Reinforcement Learning from Human Feedback

Leqi Liu
Princeton Language & Intelligence
University of Texas, Austin
April 16th, 2024

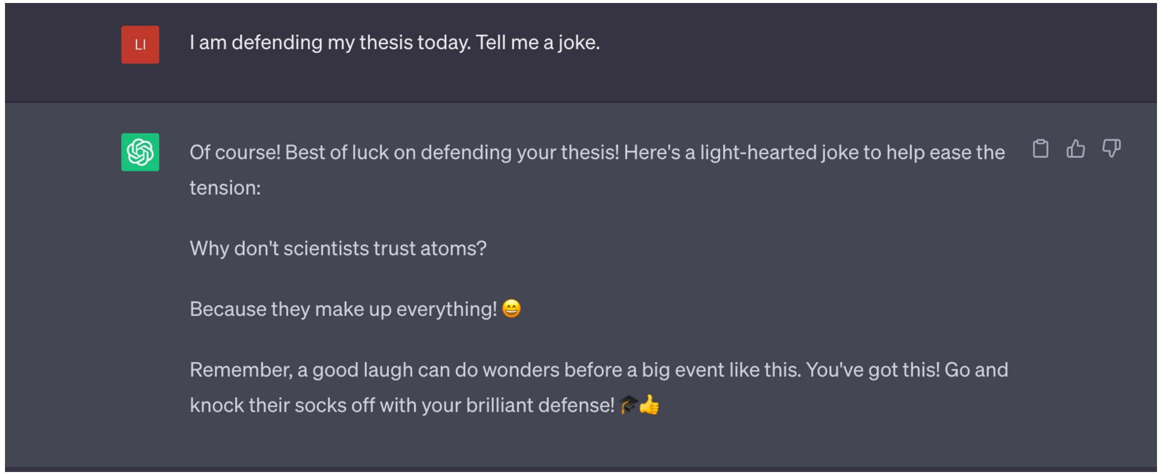
PRINCETON Language + Intelligence

Outline

- Broader landscape of Learning from Human Feedback
- Two case studies:
  - Language Modeling: **RLHF** for aligning model with **user intent**
  - Recommender systems: **Multi-armed bandits** accounting for **evolving** human preferences
- What's next?

Learning from Human Feedback

Recommender systems



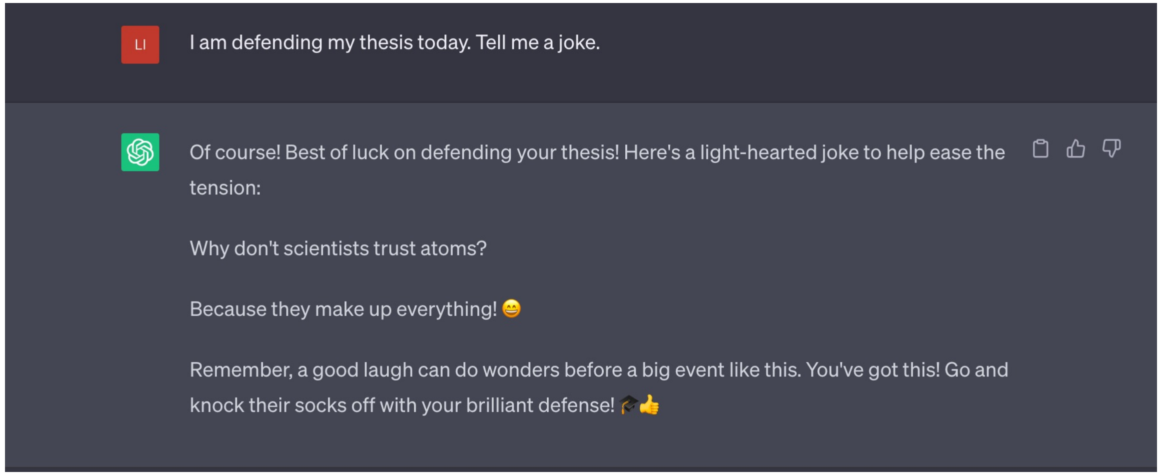
Chatbots/Language Models

Decision support systems

Self-driving cars

Learning from **Human Feedback**

User clicks, watch time…



Upvote vs not, Ending conversation,…

Expert decision, …

Driver behavioral pattern, …

Learning from **Human Feedback**

- Why do we need human feedback?

Learning from **Human Feedback**

- Why do we need human feedback?
  - Enhance the **capability** of the model
  - Improve the **utility** of the model: ML systems are deployed to interact with human users.
    - How users actually use the model? Personalization?
  - Address **safety**-related concerns: hope to align to human preferences (e.g., in the LM case)

Learning from **Human Feedback**

- Why do we need human feedback?
- What are the forms of human feedback?

Learning from **Human Feedback**

- Why do we need human feedback?
- What are the forms of human feedback? **Diverse, Rich**
    - Demonstration
    - Preference/ranking
    - Uncertainty
    - Language feedback
    - …

Learning from **Human Feedback**

- Why do we need human feedback?
- What are the forms of human feedback?
- How to collect "good" human feedback?

Learning from **Human Feedback**

- Why do we need human feedback?
- What are the forms of human feedback?
- How to collect "good" human feedback?
  - Inter-rater consistency
  - Demonstration: in the case of instruction following, what's a good way to collect (instruction, response) pairs?
  - Active query?
  - Who should we collect the data from?
  - …

Learning from **Human Feedback**

- Why do we need human feedback?
- What are the forms of human feedback?
- How to collect "good" human feedback?
- How to use human feedback?

Learning from **Human Feedback**

- Why do we need human feedback?
- What are the forms of human feedback?
- How to collect “good” human feedback?
- How to use human feedback? Feedback-type, application and goal dependent!

	Language Model	Demonstration Preference/ranking	RLHF
	Recommender System	Rating	Bandit algorithm

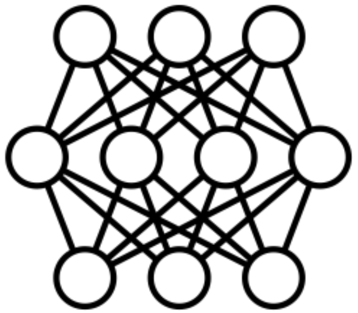
Outline

- Broader landscape of Learning from Human Feedback
- Two case studies:
  - Language Modeling: **RLHF** for aligning model with **user intent**
    - Brief overview of Language Models
    - How is it connected to RL?
    - Algorithmic space of RLHF
    - What’s next?
  - Recommender systems: Multi-armed bandits accounting for evolving human preferences
- What’s next?

LM: Next-token predictor

- Language models:

I am in Washington D.C. to attend a



P(workshop) = 0.1
P(conference) = 0.05
P(cherry) = 0.00001
…

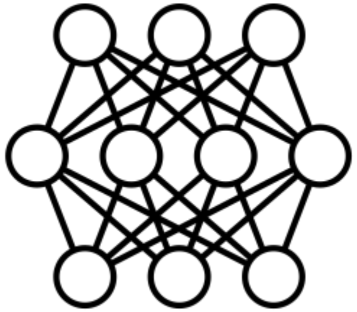
Main-stream architecture: **Transformer**

LLM: Next-token predictor

- Language models:

I am in Washington D.C. to attend a

→



→

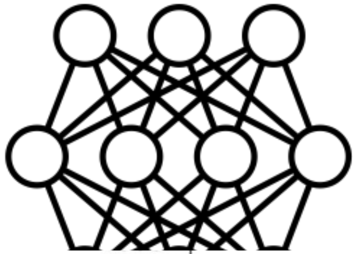
P(workshop) = 0.1
P(conference) = 0.05
P(cherry) = 0.00001
…

- Large:
  - Large amount of **data**: even small open-source models are trained on trillions of tokens.

LLM: Next-token predictor

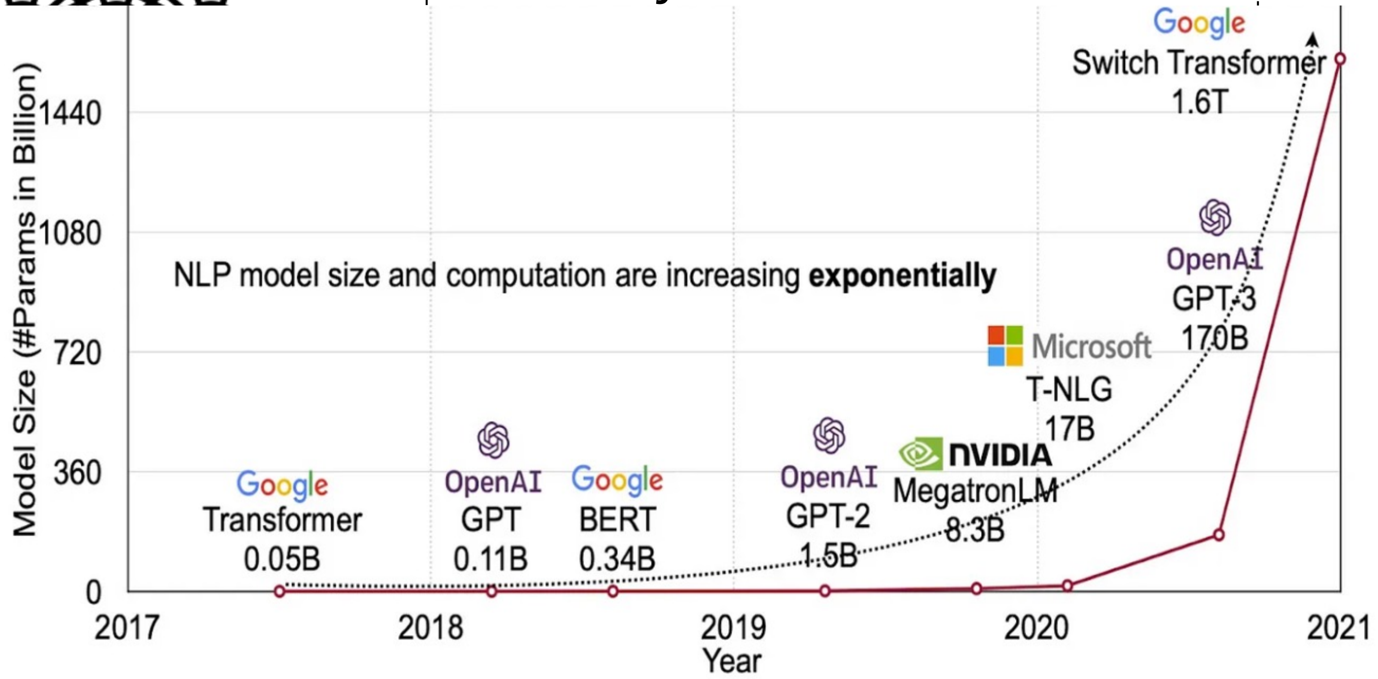
- Language models:

