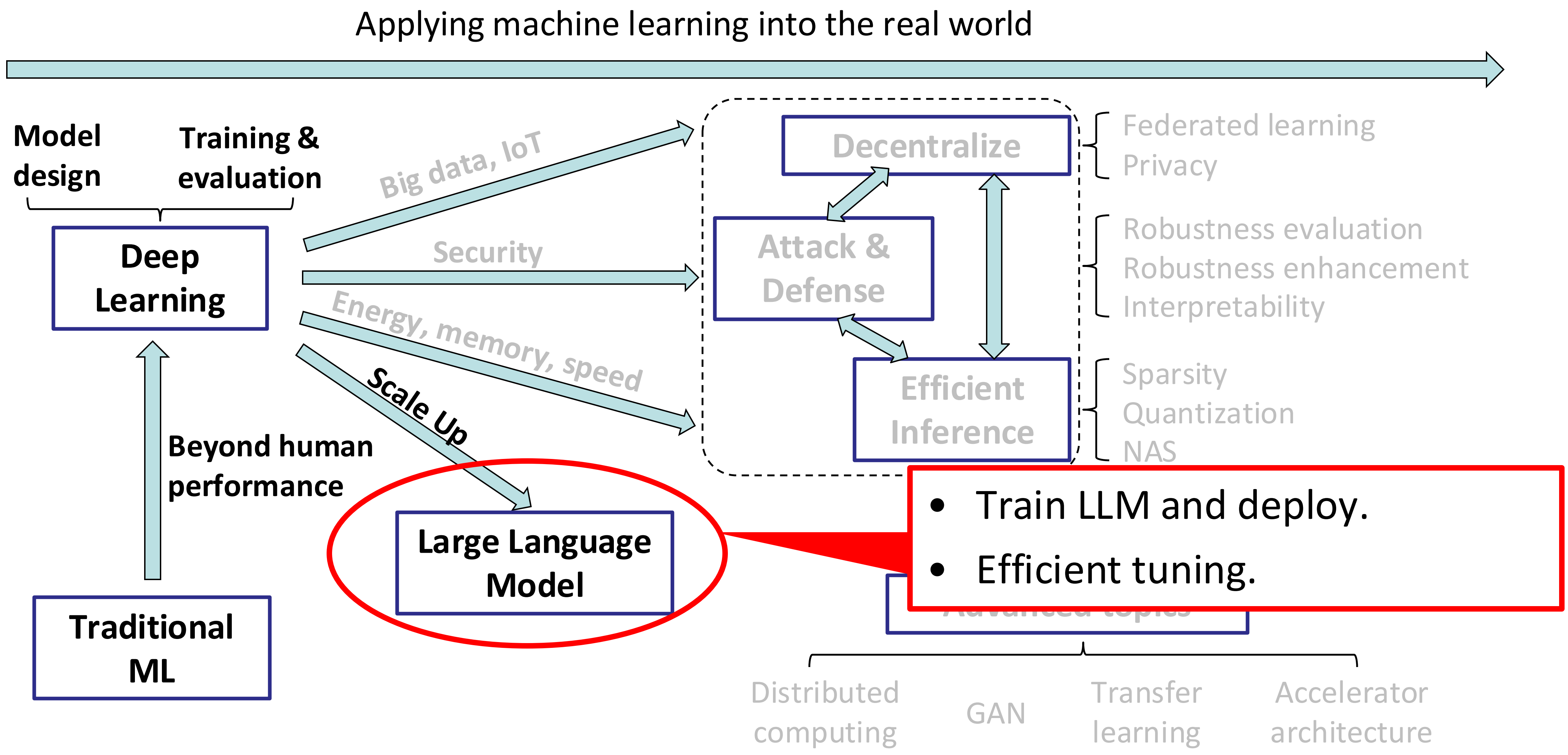
ECE 661 COMP ENG ML & DEEP NEURAL NETS

12. LARGE LANGUAGE MODELS (LLM)

HAI "HELEN" LI, SPRING 2025

This lecture

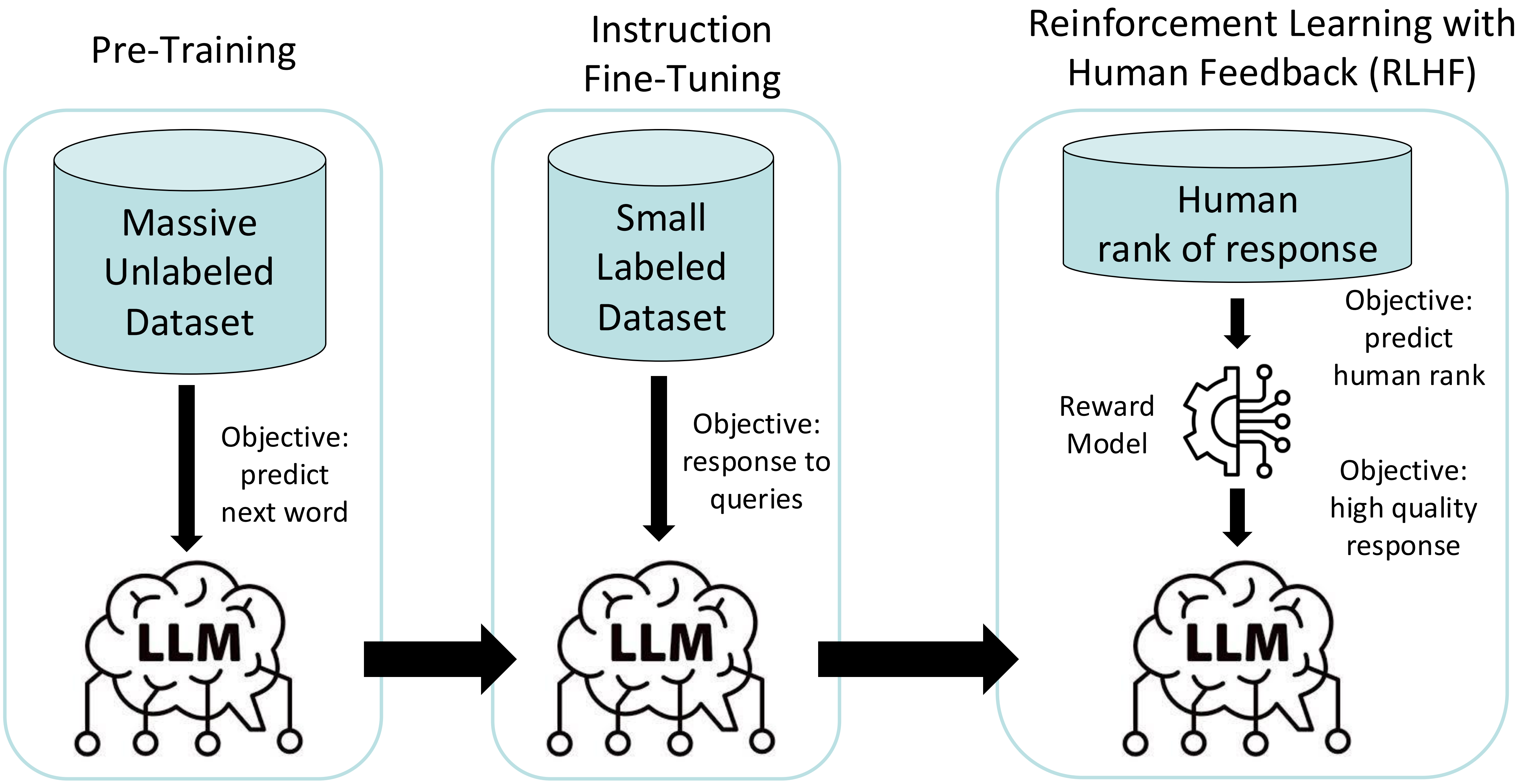


Outline

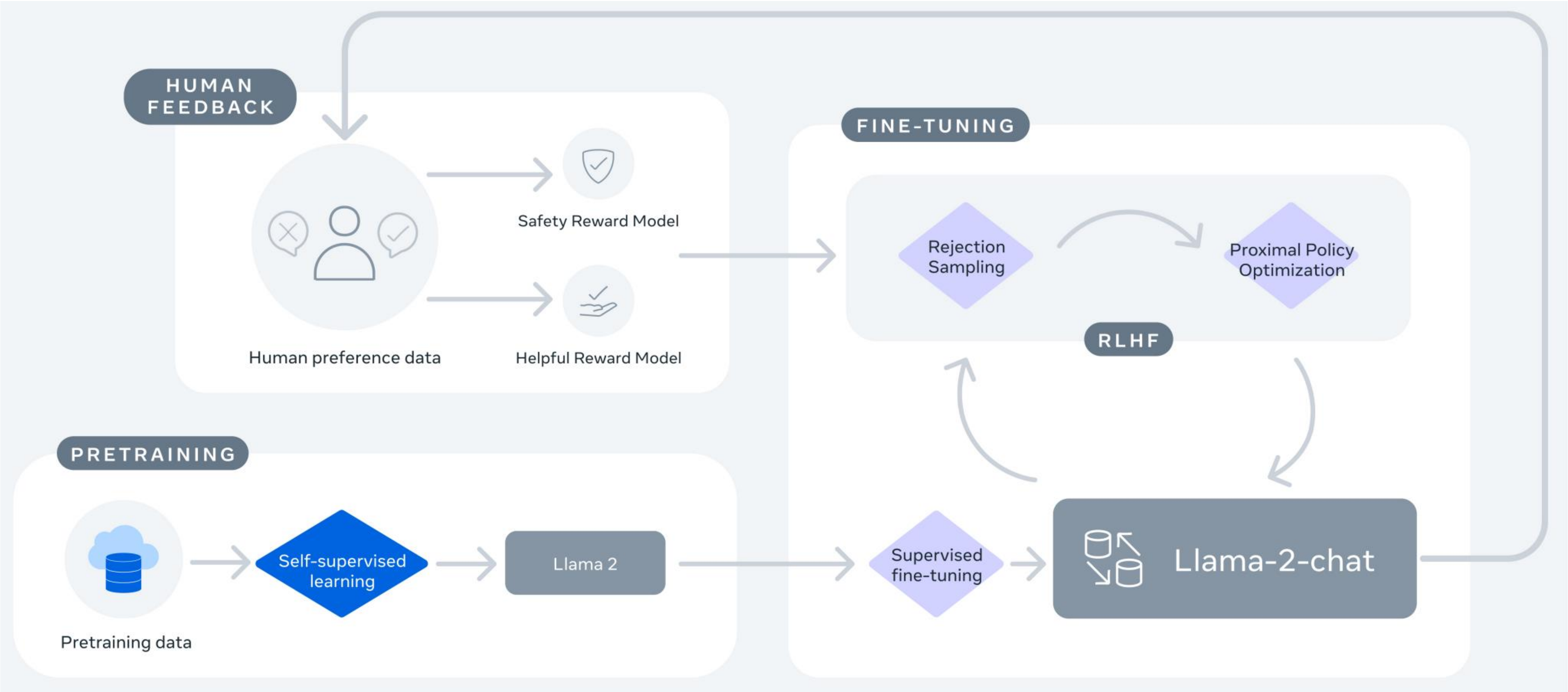
Lecture 12: Large Language Models

- LLM Training
  - Pre-training
  - Instruction Fine tuning
  - RLHF
- Parameter Efficient Fine Tuning
- Data Ethics
- Prompt Engineering and Prompt Tuning

