Reinforcement Learning with Human Feedback

Priyank Agrawal

IEOR, Columbia University

Presentation

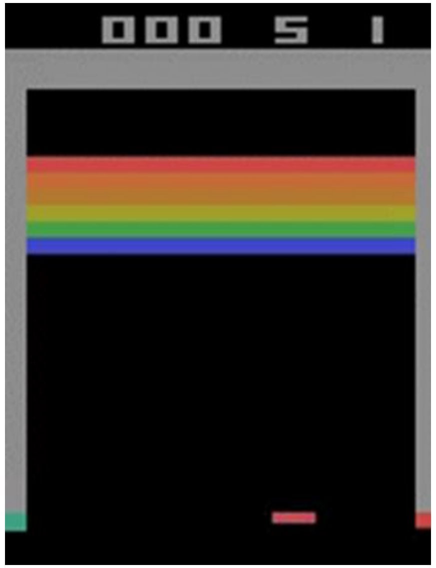
Part 1	Typical RL setting
	Case for preference-based learning
	Choice Models, BTL
	(Deep) Learning from human preferences and PbRL paper
Part 2	RLHF pipeline
	Primer for the PPO algorithm
	PPO algorithm
	DPO algorithm
Part 3	Limitations
Part 4	Conclusion and after thoughts

Part 1

- A typical RL setting
- Making a case for preference-based learning.
- Choice Models: BTL
- (Deep) Learning from human preferences and PbRL paper

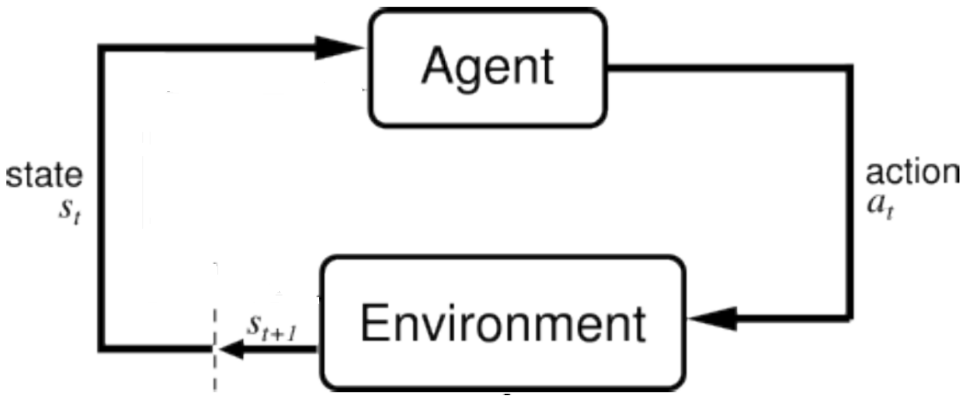
Success of Reinforcement Learning

Game playing, robotics, online shopping



Inventory management, Resource management/Queuing

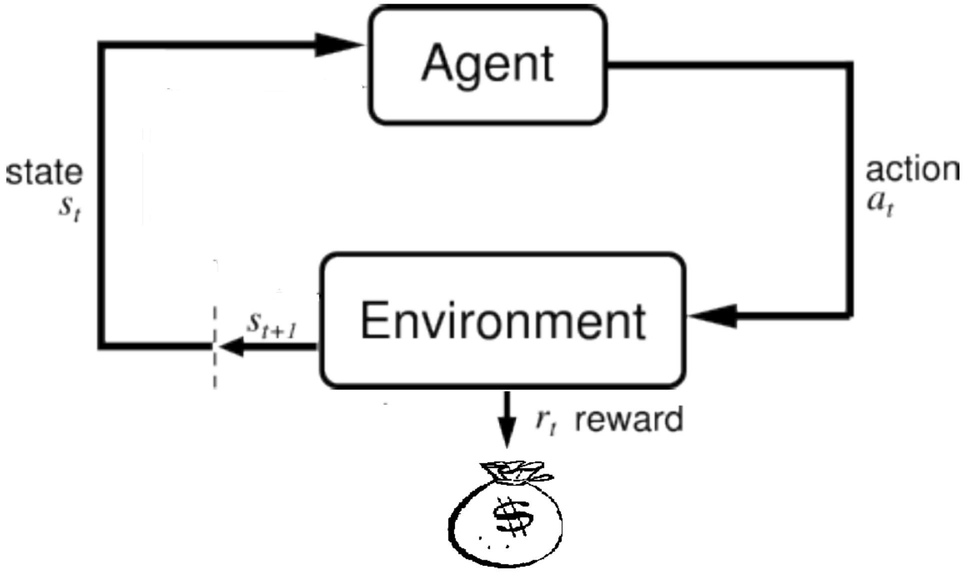
A (typical) RL instance



1. Large but tractable set of state and actions
2. Markovian transitions.

- (Largely) offline problem
- Past data = $\{s_1, a_1, s_2, a_2, \ldots s_T, a_T\}$.
- Potential goals
  - **(Task learning)** reach a goal state fast
  - **(Long-term decision making)** prioritize reaching certain “good” states often.
- Train a loss function that emphasizes the desired goal(s) and finds a good policy.

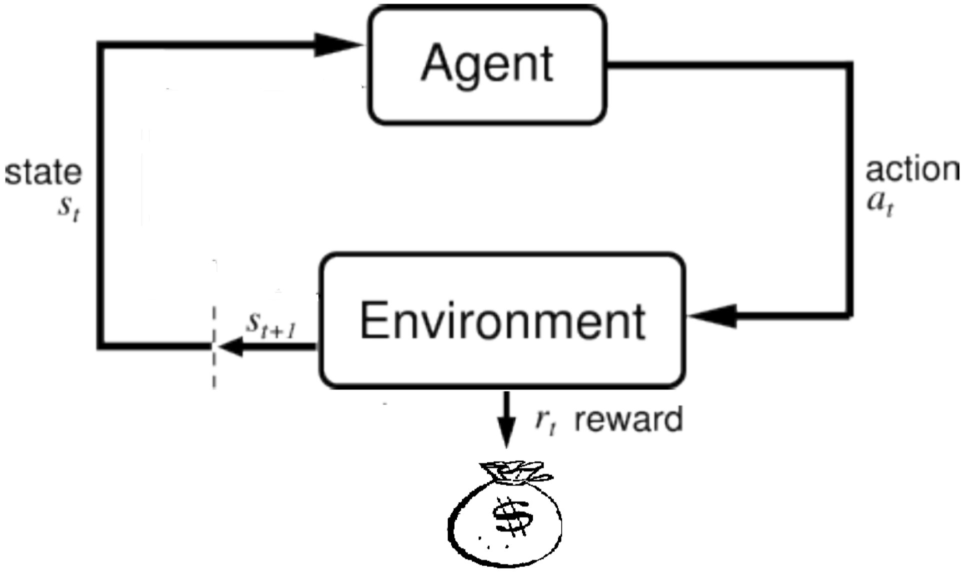
A (typical) RL instance



1. Large but tractable set of state and actions
2. Markovian transitions and rewards

- (Largely) offline problem
- Past data = $\{s_1, a_1, s_2, a_2, \ldots s_T, a_T\}$.
- Potential goals
  - **(Task learning)** reach a goal state fast
  - **(Long-term decision making)** prioritize reaching certain "good" states often.
- Train a loss function that emphasizes the desired goal(s) and finds a good policy.

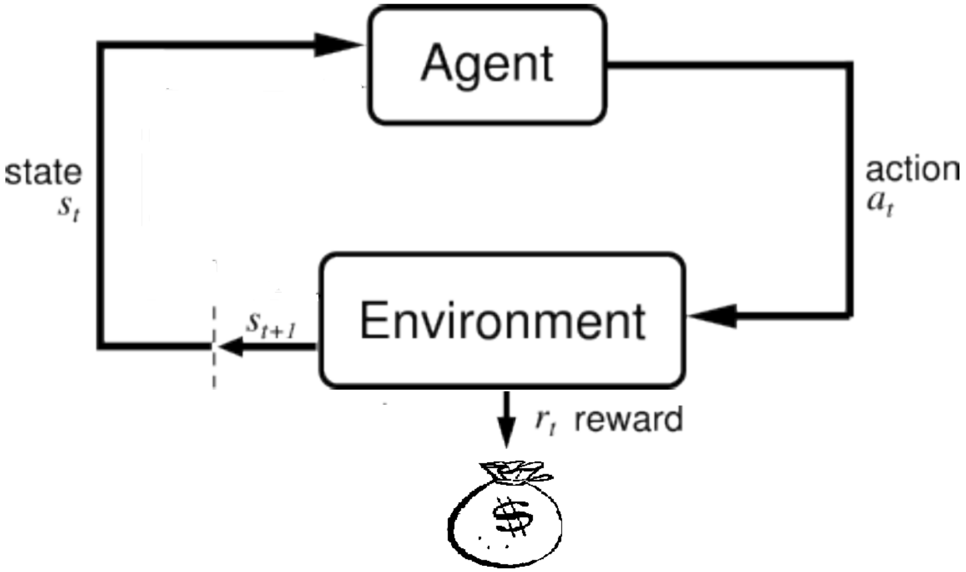
A (typical) RL instance



1. Large but tractable set of state and actions
2. Markovian transitions and rewards

- (Largely) offline problem
- Past data = $\{s_1, a_1, \boldsymbol{r_1}, s_2, a_2, \boldsymbol{r_2} \dots s_T, a_T, \boldsymbol{r_T}\}$.
- Potential goals
  - **(Task learning)** reach a goal state fast : high reward for the goal state, negative reward for non-goal states.
  - **(Long-term decision making)** prioritize reaching certain "good" states often: choice of reward selection.
- Train a loss function that emphasizes the desired goal(s) and finds a good policy.

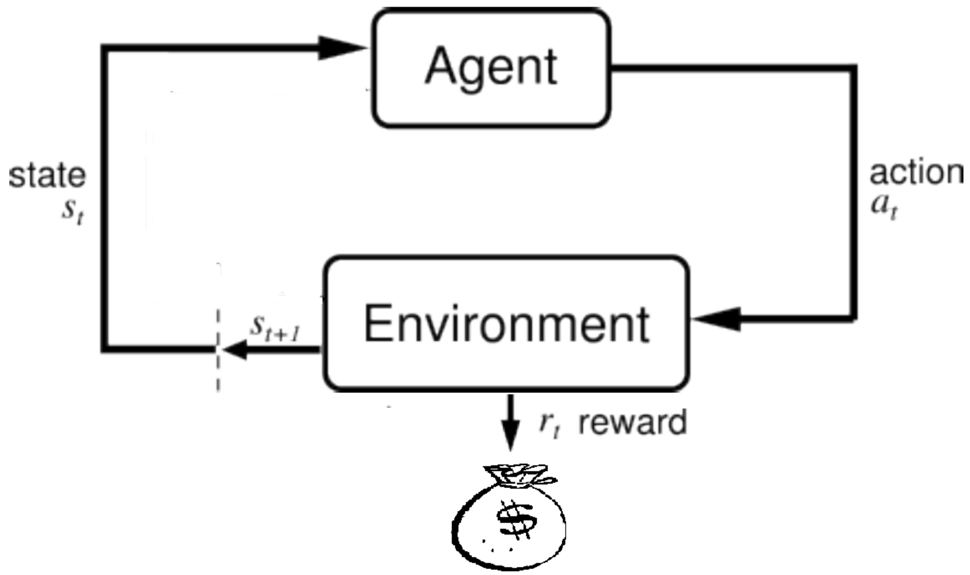
A (typical) RL instance



1. Large but tractable set of state and actions
2. Markovian transitions and rewards

- Past data = $\{s_1, a_1, \boldsymbol{r_1}, s_2, a_2, \boldsymbol{r_2} \ldots s_T, a_T, \boldsymbol{r_T}\}$.
- Define Value function $V^\pi = E_{a\sim\pi, s\sim P}[\sum_{t\geq 1} \gamma^t r(s_t, a_t)]$, $0 < \gamma \leq 1$.
- Find π such that $V^{\pi^*} - V^\pi \leq \epsilon, V^{\pi^*} = \max_{\pi'} V^{\pi'}$.

