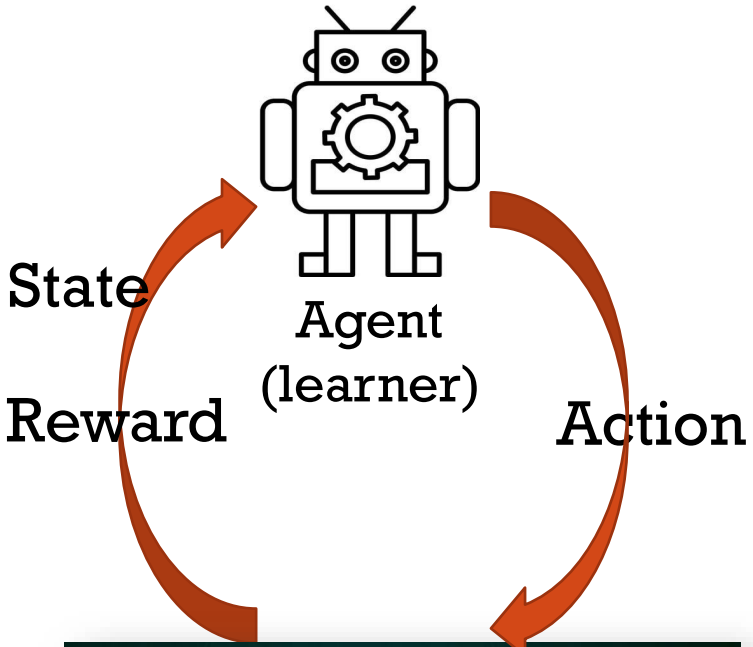
RLHF via Stochastic Zeroth-Order Policy Optimization

Lei Ying

EECS, University of Michigan, Ann Arbor

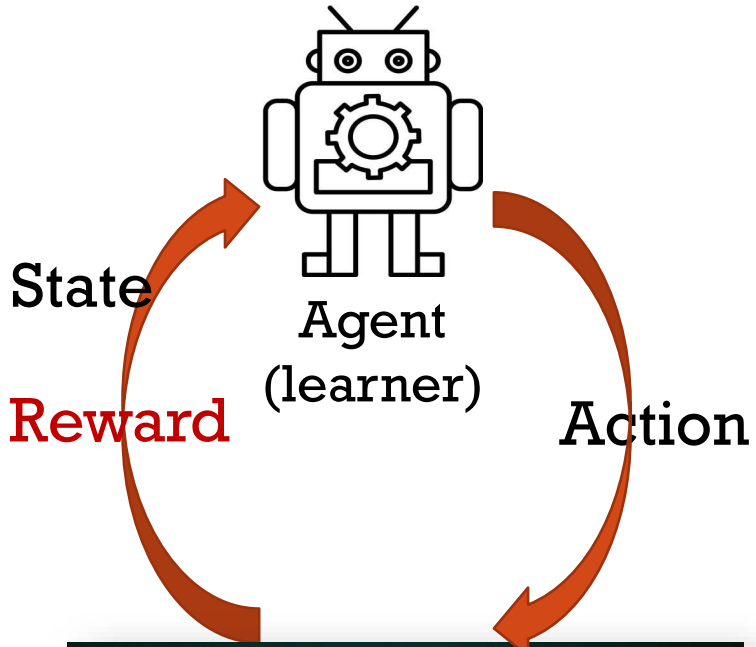
Joint work with Qining Zhang and Deyi Wang