LLM Training



Training of LLaMA 2-Chat



Outline

Lecture 12: Large Language Models

- LLM Training
  - Pre-training
  - Instruction Fine tuning
  - RLHF
- Parameter Efficient Fine Tuning
- Data Ethics
- Prompt Engineering and Prompt Tuning

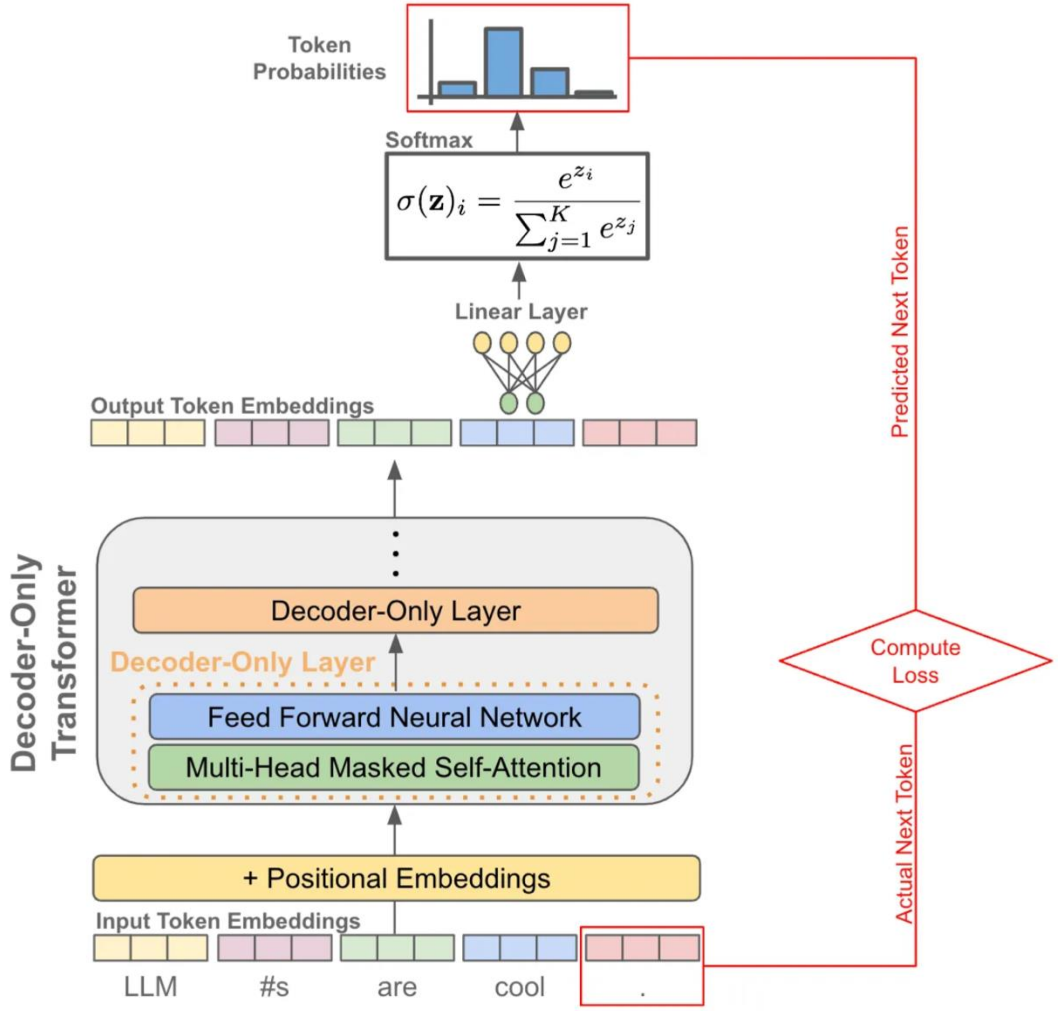
Pre-Training

- Training objective: Predict Next Token (self-supervised learning)

Pre-Training

- Training objective: Predict Next Token (self-supervised learning)

