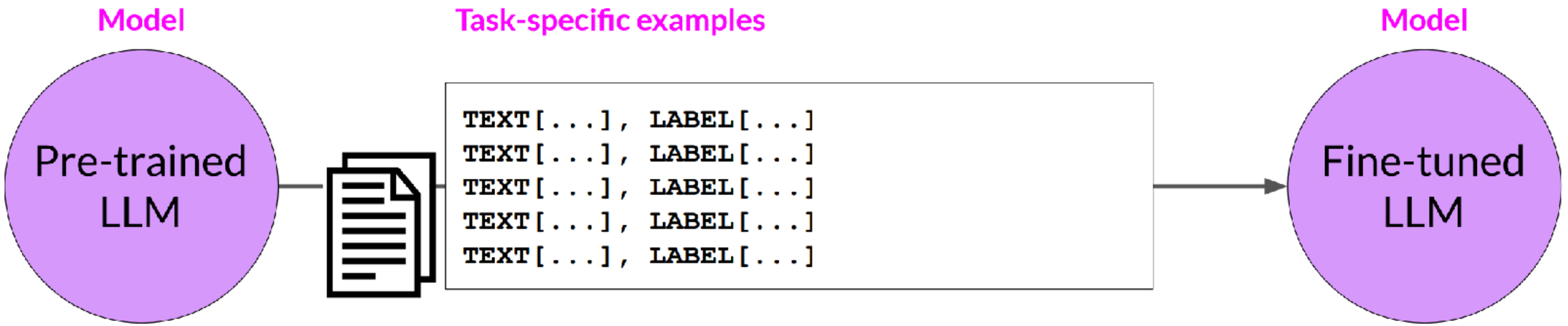


GB - TB - PB
of unstructured textual data

Conventional Finetuning

FT: a process where a pre-trained model is further trained on specific and lebelled dateset to perform a specific taks

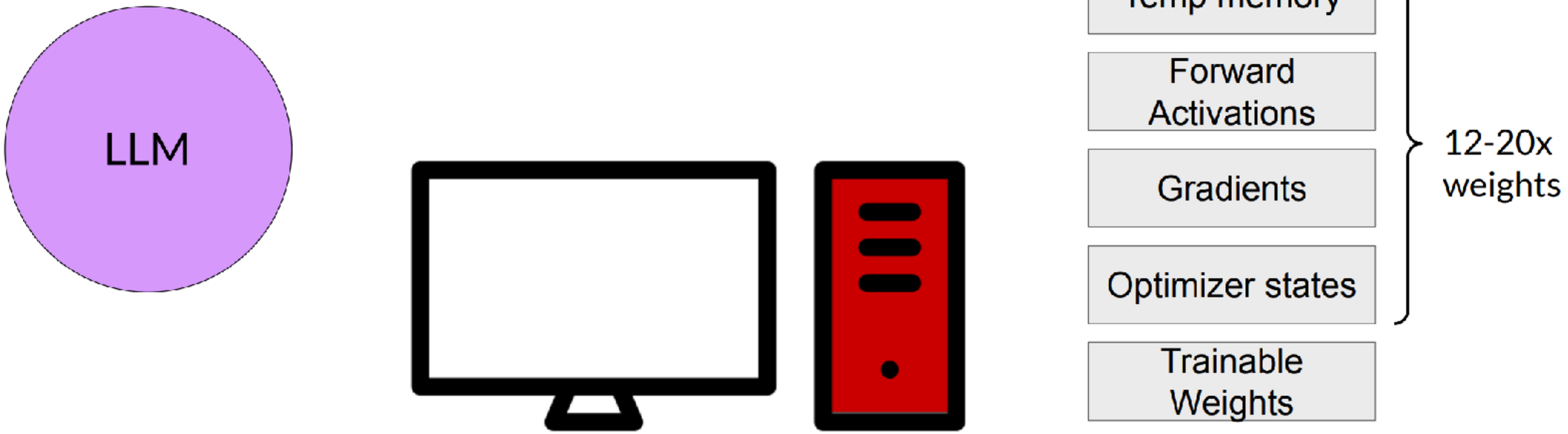
LLM fine-tuning



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

Finetuning at the Frontier

Full fine-tuning of large LLMs is challenging



Getting Smart with Finetuning

PEFT methods summary

aim to reduce the FT's computational cost and memory usage

Selective

Select subset of initial LLM parameters to fine-tune

Reparameterization

Reparameterize model weights using a low-rank representation

LoRA

Additive

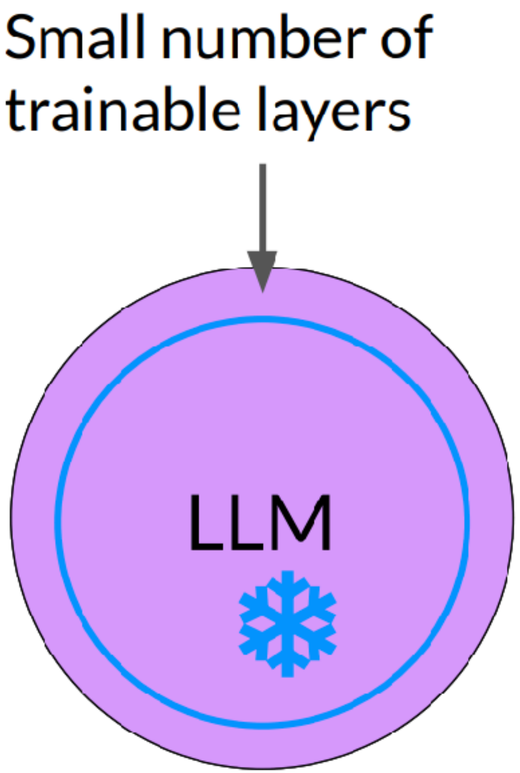
Add trainable layers or parameters to model

Adapters

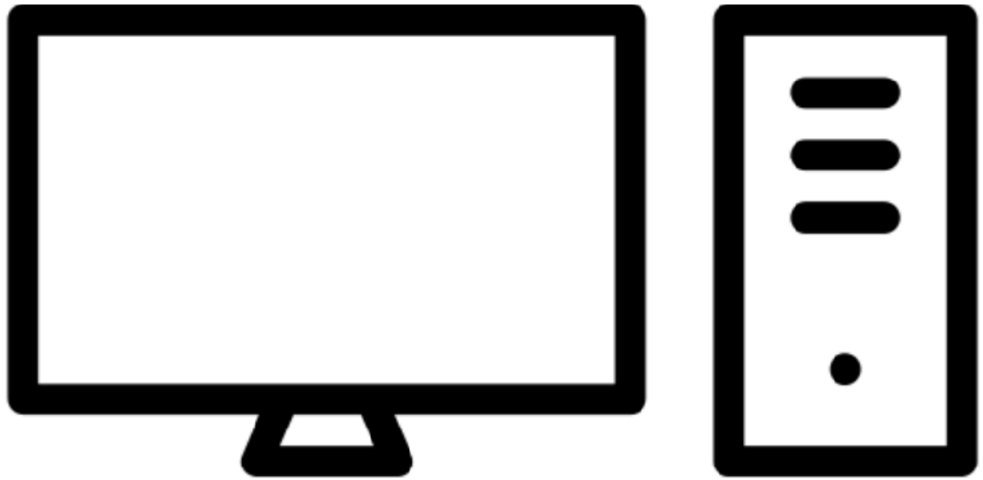
Soft Prompts

Prompt Tuning

Selective Finetuning

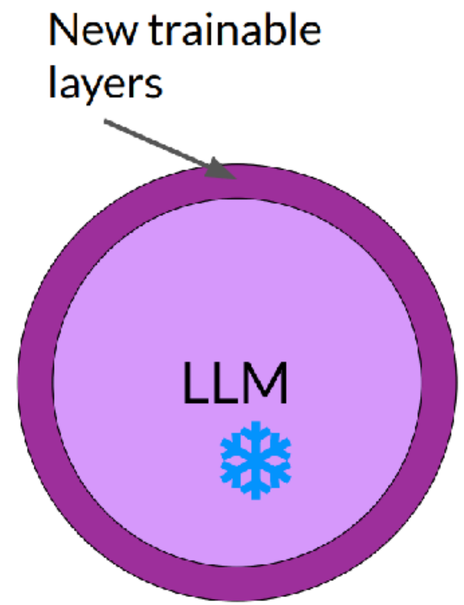


LLM with most layers frozen

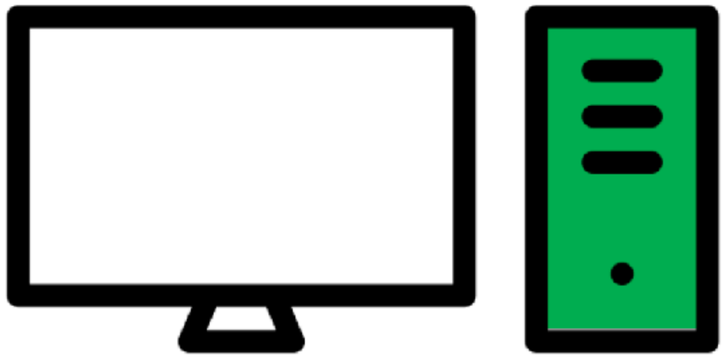


Selective Finetuning

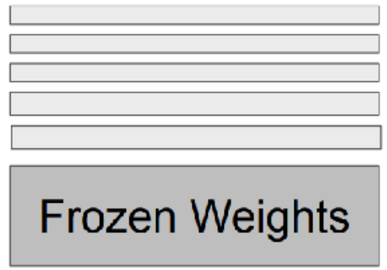
Parameter efficient fine-tuning (PEFT)