Reinforcement Learning

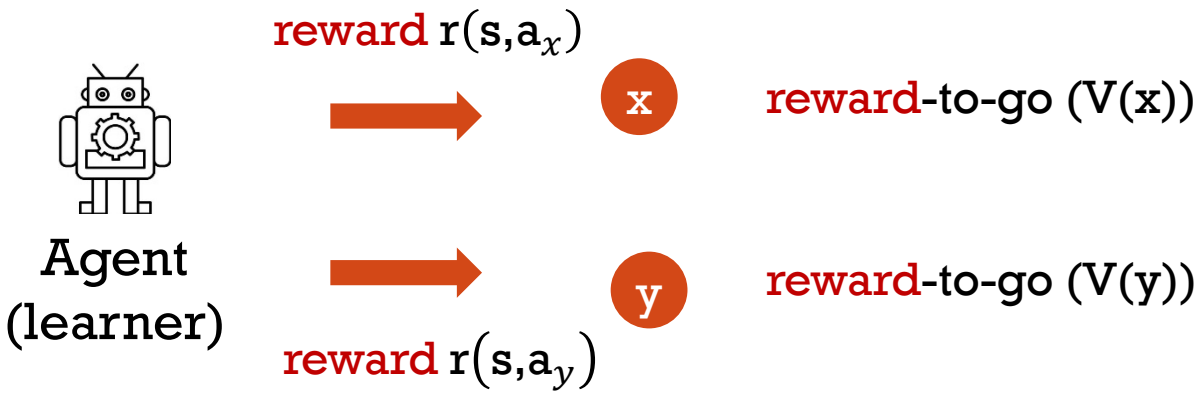


Environment

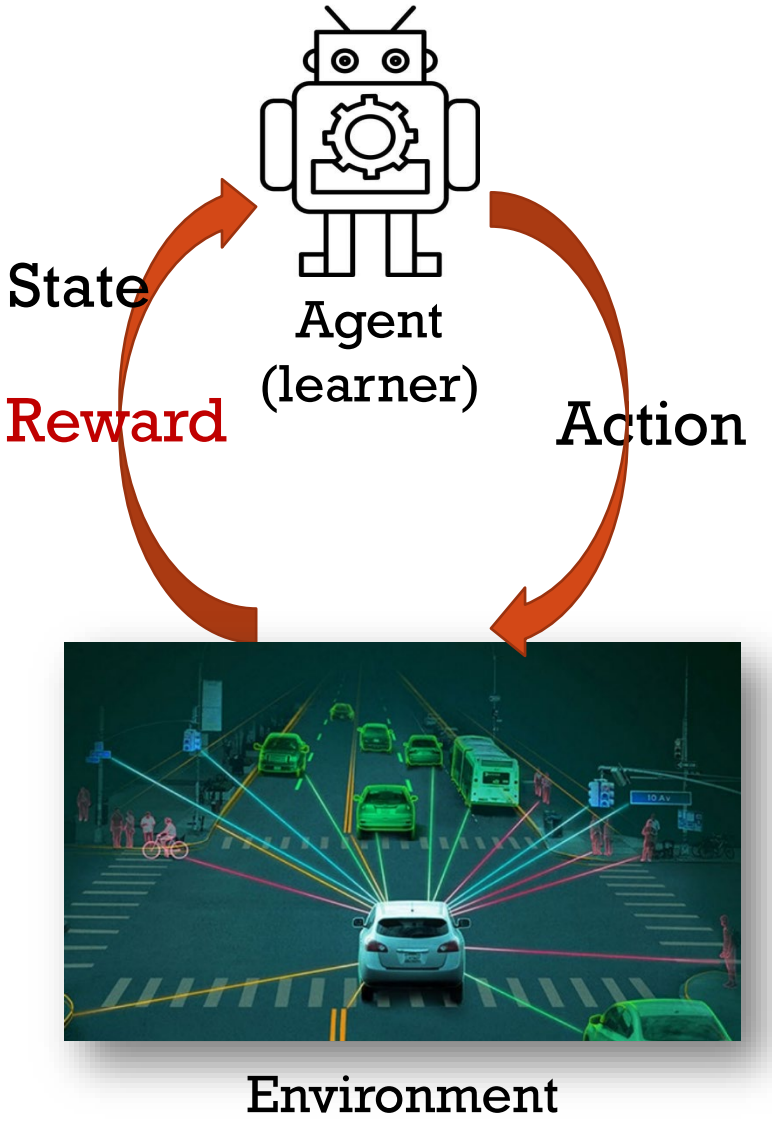
Reinforcement Learning



Value-based: Q-learning, DQN, Alpha-Zero, …



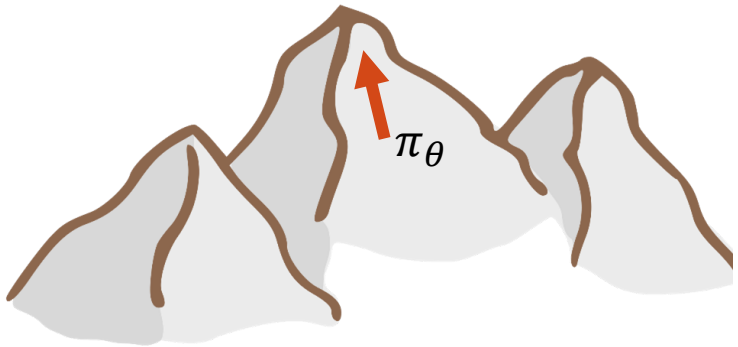
Reinforcement Learning



Value-based: Q-learning, DQN, Alpha-Zero, …

Policy-gradient-based: REINFORCE, PPO, GRPO, …

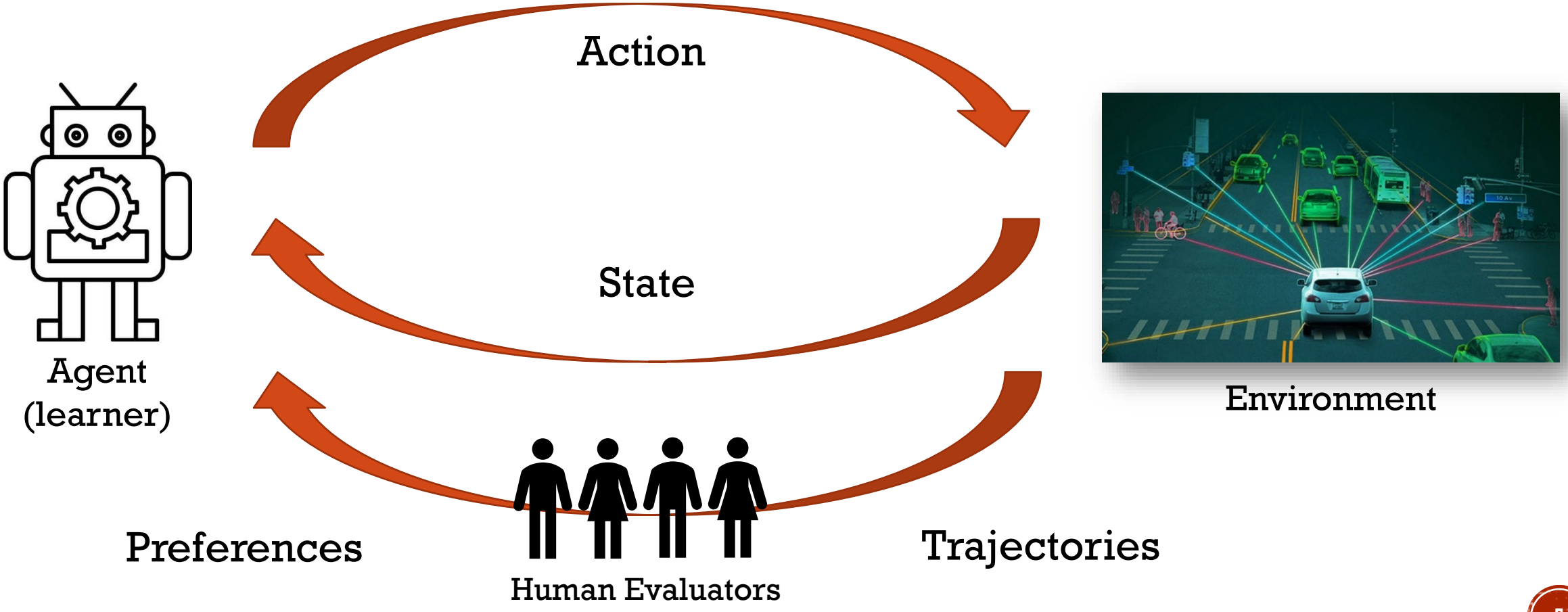
$$\nabla V(\pi_\theta) \approx \sum_t \alpha^t \nabla \log \pi_\theta(\mathbf{a}_t \mid \mathbf{x}_t) \left(\sum_{\tau=t} \alpha^{\tau-t} \mathbf{r}(\mathbf{x}_\tau, \mathbf{a}_\tau) \right)$$



RL from Human Feedback

- Learn from Rewards v.s. Learn from Preferences

- Inverse RL (Russell 1998, Ng&Russell 2000, …)
- Dueling bandits (Yue&Joachims 2000, Ailon, Karnin&Joachims, 2014, …)
- Preference-based RL (Akrour et al. 2011, Cheng et al. 2011, …)



Applications

Large Language Models

ChatGPT

Preferences over Prompt Responses

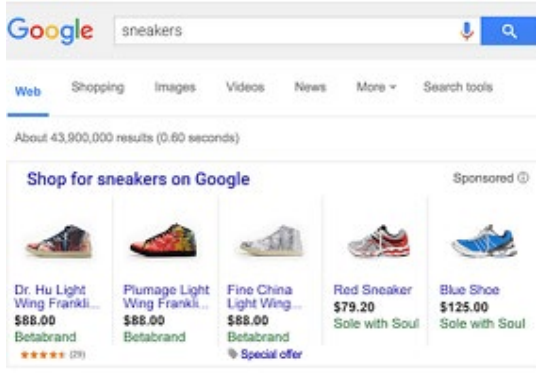
Video Game Bot

Preferences over Gaming Strategies

Robotics

Preferences over Motions

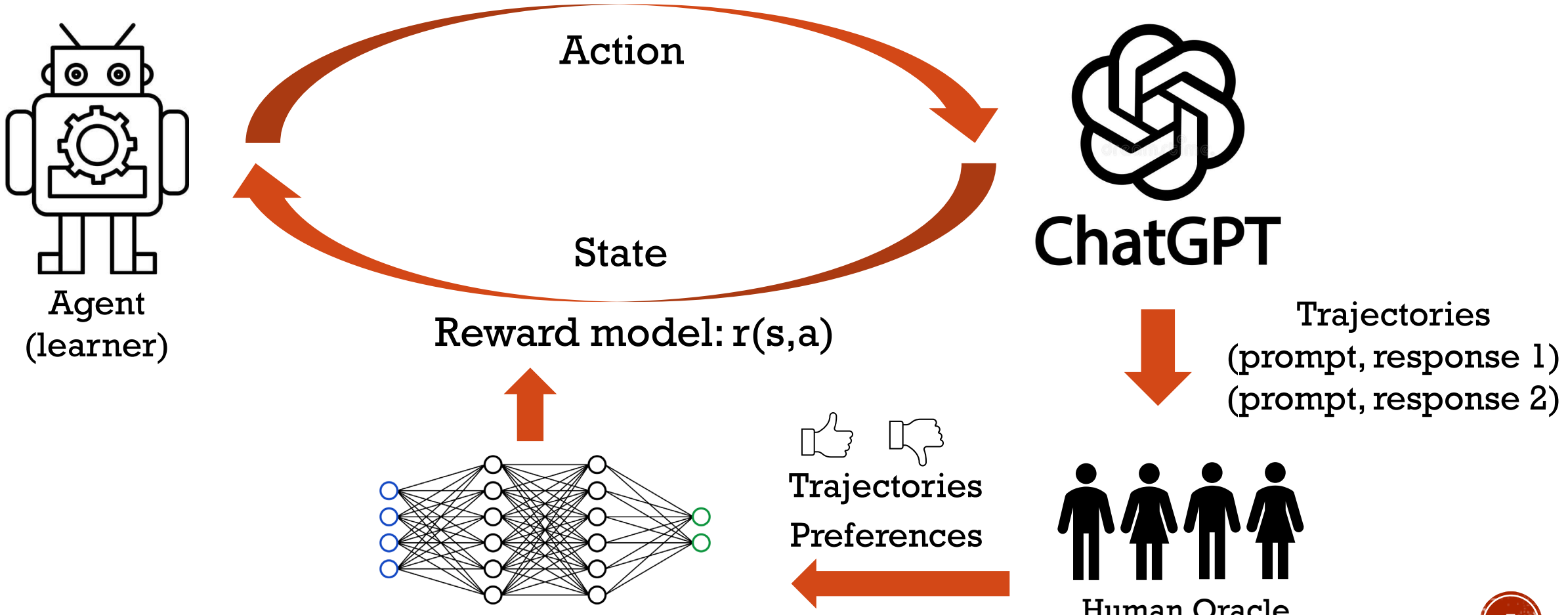
Recommendation Systems



Product Proferences

Method 1: Reward Models (or IRL)