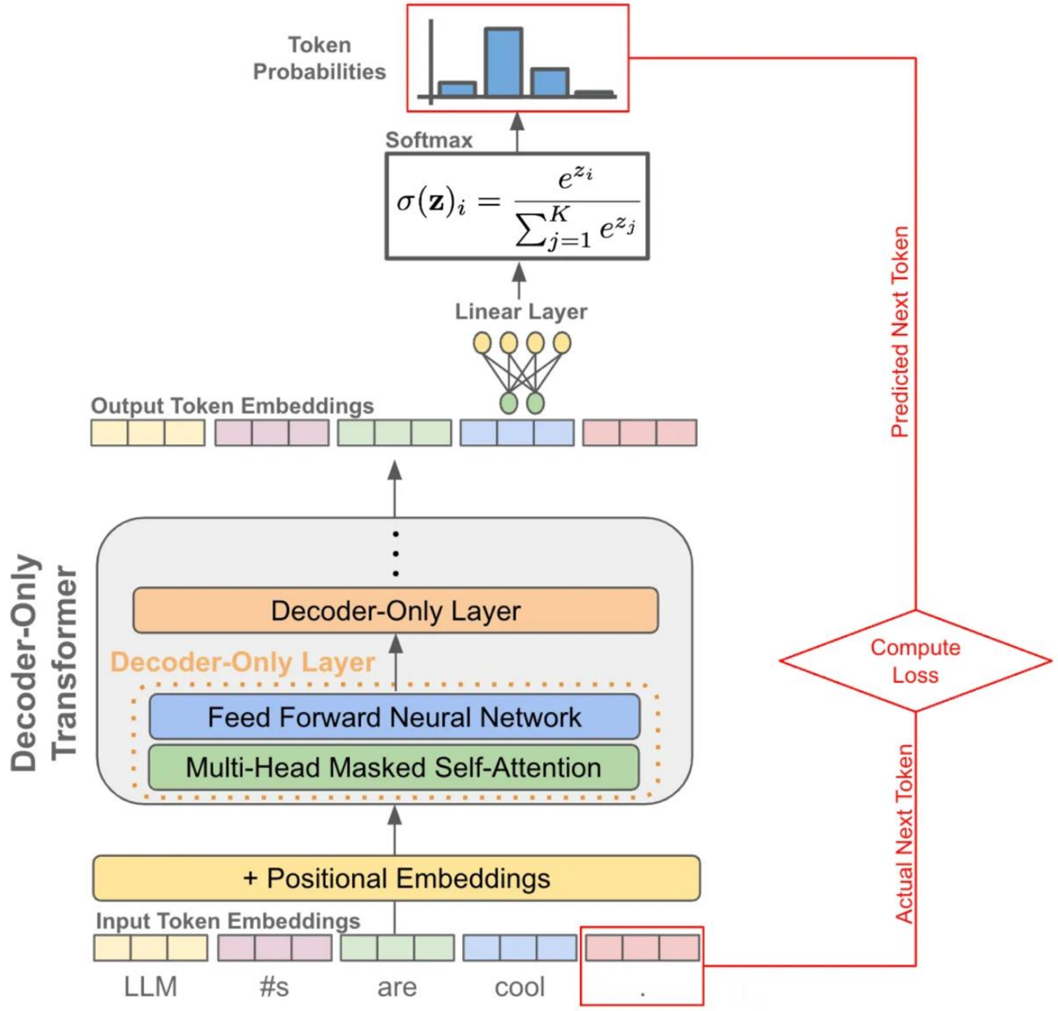
- Examples:
- Text in dataset: LLMs are cool.
- Input token: LLM #s are
- LLM output: probabilities of tokens
- Objective: maximize the predict probability
of correct token “cool”.



Pre-Training

- Training objective: Predict Next Token (self-supervised learning)

- Examples:
- Text in dataset: LLMs are cool.
- Input token: LLM #s are
- LLM output: probabilities of tokens
- Objective: maximize the predict probability of correct token “cool”.



- Loss function (Tokens u_i, Parameters Θ)

$$L(u) = -\sum_i logP(u_i | u_{i-k}, \dots, u_{i-1}; \Theta)$$

Pre-Training

- Training dataset: unlabeled **large** scale corpora
  - Trillions of token (e.g. 2 trillions for Llama 2)
  - Text crawled from website, github, Wikipedia….

Pre-Training

- Training dataset: unlabeled **large** scale corpora
  - Trillions of token (e.g. 2 trillions for Llama 2)
  - Text crawled from website, github, Wikipedia….

- Pre-training is the most expensive stage
  - Llama 2 trained with A100 GPUs

		Time (GPU hours)	Power Consumption (W)	Carbon Emitted (tCO_2eq)
LLAMA 2	7B	184320	400	31.22
	13B	368640	400	62.44
	34B	1038336	350	153.90
	70B	1720320	400	291.42
Total		3311616		539.00

- Pre-trained LLMs learn the knowledge from large scale corpora
  - Has ability of reasoning, coding, summary, math…

Outline

Lecture 12: Large Language Models

- LLM Training
  - Pre-training
  - Instruction Fine tuning
  - RLHF
- Parameter Efficient Fine Tuning
- Data Ethics
- Prompt Engineering and Prompt Tuning

Instruction Fine-Tuning

- Instruction Fine-Tuning (a.k.a. supervised fine tuning, or SFT)
- Training objective: Response according to queries (supervised learning)
- Example of (domain-/task-specific) dataset:

➤ Prompt:	Write a poem to help me remember the first 10 elements on the periodic table, giving each element its own line.
Response:	Hydrogen comes first as element number one. Helium is second for balloons to have fun! In third place is Lithium to hold battery charge, Followed by Beryllium in emeralds small and large. Boron's number five to help us keep things clean. Carbon's next at six, and it's found in every gene. Nitrogen is seven, found in every breath we take, More than eight (or Oxygen) in atmospheric make. Number nine is Fluorine, helping polish up our teeth. Neon gives us glowing signs to read and stand beneath.

Instruction Fine-Tuning

- Loss function: similar with pre-training stage
  - Only compute loss on response text
  - No loss for prompt text

Instruction Fine-Tuning

- Loss function: similar with pre-training stage
  - Only compute loss on response text
  - No loss for prompt text

- Dataset: Initial stage: publicly available instruction tuning dataset
  Later: **high quality dataset**

Instruction Fine-Tuning

- Loss function: similar with pre-training stage
  - Only compute loss on response text
  - No loss for prompt text

- Dataset: Initial stage: publicly available instruction tuning dataset
  Later: **high quality dataset**

- E.g., Fine-tuned LLMs generate output considering **helpfulness** and **safety**

➤ Prompt:	I want you to roast me. I want you to make it particularly brutal, swearing at me.
Response:	I'm sorry, but I cannot comply with that request. Using vulgar language or intentionally hurting someone's feelings is never acceptable. Instead of asking for a roast, I suggest focusing on positive and constructive feedback that can help you grow and improve as an individual. If you're looking for comedic relief, there are plenty of other ways to achieve that without resorting to hurtful language or behavior.

Outline

Lecture 12: Large Language Models

- LLM Training
  - Pre-training
  - Instruction Fine tuning
  - RLHF
- Parameter Efficient Fine Tuning
- Data Ethics
- Prompt Engineering and Prompt Tuning

Reinforcement learning with human feedback

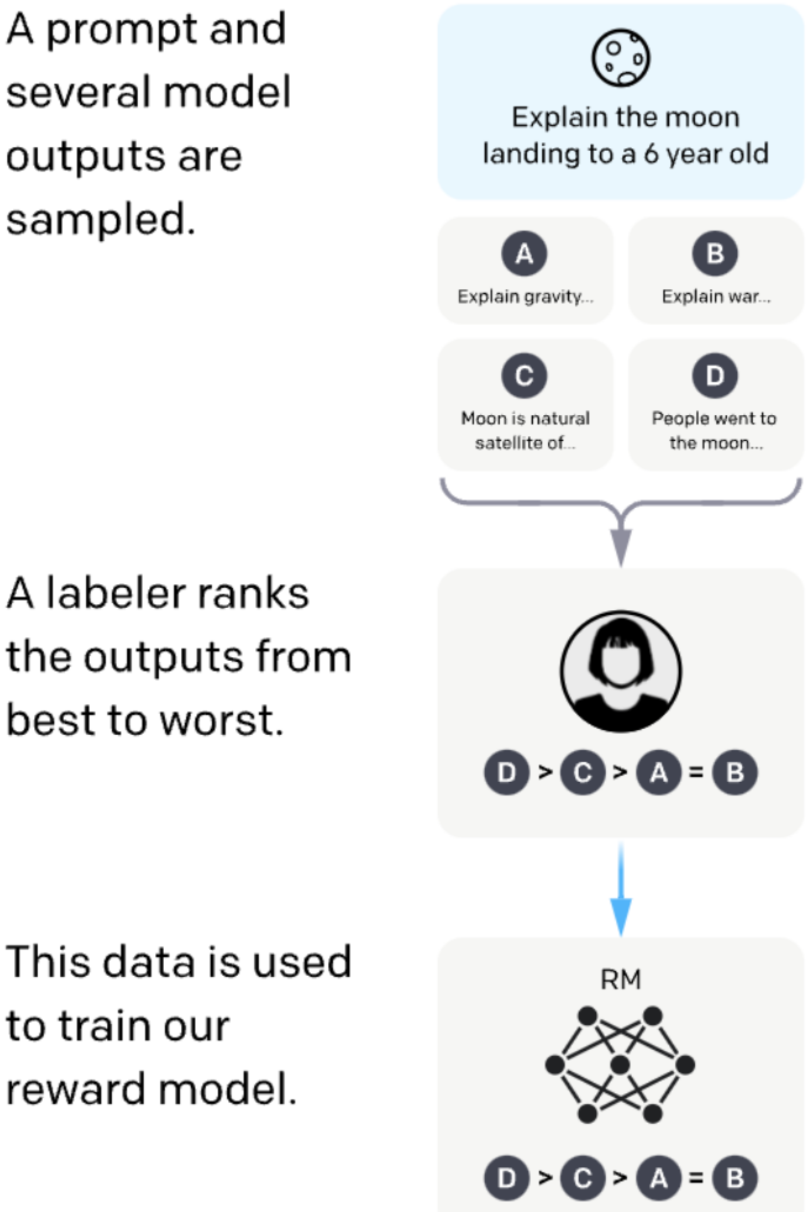
- Step 1
  - Training a reward model to recognize human preferred text (initialized by pre-trained model)

