Reinforcement Learning in the Era of LLMs

Ishika Agarwal

CS 546: Advanced NLP

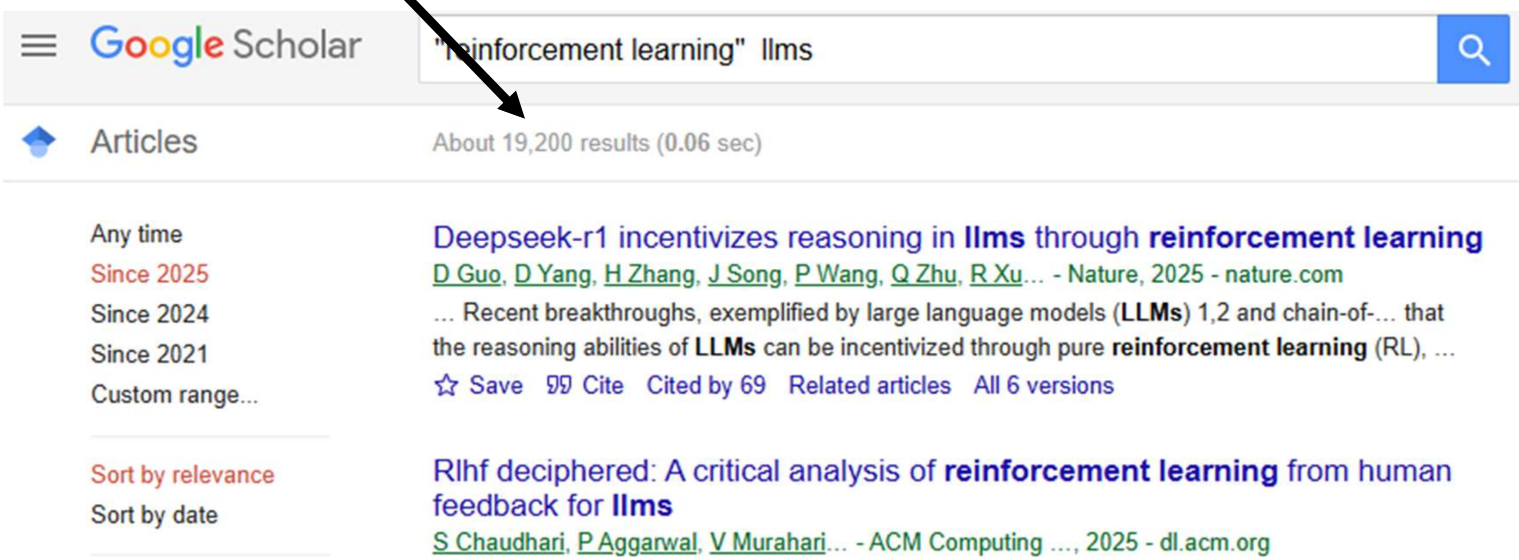
UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

CONVAI

CDS Siebel School of Computing and Data Science

Introduction

RL is a very hot topic right now!



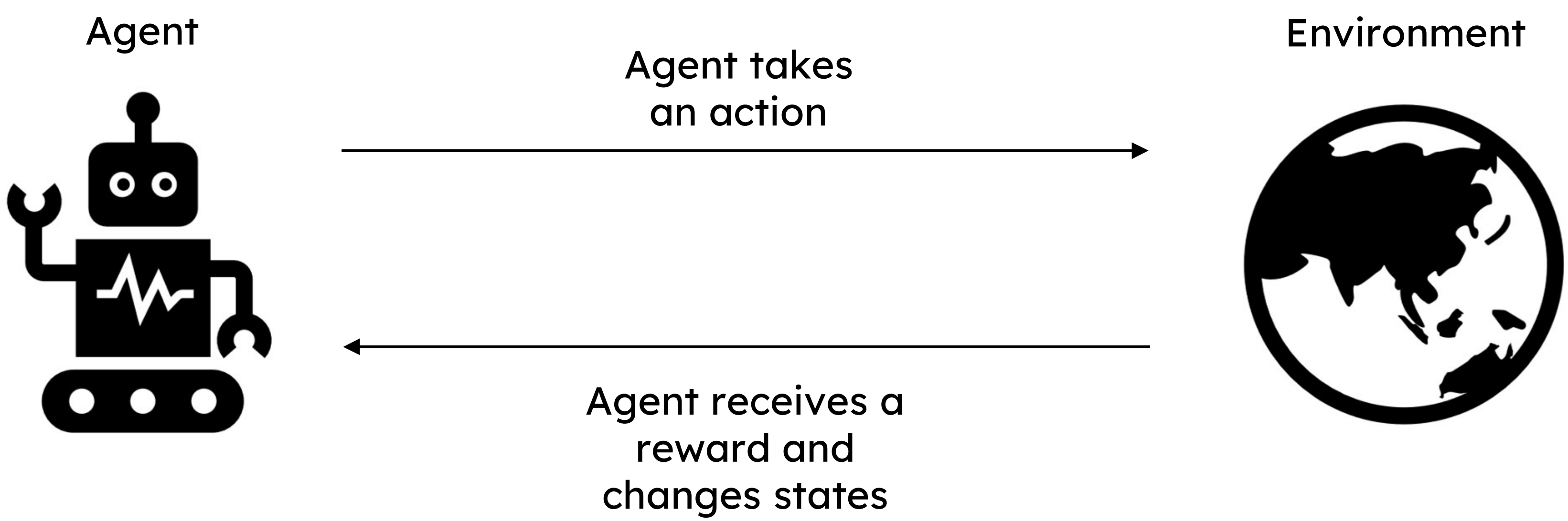
Introduction

Model development:

1. Pre-trained model (learns how to speak)
2. Instruction-tuned model (learns how to be useful)
   a. SFT
3. Generalizable/task specific models
   a. RL

What is RL?

- Learning from positive/negative reinforcement



Agenda

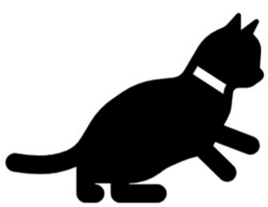
I. Basics
II. Common RL Algorithms
III. RL Applications

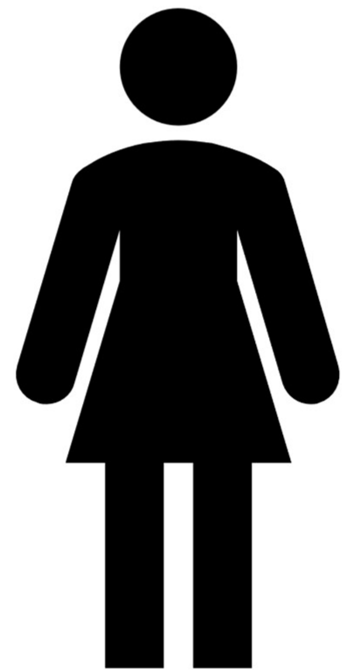
Agenda

I. Basics
   I. Terminology
   II. Bellman equations
   III. Policy gradient
II. Common RL Algorithms
III. RL Applications

Terminology

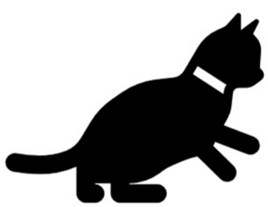
Suppose you are training your cat…

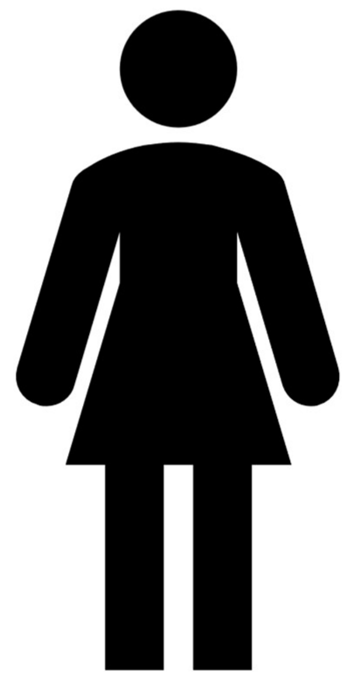




Terminology

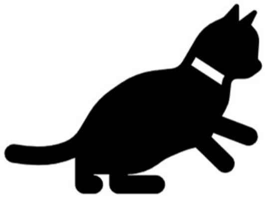
Your cat is an **agent**!