I am in Washington D.C. to attend a



P(workshop) = 0.1
P(conference) = 0.05
P(cherry) = 0.00001

- Large:
  - Large amount of **data**: even trillions of tokens.
  - **Size** of the network is large.



Credit: link

Training stages

Stage 1: **Pretraining** using next-word prediction

Pretraining is not enough for instruction following

Prompt	*Explain the moon landing to a 6 year old in a few sentences.*
Completion	GPT-3

```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

Pretrained models are not good at instruction following and understanding user intent.

Ouyang et. al. 2022

Training stages

Stage 2: **Fine-tuning** on a small dataset

Training stages

Stage 2: **Fine-tuning** on a small dataset

- Supervised fine-tuning (SFT): demonstration data

  Problem: (1) demonstration data is **expensive** to collect; (2) language generation is open-ended; (3) [we will see that] SFT is another form of pretraining, not necessarily accounting for the planning aspect; (4) …

Training stages

Stage 2: **Fine-tuning** on a small dataset

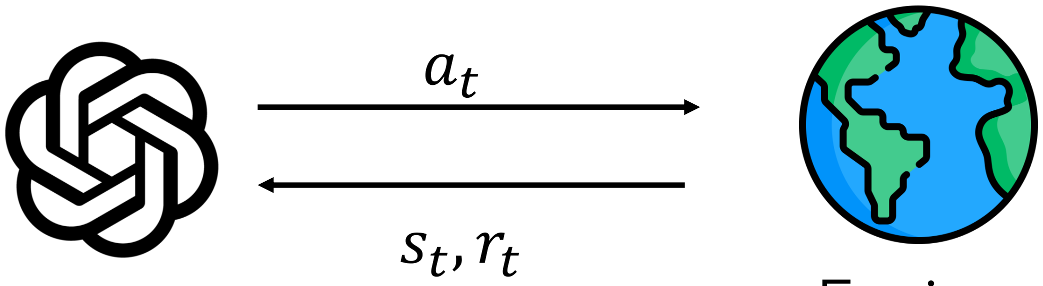
- Supervised fine-tuning (SFT): demonstration data
- RLHF (often including SFT): ranking/preference data

Notation

- Vocabulary set V, (max) length of a generated response T
- Given a prompt $x = (x_1, \dots, x_m)$, the LM generates a next-token: $x_{m+1} \sim \pi_\theta(\cdot\,|x)$ where $x_{m+1} \in V$.
- A response is denoted as $y = (x_{m+1}, \dots, x_{m+T})$ where $x_{m+t} \sim \pi_\theta(\cdot\,|x, x_1, \dots, x_{m+t-1})$.

MDP for Language Generation

- Initial state: $s_0 = x$ is the prompt
- At time $t \leq T$,
  - a_t is a token sampled from $\pi_\theta(\cdot\, | s_t)$
  - The next state $s_{t+1} = (s_t, a_t)$



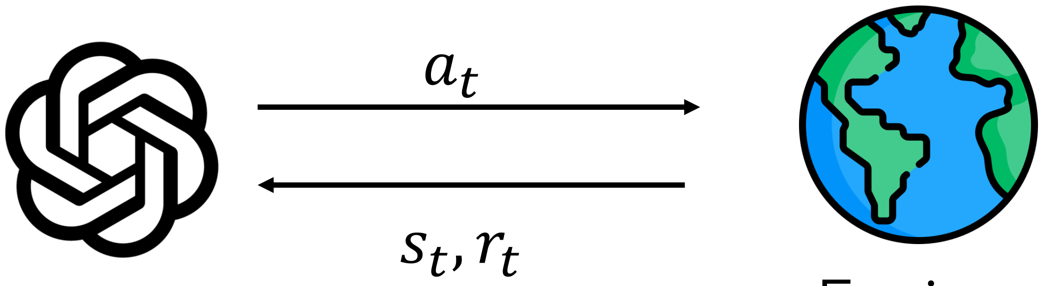
Policy
$\pi_\theta(a_t | s_t)$

Environment
Transition: specified by LM
Reward: unknown

MDP for Language Generation

- Initial state: $s_0 = x$ is the prompt
- At time $t \leq T$,
  - a_t is a token sampled from $\pi_\theta(\cdot\,|s_t)$
  - The next state $s_{t+1} = (s_t, a_t)$

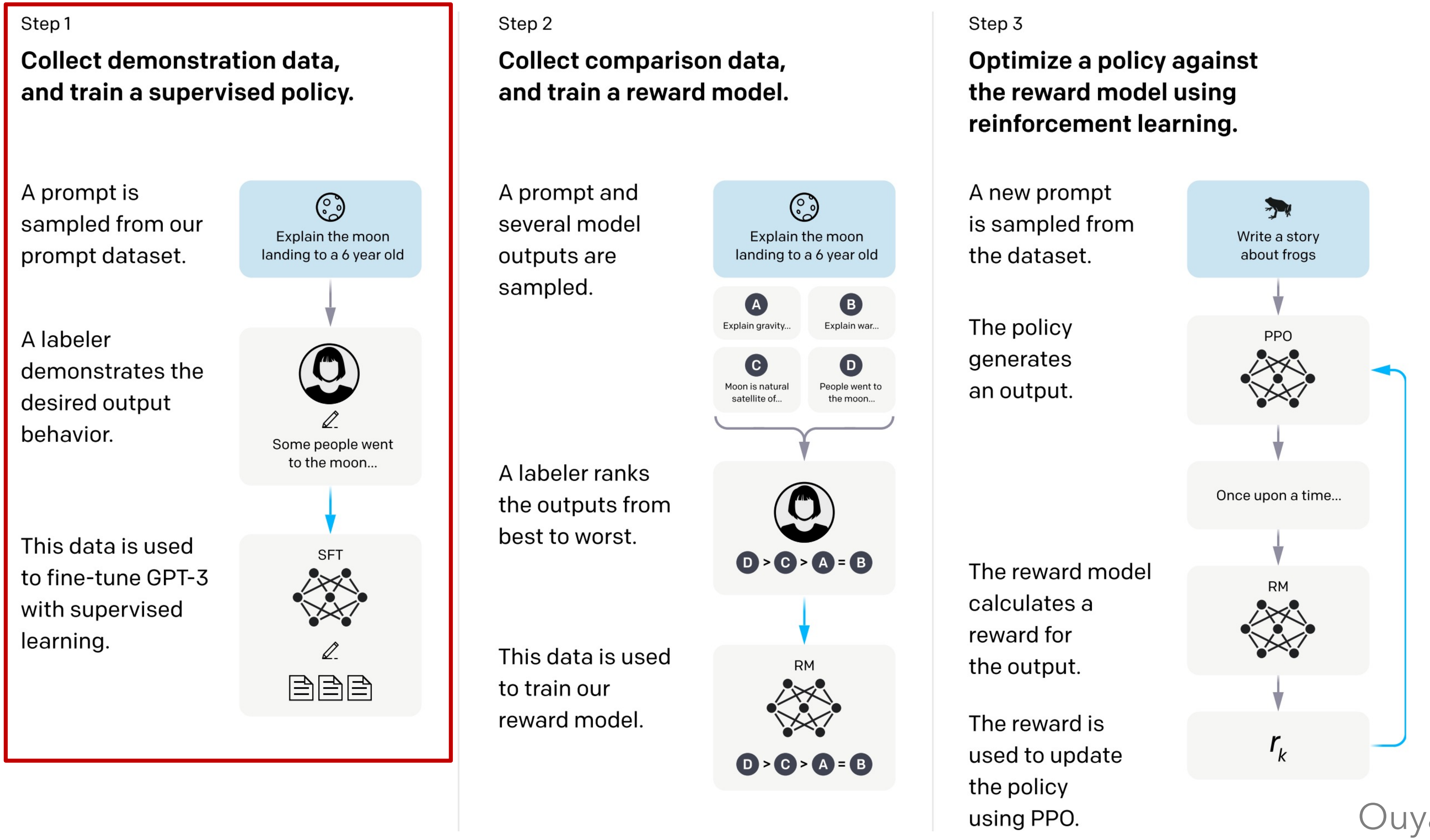
Contextual bandit formulation:
