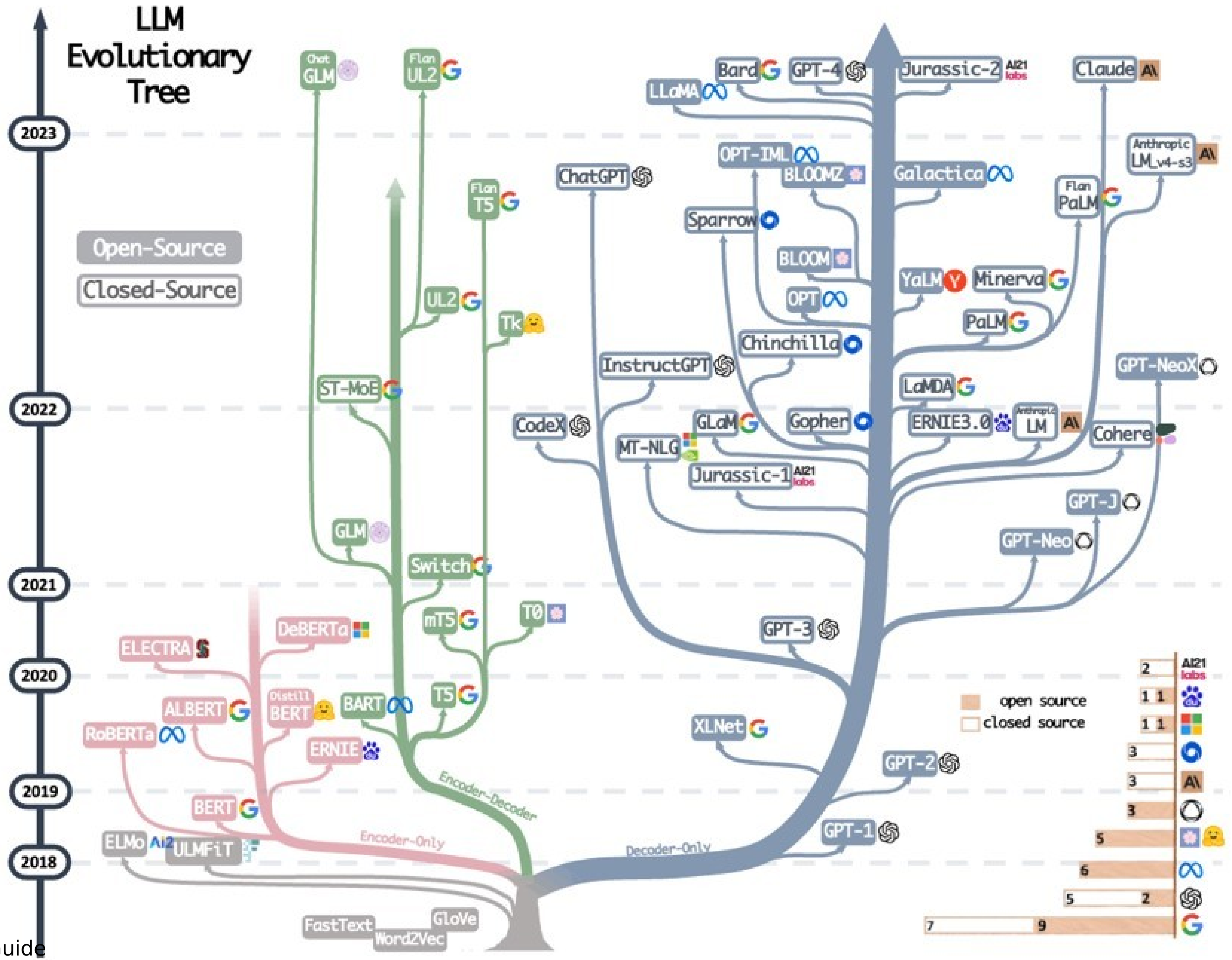
Fine-Tuning LLMs

Customizing LLMs for your needs

Ehsan Kamalinejad

LLMs

Modern language models are pretrained through self-supervised learning methods such as MLM or CLM



Source: https://github.com/Mooler0410/LLMsPracticalGuide

Pretraining LLMs

- Most of the modern LLMs are pretrained through autoregressive next token prediction.

