RLHF

Reinforcement **L**earning from **H**uman **F**eedback

昇思 MindSpore

ChatGPT背后的“功臣”

Methods

We trained this model using Reinforcement Learning from Human Feedback (RLHF), using the same methods as InstructGPT, but with slight differences in the data collection setup. We trained an initial model using supervised fine-tuning: human AI trainers provided conversations in which they played both sides—the user and an AI assistant. We gave the trainers access to model-written suggestions to help them compose their responses. We mixed this new dialogue dataset with the InstructGPT dataset, which we transformed into a dialogue format.

RLHF的效果如何——from ChatGPT

Conversation example：movie recommendation

- acknowledges the repetition and apologizes for it.
- provides more detailed information about the recommended movies

Conversation with a Machine without RLHF:

User: Hello, can you recommend a good movie to watch tonight?

Machine: Sure! What genre are you in the mood for?

User: I'm in the mood for a comedy.

Machine: How about "The Hangover"? It's a hilarious comedy that follows a group of friends who wake up from a wild night in Las Vegas and can't remember what happened.

User: Oh, I've already seen that one. Do you have any other recommendations?

Machine: How about "Bridesmaids"? It's a funny movie about a woman who becomes the maid of honor for her best friend's wedding and all the crazy events that ensue.

User: That sounds interesting. I'll give it a try. Thanks!

Conversation with a Machine modified by RLHF:

User: Hello, can you recommend a good movie to watch tonight?

Machine: Sure! What genre are you in the mood for?

User: I'm in the mood for a comedy.

Machine: How about "The Hangover"? It's a hilarious comedy that follows a group of friends who wake up from a wild night in Las Vegas and can't remember what happened.

User: Oh, I've already seen that one. Do you have any other recommendations?

Machine: I apologize for the repetition. Based on your preference for comedy, I suggest "Bridesmaids." It's a funny movie about a woman who becomes the maid of honor for her best friend's wedding and all the crazy events that ensue. It has received positive reviews and is known for its witty dialogue and comedic performances.

User: Thanks for the suggestion. Is there anything else you can recommend?

Machine: Certainly! If you're open to a different genre, I highly recommend "Inception." It's a mind-bending science fiction thriller that explores the concept of shared dreaming. The film is directed by Christopher Nolan and has garnered critical acclaim for its complex narrative and stunning visual effects.

User: Oh, I've heard about "Inception" before. It sounds intriguing. I think I'll give it a try. Thanks for the recommendation!

RLHF的效果如何——from previous work

Summarization example：reddit post

Prompt:

To pursue a Computer Sc.PhD or continue working? Especially if one has no real intention to work in academia even after grad school .

Vanilla LM:

I’ m considering pursuing a PhD in Computer Science, but I'm worried about the future. I’ m currently employed full-time, but I'm worried about the future.

+

Human Annotation:

Software Engineer with a job I’ m happy at (for now), deciding whether to pursue a PhD to improve qualifications and explore interests and a new challenge.

=

RLHF Model:

Currently employed, considering pursuing PhD in Computer Science to avoid being stuck wih no residency visa ever again . Has anyone pursued a PhD purely for the sake of research, with no intention of joining the academic world?

昇思 MindSpore

RLHF：RL & HF

Reinforcement Learning from Human Feedback

基于人工反馈的强化学习

课程目录

- RLHF简介
- 强化学习Reinforcement Learning
  - supervised learning vs reinforcement learning
  - RL中的基础概念：agent、environment；action、state、reward
  - policy gradient算法
  - PPO算法
- 人类反馈
  - reward model

RLHF简介

让语言模型变得更有“人情味”

Instruction Tuning的局限性

Instruction Finetuning

Pros:
- Simple and straightforward, generalize to unseen tasks

Cons:
- **Expensive** to collect gound-truth data for tasks
- No right answer to tasks **open-ended creative generation**
- **Equal penalization** to all token-level mistakes (even though some are worse than others)

still cannot maximize human preferences

RLHF流程

Step 1：预训练/微调一个语言模型（LM）

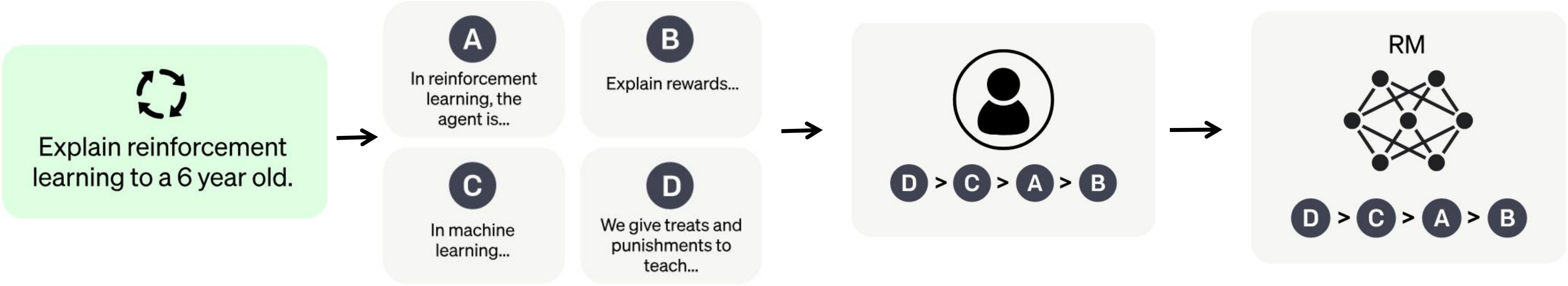
A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

