

535514: Reinforcement Learning

Lecture 17 – Proximal Policy Optimization, RLHF, and Value-Based Methods

Ping-Chun Hsieh

April 22, 2024

On-Policy vs Off-Policy Methods

	Policy Optimization	Value-Based	Model-Based	Imitation-Based
On-Policy	Exact PG REINFORCE (w/i baseline) A2C On-policy DAC TRPO Natural PG (NPG) PPO-KL & PPO-Clip RLHF by PPO-KL	Epsilon-Greedy MC Sarsa Expected Sarsa	Model-Predictive Control (MPC) PETS	IRL GAIL IQ-Learn
Off-Policy	Off-policy DPG & DDPG Twin Delayed DDPG (TD3)	Q-learning Double Q-learning DQN & DDQN C51 / QR-DQN / IQN Soft Actor-Critic (SAC)		

Recall: TRPO Algorithm

Step 1: Initialize θ_0

Step 2: For iteration $k = 0, 1, 2, \dots$

Step 2-1: Collect trajectories by running the current policy π_{θ_k}

Step 2-2: Obtain advantage $A^{\theta_k}(s, a)$ for the current policy π_{θ_k}

Step 2-3: Update the policy by solving

$$\theta_{k+1} = \arg \max_{\theta} L_{\pi_{\theta_k}}(\pi_{\theta}) \quad (\equiv \arg \max_{\theta} \mathbb{E}_{s \sim d_{\mu}^{\pi_{\theta_k}}, a \sim \pi_{\theta_k}(\cdot | s)} \left[\frac{\pi_{\theta}(a | s)}{\pi_{\theta_k}(a | s)} A^{\theta_k}(s, a) \right])$$

subject to $\bar{D}_{KL}(\pi_{\theta_k} \| \pi_{\theta}) \leq \delta$

- ▶ The TR subproblem under approximation:

$$\begin{aligned} \text{Maximize } & (\theta - \theta_k)^T \nabla_{\theta} L_{\theta_k}(\theta) |_{\theta=\theta_k} & \rightarrow \theta = \theta_k + \alpha \cdot H_{\theta_k}^{-1} \nabla_{\theta} L_{\theta_k}(\theta) |_{\theta=\theta_k} \\ \text{subject to } & \frac{1}{2} (\theta - \theta_k)^T H(\theta - \theta_k) \leq \delta \end{aligned}$$

Assignment for Lec 16:

- ▶ Could you find any part of the code that we have not discussed in this lecture?

(1) Find an initial step size α by

$$\frac{1}{2}\alpha^2(H^{-1} \nabla_{\theta} L_{\theta_k}(\theta))^T H(H^{-1} \nabla_{\theta} L_{\theta_k}(\theta)) = \delta$$

- ▶ Spend 30 minutes reading the code of TRPO
 - ▶ <https://github.com/ikostrikov/pytorch-trpo>

```
51 def trpo_step(model, get_loss, get_kl, max_kl, damping):
52     loss = get_loss()
53     grads = torch.autograd.grad(loss, model.parameters())
54     loss_grad = torch.cat([grad.view(-1) for grad in grads]).data
55
56     def Fvp(v):
57         kl = get_kl()
58         kl = kl.mean()
59
60         grads = torch.autograd.grad(kl, model.parameters(), create_graph=True)
61         flat_grad_kl = torch.cat([grad.view(-1) for grad in grads])
62
63         kl_v = (flat_grad_kl * Variable(v)).sum()
64         grads = torch.autograd.grad(kl_v, model.parameters())
65         flat_grad_grad_kl = torch.cat([grad.contiguous().view(-1) for grad in grads]).data
66
67         return flat_grad_grad_kl + v * damping
68
69     stepdir = conjugate_gradients(Fvp, -loss_grad, 10)
70
71     shs = 0.5 * (stepdir * Fvp(stepdir)).sum(0, keepdim=True)
72
73     lm = torch.sqrt(shs / max_kl)
74     fullstep = stepdir / lm[0]
75
76     negdotstepdir = (-loss_grad * stepdir).sum(0, keepdim=True)
77     print(("lagrange multiplier:", lm[0], "grad_norm:", loss_grad.norm()))
78
79     prev_params = get_flat_params_from(model)
80     success, new_params = linesearch(model, get_loss, prev_params, fullstep,
81                                     negdotstepdir / lm[0])
82     set_flat_params_to(model, new_params)
83
84     return loss
```

(2) Use line search to fine-tune the step size α to ensure improvement in the original objective function

Assignment for Lec 16:

(3) Use conjugate gradient to solve

$$Hx = \nabla_{\theta} L_{\theta_k}(\theta) \Big|_{\theta=\theta_k}$$

(Instead of taking H^{-1})

(For more details, please check
Stephen Boyd's lecture notes:

<https://web.stanford.edu/class/>

ee364b/lectures/conj_grad_slides.pdf)

```
8 def conjugate_gradients(Avp, b, nsteps, residual_tol=1e-10):
9     x = torch.zeros(b.size())
10    r = b.clone()
11    p = b.clone()
12    rdotr = torch.dot(r, r)
13    for i in range(nsteps):
14        _Avp = Avp(p)
15        alpha = rdotr / torch.dot(p, _Avp)
16        x += alpha * p
17        r -= alpha * _Avp
18        new_rdotr = torch.dot(r, r)
19        betta = new_rdotr / rdotr
20        p = r + betta * p
21        rdotr = new_rdotr
22        if rdotr < residual_tol:
23            break
24    return x
25
26
27 def linesearch(model,
28                 f,
29                 x,
30                 fullstep,
31                 expected_improve_rate,
32                 max_backtracks=10,
33                 accept_ratio=.1):
34     fval = f(True).data
35     print("fval before", fval.item())
36     for (_n_backtracks, stepfrac) in enumerate(.5**np.arange(max_backtracks)):
37         xnew = x + stepfrac * fullstep
38         set_flat_params_to(model, xnew)
39         newfval = f(True).data
40         actual_improve = fval - newfval
41         expected_improve = expected_improve_rate * stepfrac
42         ratio = actual_improve / expected_improve
43         print("a/e/r", actual_improve.item(), expected_improve.item(), ratio.item())
44
45         if ratio.item() > accept_ratio and actual_improve.item() > 0:
46             print("fval after", newfval.item())
47             return True, xnew
48
49     return False, x
```

Proximal Policy Optimization (PPO)

Schulman et al., “Proximal Policy Optimization Algorithms,” arXiv 2017

“Proximal” Viewpoint in Optimization

- ▶ Reinterpret GD from a “proximal” viewpoint

GD $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$

Proximal $x_{k+1} = \arg \min_x \left\{ f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2\alpha_k} \|x - x_k\|_2^2 \right\}$

first-order approximation proximal term

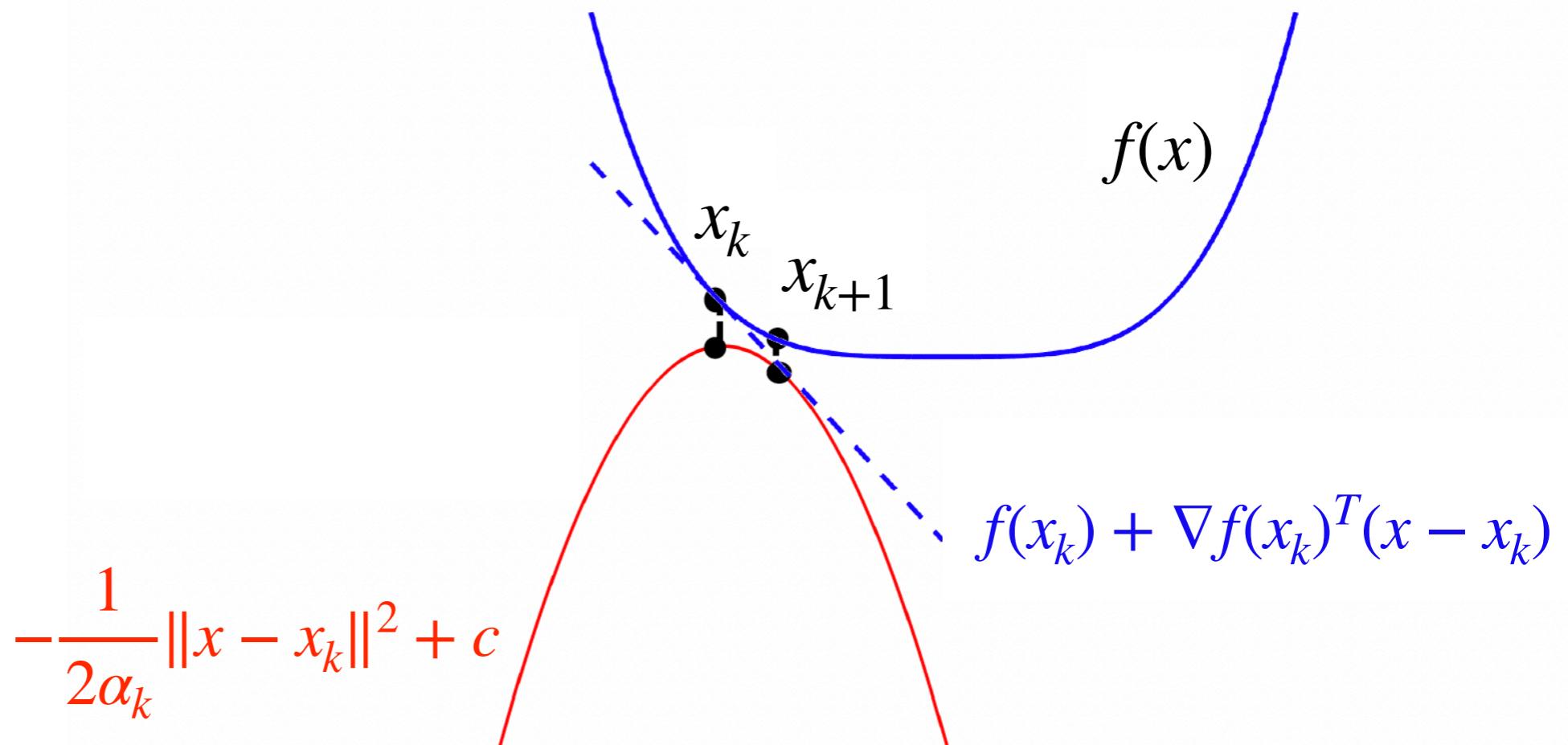
- ▶ Question: Are these two schemes equivalent?

“Proximal” Viewpoint of GD:

$$x_{k+1} = \arg \min_x \left\{ f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2\alpha_k} \|x - x_k\|_2^2 \right\}$$

$$=: g(x)$$

- ▶ **Question:** What condition does x_{k+1} need to satisfy?