- Learn from Reward v.s. Learn from Preference



RLHF: InstructGPT (Ouyang et al., 2022)

Reinforcement Learning

- Learn from rewards

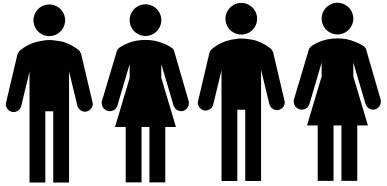
reward-hacked motorboat (**Coast Runners**)
OpenAI blog: Faulty reward functions in the wild, 2016

Method 2: Direct Preference Optimization

- DPO (Rafailov et al. 2023)

Limitations

- ❑ Highly specialized
- ❑ Limited to bandits (Rafailov et al. 2023) or deterministic MDPs (Rafailov et al. 2024)



(prompt, response) pairs

$(x, y_1), (x, y_2)$

reward

optimal policy

The Bradley-Terry model

$$\mathbb{P}((x, y_1) \succ (x, y_2)) = \left(1 + e^{-(r(x,y_1) - r(x,y_2))}\right)^{-1}$$

KL-regularized bandits

$$\frac{\pi^*(y_1|x)}{\pi^*(y_2|x)} = \frac{\pi_{\text{ref}}(y_1|x)}{\pi_{\text{ref}}(y_2|x)} \exp\left(\frac{1}{\beta}(r(x, y_1) - r(x, y_2))\right)$$

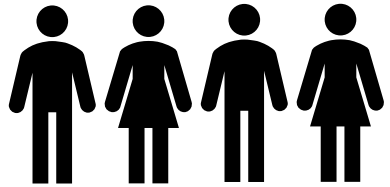
$$\mathbb{P}((x, y_1) \succ (x, y_2)) = \sigma\left(\beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} - \beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)}\right)$$

Method 2: Direct Preference Optimization

- DPO (Rafailov et al. 2023)

Both RLHF and DPO assume the BT model

General, even unknown preference models?



(prompt, response) pairs

$(x, y_1), (x, y_2)$

reward

optimal policy

The Bradley-Terry model

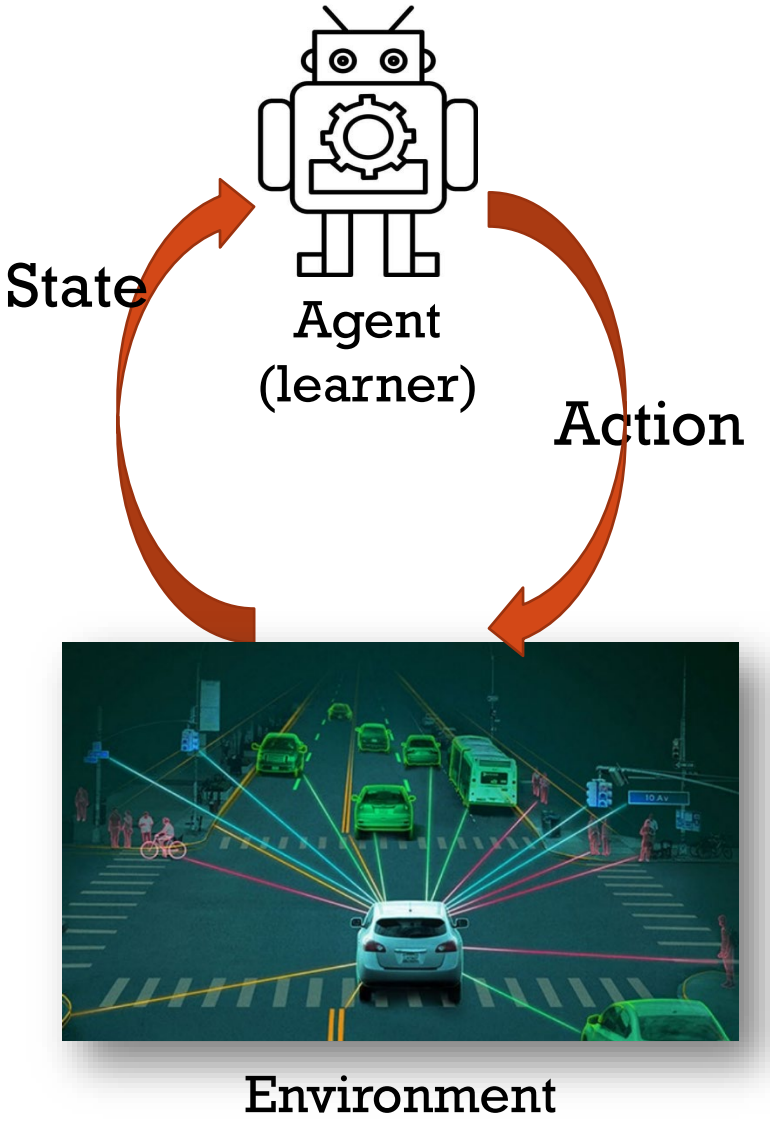
$$\mathbb{P}((x, y_1) \succ (x, y_2)) = \left(1 + e^{-(r(x,y_1) - r(x,y_2))}\right)^{-1}$$

KL-regularized bandits

$$\frac{\pi^*(y_1|x)}{\pi^*(y_2|x)} = \frac{\pi_{\text{ref}}(y_1|x)}{\pi_{\text{ref}}(y_2|x)} \exp\left(\frac{1}{\beta}(r(x, y_1) - r(x, y_2))\right)$$

$$\mathbb{P}((x, y_1) \succ (x, y_2)) = \sigma\left(\beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} - \beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)}\right)$$

Reinforcement Learning: A Third Approach



Value-based: Q-learning, DQN, DDQN, ...

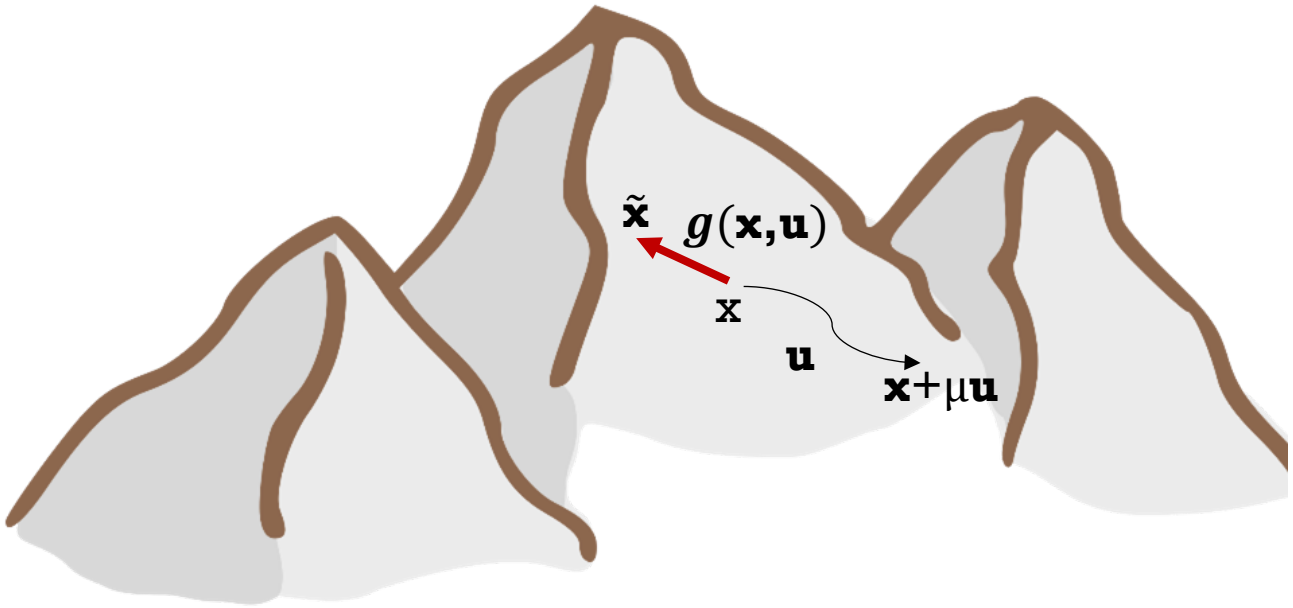
Policy-gradient-based: REINFORCE, PPO, DPG, GRPO .....