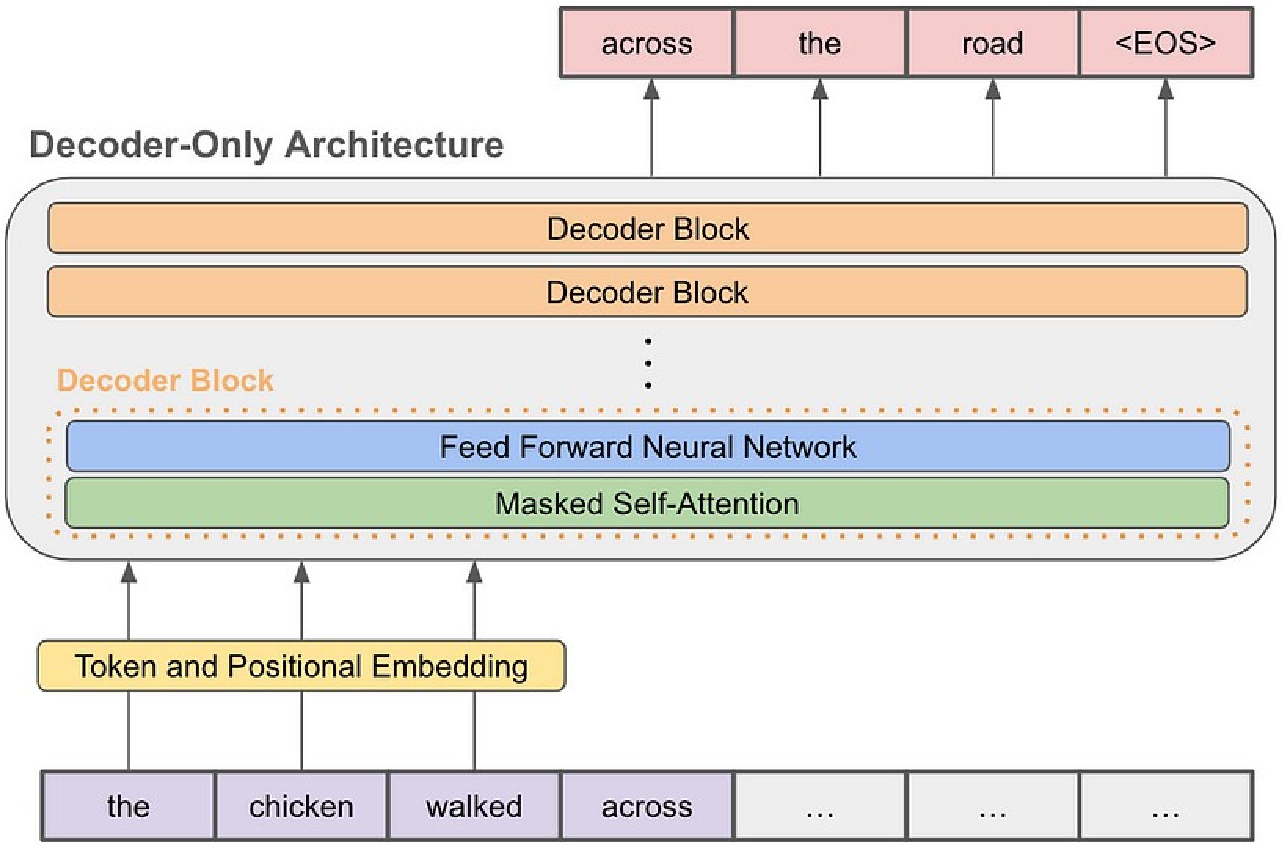


diagram: Cameron R. Wolfe Language Models: GPT and GPT-2

Pretraining LLMs

- Most of the modern LLMs are pretrained through autoregressive next token prediction.
- This allows to train on any kind of text (language, code, tabular, etc.)
- The whole internet is your playground!