A Proximal View of TRPO: Proximal Policy Optimization

$$\text{Maximize } \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_k}}, a \sim \pi_{\theta_k}(\cdot | s)} \left[\frac{\pi_\theta(a | s)}{\pi_{\theta_k}(a | s)} A^{\theta_k}(s, a) \right] \quad (\text{TRPO})$$

$$\text{subject to } \bar{D}_{KL}(\pi_{\theta_k} \| \pi_\theta) \leq \delta$$



$$\max_{\theta} \mathbb{E}_{s \sim d_\mu^{\pi_{\theta_k}}, a \sim \pi_{\theta_k}(\cdot | s)} \left[\frac{\pi_\theta(a | s)}{\pi_{\theta_k}(a | s)} A^{\theta_k}(s, a) - \beta_k D_{KL}(\pi_{\theta_k}(\cdot | s) \| \pi_\theta(\cdot | s)) \right]$$

(PPO with KL penalty)

- ▶ PPO can be also viewed as the *Lagrangian* of (TRPO) with β_k being the *Lagrange multiplier*

PPO With Adaptive KL Penalty (PPO-KL)

Proximal policy optimization (PPO):

$$\max_{\theta} \mathbb{E}_{s \sim d_{\mu}^{\pi_{\theta_k}}, a \sim \pi_{\theta_k}(\cdot | s)} \left[\frac{\pi_{\theta}(a | s)}{\pi_{\theta_k}(a | s)} A^{\theta_k}(s, a) - \beta_k D_{KL}(\pi_{\theta_k}(\cdot | s) || \pi_{\theta}(\cdot | s)) \right]$$

- ▶ The penalty coefficient β_k changes between iterations to approximately enforce the KL constraint (How?)
- ▶ Heuristic:

If $\bar{D}_{KL}(\pi_{\theta_k} || \pi_{\theta}) > 1.5d_{target}$: $\beta_{k+1} = 2\beta_k$

If $\bar{D}_{KL}(\pi_{\theta_k} || \pi_{\theta}) < d_{target}/1.5$: $\beta_{k+1} = 0.5\beta_k$

(d_{target} is a hyperparameter, e.g. $d_{target} = 0.01$)

PPO-KL Algorithm (Formally)

Input: initial policy parameters θ_0 , initial KL penalty β_0 , target KL-divergence δ

for $k = 0, 1, 2, \dots$ do

Collect trajectories and
estimate $A(s, a)$

Collect set of partial trajectories \mathcal{D}_k on policy $\pi_k = \pi(\theta_k)$

Estimate advantages $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm

Compute policy update

$$\theta_{k+1} = \arg \max_{\theta} \mathcal{L}_{\theta_k}(\theta) - \beta_k \bar{D}_{KL}(\theta || \theta_k)$$

Update the policy

by taking K steps of minibatch SGD (via Adam)

if $\bar{D}_{KL}(\theta_{k+1} || \theta_k) \geq 1.5\delta$ then

$$\beta_{k+1} = 2\beta_k$$

else if $\bar{D}_{KL}(\theta_{k+1} || \theta_k) \leq \delta/1.5$ then

$$\beta_{k+1} = \beta_k/2$$

end if

end for

Adaptively tune the
penalty coefficient

Don't use second order approximation for KL which is
expensive, use standard gradient descent

PPO With a Clipped Surrogate Objective (PPO-Clip)

- **Recall:** TRPO

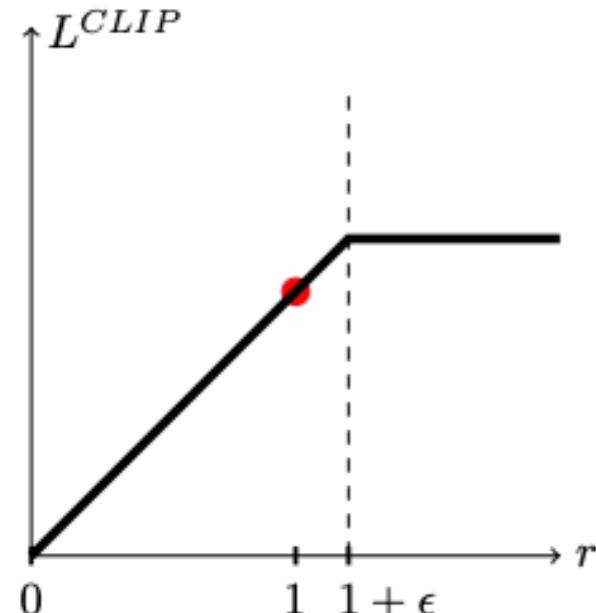
$$\max_{\theta} \mathbb{E}_{s \sim d_{\mu}^{\pi_{\theta_k}}, a \sim \pi_{\theta_k}(\cdot | s)} \left[\frac{\pi_{\theta}(a | s)}{\pi_{\theta_k}(a | s)} A^{\theta_k}(s, a) \right]$$

$$\text{subject to } \bar{D}_{KL}(\pi_{\theta_k} \| \pi_{\theta}) \leq \delta$$

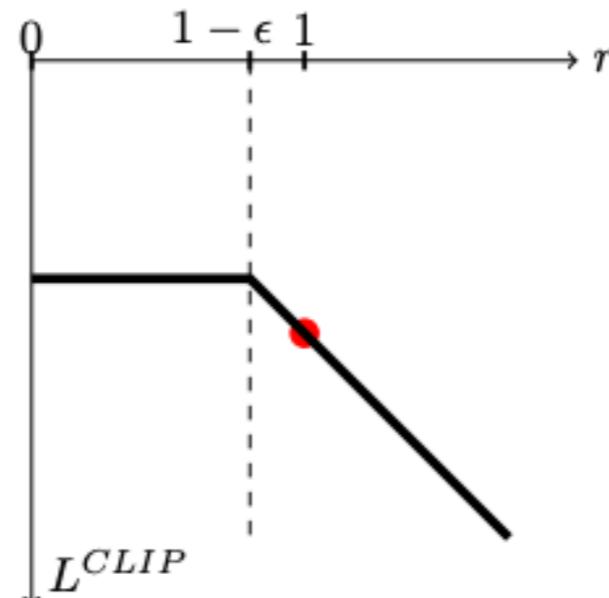
- **Idea:** Clip the ratio $\frac{\pi_{\theta}(a | s)}{\pi_{\theta_k}(a | s)}$ (instead of using KL constraint)

$$\max_{\theta} \mathbb{E}_{s \sim d_{\mu}^{\pi_{\theta_k}}, a \sim \pi_{\theta_k}(\cdot | s)} \left[\min \left\{ \frac{\pi_{\theta}(a | s)}{\pi_{\theta_k}(a | s)} A^{\theta_k}(s, a), \text{clip}\left(\frac{\pi_{\theta}(a | s)}{\pi_{\theta_k}(a | s)}, 1 - \varepsilon, 1 + \varepsilon\right) A^{\theta_k}(s, a) \right\} \right]$$

Case 1: $A > 0$



Case 2: $A < 0$



(ε is a hyperparameter, e.g. $\varepsilon = 0.2$)

Clipping prevents policy from having incentive to go far away from θ_k

PPO-Clip Algorithm (Formally)

Input: initial policy parameters θ_0 , clipping threshold ϵ
for $k = 0, 1, 2, \dots$ do

Collect trajectories and
estimate $A(s, a)$

Collect set of partial trajectories \mathcal{D}_k on policy $\pi_k = \pi(\theta_k)$

Estimate advantages $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm

Compute policy update

$$\theta_{k+1} = \arg \max_{\theta} \mathcal{L}_{\theta_k}^{CLIP}(\theta)$$

by taking K steps of minibatch SGD (via Adam), where

$$\mathcal{L}_{\theta_k}^{CLIP}(\theta) = \mathbb{E}_{\tau \sim \pi_k} \left[\sum_{t=0}^T \left[\min(r_t(\theta) \hat{A}_t^{\pi_k}, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t^{\pi_k}) \right] \right]$$

end for

PPO does not use 2nd-order derivatives

Question: What's the principle behind PPO-Clip?

(State-Wise) Performance Difference Lemma

- ▶ **(Average) Performance Difference Lemma:**

$$V^{\pi_{new}}(\mu) - V^{\pi_{old}}(\mu) = \frac{1}{1-\gamma} \mathbb{E}_{s' \sim d_{\mu}^{\pi_{new}}} \mathbb{E}_{a' \sim \pi_{new}(\cdot | s')} \left[A^{\pi_{old}}(s', a') \right]$$

- ▶ **State-Wise Performance Difference Lemma:**

$$V^{\pi_{new}}(s) - V^{\pi_{old}}(s) = \frac{1}{1-\gamma} \mathbb{E}_{s' \sim d_s^{\pi_{new}}} \mathbb{E}_{a' \sim \pi_{new}(\cdot | s')} \left[A^{\pi_{old}}(s', a') \right]$$

Question: How to ensure $\pi_{new} \geq \pi_{old}$ (i.e., $V^{\pi_{new}}(s) - V^{\pi_{old}}(s) \geq 0, \forall s$)?

Sufficient Condition for Strict Policy Improvement

Lemma (Sufficient Condition for Policy Improvement):

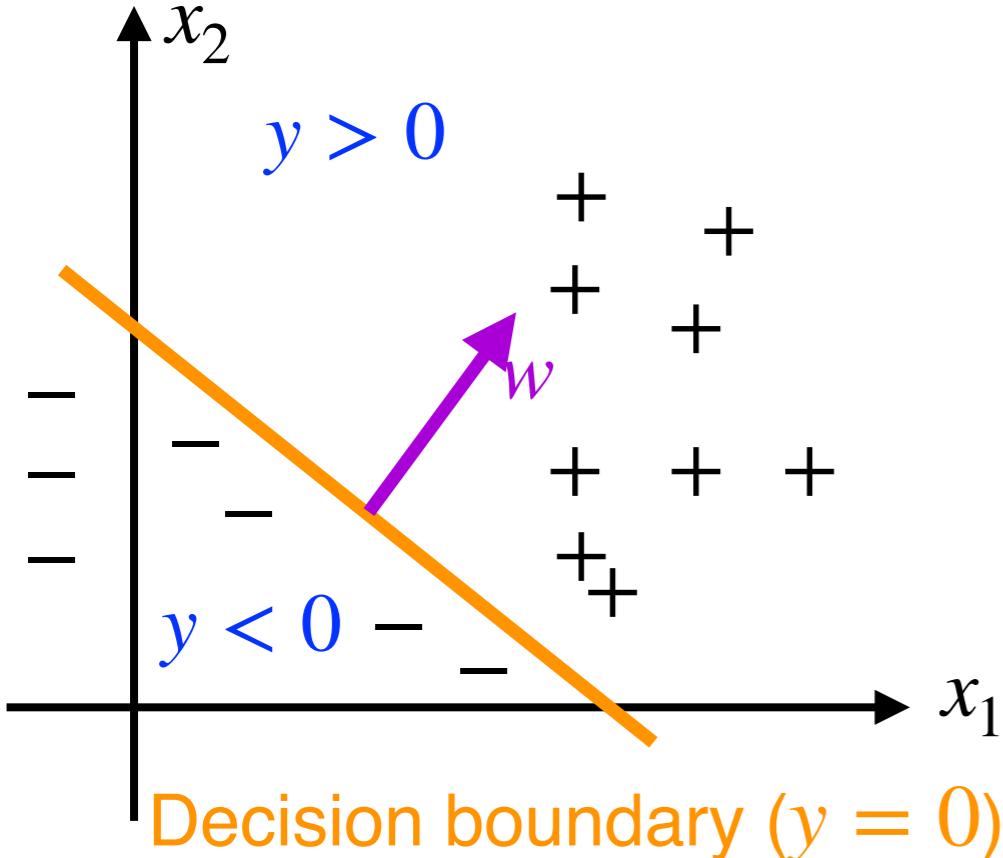
We have $\pi_{new} \geq \pi_{old}$ if

$$(\pi_{new}(a|s) - \pi_{old}(a|s)) \cdot A(s, a; \pi_{old}) \geq 0, \quad \text{for all } (s, a)$$

Proof: $V^{\pi_{new}}(s) - V^{\pi_{old}}(s) = \frac{1}{1-\gamma} \mathbb{E}_{s' \sim d_s^{\pi_{new}}} \mathbb{E}_{a' \sim \pi_{new}(\cdot|s')} \left[A^{\pi_{old}}(s', a') \right]$

Question: Have you learned anything similar in supervised learning?