Given context x,
pick an arm $y \in [V]^T$,
receives a reward $r(x, y)$



Policy
$\pi_\theta(a_t|s_t)$

Environment
Transition: specified by LM
Reward: unknown

Reinforcement Learning From Human Feedback



Ouyang et al. 2022

Step 1: SFT

Given expert demonstrations $\{(x_i, y_i)\}_{i=1}^n$, minimize the per-token loss

$$\max_\theta \sum_{i=1}^n \sum_{t=1}^T \log \pi_\theta\left(y_{i,t} \middle| x_i, y_{i,<t}\right) .$$

Cross-entropy loss treating the next-token as the label.

- still a per-token loss

Step 1: SFT

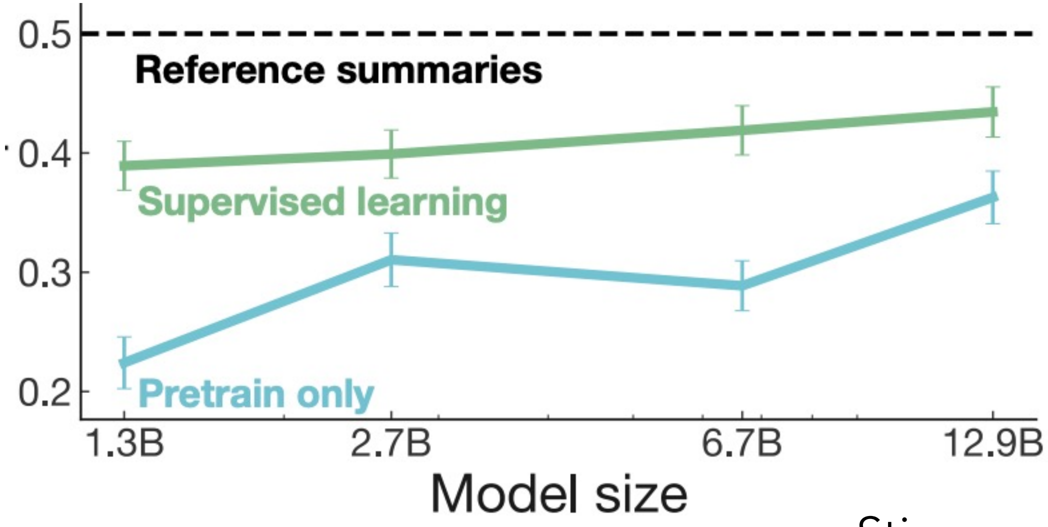
Given expert demonstrations $\{(x_i, y_i)\}_{i=1}^{n}$, minimize the per-token loss

$$\max_{\theta} \sum_{i=1}^{n} \sum_{t=1}^{T} \log \pi_\theta\left(y_{i,t} \middle| x_i, y_{i,<t}\right) .$$

Cross-entropy loss treating the next-token as the label.

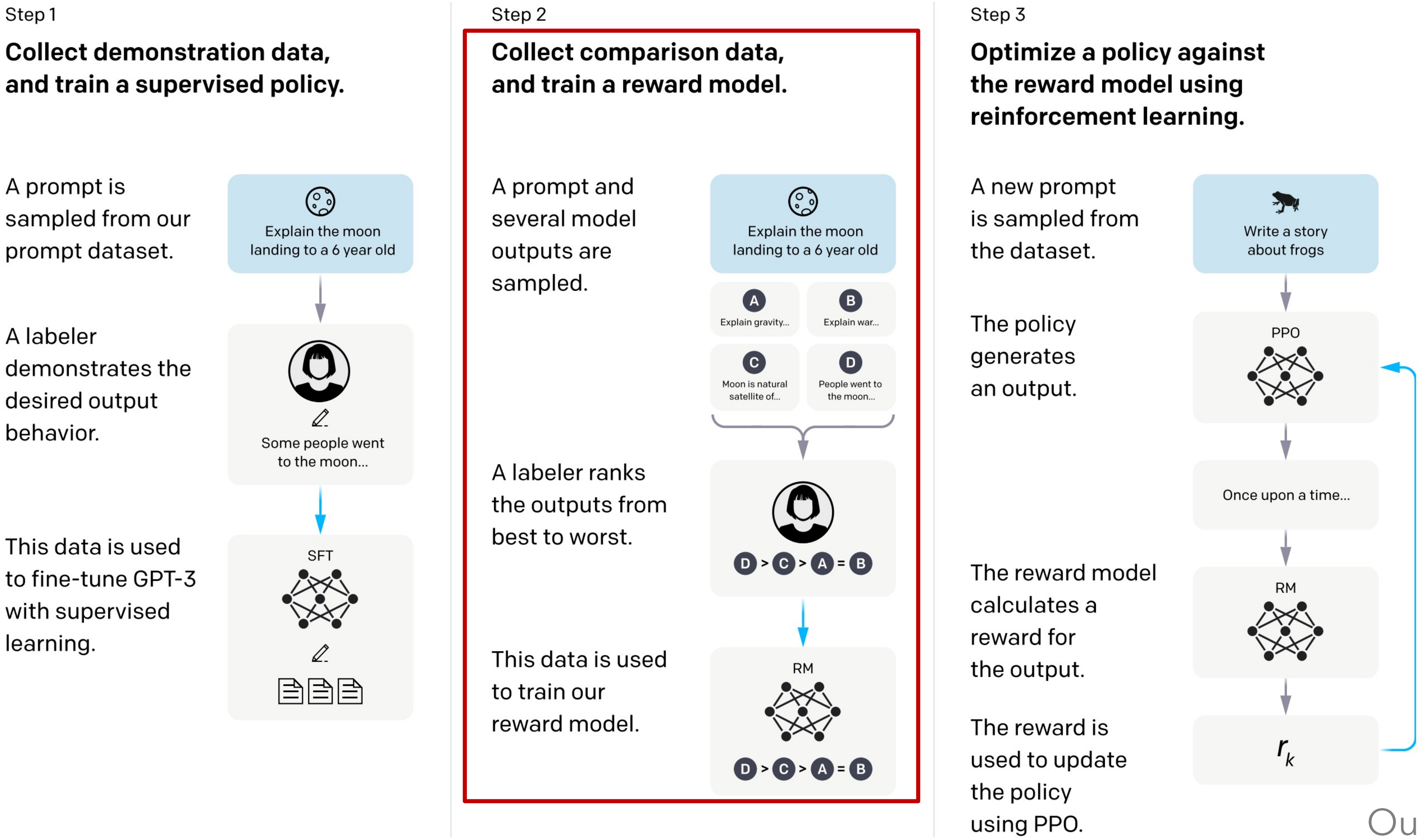
- still a per-token loss

Fraction of the time humans prefer models' summaries over the human-generated ones.



Stiennon et. al. 2020.

Reinforcement Learning From Human Feedback



Ouyang et al. 2022

Step 2: Reward Modelling

Reward Learner: r takes in (x, y) and outputs a constant.

Step 2: Reward Modelling

Reward Learner:

1. Given human feedback data $D = \{\underbrace{x_i}_{\text{Prompt}}, \overbrace{y_{i,1}}^{\text{Preferred}}, \underbrace{y_{i,2}}_{\text{Less Preferred}}\}$

2. Make the key assumption: for all (x, y_1, y_2),

$$\mathbb{P}(y_1 \succ y_2 | x) = \sigma(r(x, y_1) - r(x, y_2))$$

Bradley & Terry 1952

3. Learn the reward function through empirical risk minimization:

$$r^\star_{\text{vanilla}} \in \operatorname{argmin}_r \sum_{i=1}^{n} -\log \sigma(r(x_i, y_{i,1}) - r(x_i, y_{i,2}))$$

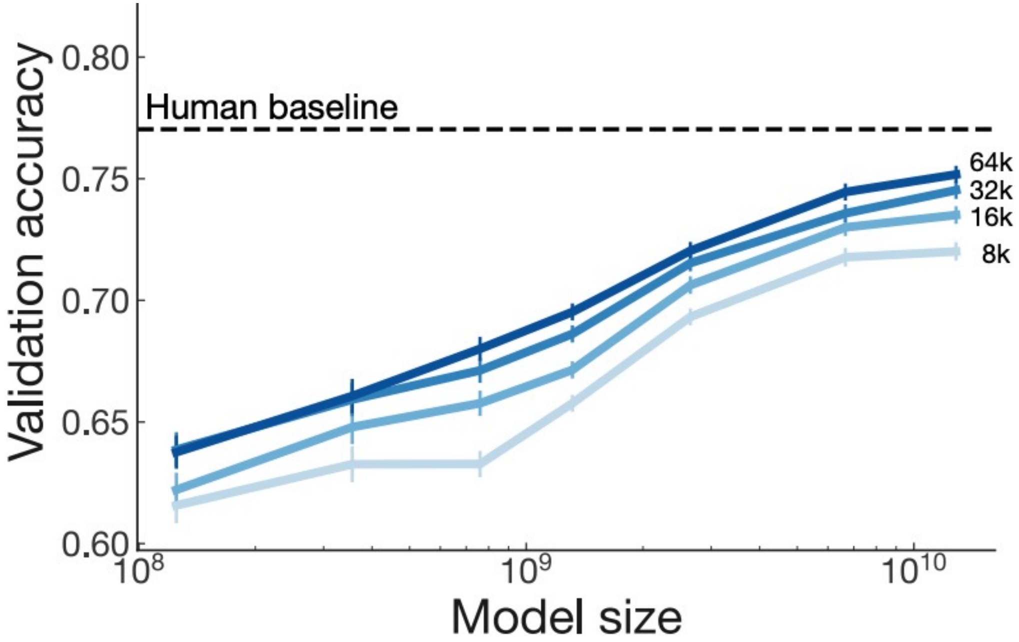
Maximum Likelihood Estimator

Step 2: Reward Modelling

Reward Learner:

Learn the reward function through empirical risk minimization:

$$r^{\star}_{\text{vanilla}} \in \text{argmin}_r \sum_{i=1}^{n} -\log \sigma(r(x_i, y_{i,1}) - r(x_i, y_{i,2}))$$



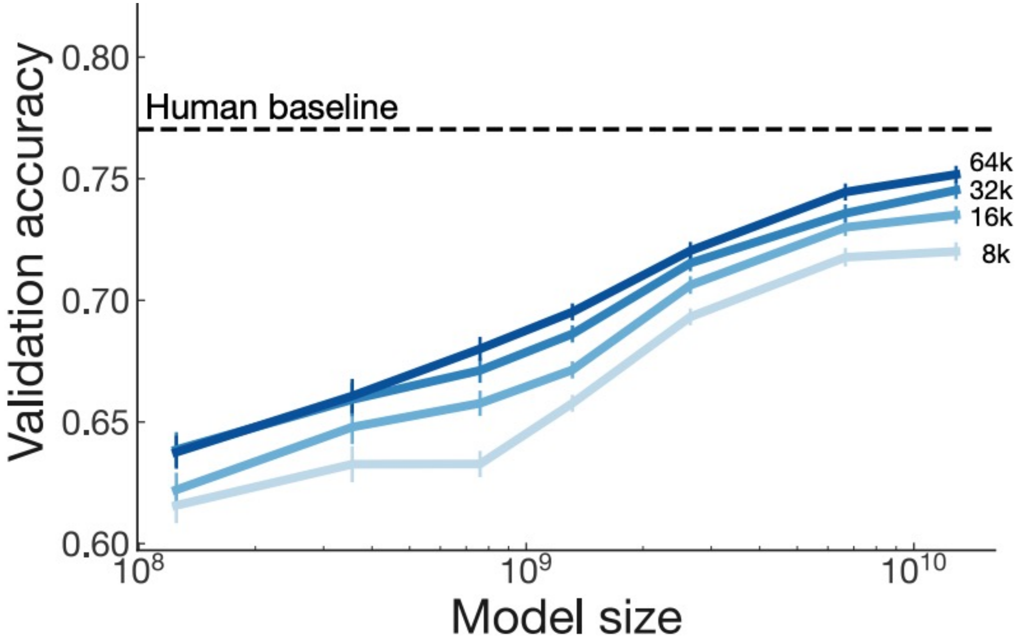
Stiennon et. al. 2020.
TLDR dataset

Step 2: Reward Modelling

Reward Learner:

Learn the reward function through empirical risk minimization:

$$r^\star_{\text{vanilla}} \in \text{argmin}_r \sum_{i=1}^{n} -\log \sigma\big(r(x_i, y_{i,1}) - r(x_i, y_{i,2})\big)$$



What's missing?

Stiennon et. al. 2020.
TLDR dataset

What is **missing**?

Reward Learner:

1. Given human feedback data $D = \{\underbrace{x_i}_{\text{Prompt}}, \overbrace{y_{i,1}}^{\text{Preferred}}, \underbrace{y_{i,2}}_{\text{Less Preferred}}, \overbrace{\boldsymbol{u_i}}^{\text{User/Annotator Identifier}}\}$

2. Make the key assumption: for all $(x, y_1, y_2, \boldsymbol{u})$,

$$\underbrace{\mathbb{P}(y_1 \succ y_2 | x, u) = \mathbb{P}(y_1 \succ y_2 | x)}_{\textbf{A1: Preference Uniformity}} = \underbrace{\sigma(r(x, y_1) - r(x, y_2))}_{\text{A2: Bradley-Terry}}$$

3. Learn the reward function through empirical risk minimization:

$$r^\star_{\text{vanilla}} \in \text{argmin}_r \sum_{i=1}^{n} -\log \sigma(r(x_i, y_{i,1}) - r(x_i, y_{i,2}))$$

What’s the problem with assuming preference uniformity?

1. Human preferences are naturally **diverse** and **subjective**. There is no “objectively correct” preference.

What's the problem with assuming preference uniformity?

1. Human preferences are naturally **diverse** and **subjective**. There is no "objectively correct" preference.
2. Deterministic LM under $r^{\star}_{\text{vanilla}}$ is equivalent to **majority voting**.

Human feedback on x_i = "I like"

User	dog	cat
1	Preferred	Less Preferred
2	Preferred	Less Preferred
3	Less Preferred	Preferred

Under $r^{\star}_{\text{vanilla}}$, for all three user,

$$\mathbb{P}(\text{dog} \succ \text{cat} \mid \text{I like}) = \frac{2}{3}.$$

Generated text for all users:

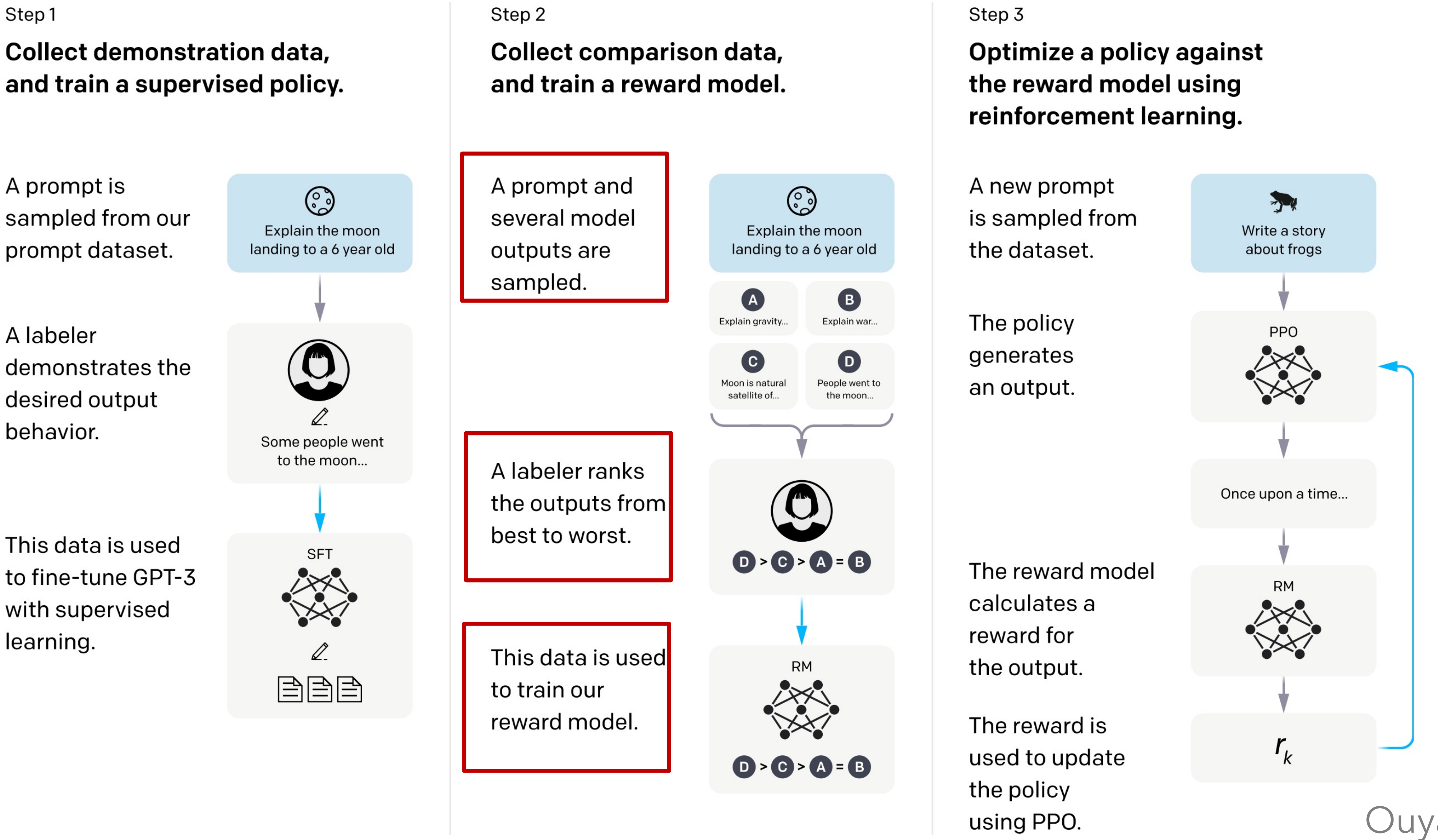
"I like dog"

What's the problem with assuming preference uniformity?

Human preferences are naturally **diverse** and **subjective**. There is no "objectively correct" preference.

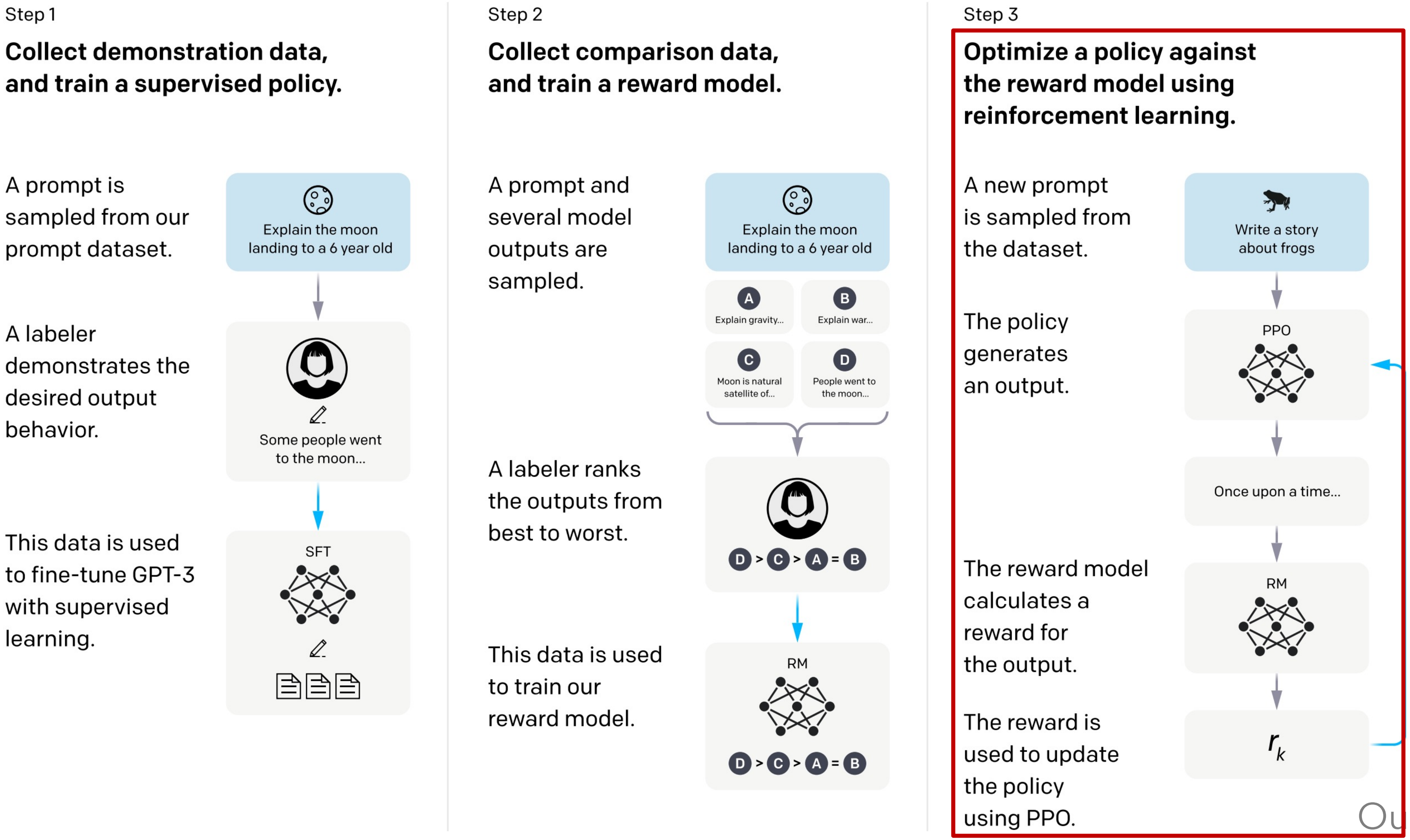
Check out our new paper on
Personalized Language Modeling from Personalized Human Feedback
arXiv: 2402.05133

Reinforcement Learning From Human Feedback



Ouyang et al. 2022