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

RLHF流程

Step 2：基于人类偏好，训练一个打分模型给语言模型的输出进行打分



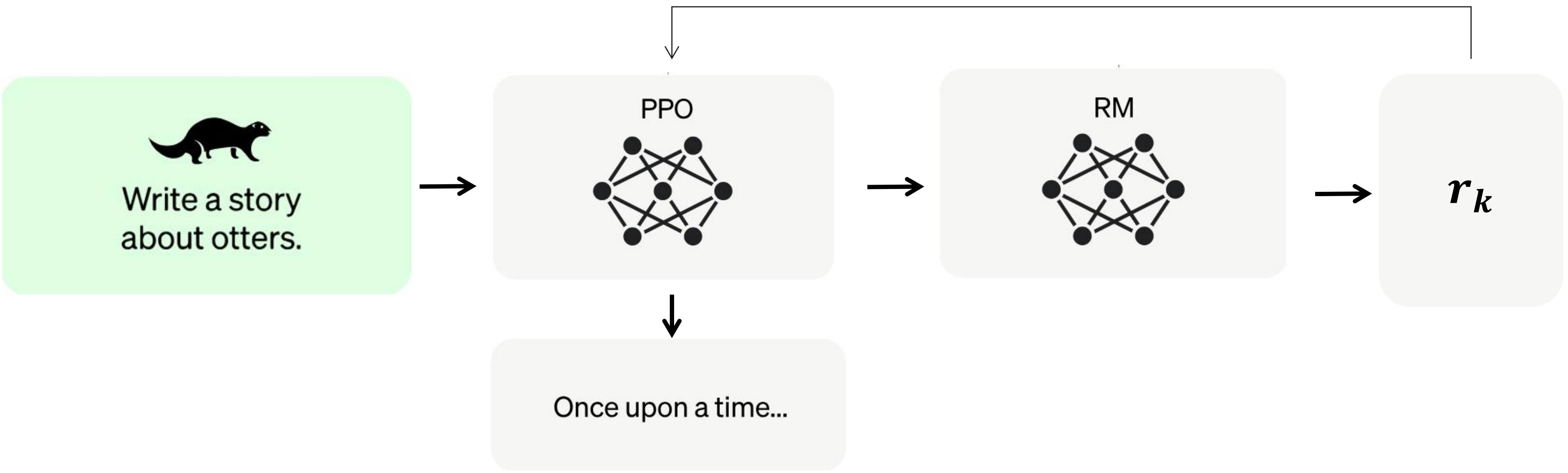
A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

RLHF流程

Step 3：基于人类偏好打分，使用强化学习（RL）微调语言模型（LM）



A new prompt is sampled from the dataset.

- The PPO model is initialized from the supervised policy.
- The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