A (typical) RL instance



1. Large but tractable set of state and actions
2. Markovian transitions and rewards

- Past data = $\{s_1, a_1, \boldsymbol{r_1}, s_2, a_2, \boldsymbol{r_2} \ldots s_T, a_T, \boldsymbol{r_T}\}$.
- Define Value function $V^{\pi} = E_{a\sim\pi, s\sim P}[\sum_{t\geq 1} \gamma^t r(s_t, a_t)]$, $0 < \gamma \leq 1$.
- Find π such that $V^{\pi^*} - V^{\pi} \leq \epsilon, V^{\pi^*} = \max_{\pi'} V^{\pi'}$.
- Reward feedback can be provided by human labelers, machine etc.
- Primary difference that we consider is that there is trajectory level preference feedback.

A case for Preference based learning

Typical RL instance requires significant reward-engineering, domain knowledge and definition of a compact reward function.

- **Reward hacking**
  - Example: Say a robotic vacuum cleaner learns to hide dirt instead of actually, removing the dirt.
  - Designing "unhackable" reward functions.

'' A Survey of Preference-Based Reinforcement Learning Methods'', JMLR 2017.
"Defining and Characterizing reward hacking", NeurIPS 2022.

A case for Preference based learning

Typical RL instance requires significant reward-engineering, domain knowledge and definition of a compact reward function.

- **Reward hacking**
  - Example: Say a robotic vacuum cleaner learns to hide dirt instead of actually, removing the dirt.
  - Designing "unhackable" reward functions.
- **Reward shaping**
  - Example: robot picking a glass what is a good reward function? Goal "image" of glass in air? What if the glass has a dark liquid or background changes?

'' A Survey of Preference-Based Reinforcement Learning Methods'', JMLR 2017.
"Defining and Characterizing reward hacking", NeurIPS 2022.

A case for Preference based learning

Typical RL instance requires significant reward-engineering, domain knowledge and definition of a compact reward function.

- **Reward hacking**
  - Example: Say a robotic vacuum cleaner learns to hide dirt instead of actually, removing the dirt.
  - Designing “unhackable” reward functions.
- **Reward shaping**
  - Example: robot picking a glass what is a good reward function? Goal “image” of glass in air? What if the glass has a dark liquid or background changes?
- **Multi-objective reward**
  - Example: Economic policies that prioritize economic growth without letting inflation grow too much.

‘’ A Survey of Preference-Based Reinforcement Learning Methods’’, JMLR 2017.
“Defining and Characterizing reward hacking”, NeurIPS 2022.

A case for Preference based learning

Preference-based learning

- $\tau_1 = \{s_1, a_1, s_2, a_2, \dots, s_t, a_t\}$ and $\tau_2 = \{s'_1, a'_1, s'_2, a'_2, \dots, s'_t, a'_t\}$. Typical feedback $\{\tau_1 \geq \tau_2\}$.
- Can utilize expert feedback, non-expert "common-sense" feedback
- Comparing two options is often easier than generating an expert trajectory **(Imitation learning)** or finding a reward function first from human demonstrations **(Inverse RL)** to train an agent.

'' A Survey of Preference-Based Reinforcement Learning Methods'', JMLR 2017.
"Defining and Characterizing reward hacking", NeurIPS 2022.

Preference modeling example

- There are 30 basketball teams in the NBA, each playing 82 games in the regular season (so there are 1230 total games).
- We observe, at the end of the regular season, which two teams (i, j) played in each game, and whether team i or team j won.
- **How can we rank the teams and/or determine the strength of each team?**

''https://web.stanford.edu/class/archive/stats/stats200/stats200.1172/Lecture24.pdf''

Preference modeling example

- There are 30 basketball teams in the NBA, each playing 82 games in the regular season (so there are 1230 total games).
- We observe, at the end of the regular season, which two teams (i, j) played in each game, and whether team i or team j won.
- **How can we rank the teams and/or determine the strength of each team?**
- The simplest strategy might be to compare the number of games won by each team.

''https://web.stanford.edu/class/archive/stats/stats200/stats200.1172/Lecture24.pdf"

Preference modeling example

- There are 30 basketball teams in the NBA, each playing 82 games in the regular season (so there are 1230 total games).
- We observe, at the end of the regular season, which two teams (i, j) played in each game, and whether team i or team j won.
- **How can we rank the teams and/or determine the strength of each team?**
- The simplest strategy might be to compare the number of games won by each team.
- However, the NBA season is structured so that every team plays every other team a different number of times (between 2 and 4).
- The teams have different “strengths of schedule”, meaning that some teams play stronger opponents more frequently than do other teams.
- These teams might have worse win-loss records, but in fact be better than other teams that won more games against weaker opponents.

’’https://web.stanford.edu/class/archive/stats/stats200/stats200.1172/Lecture24.pdf”

Bradley-Terry Model (BTL)

- Let $\beta_i \in R$, denote the strength of team i.
- Let the outcome of the game between teams i, j be determined by $\beta_i - \beta_j$.
- Then **Bradley Terry Model** assumes the outcome as an independent Bernoulli random variable with distribution Bernoulli(p_{ij}), where the log-odds corresponding to the probability p_{ij} that the team i beats team j is modeled as,

$$\log\left(\frac{p_{ij}}{1-p_{ij}}\right) = \beta_i - \beta_j.$$

$$p_{ij} = \frac{e^{\beta_i - \beta_j}}{1 + e^{\beta_i - \beta_j}} = \frac{e^{\beta_i}}{e^{\beta_i} + e^{\beta_j}}.$$

''https://web.stanford.edu/class/archive/stats/stats200/stats200.1172/Lecture24.pdf"

Bradley-Terry Model (BTL)

- Let $\beta_i \in R$, denote the strength of team i.
- Let the outcome of the game between teams i, j be determined by $\beta_i - \beta_j$.
- Then **Bradley Terry Model** assumes the outcome as an independent Bernoulli random variable with distribution Bernoulli(p_{ij}), where the log-odds corresponding to the probability p_{ij} that the team i beats team j is modeled as,

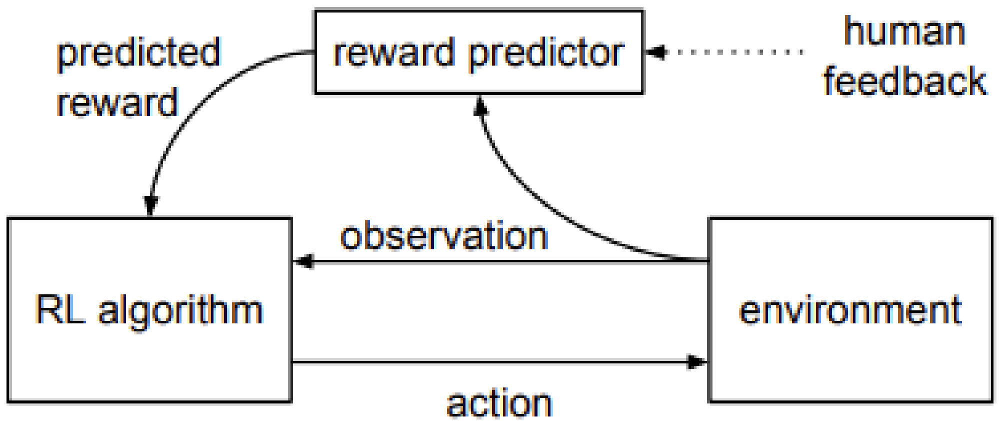
$$\log\left(\frac{p_{ij}}{1-p_{ij}}\right) = \beta_i - \beta_j.$$

$$p_{ij} = \frac{e^{\beta_i - \beta_j}}{1 + e^{\beta_i - \beta_j}} = \frac{e^{\beta_i}}{e^{\beta_i} + e^{\beta_j}}.$$

- Invariant under constant scaling, outcomes independent of non-competing teams
- Model can be enhanced by parametrizations and link functions $\beta_i \leftarrow \sigma\big(f(\beta_i)\big)$.

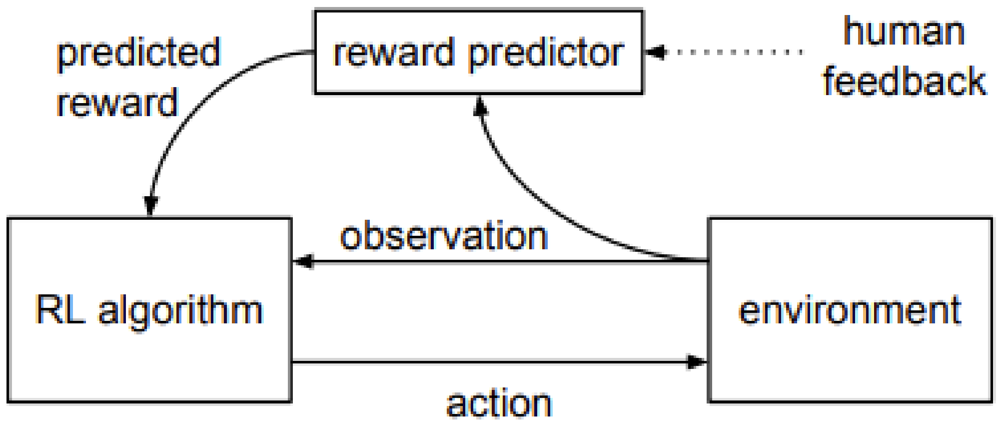
''https://web.stanford.edu/class/archive/stats/stats200/stats200.1172/Lecture24.pdf"

Application of BTL: RL with learning from human preferences [2017]



“Deep Reinforcement Learning from Human Preferences”, 2017

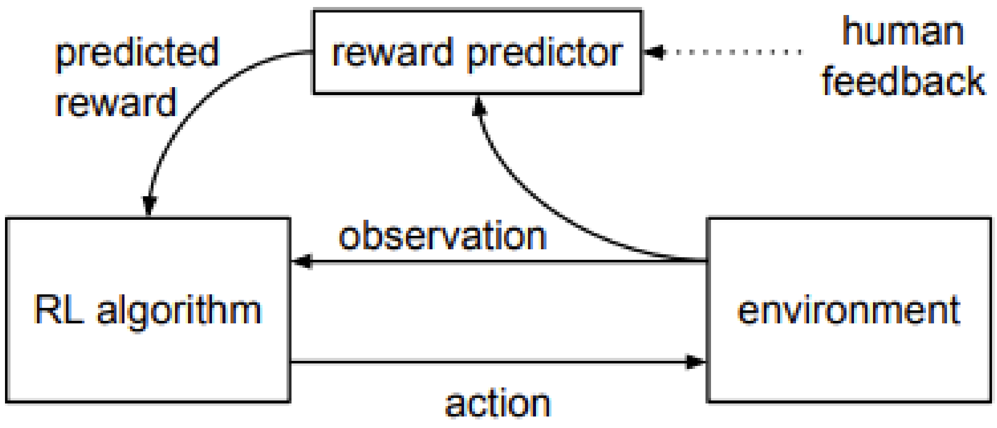
Application of BTL: RL with learning from human preferences [2017]