Our Idea: Connect Policy Improvement & Binary Classification



	Binary Classification	Policy Improvement in RL
Feature	x	State-action pair
Label	$t(x)$ (+ or -)	Sign of $A^\pi(s, a)$
Classifier	$y(x) = w^T x + w_0$	$\pi_{new}(a s) - \pi_{old}(a s)$
Goal	$y(x) \cdot t(x) \geq 0$	
Loss Function	$\max\{0, \varepsilon - y(x) \cdot t(x)\}$	Hinge loss

Rewriting PPO-Clip's Objective as Hinge Loss

$$J_{PPO}(\theta) = \frac{1}{|D|} \sum_{(s,a) \in D} \min \left\{ \frac{\pi_\theta(a | s)}{\pi_{\theta_k}(a | s)} A(s, a; \theta_k), \text{clip}\left(\frac{\pi_\theta(a | s)}{\pi_{\theta_k}(a | s)}, 1 - \varepsilon, 1 + \varepsilon\right) A(s, a; \theta_k) \right\}$$

$$L_{Hinge}(\theta) = \frac{1}{|D|} \sum_{(s,a) \in D} |A(s, a; \theta_k)| \cdot \max \left\{ 0, \varepsilon - \text{sign}(A(s, a; \theta_k)) \cdot \left(\frac{\pi_\theta(a | s)}{\pi_{\theta_k}(a | s)} - 1 \right) \right\}$$

Theorem:

$$J_{PPO}(\theta) = -L_{Hinge}(\theta) + \text{constant}$$

(This also implies that $\nabla_\theta J_{PPO}(\theta) = -\nabla_\theta L_{Hinge}(\theta)$)

- ▶ The perspective of “*hinge loss*” for PPO-Clip can enjoy nice theoretical convergence guarantees:

PPO-Clip Attains Global Optimality: Towards Deeper Understandings of Clipping

Nai-Chieh Huang, Ping-Chun Hsieh, Kuo-Hao Ho, I-Chen Wu

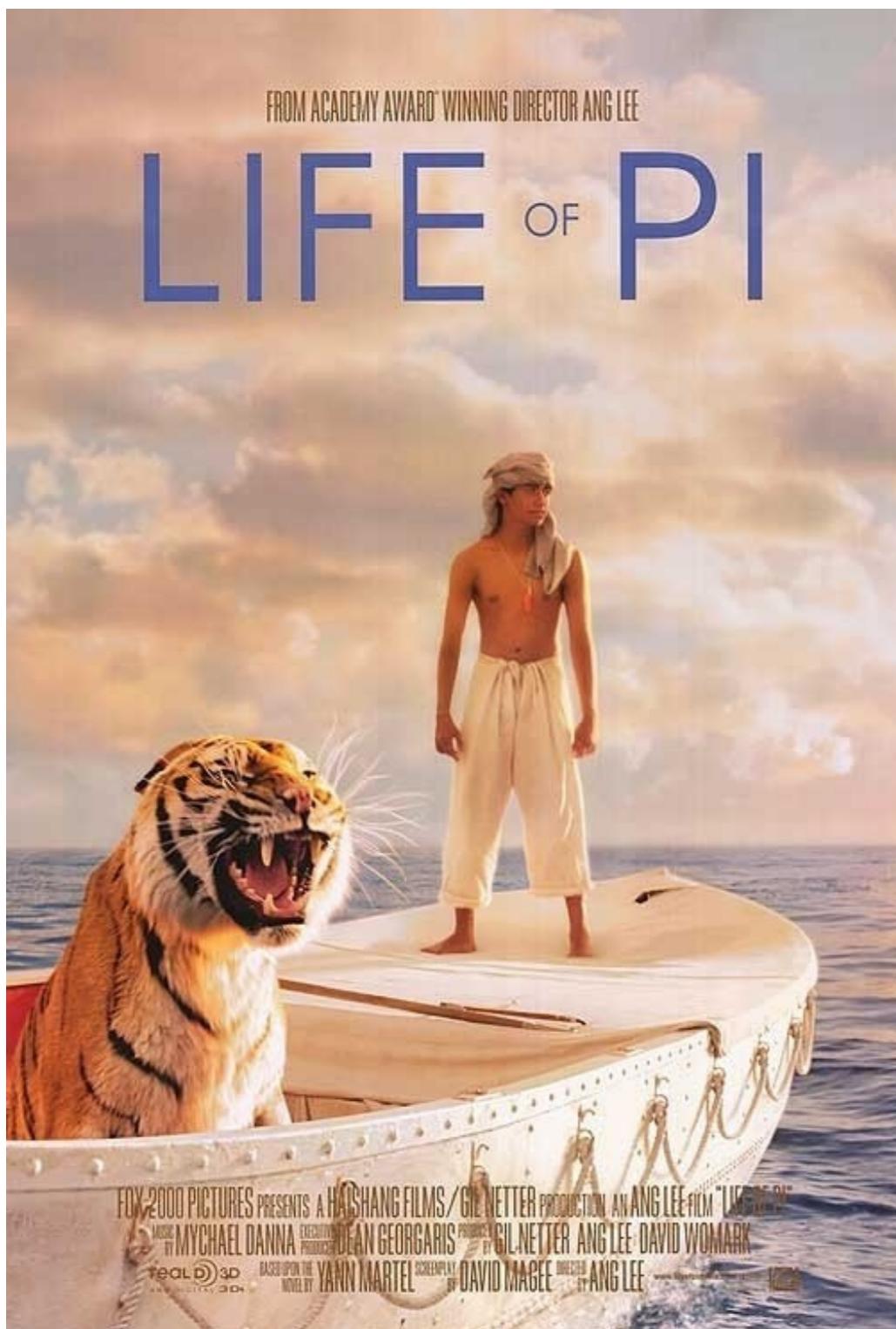
Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
`{naich.cs09, pinghsieh}@nycu.edu.tw`

Abstract

Proximal Policy Optimization algorithm employing a clipped surrogate objective (PPO-Clip) is a prominent exemplar of the policy optimization methods. However, despite its remarkable empirical success, PPO-Clip lacks theoretical substantiation to date. In this paper, we contribute to the field by establishing the first global convergence results of a PPO-Clip variant in both tabular and neural function approximation settings. Our findings highlight the $O(1/\sqrt{T})$ min-iterate convergence rate specifically in the context of neural function approximation. We tackle the inherent challenges in analyzing PPO-Clip through three central concepts: (i) We introduce a generalized version of the PPO-Clip objective, illuminated by its connection with the hinge loss. (ii) Employing entropic mirror descent, we establish asymptotic convergence for tabular PPO-Clip with direct policy parameterization. (iii) Inspired by the tabular analysis, we streamline convergence analysis by introducing a two-step policy improvement approach. This decouples policy search from complex neural policy parameterization using a regression-based update scheme. Furthermore, we gain deeper insights into the efficacy of PPO-Clip by interpreting these generalized objectives. Our theoretical findings also mark the first characterization of the influence of the clipping mechanism on PPO-Clip convergence. Importantly, the clipping range affects only the pre-constant of the convergence rate.

main variants: PPO-KL and PPO-Clip, each with distinct characteristics. PPO-KL adds a Kullback-Leibler divergence penalty to the objective, while PPO-Clip integrates probability ratio clipping. These variants showcase remarkable performance across various environments, with PPO standing out for its computational efficiency (Chen, Peng, and Zhang 2018; Ye et al. 2020; Byun, Kim, and Wang 2020).

Given the empirical success of these policy optimization algorithms, recent works have made significant strides in enhancing their theoretical guarantees. In particular, (Agarwal et al. 2020; Bhandari and Russo 2019) prove the global convergence result of the policy gradient algorithm under different settings. Additionally, (Mei et al. 2020) establishes the convergence rates of the softmax policy gradient in both the standard and the entropy-regularized settings. Furthermore, it has been shown that various policy gradient algorithms also enjoy global convergence (Fazel et al. 2018; Liu et al. 2020; Wang et al. 2021). In the context of TRPO and PPO, (Shani, Efroni, and Mannor 2020) have utilized the mirror descent method to establish the convergence rate of adaptive TRPO under both the standard and entropy-regularized settings. Furthermore, (Liu et al. 2019) have provided the convergence rate of PPO-KL and TRPO under neural function approximation. By contrast, despite that PPO-Clip is a computationally efficient and empirically successful method, the



Strict Policy Improvement?

- ▶ Version 1: Monotonic improvement in average performance (e.g., TRPO)
- ▶ Version 2: Monotonic state-wise policy improvement (e.g., PPO-Clip), with extension to NN policy networks
- ▶ Which version of the story do you prefer?

Assignment for this lecture:

- ▶ Spend 30 minutes going through the idea of PPO
- ▶ Spend 30 minutes reading the code of PPO-Clip
 - ▶ <https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail/>
- ▶ Could you explain the purpose of each line?
- ▶ Could you find any part of the code that we have not discussed in this lecture? (Hint: There are at least 2!)

We will discuss this next time!

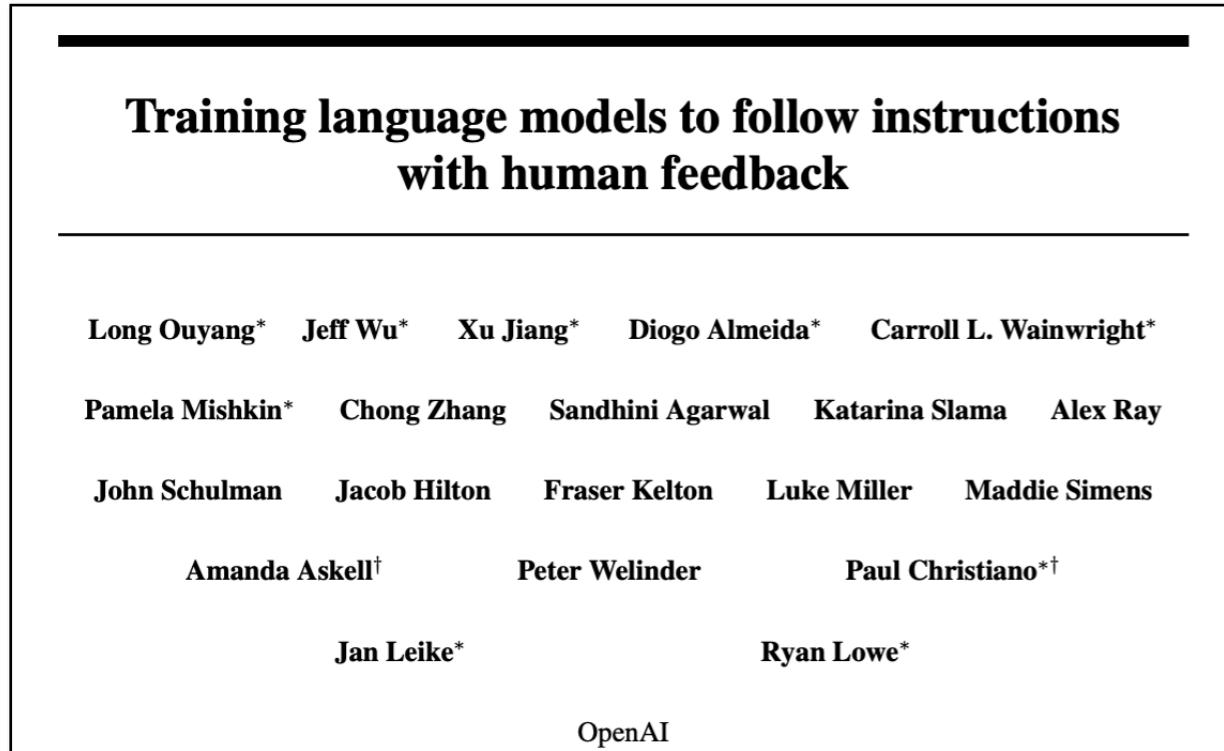
Implementing PPO-Clip in 90 lines:

```
7  class PPO():
8      def __init__(self,
9          actor_critic,
10         clip_param,
11         ppo_epoch,
12         num_mini_batch,
13         value_loss_coef,
14         entropy_coef,
15         lr=None,
16         eps=None,
17         max_grad_norm=None,
18         use_clipped_value_loss=True):
19
20     self.actor_critic = actor_critic
21
22     self.clip_param = clip_param
23     self.ppo_epoch = ppo_epoch
24     self.num_mini_batch = num_mini_batch
25
26     self.value_loss_coef = value_loss_coef
27     self.entropy_coef = entropy_coef
28
29     self.max_grad_norm = max_grad_norm
30     self.use_clipped_value_loss = use_clipped_value_loss
31
32     self.optimizer = optim.Adam(actor_critic.parameters(), lr=lr, eps=eps)
33
34     def update(self, rollouts):
35         advantages = rollouts.rewards[:-1] - rollouts.value_preds[:-1]
36         advantages = (advantages - advantages.mean()) / (
37             advantages.std() + 1e-05)
38
39         value_loss_epoch = 0
40         action_loss_epoch = 0
41         dist_entropy_epoch = 0
42
43         for e in range(self.ppo_epoch):
44             if self.actor_critic.is_recurrent:
45                 data_generator = rollouts.recurrent_generator(
46                     advantages, self.num_mini_batch)
47             else:
48                 data_generator = rollouts.feed_forward_generator(
49                     advantages, self.num_mini_batch)
50
51             for sample in data_generator:
52                 obs_batch, recurrent_hidden_states_batch, actions_batch, \
53                 value_preds_batch, return_batch, masks_batch, old_action_log_probs_batch, \
54                 adv_targ = sample
55
56                 # Reshape to do in a single forward pass for all steps
57                 values, action_log_probs, dist_entropy, _ = self.actor_critic.evaluate_actions(
58                     obs_batch, recurrent_hidden_states_batch, masks_batch,
59                     actions_batch)
59
60
61                 ratio = torch.exp(action_log_probs -
62                                   old_action_log_probs_batch)
63                 surr1 = ratio * adv_targ
64                 surr2 = torch.clamp(ratio, 1.0 - self.clip_param,
65                                     1.0 + self.clip_param) * adv_targ
66                 action_loss = -torch.min(surr1, surr2).mean()
67
68                 if self.use_clipped_value_loss:
69                     value_pred_clipped = value_preds_batch + \
70                         (values - value_preds_batch).clamp(-self.clip_param, self.clip_param)
71                     value_losses = (values - return_batch).pow(2)
72                     value_losses_clipped = (
73                         value_pred_clipped - return_batch).pow(2)
74                     value_loss = 0.5 * torch.max(value_losses,
75                                         value_losses_clipped).mean()
76
77                 else:
78                     value_loss = 0.5 * (return_batch - values).pow(2).mean()
79
80                     self.optimizer.zero_grad()
81                     (value_loss * self.value_loss_coef + action_loss -
82                      dist_entropy * self.entropy_coef).backward()
83                     nn.utils.clip_grad_norm_(self.actor_critic.parameters(),
84                                         self.max_grad_norm)
85
86                     self.optimizer.step()
87
88                     value_loss_epoch += value_loss.item()
89                     action_loss_epoch += action_loss.item()
90                     dist_entropy_epoch += dist_entropy.item()
91
92                     num_updates = self.ppo_epoch * self.num_mini_batch
93
94                     value_loss_epoch /= num_updates
95                     action_loss_epoch /= num_updates
96                     dist_entropy_epoch /= num_updates
97
98             return value_loss_epoch, action_loss_epoch, dist_entropy_epoch
```