RLHF流程

RLHF流程的重点在两个地方：

- 语言模型的强化学习微调是怎样的？
- 奖励模型（Rward Model）怎么训练出来的？人类偏好的打分数据如何处理？

Intrduction to RL

用于对复杂问题进行决策的机器学习方法

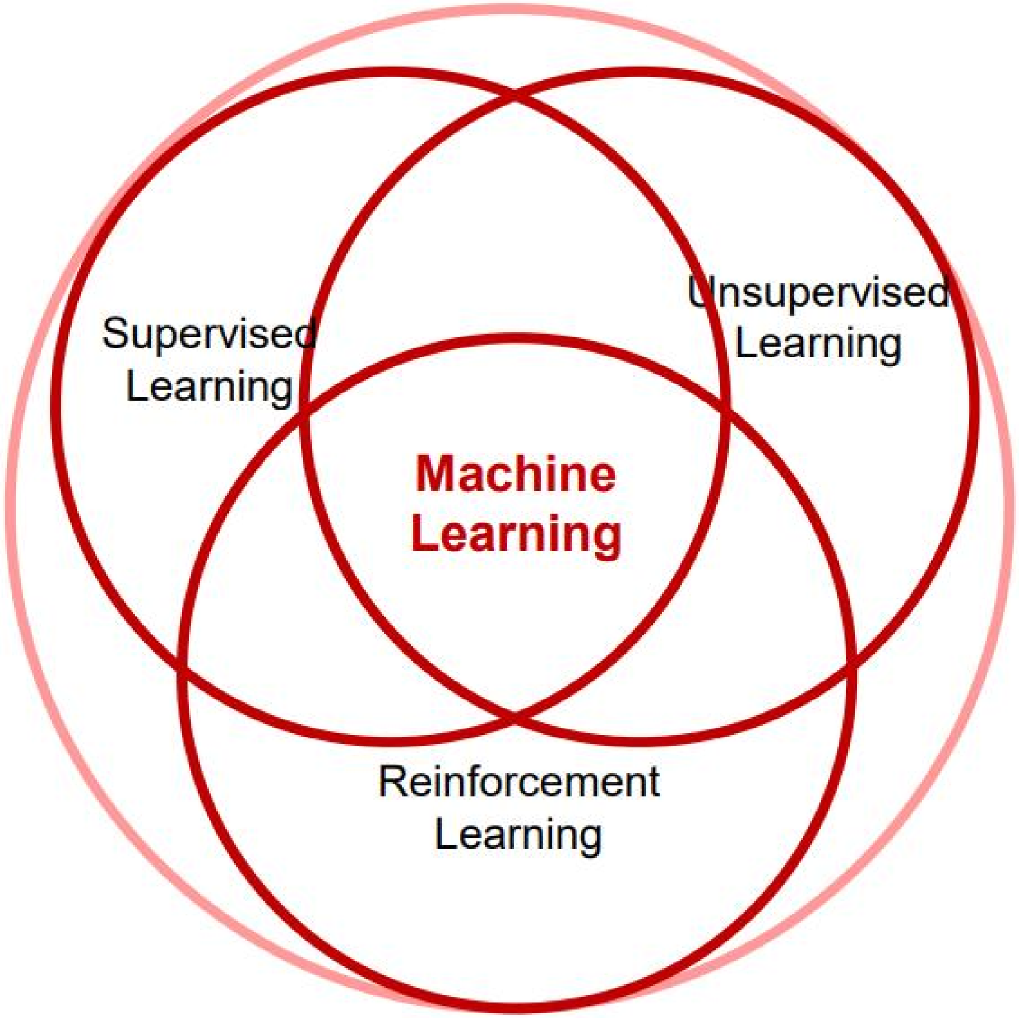
机器学习的三大领域

supervised learning：

- 基于**数据集**完成训练
- 数据集需要**标注**，计算标签和预测之间的**误差（损失）**
- 训练过程中预测的对错可以**直接实时得到**

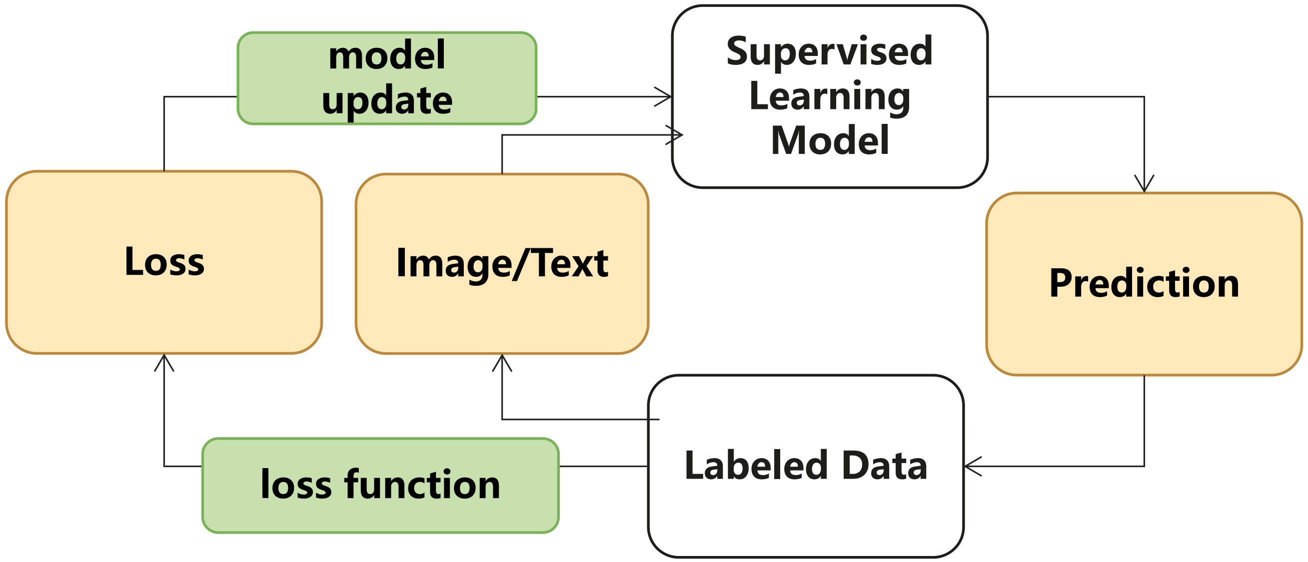
reinforcement learning：

- 不需要标注数据集，需要**环境（游戏）**
- 和环境交互中，由环境反馈**奖励信号**
- 环境反馈的奖励信号是**延迟的**
- 智能体的动作会**影响到和后续的状态和数据**