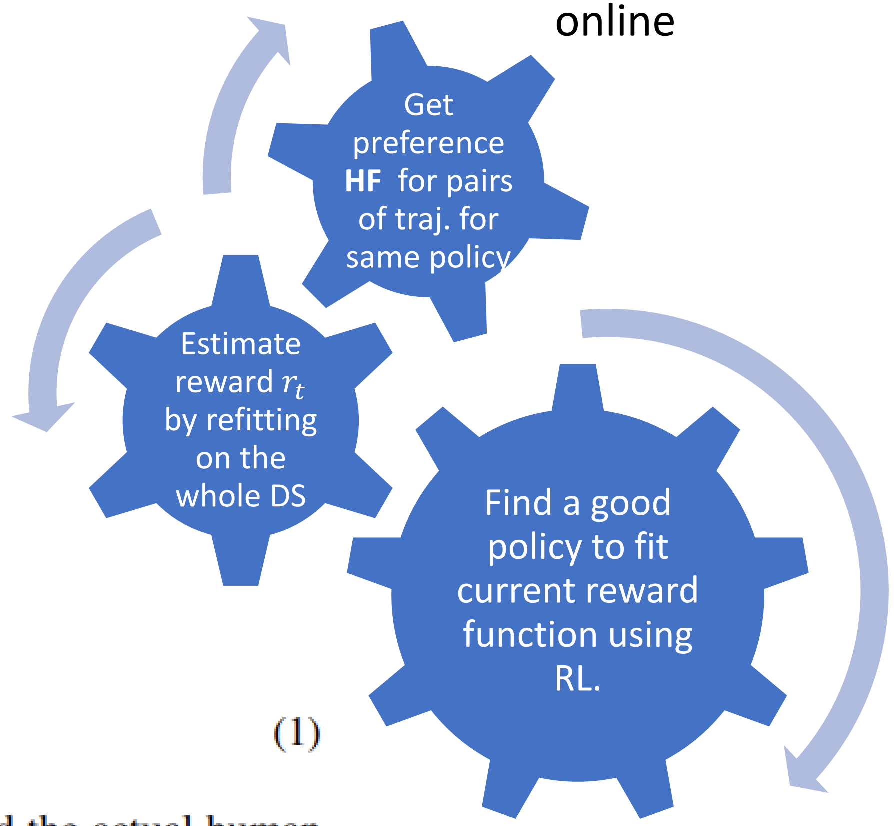
Done asynchronously but online

“Deep Reinforcement Learning from Human Preferences”, 2017

Application of BTL: RL with learning from human preferences [2017]



Done asynchronously but online



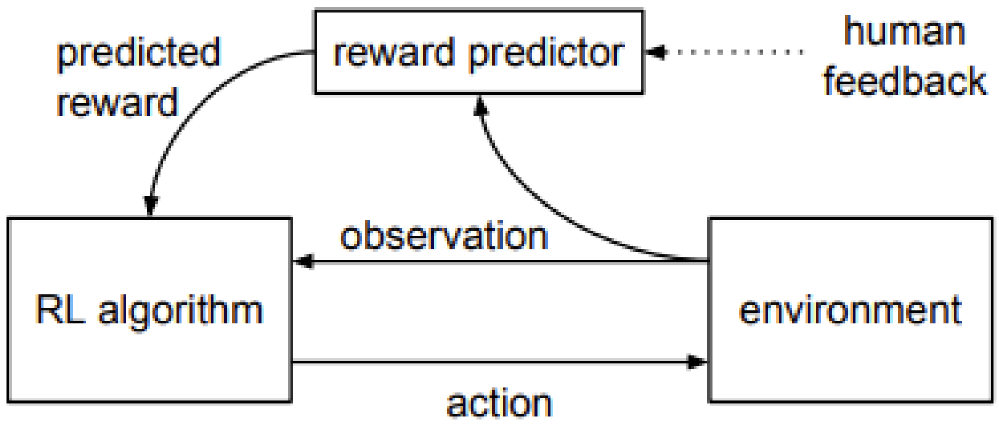
$$\hat{P}[\sigma^1 \succ \sigma^2] = \frac{\exp \sum \hat{r}(o_t^1, a_t^1)}{\exp \sum \hat{r}(o_t^1, a_t^1) + \exp \sum \hat{r}(o_t^2, a_t^2)}. \tag{1}$$

We choose $\hat{r}$ to minimize the cross-entropy loss between these predictions and the actual human labels:

$$\text{loss}(\hat{r}) = -\sum_{(\sigma^1, \sigma^2, \mu) \in \mathcal{D}} \mu(1) \log \hat{P}[\sigma^1 \succ \sigma^2] + \mu(2) \log \hat{P}[\sigma^2 \succ \sigma^1].$$

"Deep Reinforcement Learning from Human Preferences", 2017

Application of BTL: RL with learning from human preferences [2017]



Superior convergence rates in environments like Atari, MuJuCo, etc

Done asynchronously but online

"Deep Reinforcement Learning from Human Preferences", 2017

Part 2

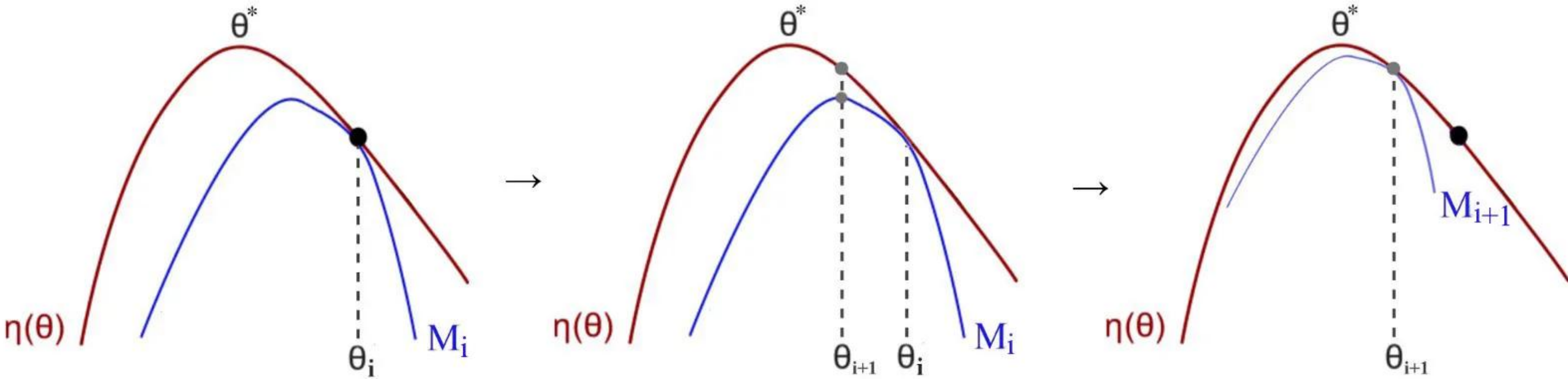
- PPO algorithm
- RLHF
- DPO algorithm

A primer to the (Proximal Policy Optimization) PPO algorithm: Non-RL view

<u>Minorize-Maximization (MM) Algorithm</u>

How to optimize a function like $f(\theta) = V^{\pi_\theta}$*?*

Steps in MM algorithm



The above method guarantees that $f(\theta_i)$ *converges to a local optima or saddle point as* $i \rightarrow \infty$.

$$f(\theta_{i+1}) \geq g(\theta_{i+1}|\theta_i) \geq g(\theta_i|\theta_i) = f(\theta_i).$$

"https://jonathan-hui.medium.com/rl-proximal-policy-optimization-ppo-explained-77f014ec3f12"

A primer to the PPO algorithm: Non-RL view

<u>Minorize-Maximization (MM) Algorithm</u>

How to optimize a function like $\eta(\theta) = V^{\pi_\theta}$*?*

Steps in MM algorithm

- The algorithm proceeds in iteration $i = 1,2,3, \ldots$
- Let $M_i = g(\theta|\theta_i)$ be a surrogate which be **minorized version** of the objective function $f(\theta)$, satisfying
  - $g(\theta|\theta_i) \leq f(\theta) \forall\, \theta$.
  - $g(\theta_i|\theta_i) = f(\theta_i)$.
- The algorithm maximizes $g(\theta|\theta_i)$ instead:
  - $\theta_{i+1} = argmax_\theta g(\theta|\theta_i)$.

The above method guarantees that $f(\theta_i)$ *converges to a local optima or saddle point as* $i \to \infty$.

$$f(\theta_{i+1}) \geq g(\theta_{i+1}|\theta_i) \geq g(\theta_i|\theta_i) = f(\theta_i).$$

"https://jonathan-hui.medium.com/rl-proximal-policy-optimization-ppo-explained-77f014ec3f12"

A primer to the PPO algorithm: Non-RL view

- *<u>If $g(\theta) = f(\theta)$, that is if we optimize V^{π_θ} directly then we get the family of the policy gradient algorithms.</u>*
  - Examples include REINFORCE [Williams 1988, Williams 1992] , DQN [2016], among others
  - Practical implementations still involved formulations (e.g. Baseline trick) and engineering heuristics (DQN for Atari)

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s)$$

policy gradient (steepest direction to maximize rewards)

$$g = \nabla_\theta J(\pi_\theta) = \mathop{\mathrm{E}}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t \nabla_\theta \log \pi_\theta(a_t|s_t) A^{\pi_\theta}(s_t, a_t)\right]$$

$$\theta_{k+1} = \theta_k + \alpha\, g$$

take a gradient step in updating the policy

“https://jonathan-hui.medium.com/rl-proximal-policy-optimization-ppo-explained-77f014ec3f12”

A primer to the PPO algorithm: Non-RL view

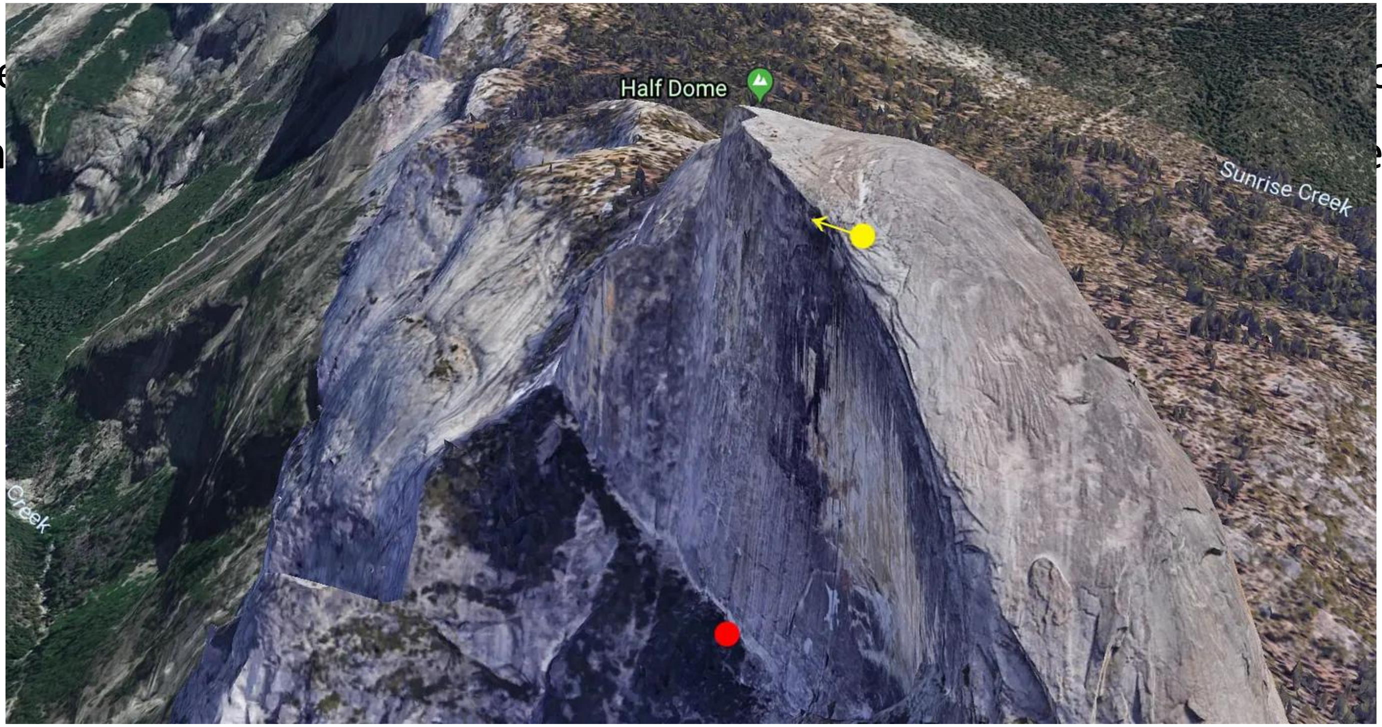
- <u>*If $g(\theta) = f(\theta)$, that is if we optimize V^{π_θ} directly then we get the family of the policy gradient algorithms.*</u>
  - Examples include REINFORCE [Williams 1988, Williams 1992] , DQN [2016], among others
  - Practical implementations still involved formulations (e.g. Baseline trick) and engineering heuristics (DQN for Atari)