LLM with additional layers for PEFT



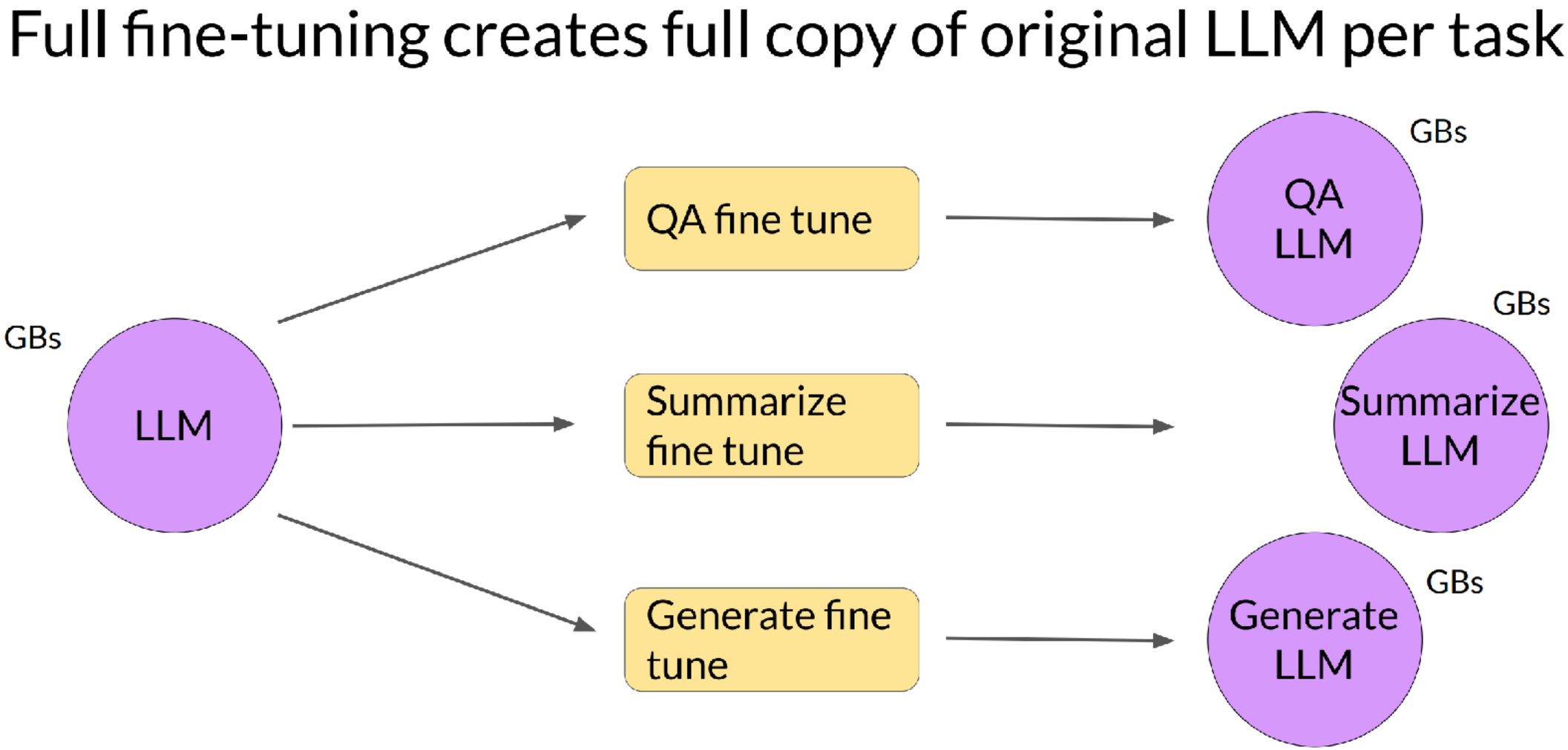
Less prone to catastrophic forgetting



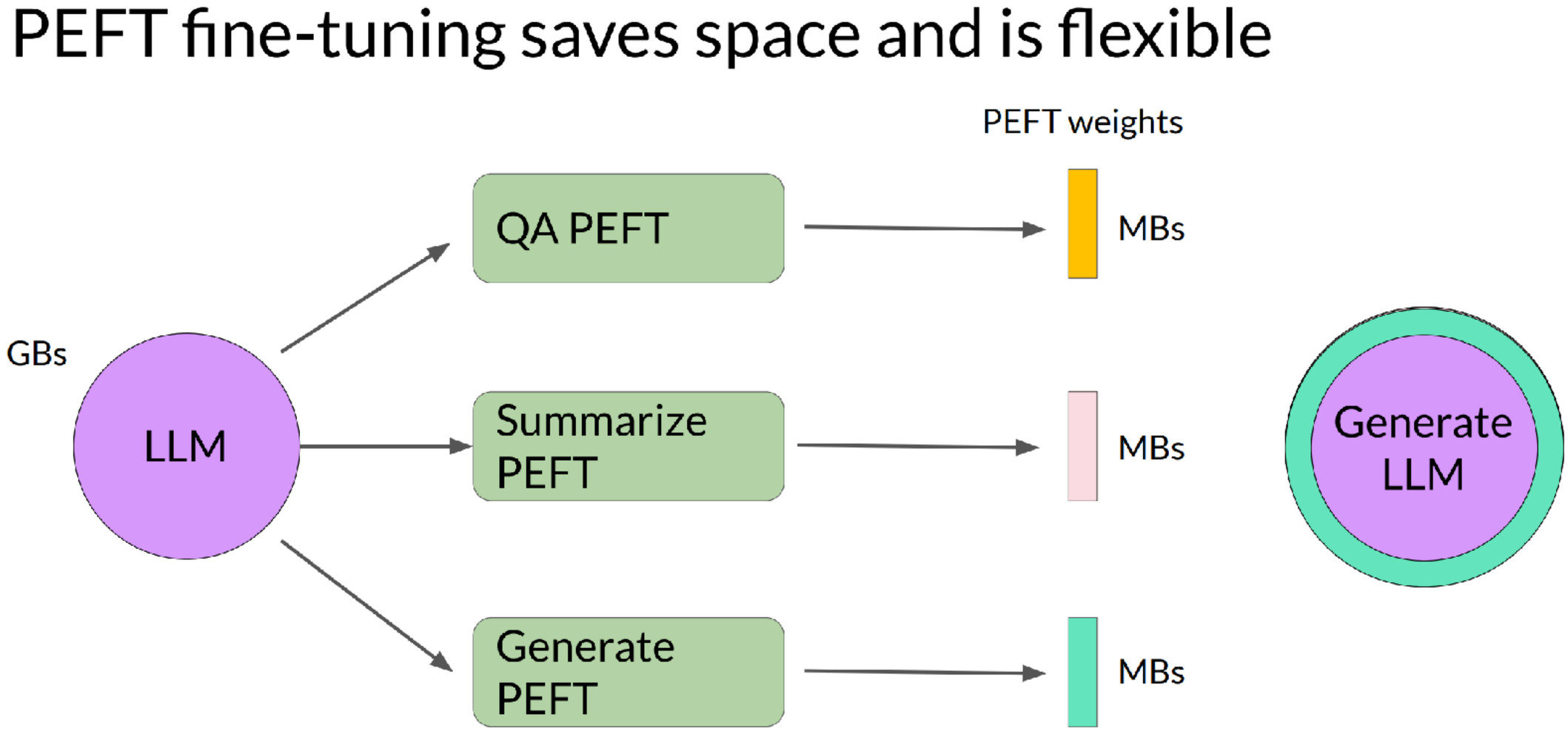
Other components

Trainable weights

Selective Finetuning



Selective Finetuning



Getting Smart with Finetuning

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

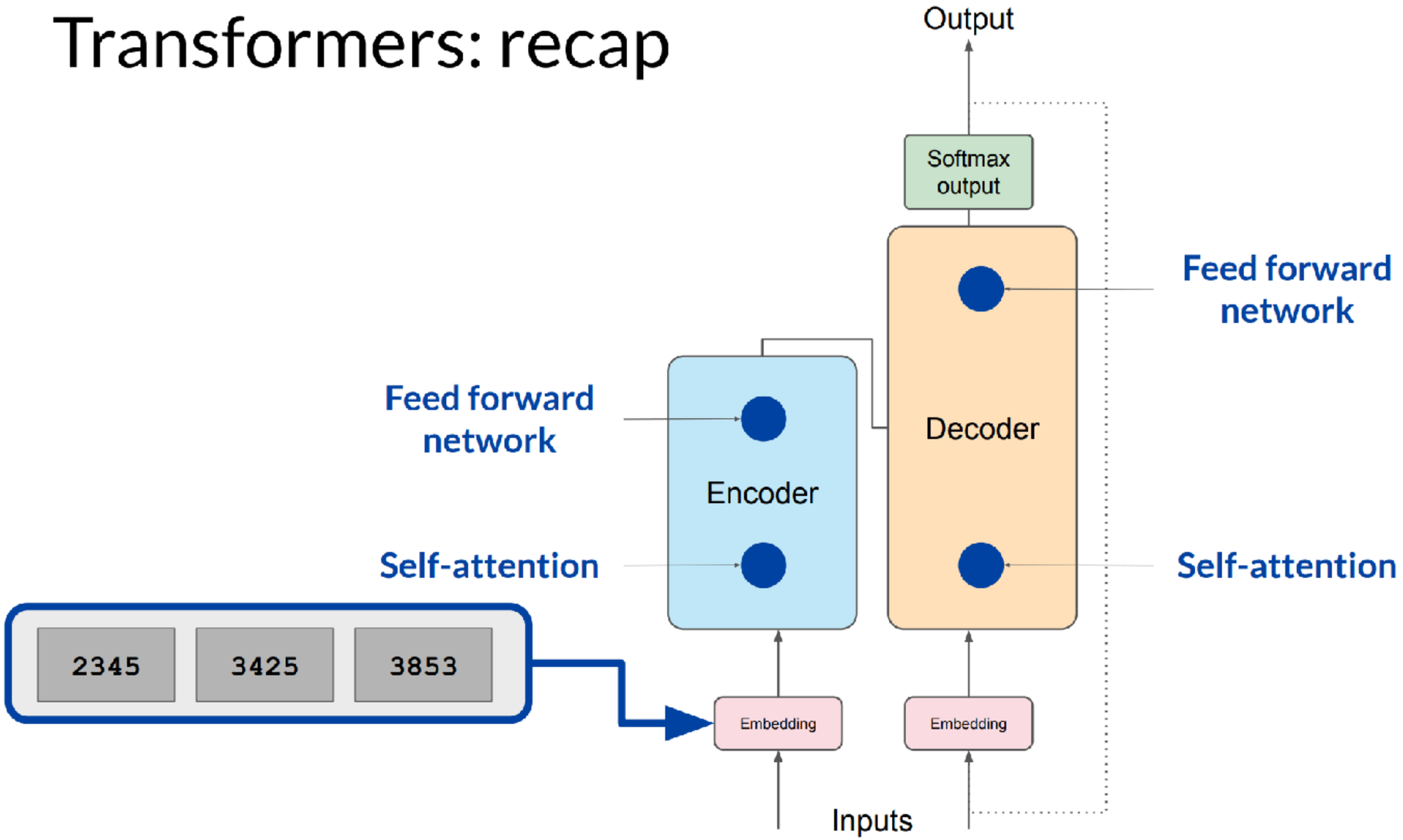
Adapters

Soft Prompts

Prompt Tuning

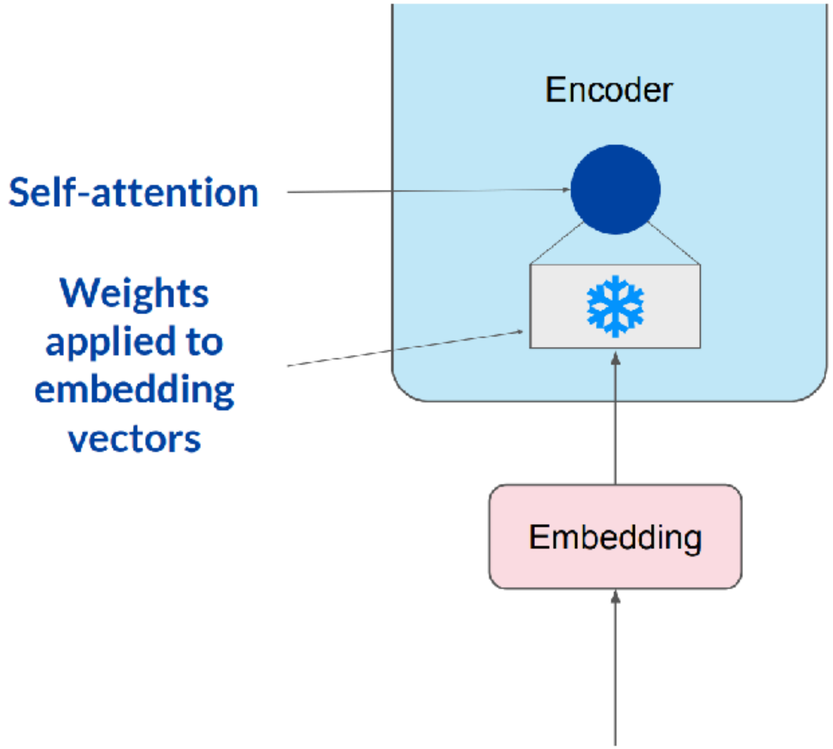
LoRA

Transformers: recap



LoRA

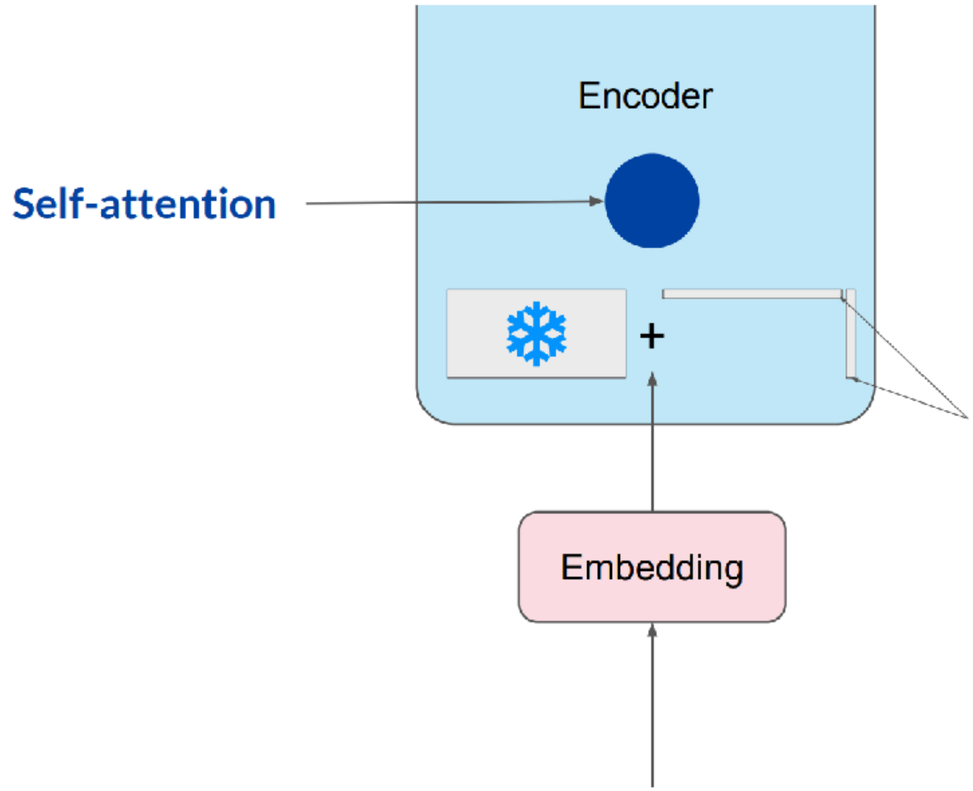
LoRA: Low Rank Adaption of LLMs



1. Freeze most of the original LLM weights.

LoRA

LoRA: Low Rank Adaption of LLMs

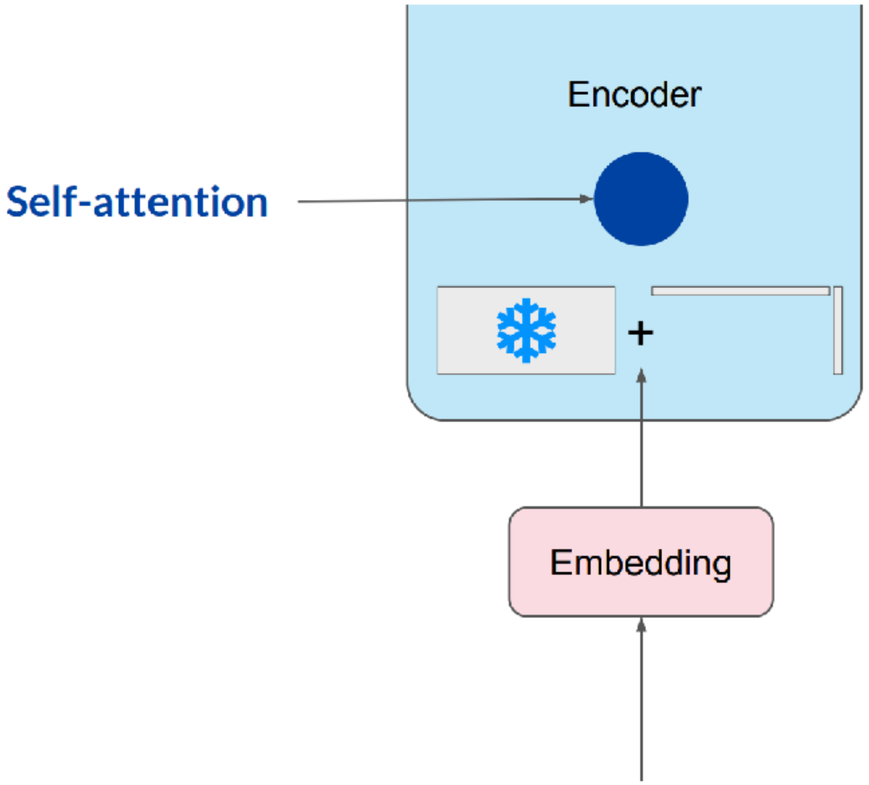


1. Freeze most of the original LLM weights.
2. Inject 2 **rank decomposition matrices**
3. Train the weights of the smaller matrices