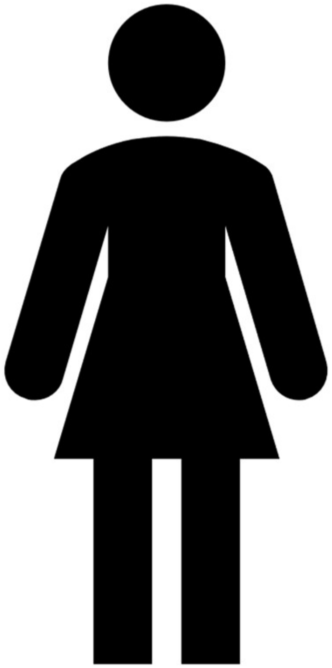




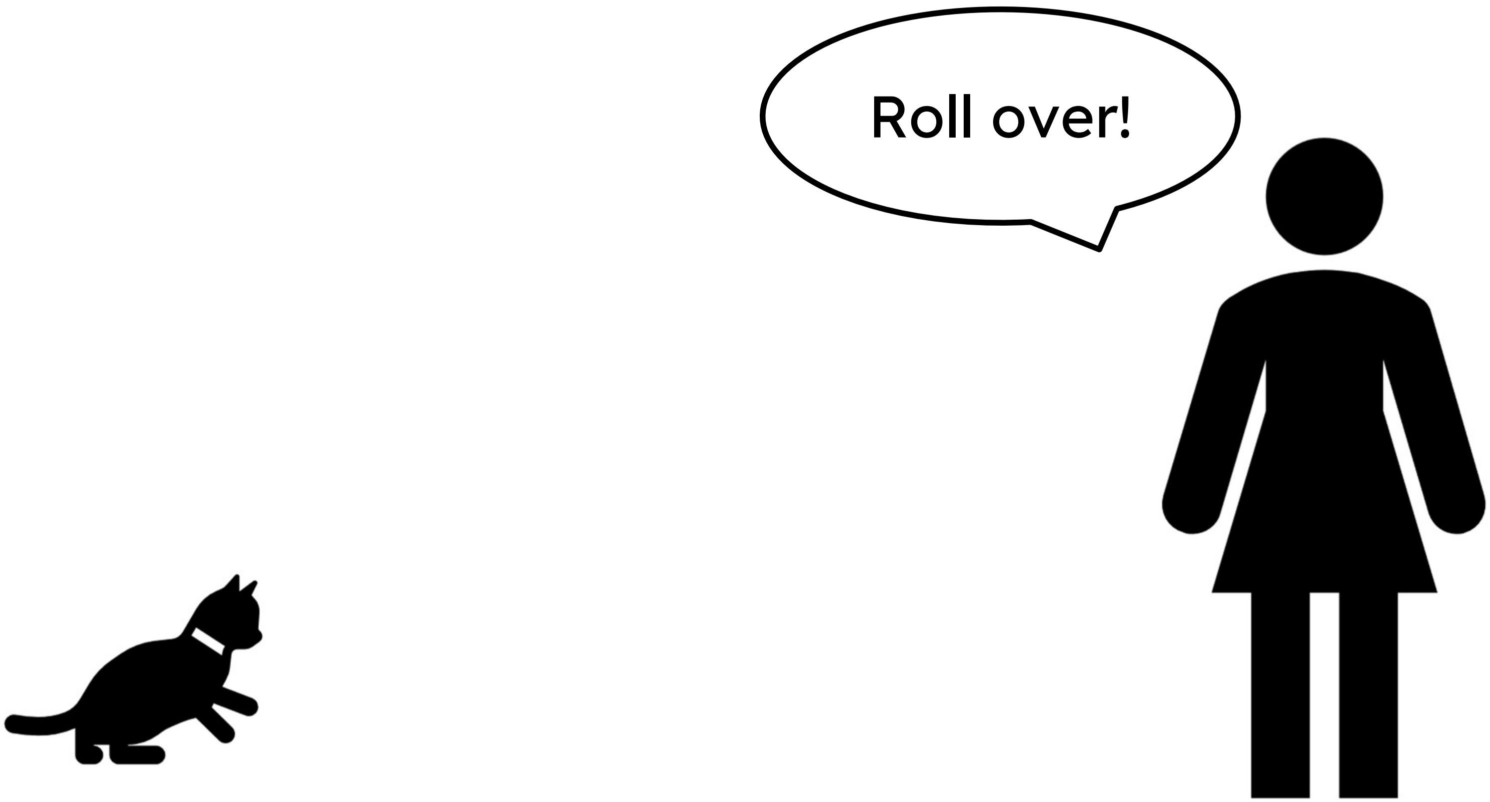
Terminology

Your cat is an agent! And its **state** is *sitting*.



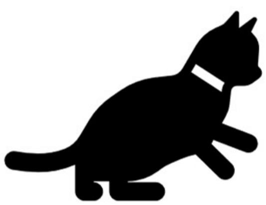


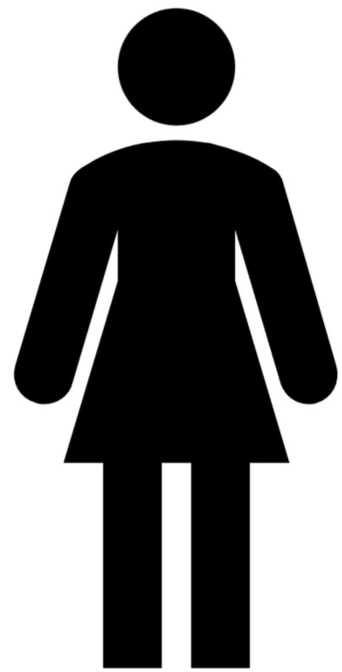
Terminology



Terminology

Whatever your cat does next is an **action**.



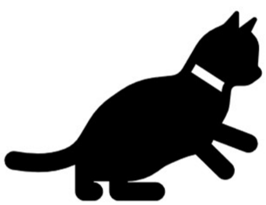


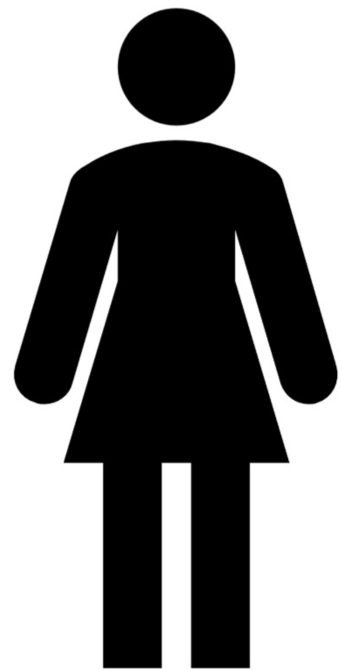
Terminology

Whatever your cat does next is an action. It will sample its next action from its **policy.**

Terminology

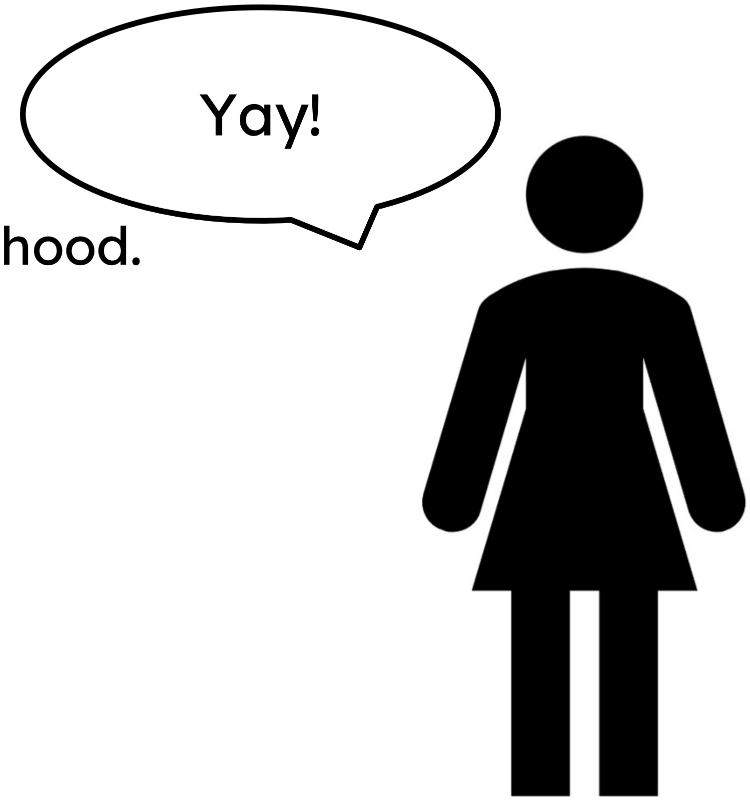
Your cat makes the **action** of rolling over.

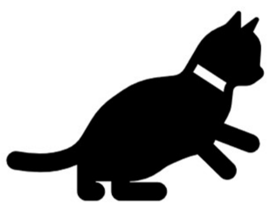




Terminology

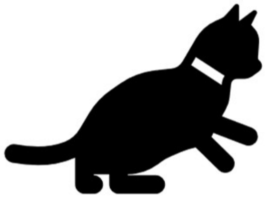
Cat receives a positive **reward**.
It realizes that next time, it
should roll over with higher likelihood.

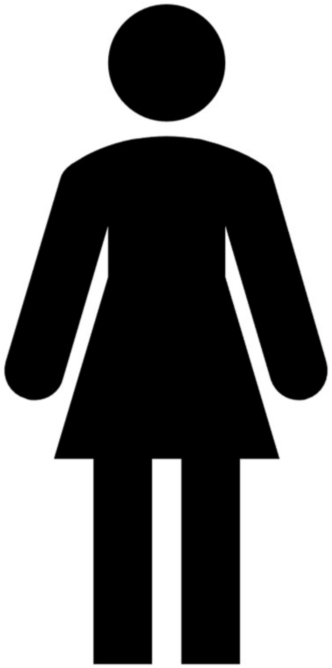




Terminology

Your cat will take a series of actions: a **trajectory**.