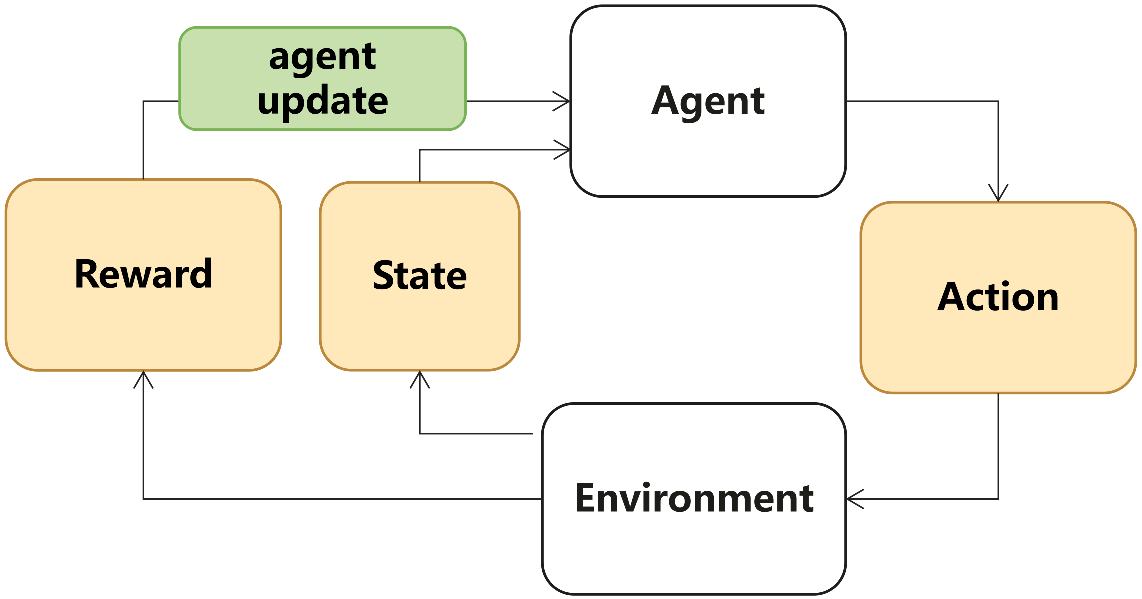
Supervised learning vs reinforcement learning

supervised learning



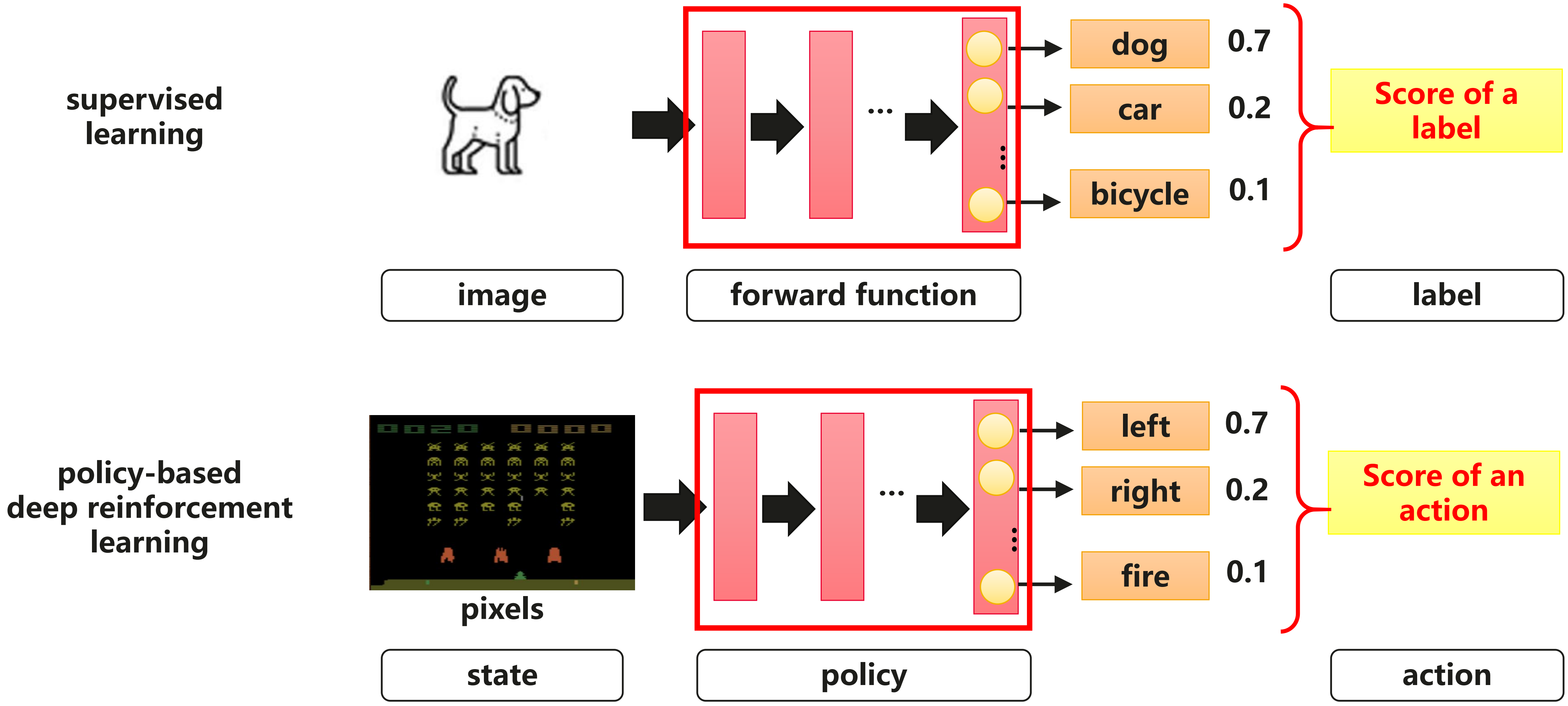
objective:
maximise the likelihood between logit and label