Reinforcement Learning From Human Feedback (RLHF)

Some of the material is from the ICML 2023 Tutorial by Nathan Lambert
(<https://icml.cc/media/icml-2023/Slides/21554.pdf>)

RLHF for Language Models (LM), e.g., ChatGPT



The Main Ideas of ChatGPT Comes
From InstructGPT:
<https://arxiv.org/abs/2203.02155>

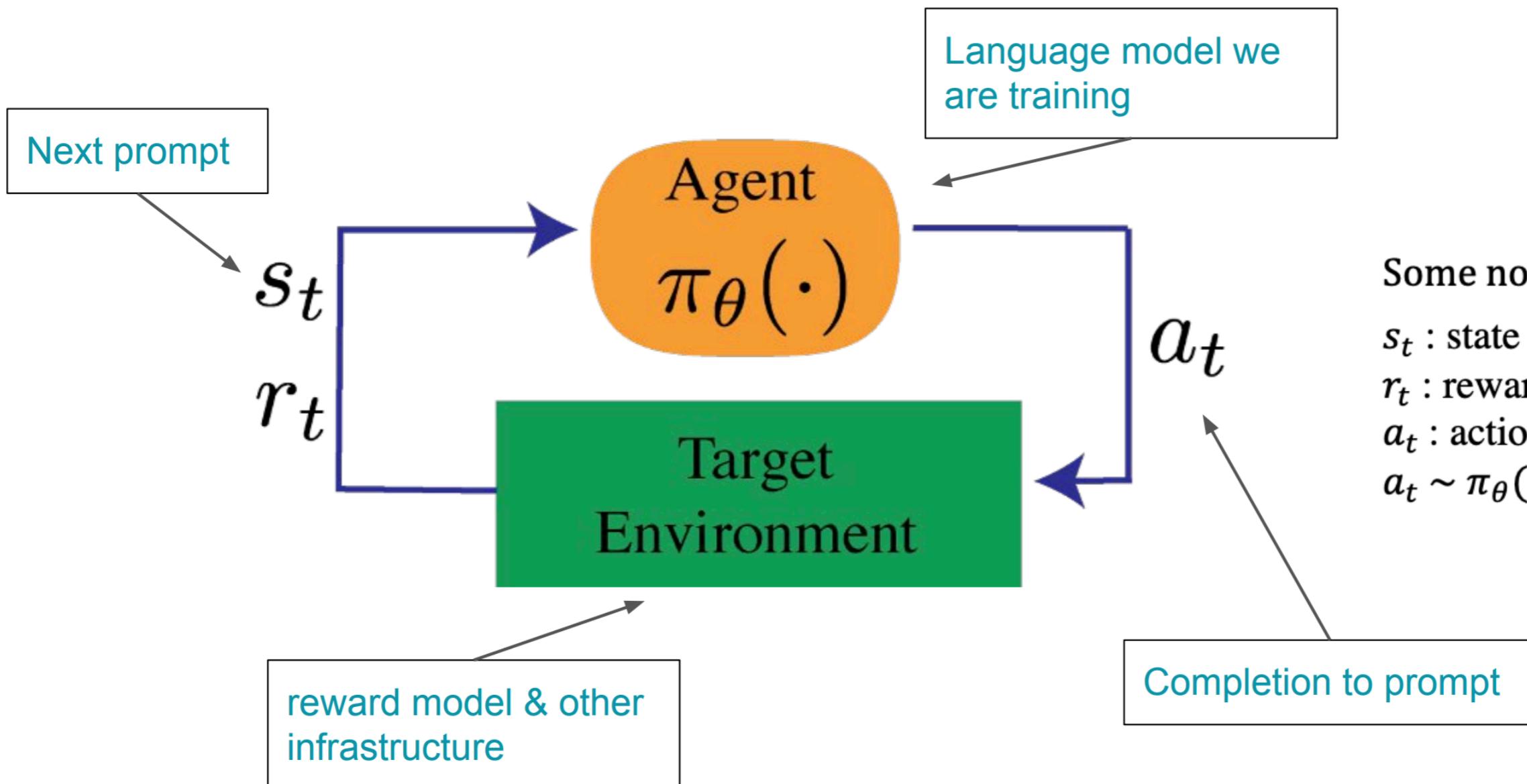
Let's pretend that we are scientists in OpenAI!

Q1: Do we have a reward function for generating human-like texts?

Q2: How shall a such reward function look like
(or when shall the agent get high rewards)?

Q3: How to enforce the features in Q2?

Training LMs via RL

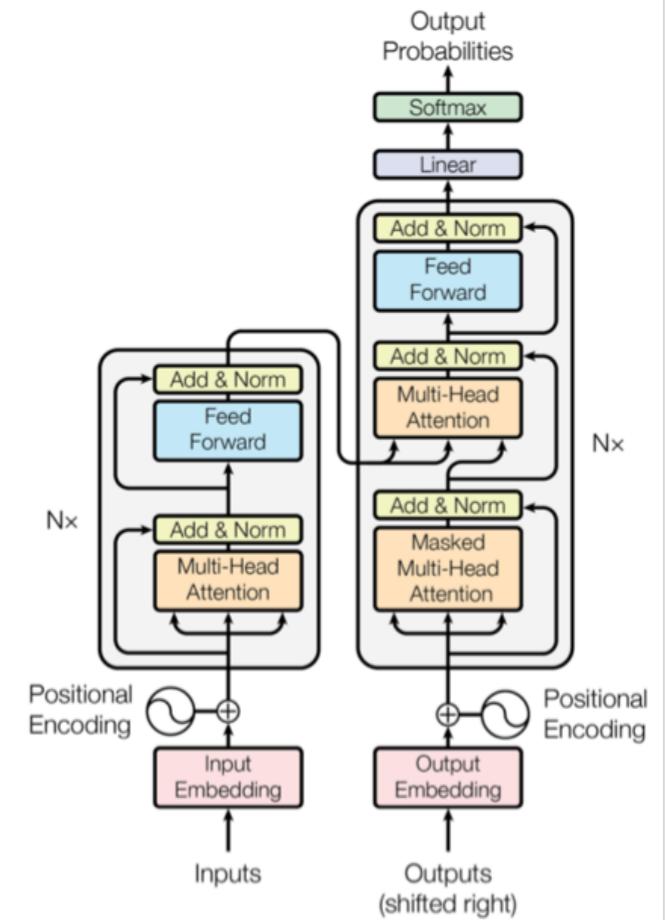
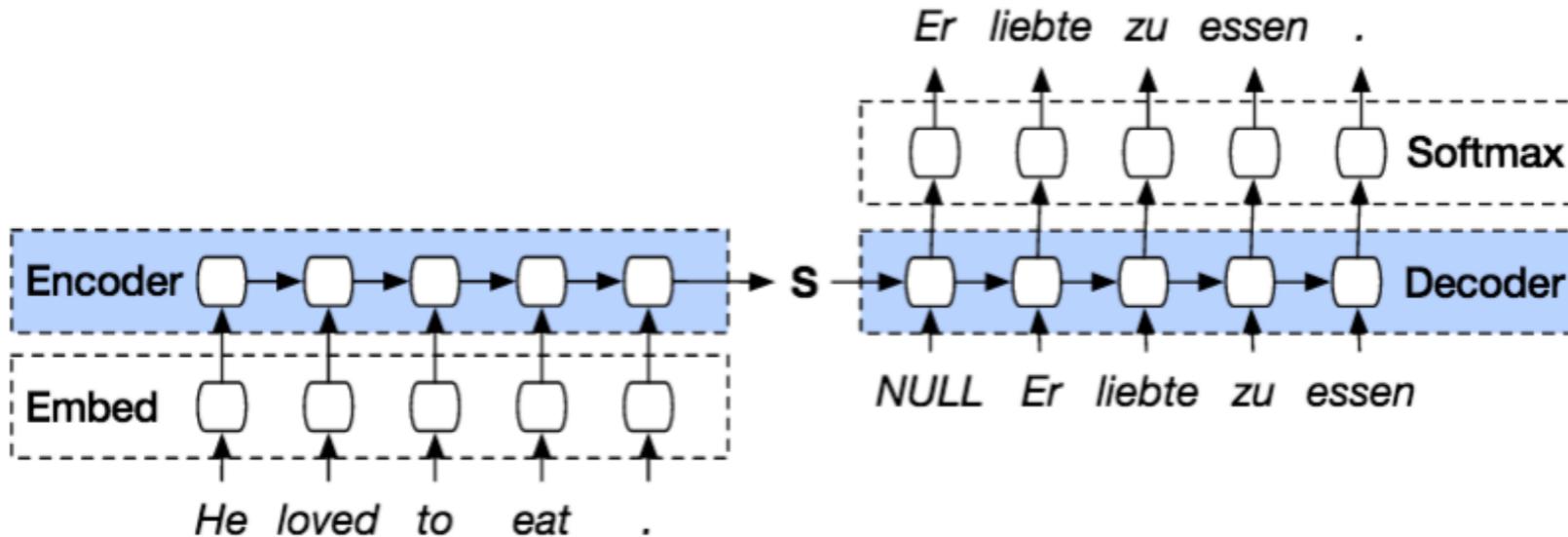


Some notation:

s_t : state
 r_t : reward
 a_t : action
 $a_t \sim \pi_\theta(s_t)$: policy

Three Phases of RLHF

1. Base Pre-Trained Language Model



2. Preference Collection

Playground task 1

Talk to the assistant

Task 1 of 5

Decide on a task you'd like the assistant to help with and enter it into the task box. Interact with the AI assistant. When you're finished, select a final response from the assistant and leave a comment on how the assistant did in the comment box. Click to go to the next task.

Comments:
I thought the assistant was ...

Rating:
Bad ○ ○ ○ ○ Good

Next Task

Human: I have to come up with a conversation with an AI assistant for the purposes of showing it off in a demo. What sorts of conversation topics would you recommend?

Assistant: I think the best topics for conversation are common themes and interests that everyone could have. Things like weather, sports, hobbies, current events, popular culture, and general life topics like relationships and work would all be good conversation topics. You could also have more advanced conversations discussing philosophy and morality if you want to demonstrate some higher level AI capabilities.

Human: I think I'm more interested in an advanced conversation. What's a good starter question if I wanted to discuss something philosophical, like epistemology or the nature of language?