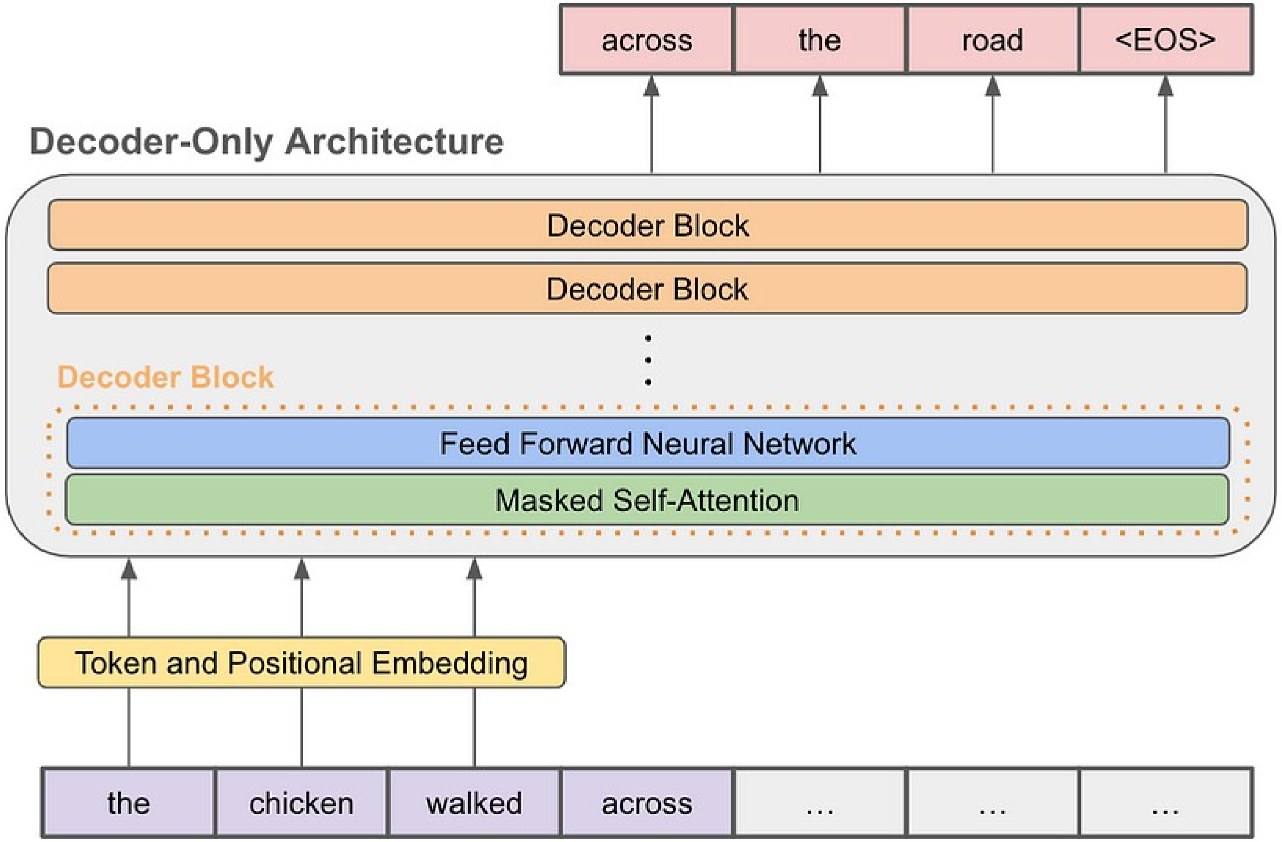