reinforcement learning



objective:
maximise the reward

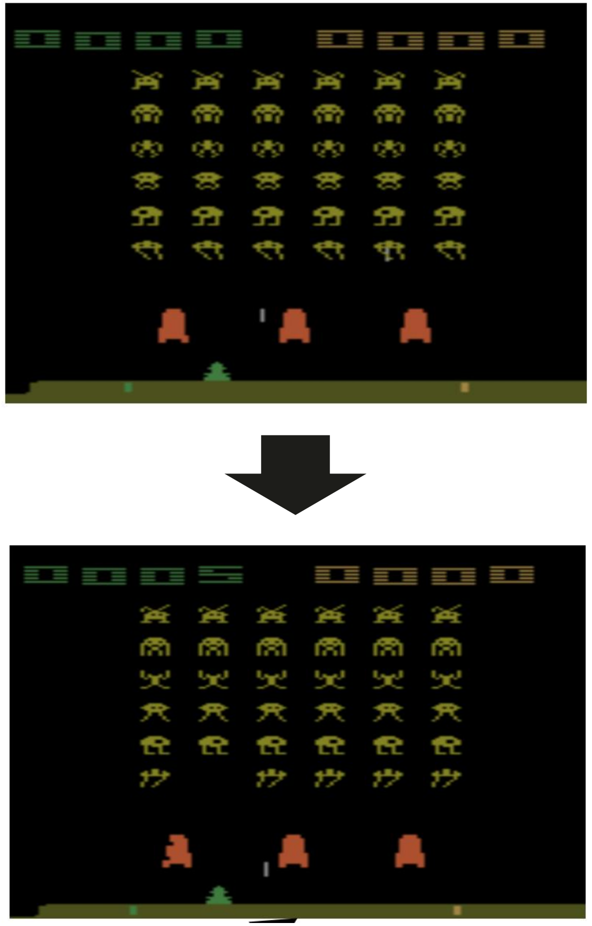
Supervised learning vs reinforcement learning



强化学习（Reinforcement learning）

强化学习是一种用于**决策（decision making）**的机器学习方法

智能体（agent）进行某个动作	飞船开火
状态（state）受到动作的影响，发生改变	敌人被击中
环境反馈**奖励信号（reward）**	成功击落敌人，积1分
不断重复上述步骤，直到完成一个**episode**	不断重复，直至飞创坠毁，游戏结束
目标：选择能够将reward per episode最大化的动作	目标：最大化平均一轮游戏的得分



任务复杂，没有固定获胜的唯一解

为什么要用强化学习优化语言模型?

人类偏好中有一些比较抽象的概念因人而异，不好定义，没办法计算”损失“

- *what is funny?*
- *what is ethical?*
- *what is safe?*
- *what is ‘sound natural to human’*