Perturbation-based policy optimization: Zeroth-order methods and evolution strategies

Stochastic Zeroth-Order Policy Optimization (SZPO)

- Zeroth-order optimization (two-point method) (Nesterov, 2011; Ghadimi&Lan, 2013; Duchi et al., 2014)

$$\max f(\mathbf{x})$$



- Perturb

$$\mathbf{u} \sim \text{Gaussian}(0, \mathbf{I})$$

- Compare

$$\frac{f(\mathbf{x}+\mu\mathbf{u}) - f(\mathbf{x})}{\mu}$$

- Move

$$\tilde{\mathbf{x}} = \mathbf{x} + \beta \boldsymbol{g}(\mathbf{x},\mathbf{u})$$

$$\boldsymbol{g}(\mathbf{x},\mathbf{u}) = \frac{f(\mathbf{x}+\mu\mathbf{u}) - f(\mathbf{x})}{\mu}\mathbf{u}$$

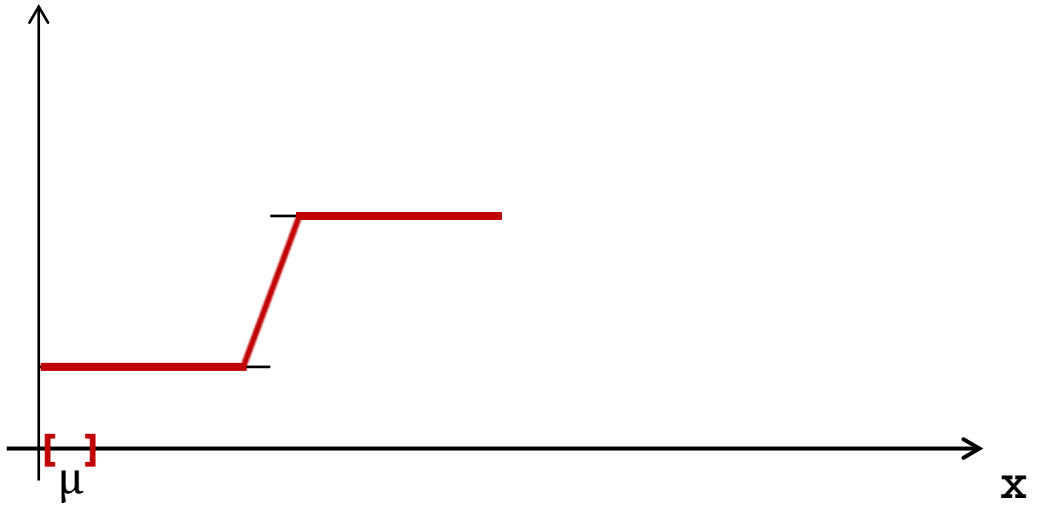
Why does it work?

$$\tilde{\mathbf{x}} = \mathbf{x} + \beta \boldsymbol{g}(\mathbf{x},\mathbf{u})$$
$$\boldsymbol{g}(\mathbf{x},\mathbf{u}) = \frac{f(\mathbf{x}+\mu\mathbf{u}) - f(\mathbf{x})}{\mu}\mathbf{u}$$

- Gradient of something?

Gaussian smoothing $f_\mu(\mathbf{x})=\mathrm{E}[f(\mathbf{x}+\mu\mathbf{u})]$, $\mathbf{u} \sim$ Gaussian$(0, \mathbf{I})$

$$\mathrm{E}[\boldsymbol{g}(\mathbf{x},\mathbf{u})] = \nabla f_\mu(\mathbf{x})$$



Why does it work?

$$\tilde{\mathbf{x}} = \mathbf{x} + \beta \boldsymbol{g}(\mathbf{x},\mathbf{u})$$
$$\boldsymbol{g}(\mathbf{x},\mathbf{u}) = \frac{f(\mathbf{x}+\mu\mathbf{u}) - f(\mathbf{x})}{\mu}\mathbf{u}$$

- Gradient of something?

Smoothed function $f_\mu(\mathbf{x})=\mathrm{E}[f(\mathbf{x}+\mu\mathbf{u})]$, $\mathbf{u} \sim$ Gaussian(0, $\mathbf{I}$)

$$\mathrm{E}[\boldsymbol{g}(\mathbf{x},\mathbf{u})] = \nabla f_\mu(\mathbf{x})$$

- (Proof) Stein's Lemma: Fixing x and sampling $\mathbf{u} \sim$ Gaussian(0, $\mathbf{I}$)

$$\mathrm{E}[\mathbf{u}f(\mathbf{x}+\mu\mathbf{u})] = \mathrm{E}[\mu\nabla f(\mathbf{x}+\mu\mathbf{u})]$$

Why does it work?

$$\tilde{\mathbf{x}} = \mathbf{x} + \beta \boldsymbol{g}(\mathbf{x},\mathbf{u})$$
$$\boldsymbol{g}(\mathbf{x},\mathbf{u}) = \frac{f(\mathbf{x}+\mu\mathbf{u}) - f(\mathbf{x})}{\mu}\mathbf{u}$$

- Gradient of something?

Smoothed function $f_\mu(\mathbf{x})=\mathrm{E}[f(\mathbf{x}+\mu\mathbf{u})]$, $\mathbf{u} \sim$ Gaussian(0, $\mathbf{I}$)

$$\mathrm{E}[\boldsymbol{g}(\mathbf{x},\mathbf{u})] = \nabla f_\mu(\mathbf{x})$$

- (Proof) Stein's Lemma: Fixing x and sampling $\mathbf{u} \sim$ Gaussian(0, $\mathbf{I}$)

$$\mathrm{E}[\mathbf{u} f(\mathbf{x}+\mu\mathbf{u})] = \mu \nabla \mathrm{E}[f(\mathbf{x}+\mu\mathbf{u})]$$

Why does it work?

$$\tilde{\mathbf{x}} = \mathbf{x} + \beta \boldsymbol{g}(\mathbf{x},\mathbf{u})$$
$$\boldsymbol{g}(\mathbf{x},\mathbf{u}) = \frac{f(\mathbf{x}+\mu\mathbf{u}) - f(\mathbf{x})}{\mu}\mathbf{u}$$

- Gradient of something?

Smoothed function $f_\mu(\mathbf{x})=\mathrm{E}[f(\mathbf{x}+\mu\mathbf{u})]$, $\mathbf{u} \sim$ Gaussian$(0, \mathbf{I})$

$$\mathrm{E}[\boldsymbol{g}(\mathbf{x},\mathbf{u})] = \nabla f_\mu(\mathbf{x})$$

- (Proof) Stein's Lemma: Fixing $\mathbf{x}$ and sampling $\mathbf{u} \sim$ Gaussian$(0, \mathbf{I})$

$$\mathrm{E}[\frac{1}{\mu}(\mathbf{u}f(\mathbf{x}+\mu\mathbf{u}) - \mathbf{u}f(\mathbf{x}))] = \mu\nabla\,\mathrm{E}[f(\mathbf{x}+\mu\mathbf{u})]$$

Stochastic Zeroth-Order Policy Optimization (SZPO)

- Zeroth-order optimization

$$\max f(\mathbf{x})$$

  - Perturb

$$\mathbf{u} \sim \text{Gaussian}(0, \mathbf{I})$$

  - Compare

$$\frac{f(\mathbf{x}+\mu\mathbf{u}) - f(\mathbf{x})}{\mu}$$

  - Move

$$\tilde{\mathbf{x}} = \mathbf{x} + \beta \boldsymbol{g}(\mathbf{x},\mathbf{u})$$

$$\boldsymbol{g}(\mathbf{x},\mathbf{u}) = \frac{f(\mathbf{x}+\mu\mathbf{u}) - f(\mathbf{x})}{\mu}\mathbf{u}$$