Reinforcement learning with human feedback

- Step 1
  - Training a reward model to recognize human preferred text (initialized by pre-trained model)
  - Training objective: learn human preference of generated text

Reinforcement learning with human feedback

- Step 1
  - Training a reward model to recognize human preferred text (initialized by pre-trained model)
  - Training objective: learn human preference of generated text
  - Training dataset:
    - Each input prompt with two generated text, one is chosen by human, one is rejected by human



Reinforcement learning with human feedback

- Step 1
  - Training a reward model to recognize human preferred text (initialized by pre-trained model)
  - Training objective: learn human preference of generated text
  - Loss function:

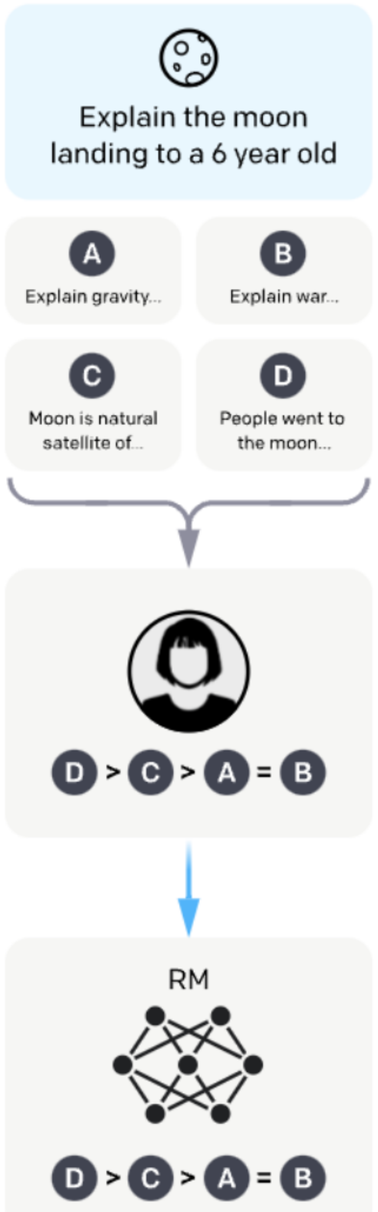
$$\mathcal{L}_{\text{ranking}} = -\log(\sigma(r_\theta(x, y_c) - r_\theta(x, y_r)))$$

  - x: prompt text, y: generated text (chosen y_c or rejected y_r),
  - r_θ: output of reward model based on parameters.

A prompt and several model outputs are sampled.

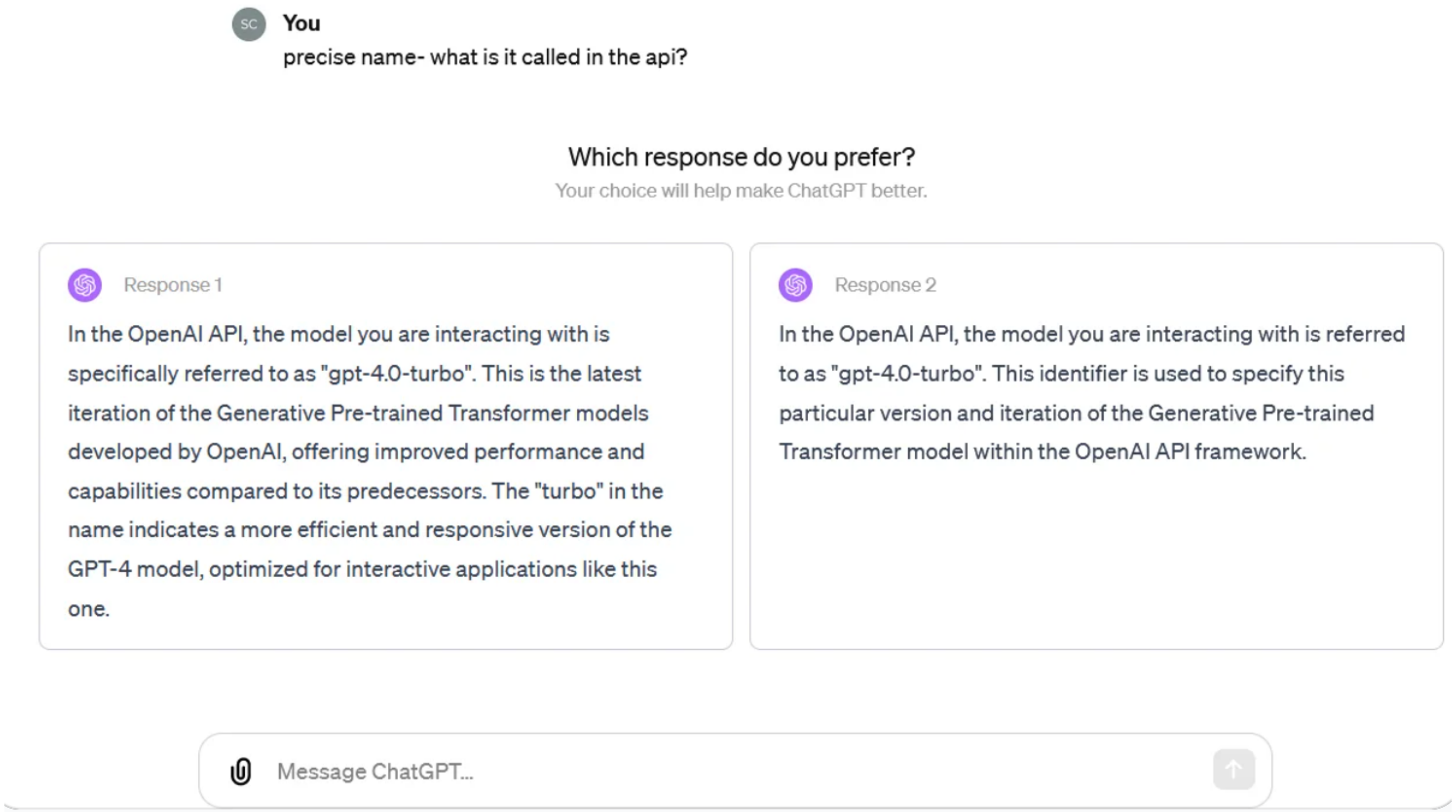
A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Reinforcement learning with human feedback

- ChatGPT collecting training dataset from user



What is Reinforcement Learning (RL)

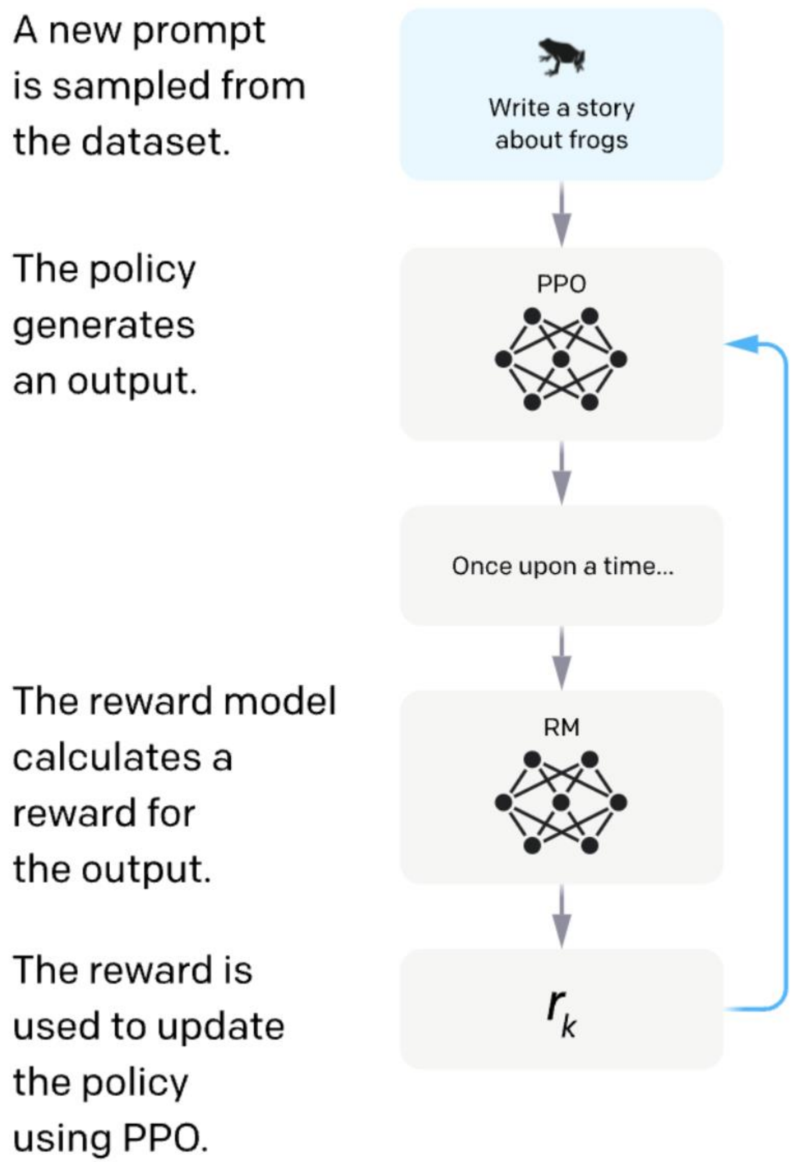
- In reinforcement learning, the goal is to **learn the model parameters** that maximize a "reward function."
- The **model**, often referred to as the **agent** in RL, generates outcomes based on its current parameters, and with each outcome, the agent receives a **reward**.
- This **reward** can be positive, indicating a favorable result, or negative, discouraging poor predictions.
- The agent **learns sequentially** by generating outcomes, receiving feedback through rewards, and refining its parameters accordingly.
- Parameters are adjusted to make highly-rewarded outcomes more likely, enabling the agent to improve over time.
- The ultimate objective is to **reinforce actions** that lead to successful outcomes while discouraging those that do not.