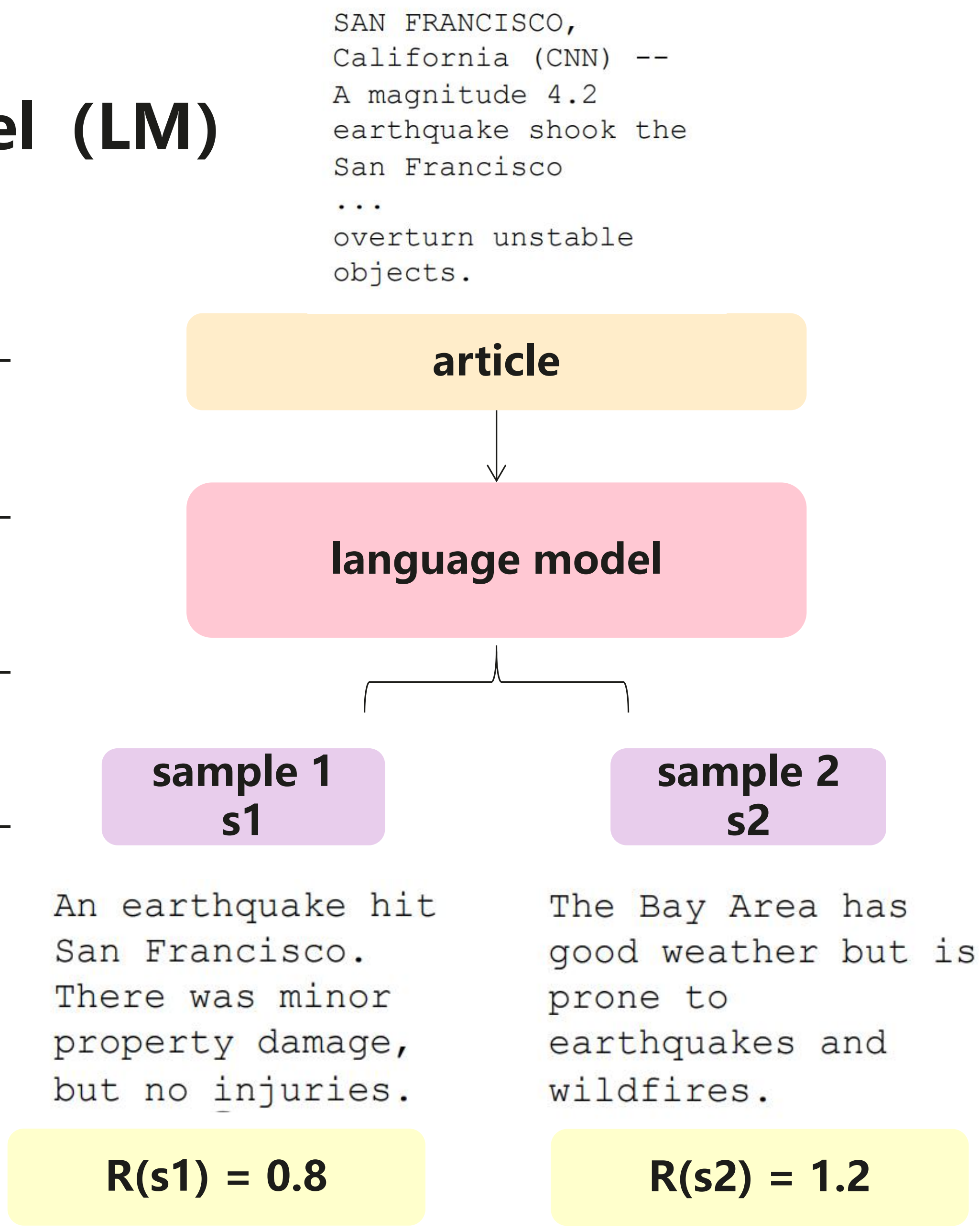
这种针对复杂问题的训练优化可以交给强化学习解决。

RL for language model（LM）

智能体（agent）进行某个动作	语言模型基于prompt进行回答
状态（state）受到动作的影响，发生改变	人类接收到模型的对话
环境反馈**奖励信号（reward）**	根据人类偏好回答打分
不断重复上述步骤，直到完成一个**episode**	让模型不断生成回复，直到收集到足够多的样本数据
目标：最大化reward per episode	目标：最大化reward per prompt

The language model is optimized by maximizing the expected reward of samples.

$$\mathbb{E}_{\hat{s}\sim p_\theta(s)}[R(\hat{s})] \;=\; \frac{1}{m}\sum_{1}^{M} R(s_i)$$



Policy gradient and PPO

value function, policy, model

一个智能体（agent）会包含以下至少一种元素

- **value function**：how good is each state and/or action
- **policy**：agent’s behavior function
- **model**：agent’s representation of the environment