- Stochastic Zeroth-order policy optimization (SZPO)

$$\max V(\pi_\theta)$$

  - Perturb

$$\mathbf{u} \sim \text{Gaussian}(0, \mathbf{I})$$

  - Compare

$$\frac{V(\pi_{\theta+\mu u}) - V(\pi_\theta)}{\mu}$$

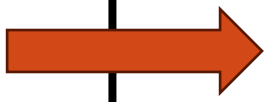
  - Move

$$\tilde{\theta} = \theta + \beta \boldsymbol{g}(\theta,\mathbf{u})$$

$$\boldsymbol{g}(\theta,\mathbf{u}) = \frac{V(\pi_{\theta+\mu\mathbf{u}}) - V(\pi_\theta)}{\mu}\mathbf{u}$$

Stochastic Zeroth-Order Policy Optimization (SZPO)

Learn from human feedback via the link function (e.g., the BT model)



- Stochastic Zeroth-order policy optimization (SZPO)

$$\max V(\pi_\theta)$$

  - Perturb

$$\mathbf{u} \sim \text{Gaussian}(0, \mathbf{I})$$

  - Compare

$$\frac{V(\pi_{\theta+\mu u}) - V(\pi_\theta)}{\mu}$$

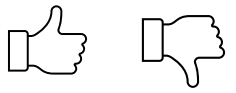
  - Move

$$\tilde{\theta} = \theta + \beta \boldsymbol{g}(\theta, \mathbf{u})$$

$$\boldsymbol{g}(\theta, \mathbf{u}) = \frac{V(\pi_{\theta+\mu\mathbf{u}}) - V(\pi_\theta)}{\mu}\mathbf{u}$$

A pair of trajectories generated by policies π and $\tilde{\pi}$

We have preferences from human feedback

τ_π  $\tau_{\tilde{\pi}}$

We need the value difference

$V(\pi) - V(\tilde{\pi}) = E[r(\tau_\pi)] - E[r(\tau_{\tilde{\pi}})]$

Link function: the Bradley-Terry (BT) model

$$\mathbb{P}(\tau_\pi \succ \tau_{\tilde{\pi}}) = \left(1 + e^{r(\tau_\pi) - r(\tau_{\tilde{\pi}})}\right)^{-1}$$

Unknown link function?

When is SZPO for RLHF Useful?

Result 1: Unknown preference models (link functions)

Qining Zhang and Lei Ying. "Provable Reinforcement Learning from Human Feedback with an Unknown Link Function." *arXiv preprint arXiv:2506.03066* (2025).

- SZPO

$$\max V(\pi_\theta)$$

  - Perturb

$$\mathbf{u} \sim \text{Gaussian}(0, \mathbf{I})$$

  - Compare

$$o = \mathbf{sgn}\left(V(\pi_{\theta+\mu u}) - V(\pi_\theta)\right)$$

  - Move

$$\tilde{\theta} = \theta + \beta \boldsymbol{g}(\theta, \mathbf{u})$$

$$\boldsymbol{g}(\theta, \mathbf{u}) = o\mathbf{u}$$

- Sign-SZPO

$$\max V(\pi_\theta)$$

- Perturb

$$\mathbf{u} \sim \text{Gaussian}(0, \mathbf{I})$$

- Compare

$$o = \mathbf{sgn}\left(V(\pi_{\theta+\mu u}) - V(\pi_\theta)\right)$$

- Move

$$\tilde{\theta} = \theta + \beta \boldsymbol{g}(\theta, \mathbf{u})$$

$$\boldsymbol{g}(\theta, \mathbf{u}) = o\mathbf{u}$$

qualitative instead of quantitative

same direction but different step sizes

Sign-SZPO for Unknown Link Functions

Sign-SZPO: $\tilde{\theta} = \theta + \beta \boldsymbol{g}(\theta, \mathbf{u})$

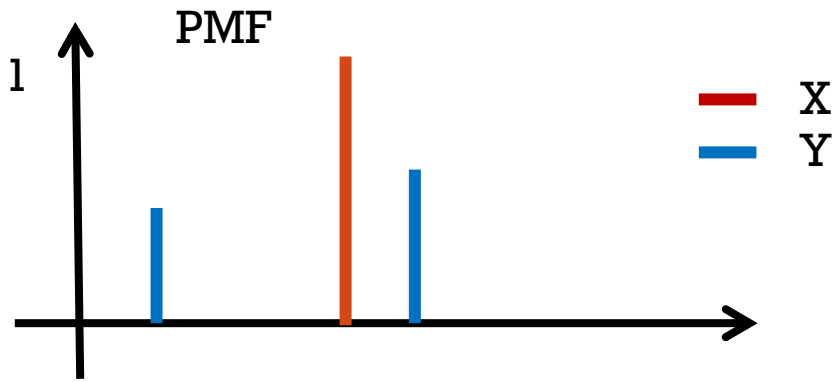
$\boldsymbol{g}(\theta, \mathbf{u}) = o\mathbf{u}$

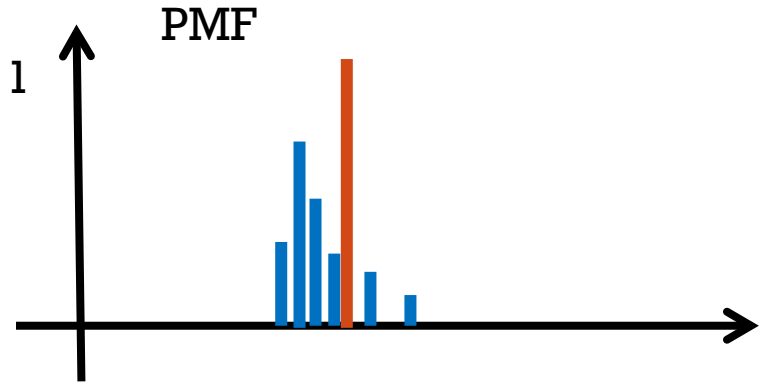
$o = \mathrm{sgn}\left(V(\pi_{\theta+\mu u}) - V(\pi_\theta)\right)$

- Estimation: Estimate whether $E[X] > E[Y]$ from $\{1(X_i > Y_i)\}_i$ → $P(X > Y)$

Randomness in human feedback → Multiple human evaluators

Randomness from the environment → Batch comparison $P\left(\frac{1}{D}\sum_{i=1}^{D} X_i > \frac{1}{D}\sum_{i=1}^{D} Y_i\right)$





Sign-SZPO for Unknown Link Functions

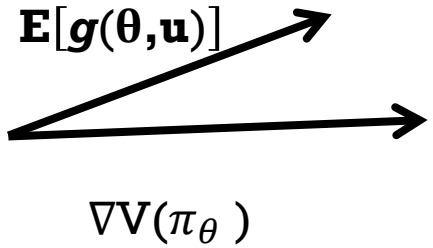
Sign-SZPO: $\tilde{\theta} = \theta + \beta \boldsymbol{g}(\theta,\mathbf{u})$

$\boldsymbol{g}(\theta,\mathbf{u}) = o\mathbf{u}$

$o = \mathrm{sgn}\left(V(\pi_{\theta+\mu u}) - V(\pi_\theta)\right)$

- Convergence

$\mathbf{E}[\boldsymbol{g}(\theta,\mathbf{u})] = \nabla$?