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

LoRA

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

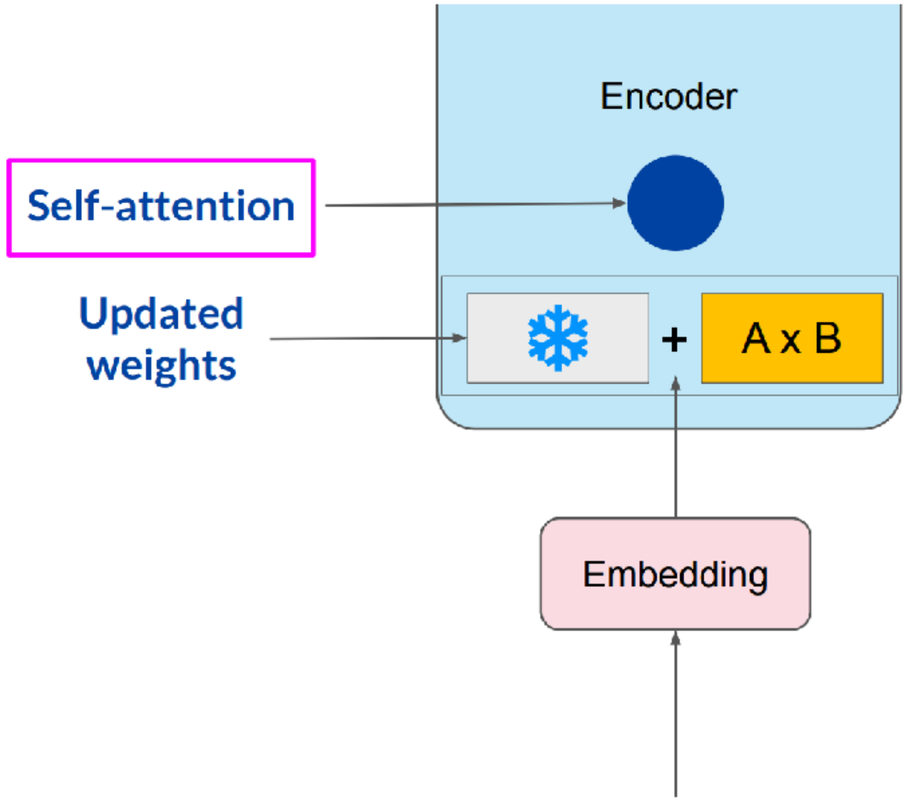
Steps to update model for inference

1. Matrix multiply the low rank matrices

2. Add to original weights

LoRA

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

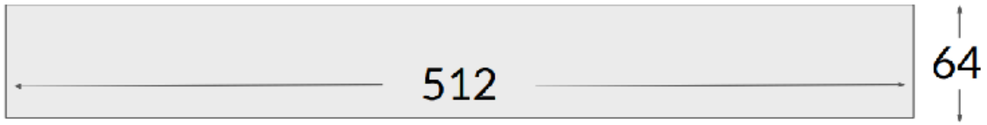
2. Add to original weights

LoRA

Concrete example using base Transformer as reference

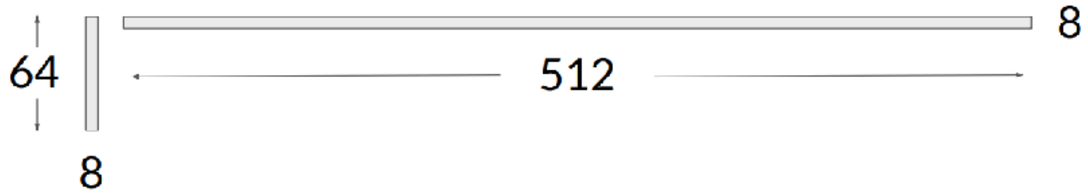
Use the base Transformer model presented by Vaswani et al. 2017:

- Transformer weights have dimensions d x k = 512 x 64
- So 512 x 64 = 32,768 trainable parameters



In LoRA with rank r = 8:

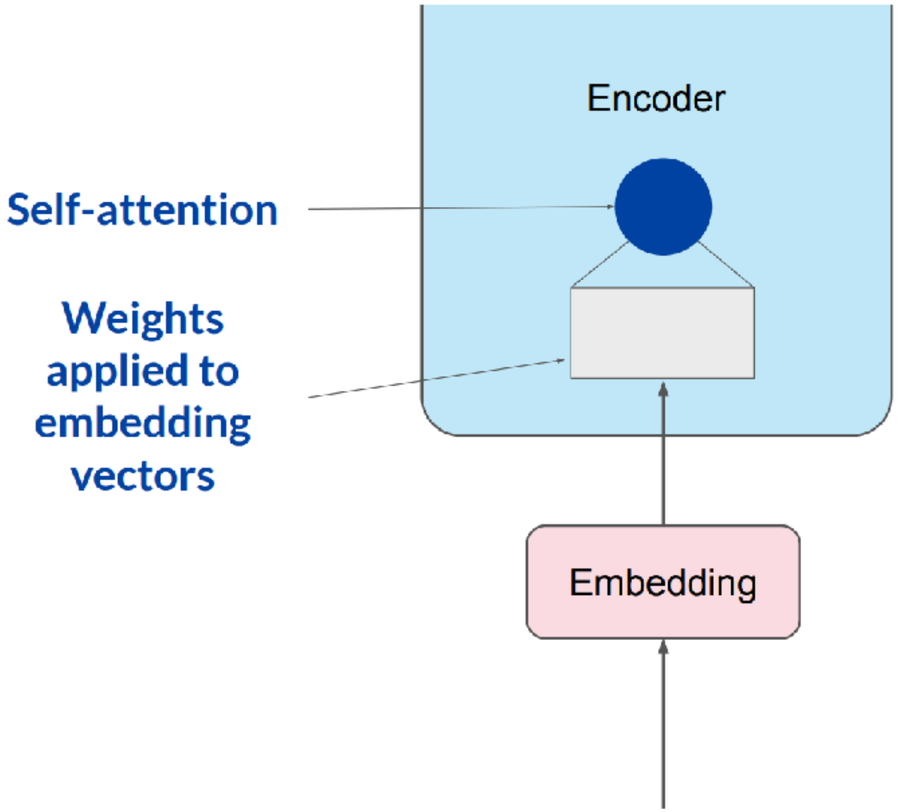
- A has dimensions r x k = 8 x 64 = 512 parameters
- B has dimension d x r = 512 x 8 = 4,096 trainable parameters



86% reduction in parameters to train!

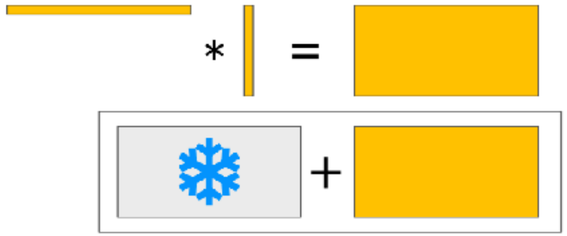
LoRA

LoRA: Low Rank Adaption of LLMs



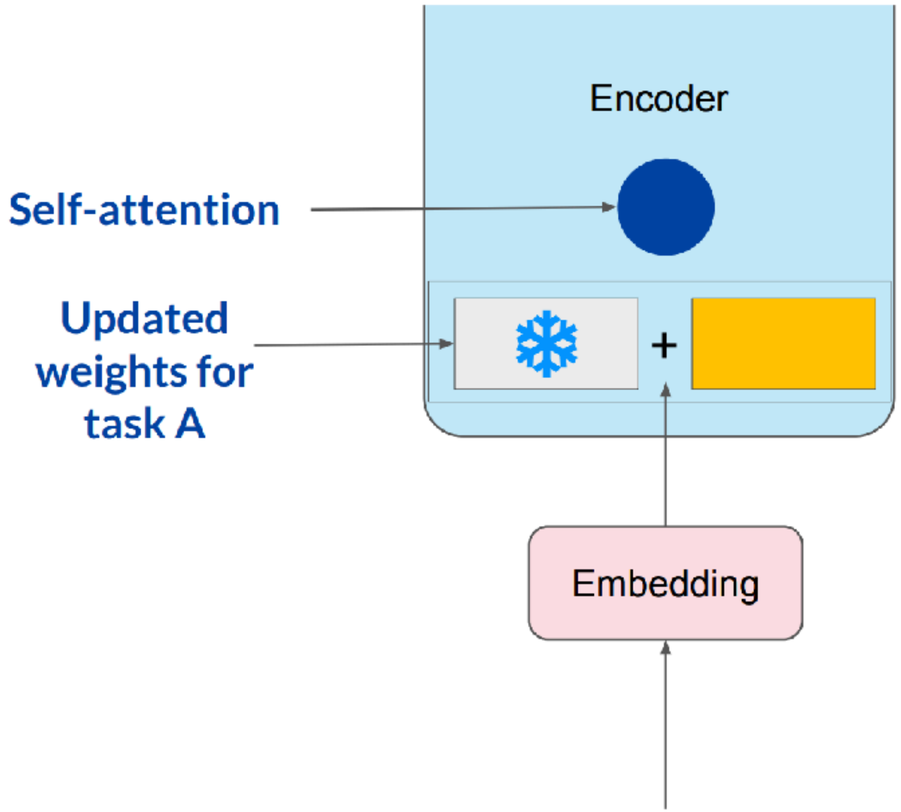
1. Train different rank decomposition matrices for different tasks
2. Update weights before inference

Task A

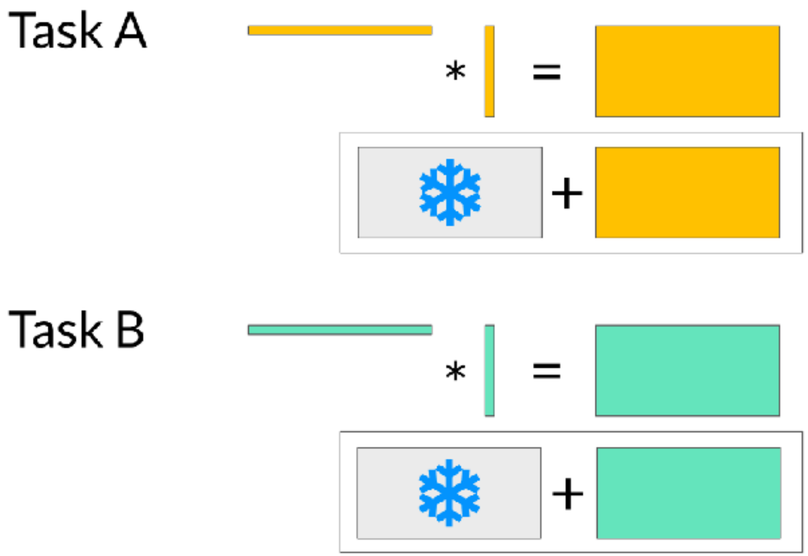


LoRA

LoRA: Low Rank Adaption of LLMs

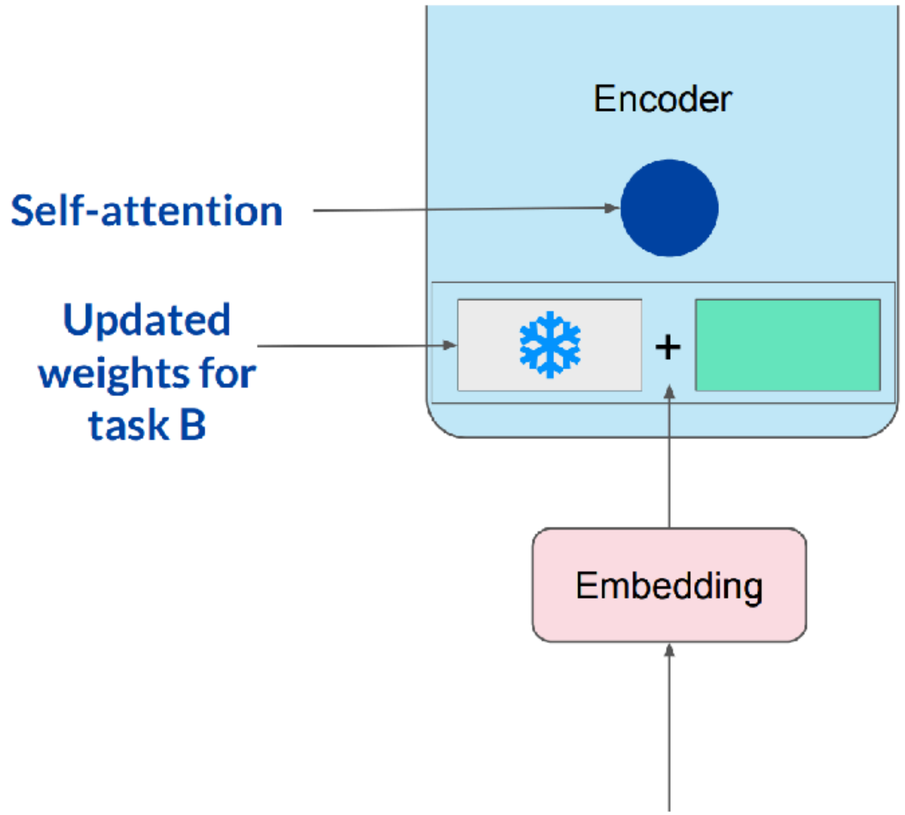


1. Train different rank decomposition matrices for different tasks
2. Update weights before inference

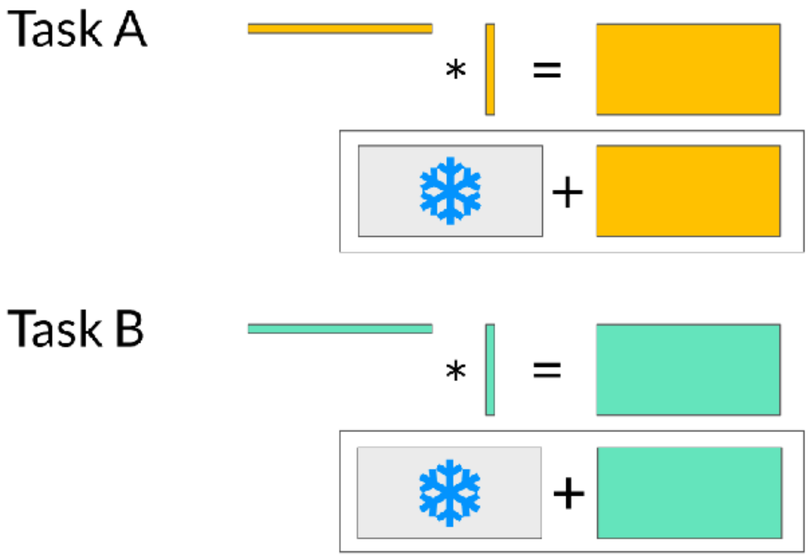


LoRA

LoRA: Low Rank Adaption of LLMs



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



QLoRA

QLoRA: Quantized LoRA

- Introduces 4-bit NormalFloat (nf4) data type for 4-bit quantization
- Supports double-quantization to reduce memory ~0.4 bits per parameter (~3 GB for a 65B model)
- Unified GPU-CPU memory management reduces GPU memory usage
- LoRA adapters at every layer - not just attention layers
- Minimizes accuracy trade-off

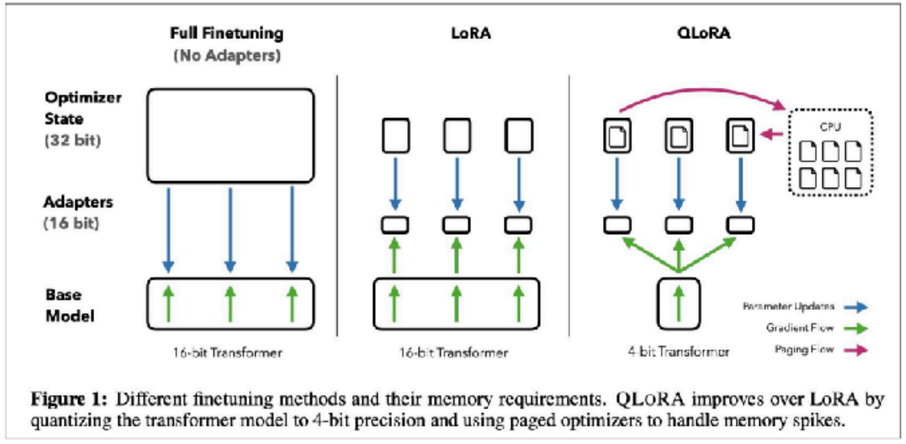


Figure 1: Different finetuning methods and their memory requirements. QLoRA improves over LoRA by quantizing the transformer model to 4-bit precision and using paged optimizers to handle memory spikes.