估计在当前state（进行某个action）好不好

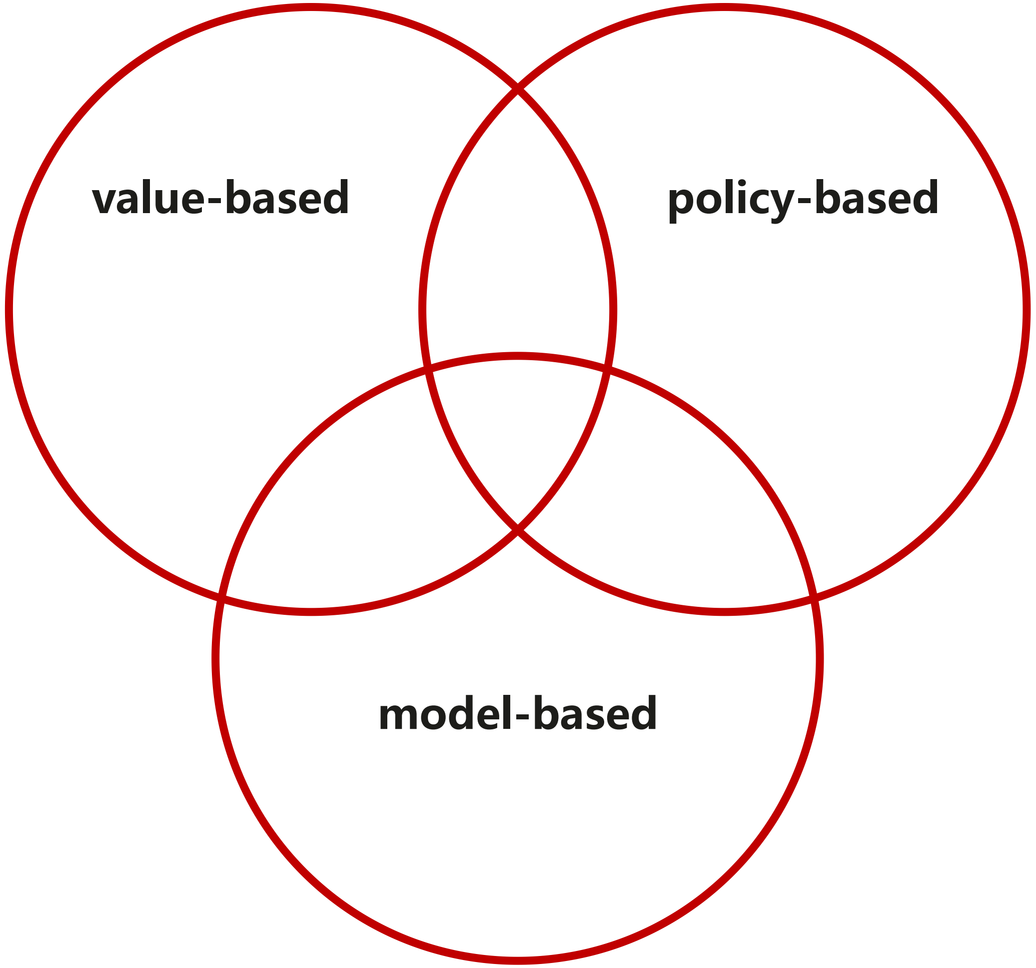
一般会用一个数值（value表示）

模拟智能体如何对动作进行决策

猜测环境如何对action进行交互

强化学习分类

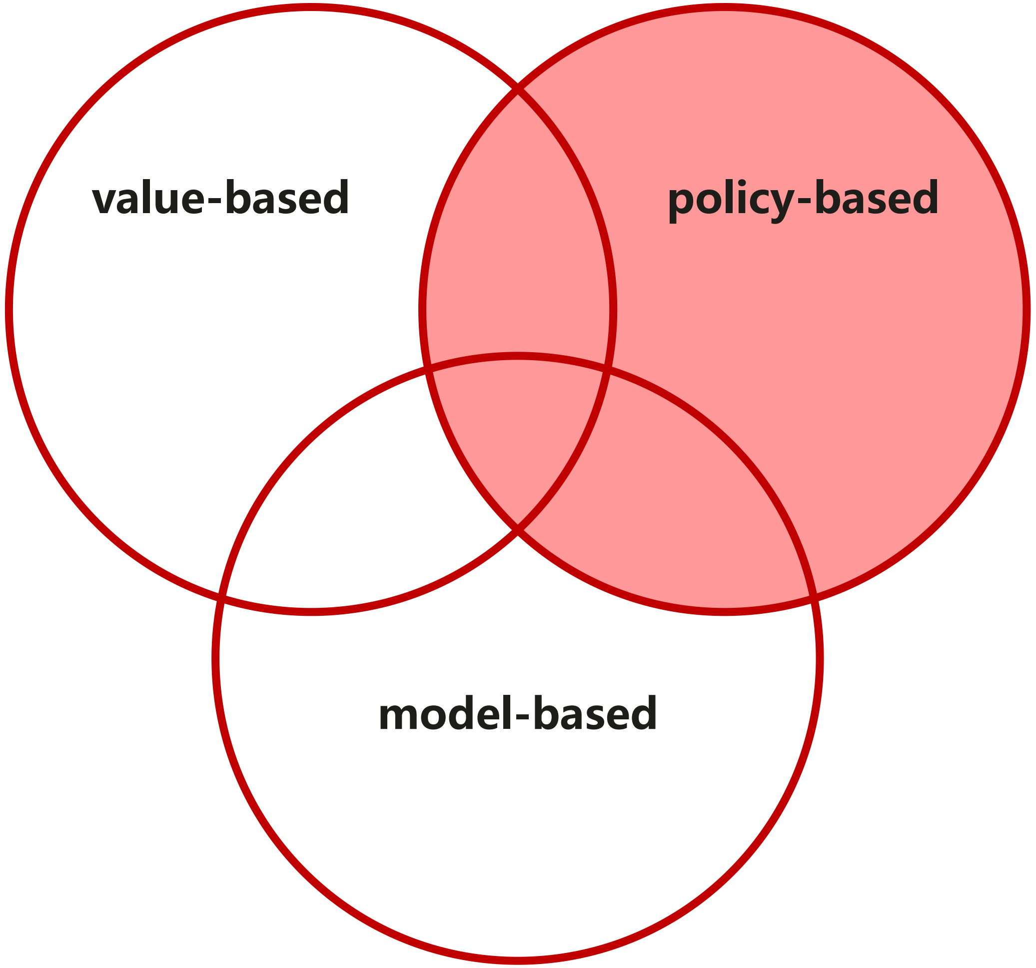
- **value-based：**
  - 根据当前的state，比较每个动作的期待值（Q-value），选择最大的Q对应的作为本次选择动作
  - 最大化期待值$Q^*(s,a)$
  - value function, no policy(implicit)
- **policy-based：**
  - 根据当前的state，计算每个动作对应的概率p(s, a)，根据概率选取动作
  - 选择可以达到最大reward的策略π^*
  - policy, no value function
- **model-based：**
  - 对环境进行建模，根据猜测环境反馈决定是否进行某个动作
  - policy and/or value function, model