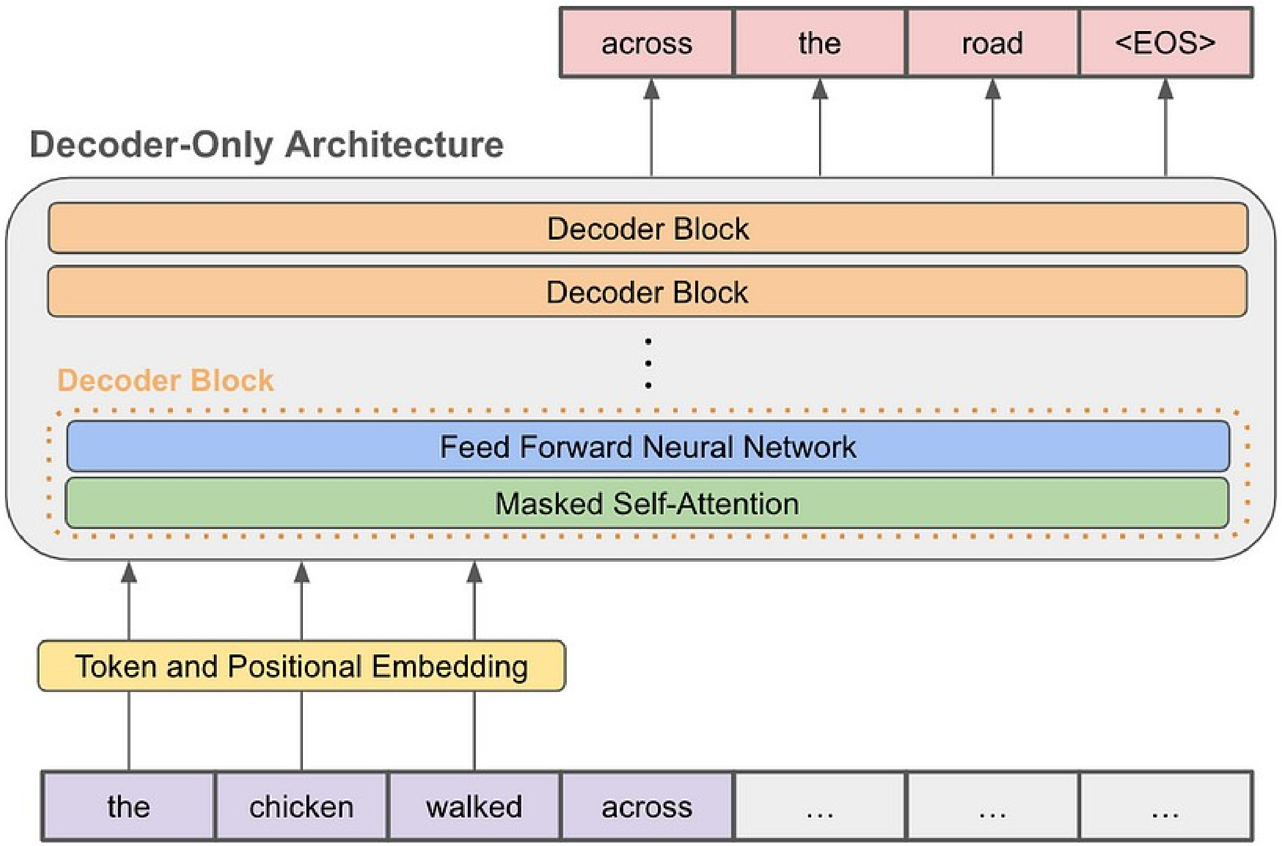