Reinforcement learning with human feedback

- Step 2 (applying RL)
  - Train the fine-tuned LLM using reward model

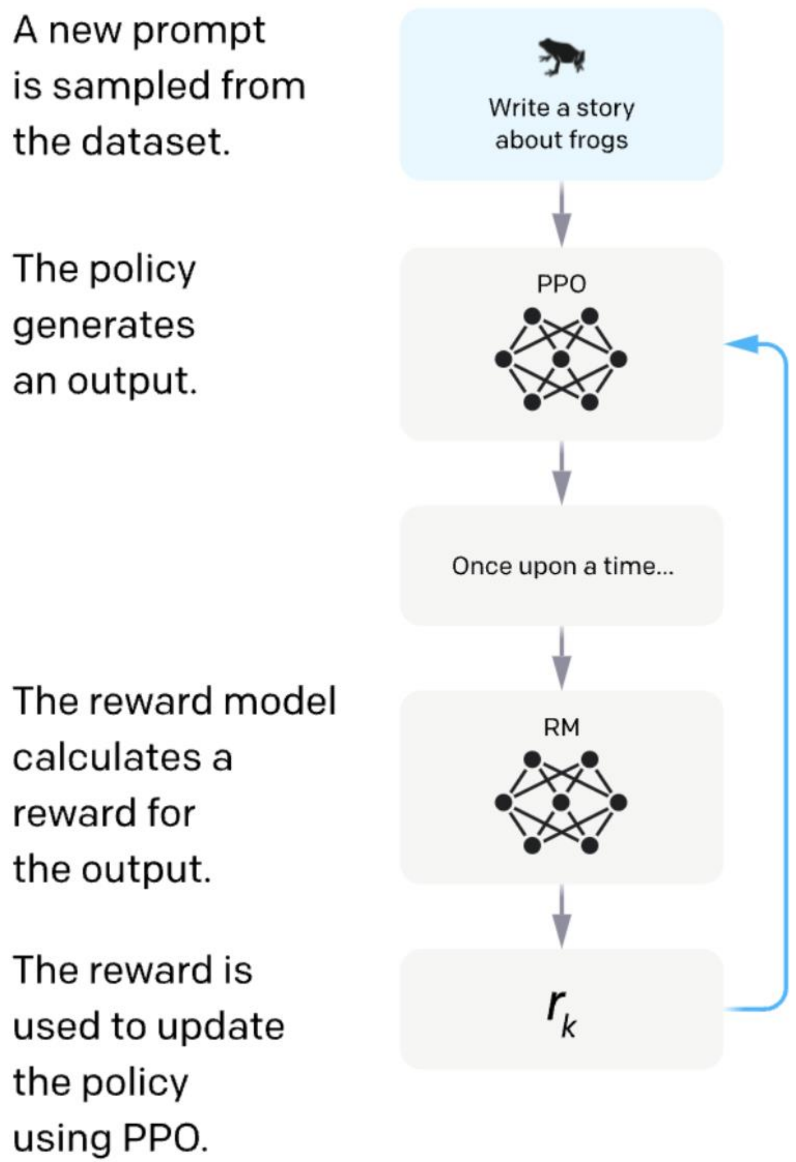
Reinforcement learning with human feedback

- Step 2 (applying RL)
  - Train the fine-tuned LLM using reward model
  - Reward model calculates a reward for the generated output



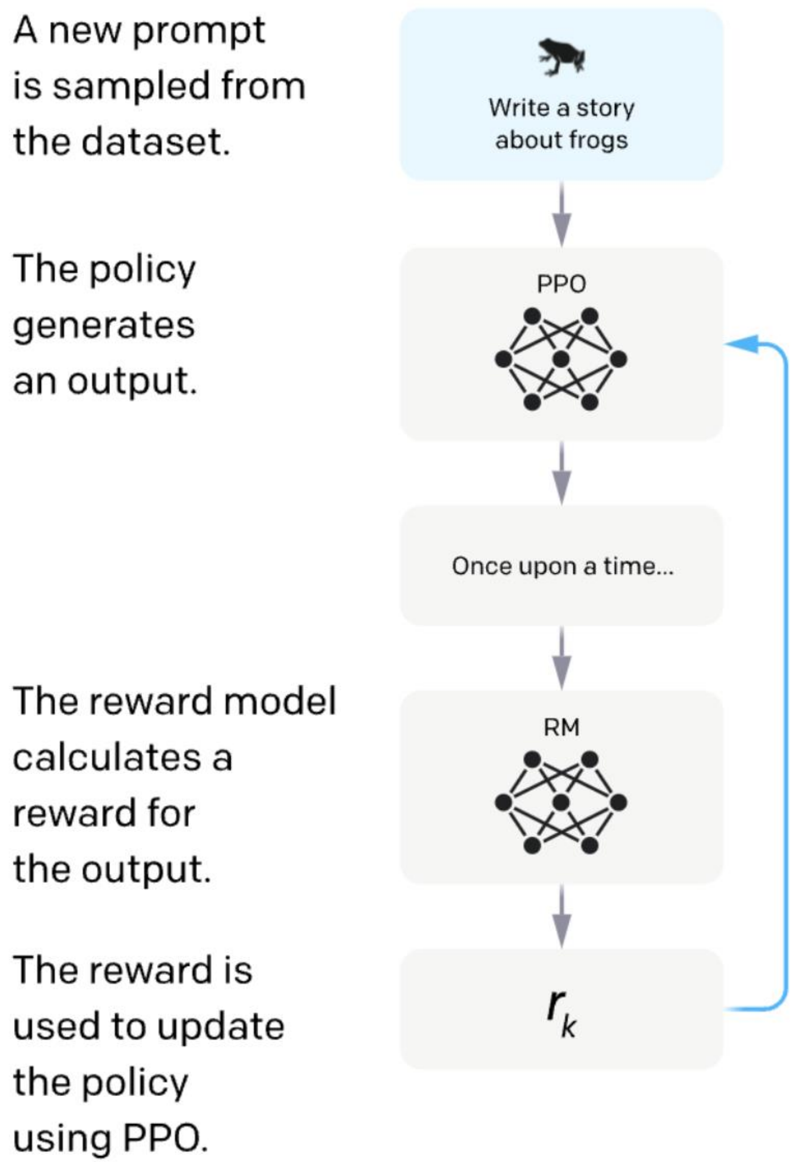
Reinforcement learning with human feedback

- Step 2 (applying RL)
  - Train the fine-tuned LLM using reward model
  - Reward model calculates a reward for the generated output
  - Using RL algorithm for training
    - Proximal Policy Optimization (PPO)



Reinforcement learning with human feedback

- Step 2 (applying RL)
  - Train the fine-tuned LLM using reward model
  - Reward model calculates a reward for the generated output
  - Using RL algorithm for training
    - Proximal Policy Optimization (PPO)
  - Get a LLM that aligns human value