Terminology – recap

Agent	The learner or decision maker
Environment	The (contained) area the agent makes decisions in
State	The representation of the current situation
Action	The choice the agent makes
Reward	The feedback the agent receives
Policy	The current mapping from states to actions
Trajectory	A set of actions

Terminology – draw parallels to language

CONVAI

Agent	The learner or decision maker	Language model
Environment	The (contained) area the agent makes decisions in	Language
State	The representation of the current situation	The current context of the model ($s_{<t}$)
Action	The choice the agent makes	The token s_t generated at index t
Reward	The feedback the agent receives	Outcome or process reward
Policy	The current mapping from states to actions	Language model
Trajectory	A set of actions	Also, $s_{<t}$

Bellman Equations

Two things that make RL work (and difficult…):

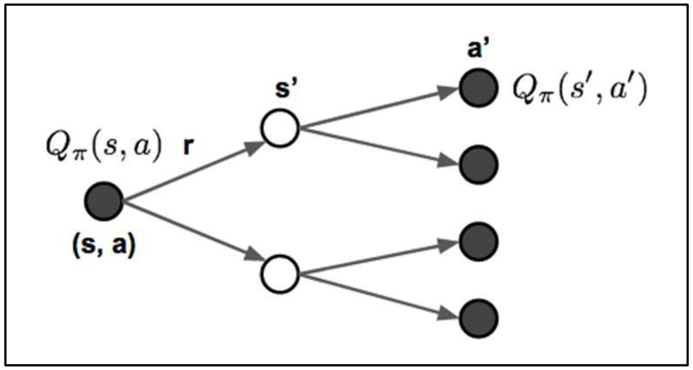
1. Q function

2. Value function

Bellman Equations

Two things that make RL work (and difficult…):

1. Q function
   - What is the expected reward of taking action a in state s?
   - $Q_\pi(s,a)$: the current reward plus the expected reward of trajectory
   - $Q_\pi(s,a) = r(s,a) + \gamma \sum_{s'} P(s'|a,s) \sum_{a'} \pi(a'|s') Q_\pi(s',a')$



https://lilianweng.github.io/posts/2018-02-19-rl-overview/

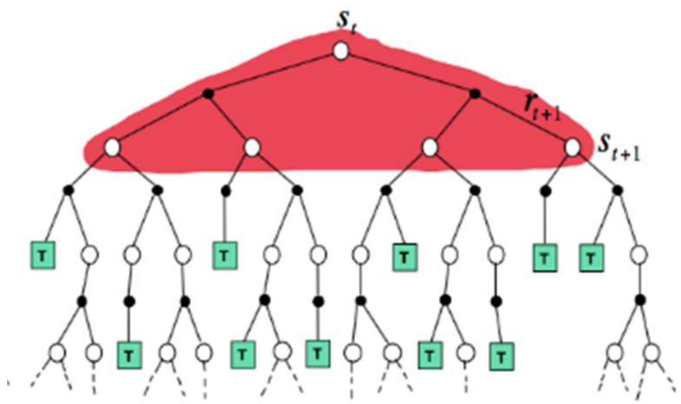
2. Value function

Bellman Equations

Two things that make RL work (and difficult…):

1. Q function
   - What is the expected reward of taking action a in state s?
   - $Q_\pi(s,a)$: the current reward plus the expected reward of trajectory
   - $Q_\pi(s,a) = r(s,a) + \gamma \sum_{s'} P(s'|a,s) \sum_{a'} \pi(a'|s') Q_\pi(s',a')$

2. Value function
   - What is the expected reward of being in state s?
   - $V_\pi(s)$: the average reward of the trajectory
   - $V_\pi(s) = \sum_a \pi(a|s)[r(s,a) + \gamma \sum_{s'} P(s'|a,s)\, V_\pi(s')$



https://davidstarsilver.wordpress.com/wp-content/uploads/2025/04/lecture-4-model-free-prediction-.pdf

Policy Gradient

- In RL we want to maximize the expected return:

$$J(\pi) = E_{\tau \sim \pi}[R(\tau)]$$

- The policy gradient comes out to be:

$$\nabla_\theta J(\pi_\theta) = E_{\tau \sim \pi_\theta}[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)\, R(\tau)]$$

  - $\nabla_\theta \log \pi_\theta(a_t|s_t)$ → where do I go to make this action more likely?
  - $R(\tau)$ → how good is this trajectory?

Policy Gradient

- In RL we want to maximize the expected return:

$$J(\pi) = E_{\tau\sim\pi}[R(\tau)]$$

- The policy gradient comes out to be:

$$\nabla_\theta J(\pi_\theta) = E_{\tau\sim\pi_\theta}[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)\, R(\tau)]$$

  - $\nabla_\theta \log \pi_\theta(a_t|s_t)$ → where do I go to make this action more likely?
  - $R(\tau)$ → how good is this trajectory?
  - Intuition check!! If there are good actions with low probability, the policy gradient will be…?
    - Bigger or smaller?

Agenda

I. Basics
  I. Terminology
  II. Bellman equations
  III. Policy gradient
II. Common RL Algorithms
  I. PPO
  II. DPO
  III. GRPO
III. RL Applications

Vanilla Policy Optimization

- Receive a reward at the end of the entire trajectory

$$L_{VPG} = -E_{\tau \sim \pi_\theta}[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)\, R(\tau)]$$

- Once you and your cat are done playing for an hour, the cat receives a treat/no treat

Vanilla Policy Optimization

- Receive a reward at the end of the entire trajectory

$$L_{VPG} = -E_{\tau \sim \pi_\theta}[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)\, R(\tau)]$$

- Once you and your cat are done playing for an hour, the cat receives a treat/no treat
- Problems:
  - Rewards are very sparse!
  - Can we give rewards for each action instead…?

Proximal Policy Optimization (PPO)

- Approximate the reward at each step + future steps:

$$L_{PPO} = -E_{\tau\sim\pi_\theta}[\sum_{t=0}^{T} \quad r_t(\theta)A(s_t, a_t), \qquad]$$

- $A(s_t, a_t)$: advantage function
  - Tells you how much better a_t is for s_t compared to other actions
- $r_t(\theta)$ is NOT reward, it is a ratio: $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$
  - Measures how drastically the new policy is changing compared to the old one

Proximal Policy Optimization (PPO)

- Approximate the reward at each step + future steps:

$$L_{PPO} = -E_{\tau\sim\pi_\theta}[\sum_{t=0}^{T} \min(r_t(\theta)A(s_t,a_t), CLIP(r_t(\theta), 1-\epsilon, 1+\epsilon)A(s_t,a_t))]$$

- $A(s_t, a_t)$: advantage function
  - Tells you how much better a_t is for s_t compared to other actions
- $r_t(\theta)$ is NOT reward, it is a ratio: $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$
  - Measures how drastically the new policy is changing compared to the old one
  - Clipping ensures the model doesn't stray too far from the previous model

Proximal Policy Optimization (PPO)

- Approximate the reward at each step + future steps:

$$L_{PPO} = -E_{\tau \sim \pi_\theta}[\sum_{t=0}^{T} \min(r_t(\theta)A(s_t, a_t), CLIP(r_t(\theta), 1-\epsilon, 1+\epsilon)A(s_t, a_t))]$$

- $A(s_t, a_t)$: advantage function
  - Tells you how much better a_t is for s_t compared to other actions
- $r_t(\theta)$ is NOT reward, it is a ratio: $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$
  - Measures how drastically the new policy is changing compared to the old one
  - Clipping ensures the model doesn't stray too far from the previous model

Advantage function…?

$$L_{PPO} = -E_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{T} r_t(\theta) A(s_t, a_t)\right]$$

- $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$
  - The expected return of taking action a_t minus the expected return of being in state s_t
  - Intuitively:
    - How much better is action a_t at s_t compared to other actions
    - For a given trajectory, what if I took action a_t at s_t instead? Would it be better?
  - Intuition check!! What does it mean when the advantage is…
    - Negative?
    - Zero?

Advantage function…?

$$L_{PPO} = -E_{\tau \sim \pi_\theta}[\sum_{t=0}^{T} \quad r_t(\theta)A(s_t, a_t) \quad]$$

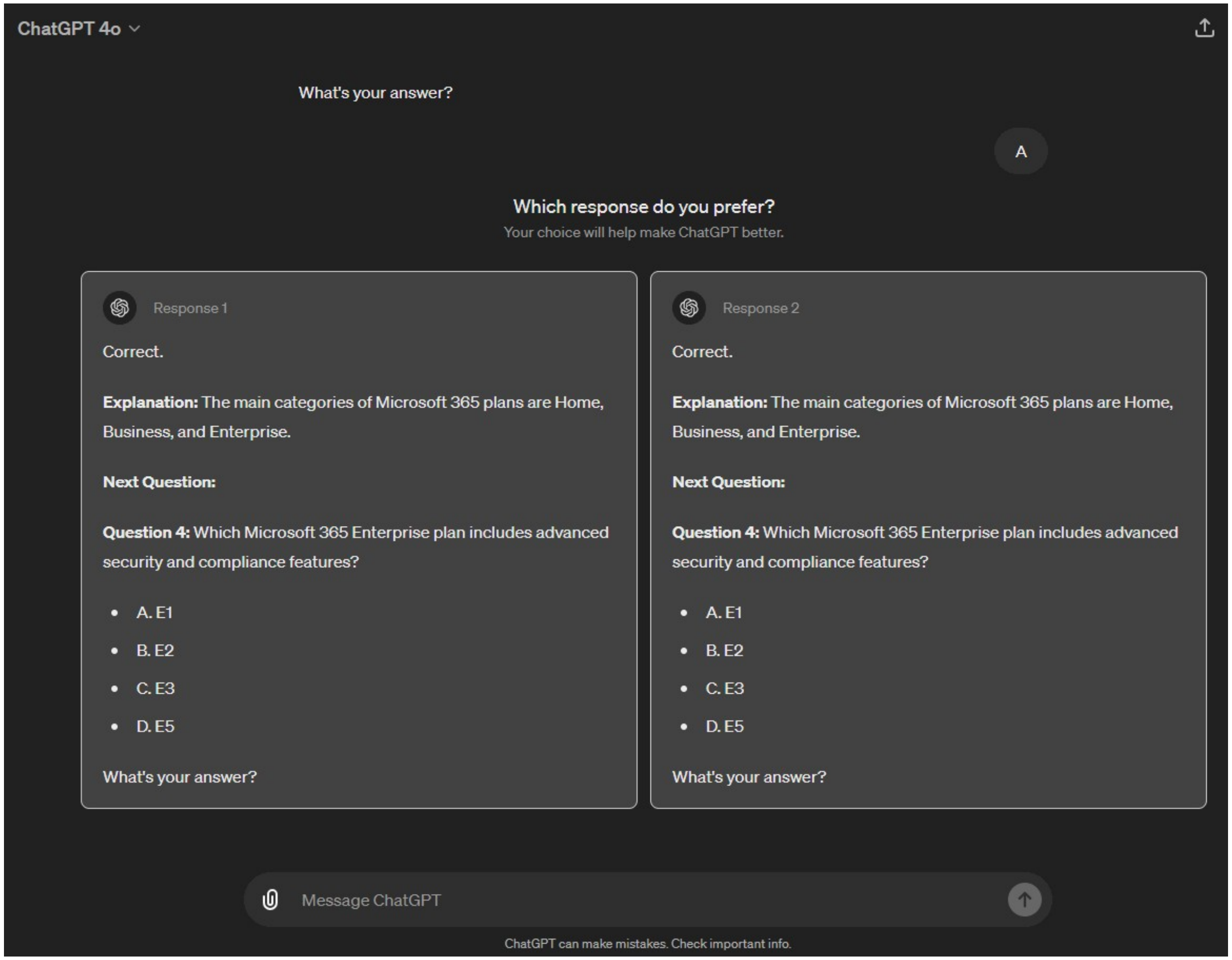
- $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$
  - The expected return of taking action a_t minus the expected return of being in state s_t
  - $Q(s_t, a_t)$ is given ("yay!" when the cat rolls over)
  - $V(s_t)$ is estimated with a neural network
    - This could be expensive if the state space gets really large