  - First order methods assume the Value function surface to be flat. High curvature can be bad for learning

"https://jonathan-hui.medium.com/rl-proximal-policy-optimization-ppo-explained-77f014ec3f12"

A primer to the PPO algorithm: Non-RL view

- *If $g(\theta) = f(\theta)$, that is if we optimize V^{π_θ} directly then we get the family of the policy gradient algorithms.*
  - Examples include … others
  - Practical implem… …eering heuristics (DQN for Atari)



  - Susceptible to learning rate schedule, large policy changes

"https://jonathan-hui.medium.com/rl-proximal-policy-optimization-ppo-explained-77f014ec3f12"

A primer to the PPO algorithm:

- **Choice of $\boldsymbol{g(\theta)}$ that minorizes $\boldsymbol{f(\theta)}$**
  - Key idea: We find π' that **locally** improves $J(\pi')$ when compared to $J(\pi_{old})$ for some π_{old}.
  - Suppose the objective is $\max_{\pi'} J(\pi') = E_{a\sim\pi', s\sim P}[\sum_{t\geq 1} \gamma^t r_t]$
  - We only care about the argmax policy
  - So instead consider the objective $\mathrm{f}(\theta) = \max_{\pi'} J(\pi') - J(\pi)$
  - We will find a function $g(\theta)$ that minorizes $\mathrm{f}(\theta)$: [result of the famous TRPO 2015 paper]

$$J(\pi') - J(\pi) \geq \underbrace{\mathcal{L}_\pi(\pi') - C\sqrt{\mathop{\mathrm{E}}_{s\sim d^\pi}[D_{KL}(\pi'||\pi)[s]]}}_{\mathrm{M}}$$

  - Choice of $L_\pi(\pi')$ is very specific.
  - Key point M is non-negative therefore we have monotonic improvement

"https://jonathan-hui.medium.com/rl-proximal-policy-optimization-ppo-explained-77f014ec3f12"

A primer to the PPO algorithm

$$\max_{\pi'} \mathcal{L}_{\pi}(\pi') - C\sqrt{\mathop{\mathrm{E}}_{s\sim d^{\pi_k}}[D_{KL}(\pi'||\pi)[s]]}$$

or

$$\max_{\pi'} \mathcal{L}_{\pi}(\pi')$$

$$\text{s.t.} \mathop{\mathrm{E}}_{s\sim d^{\pi}}\left[D_{KL}(\pi'||\pi)[s]\right] \leq \delta$$

A primer to the PPO algorithm

$$\max_{\pi'} \mathcal{L}_{\pi}(\pi') - C\sqrt{\mathop{\mathrm{E}}_{s\sim d^{\pi_k}}\left[D_{KL}(\pi'||\pi)[s]\right]}$$

or

$$\max_{\pi'} \mathcal{L}_{\pi}(\pi')$$
$$\text{s.t.} \mathop{\mathrm{E}}_{s\sim d^{\pi}}\left[D_{KL}(\pi'||\pi)[s]\right] \leq \delta$$

- Approximate the expected advantage function locally around the current policy.
- The accuracy decreases when the new policy and the current policy diverge from each other.
- KL term acts as an upper bound for the error.

$$\underset{\theta}{\text{maximize}} \quad \underbrace{\hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right]}_{\mathcal{L}}$$

$$\text{subject to} \quad \hat{\mathbb{E}}_t[\mathrm{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]] \leq \delta.$$

The PPO algorithm [2017]

- The two variants of the PPO algorithm

Algorithm 4 PPO with Adaptive KL Penalty

Input: initial policy parameters θ_0, initial KL penalty β_0, target KL-divergence δ
for $k = 0, 1, 2, ...$ **do**
  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

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

  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

The PPO algorithm [2017]

- The two variants of the PPO algorithm

$$r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_k}(a_t|s_t)$$

Algorithm 5 PPO with Clipped Objective

Input: initial policy parameters θ_0, clipping threshold ϵ

for $k = 0, 1, 2, ...$ **do**

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

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

end for

The PPO algorithm [2017]

- The PPO algorithm is easy to implement in practice and works well.
- Only a few lines of code change from the vanilla policy gradient algorithm (clipped version works well)
- Paper shows it to perform better or similar than contemporary algorithms on variety of tasks.
- TRPO[2015] introduced the idea of using a surrogate loss to optimize the value function. PPO simplifies the implementation with stronger performance.

RLHF Pipeline and DPO

- **<u>Key takeaway from PPO: we have a surrogate reward function that approximately minorizes the reward function of our interest, we optimize the surrogate reward function instead.</u>**
- Foundation models may not have “human-need-aligned” output therefore they need to be fine-tuned to satisfy ethics/safety/security constraints or for a specific use-case.
- RLHF in the context of foundation model fine-tuning refers to a 3 step process, where PPO is used in the 3rd step.
- **Direct Preference Optimization (DPO)** is an efficient way of combining the 2nd and the 3rd steps of the RLHF pipeline.

Slides from Rafael Rafailov Archit Sharma Eric Mitchell

“https://web.stanford.edu/class/cs234/CS234Spr2024/slides/dpo_slides.pdf”

Direct Preference Optimization: A New RLHF Approach

Rafael Rafailov Archit Sharma Eric Mitchell

Stanford University

RLHF: Reinforcement Learning From Human Feedback

RLHF: Reinforcement Learning From Human Feedback

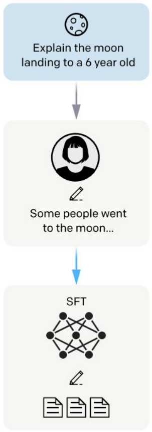
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

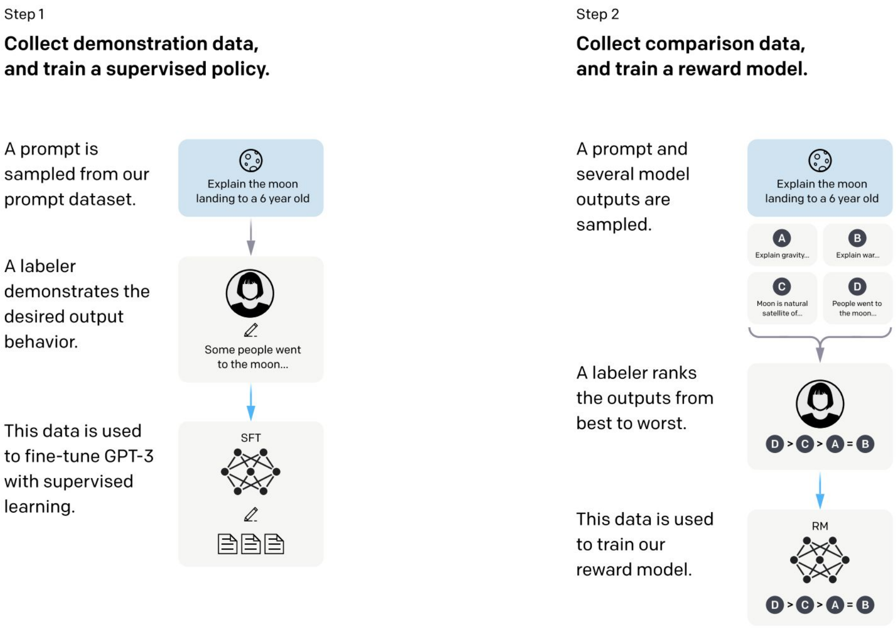
A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.



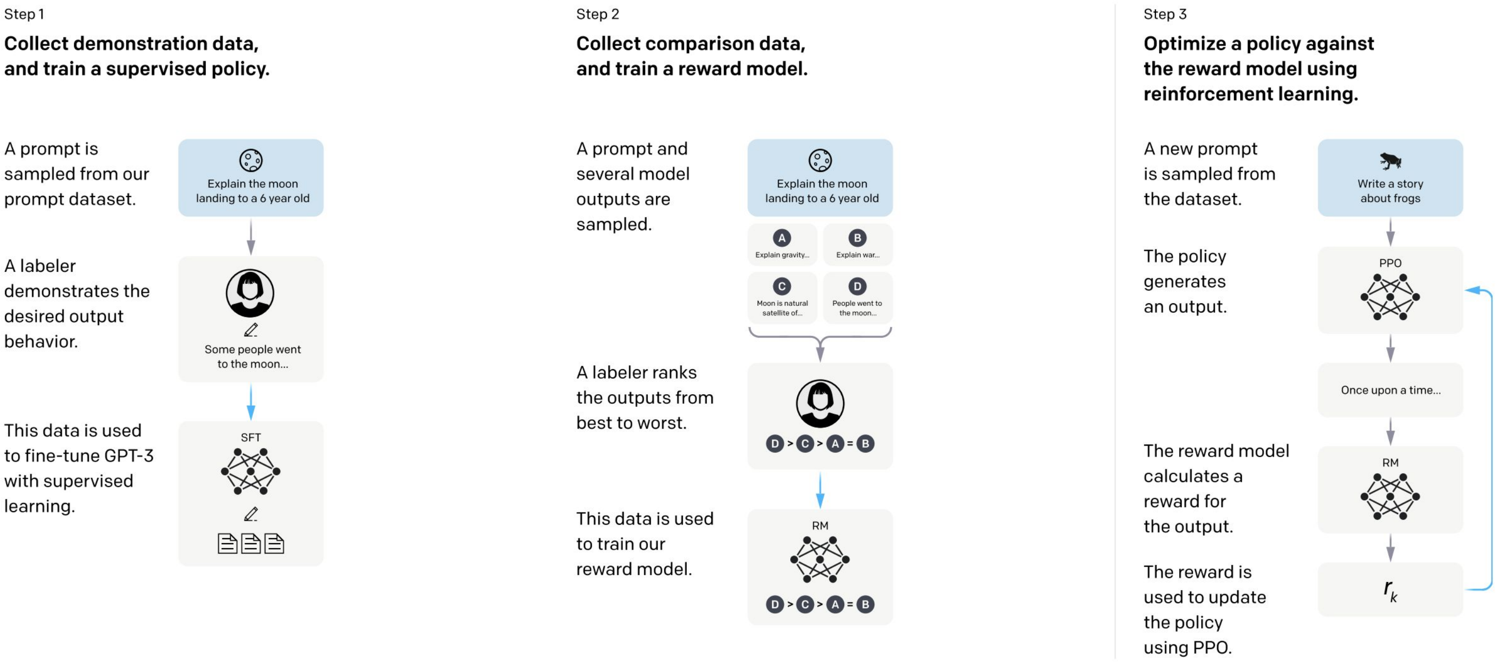
Training language models to follow instructions with human feedback, Ouyang et. al. 2022