Reinforcement Learning From Human Feedback



Ouyang et al. 2022

Step 3: Policy optimization

Learning Objective:

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

Why?

- Maximizing the reward
- Be close to the SFT policy (e.g., reduce reward hacking)

Step 3: Policy optimization

Learning Objective:

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

Why?

- Maximizing the reward
- Be close to the SFT policy (e.g., reduce reward hacking)

Caution: $KL(\pi_\theta, \pi_{\text{ref}}) = \mathbb{E}_{y \sim \boldsymbol{\pi_\theta}(y|x)}\left[\log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right]$; forcing $\pi_\theta(y|x)$ to be 0 if $\pi_{\text{ref}}(y|x)$=0.

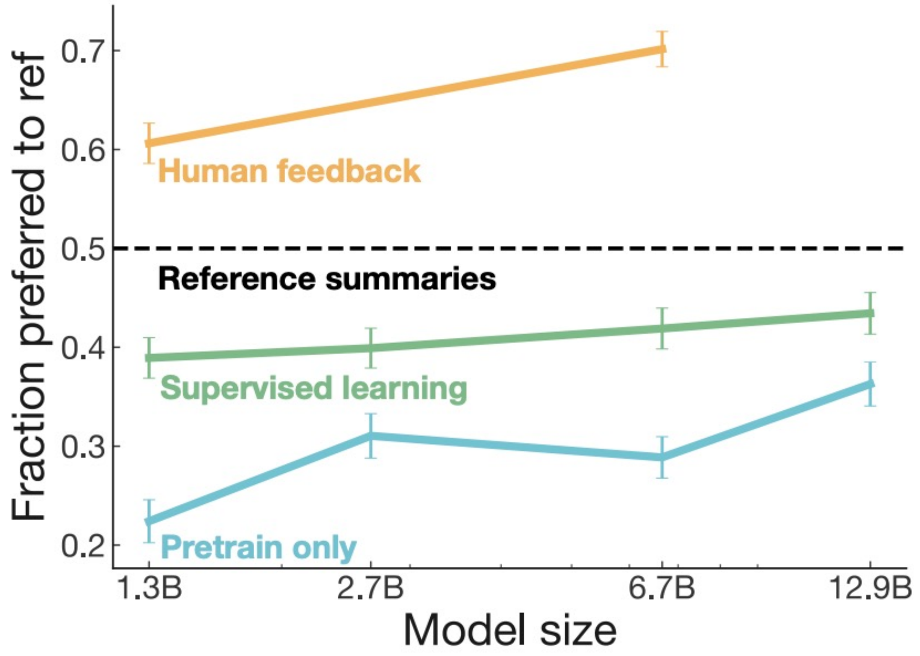
Step 3: Policy optimization

Learning Objective:

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

How?

- Best-of-N: surprisingly strong
- Proximal Policy Optimization:
  - [arXiv: 1707.06347]
  - [arXiv:1506.02438]



In general, hard to optimize!

DPO: Direct Preference Optimization

PPO is hard to train… Can we work with a purely supervised loss?

Learning Objective:

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

Turns out: there is a **closed** form of the optimal policy!

$$\pi(y|x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y|x) \exp(\frac{1}{\beta} r(x, y))$$

Rafailov et. al. 2023

DPO: Direct Preference Optimization

PPO is hard to train… Can we work with a purely supervised loss?

Learning Objective:

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

Turns out: there is a **closed** form of the optimal policy!

$$\pi(y|x) = \frac{1}{Z(x)}\pi_{\text{ref}}(y|x)\exp(\frac{1}{\beta}r(x,y))$$

Re-arrange terms and plug-in to the reward model loss:

$$\min_r - \mathbb{E}_{(x,y_{i,1},y_{i,2})\sim D}[\log\sigma(\beta\frac{\pi(y_{i,1}|x_i)}{\pi_{\text{ref}}(y_{i,1}|x_i)} - \beta\frac{\pi(y_{i,2}|x_i)}{\pi_{\text{ref}}(y_{i,2}|x_i)})]$$

Rafailov et. al. 2023

DPO: Direct Preference Optimization

PPO is hard to train… C
loss?

Learning Objective:

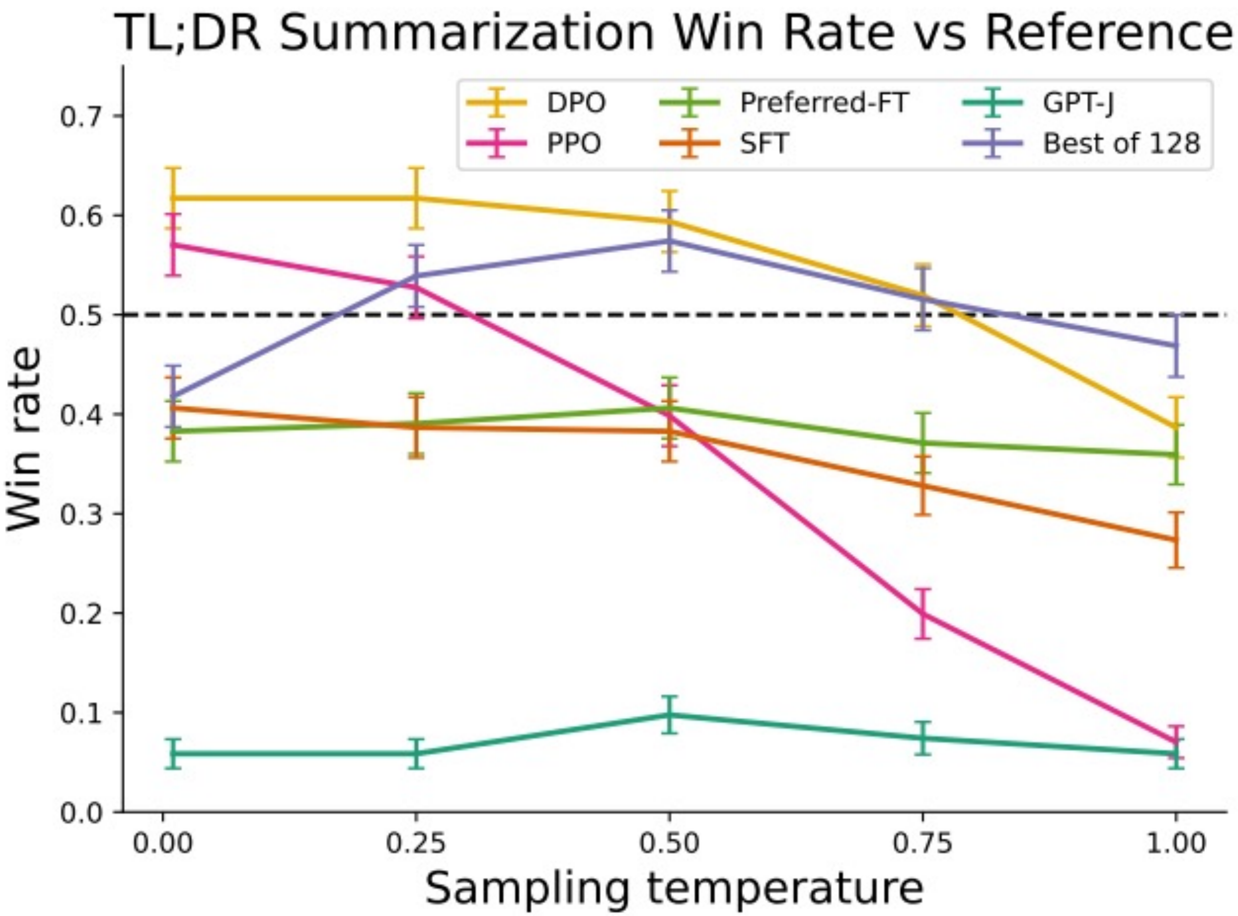
$$\max_{\theta} \mathbb{E}_{x \sim D, y}$$