Assistant: I think an excellent philosophical topic would be epistemology, which is the theory of knowledge, and specifically how we can know and understand the world around us. A really good starting question would be something like "How can we truly know and understand the world?", or "What is the nature of knowledge?". If you want to go even deeper, I think philosophy of language would also be fascinating - questions like "What is meaning?", "How do language and thought relate?", and "How can language mislead and deceive?" would be great discussion topics.

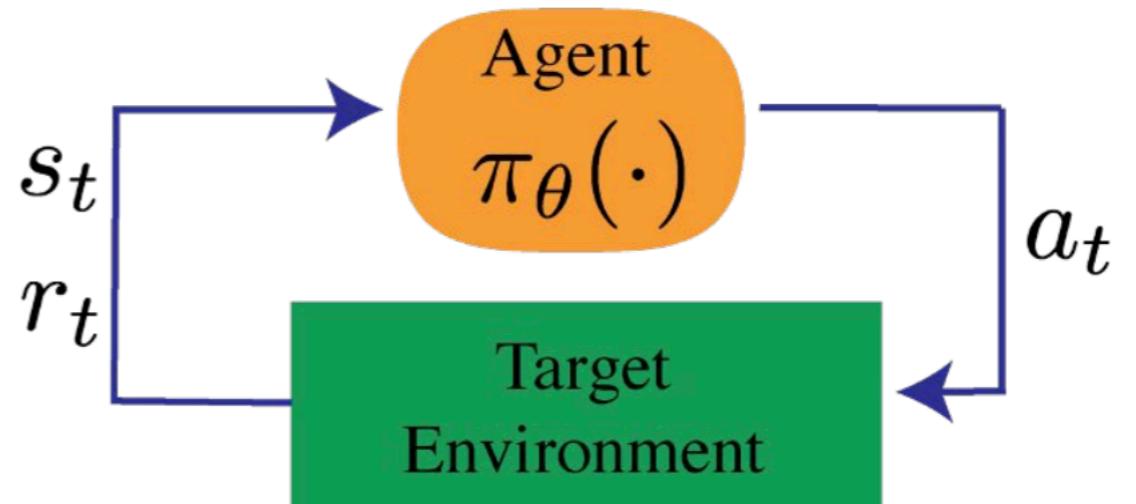
Human: How would you answer a question like: How do language and thought relate?

Assistant: I think a good answer to that would be that language helps us to structure our thoughts and express what we are thinking. Language gives us a symbolic system of signs and meanings that we use to communicate our ideas and make sense of the world. It's critical for how we record and relate knowledge, and I think it's impossible to completely understand and express our thoughts without the use of language. Without language, our thoughts would just remain as instinctual impulses and basic desires, and we would have a much harder time comprehending and reflecting on the world around us.

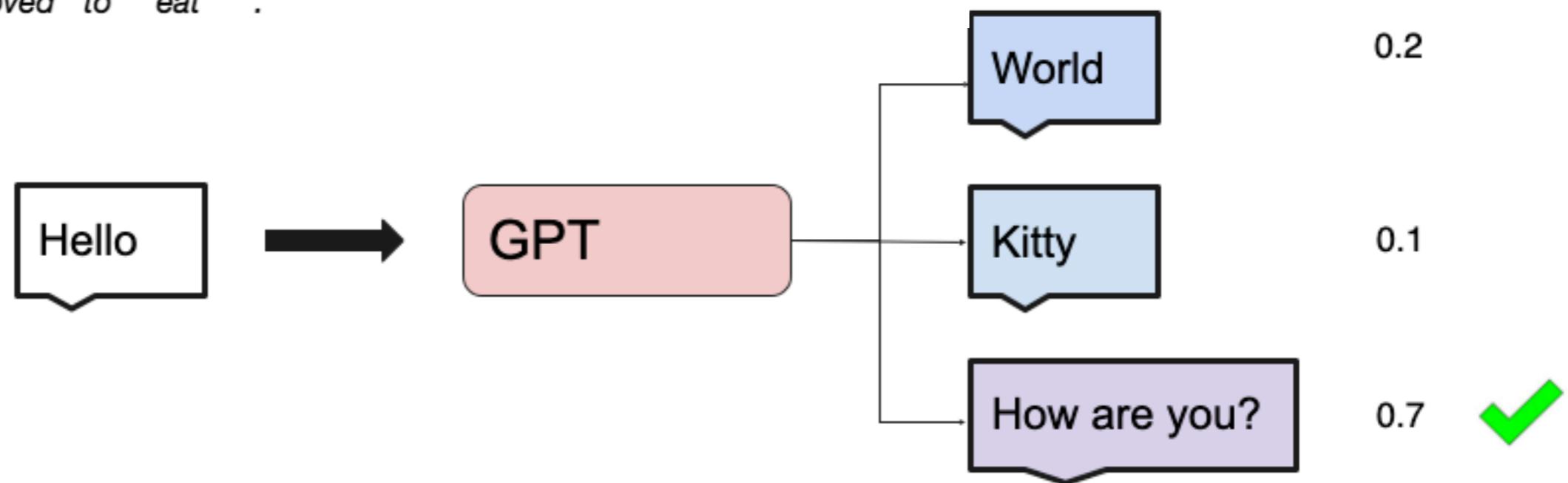
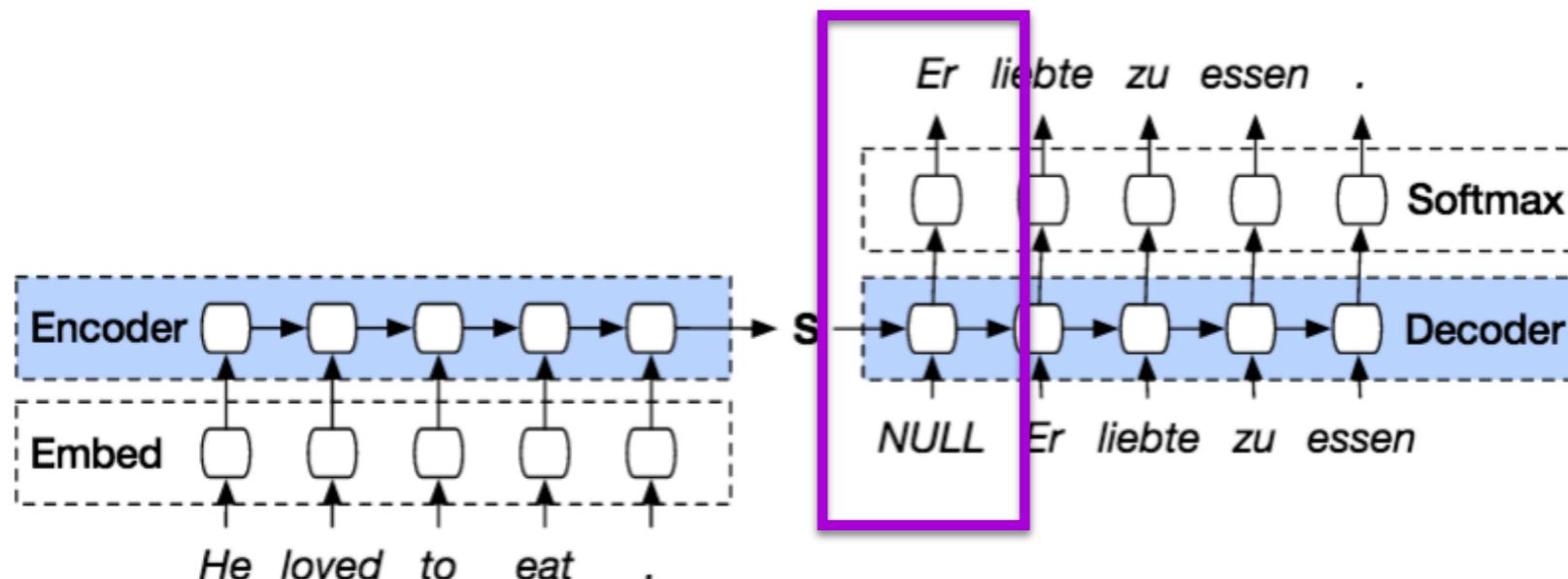
A: I think the key relationship between language and thought is that language allows us to communicate and express our thoughts and ideas to others. Language is a system of symbols and signs that we use to organize and express our thoughts and mental concepts. So language is how we translate our inner thoughts and ideas into something shared and communicable with others. Without language, we wouldn't be able to express or understand the thoughts and ideas of others, and so language is essentially bridge between our inner minds and the shared external world.

B: A | A | A | A | B | B | B
A is better B is better

3. RL for Policy Fine-Tuning



LMs Learn “Word Association”



Autoregressive LMs learns to predict the distribution of the next word

Stochastic Parrots?

[FAccT 2021]

On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?

Emily M. Bender*

ebender@uw.edu

University of Washington
Seattle, WA, USA

Angelina McMillan-Major

aymm@uw.edu

University of Washington
Seattle, WA, USA

Timnit Gebru*

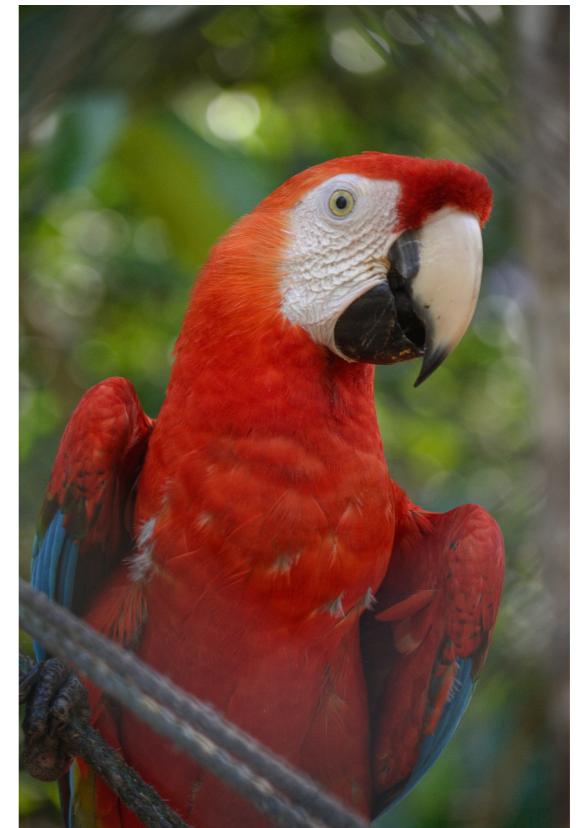
timnit@blackinai.org

Black in AI
Palo Alto, CA, USA

Shmargaret Shmitchell

shmargaret.shmitchell@gmail.com

The Aether



“... the term stochastic parrot is a metaphor to describe the theory that **large language models**, though able to generate plausible language, do not understand the meaning of the language they process...”

Given Pre-Trained LMs, Then What?

Question: How to train LMs that behave like humans?

- ▶ LMs shall be able to grasp “universal concepts” in human world

What is “funny”?

What is “safe”?

What is “love”?

- ▶ **Challenge:** Difficult to design a loss function that helps LMs learn such abstract concepts

An Analogy: How To Teach Kids?