RLHF: Reinforcement Learning From Human Feedback



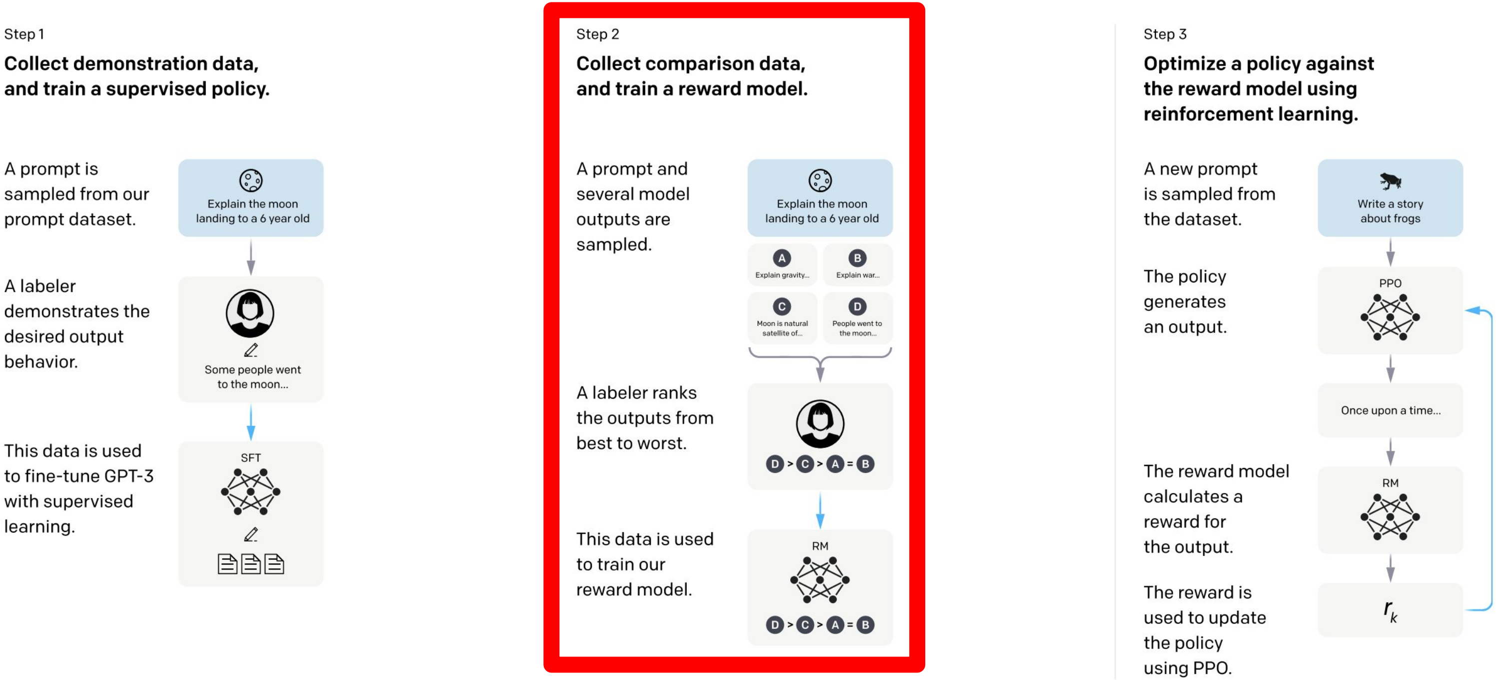
Training language models to follow instructions with human feedback, Ouyang et. al. 2022

RLHF: Reinforcement Learning From Human Feedback



Training language models to follow instructions with human feedback, Ouyang et. al. 2022

RLHF: Learning a **reward model** from human feedback



Training language models to follow instructions with human feedback, Ouyang et. al. 2022

RLHF: Learning a **reward model** from human feedback

RLHF: Learning a **reward model** from human feedback

Feedback comes as **preferences over model samples:** $\mathcal{D} = \{x^i, y_w^i, y_l^i\}$

RLHF: Learning a **reward model** from human feedback

Feedback comes as **preferences over model samples:** $\mathcal{D} = \{x^i, y_w^i, y_l^i\}$

Prompt

Preferred response

Dispreferred response

RLHF: Learning a **reward model** from human feedback

Feedback comes as **preferences over model samples:** $\mathcal{D} = \{x^i, y_w^i, y_l^i\}$

Prompt

Preferred response

Dispreferred response

Bradley-Terry Model connects rewards to preferences:

RLHF: Learning a **reward model** from human feedback

Feedback comes as **preferences over model samples:** $\mathcal{D} = \{x^i, y_w^i, y_l^i\}$

Prompt

Preferred response

Dispreferred response

Bradley-Terry Model connects rewards to preferences:

$$p(y_w \succ y_l \mid x) = \sigma(r(x, y_w) - r(x, y_l))$$

RLHF: Learning a **reward model** from human feedback

Feedback comes as **preferences over model samples:** $\mathcal{D} = \{x^i, y_w^i, y_l^i\}$

Prompt

Preferred response

Dispreferred response

Bradley-Terry Model connects rewards to preferences:

Reward assigned to **preferred** and **dispreferred** responses

$$p(y_w \succ y_l \mid x) = \sigma(r(x, y_w) - r(x, y_l))$$

RLHF: Learning a **reward model** from human feedback

Feedback comes as **preferences over model samples:** $\mathcal{D} = \{x^i, y_w^i, y_l^i\}$

Prompt

Preferred response

Dispreferred response

Bradley-Terry Model connects rewards to preferences:

Reward assigned to **preferred** and **dispreferred** responses

$$p(y_w \succ y_l \mid x) = \sigma(r(x, y_w) - r(x, y_l))$$

Train the reward model by **minimizing negative log likelihood:**

RLHF: Learning a **reward model** from human feedback

Feedback comes as **preferences over model samples:** $\mathcal{D} = \{x^i, y_w^i, y_l^i\}$

Prompt

Preferred response

Dispreferred response

Bradley-Terry Model connects rewards to preferences:

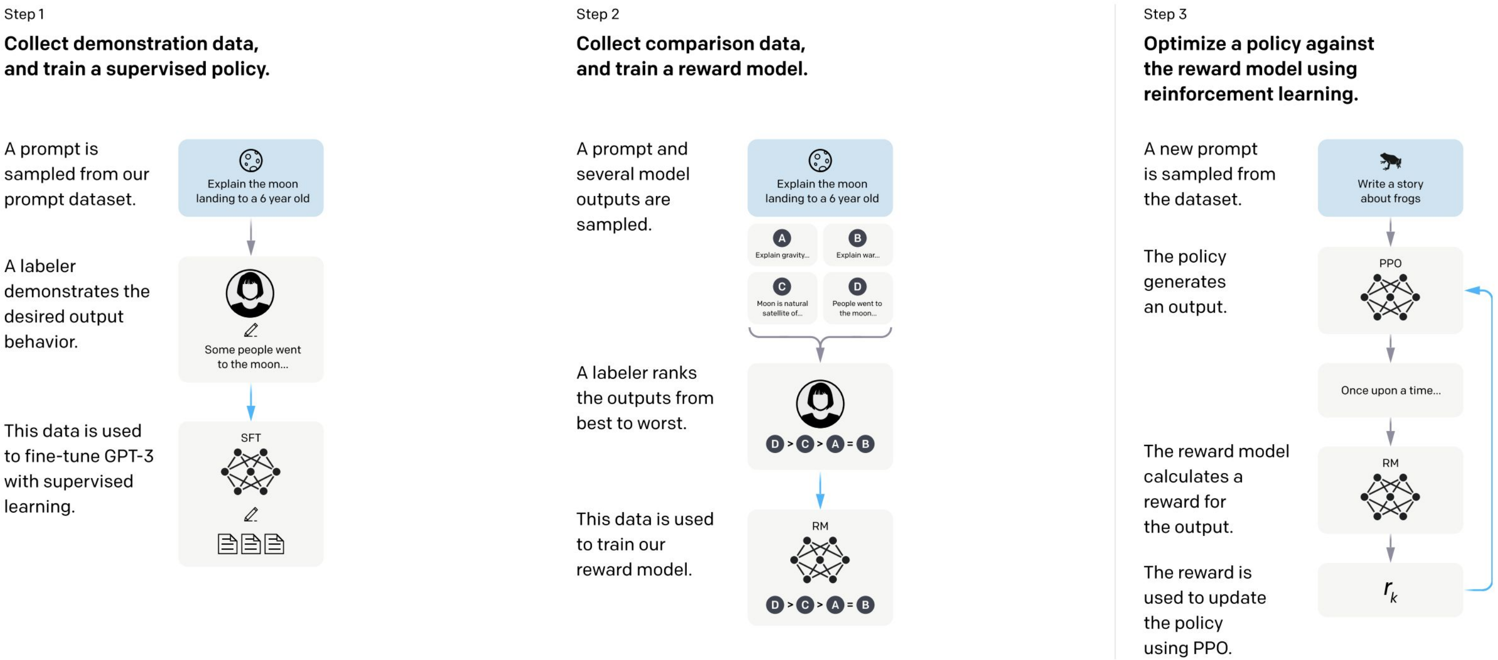
Reward assigned to **preferred** and **dispreferred** responses

$$p(y_w \succ y_l \mid x) = \sigma(r(x, y_w) - r(x, y_l))$$

Train the reward model by **minimizing negative log likelihood:**

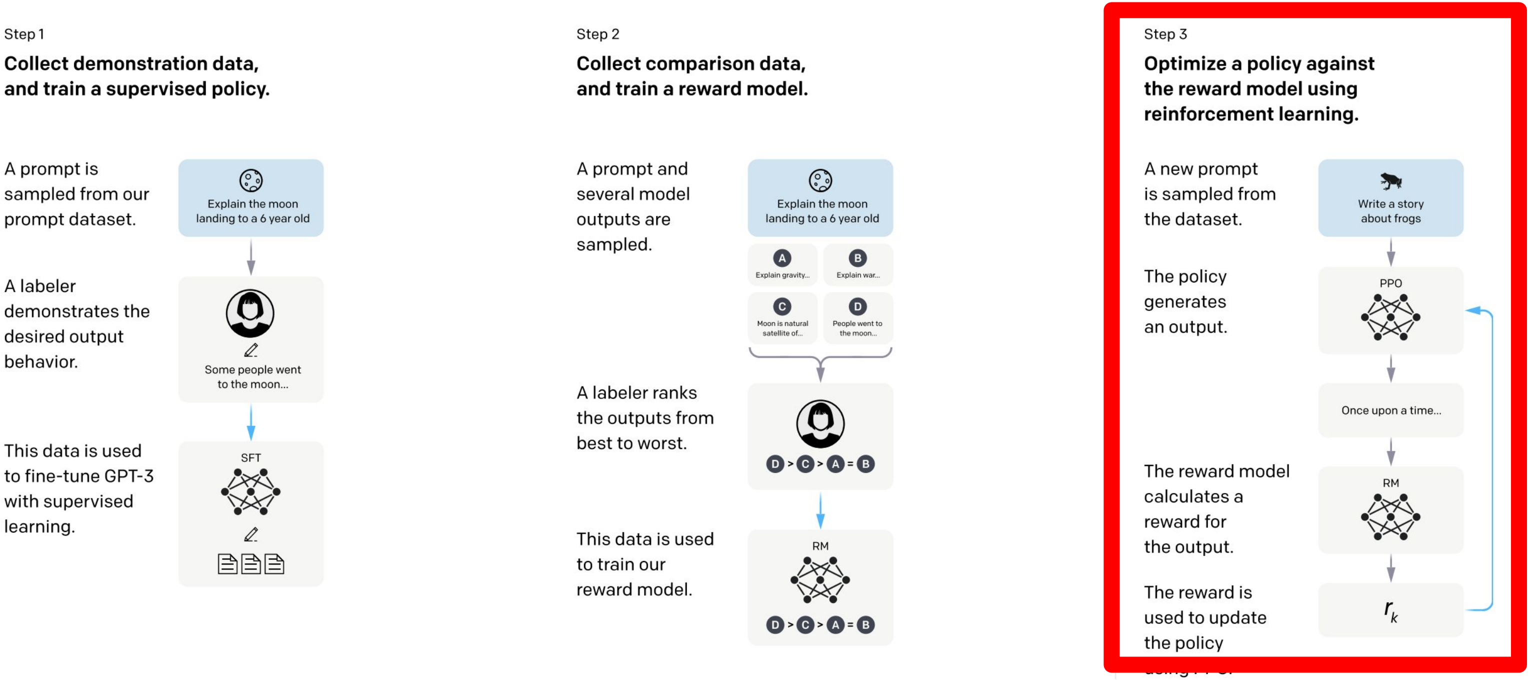
$$\mathcal{L}_R(\phi, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l)) \right]$$