Turns out: there is a **closed**

$$\pi(y|x) = \frac{1}{Z(x}$$

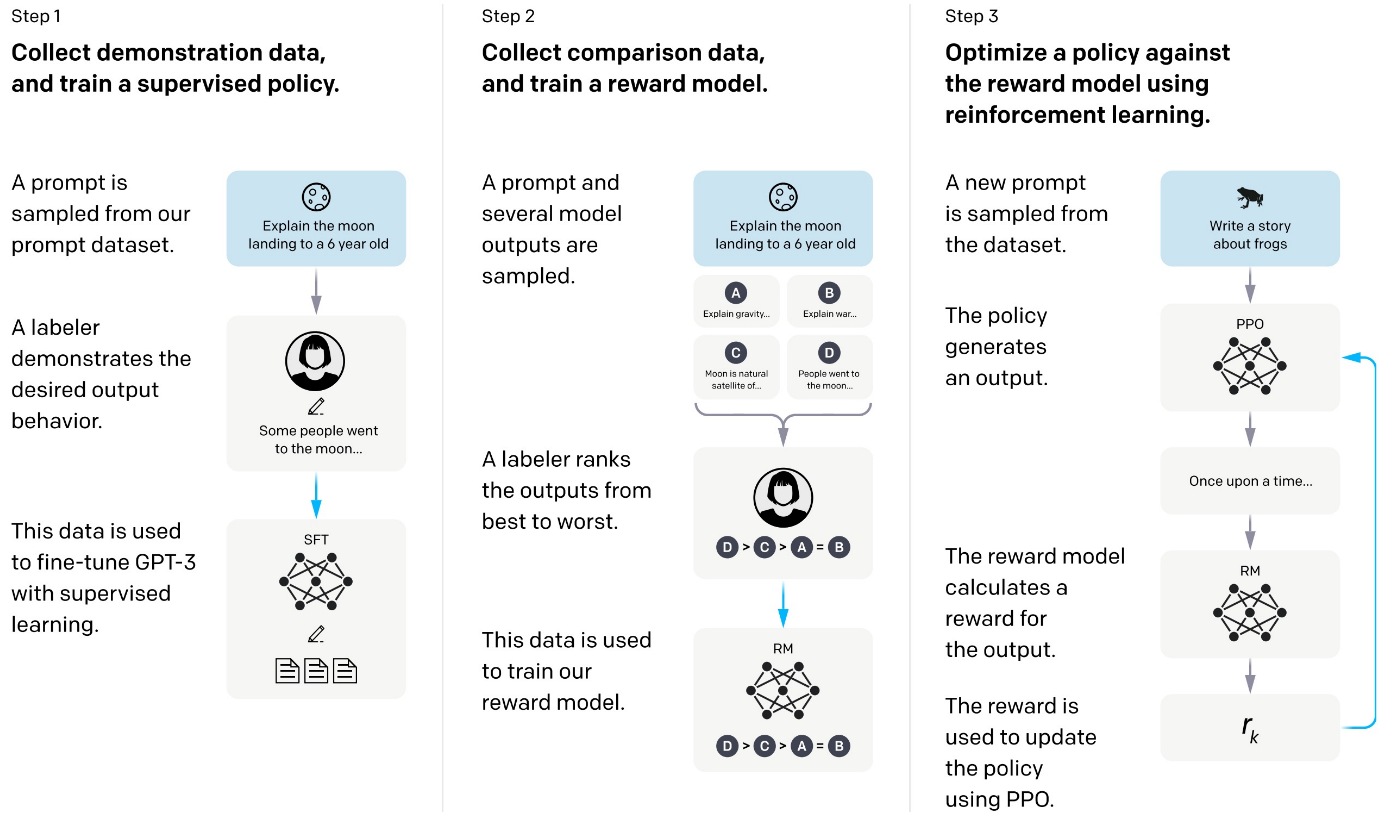
Re-arrange terms and plug

$$\min_r - \mathbb{E}_{(x, y_{i,1}, y_{i,2}}$$



Rafailov et. al. 2023

A common paradigm in RL



Model-based approach

Some characteristics:

- Large action space
- Information in the form of comparison
- "sparse" reward
- …

Research landscape of RLHF

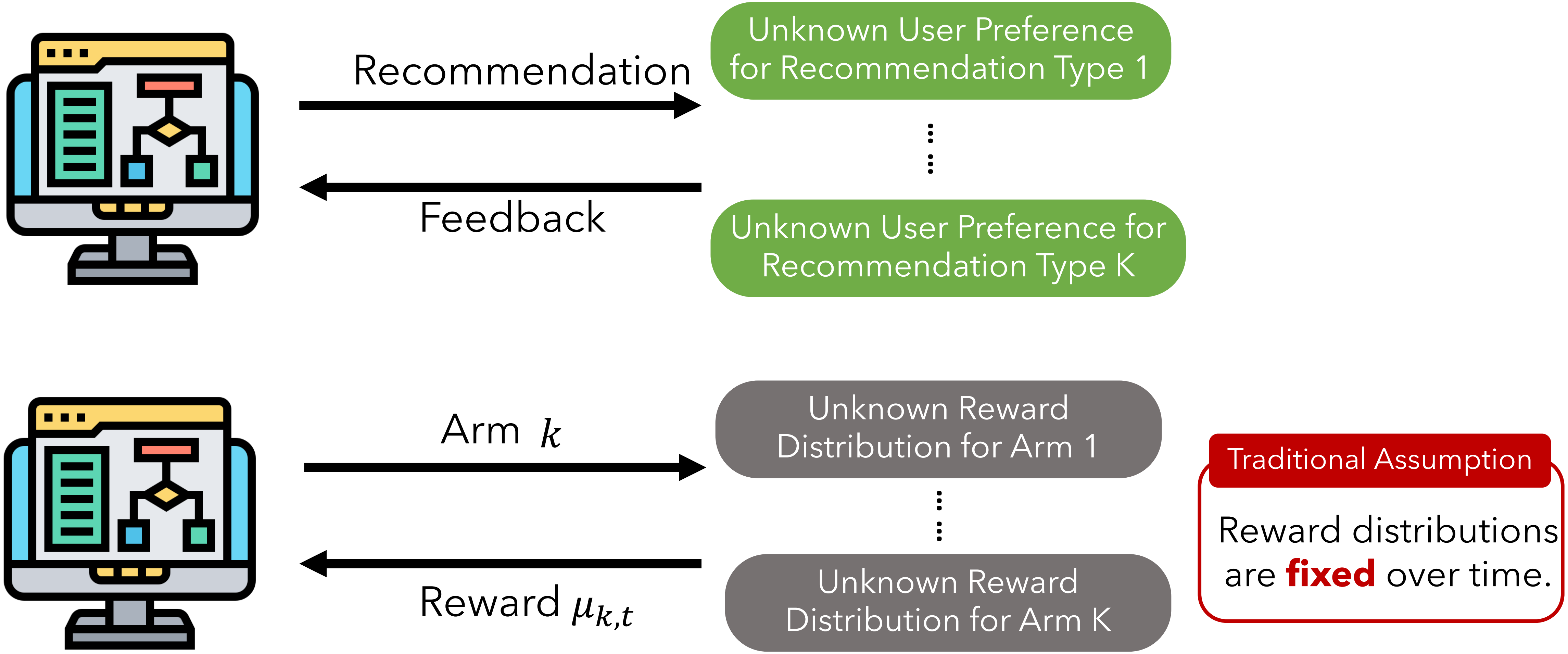
LM alignment and RLHF is an active developing area:

- Effectiveness:
  - New learning objective?
  - Multi-turn instead of single-turn interaction?
  - Personalization?
  - …
- Efficiency:
  - Sample: Self-play? AI instead of human feedback? Actively query human?
  - Compute: Can we perform RLHF without optimizing a language model (at a large scale)?

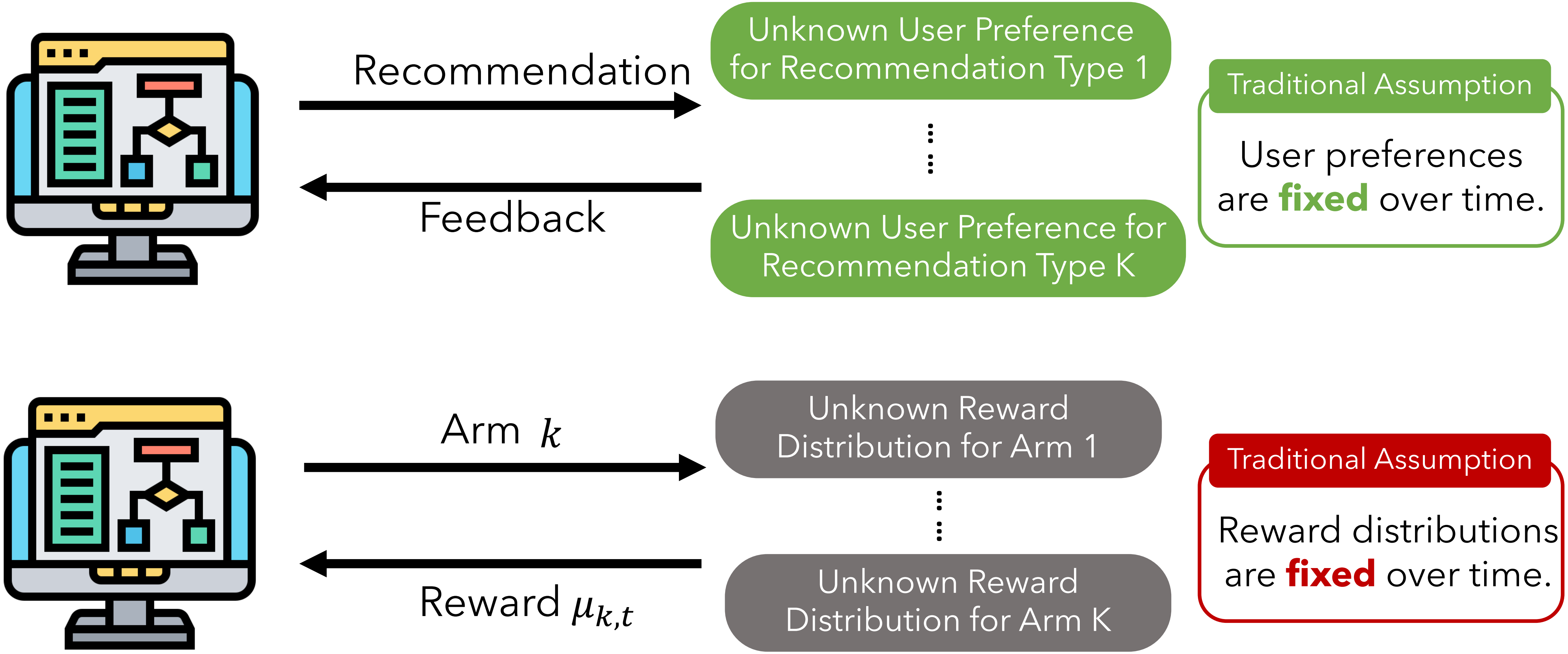
Outline

- Broader landscape of Learning from Human Feedback
- Two case studies:
  - Language Modeling: RLHF for aligning model with user intent
  - Recommender systems: **Multi-armed bandits** accounting for **evolving** human preferences
    - Multi-armed bandits for modeling recommender system
    - A specific variant that accounts for user preference dynamics
- What's next?

Recommender systems as multi-armed bandits



Recommender systems as multi-armed bandits



Evolving preferences

Existing psychology and marketing research suggests that people have evolving preferences. [Tucker, 1964; McConnell, 1968; …]


No publicly available experimental framework for testing this hypothesis in bandit settings.

Evolving preferences

Existing psychology and marketing research suggests that people have evolving preferences. [Tucker, 1964; McConnell, 1968; …]

No publicly available experimental framework for testing this hypothesis in bandit settings.

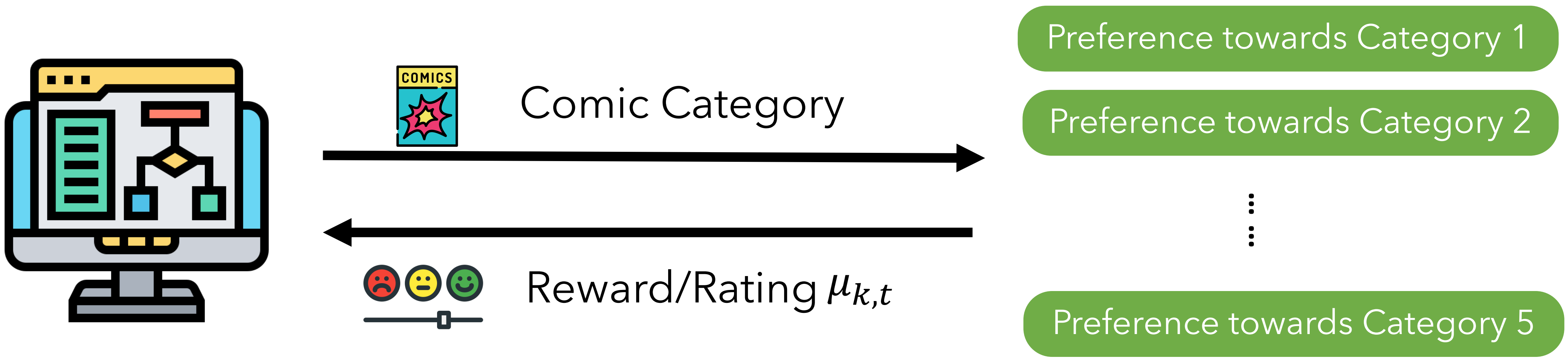
We developed an open-source library:

A toolkit for conducting human subject studies in multi-armed bandits

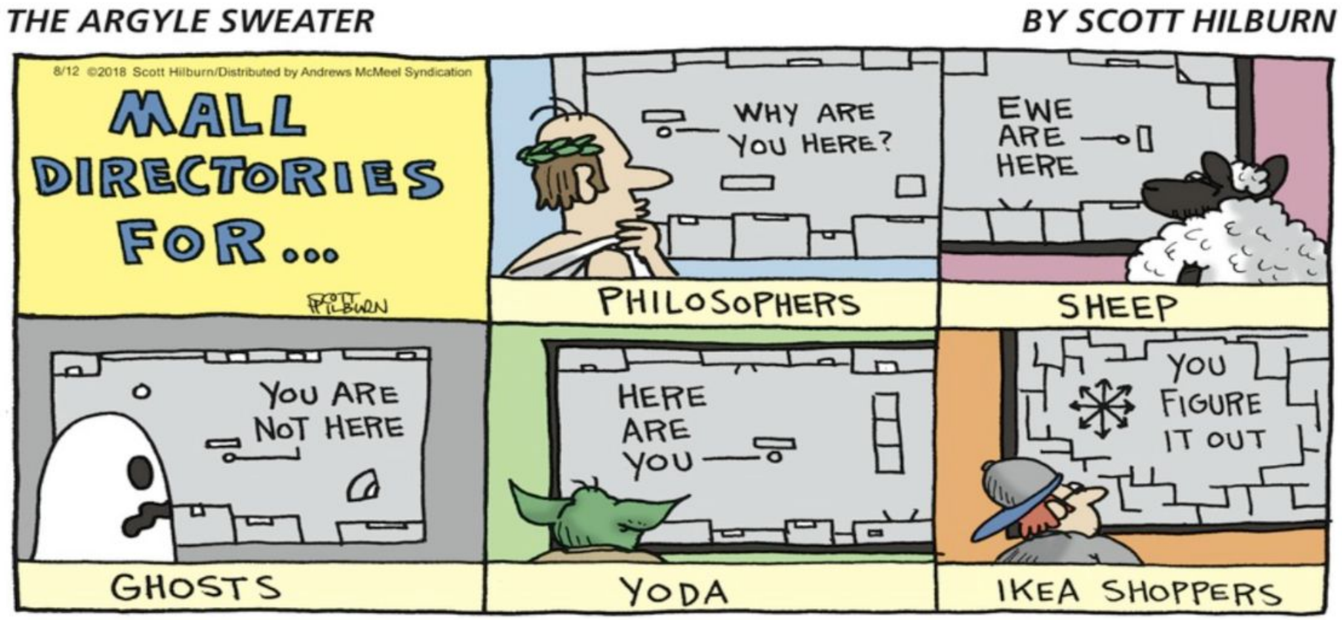
Usage

Identify reward assumptions that better capture user preferences.

Evolving preferences



Evolving preferences