What does the sign tell us?

$$\mathrm{sgn}\left(V(\pi_{\theta+\mu\mathbf{u}}) - V(\pi_\theta)\right) \approx \mathrm{sgn}(\langle \nabla V(\pi_\theta), \mu u \rangle)$$

Sign-SZPO for Unknown Link Functions

Sign-SZPO: $\tilde{\theta} = \theta + \beta \boldsymbol{g}(\theta, \mathbf{u})$

$\boldsymbol{g}(\theta, \mathbf{u}) = o\mathbf{u}$

$o = \mathrm{sgn}\left(V(\pi_{\theta+\mu u}) - V(\pi_\theta)\right)$

- Let us look at the inner product

$$\begin{aligned}
&E[\langle \nabla V(\pi_\theta), g(\theta, u)\rangle] \\
&= E\left[\mathrm{sgn}\left(V(\pi_{\theta+\mu u}) - V(\pi_\theta)\right)\langle \nabla V(\pi_\theta), \mathbf{u}\rangle\right] \\
&= E[\mathrm{sgn}(\langle \nabla V(\pi_\theta), \mu \mathbf{u}\rangle)\langle \nabla V(\pi_\theta), \mathbf{u}\rangle] + E\left[\left(\mathrm{sgn}\left(V(\pi_{\theta+\mu u}) - V(\pi_\theta)\right) - \mathrm{sgn}(\langle \nabla V(\pi_\theta), \mu \mathbf{u}\rangle)\right)\langle \nabla V(\pi_\theta), \mathbf{u}\rangle\right]
\end{aligned}$$

$$E[|\langle \nabla V(\pi_\theta), \mathbf{u}\rangle|] = \sqrt{\frac{2}{\pi}}\|\nabla V(\pi_\theta)\|_2$$

well-aligned

Sign-SZPO for Unknown Link Functions

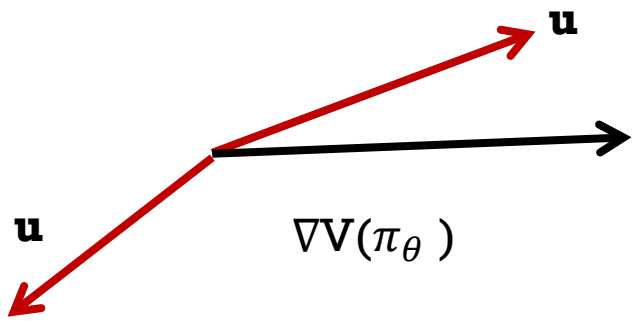
Sign-SZPO: $\tilde{\theta} = \theta + \beta \boldsymbol{g}(\theta, \mathbf{u})$
$\boldsymbol{g}(\theta, \mathbf{u}) = o\mathbf{u}$
$o = \mathrm{sgn}\left(V(\pi_{\theta+\mu u}) - V(\pi_\theta)\right)$

- Let us look at the inner product

$$\mathrm{E}[\langle \nabla V(\pi_\theta), g(\theta, u)\rangle]$$
$$= \mathrm{E}\left[\mathrm{sgn}\left(V(\pi_{\theta+\mu u}) - V(\pi_\theta)\right)\langle \nabla V(\pi_\theta), u\rangle\right]$$
$$= \mathrm{E}[\mathrm{sgn}(\langle \nabla V(\pi_\theta), \mu \mathbf{u}\rangle)\langle \nabla V(\pi_\theta), \mathbf{u}\rangle] + \mathrm{E}\left[\left(\mathrm{sgn}\left(V(\pi_{\theta+\mu u}) - V(\pi_\theta)\right) - \mathrm{sgn}(\langle \nabla V(\pi_\theta), \mu \mathbf{u}\rangle)\right)\langle \nabla V(\pi_\theta), \mathbf{u}\rangle\right]$$

$$V(\pi_{\theta+\mu \mathbf{u}}) - V(\pi_\theta) \in [\langle \nabla V(\pi_\theta), \mathbf{u}\rangle \mu - O(\mu^2 \|\mathbf{u}\|_2^2), \langle \nabla V(\pi_\theta), \mathbf{u}\rangle \mu + O(\mu^2 \|\mathbf{u}\|_2^2)]$$

Case 1: If $\langle \nabla V(\pi_\theta), \mathbf{u}\rangle = \omega(\pm\mu)$ i.e., $\mathbf{u}$ aligns with $\pm \nabla V(\pi_\theta)$



Sign-SZPO for Unknown Link Functions

Sign-SZPO: $\tilde{\theta} = \theta + \beta g(\theta, \mathbf{u})$

$g(\theta, \mathbf{u}) = o\mathbf{u}$

$o = \mathrm{sgn}\left(V(\pi_{\theta+\mu u}) - V(\pi_\theta)\right)$

- Let us look at the inner product

$$
\begin{aligned}
&E[\langle \nabla V(\pi_\theta), g(\theta,u)\rangle] \\
&= E\left[\mathrm{sgn}\left(V(\pi_{\theta+\mu u}) - V(\pi_\theta)\right)\langle \nabla V(\pi_\theta), u\rangle\right] \\
&= E[\mathrm{sgn}(\langle \nabla V(\pi_\theta), \mu \mathbf{u}\rangle)\langle \nabla V(\pi_\theta), \mathbf{u}\rangle] + E\left[\left(\mathrm{sgn}\left(V(\pi_{\theta+\mu u}) - V(\pi_\theta)\right) - \mathrm{sgn}(\langle \nabla V(\pi_\theta), \mu \mathbf{u}\rangle)\right)\langle \nabla V(\pi_\theta), \mathbf{u}\rangle\right] \\
&\qquad\qquad \leq 2 \qquad\qquad = O(\mu)
\end{aligned}
$$

Case 2: If $\langle \nabla V(\pi_\theta), \mathbf{u}\rangle = O(\mu)$

$$E[\langle \nabla V(\pi_\theta), g(\theta,u)\rangle] = \sqrt{\frac{2}{\pi}}\,\|\nabla V(\pi_\theta)\|_2 + O(\mu)$$

Convergence Result

Rate of convergence to a stationary point (smoothness and carefully chosen μ)

$$\tilde{O}\left(\sqrt{d}\left(\sqrt{\frac{H}{T}}+\max\left\{\frac{1}{\sigma'(0)},1\right\}\frac{1}{N^{\frac{1}{4}}}+\sqrt{\epsilon_D^*}\right)\right)$$

match the zeroth-order stochastic gradient

price for the zeroth-order (Duchi et al., 2014)

- H: planning horizon
- T: number of policy gradient iterations
- d: number of parameters

Convergence Result

Rate of convergence to a stationary point (smoothness and carefully chosen μ)

$$\tilde{O}\left(\sqrt{d}\left(\sqrt{\frac{H}{T}}+\max\left\{\frac{1}{\sigma'(0)},1\right\}\frac{1}{N^{\frac{1}{4}}}+\sqrt{\epsilon_D^*}\right)\right)$$

Distinguishability: infer the sign of the difference using preferences

$$P\left(\sum_{i=1}^{D} X_i > \sum_{i=1}^{D} Y_i\right) > 0.5 \quad \text{when } E[X]-E[Y] > \epsilon_D^*$$

Convergence Result

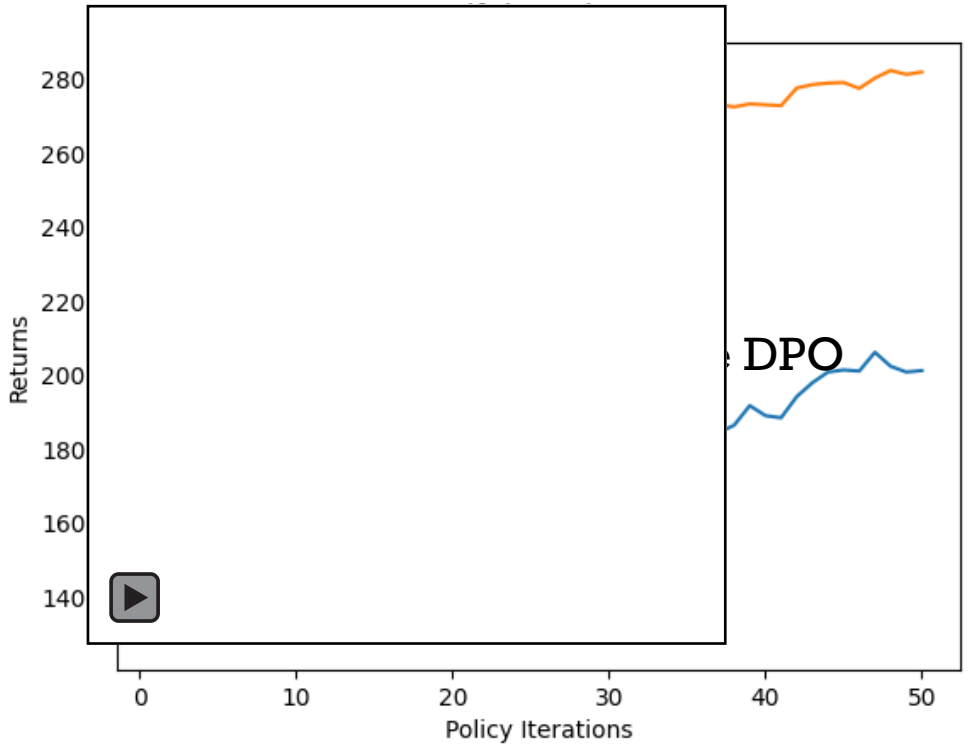
Rate of convergence to a stationary point (smoothness and carefully chosen μ)