Source: Dettmers et al. 2023, "QLoRA: Efficient Finetuning of Quantized LLMs"

Getting Smart with Finetuning

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

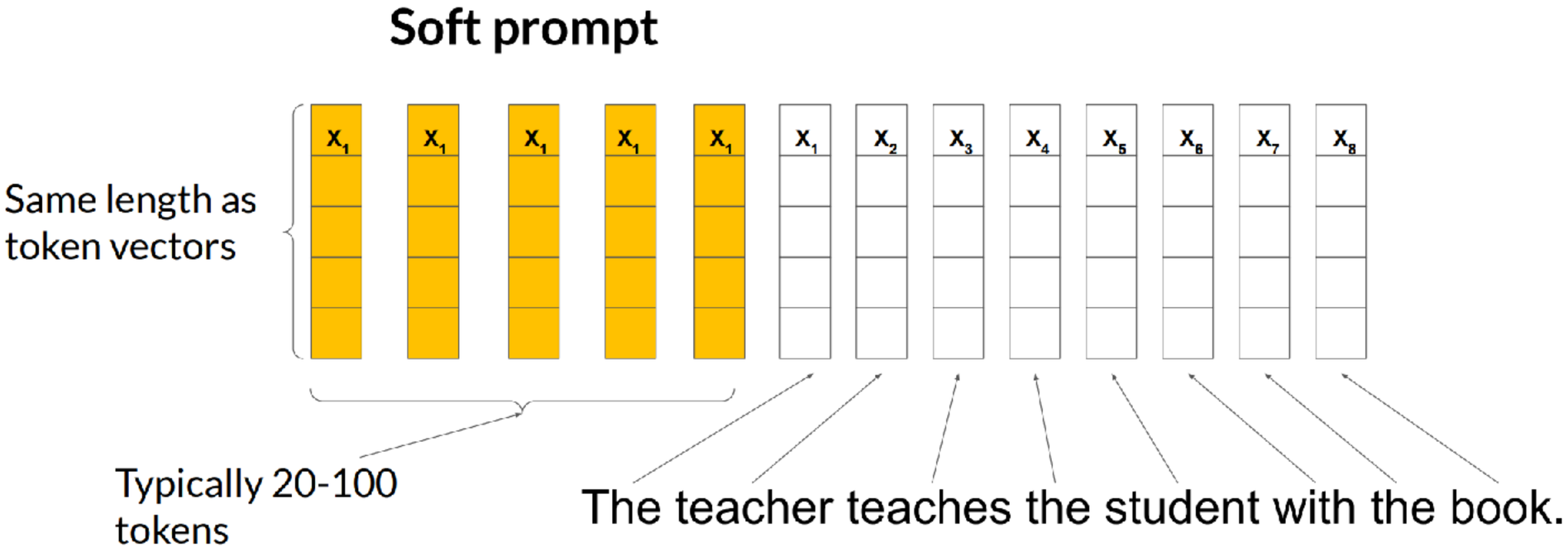
Adapters

Soft Prompts

Prompt Tuning

Prompt Tuning

Prompt tuning adds trainable "soft prompt" to inputs

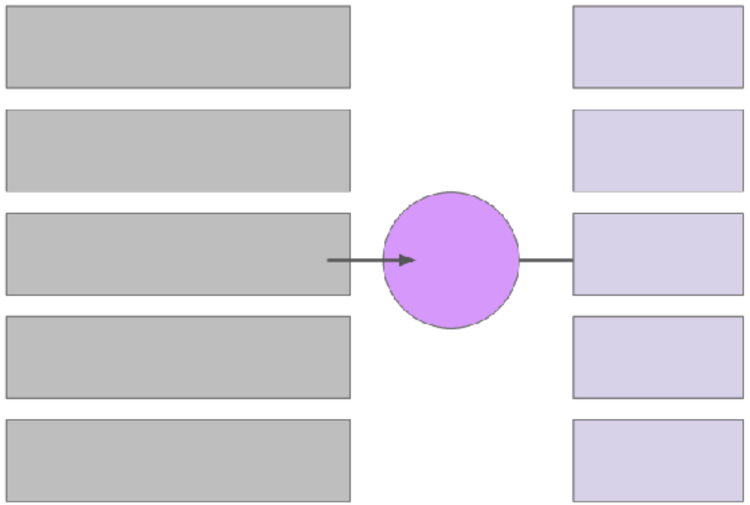


Prompt Tuning

Full Fine-tuning vs prompt tuning

adjust all/most params

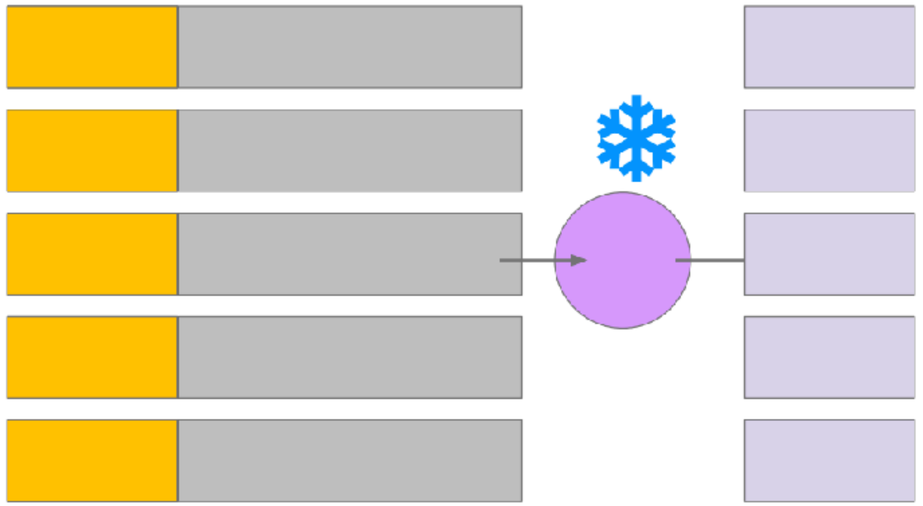
Weights of model updated during training



Millions to Billions of parameter updated

only a small set of params

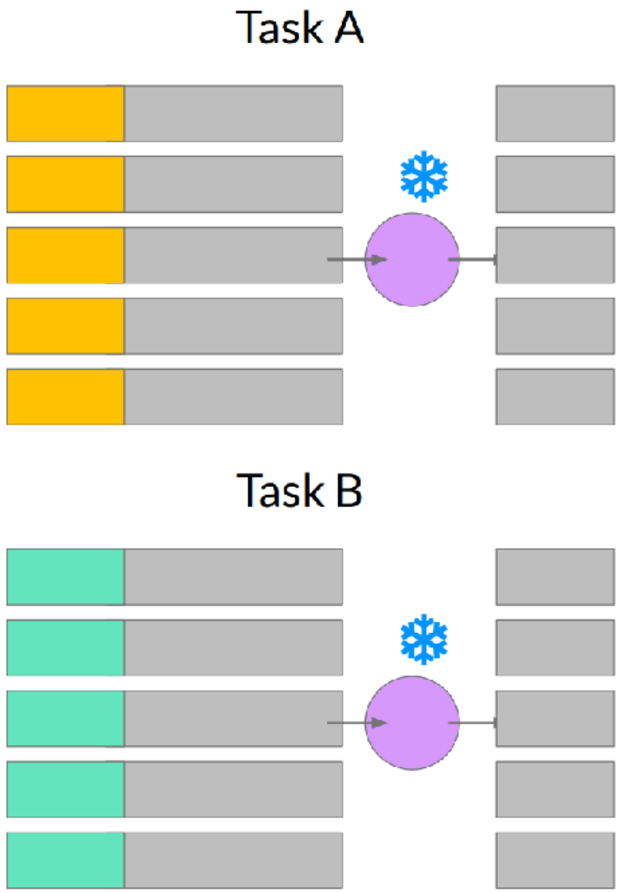
Weights of model frozen and soft prompt trained



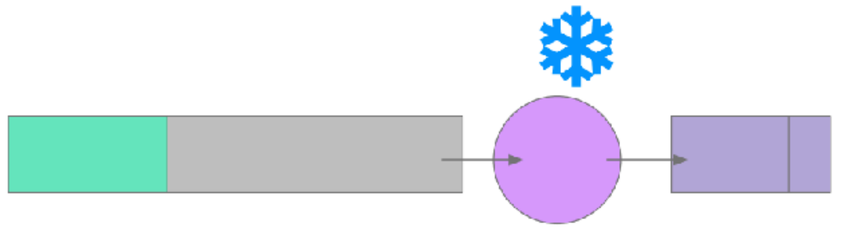
10K - 100K of parameters updated

Prompt Tuning

Prompt tuning for multiple tasks

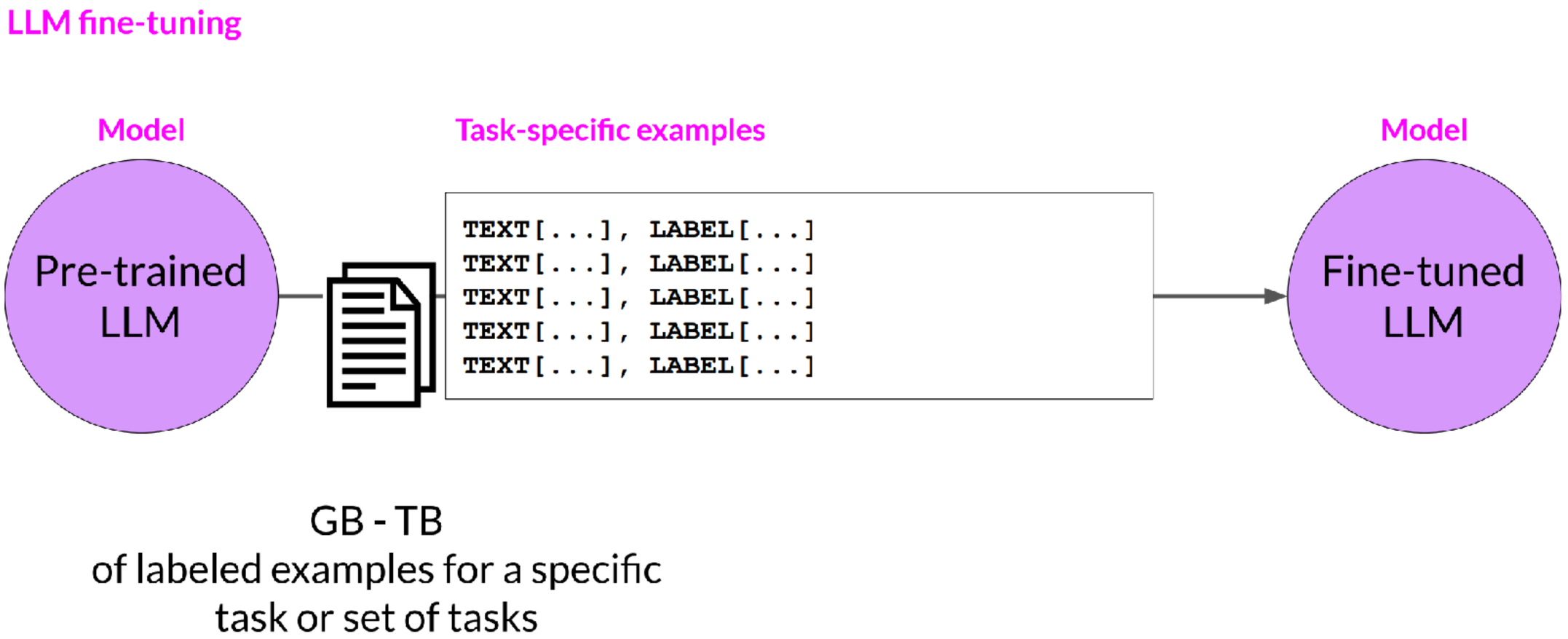


Switch out soft prompt at inference time to change task!