RLHF: Reinforcement Learning From Human Feedback



Training language models to follow instructions with human feedback, Ouyang et. al. 2022

RLHF: Learning a **policy** that optimizes the **reward**



Training language models to follow instructions with human feedback, Ouyang et. al. 2022

RLHF: Learning a **policy** that optimizes the **reward**

Now we have a **reward model** r_ϕ that represents* **goodness according to humans**

RLHF: Learning a **policy** that optimizes the **reward**

Now we have a **reward model** r_ϕ that represents* **goodness according to humans**

Now, learn a policy π_θ achieving **high reward**

RLHF: Learning a **policy** that optimizes the **reward**

Now we have a **reward model** r_ϕ that represents* **goodness according to humans**

Now, learn a policy π_θ achieving **high reward**

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}\left[r_\phi(x, y)\right]$$

RLHF: Learning a **policy** that optimizes the **reward**

Now we have a **reward model** r_ϕ that represents* **goodness according to humans**

Now, learn a policy π_θ achieving **high reward**

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}\left[r_\phi(x, y)\right]$$

Sample from policy

Want high reward…

RLHF: Learning a **policy** that optimizes the **reward**

Now we have a **reward model** r_ϕ that represents* **goodness according to humans**

Now, learn a policy π_θ achieving **high reward** while **staying close** to original model π_{ref}

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}\left[r_\phi(x, y)\right]$$

Sample from policy

Want high reward…

RLHF: Learning a **policy** that optimizes the **reward**

Now we have a **reward model** r_ϕ that represents* **goodness according to humans**

Now, learn a policy π_θ achieving **high reward** while **staying close** to original model π_{ref}

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

Sample from policy

Want high reward…

…but keep KL to original model small!

Direct Preference Optimization

Direct Preference Optimization

RLHF Objective

(get **high reward**, stay **close** to reference model)

Direct Preference Optimization

RLHF Objective

(get **high reward**, stay **close** to reference model)

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi(y|x)}\left[r(x,y)\right] - \beta\mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid x)\|\pi_{\mathrm{ref}}(\cdot \mid x))$$

Direct Preference Optimization

any reward function

RLHF Objective

(get **high reward**, stay **close** to reference model)

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi(y|x)} \left[r(x, y)\right] - \beta \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid x) \| \pi_{\mathrm{ref}}(\cdot \mid x))$$

Direct Preference Optimization

any reward function

RLHF Objective

(get **high reward**, stay **close** to reference model)

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi(y|x)} \left[r(x,y) \right] - \beta \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid x) \| \pi_{\mathrm{ref}}(\cdot \mid x))$$

Closed-form Optimal Policy

(write **optimal policy** as function of **reward function**; from prior work)

Direct Preference Optimization

any reward function

RLHF Objective

(get **high reward**, stay **close** to reference model)

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi(y|x)}\left[r(x,y)\right] - \beta\mathbb{D}_{\text{KL}}(\pi(\cdot \mid x)\|\pi_{\text{ref}}(\cdot \mid x))$$

Closed-form Optimal Policy

(write **optimal policy** as function of **reward function**; from prior work)

$$\pi^*(y \mid x) = \frac{1}{Z(x)}\pi_{\text{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r(x,y)\right)$$

Direct Preference Optimization

any reward function

RLHF Objective

(get **high reward**, stay **close** to reference model)

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi(y|x)} \left[r(x,y)\right] - \beta \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid x) \| \pi_{\mathrm{ref}}(\cdot \mid x))$$

Closed-form Optimal Policy

(write **optimal policy** as function of **reward function**; from prior work)

$$\pi^*(y \mid x) = \frac{1}{Z(x)} \pi_{\mathrm{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x,y)\right)$$

with $Z(x) = \sum_y \pi_{\mathrm{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x,y)\right)$

Direct Preference Optimization

any reward function

RLHF Objective

(get **high reward**, stay **close** to reference model)

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi(y|x)} \left[r(x,y)\right] - \beta \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid x) \| \pi_{\mathrm{ref}}(\cdot \mid x))$$

Closed-form Optimal Policy

(write **optimal policy** as function of **reward function**; from prior work)

$$\pi^*(y \mid x) = \frac{1}{Z(x)} \pi_{\mathrm{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x,y)\right)$$

with $Z(x) = \sum_y \pi_{\mathrm{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x,y)\right)$

Note **intractable sum** over possible responses; can't immediately use this

Direct Preference Optimization

any reward function

RLHF Objective

(get **high reward**, stay **close** to reference model)

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi(y|x)} \left[r(x, y)\right] - \beta \mathbb{D}_{\text{KL}}(\pi(\cdot \mid x) \| \pi_{\text{ref}}(\cdot \mid x))$$

Closed-form Optimal Policy

(write **optimal policy** as function of **reward function**; from prior work)

$$\pi^*(y \mid x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right)$$

with $Z(x) = \sum_y \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right)$

Note **intractable sum** over possible responses; can't immediately use this

Rearrange

(write **any reward function** as function of **optimal policy**)

Direct Preference Optimization

any reward function

RLHF Objective

(get **high reward**, stay **close** to reference model)

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi(y|x)} \left[r(x,y)\right] - \beta \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid x) \| \pi_{\mathrm{ref}}(\cdot \mid x))$$

Closed-form Optimal Policy

(write **optimal policy** as function of **reward function**; from prior work)

$$\pi^*(y \mid x) = \frac{1}{Z(x)} \pi_{\mathrm{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x,y)\right)$$

with $Z(x) = \sum_y \pi_{\mathrm{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x,y)\right)$

Note **intractable sum** over possible responses; can't immediately use this

Rearrange

(write **any reward function** as function of **optimal policy**)

$$r(x,y) = \underbrace{\beta \log \frac{\pi^*(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} + \beta \log Z(x)}_{\text{some parameterization of a reward function}}$$

Direct Preference Optimization

any reward function

RLHF Objective

(get **high reward**, stay **close** to reference model)

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi(y|x)} [r(x,y)] - \beta \mathbb{D}_{\mathrm{KL}}(\pi(\cdot \mid x) \| \pi_{\mathrm{ref}}(\cdot \mid x))$$

Closed-form Optimal Policy

(write **optimal policy** as function of **reward function**; from prior work)

$$\pi^*(y \mid x) = \frac{1}{Z(x)} \pi_{\mathrm{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x,y)\right)$$

with $Z(x) = \sum_y \pi_{\mathrm{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x,y)\right)$

Note **intractable sum** over possible responses; can't immediately use this

Rearrange

(write **any reward function** as function of **optimal policy**)

Ratio is **positive** if policy likes response more than reference model, **negative** if policy likes response less than ref. model

$$r(x,y) = \underbrace{\beta \log \frac{\pi^*(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} + \beta \log Z(x)}_{\text{some parameterization of a reward function}}$$

Direct Preference Optimization: Putting it together

Direct Preference Optimization: Putting it together

A loss function on
<u>reward functions</u>

Direct Preference Optimization: Putting it together

A loss function on
reward functions

+

A transformation
between reward
functions and policies

Direct Preference Optimization: Putting it together

A loss function on
reward functions

+

A transformation
between reward
functions and policies

=

A loss function
on policies

Direct Preference Optimization: Putting it together

A loss function on
<u>reward functions</u>

+

A transformation
between <u>reward</u>
<u>functions</u> and <u>policies</u>

=

A loss function
on <u>policies</u>

Derived from the Bradley-Terry model of human preferences:

$$\mathcal{L}_R(r, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma(r(x, y_w) - r(x, y_l))\right]$$

Direct Preference Optimization: Putting it together

Derived from the Bradley-Terry model of human preferences:

A loss function on <u>reward functions</u>

$$\mathcal{L}_R(r, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma(r(x, y_w) - r(x, y_l))\right]$$

+

A transformation between <u>reward functions</u> and <u>policies</u>

$$r_{\pi_\theta}(x, y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x)$$

=

A loss function on <u>policies</u>

Direct Preference Optimization: Putting it together

A loss function on <u>reward functions</u>

Derived from the Bradley-Terry model of human preferences:

$$\mathcal{L}_R(r, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma(r(x, y_w) - r(x, y_l))\right]$$

+

A transformation between <u>reward</u> <u>functions</u> and <u>policies</u>

$$r_{\pi_\theta}(x, y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x)$$

=

Reward of **preferred** response

Reward of **dispreferred** response

A loss function on <u>policies</u>

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right]$$

Direct Preference Optimization: Putting it together

A loss function on <u>reward functions</u>

Derived from the Bradley-Terry model of human preferences:

$$\mathcal{L}_R(r, \mathcal{D}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma(r(x, y_w) - r(x, y_l))\right]$$

+

A transformation between <u>reward functions</u> and <u>policies</u>

$$r_{\pi_\theta}(x, y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x)$$

When substituting, the **log Z term cancels**, because the loss only cares about **difference** in rewards

=

A loss function on <u>policies</u>

Reward of **preferred** response

Reward of **dispreferred** response

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right]$$

Direct Preference Optimization: Putting it together

Derived from the Bradley-Terry model of human preferences:

A loss function on <u>reward functions</u>

$$\mathcal{L}_R(r, \mathcal{D}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma(r(x, y_w) - r(x, y_l))\right]$$

+

A transformation between <u>reward</u> <u>functions</u> and <u>policies</u>

$$r_{\pi_\theta}(x, y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x)$$

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right]$$

Reward of **preferred** response

Reward of **dispreferred** response

RLHF Pipeline and DPO

- <u>**Key takeaway:**</u> **Direct Preference Optimization (DPO)** fits an implicit reward function and optimizes for a good policy.
- DPO has a better performance than PPO (PPO + explicit reward function fitting)
- DPO is more robust to reward hacking as compared to PPO (PPO + explicit reward function fitting)

"https://web.stanford.edu/class/cs234/CS234Spr2024/slides/dpo_slides.pdf"

Part 3

- Limitations
- Practical aspects: the optimal design problem and credit assignment

Limitations

- BTL model
  - Does not allow non-transitive preferences

  Some may like apples over bananas, oranges over apples but bananas over oranges.