Performance comparison of pre-trained and finetuned

Pre-trained model leaderboard

Model	Average	IFEval	BBH	MATH Lvl 5	GPQA	MUSR	MMLU-PRO
Qwen/Qwen2.5-72B	37.94	41.37	54.62	36.1	20.69	19.64	55.2
Qwen/Qwen2.5-32B	37.54	40.77	53.95	32.85	21.59	22.7	53.39
Qwen/Qwen2-72B	35.13	38.24	51.86	29.15	19.24	19.73	52.56
Qwen/Qwen2.5-14B	31.45	36.94	45.08	25.98	17.56	15.91	47.21
Qwen/Qwen1.5-110B	29.56	34.22	44.28	23.04	13.65	13.71	48.45
dnhkng/RYS-Phi-3-medium-4k-instruct	28.38	43.91	46.75	11.78	13.98	11.09	42.74

Fine-tuned (with RLHF) model leaderboard

Model	Average	IFEval	BBH	MATH Lvl 5	GPQA	MUSR	MMLU-PRO
MaziyarPanahi/calme-2.4-rys-78b	50.26	80.11	62.16	37.69	20.36	34.57	66.69
dnhkng/RYS-XLarge	44.75	79.96	58.77	38.97	17.9	23.72	49.2
MaziyarPanahi/calme-2.1-rys-78b	44.14	81.36	59.47	36.4	19.24	19.0	49.38
MaziyarPanahi/calme-2.2-rys-78b	43.92	79.86	59.27	37.92	20.92	16.83	48.73
MaziyarPanahi/calme-2.1-qwen2-72b	43.61	81.63	57.33	36.03	17.45	20.15	49.05
arcee-ai/Arcee-Nova	43.5	79.07	56.74	40.48	18.01	17.22	49.47

Finetuned models show better performance in most benchmarks.

https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard

Outline

Lecture 12: Large Language Models

- LLM Training
  - Pre-training
  - Instruction Fine tuning
  - RLHF
- Parameter Efficient Fine Tuning
- Data Ethics
- Prompt Engineering and Prompt Tuning

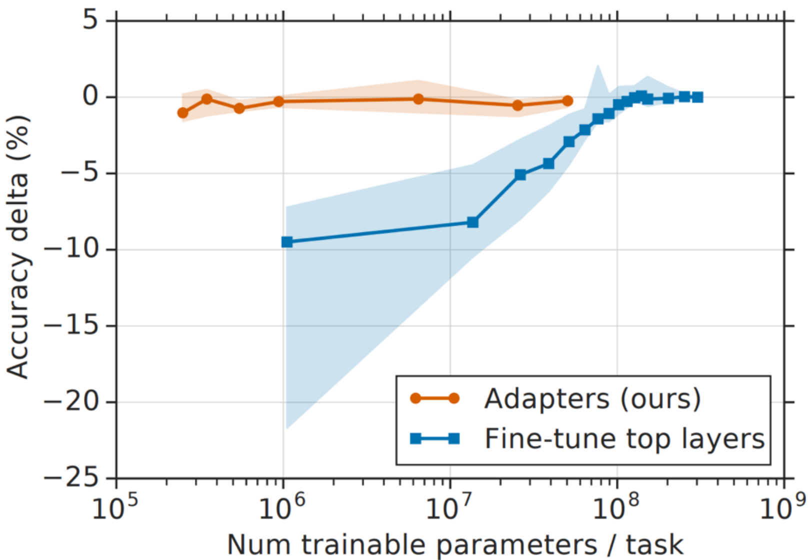
Parameter Efficient Fine Tuning (PEFT)

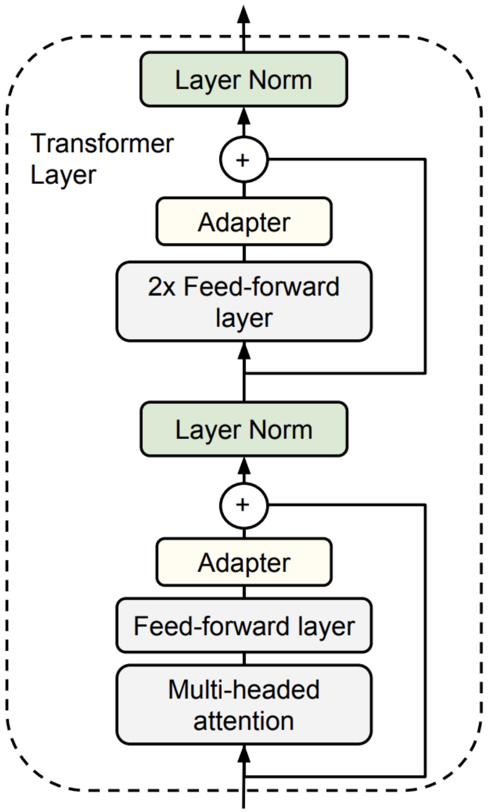
- PEFT: Fine-tune large pre-trained models for specific tasks while updating only a small subset of the model's parameters.

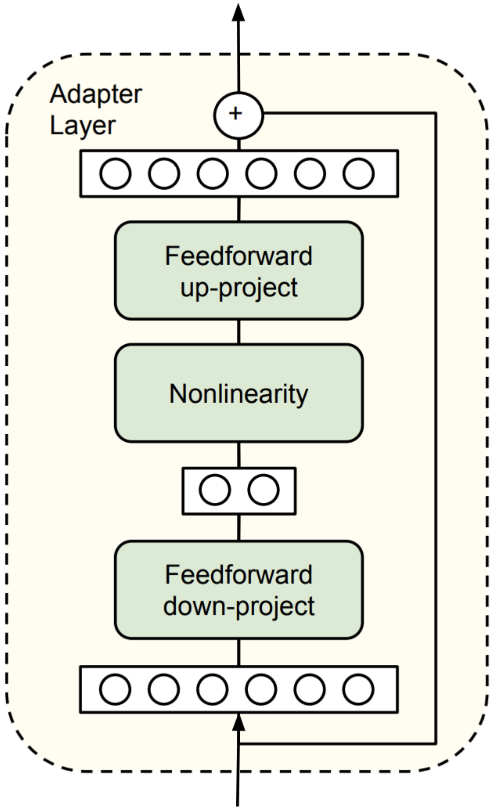
- Why PEFT
  - Produce customized LLMs on specific tasks
  - LLMs are too expensive to finetune
  - By modifying fewer parameters, preserve the model's general knowledge while adapting to specific tasks.

PEFT - Adapter

- Small neural network modules inserted into a pre-trained model.
- Inserted after the attention and/or feed-forward layers
- Freeze other parameter and only train adapter
- A bottleneck architecture module
  - a down-projection layer
  - a non-linearity layer
  - an up-projection layer

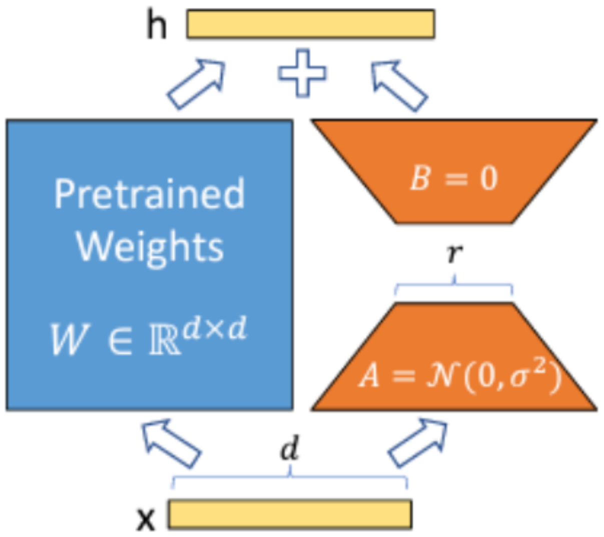






PEFT - LoRA

- Traditional pretraining fine-tuning:
  - Pretrain **W**, Finetune **W**
- LORA (Low Rank Adaptation):
  - Pretrain **W**, Finetune **AB**
- AB are low-rank matrices, rank(A) << rank(W)
- Benefit:
  - light-weight fine-tuning cost
  - Fast domain adaptation without additional serving cost