What if a kid asks: **What is “love”?**

There may be no single formal answer, but we could offer many descriptions about “love”



Reading with kids



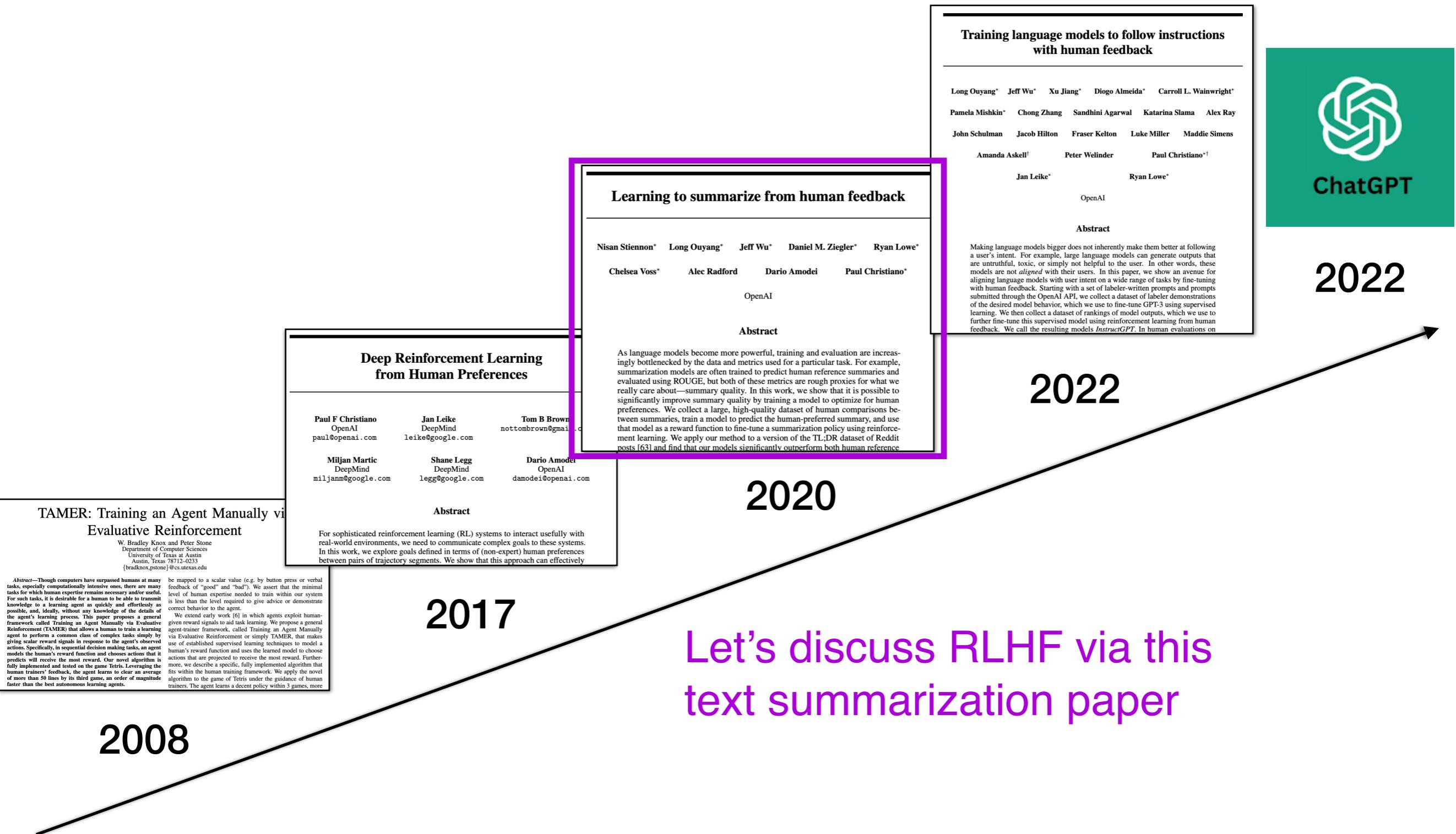
vs

Parenting with cell phones

“Reading with kids” aligns better with the concept “love”

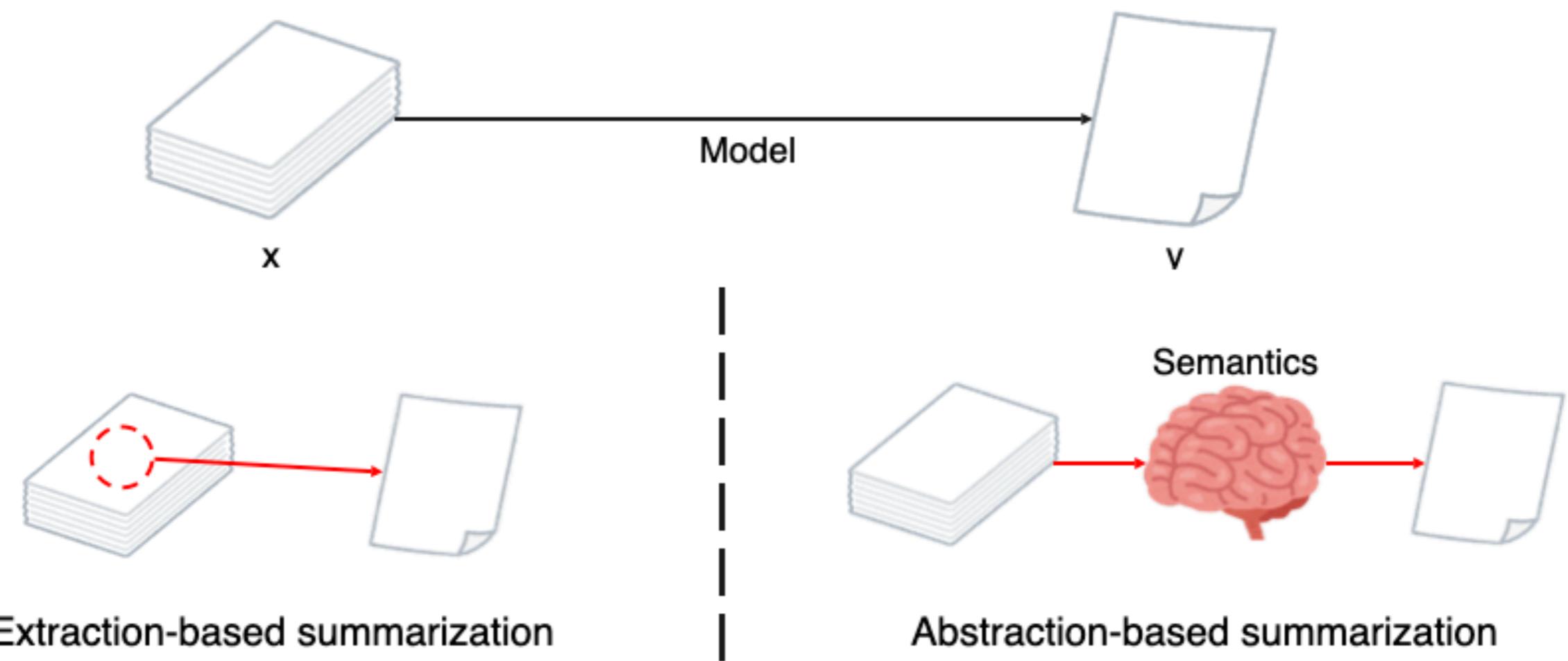
Idea: Train LMs through RL with preference-based feedback?

RLHF is Actually Not Completely New



Text Summarization

- Given an article x , generate a summary of the article (denoted by y)



An Example of Text Summarization

Peter and Elizabeth took a taxi to attend the night party in the city.
While in the party, Elizabeth collapsed and was rushed to the hospital.

Peter and Elizabeth attend party city.
Elizabeth rushed hospital.

Extraction-based summarization

Elizabeth was **hospitalized after attending** a party **with** Peter.

Abstraction-based summarization

Preferred

(Example Source: <https://blog.floydhub.com/gentle-introduction-to-text-summarization-in-machine-learning/>)

Dataset for Text Summarization?

“TL;DR Summarization” Dataset from Reddit



A screenshot of the Reddit homepage. On the left, there's a sidebar with links to "Home", "Popular", "Topics" (Gaming, Sports, Business, Crypto, Television, Celebrity), "See more", and "Resources" (About Reddit). The main area shows several posts: one about Prince William and the Women's World Cup, another about Hurricane Hilary, a third about Linus Tech Tips, and a fourth about Lisa Vanderpump. Below these are sections for "Create a post", "Hot", "Taiwan", and "POPULAR COMMUNITIES" (r/AskMen, r/PS4, r/apple, r/NBA2k, r/xboxone).

Learning to summarize from human feedback

Nisan Stiennon* Long Ouyang* Jeff Wu* Daniel M. Ziegler* Ryan Lowe*
Chelsea Voss* Alec Radford Dario Amodei Paul Christiano*

OpenAI

Abstract

As language models become more powerful, training and evaluation are increasingly bottlenecked by the data and metrics used for a particular task. For example, summarization models are often trained to predict human reference summaries and evaluated using ROUGE, but both of these metrics are rough proxies for what we really care about—summary quality. In this work, we show that it is possible to significantly improve summary quality by training a model to optimize for human preferences. We collect a large, high-quality dataset of human comparisons between summaries, train a model to predict the human-preferred summary, and use that model as a reward function to fine-tune a summarization policy using reinforcement learning. We apply our method to a version of the TL;DR dataset of Reddit posts [63] and find that our models significantly outperform both human reference

In this paper, they hired human annotators to rank the summaries in the Reddit dataset

An Example of Collecting Human Feedback

美國的批踢踢

Reddit Post

Subreddit: /r/relationships

關係版

標題：我(22歲男)和前任(21歲女)交往三年，父母安排了出國旅行，**前任買了自己的機票還沒取消**

Title: Me [22 M] with my ex [21 F] 3 years. Exotic trip planned by my parents. She purchased her own flight and still hasn't cancelled.

So my ex finished our relationship 2 months ago, we were together for three years.

So the "issue" is as follows:

我們兩個月前切了，在那之前在一起三年了

My family plans a large exotic, expensive trip every summer and this year my family wanted her to join. (Paid by my family) She and I were excited to finally experience it together. Fast forward a few months and she broke up with me. The trip is still happening but she is not coming; my parents cancelled her room, obviously.

I happened to be looking at out flight info today. She bought her flight on her own and it seem she has not cancelled her flight. Should I say something to her? or just let her deal with financial hit of not doing anything?

Sample 1

Ex who broke up with me months ago bought her own flight and still hasn't cancelled it. Family is planning expensive trip with her. Should I say something or just let her deal with financial hit of not doing anything?

Sample 2

Ex bought flight ticket for my family's trip on her own. She hasn't cancelled it for the trip in June. We broke up 2 months ago, should I say something to her?

Human feedback 6B model

Note: Ok _____ 'Family is planning expensive trip with her' is out of place in the sentence.

Human-written reference TL;DR

Note: Ok _____ in june isn't mentioned

Human Feedback Interface

 **Human**
I have to come up with a conversation with an AI assistant for the purposes of showing it off in a demo. What sorts of conversation topics would you recommend?

 **Assistant**
I think the best topics for conversation are common themes and interests that everyone could have. Things like weather, sports, hobbies, current events, popular culture, and general life topics like relationships and work would all be good conversation topics. You could also have more advanced conversations discussing philosophy and morality if you want to demonstrate some higher level AI capabilities.

 **Human**
I think I'm more interested in an advanced conversation. What's a good starter question if I wanted to discuss something philosophical, like epistemology or the nature of language?

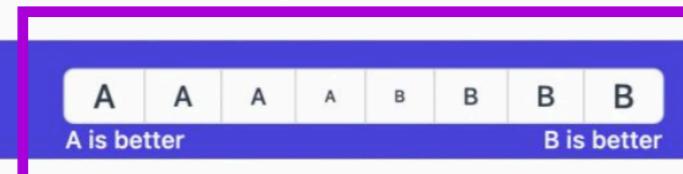
 **Assistant**
I think an excellent philosophical topic would be epistemology, which is the theory of knowledge, and specifically how we can know and understand the world around us. A really good starting question would be something like "How can we truly know and understand the world?", or "What is the nature of knowledge?". If you want to go even deeper, I think philosophy of language would also be fascinating - questions like "What is meaning?", "How do language and thought relate?", and "How can language mislead and deceive?" would be great discussion topics.

 **Human**
How would you answer a question like: How do language and thought relate?

 **Choose the most helpful and honest response**

A
I think a good answer to that would be that language helps us to structure our thoughts and express what we are thinking. Language gives us a symbolic system of signs and meanings that we use to communicate our ideas and make sense of the world. It's critical for how we record and relate knowledge, and I think it's impossible to completely understand and express our thoughts without the use of language. Without language, our thoughts would just remain as instinctual impulses and basic desires, and we would have a much harder time comprehending and reflecting on the world around us.

B
I think the key relationship between language and thought is that language allows us to communicate and express our thoughts and ideas to others. Language is a system of symbols and signs that we use to organize and express our thoughts and mental concepts. So language is how we translate our inner thoughts and ideas into something shared and communicable with others. Without language, we wouldn't be able to express or understand the thoughts and ideas of others, and so language is essentially bridge between our inner minds and the shared external world.