diagram: Cameron R. Wolfe Language Models: GPT and GPT-2

Pretraining LLMs

- Most of the modern LLMs are pretrained through autoregressive next token prediction.
- This allows to train on any kind of text (language, code, tabular, etc.)
- The whole internet is your playground!
- The task is hard!



2 + 2 = <?>
Must know math
The president of US in 1983 was <?> Must know history

diagram: Cameron R. Wolfe Language Models: GPT and GPT-2

Foundational Models

- Foundational LLMs are pretrained on large data
- They are versatile and adaptable base for multiple purposes
- In their raw form they do not follow instruction or answer question
- While the knowledge is stored in them, one needs to extract it

Fine-Tuning LLM

To extract the knowledge stored in LLMs, one can do fine-tuning

Two main fine-tuning methods:

- Supervised Fine-Tuning (SFT)
- Reinforcement Learning with Human Feedback (RLHF)

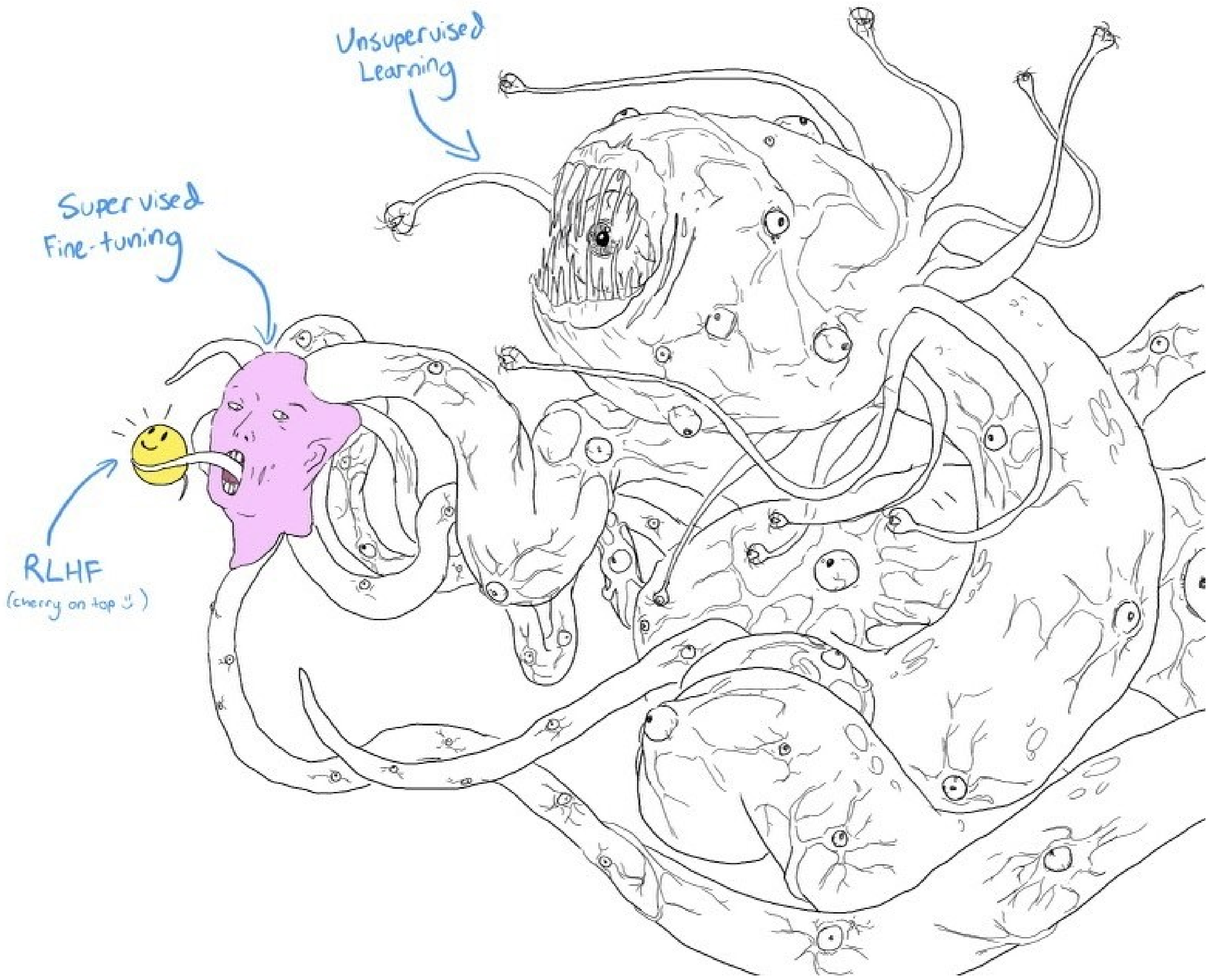
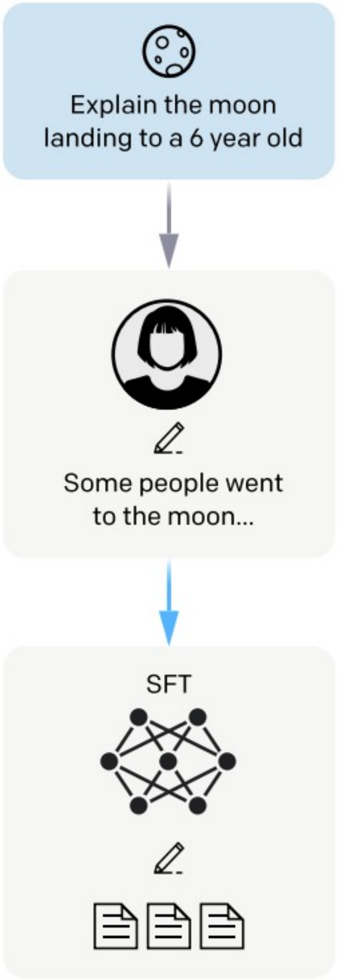