Advantage function…?

$$L_{PPO} = -E_{\tau \sim \pi_\theta}[\sum_{t=0}^{T} r_t(\theta) A(s_t, a_t)]$$

- $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$
  - The expected return of taking action a_t minus the expected return of being in state s_t
  - $Q(s_t, a_t)$ is given (“yay!” when the cat rolls over)
  - $V(s_t)$ is estimated with a neural network
    - This could be expensive if the state space gets really large
- Can the model autonomously figure out which actions are better than others?
  - Preference pairs
  - Group optimization

Direct Preference Optimization (DPO)



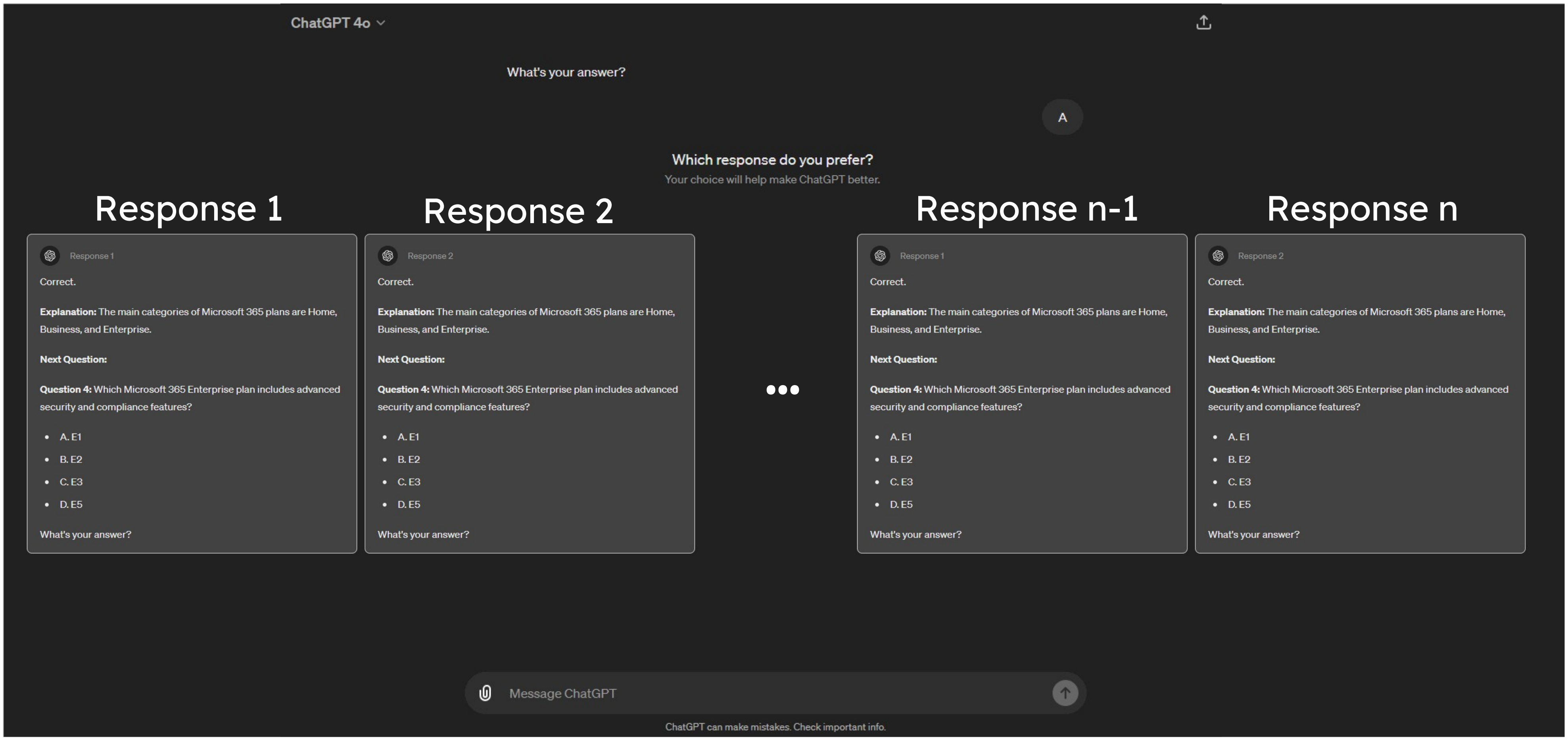
Direct Preference Optimization (DPO)

- Alternative to RLHF:

$$L_{DPO} = -E_{x,y^+,y^-}[\log \sigma(\beta \frac{\pi_\theta(y^+|x)}{\pi_{\theta_{ref}}(y^+|x)} - \beta \frac{\pi_\theta(y^-|x)}{\pi_{\theta_{ref}}(y^-|x)})]$$

- x, y^+, y^-: input, desired response, undesired response
- β controls how far π_θ can move from $\pi_{\theta_{\mathrm{ref}}}$
- Move in the direction of favoring y^+ and away from favoring y^-
    - Your cat learns to $y^+ =$ “meow at you” ($y^- =$ “jump at you”) when you come home

Group Relative Policy Optimization (GRPO)



Group Relative Policy Optimization (GRPO)

- PPO-style loss over a group of responses

$$L_{GRPO} = \frac{1}{G}\sum_{i=1}^{G} L_{PPO}$$

- $\tilde{A}(s_t, a_t) = R_t - \bar{R}$
  - Samples G completions
  - Scores each of them
  - Advantage of completion i is the difference in the reward of i and the mean reward

Group Relative Policy Optimization (GRPO)

- PPO-style loss over a group of responses

$$L_{GRPO} = \frac{1}{G}\sum_{i=1}^{G}\sum_{t=0}^{T}\min\left(r_t(\theta)\tilde{A}(s_t, a_t), CLIP(r_t(\theta), 1-\epsilon, 1+\epsilon)\tilde{A}(s_t, a_t)\right)$$

- $\tilde{A}(s_t, a_t) = R_t - \bar{R}$
  - Samples G completions
  - Scores each of them
  - Advantage of completion i is the difference in the reward of i and the mean reward

Recap

PPO:

- Improves the RL policy without deviating too much (ensuring no performance degradation)

DPO:

- Eliminates the need for an explicit value function (expected rewards)
- Advantage based on preference pairs

GRPO:

- Also eliminates the need for an explicit value function
- Advantage based on relative rewards within a group

Agenda

I. Basics
   I. Terminology
   II. Bellman equations
   III. Policy gradient
II. Common RL Algorithms
   I. PPO
   II. DPO
   III. GRPO
III. RL Applications
   I. PPO: RLHF
   II. DPO: Reasoning
   III. GRPO: Self-Correction