  - Even if individuals have transitive preferences, then the expected preferences may not be transitive.

  How to aggregate preferences over a population but still personalize.

Limitations

- BTL model
  - Does not allow non-transitive preferences
  - Even if individuals have transitive preferences, then the expected preferences may not be transitive.
- RLHF pipeline/DPO pipeline
  - Success depends heavily on the choice of π_{ref}

  This was the policy of the already fine tuned foundation model.
  - Hard to quantify how good is the optimized policy wrt π_{ref}

  How useful is the RLHF/DPO pipeline?

Limitations

- BTL model
  - Does not allow non-transitive preferences
  - Even if individuals have transitive preferences, then the expected preferences may not be transitive.
- RLHF pipeline/DPO pipeline
  - Success depends heavily on the choice of π_{ref}
  - Hard to quantify how good is the optimized policy wrt π_{ref}
- About Data acquisition
  - Which contexts/prompts need fine tuning and what token sequences to offer to humans to label?
  - The human labeling is costly and slow.

Limitations

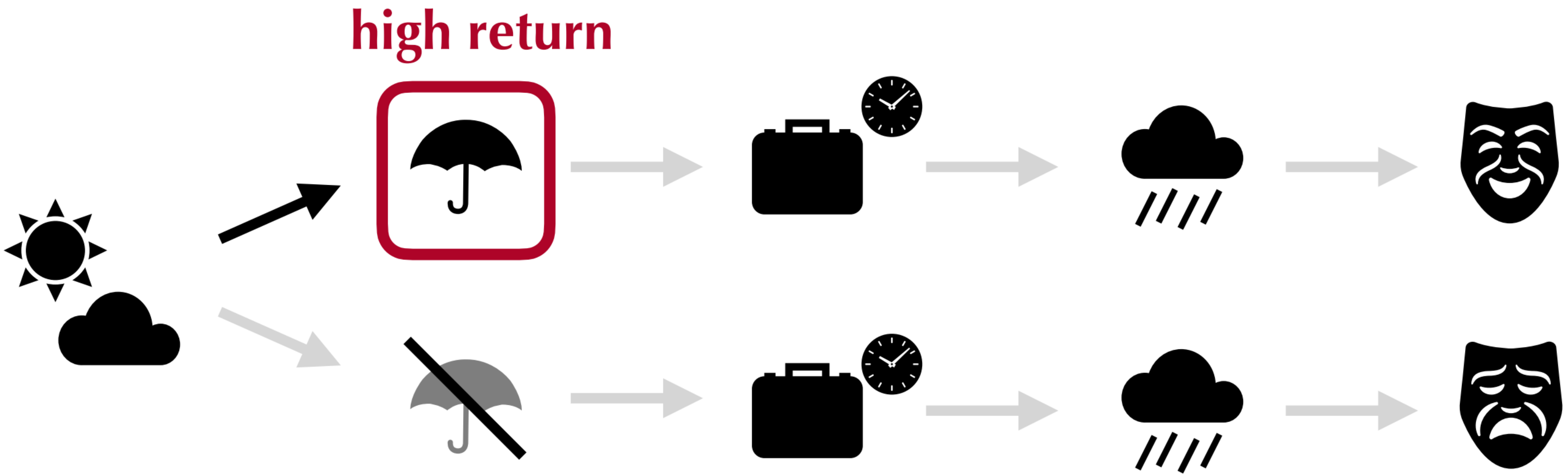
The issue of Hindsight Credit assignment: Broader limitation with preference modeling.

Slides from “https://runzhe-yang.science/princeton-cs/demo/hindsight_credit_assignment.pdf”

Value Function Problem

$$V^\pi(x) \stackrel{\text{def}}{=} \mathbb{E}_{\tau\sim\mathcal{T}(x,\pi)}\Big[Z(\tau)\Big], \qquad Q^\pi(x,a) \stackrel{\text{def}}{=} \mathbb{E}_{\tau\sim\mathcal{T}(x,a,\pi)}\Big[Z(\tau)\Big].$$

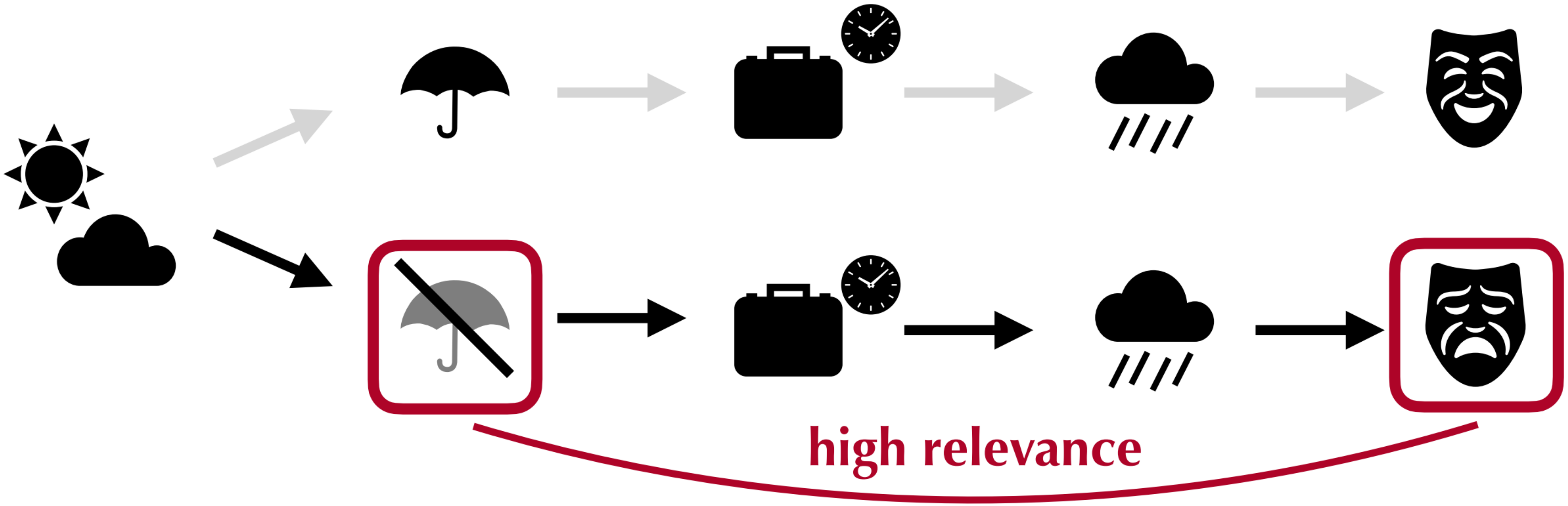
"how does the **current action** affect **future outcomes**?"



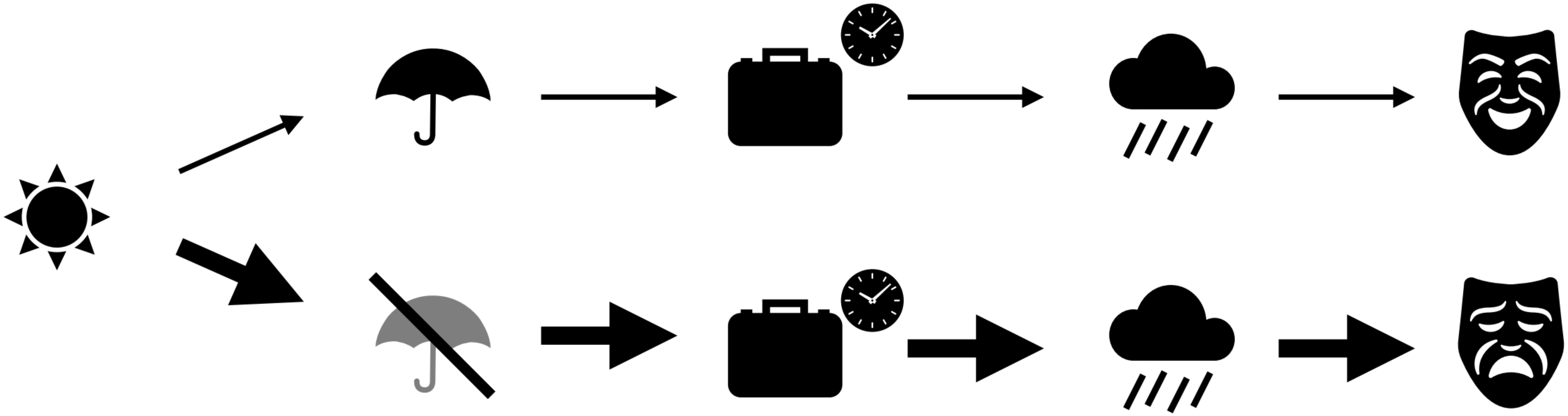
Credit Assignment Problem

$$I(A_t; f(\tau_{t:\infty})|X_t = x) = \mathbb{E}_{\tau\sim\mathcal{T}(x,\pi)}\left[\log\left(\frac{\mathbb{P}(A = A_t|f(\tau) = f(\tau_{t:\infty}), X_t = x)}{\mathbb{P}(A = A_t|X_t = x)}\right)\right]$$

"given an **outcome**, how *relevant* were **past decisions**?"

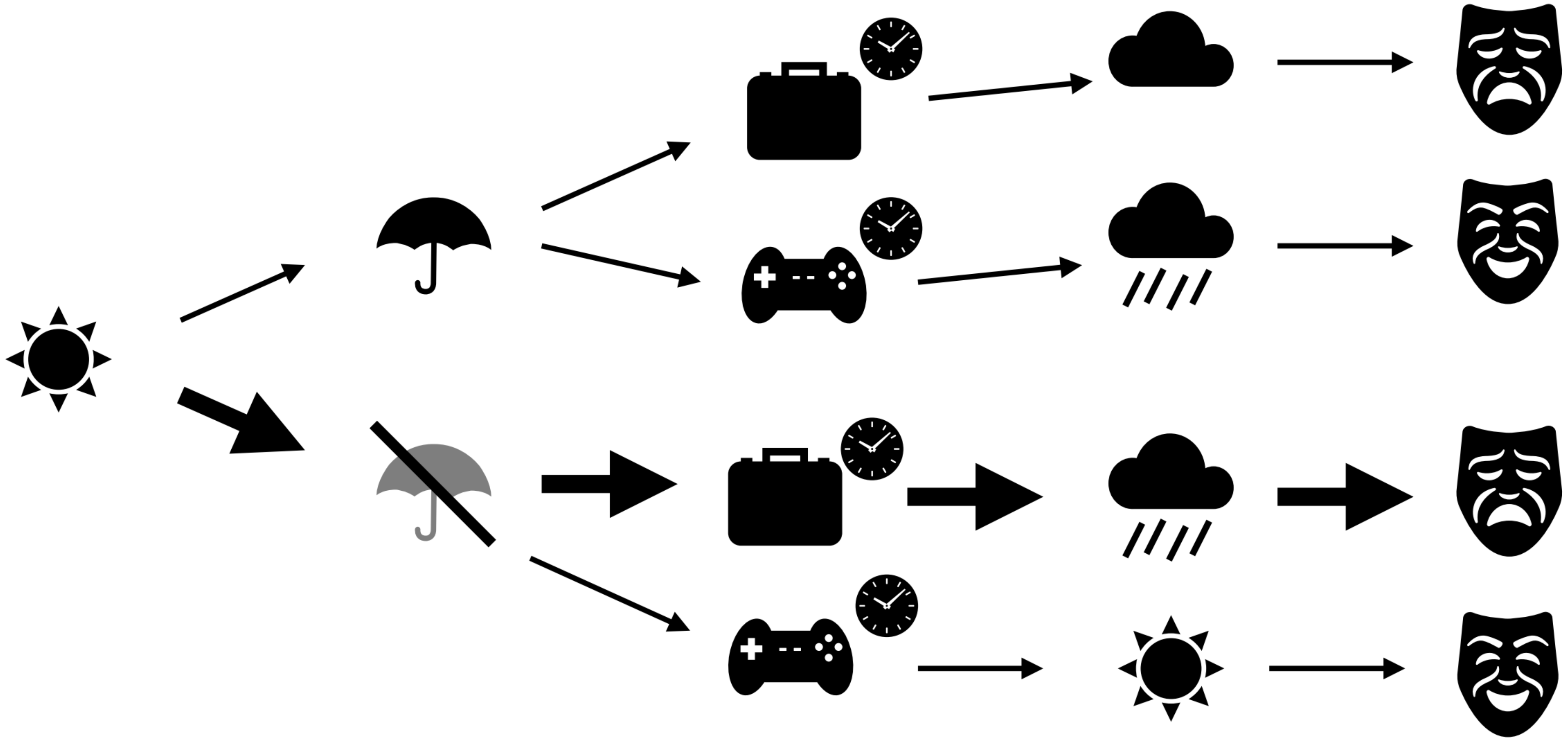


Credit Assignment Problem - Why is it important?