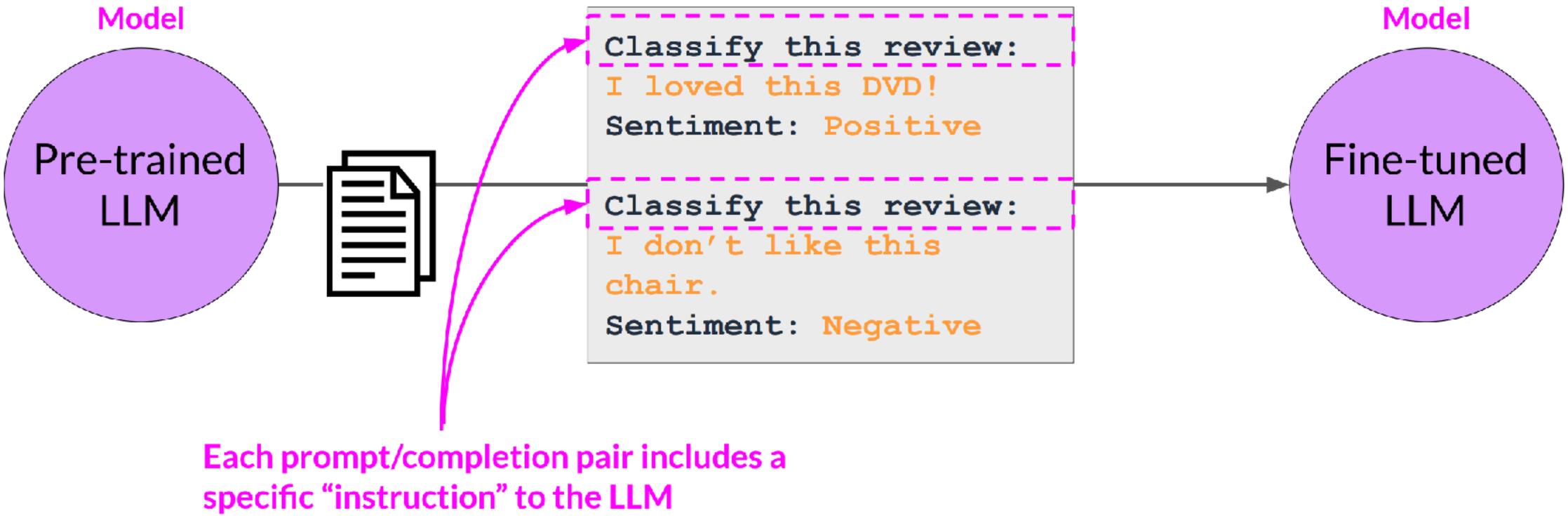


Instruction Finetuning

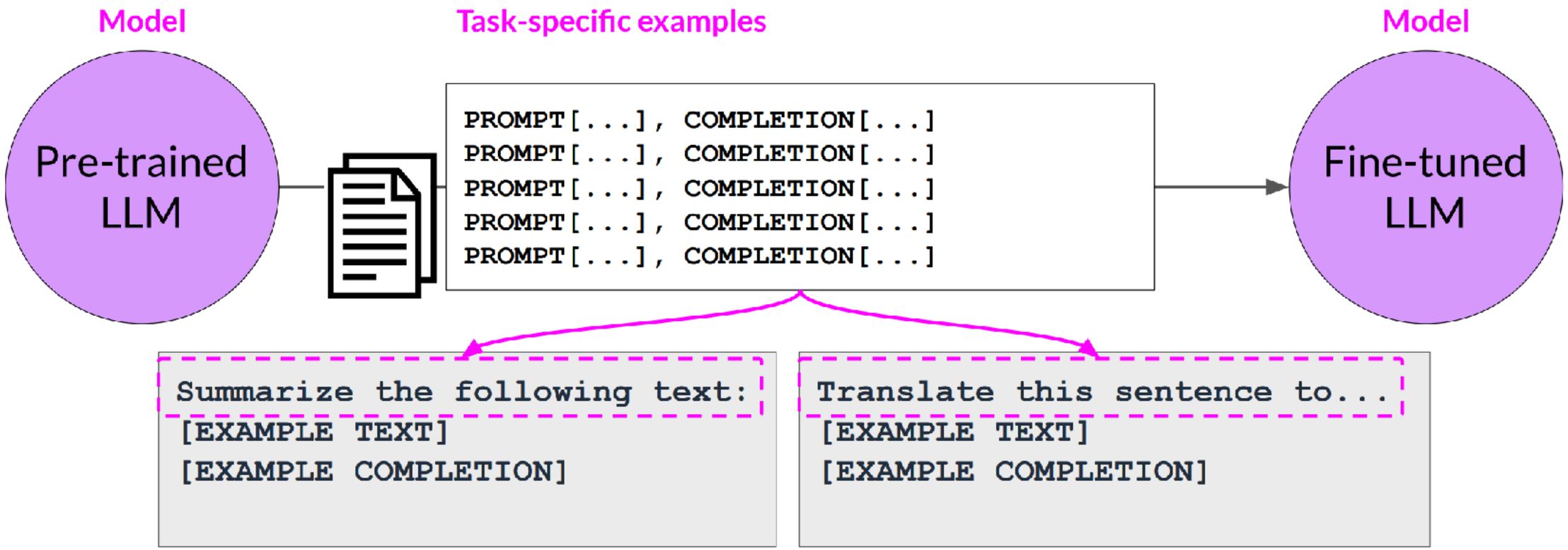
Conventional Finetuning



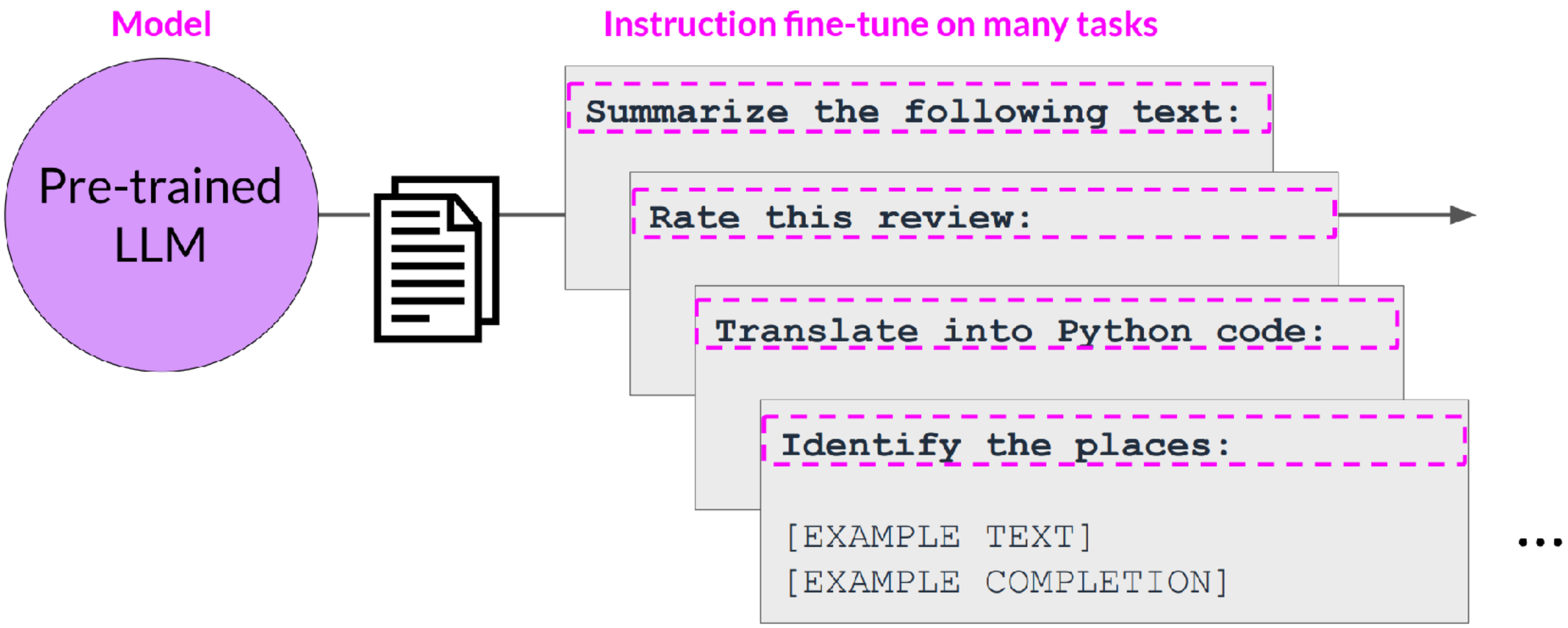
Instruction Finetuning



Instruction Finetuning



Instruction Finetuning



Instruction Finetuning

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"