PPO

Training language models to follow instructions with human feedback

Long Ouyang* Jeff Wu* Xu Jiang* Diogo Almeida* Carroll L. Wainwright*

Pamela Mishkin* Chong Zhang Sandhini Agarwal Katarina Slama Alex Ray

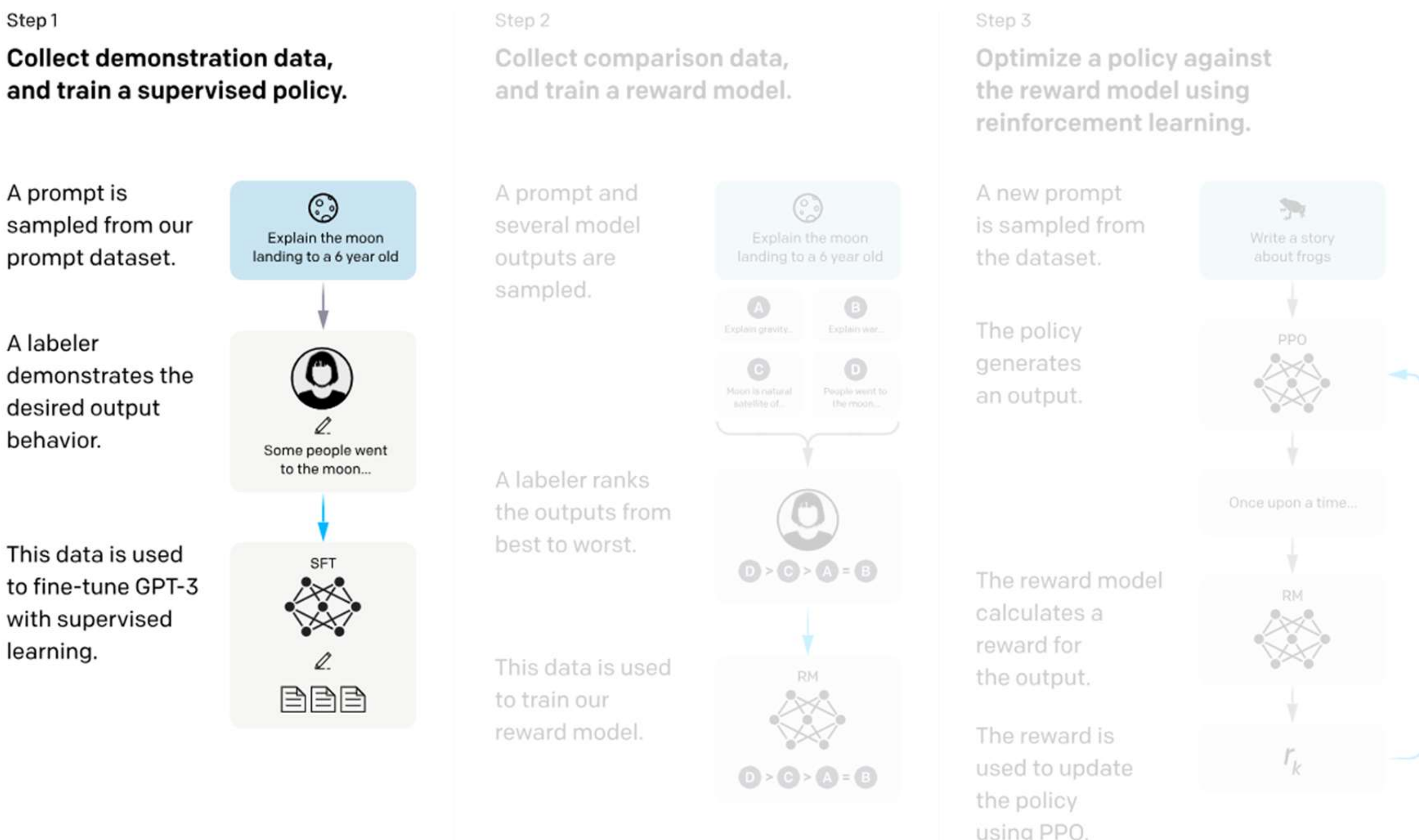
John Schulman Jacob Hilton Fraser Kelton Luke Miller Maddie Simens

Amanda Askell† Peter Welinder Paul Christiano*†

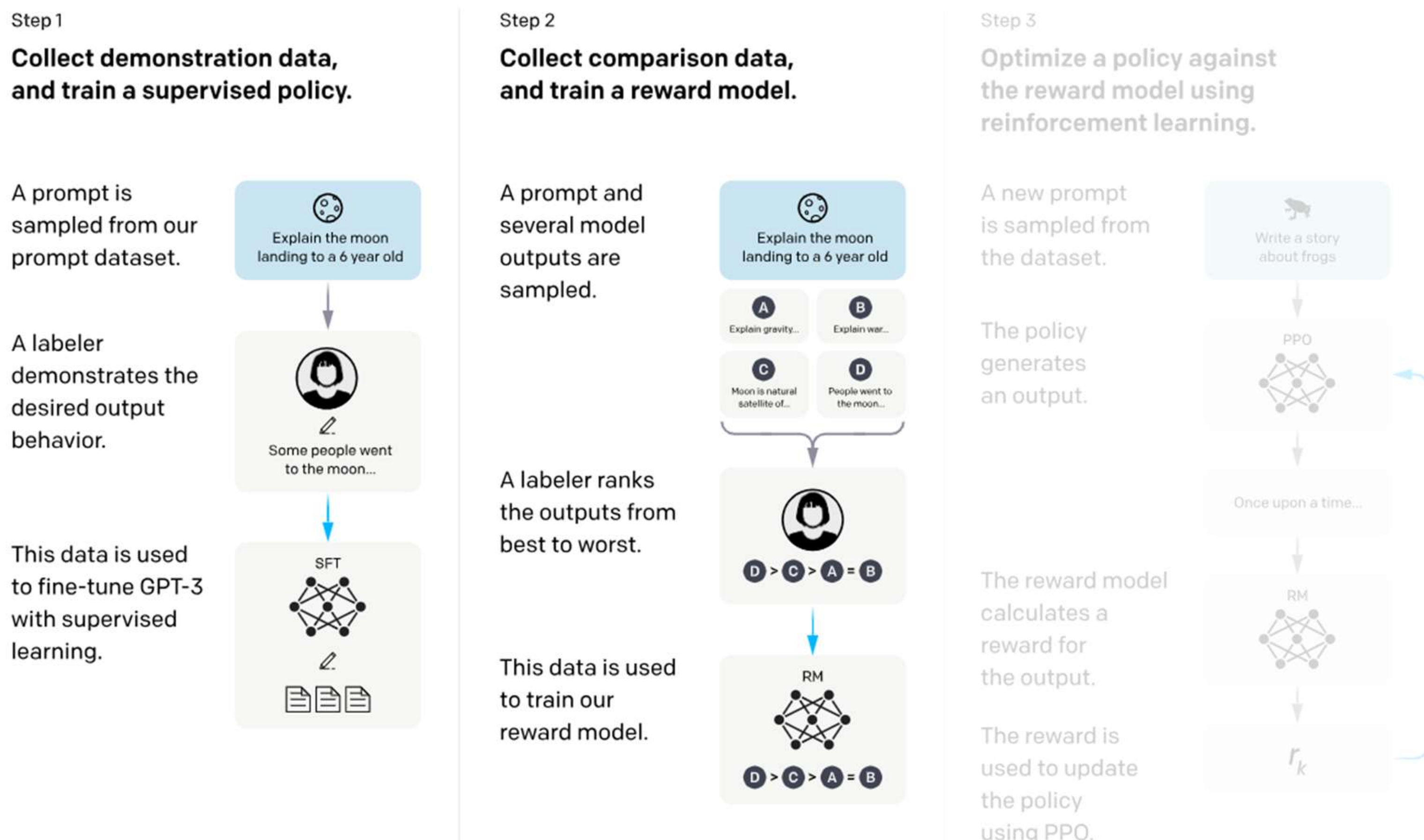
Jan Leike* Ryan Lowe*

OpenAI

RLHF

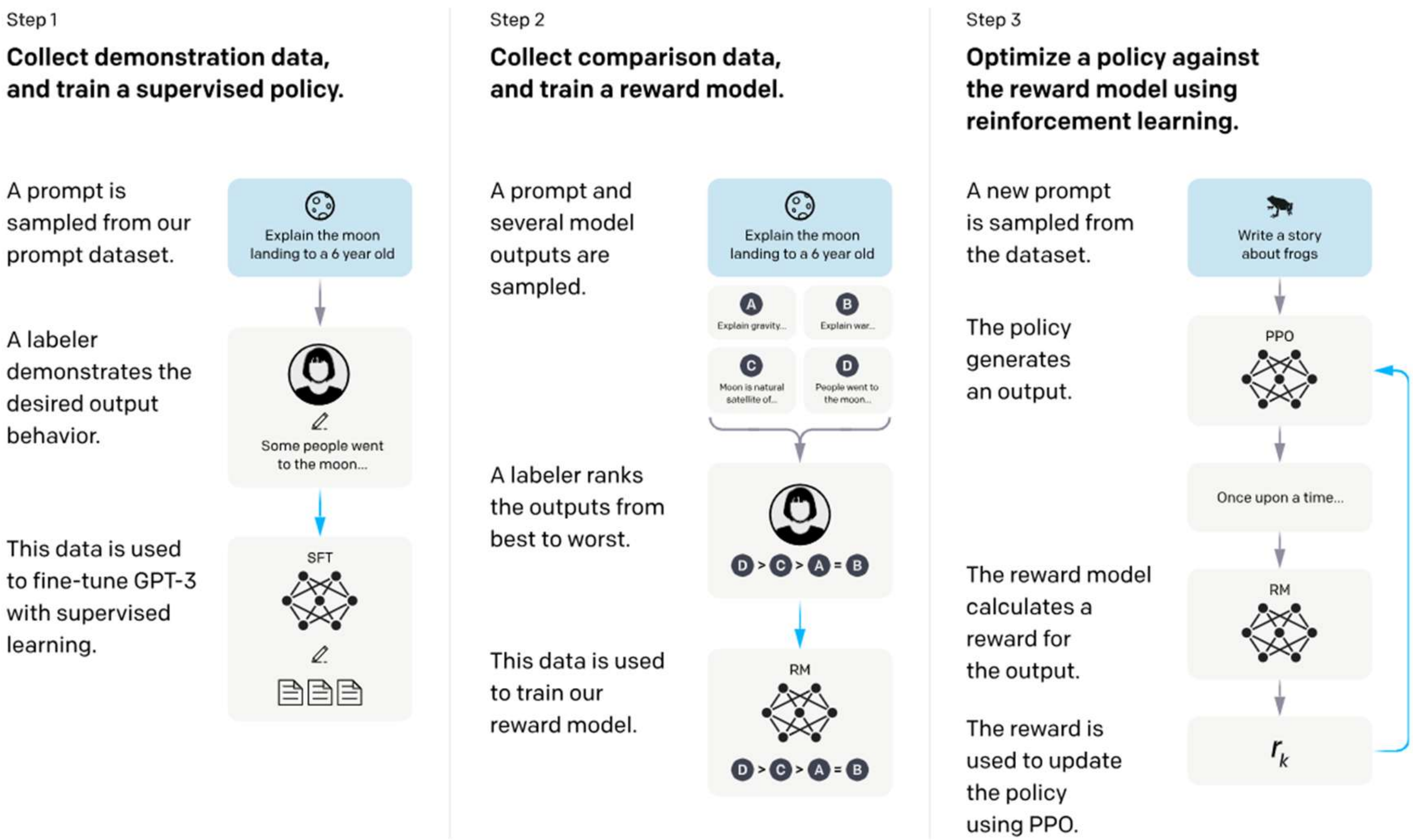

Step 1
Collect demonstration data, and train a supervised policy.
A prompt is sampled from our prompt dataset.
Explain the moon landing to a 6 year old
A labeler demonstrates the desired output behavior.
Some people went to the moon...
This data is used to fine-tune GPT-3 with supervised learning.
SFT
Step 2
Collect comparison data, and train a reward model.
A prompt and several model outputs are sampled.
Explain the moon landing to a 6 year old
A
B
C
D
A labeler ranks the outputs from best to worst.
D > C > A = B
This data is used to train our reward model.
RM
D > C > A = B
Step 3
Optimize a policy against the reward model using reinforcement learning.
A new prompt is sampled from the dataset.
Write a story about frogs
The policy generates an output.
PPO
Once upon a time...
The reward model calculates a reward for the output.
RM
The reward is used to update the policy using PPO.
r_k