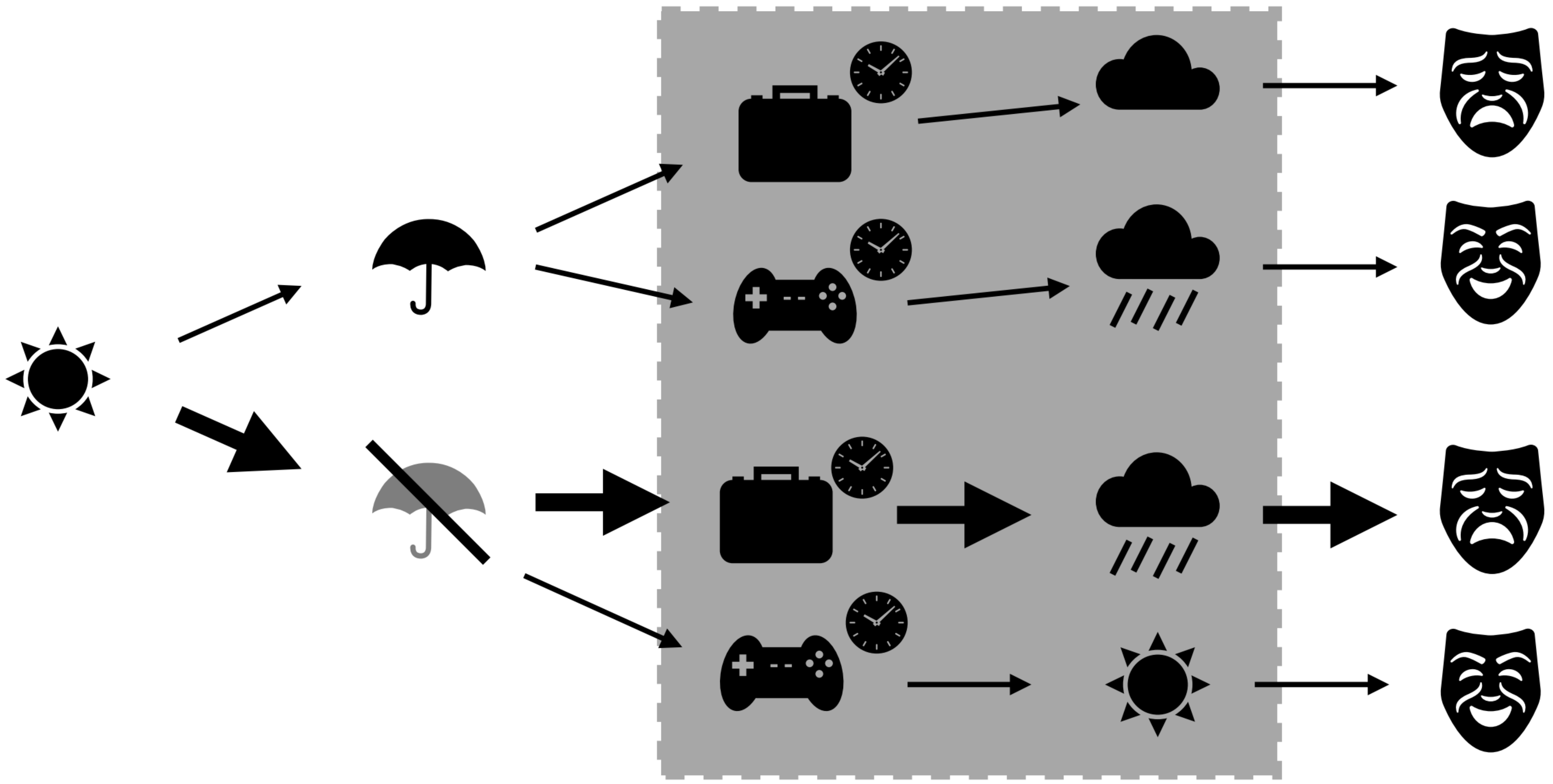
Rare events require an infeasible number of samples to obtain an accurate estimate.

Credit Assignment Problem - Why is it challenging?



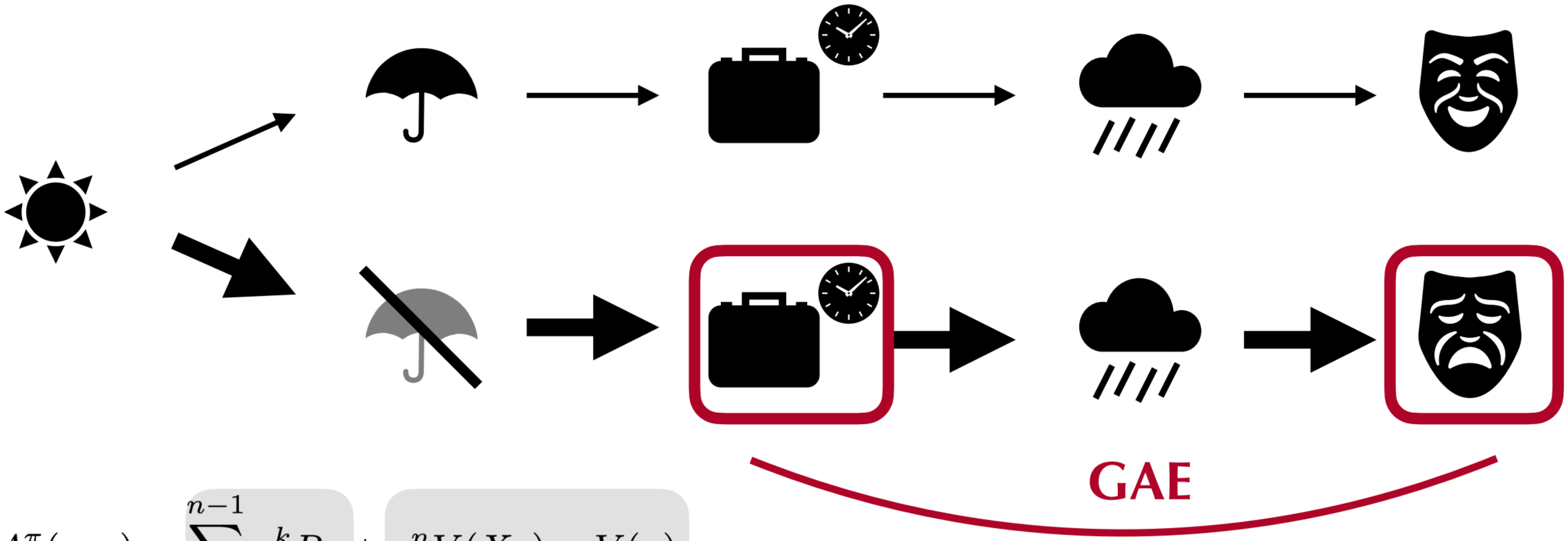
Issue 1: Variance - low sample efficiency

Credit Assignment Problem - Why is it challenging?



Issue 2: Partial observability - cannot bootstrap.

Credit Assignment Problem - Why is it challenging?

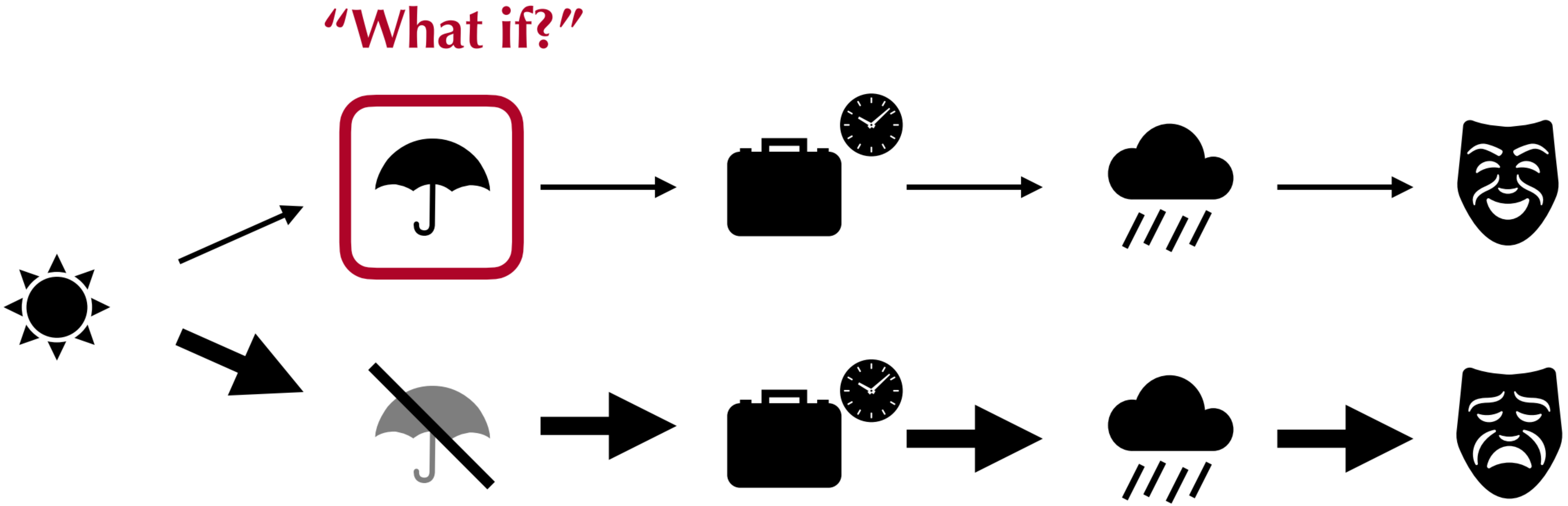


$$A^\pi(x,a) \approx \underbrace{\sum_{k=0}^{n-1}\gamma^k R_k}_{\text{variance}} + \underbrace{\gamma^n V(X_n) - V(x).}_{\text{bias}}$$

—> best *n*?

Issue 3: Time as a proxy - rely on *time* as the sole metric.

Credit Assignment Problem - Why is it challenging?



Issue 4: No counterfactuals - only update actions serendipitously occur.

Part 4

- Conclusion and after thoughts

Conclusion and After-thoughts

<u>Summary</u>

- Preference feedback is powerful way of awarding rewards in RL.
- PPO is a good, policy gradient style algorithm.
- RLHF in the context of LLMS refer to a pipeline of three step procedure useful in fine-tuning the foundation model for a specific task
- DPO is an efficient way of combining the steps 2 and 3 of RLHF for LLM.