Rate your enjoyment of the comic between 1 and 9.

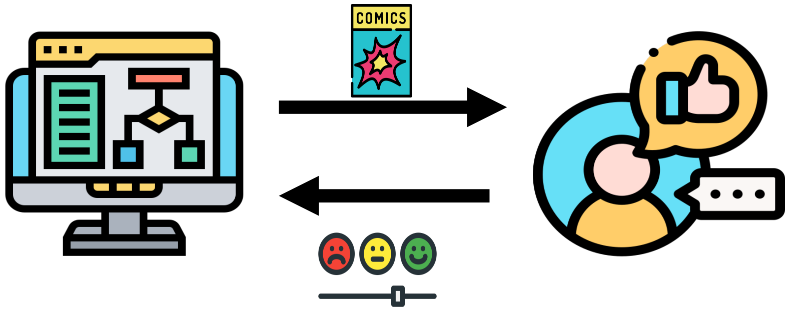
① *Likert scale slider for rating.*

Score: 6

1	3	5	7	9
Disliked a lot	*Somewhat disliked*	*Neutral*	*Somewhat enjoyed*	*Enjoyed a lot*

How many unique characters (with a face and/or body) are in this comic? (Put 5 if there are more than 5). 5

② ***Attention check question(s).***



Evolving preferences

REPEAT

$(\underbrace{1 \ldots 1}_{m \text{ times}}, \ldots, K \ldots K)$

CYCLE

$(\underbrace{1 \ldots K, \ldots, 1 \ldots K}_{m \text{ cycles}})$

Comic Category k

Reward/Rating $\mu_{k,t}$

Preference towards Category 1

Preference towards Category 2

⋮

Preference towards Category 5

Understanding the validity of assumptions on human preferences in MABs. **Leqi***, Zhou*, Lipton, Kılınç-Karzan, Montgomery. CHI ’23

Evolving preferences

REPEAT

$(\underbrace{1 \dots 1}_{m \text{ times}}, \dots, K \dots K)$

CYCLE

$(\underbrace{1 \dots K, \dots, 1 \dots K}_{m \text{ cycles}})$

Comic Category k →

Preference towards Category 1

Preference towards Category 2

⋮

Preference towards Category 5

← Reward/Rating $\mu_{k,t}$

Key Characteristics

Each arm is pulled the **same** number of times.

Understanding the validity of assumptions on human preferences in MABs. **Leqi***, Zhou*, Lipton, Kılınç-Karzan, Montgomery. CHI ’23

Evolving preferences

REPEAT

$(\underbrace{1 \ldots 1}_{m \text{ times}}, \ldots, K \ldots K)$

CYCLE

$(\underbrace{1 \ldots K, \ldots, 1 \ldots K}_{m \text{ cycles}})$

Comic Category k →

Preference towards Category 1

Preference towards Category 2

⋮

Preference towards Category 5

← Reward/Rating $\mu_{k,t}$

Key Characteristics

Each arm is pulled the **same** number of times.

Test Statistic τ_k

Difference between mean rating for each comic category k under CYCLE and REPEAT.

Evolving preferences

REPEAT

$(\underbrace{1 \ldots 1}_{m \text{ times}}, \ldots, K \ldots K)$

CYCLE

$(\underbrace{1 \ldots K, \ldots, 1 \ldots K}_{m \text{ cycles}})$

Comic Category k

Reward/Rating $\mu_{k,t}$

Preference towards Category 1

Preference towards Category 2

⋮