RLHF



Step 1
Collect demonstration data, and train a supervised policy.
A prompt is sampled from our prompt dataset.
Explain the moon landing to a 6 year old
A labeler demonstrates the desired output behavior.
Some people went to the moon...
This data is used to fine-tune GPT-3 with supervised learning.
SFT
Step 2
Collect comparison data, and train a reward model.
A prompt and several model outputs are sampled.
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...
A labeler ranks the outputs from best to worst.
D > C > A = B
This data is used to train our reward model.
RM
D > C > A = B
Step 3
Optimize a policy against the reward model using reinforcement learning.
A new prompt is sampled from the dataset.
Write a story about frogs
The policy generates an output.
PPO
Once upon a time...
The reward model calculates a reward for the output.
RM
The reward is used to update the policy using PPO.
r_k

RLHF



Step 1
Collect demonstration data, and train a supervised policy.
A prompt is sampled from our prompt dataset.
Explain the moon landing to a 6 year old
A labeler demonstrates the desired output behavior.
Some people went to the moon...
This data is used to fine-tune GPT-3 with supervised learning.
SFT
Step 2
Collect comparison data, and train a reward model.
A prompt and several model outputs are sampled.
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...
A labeler ranks the outputs from best to worst.
D > C > A = B
This data is used to train our reward model.
RM
D > C > A = B
Step 3
Optimize a policy against the reward model using reinforcement learning.
A new prompt is sampled from the dataset.
Write a story about frogs
The policy generates an output.
PPO
Once upon a time...
The reward model calculates a reward for the output.
RM
The reward is used to update the policy using PPO.
r_k

RLHF

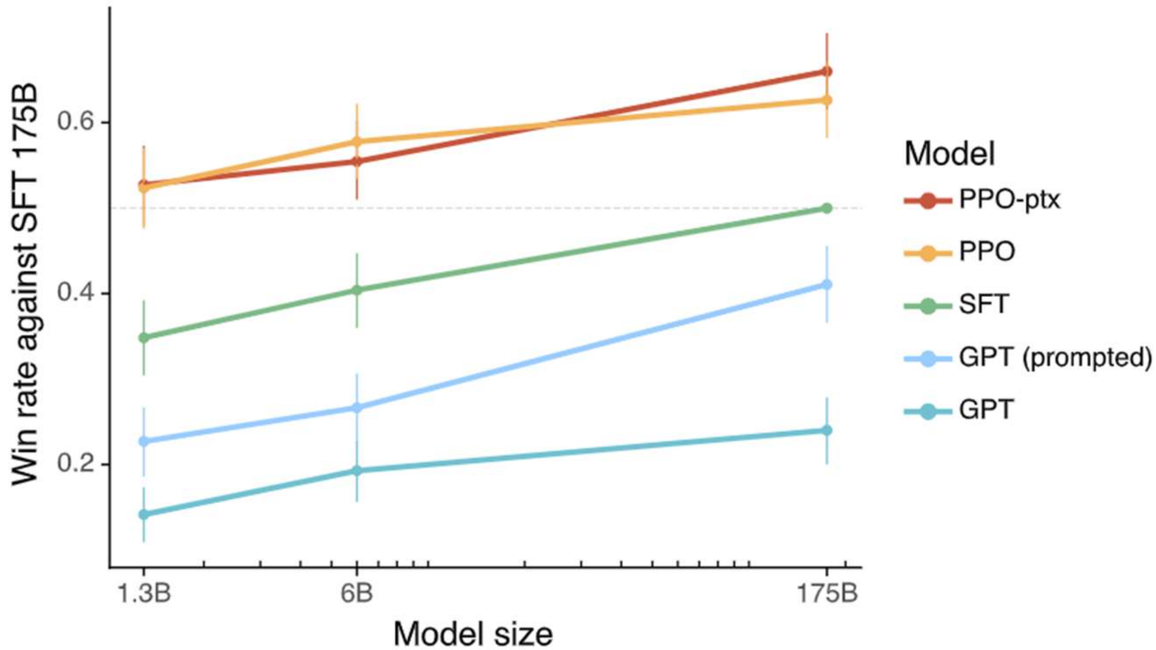


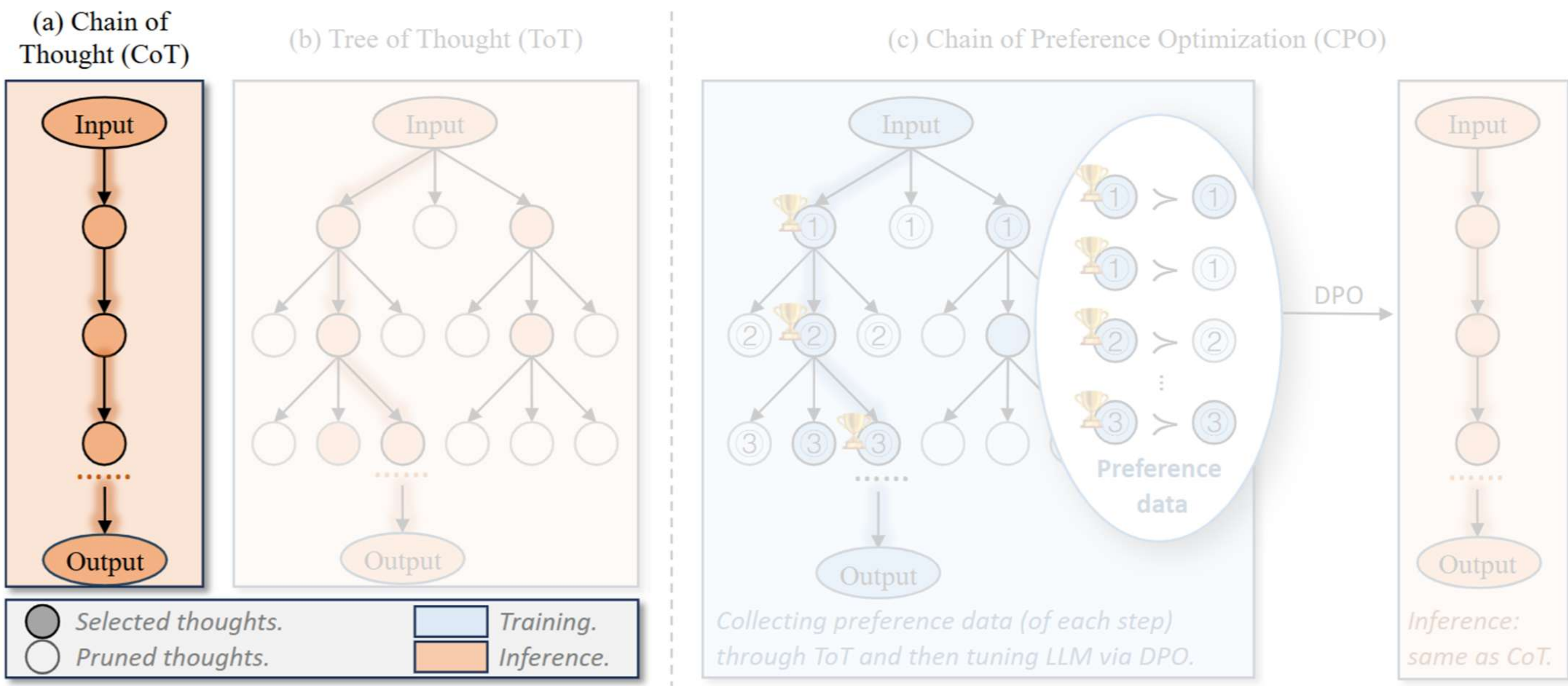
Figure 1: Human evaluations of various models on our API prompt distribution, evaluated by how often outputs from each model were preferred to those from the 175B SFT model. Our InstructGPT models (PPO-ptx) as well as its variant trained without pretraining mix (PPO) significantly outperform the GPT-3 baselines (GPT, GPT prompted); outputs from our 1.3B PPO-ptx model are preferred to those from the 175B GPT-3. Error bars throughout the paper are 95% confidence intervals.