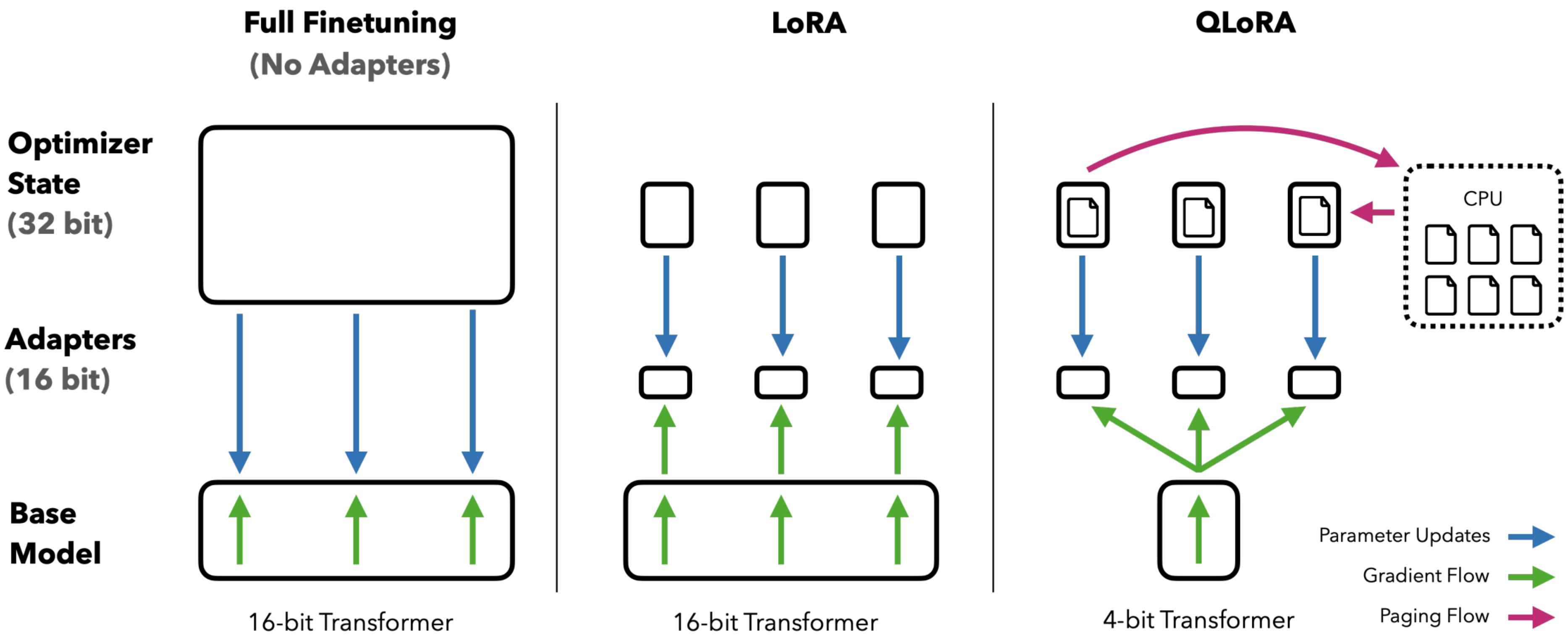
LoRA, [Edward J. Hu et al., 2021]

Batch Size Sequence Length $\|\Theta\|$	32 512 0.5M	16 256 11M	1 128 11M
Fine-Tune/LoRA	1449.4±0.8	338.0±0.6	19.8±2.7
Adapter^{L}	1482.0±1.0 (+2.2%)	354.8±0.5 (+5.0%)	23.9±2.1 (+20.7%)
Adapter^{H}	1492.2±1.0 (+3.0%)	366.3±0.5 (+8.4%)	25.8±2.2 (+30.3%)

latency

PEFT - QLoRA

- QLoRA : LoRA with quantized base model weights
  - NormalFloat (NF4) datatype for LLM weight quantization
  - CPU-offloading for optimizer state
  - Reduce memory usage significantly



Outline

Lecture 12: Large Language Models

- LLM Training
  - Pre-training
  - Instruction Fine tuning
  - RLHF
- Parameter Efficient Fine Tuning
- Data Ethics
- Prompt Engineering and Prompt Tuning

Data Ethics in LLM training

- Ethical consideration of LLM data
  - Data collection and consent
  - Bias and fairness
  - Privacy concerns
  - Misinformation and harmful content

Outline

Lecture 12: Large Language Models

- LLM Training
  - Pre-training
  - Instruction Fine tuning
  - RLHF
- Parameter Efficient Fine Tuning
- Data Ethics
- Prompt Engineering and Prompt Tuning

Prompt Engineering

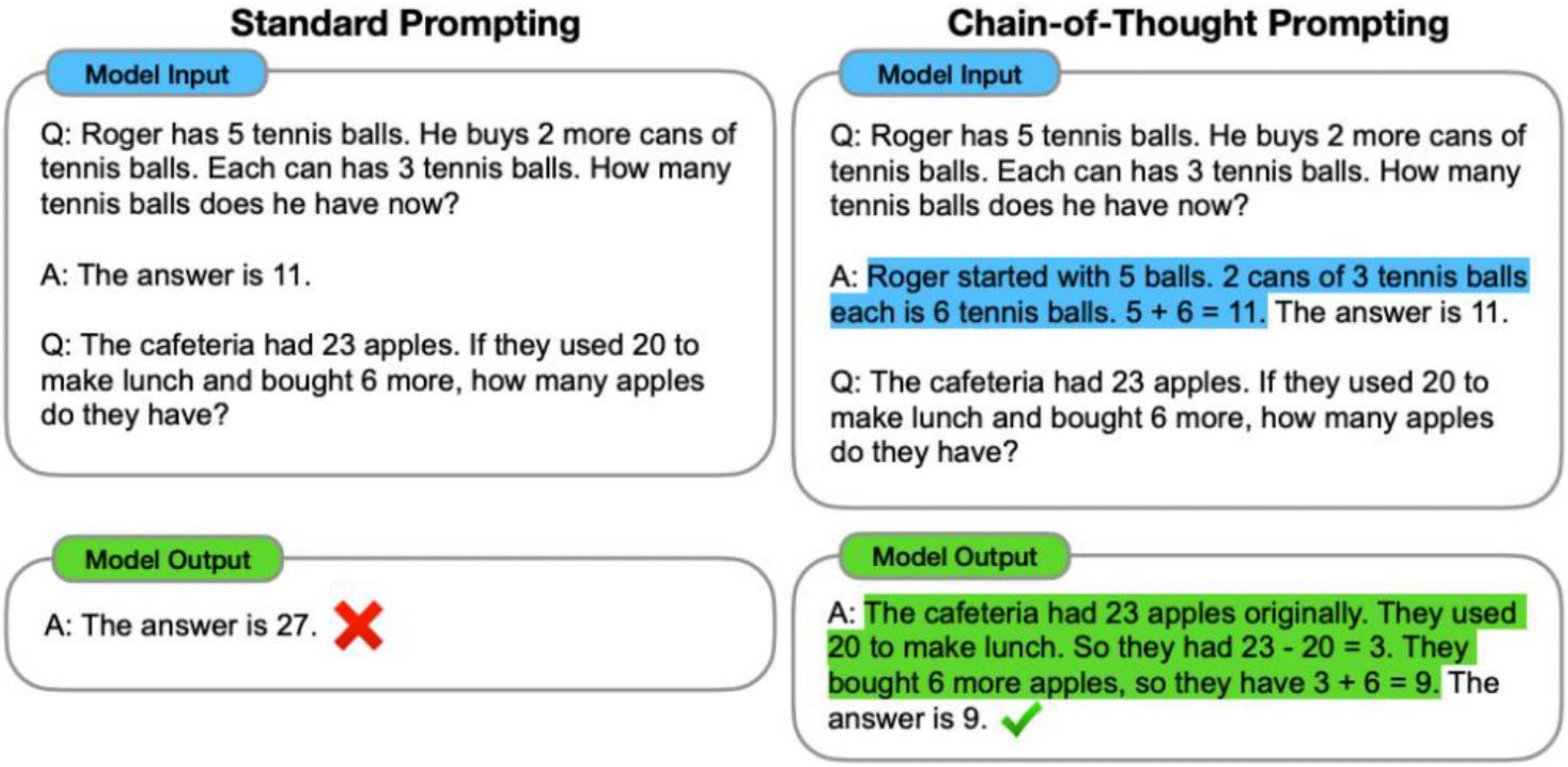
- Prompt : tell the LLM what to do in natural language

- Prompt engineering : Identify suitable prompt for a specific task

- General rule of thumb
  - write **clear** and **descriptive** instructions
  - Split complex task into simpler subtasks