强化学习分类

- **value-based：**
  - 根据当前的state，比较每个动作的期待值（Q-value），选择最大的Q对应的作为本次选择动作
  - 最大化期待值$Q^*(s,a)$
  - value function, no policy(implicit)
- **policy-based：**
  - 根据当前的state，计算每个动作对应的概率p(s, a)，根据概率选取动作
  - 选择可以达到最大reward的策略π^*
  - policy, no value function
- **model-based：**
  - 对环境进行建模，根据猜测环境反馈决定是否进行某个动作
  - policy and/or value function, model

policy gradient



Policy gradient

- Supervised learning: use **gradient descent** method to **minimize the loss**
- Reinforcement learning: use **gradient ascent** method to **maximize the reward**
  - $\boldsymbol{\theta}$:parameter of model
  - $\boldsymbol{p_\theta(s_i)}$**:** probability of the ith language model sample
  - **m:** number of iterations

$$\theta^{new} \leftarrow \theta^{old} + \eta \nabla \bar{R}_{\theta^{old}}$$

$$\nabla \overline{R_\theta} = \mathbb{E}_{\hat{s} \sim p_\theta(s)}[R(\hat{s})\, \nabla_\theta \log p_\theta(\hat{s})] \approx \frac{1}{m} \sum_{i=1}^{m} R(s_i)\, \nabla_\theta \log p_\theta(s_i)$$

The objective of reinforcement learning is to reinforce good actions by increasing the probability they happen again.

If R is +++ → **good action** → **take gradient steps to maximize $p_\theta(s_i)$**

If R is --- → **bad action** → **take gradient steps to minimize $p_\theta(s_i)$**

RL algorithm - Proximal Policy Optimization (PPO)

Policy Gradient → On-policy → Off-policy → Add constraint

- on-policy: The model being optimized and the model interacting with the human is **the same**.
  - **Have to recollect samples every time the model is updated**
- off-policy: The model being optimized and the model interacting with the human is **different**.
  - **No need to recollect samples**

on-policy to off-policy

$$\nabla \overline{R}_\theta = E_{\hat{s}\sim p_{\theta'}(s)}\left[\frac{p_\theta(s)}{p_{\theta'}(s)} R_{\theta'}(\hat{s}) \nabla log p_\theta(\hat{s})\right]$$

$$\overline{R}_\theta = E_{\hat{s}\sim p_{\theta'}(s)}\left[\frac{p_\theta(s)}{p_{\theta'}(s)} R_{\theta'}(\hat{s})\right]$$

Importance sampling: use a different distribution $p_{\theta'}(s)$ to model $p_\theta(s)$

add constraint

$$\overline{R}_\theta^{PPO} = \overline{R}_\theta - \beta KL(\theta, \theta')$$

KL divegence: penalize the divergence between θ, θ'

PPO2

$$\overline{R}_\theta^{PPO2} \approx \sum min\left(\frac{p_\theta(s)}{p_{\theta'}(s)} R_{\theta'}(s_i), clip\left(\frac{p_\theta(s)}{p_{\theta'}(s)}, 1-\varepsilon, 1+\varepsilon\right) R_{\theta'}(s_i)\right)$$

Human feedback

在评价标准中加入人类的视角

Human preference modelling

R(s): arbitrary, non-differentiable reward function

- **problem 1**: expensive human-in-loop
- **solution 1**: model human preferences as a separate NLP problem
- **problem 2**: noisy human feedbacks
- **solution 2**: use pair-wise comparison instead of direct scoring

An earthquake hit San Francisco. There was minor property damage, but no injuries.

s_1

$$R(s_1) = 8.0$$

The Bay Area has good weather but is prone to earthquakes and wildfires.

s_2

$$R(s_2) = 1.2$$

A 4.2 magnitude earthquake hit San Francisco, resulting in massive damage.

s_3

$$R(s_3) = \quad 4.1? \quad 6.6? \quad 3.2?$$

Human preference modelling

R(s): arbitrary, non-differentiable reward function

- **problem 1**: expensive human-in-loop
- **solution 1**: **model human preferences** as a separate NLP problem
- **problem 2**: noisy human feedbacks
- **solution 2**: ask for **pair-wise comparison** instead of direct scoring

An earthquake hit San Francisco. There was minor property damage, but no injuries.	>	A 4.2 magnitude earthquake hit San Francisco, resulting in massive damage.	>	The Bay Area has good weather but is prone to earthquakes and wildfires.
s_1		s_3		s_2

Thanks