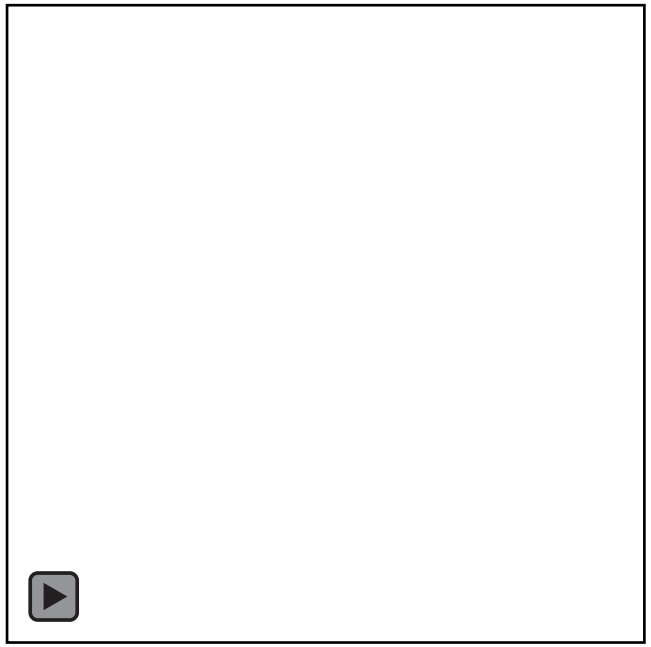
$$\tilde{O}\left(\sqrt{d}\left(\sqrt{\frac{H}{T}}+\max\left\{\frac{1}{\sigma'(0)},1\right\}\frac{1}{N^{\frac{1}{4}}}+\sqrt{\epsilon_D^*}\right)\right)$$

error due to limited expertise and a finite number of evaluators

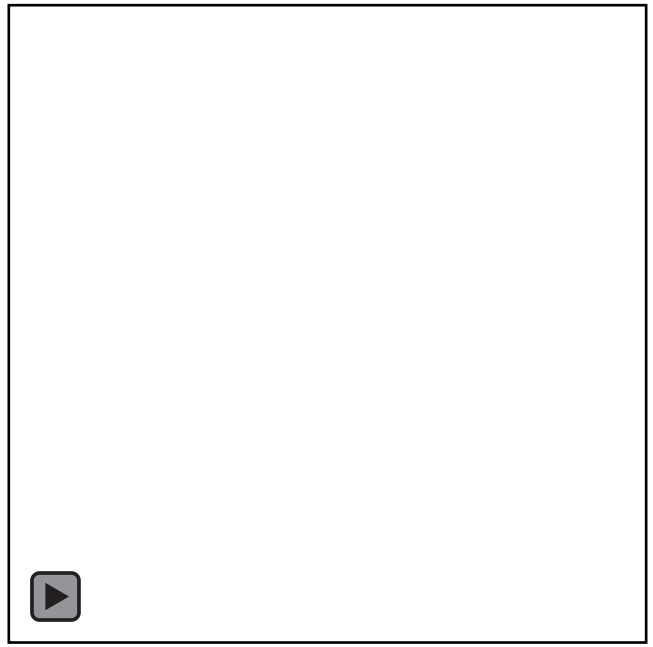
- N: number of trajectory-set pairs per policy iteration
- $\sigma()$: the unknown link function, $\sigma'(0)=\frac{1}{4}$ under the BT model

HalfCheetah-v5





Online DPO
(link function mismatch)



SZPO (Our algorithm)

When is SZPO for RLHF Useful?

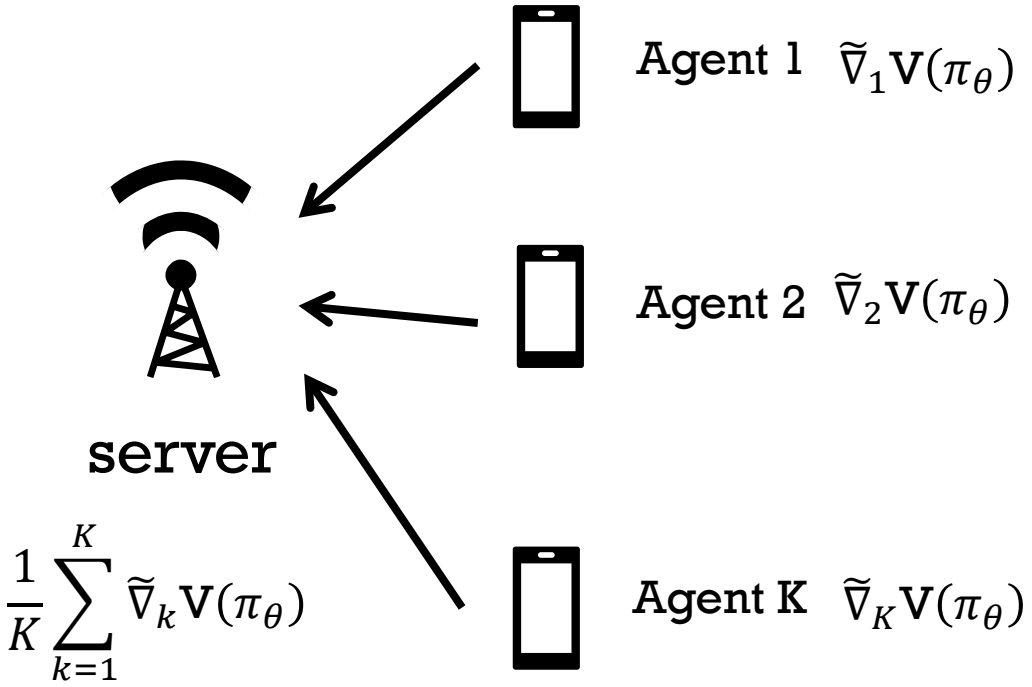
Result 1: Unknown preference models (link functions)

Qining Zhang and Lei Ying. "Provable Reinforcement Learning from Human Feedback with an Unknown Link Function." *arXiv preprint arXiv:2506.03066* (2025).

Result 2: Federated RLHF

Deyi Wang, Qining Zhang, Lei Ying. "Efficient Federated RLHF via Zeroth-Order Policy Optimization." *arXiv preprint arXiv:2604.17747* (2026).