Image: Helen Toner Twitter

Diagram: arXiv:2203.02155

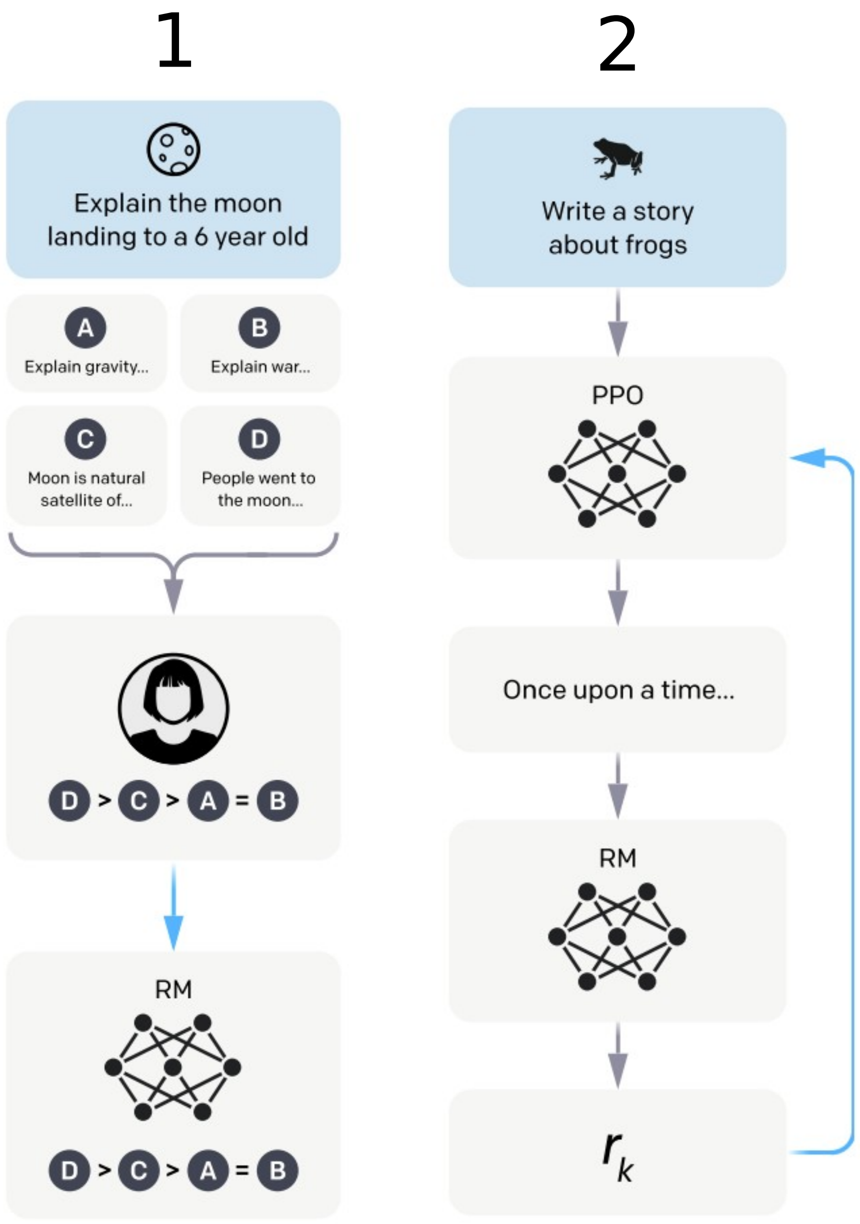
Supervised Fine-Tuning

- Create a diverse prompt dataset compatible with your task
- Sample the prompt dataset
- Give sample prompts to labelers to create the desire output
- Fine-tune the base model with the query-response pairs