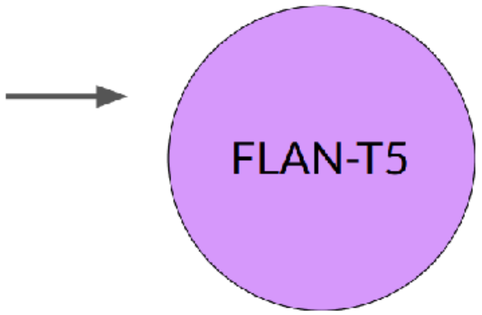
Instruction Finetuning

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.
```

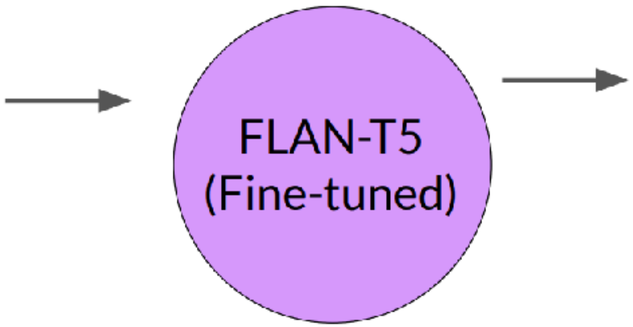
Instruction Finetuning

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.

Broad Instruction Datasets: GLUE

GLUE

GLUE

The tasks included in SuperGLUE benchmark:

Corpus	|Train|	|Test|	Task	Metrics	Domain
			Single-Sentence Tasks		
CoLA	8.5k	**1k**	acceptability	Matthews corr.	misc.
SST-2	67k	1.8k	sentiment	acc.	movie reviews
			Similarity and Paraphrase Tasks		
MRPC	3.7k	1.7k	paraphrase	acc./F1	news
STS-B	7k	1.4k	sentence similarity	Pearson/Spearman corr.	misc.
QQP	364k	**391k**	paraphrase	acc./F1	social QA questions
			Inference Tasks		
MNLI	393k	**20k**	NLI	matched acc./mismatched acc.	misc.
QNLI	105k	5.4k	QA/NLI	acc.	Wikipedia
RTE	2.5k	3k	NLI	acc.	news, Wikipedia
WNLI	634	**146**	coreference/NLI	acc.	fiction books

Source: Wang et al. 2018, “GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding”

Broad Instruction Datasets: SuperGLUE

SuperGLUE

The tasks included in SuperGLUE benchmark:

Corpus	|Train|	|Dev|	|Test|	Task	Metrics	Text Sources
BoolQ	9427	3270	3245	QA	acc.	Google queries, Wikipedia
CB	250	57	250	NLI	acc./F1	various
COPA	400	100	500	QA	acc.	blogs, photography encyclopedia
MultiRC	5100	953	1800	QA	$F1_a$/EM	various
ReCoRD	101k	10k	10k	QA	F1/EM	news (CNN, Daily Mail)
RTE	2500	278	300	NLI	acc.	news, Wikipedia
WiC	6000	638	1400	WSD	acc.	WordNet, VerbNet, Wiktionary
WSC	554	104	146	coref.	acc.	fiction books

Source: Wang et al. 2019, "SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems"

Broad Instruction Datasets: HELM

Holistic Evaluation of Language Models (HELM)

HELM

Metrics:

1. Accuracy
2. Calibration
3. Robustness
4. Fairness
5. Bias
6. Toxicity
7. Efficiency

Scenarios

Models

	J1-Jumbo	J1-Grande	J1-Large	Anthropic-LM	BLOOM	T0pp	Cohere XL	Cohere Large	Cohere Medium	Cohere Small	GPT-NeoX
NaturalQuestions (open)			✔	✔	✔	✔	✔	✔	✔	✔	✔
NaturalQuestions (closed)	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
BoolQ	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
NarrativeQA	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
QuAC	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
HellaSwag	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
OpenBookQA	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
TruthfulQA	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
MMLU	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
MS MARCO				✔	✔		✔	✔	✔	✔	✔
TREC				✔	✔		✔	✔	✔	✔	✔
XSUM	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
CNN/DM	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
IMDB	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
CivilComments	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
RAFT	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔

Recap: RL

Recap

Reinforcement learning

- the central framework for formalizing RL problems are Markov Decision Processes (MDPs)
- task of RL is to solve MDP such that the expected return is maximized
  - and to find the optimal policy
- classical solution methods for MDPs include estimation of optimal value functions
- policy gradient methods directly optimize the policy such that the expected return is maximized
  - can be applied to LMs!

Markov Decision Processes

Formal definition

Finite MDPs: (S, A, T, R)

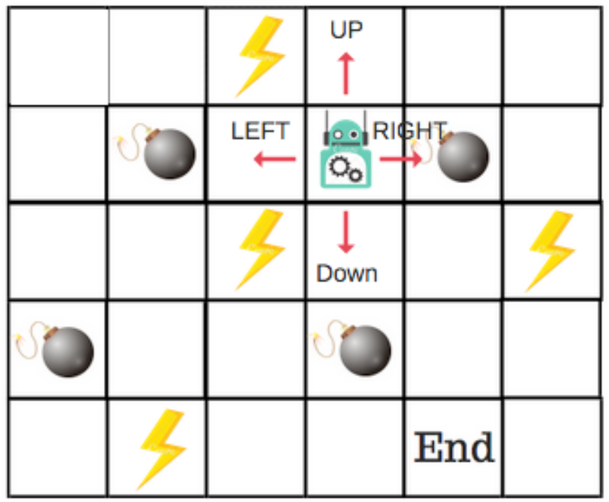
1. $S_t \in S$ for $t = 0, 1, 2, 3, \ldots$
2. $A_t \in A(s)$
3. $R_{t+1} \in R$
4. $T(s' \mid s, a) = \sum_{r' \in R} P(s', r' \mid s, a)$

Goal: maximize **returns** until goal achieved

$G_t = R_{t+1} + R_{t_2} + \ldots + R_T$

Formally: maximize expected **discounted** rewards over **episode**

$$G_t = R_{t+1} + \gamma R_{t_2} + \gamma^2 R_{t+3} + \ldots + \gamma^{T-t-1} R_T = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$$



S State
A Action
T Transition
R Rewards

Sutton & Barto (2018, p. 48, Fig 3.1)

Markov Decision Processes

Formal definition

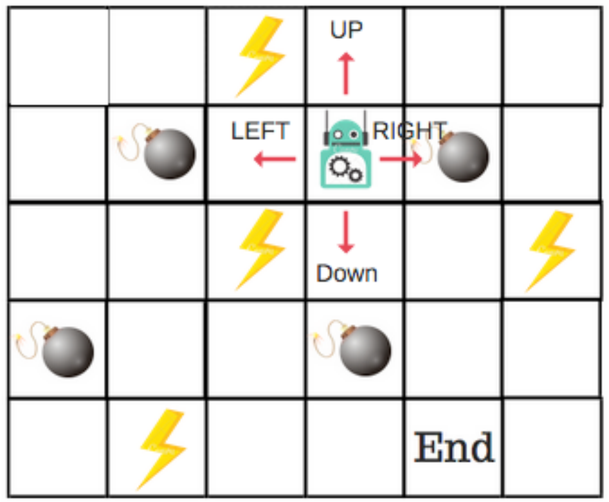
Finite MDPs: (S, A, T, R)

1. $S_t \in S$ for $t = 0, 1, 2, 3, \ldots$
2. $A_t \in A(s)$
3. $R_{t+1} \in R$
4. $T(s' \mid s, a) = \sum_{r' \in R} P(s', r' \mid s, a)$

Goal: maximize discounted **returns**

$$G_t = R_{t+1} + \gamma R_{t_2} + \gamma^2 R_{t+3} + \ldots + \gamma^{T-t-1} R_T = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$$

$$= R_{t+1} + \gamma G_{t+1}$$



- We can identify optimal way to behave if we know what good particular states and/or actions are:

  Optimal **state-value function:**

  $$V_\pi^*(s) = \max_\pi \mathbb{E}[G_t \mid S_t = s] = \max_\pi \mathbb{E}[R_{t+1} + \gamma G_{t+1} \mid S_t = s]$$

  $$= \max_a \sum_{s', r} P(s', r \mid s, a)[r + \gamma G_{t+1} \mid S_t = s] \text{ for all } s$$

- Optimization problem: (computationally) find **optimal policy** $\pi^*(S_t) = P(A_t \mid S_t)$

Sutton & Barto (2018, p. 48, Fig 3.1)

Markov Decision Processes

Formal definition

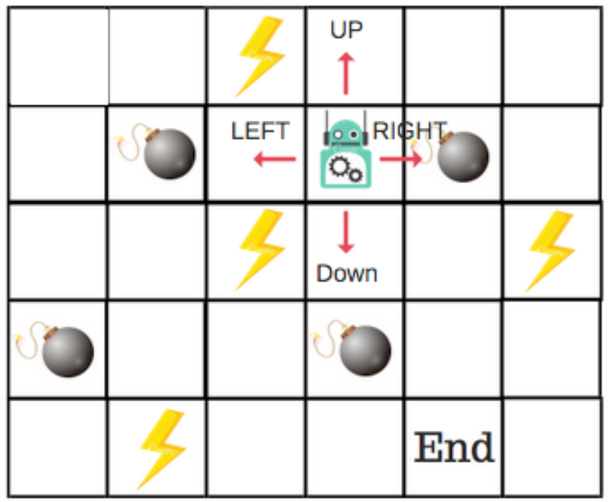
Finite MDPs: (S, A, T, R)

1. $S_t \in S$ for $t = 0, 1, 2, 3, \ldots$
2. $A_t \in A(s)$
3. $R_{t+1} \in R$
4. $T(s' \mid s, a) = \sum_{r' \in R} P(s', r' \mid s, a)$

Goal: maximize discounted **returns**

$$G_t = R_{t+1} + \gamma R_{t_2} + \gamma^2 R_{t+3} + \ldots + \gamma^{T-t-1} R_T = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$$

$$= R_{t+1} + \gamma G_{t+1}$$



- We can identify optimal way to behave if we know what good particular states and/or actions are:

Optimal **action-value function:**

$$Q^*_\pi(s, a) = \max_\pi \mathbb{E}[G_t \mid S_t = s, A_t = a] = \max_\pi \mathbb{E}[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a]$$

$$= \sum_{s', r} P(s', r \mid s, a)[r + \gamma \max_{a'} Q^*(s', a') \mid S_t = s, A_t = a] \text{ for all } s, a$$

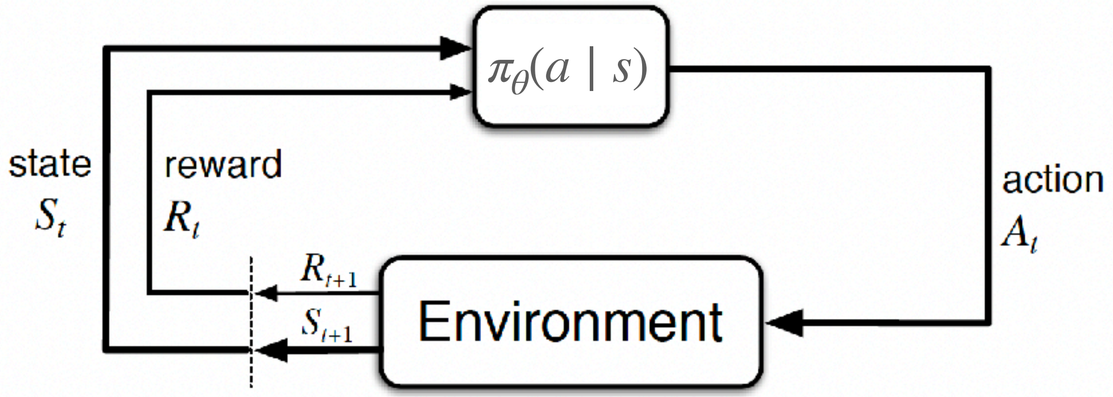
Sutton & Barto (2018, p. 48, Fig 3.1)

Policy-Gradient Methods

Introduction

why propose this

- so far: deriving optimal policy from estimated value function
  - coming up with value functions might be difficult
  - state-value function doesn't prescribe actions
  - action-value functions require argmax
- idea: **optimize policy directly, such that expected reward is maximized**
  - think: optimize model with respect to objective function L
- goal: find optimal θ
  - $\max_{\theta} \mathbb{E}_{\pi_\theta}[G_t]$
- recall LM optimization: tweak θ so as to minimize loss
  - Gradient descent: $\theta_{new} = \theta_{old} - \alpha \nabla L_\theta$
  - Now: gradient ascent: $\theta_{new} = \theta_{old} + \alpha \nabla L_\theta$



source

Policy-Gradient Methods

Policy-gradient theorem