DPO

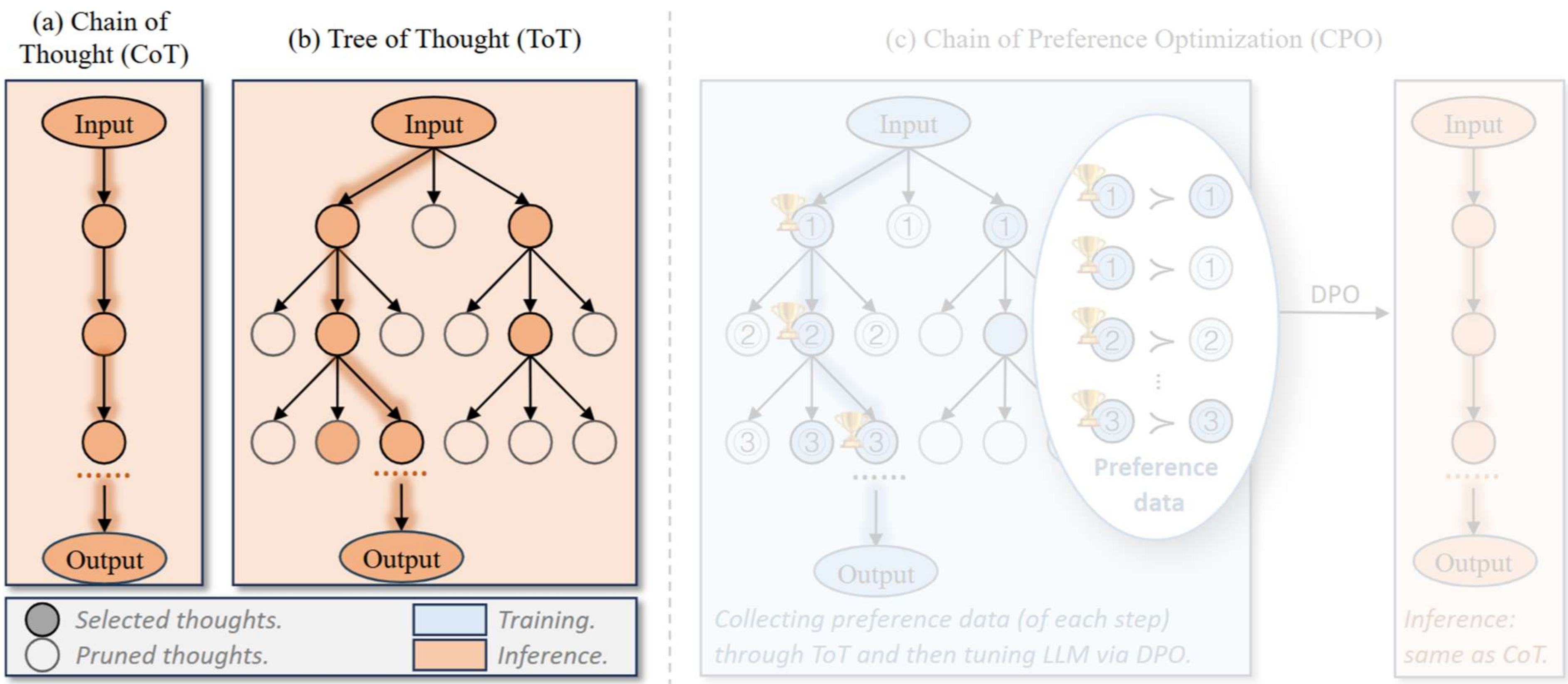
Chain of Preference Optimization: Improving Chain-of-Thought Reasoning in LLMs

Xuan Zhang[*12], Chao Du[†1], Tianyu Pang[1], Qian Liu[1], Wei Gao[2], Min Lin[1]
[1]Sea AI Lab, Singapore
[2]School of Computing and Information Systems, Singapore Management University
xuanzhang.2020@phdcs.smu.edu.sg; weigao@smu.edu.sg;
{duchao, liuqian, tianyupang, linmin}@sea.com

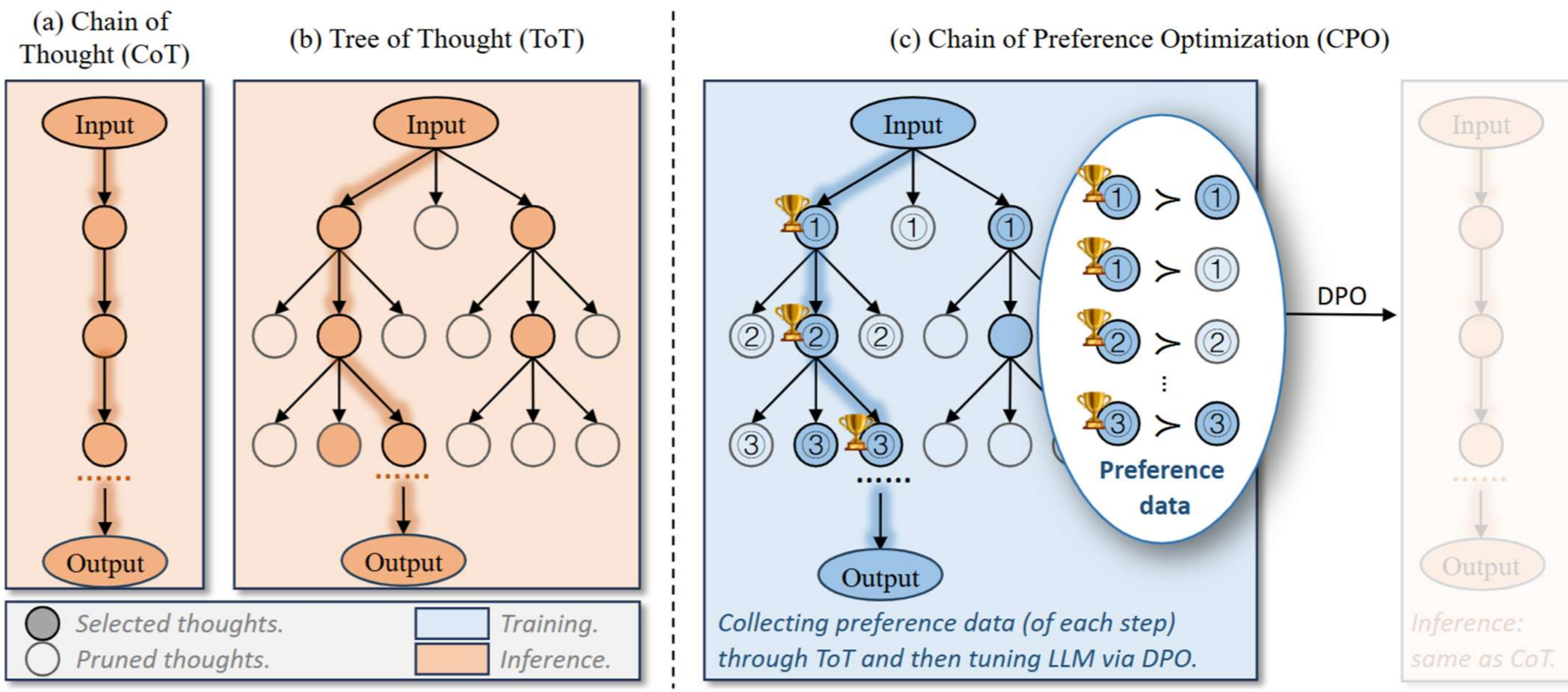
Chain of Preference Optimization



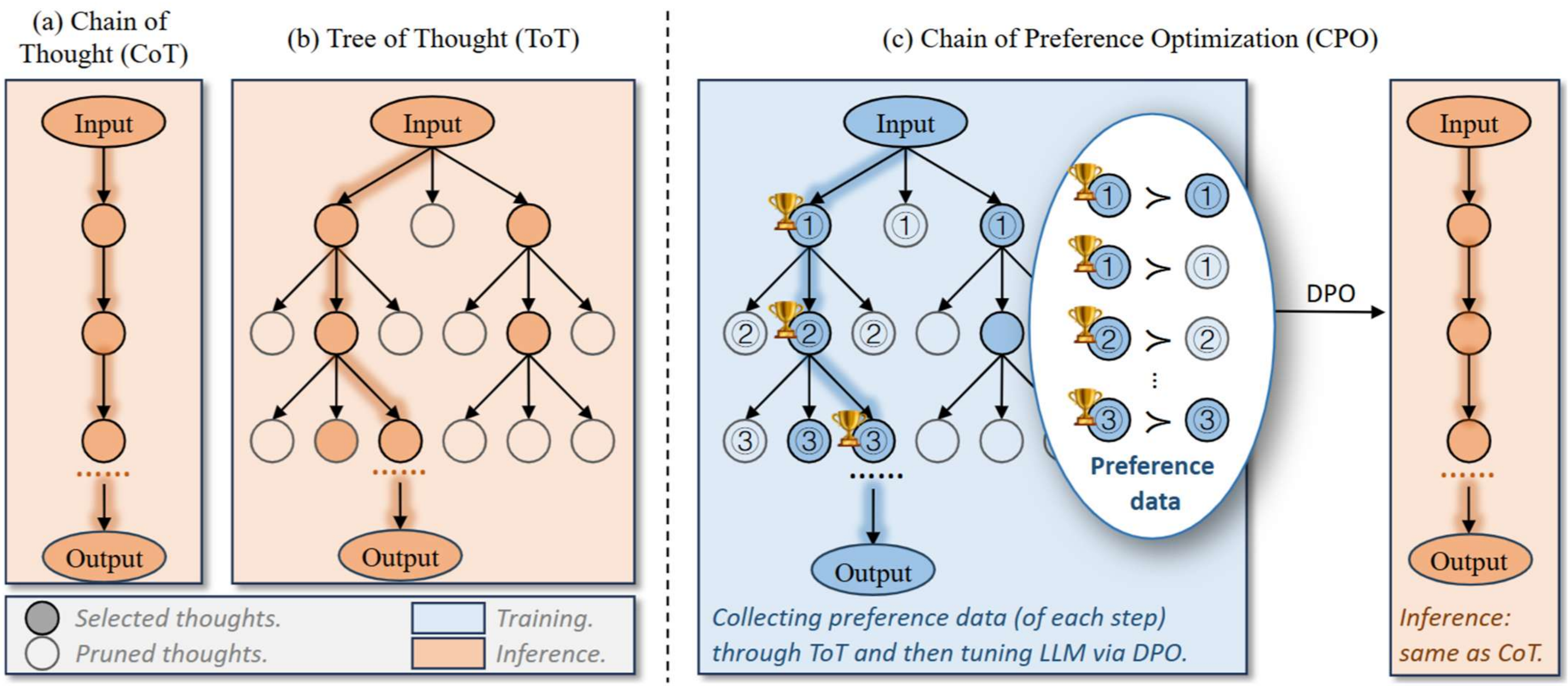
Chain of Preference Optimization



Chain of Preference Optimization



Chain of Preference Optimization



Chain of Preference Optimization

Table 1: Experimental results for ToT, CoT, TS-SFT, and our proposed CPO across complex task including question answering, fact verification, and arithmetic reasoning are presented. * mean significantly better than the best baseline (TS-SFT) with $p < 0.01$. **Bold** denotes the best methoc and the second best if the top method is ToT.