Result 2: Efficient Federated RLHF

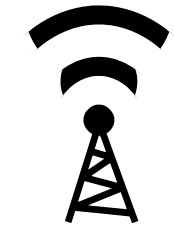


FedAvg

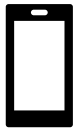
- Compute the gradient at each agent based on the local dataset
- Communicate the gradients to the server
- The server averages the gradients and updates the model

Expensive to compute, store, and communicate

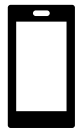
Result 2: Efficient Federated RLHF



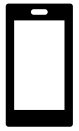
server



Agent 1



Agent 2

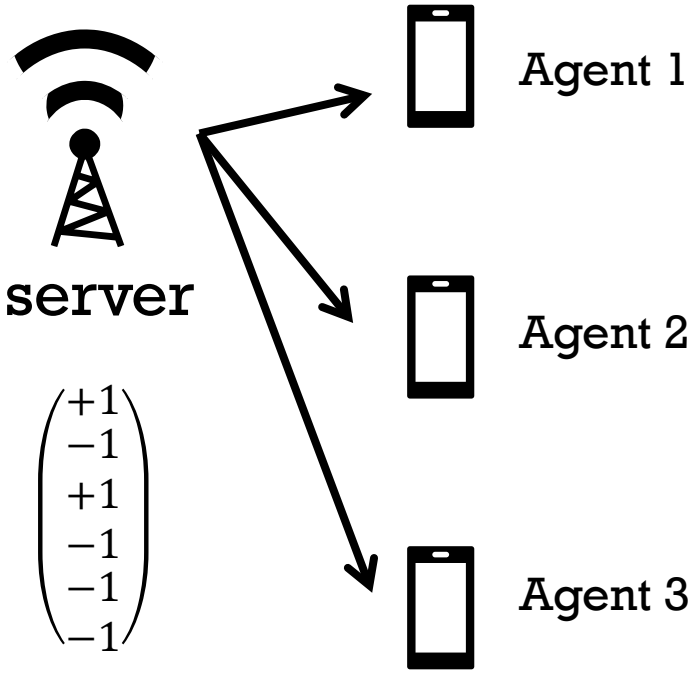


Agent K

Par-S^{2}ZPO

- ❑ Idea 1: Zeroth-order policy optimization
  - ❑ no need to compute and store gradients
- ❑ Idea 2: Bernoulli perturbation $\{+1, -1\}$ (SPSA (Spall 87, 92))
  - ❑ Easy to store and communicate
- ❑ Idea 3: Partitioned zeroth order
  - ❑ Further reduce memory complexity

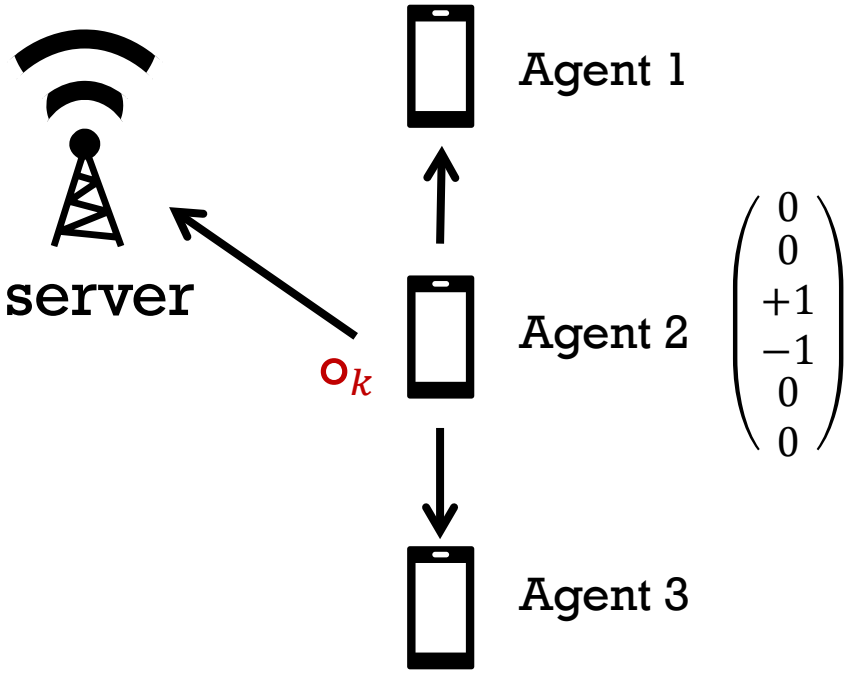
Result 2: Efficient Federated RLHF



Par-S^{2}ZPO

- Server: sample a Rademacher perturbation vector and broadcast

Result 2: Efficient Federated RLHF



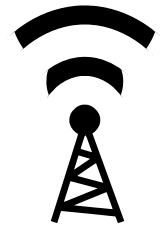
Par-S^{2}ZPO

- Server: sample a Rademacher perturbation vector and broadcast
- Agent k:
  - Extract its perturbation vector $\mathbf{u}_k$ and compare to evaluate

$$o_k = \operatorname{sgn}\left(V(\pi_{\theta+\mu\mathbf{u}_k}) - V(\pi_{\theta})\right)$$

  - Broadcast one-bit o_k

Par-S^{2}ZPO



server

Agent 1

Agent 2

Agent 3

Par-S^{2}ZPO

- Server: sample a Rademacher perturbation vector and broadcast
- Agent k:
  - Extract its perturbation vector $\mathbf{u}_k$ and compare to evaluate

$$o_k = \mathrm{sgn}\big(V(\pi_{\boldsymbol{\theta}+\mu\mathbf{u}_k}) - V(\pi_{\boldsymbol{\theta}})\big)$$

  - Broadcast one-bit feedback o_k
- Server and agents: update the model with

$$\widetilde{\boldsymbol{\theta}} = \boldsymbol{\theta} + \beta \sum_{k} o_k \mathbf{u}_k$$

Par-S^{2}ZPO

Communication	Memory	Computation
2d bits per policy update	$\frac{d}{K}$ bits per agent	no back propagation

Performance?

- ❑ The impact of binary perturbation
- ❑ The impact of distributed evaluation

Environments	$K = 1$ (G)	$K = 1$ (B)	$K = 5$ (B)	$K = 10$ (B)	$K = 15$ (B)
Half Cheetah	4375.40 ± 30.90	4394.77 ± 27.97	4398.68 ± 31.66	4422.25 ± 24.62	4424.69 ± 29.49
Hopper	2557.88 ± 119.43	2624.65 ± 78.81	2658.00 ± 76.46	2594.85 ± 102.66	2583.12 ± 84.68
Swimmer	94.85 ± 4.53	91.37 ± 1.74	94.57 ± 1.96	92.59 ± 1.32	91.80 ± 1.37
Walker2D	2376.34 ± 58.49	2433.64 ± 54.93	2649.10 ± 42.00	2552.52 ± 60.18	2651.27 ± 50.74

Par-S^2ZPO

Communication	Memory	Computation
2d bits per policy update	$\frac{d}{K}$ bits per agent	no back propagation

The "rate of convergence" to a stationary point

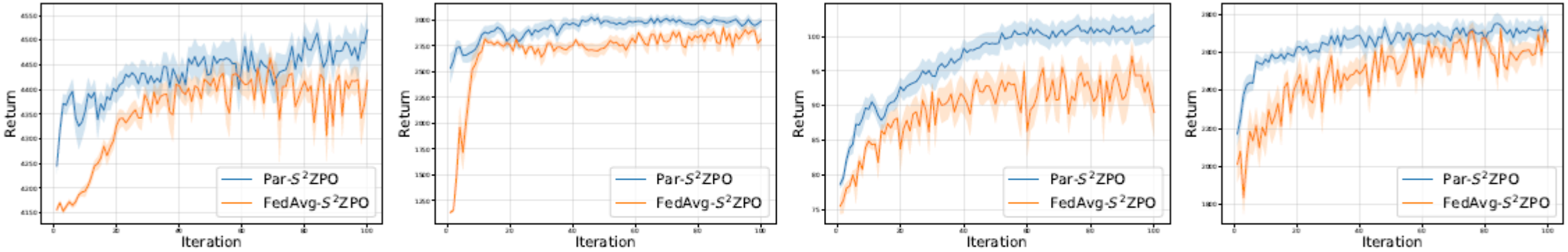
$$\tilde{O}\left(\sqrt{d}\left(\sqrt{\frac{2NDH}{M}}+\sqrt{\varsigma_P^{-1}\left(\frac{4}{N}\right)}+\sqrt{\epsilon_D^*}\right)\right)$$

M=2DNKT is the sample complexity

- D: batch size
- N: number of batches
- K: number of agents
- T: number of policy iterations

Par-S^2ZPO

Communication	Memory	Computation
2d bits per policy update	$\frac{d}{K}$ bits per agent	no back propagation



(a) Half Cheetah (b) Hopper (c) Swimmer (d) Walker 2D

Figure 3: Par-S^2ZPO versus FedAvg-S^2ZPO

Thanks!
Questions?