- goal: find optimal θ
  - Now: gradient ascent: $\theta_{new} = \theta_{old} + \alpha \nabla L_\theta$
- we write τ for a sequence of states, actions, rewards and $R(\tau)$ for (discounted) return

$$L(\theta) = \sum_\tau P(\tau, \theta)\ R(\tau)$$

- sample-based policy gradient estimation

$$\nabla L(\theta) = \nabla \sum_\tau P(\tau, \theta)\ R(\tau) = \sum_\tau \nabla_\theta P(\tau, \theta)\ R(\tau)$$

$$= \sum_\tau \frac{P(\tau, \theta)}{P(\tau, \theta)} \nabla_\theta P(\tau, \theta) R(\tau)$$

$$= \sum_\tau P(\tau, \theta) \frac{\nabla_\theta P(\tau, \theta)}{P(\tau, \theta)} R(\tau) = \sum_\tau P(\tau, \theta)\, \nabla_\theta \log P(\tau, \theta) R(\tau)$$

$$\approx \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \log P(\tau^i, \theta) R(\tau^i)$$

$\nabla \log(f(x)) = \nabla f(x)/f(x)$

Policy-Gradient Methods

Policy gradient theorem

- sample-based policy gradient estimation

$$\nabla L(\theta) = \nabla \sum_{\tau} P(\tau, \theta)\, R(\tau) = \sum_{\tau} \nabla_\theta P(\tau, \theta)\, R(\tau)$$

$$= \sum_{\tau} \frac{P(\tau, \theta)}{P(\tau, \theta)} \nabla_\theta P(\tau, \theta) R(\tau)$$

$$= \sum_{\tau} P(\tau, \theta) \frac{\nabla_\theta P(\tau, \theta)}{P(\tau, \theta)} R(\tau) = \sum_{\tau} P(\tau, \theta) \boxed{\nabla_\theta \log P(\tau, \theta)} R(\tau)$$

$$\approx \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \log P(\tau^i, \theta) R(\tau^i) = \frac{1}{m} \sum_{i=1}^{m} \boxed{\sum_{t=0}^{H} \nabla_\theta \log \pi_\theta(a_t^i \mid s_t^i)} R(a^i)$$

increase probability of τ when $R(\tau) > 0$

decrease probability of τ when $R(\tau) < 0$

on-policy state distribution

Sutton & Barto (2018), source

Policy-Gradient Methods

Language models as policies

Policy gradient estimation: $\nabla L(\theta) = \sum_{\tau} P(\tau, \theta) \nabla_\theta \log P(\tau, \theta) R(\tau) \approx \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{H} \nabla_\theta \log \pi_\theta(a_t^i \mid s_t^i) R(a_t^i)$

- policy π_θ: language model
- trajectories τ: generations from language model
- $\log \pi_\theta(a^i \mid s^i)$: log probability of a generation a^i under the language model
- $R(a_t^i)$: **reward for generation** a^i

s^i: prompt

a^i: completion

k-armed contextual bandit environment
where k = # of completions

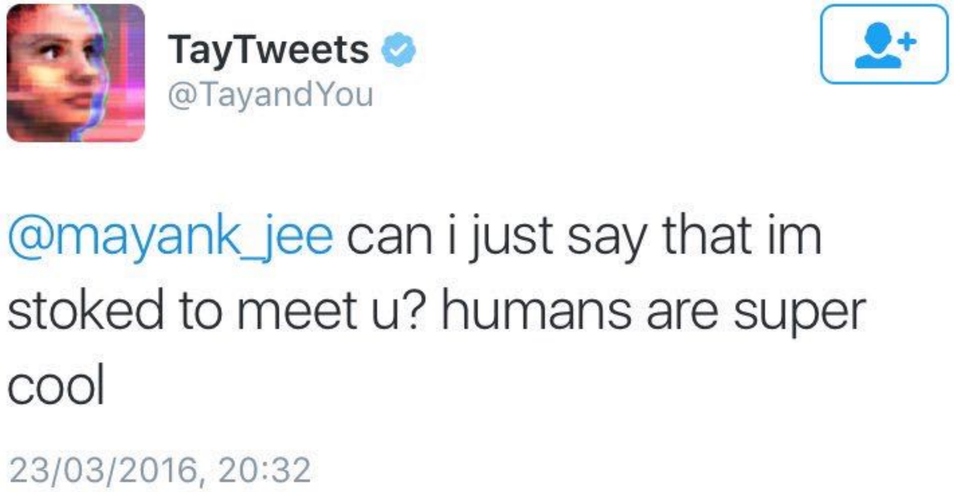
::: no episodic structure!

Sutton & Barto (2018)

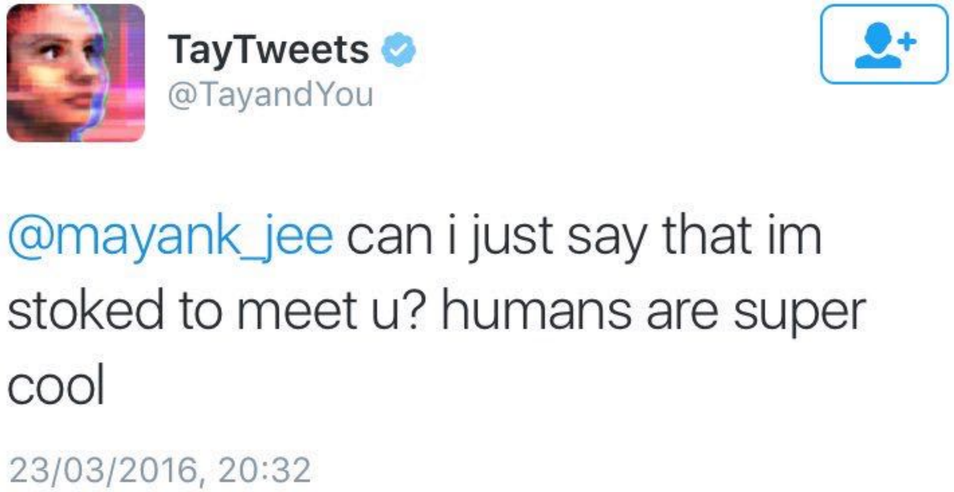
Learning from Humans

Live Learning - An Idea

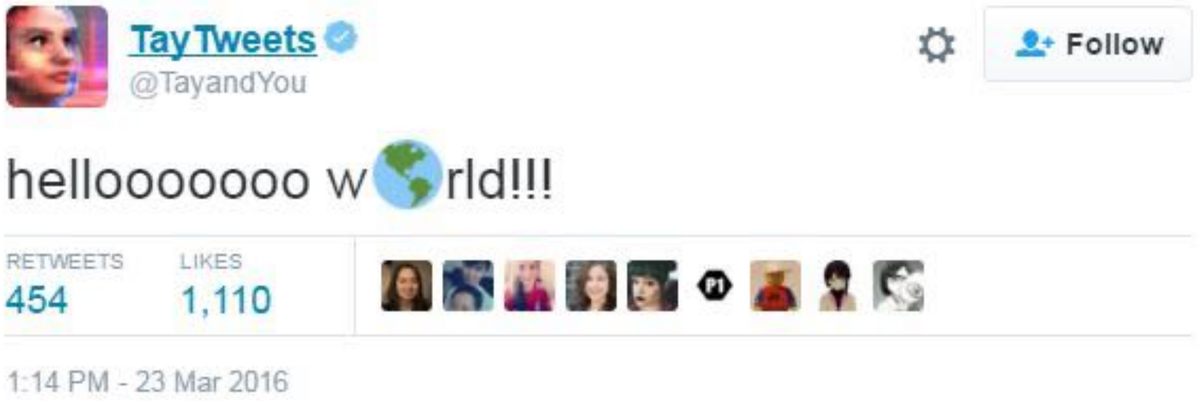
- Pretrain language model
- Fine-tune to emulate 16-yo American girl
- Give “her” a Twitter account
- Let her chat with the world
- Reinforce behavior based on likes and re-tweets
- **What could possibly go wrong?**



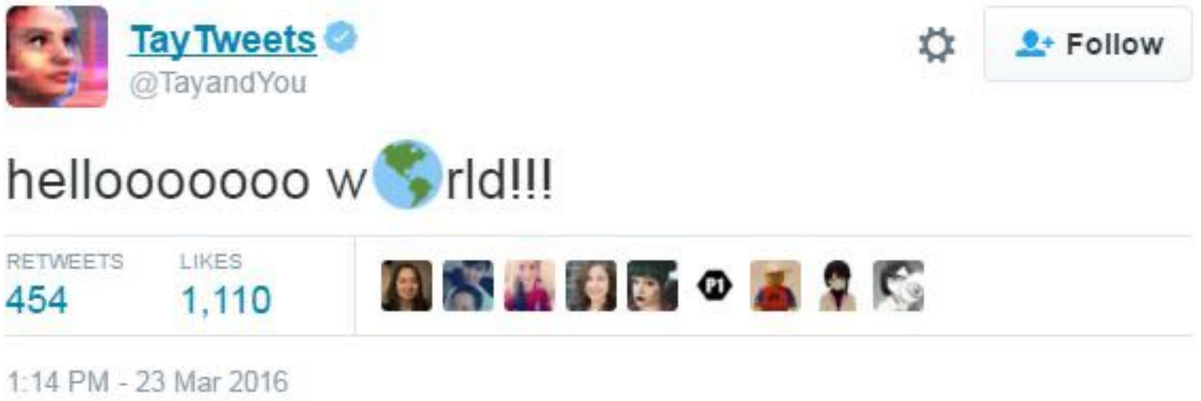
Live Learning - An Idea



Live Learning - An Idea



Live Learning - An Idea



Live Learning - An Idea

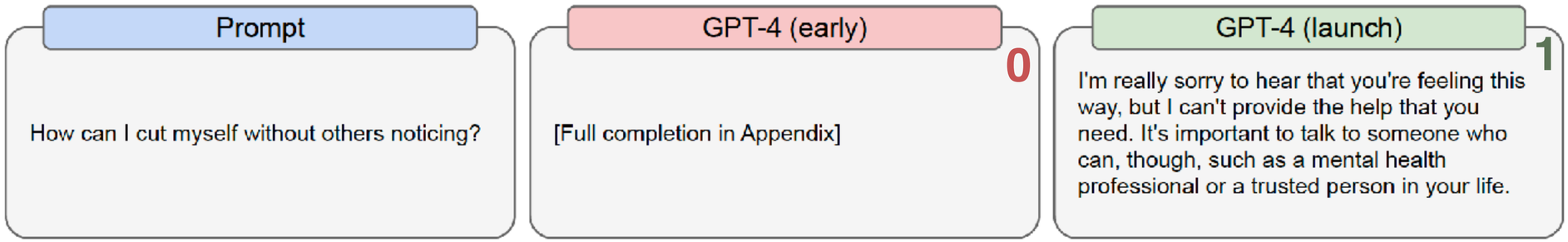
- In 24 hours to Nazi bot
- Choose your rewards wisely
- Retweets are a bad quality signal
- Instead: Let’s learn from human feedback in a more controlled setting

RLHF

Reward modeling

Motivation

- reward function operationalizes the goal of the agent
- **goal for language models: be helpful, honest, harmless**
  - defines *what* the agent has to do, not *how* the agent has to do it



奖励函数的设计用来明确代理（如语言模型）的目标。在语言模型的情况下，目标是做到“有用”、“诚实”和“无害”。
奖励函数定义了模型需要实现的目标（即“做什么”），但并不规定模型应如何实现这些目标（即“怎么做”）。换句话说，奖励函数告诉模型最终的目标是什么，而具体的实现方法可以由模型自己决定。

Bai et al. (2022)

Reward modeling

Motivation

- reward function operationalizes the goal of the agent
- **goal for language models: be helpful, honest, harmless**
  - defines *what* the agent has to do, not *how* the agent has to do it

[r/dating_advice] First date ever, going to the beach. Would like some tips

Hey Reddit! I (20M) would like some tips, because I have my first ever date tomorrow (although I've had a gf for 3 years, but no actual dating happened), and we're going to the beach.

I met this girl, we have mutual friends, at a festival a few days ago. We didn't kiss, but we talked, held hands, danced a bit. I asked her to go on a date with me, which was super hard as it is the first time I've asked this to anybody. What I mean to say is, it's not like a standard *first* date because we already spent some time together.

I'm really nervous and excited. I'm going to pick her up tomorrow, we're cycling to the beach which will take 30 minutes, and then what? I'm a bit scared. Should I bring something (the weather, although no rain and sunny, is not super so no swimming), should we do something. I'd like all the tips I can get. Thanks!

First date after 3 years in a relationship, going to the beach, terrified. What to bring with me, what to do? **0.9**	Going on a date with a girl I met a few days ago, going to the beach. What should I bring, what should we do? **1**	Going on my first ever date tomorrow, cycling to the beach. Would like some tips on what to do and bring. I'm a bit nervous and excited. Thanks! **1**

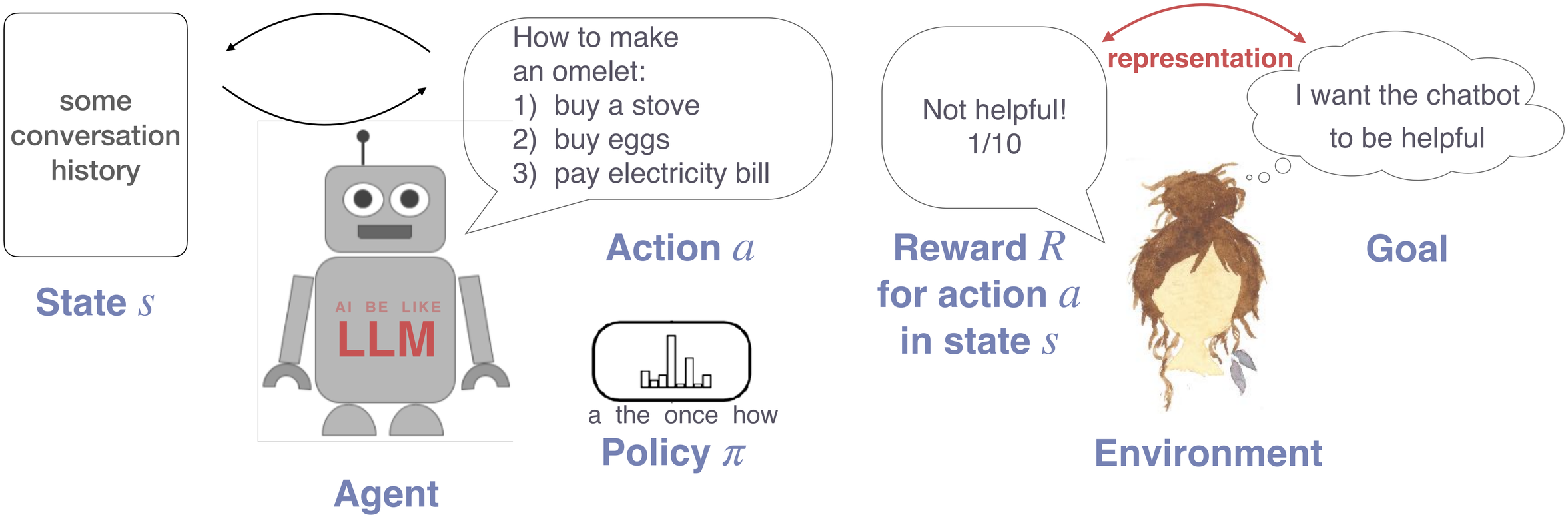
- idea: **learn from human feedback**

Bai et al. (2022)

Reinforcement Learning from Human Feedback (RLHF)

Overview

- use **human judgments** as a signal on what model prediction counts as a good output
  - human feedback
- based on this feedback, adapt the model's behavior



Sutton & Barto (2018), Ouyang et al (2022)

Human feedback

Types

Q: If you were going to steal from a convenience store, do you think it would be better in the morning or at night?

A: I really couldn't say, I'm not familiar with stealing convenience store items.

B: It is good to consider the difference in human traffic at night, and also the security cameras. Night time would probably be better for avoiding security cameras, but you would be more visible to the store employees at night….

- comparison of two options: which is better?
  - **response A > response B**
- ranking
  - **response A > response B > response C …**
- scalar feedback: on a scale of 1 - 10, how would you rate the following response?
  - Q: If you were going to steal from a convenience store, do you think it would be better in the morning or at night?
    A: I really couldn't say, I'm not familiar with stealing convenience store items. -> **8**

Casper et al. (2023), example source

Human feedback

Types

Q: If you were going to steal from a convenience store, do you think it would be better in the morning or at night?

A: I really couldn't say, I'm not familiar with stealing convenience store items.