		ToT [8]		CoT [1]		TS-SFT [11]		CPO (ours)	
		Acc. (%)↑	Latency (s/ins.)↓	Acc. (%)↑	Latency (s/ins.)↓	Acc. (%)↑	Latency (s/ins.)↓	Acc. (%)↑	Latency (s/ins.)↓
LLaMA2-7B									
Question Answering	Bam.	**33.6**	1168.4	29.6	37.2	30.4	36.5	**32.0***	38.2
	2Wiki.	28.6	847.6	26.3	35.7	27.6	35.5	**29.7***	35.7
	Hot.	23.0	1100.7	21.0	45.5	22.7	44.8	**24.0***	41.1
Fact Verification	FVR.	47.3	2087.1	45.8	33.8	47.5	34.0	**53.2***	36.8
	FVRS.	47.5	2539.5	44.3	40.6	46.0	40.4	**49.0***	41.2
	Vita.	50.7	2639.3	47.3	35.9	51.0	40.1	**52.7***	40.1
Arithmetic	SVA.	42.7	1861.1	37.7	33.3	43.1	30.2	**46.0***	32.1
Average Performance		39.1	1749.1	36.0	37.4	38.3	37.4	**40.9***	37.9

GRPO

Maximizing Confidence Alone Improves Reasoning

Mihir Prabhudesai* Carnegie Mellon University

Lili Chen* Carnegie Mellon University

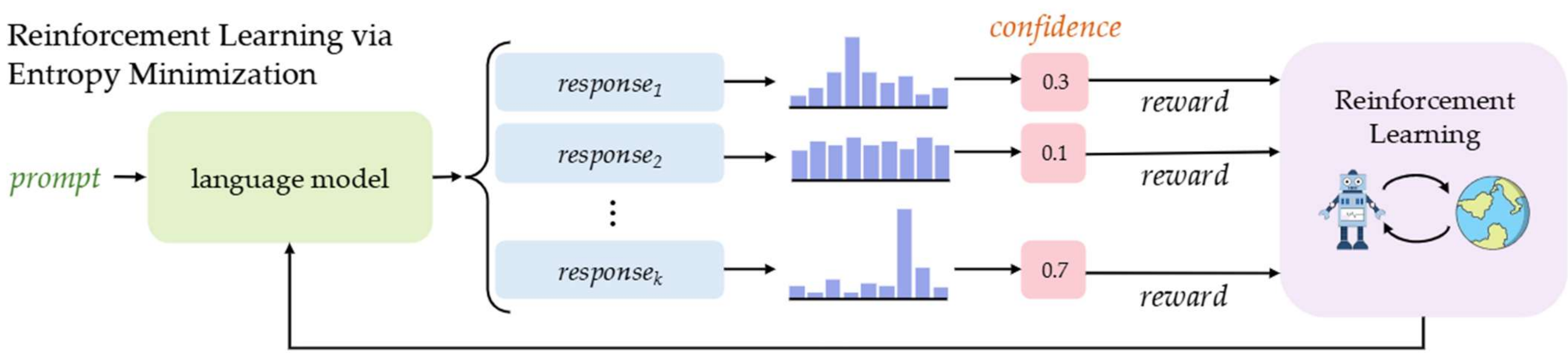
Alex Ippoliti* Carnegie Mellon University

Katerina Fragkiadaki Carnegie Mellon University

Hao Liu Carnegie Mellon University

Deepak Pathak Carnegie Mellon University

Maximizing Confidence



Maximizing Confidence

Give confidence (negative entropy) as a reward to GRPO:

- Multiply the probability of a token with its log probability
  - $p_t(v) \log p_t(v)$ where v is a token at positive t
- Sum across all tokens in the vocabulary at position t
  - $\sum_{v \in V} p_t(v) \log p_t(v)$ where V is the vocabulary
  - Most of them will be close to 0
- Average across tokens in the sequence:
  - $R(y_{pred}) = \frac{1}{T} \sum_{t=1}^{T} \sum_{v \in V} p_t(v) \log p_t(v)$
  - This is the final reward function!

Maximizing Confidence

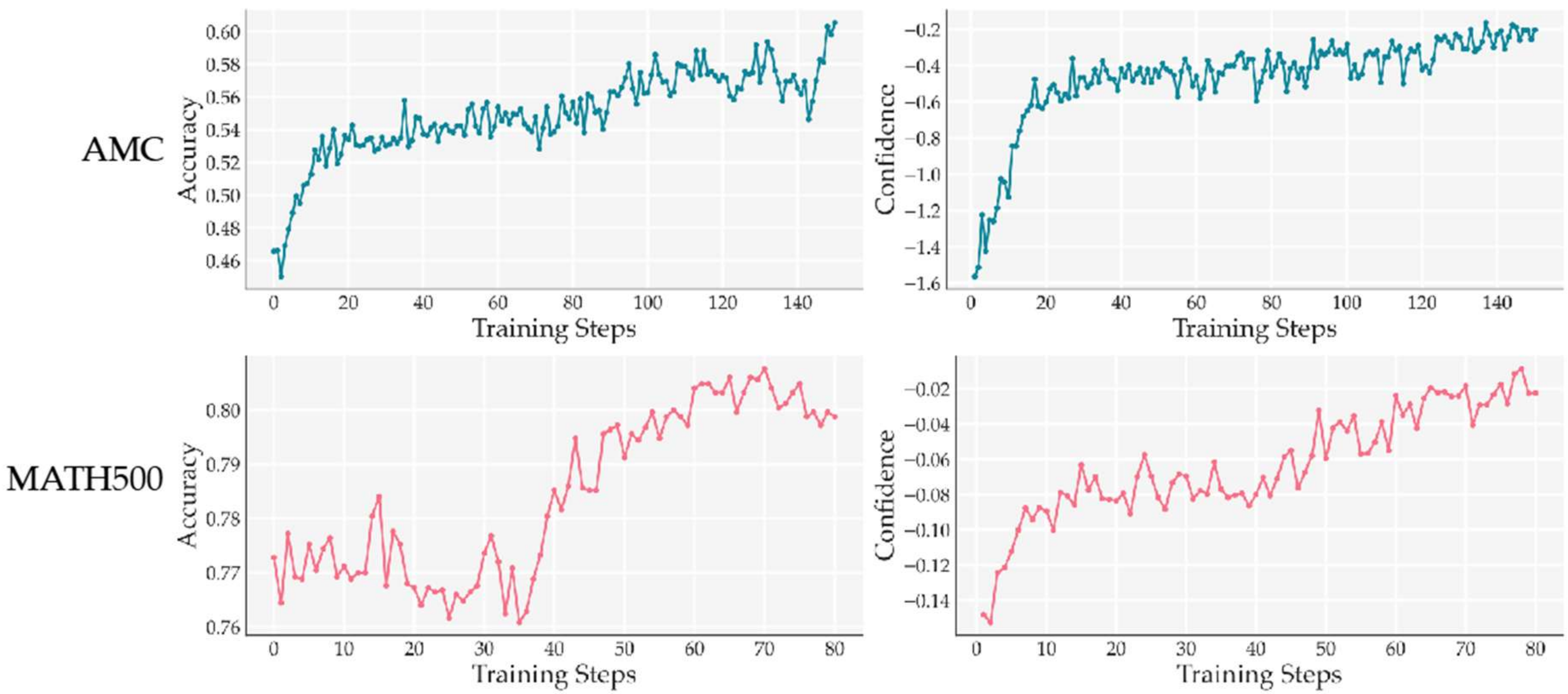


Figure 3: Accuracy and confidence over the course of training. The trends indicate that accuracy and confidence are indeed highly correlated and therefore it is natural to use confidence as a reward.

Conclusion

Conclusion

Welcome to the Era of Experience

David Silver, Richard S. Sutton*

Abstract

We stand on the threshold of a new era in artificial intelligence that promises to achieve an unprecedented level of ability. A new generation of agents will acquire superhuman capabilities by learning predominantly from experience. This note explores the key characteristics that will define this upcoming era.

Conclusion

LLMs are changing a lot of how we use RL:

- Infinite state and action spaces
- Human-data to experience-data
  - Human asks a question, agent responds, human gives feedback
  - Agent interacts with its environment and receives signals from the environment

Conclusion

In the era of experience,

- Agents will not solely require on human feedback to improve
- Example:
  - A health and wellness agent prescribes a diet and exercise regiment
  - Human reward: “yes this is working” / “no this isn’t working”
  - Experience reward: heart rate, sleep patterns, blood work results, etc.
- We can achieve agents that are better than humans

Conclusion

In the era of experience,

- Agents will not solely require on human feedback to improve
- Example:
  - A health and wellness agent prescribes a diet and exercise regiment
  - Human reward: “yes this is working” / “no this isn’t working”
  - Experience reward: heart rate, sleep patterns, blood work results, etc.
- We can achieve agents that are better than humans

<u>My opinion</u>: easier to hype, harder to adopt; we’re living in exciting times!

Reinforcement Learning in the Era of LLMs

Ishika Agarwal

CS 546: Advanced NLP

University of Illinois Urbana-Champaign

CONVAI

Siebel School of Computing and Data Science
CDS