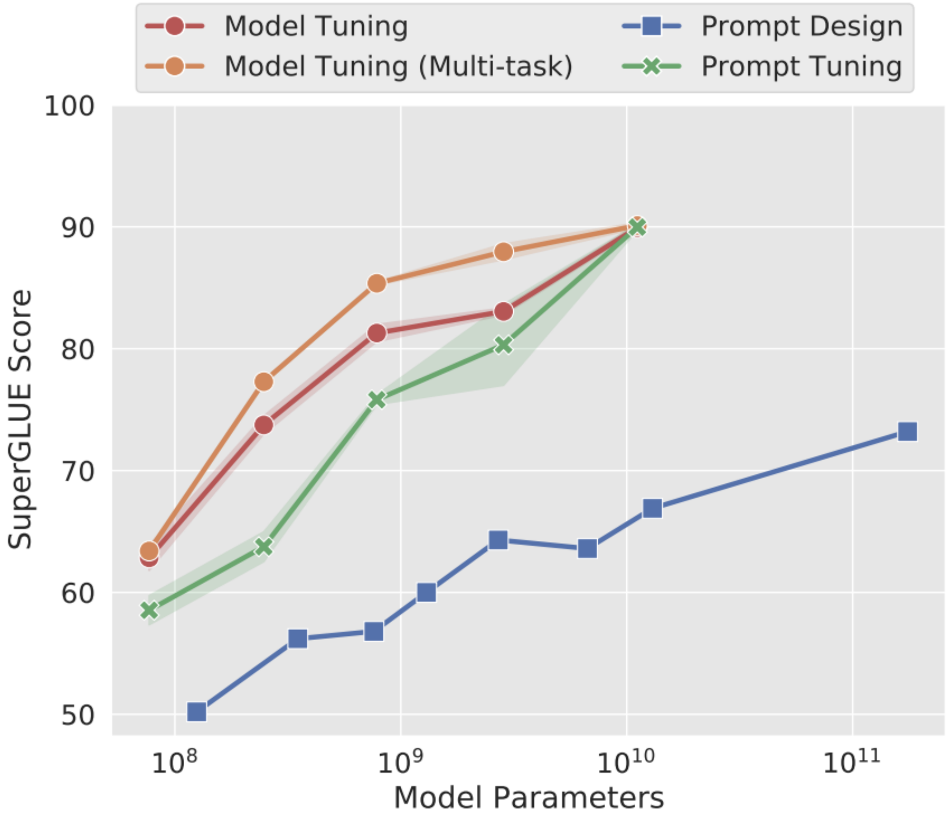
Prompt Engineering

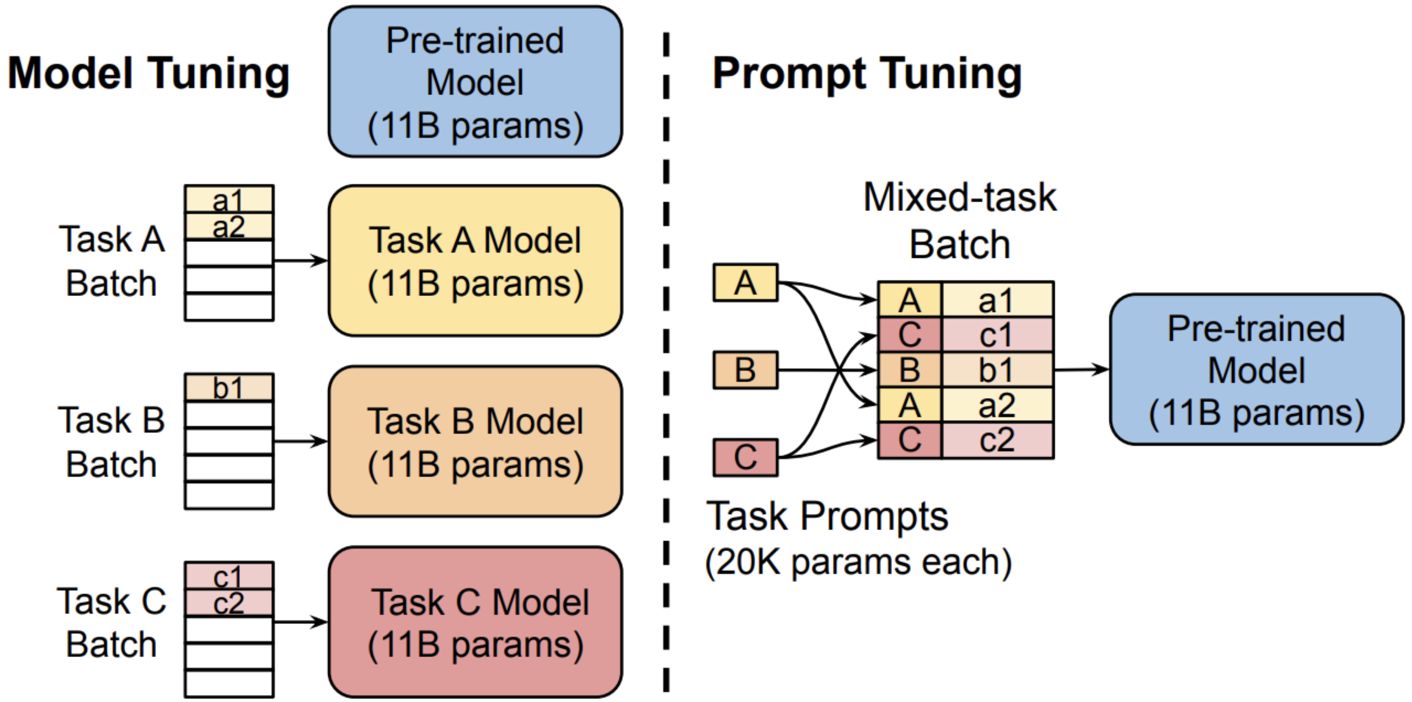
- Chain of thought prompting
    - Ask the model to work step-by-step



Prompt Tuning

- From discrete prompt to continuous trainable prompt
- learning a small set of continuous task-specific vectors (called "soft prompts") that are prepended to the input sequence.
- Extremely parameter-efficient (often <0.1% of model parameters).





In this lecture we learned

- LLM Training
  - Pre-training
  - Instruction Fine tuning
  - RLHF
- Parameter Efficient Fine Tuning
- Data Ethics
- Prompt Engineering and Prompt Tuning

Further reading

- Vaswani, Ashish, et al. "Attention is all you need." arXiv preprint arXiv:1706.03762 (2017).
- So, David R., Chen Liang, and Quoc V. Le. "The evolved transformer.” (2019)
- Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding.” (2018)
- Radford, Alec, et al. "Language models are unsupervised multitask learners." OpenAI blog 1.8 (2019): 9.
- Brown, Tom B., et al. "Language models are few-shot learners." arXiv preprint arXiv:2005.14165 (2020).
- Touvron, Hugo, et al. "Llama 2: Open foundation and fine-tuned chat models." *arXiv preprint arXiv:2307.09288* (2023).
- Wei, Jason, et al. "Chain-of-thought prompting elicits reasoning in large language models." *NeurIPS* (2022): 24824-24837.
- Li, Zhiyuan, et al. "Chain of thought empowers transformers to solve inherently serial problems." *arXiv preprint arXiv:2402.12875* (2024).

Reference

- Child, Rewon, et al. "Generating long sequences with sparse transformers.” 2019
- Wang, Sinong, et al. "Linformer: Self-attention with linear complexity." *2020.*
- Delvin, Jacob, et al. “Bert: Pre-training of deep bidirectional transformers for language understanding” NAACL 2019
- Lan, Zhenzhong, et al. "Albert: A lite bert for self-supervised learning of language representations.” 2019.
- Liu, Yinhan, et al. "Roberta: A robustly optimized bert pretraining approach.” 2019.
- Sanh, Victor, et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.” 2019.
- Jiao, Xiaoqi, et al. "Tinybert: Distilling bert for natural language understanding.” 2019.
- Tambe, Thierry, et al. "EdgeBERT: Optimizing On-Chip Inference for Multi-Task NLP." *2020.*
- Zafrir, Ofir, et al. "Q8bert: Quantized 8bit bert." *2019.*
- Radford, Alec, et al. "Improving language understanding by generative pre-training." 2018.
- Radford, Alec, et al. “Language models are unsupervised multitask learners.” 2018.
- Brown, Tom B., et al. "Language models are few-shot learners." 2020.
- Yang, Zhilin, et al. "Xlnet: Generalized autoregressive pretraining for language understanding." NeurIPS 2019.
- Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." 2020.
- Touvron, Hugo, et al. "Llama 2: Open foundation and fine-tuned chat models." arXiv preprint arXiv:2307.09288 (2023).
- Wei, Jason, et al. "Chain-of-thought prompting elicits reasoning in large language models." NeurIPS (2022): 24824-24837.

- https://www.kdnuggets.com/2019/09/bert-roberta-distilbert-xlnet-one-use.html
- https://www.theguardian.com/technology/2019/feb/14/elon-musk-backed-ai-writes-convincing-news-fiction