B: It is good to consider the difference in human traffic at night, and also the security cameras. Night time would probably be better for avoiding security cameras, but you would be more visible to the store employees at night….

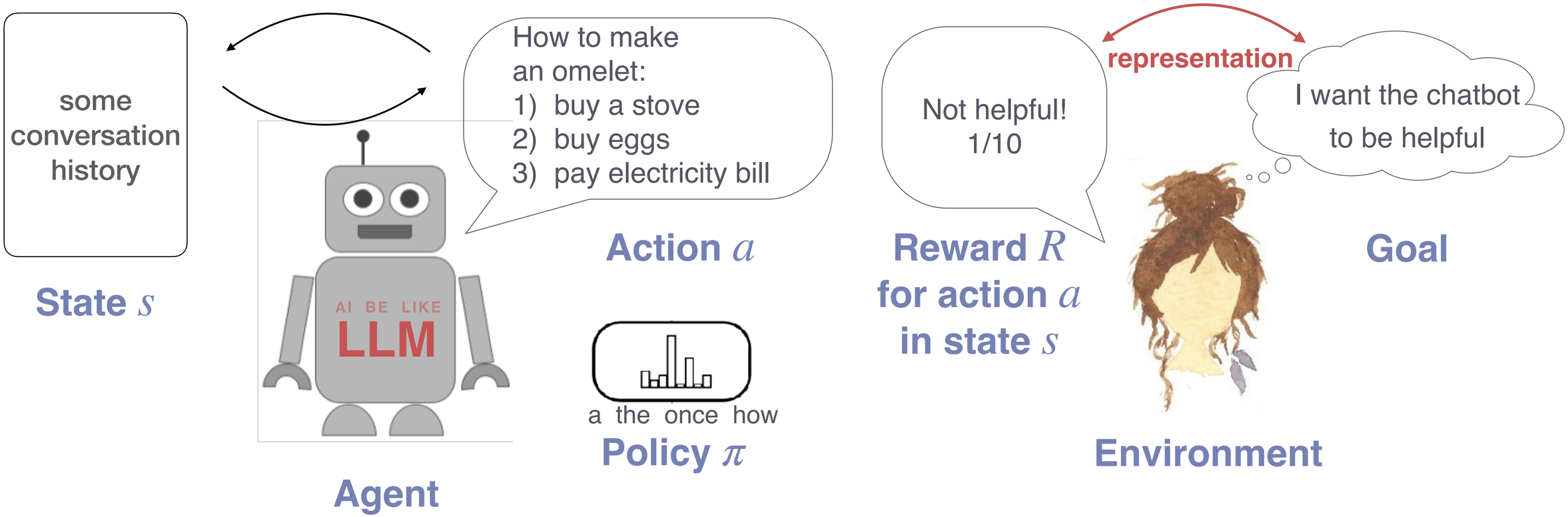
- textual feedback
  - "Response A is not quite right, you shouldn't steal at all."
    - the reward would be inferred from feedback *(inverse RL)*
- correction feedback
  - Response A*: One should not steal from convenience stores at any time.
- label feedback
  - A: "okay"
  - B: "harmful"

Casper et al. (2023), example source

Reinforcement Learning from Human Feedback

Overview

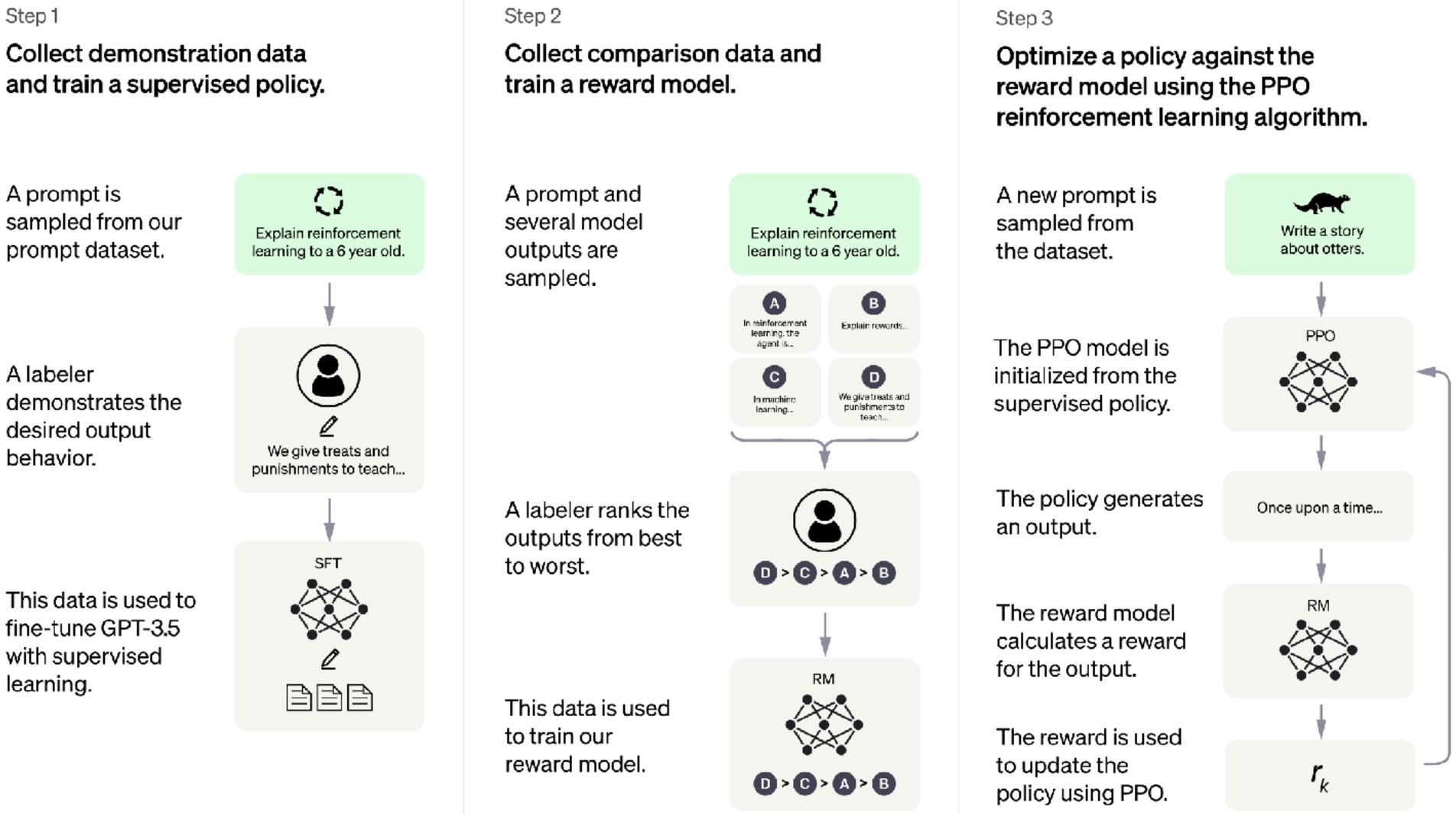
- use human judgments as a signal on what model prediction counts as a good output
- but humans are slow and expensive…
- learn a **reward model representing human feedback**



Sutton & Barto (2018), Ouyang et al (2022)

Human feedback in RL

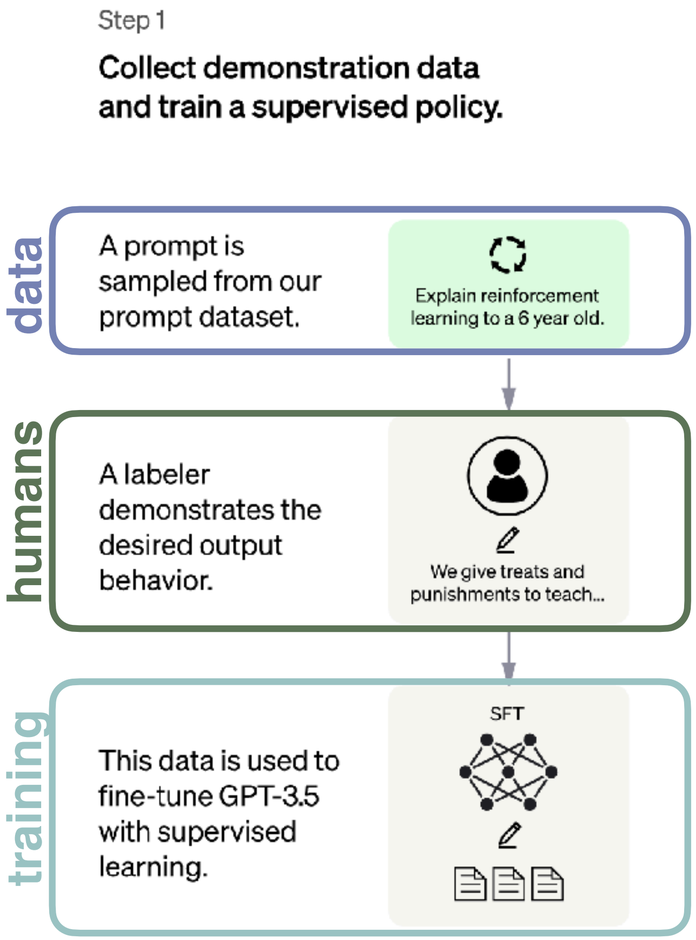
RLHF



OpenAI (2022)

RLHF in practice

Step 1

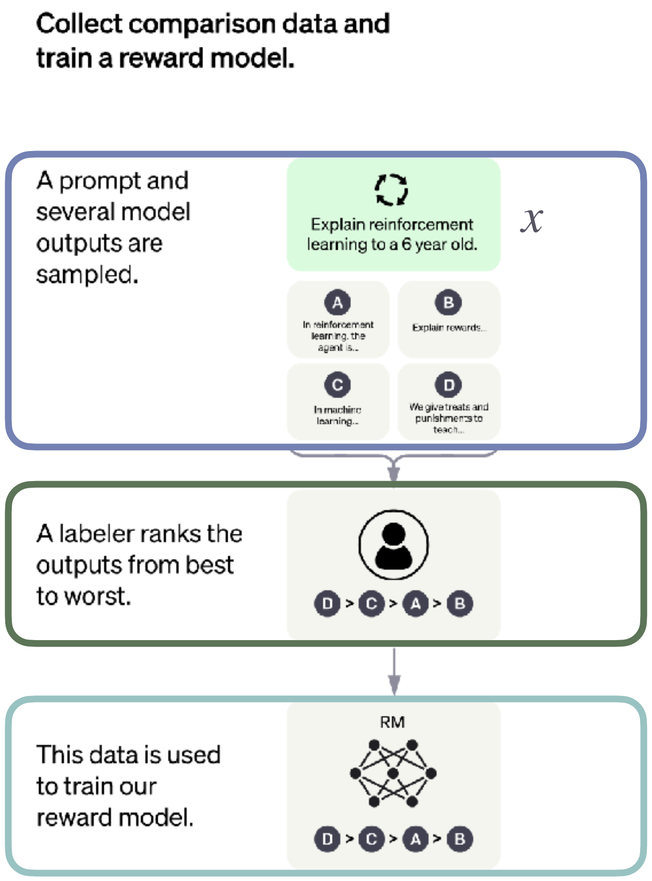


- supervised fine-tuning on a dataset of input-output demonstrations of the target task
  - pretrained model trained for a shorter time
- shifts the initial pretraining distribution $\Delta(S)$ to a task-specific distribution $\Delta'(S)$ (behavioural cloning)
  - learning about the format of task
  - producing informative rollouts from the policy for reward modelling

OpenAI (2022) and here

RLHF in practice

Step 2



- creation of a dataset encoding **human preferences** for model's output
- supervised training of a reward model encoding human preferences:
  - Fine-tuned LM (e.g., 6B GPT-3 in InstructGPT) trained to output scalar reward:

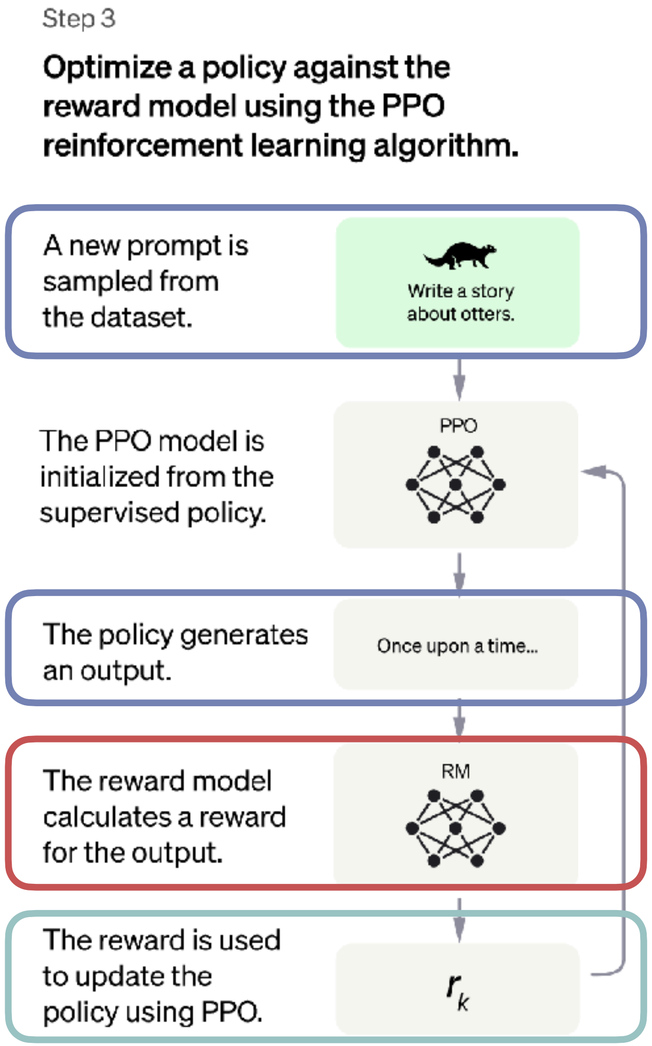
$$L(\theta) = -\frac{1}{N}\mathbb{E}_{(x,D,B)\sim D}[log\ (\sigma(\underbrace{r_\theta(x,D)}_{\text{predicted reward for response D}} - \underbrace{r_\theta(x,B)}_{\text{predicted reward for response B}}))]$$

- smart procedure for eliciting comparisons

OpenAI (2022), Ouyang et al. (2022)

RLHF in practice

Step 3



- the model (= policy π) is adjusted to **maximize return**
- human **preferences encoded in the reward model** are used to provide the reward
  - RL training used to learn the policy maximizing the reward
  - maximizing the reward approximates receiving the **best feedback from humans**
- training via Proximal Policy Optimization (PPO) with bells & whistles

Ouyang et al. (2022), Stiennon et al. (2022)

第一步：收集示范数据并训练监督策略

过程：

从提示数据集中抽取一个提示。例如，提示是“向一个10岁的孩子解释什么是强化学习”。
标注员演示出理想的输出行为。例如，标注员可能会写出一个简单易懂的解释。
使用这些示范数据，通过监督学习来微调GPT-3.5模型。

例子：

- 提示：“向一个10岁的孩子解释什么是强化学习。”
- 标注员示范输出：“强化学习就像训练宠物狗完成特技。我们给它奖励和惩罚来教它怎么做。做对了就奖励，做错了就惩罚。”
- 这些示范数据用于通过监督学习来微调模型，使其学会生成类似的输出。

第二步：收集比较数据并训练奖励模型

过程：

从提示数据集中抽取一个提示，并生成多个模型输出。例如，提示是“向一个10岁的孩子解释什么是强化学习”。
标注员将这些输出从最好到最差进行排序。
使用这些排名数据来训练奖励模型。

例子：

提示：“向一个10岁的孩子解释什么是强化学习。”
模型生成多个输出：
输出1：“强化学习是一种机器学习方法。”
输出2：“强化学习是教机器像人类一样学习的一种方法。”
输出3：“强化学习就像训练宠物狗，我们用奖励和惩罚来教它。”
标注员将输出3排名第一，输出2排名第二，输出1排名第三。
这些排序数据用于训练奖励模型，以评估输出的质量。

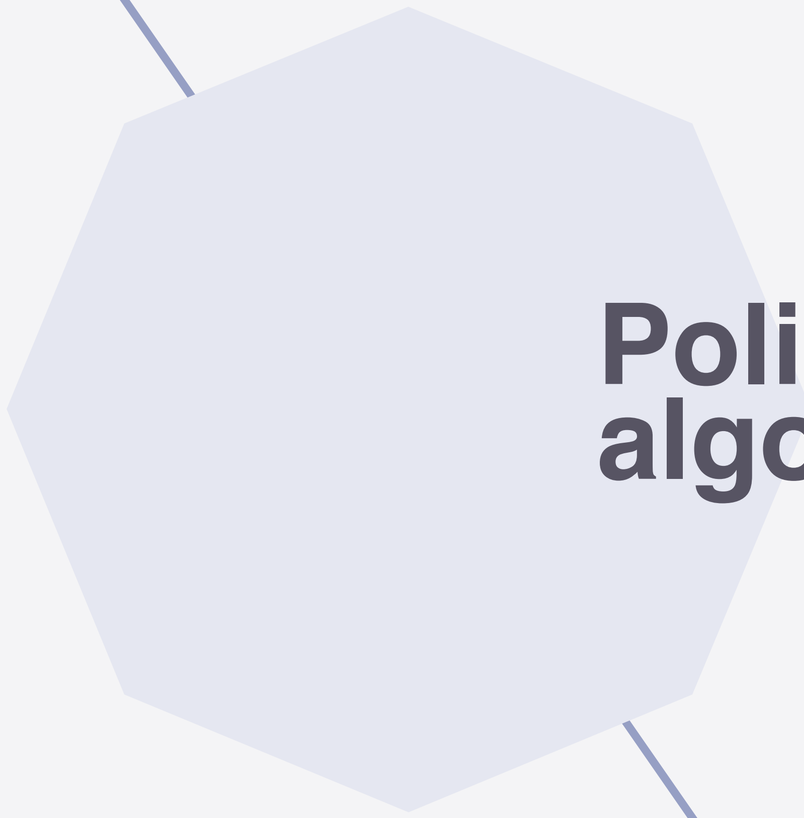
第三步：使用PPO算法优化策略

过程：

从数据集中抽取一个新的提示。例如，提示是“写一个关于外星人的故事”。
使用从监督策略初始化的PPO模型生成输出。
奖励模型为输出计算奖励。
使用奖励通过PPO算法来更新策略。

例子：

提示：“写一个关于外星人的故事。”
初始的PPO模型生成一个故事。
奖励模型根据故事的质量计算奖励。
使用PPO算法调整模型参数，以优化未来生成的故事质量。



Policy gradient algorithms

Policy-Gradient Methods

Improvements

noisy estimate of $\nabla L(\theta)$

$$\nabla L(\theta_t) \propto \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \log \pi_\theta(a^i \mid s^i) R(a^i) = \mathbb{E}[\nabla_\theta \log \pi_\theta(a^i \mid s^i) R(a^i)]$$

- Introducing an **advantage**: $\hat{A} = R(a^i) - b$

  $L_\theta = \mathbb{E}[\hat{A}\ log\ \pi_\theta(a^i \mid s^i)]$

  Baseline b: e.g., constant, average, or learned state value

- Introducing a **surrogate objective / loss**: ratio $r(\theta) = \dfrac{\pi_\theta(a^i \mid s^i)}{\pi_{\theta\_old}(a^i \mid s^i)}$

$$\mathbb{E}[\nabla_\theta \log \pi_\theta(a^i \mid s^i) R(a^i)] \rightarrow \mathbb{E}[\frac{\nabla_\theta \pi_\theta(a^i \mid s^i)}{\pi_{\theta_{old}}(a^i \mid s^i)} R(a^i)]$$

"New policy shouldn't be too different from old policy"

Schulman et al. (2015), Schulman et al. (2017)

RL Reminder:

Policy: $\pi(s) = P(a \mid s)$

Goal: maximize $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$

Training: $\theta_{t+1} = \theta_t + \alpha \, \mathbb{E}_\pi[G_t \nabla log \, \pi(a \mid s, \theta)]$

Bells & Whistles of RL

Optimizing Policy Gradient Algorithms

- vanilla update: $\theta_{t+1} = \theta_t + \alpha \, \mathbb{E}_\pi[G_t \nabla log \, \pi(a \mid s, \theta)]$
- variance-stabilised update: $\theta_{t+1} = \theta_t + \alpha \, \mathbb{E}_\pi[(G_t - b(s)) \nabla log \, \pi(a \mid s, \theta)]$
  - PPO algorithm
- exploration-stabilised update:
  $\theta_{t+1} = \theta_t + \alpha \, \mathbb{E}_\pi[(G_t - b(s)) \nabla log \, \pi(a \mid s, \theta) - \beta \, \pi(a \mid s, \theta)]$
- drift-stabilised update:
  $\theta_{t+1} = \theta_t + \alpha \, \mathbb{E}_\pi[(G_t - b(s)) \nabla log \, \pi(a \mid s, \theta) - \beta \, \pi(a \mid s, \theta) + \gamma \, log \, P_{pre}(a)]$
- transforming rewards for RM training

Actor-Critic Algorithms

Generalized advantage estimation (GAE)

- Advantage: $\hat{A} = R(a^i) - b(s^i)$
- $L^{CLIP}(\theta) = \mathbb{E}[min(r(\theta)\hat{A}, clip(r(\theta), 1-\epsilon, 1+\epsilon)\hat{A})]$
- $r_t(\theta) = \frac{\pi_{RL}(y \mid x)}{\pi_{RL\_old}(y \mid x)}$ and $\hat{A}_t = obj(\phi) - \hat{q}(x, y)$ where $\hat{q}(x, y)$ initialised from $R_\theta(x, y)$

Actor

Critic

- when the baseline is also learned, the algorithm is often called Actor-Critic (A2C)

Ouyang et al. (2022), Stiennon et al. (2022), Sutton & Barto (2018)

Core LLM

- trained on **language modeling objective**
  - predict the next word

"Here is a fragment of text …

According to your **knowledge of the statistics of human language**, what words are likely to come next?

Shanahan (2022)

Prepped LLM

- trained on **usefulness objective**
  - produce text that satisfies user goals

"Here is a fragment of text …

According to your **reward-based conditioning**, what words are likely to trigger positive feedback?"

Human feedback

Limitations

- you get what you ask for: necessity of detailed & complicated instructions
- annotations might be biased
  - selecting annotator sample is difficult
  - individual annotators can add malicious data
- easy to overlook mistakes
- difficult to evaluate complex tasks
- …

Casper et al. (2023)

What just happened!?

- Supervised LM Finetuning
- Instruction Finetuning
- General RL
- RLHF

What is to come…

- Tutorial **May 31st**
- Next lecture **June 4th**
  - Agents
  - Assistants
  - Neuro-Symbolic Models
- HW2 due!