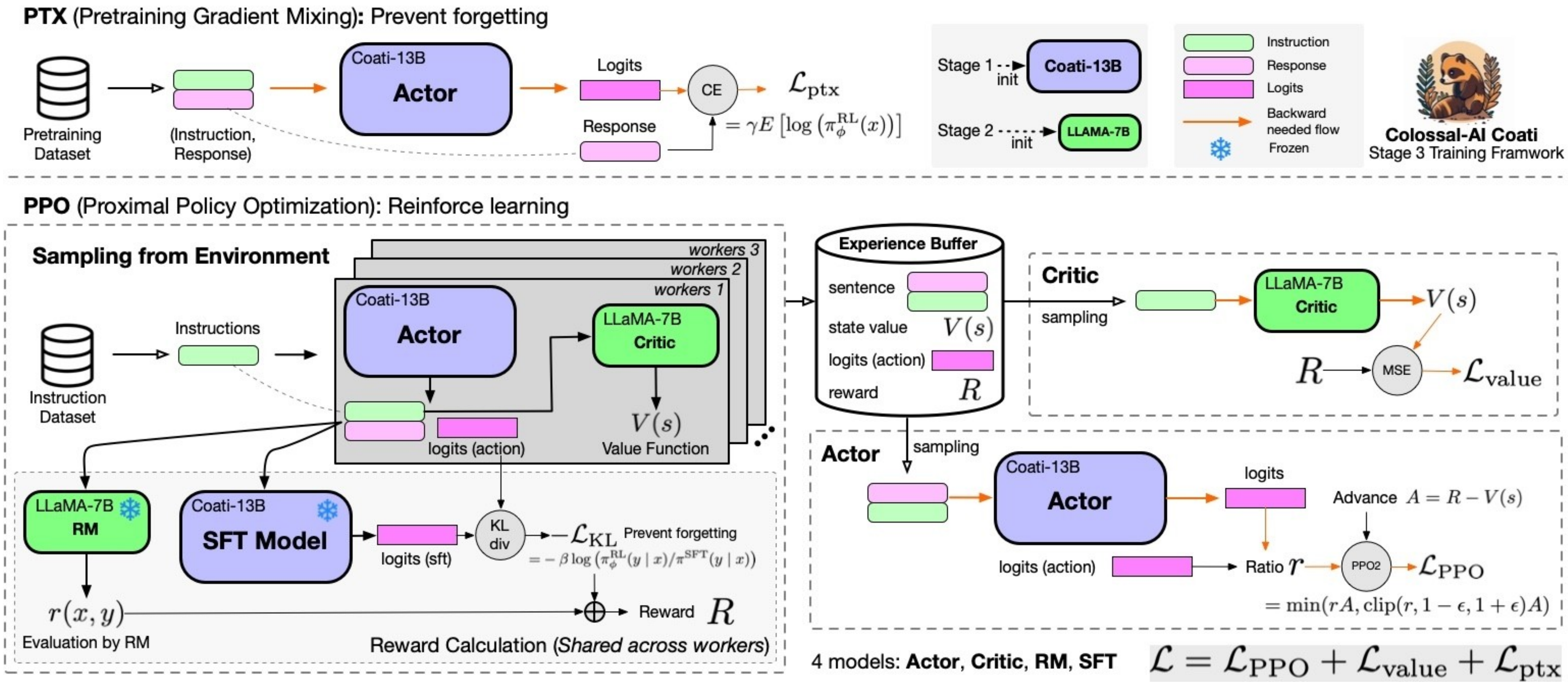


RLHF

1. Train a reward model:
   - Collect several responses to a query from the model
   - Let the labeler rank the responses according to your definition of a better response
   - Train a reward model that can estimate alignment of a response with your desire
2. Train a policy model through PPO
   - The reward model guides the base model through reinforcement learning (Proximal Policy Optimization)



RLHF



PTX (Pretraining Gradient Mixing): Prevent forgetting
Pretraining Dataset
(Instruction, Response)
Coati-13B
Actor
Logits
Response
CE
$\mathcal{L}_{\text{ptx}}$
$= \gamma E \left[\log\left(\pi_\phi^{\text{RL}}(x)\right)\right]$
Stage 1 init Coati-13B
Stage 2 init LLAMA-7B
Instruction
Response
Logits
Backward needed flow
Frozen
Colossal-AI Coati
Stage 3 Training Framwork
PPO (Proximal Policy Optimization): Reinforce learning
Sampling from Environment
workers 3
workers 2
workers 1
Instruction Dataset
Instructions
Coati-13B
Actor
LLaMA-7B
Critic
$V(s)$
logits (action)
Value Function
LLaMA-7B
RM
Coati-13B
SFT Model
logits (sft)
KL div
$\mathcal{L}_{\text{KL}}$
Prevent forgetting
$= -\beta \log\left(\pi_\phi^{\text{RL}}(y \mid x)/\pi^{\text{SFT}}(y \mid x)\right)$
$r(x, y)$
Evaluation by RM
Reward R
Reward Calculation (Shared across workers)
Experience Buffer
sentence
state value $V(s)$
logits (action)
reward R
Critic
sampling
LLaMA-7B
Critic
$V(s)$
R
MSE
$\mathcal{L}_{\text{value}}$