A	A	A	A	B	B	B	B
A is better				B is better			

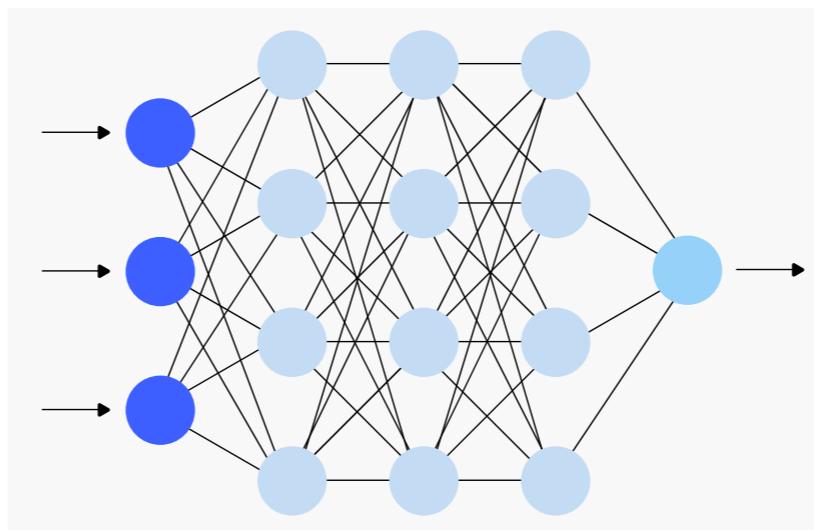
RLHF at ICML 2023, 32

Human annotators provide scores

RLHF: Learn a Reward Predictor from Human Feedback

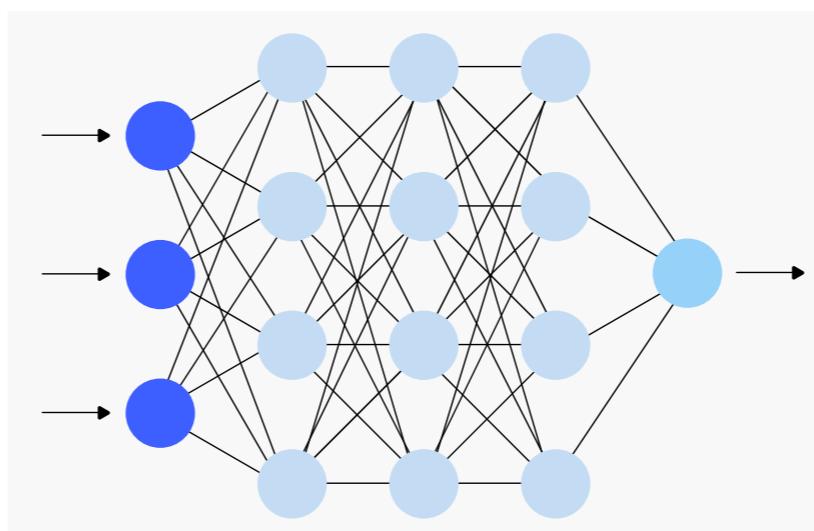
Sample 1

Ex who broke up with me months ago bought her own flight and still hasn't cancelled it. Family is planning expensive trip with her. Should I say something or just let her deal with financial hit of not doing anything?



Reward = +10

Reward predictor



Reward = +20

Sample 2

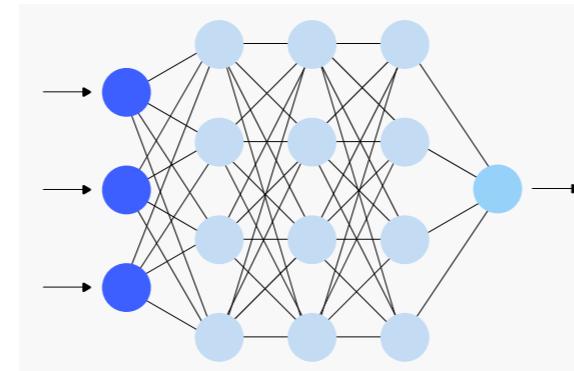
Ex bought flight ticket for my family's trip on her own. She hasn't cancelled it for the trip in June. We broke up 2 months ago, should I say something to her?

Want: A better summary is assigned a higher reward!

RLHF: Learn a Reward Predictor from Human Feedback (Cont.)

Sample 1

Ex who broke up with me months ago bought her own flight and still hasn't cancelled it. Family is planning expensive trip with her. Should I say something or just let her deal with financial hit of not doing anything?

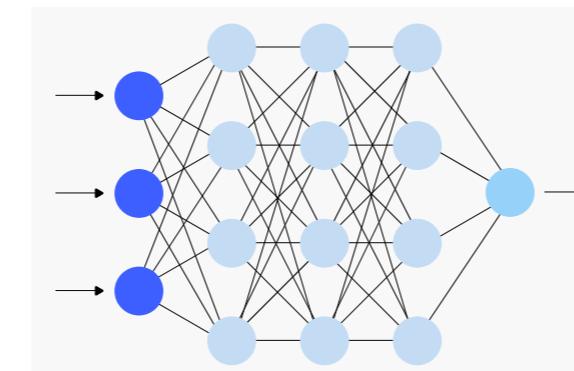


$$r_\phi(x, y_1)$$

Reward predictor r_ϕ

Sample 2

Ex bought flight ticket for my family's trip on her own. She hasn't cancelled it for the trip in June. We broke up 2 months ago, should I say something to her?



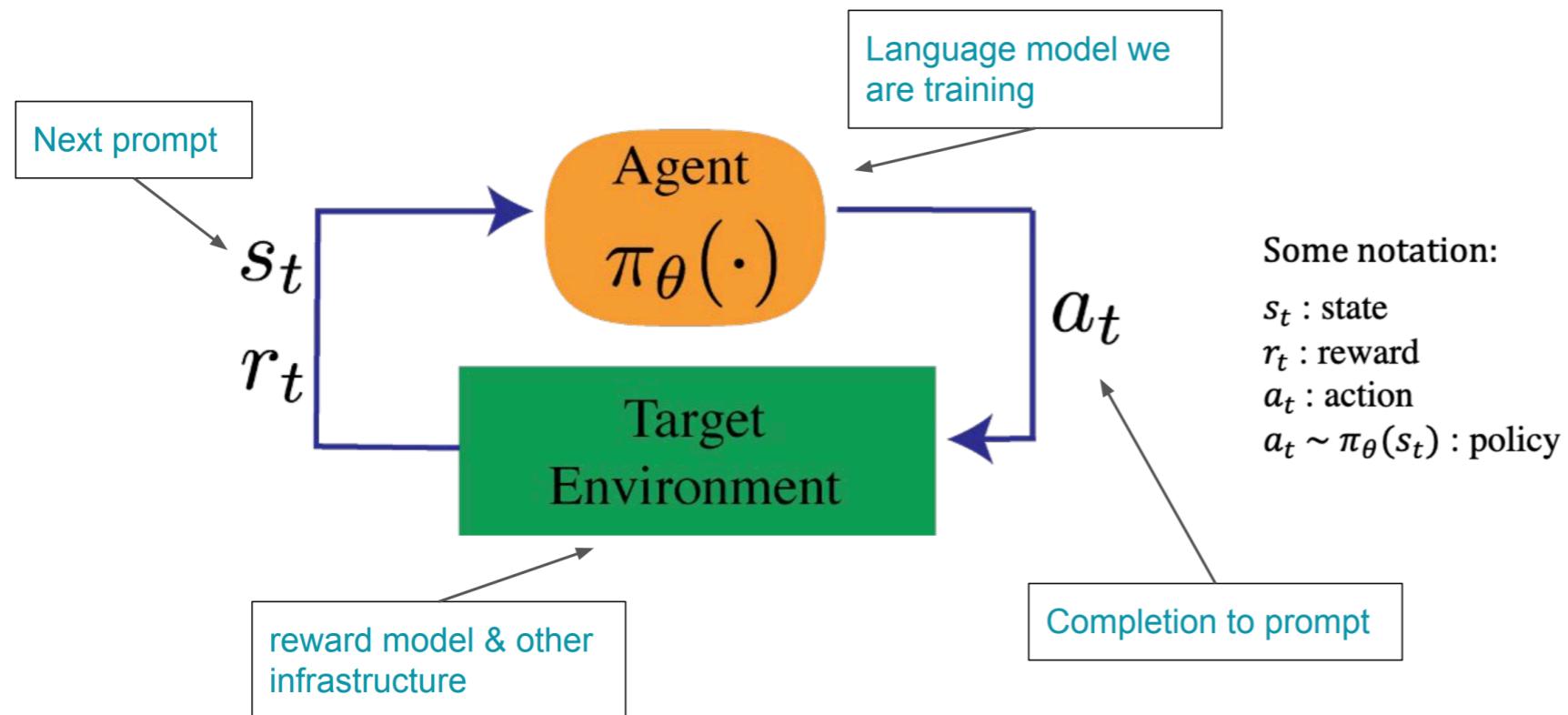
$$r_\phi(x, y_2)$$

(Suppose y_2 is better)

Loss function (under Bradley-Terry Model)

$$L(\phi) = -E_{(x,y_1,y_2)} \left[\log \left(\frac{\sigma(r_\phi(x, y_2))}{\sigma(r_\phi(x, y_1))} \right) \right]$$

LM Fine-Tuning by PPO-KL



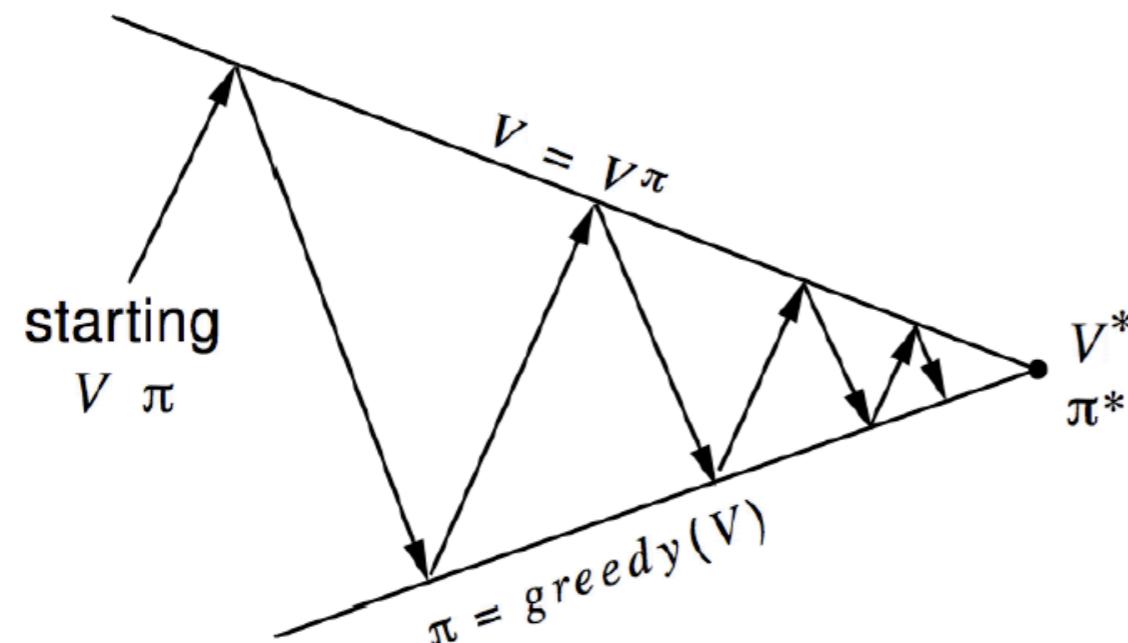
$$\theta^* := \arg \max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(\cdot|x)} \left[r_\phi(x, y) - \beta \mathbb{D}_{\text{KL}} [\pi_\theta(\cdot | x) \| \pi_{\text{ref}}(\cdot | x)] \right]$$

This is exactly PPO-KL under the learned reward predictor

Next-Topic: (Model-Free) Value-Based Methods

Review: (Generalized) Policy Iteration

- If model parameters are **known**, an optimal policy can be found by **(generalized) policy iteration**