Actor
sampling
Coati-13B
Actor
logits
Advance $A = R - V(s)$
logits (action)
Ratio r
PPO2
$\mathcal{L}_{\text{PPO}}$
$= \min(rA, \text{clip}(r, 1-\epsilon, 1+\epsilon)A)$
4 models: Actor, Critic, RM, SFT
$\mathcal{L} = \mathcal{L}_{\text{PPO}} + \mathcal{L}_{\text{value}} + \mathcal{L}_{\text{ptx}}$

SFT vs RLHF

- It is harder to label data for SFT vs RLHF

SFT vs RLHF

- It is harder to label data for SFT vs RLHF
- It is easier to crawl data for SFT vs RLHF

SFT vs RLHF

- It is harder to label data for SFT vs RLHF
- It is easier to crawl data for SFT vs RLHF
- RLHF data is collected online while SFT data is collected offline

SFT vs RLHF

- It is harder to label data for SFT vs RLHF
- It is easier to crawl data for SFT vs RLHF
- RLHF data is collected online while SFT data is collected offline
- RLHF suffers less from “catastrophic forgetting”

SFT vs RLHF

- It is harder to label data for SFT vs RLHF
- It is easier to crawl data for SFT vs RLHF
- RLHF data is collected online while SFT data is collected offline
- RLHF suffers less from “catastrophic forgetting”
- RLHF setup is a lot more complex compared to SFT

Important Questions

- How much data is needed for SFT or RLHF on top of a foundational model?
- How much improvement one can get by doing domain/problem specialization?
- What are the scaling laws for fine-tuning?
- How effective are methods such as PEFT in fine-tuning?

Thank you.
Questions?