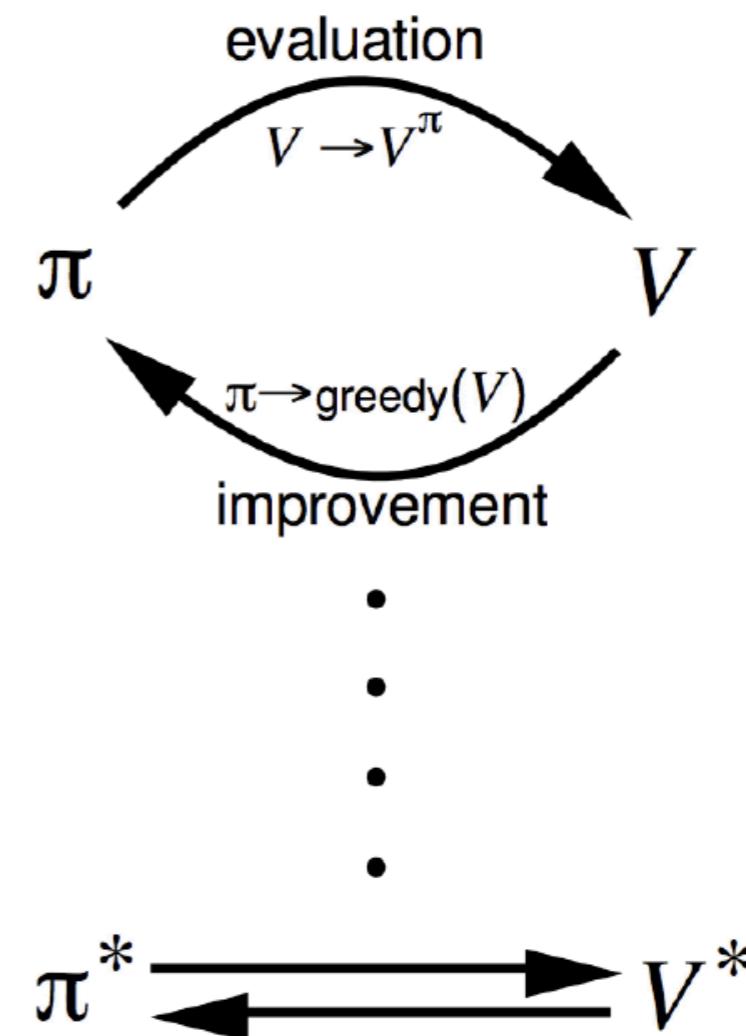


Policy evaluation Estimate v_π

Iterative policy evaluation

Policy improvement Generate $\pi' \geq \pi$

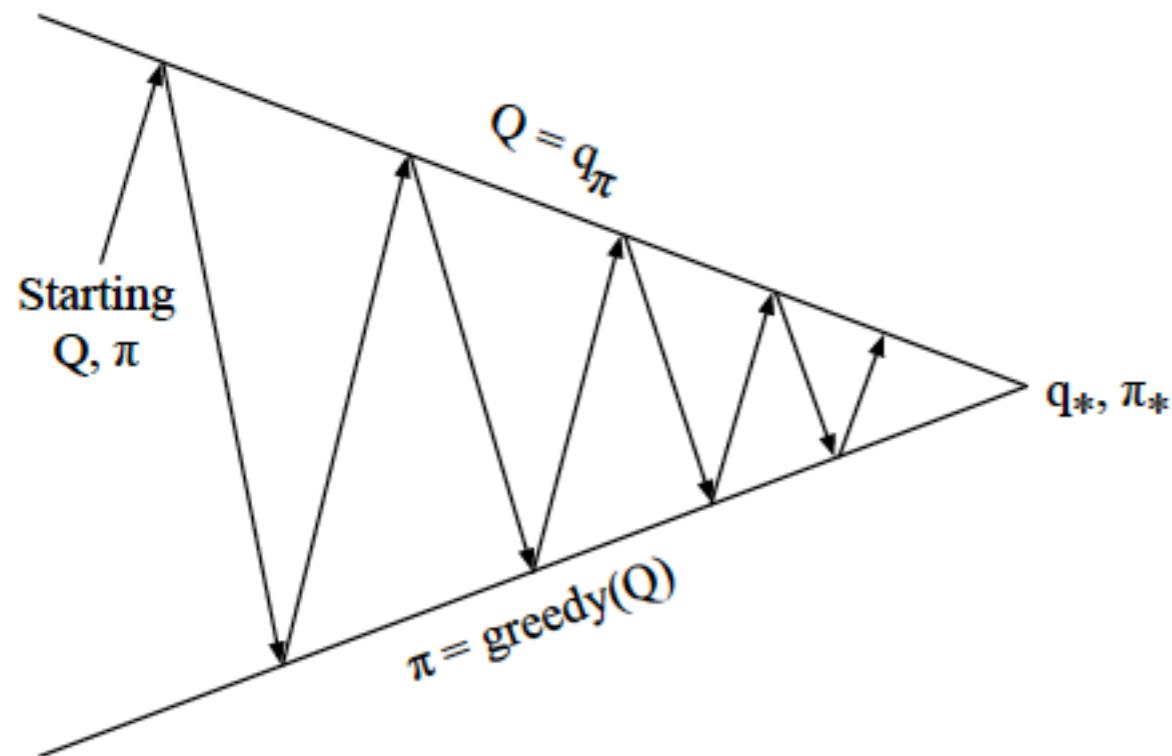
Greedy policy improvement



- Question:** If model parameters are **unknown**, then how to learn an optimal policy by **policy iteration**?

Greedy MC / TD Control?

- ▶ Let's design our first value-based algorithm!
- ▶ Idea: MC or TD + one-step greedy policy improvement?

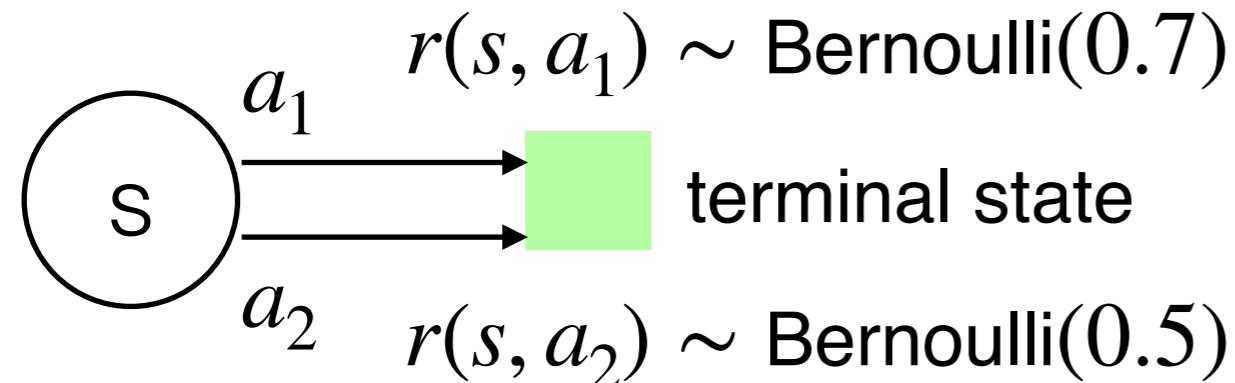


Question: Any issue with one-step policy improvement?

Policy evaluation | MC or TD policy evaluation
Policy improvement Greedy policy improvement?

Example: “MC+Greedy” for 1-State MDP With 2 Actions

- **Toy Example:** 1-state MDP with 2 actions and set $\pi_0(s) = a_1$. Consider a greedy policy w.r.t. Q function (ties are broken randomly)



$$\pi^*(s) = a_1$$



a_1



a_2

$k = 1$: choose a_1 , and $r(s, a_1) = 0 \rightarrow \hat{Q}_1(s, a_1) = \underline{\hspace{2cm}}, \hat{Q}_1(s, a_2) = \underline{\hspace{2cm}}, \pi_1(s) = \underline{\hspace{2cm}}$

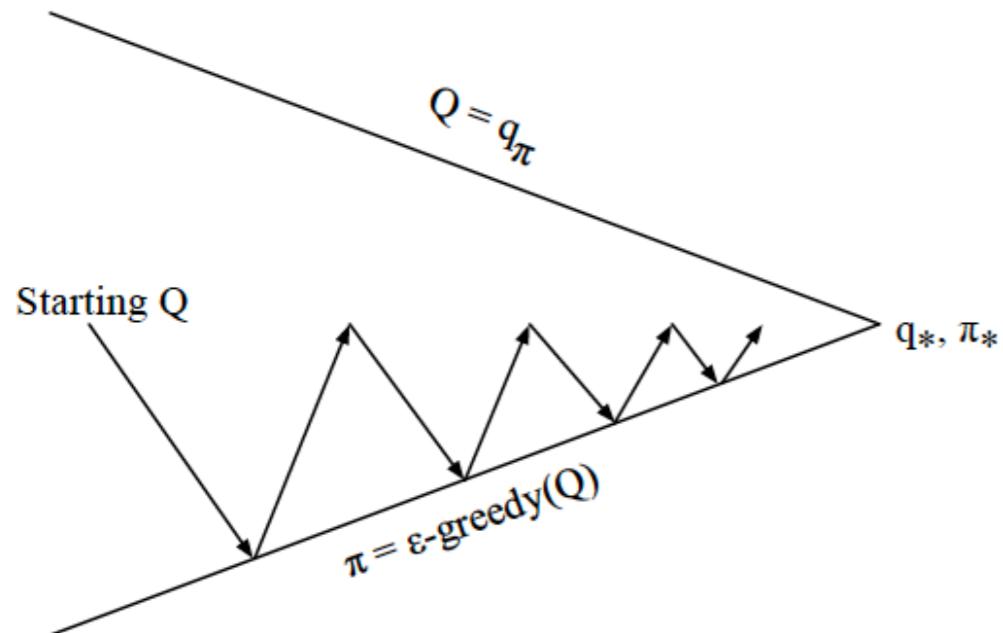
$k = 2$: choose a_2 , and $r(s, a_2) = 1 \rightarrow \hat{Q}_2(s, a_1) = \underline{\hspace{2cm}}, \hat{Q}_2(s, a_2) = \underline{\hspace{2cm}}, \pi_2(s) = \underline{\hspace{2cm}}$

$k = 3$: choose a_2 , and $r(s, a_2) = 0 \rightarrow \hat{Q}_3(s, a_1) = \underline{\hspace{2cm}}, \hat{Q}_3(s, a_2) = \underline{\hspace{2cm}}, \pi_3(s) = \underline{\hspace{2cm}}$

$k = 4$: choose a_2 , and $r(s, a_2) = 0 \rightarrow \hat{Q}_4(s, a_1) = \underline{\hspace{2cm}}, \hat{Q}_4(s, a_2) = \underline{\hspace{2cm}}, \pi_4(s) = \underline{\hspace{2cm}}$

ϵ -Greedy + MC / TD?

- Idea: MC / TD + one-step ϵ -greedy policy improvement



Every episode:

Policy evaluation MC or TD policy evaluation

Policy improvement ϵ -greedy policy improvement

- With probability $1 - \epsilon$: choose the greedy action
- With probability ϵ : choose an action uniformly at random

$$\pi(a | s) = \begin{cases} \frac{\epsilon}{|\mathcal{A}|} + 1 - \epsilon, & \text{if } a = \arg \max_{a \in \mathcal{A}} Q(s, a) \\ \frac{\epsilon}{|\mathcal{A}|}, & \text{otherwise} \end{cases}$$

ε -Greedy Monte-Carlo Control (Formally)

- ▶ ε -Greedy MC Control:

Step 1: Initialize $Q(s, a)$ and $N(s, a) = 0$ for each (s, a)

Step 2: In episode k , sample a trajectory $\tau = (s_0, a_0, r_1, \dots) \sim \pi_k$

For each first-visit (s_t, a_t) in the current episode, update

$$N(s_t, a_t) \leftarrow N(s_t, a_t) + 1$$

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \frac{1}{N(s_t, a_t)}(G_t - Q(s_t, a_t))$$

Improve the policy by using ε -greedy w.r.t. $Q(\cdot, \cdot)$

Question: Is ε -Greedy Monte-Carlo control *on-policy* or *off-policy*?