Preference towards Category 5

Key Characteristics

Each arm is pulled the **same** number of times.

Test Statistic τ_k

Difference between mean rating for each comic category k under CYCLE and REPEAT.

User Preference for Category k

Evolving preferences

REPEAT

$(\underbrace{1 \ldots 1}_{m \text{ times}}, \ldots, K \ldots K)$

CYCLE

$(\underbrace{1 \ldots K, \ldots, 1 \ldots K}_{m \text{ cycles}})$

Comic Category k →

← Reward/Rating $\mu_{k,t}$

Preference towards Category 1

Preference towards Category 2

⋮

Preference towards Category 5

Key Characteristics

Each arm is pulled the **same** number of times.

Test Statistic τ_k

Difference between mean ~~rating~~ for each ~~comic category~~ k under CYCLE and REPEAT.

reward **arm**

Evolving preferences

REPEAT

$(\underbrace{1 \ldots 1}_{m \text{ times}}, \ldots, K \ldots K)$

CYCLE

$(\underbrace{1 \ldots K, \ldots, 1 \ldots K}_{m \text{ cycles}})$

Comic Category k →

← Reward/Rating $\mu_{k,t}$

Preference towards Category 1

Preference towards Category 2

⋮

Preference towards Category 5

Key Characteristics

Each arm is pulled the **same** number of times.

Test Statistic τ_k

Difference between mean reward for each arm k under CYCLE and REPEAT.

$\tau_k = 0$ → Preference remains the same under CYCLE and REPEAT.

$\tau_k \neq 0$ → Evolving preference exists.

Evolving preferences

REPEAT

$(\underbrace{1 \dots 1}_{m \text{ times}}, \dots, K \dots K)$

CYCLE

$(\underbrace{1 \dots K, \dots, 1 \dots K}_{m \text{ cycles}})$

Comic Category k →

← Reward/Rating $\mu_{k,t}$

Preference towards Category 1

Preference towards Category 2

⋮

Preference towards Category 5

Key Characteristics

Each arm is pulled the **same** number of times.

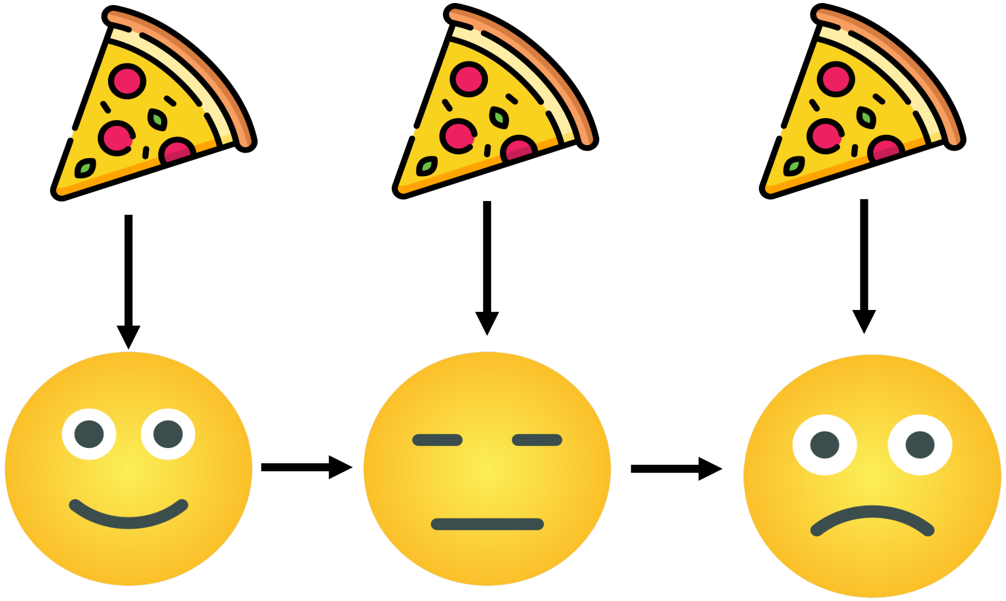
Test Statistic τ_k

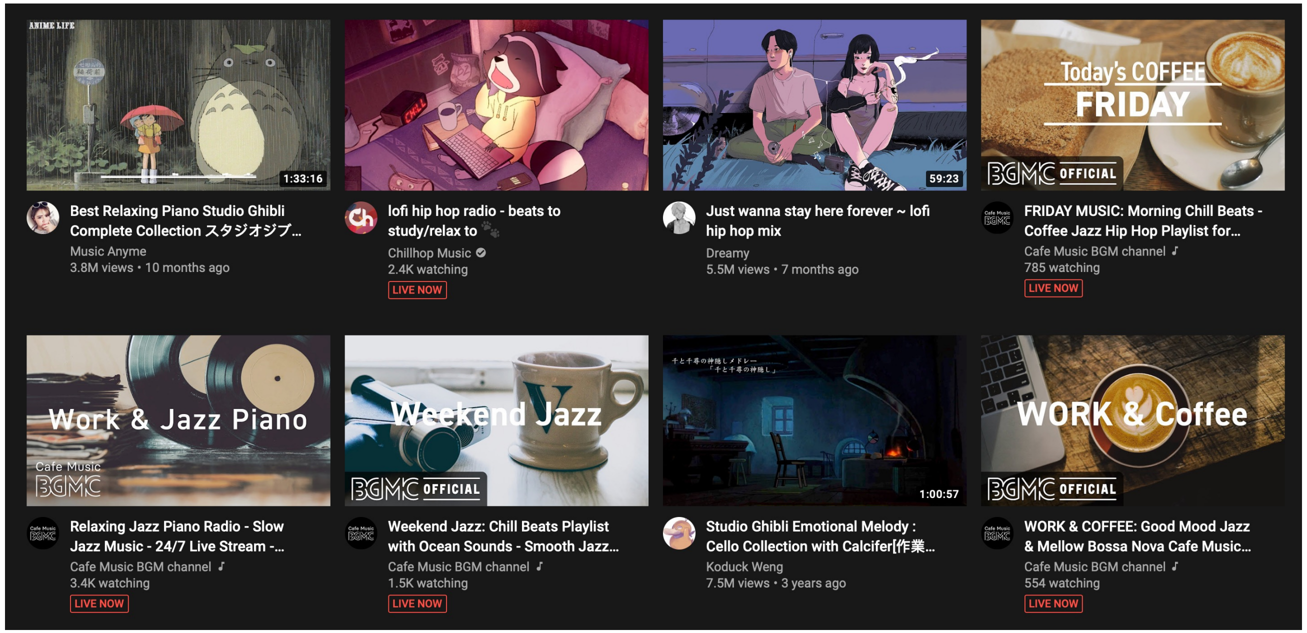
Difference between mean reward for each arm k under CYCLE and REPEAT.

	Family	Gag	Conservative	Office	Liberal
τ_k	0.784	0.635	1.274	0.552	1.475
p-value	<0.001*	<0.001*	<0.001*	<0.001*	<0.001*

Satiation

Short-term enjoyment declines due to repetitive exposure to the same item.

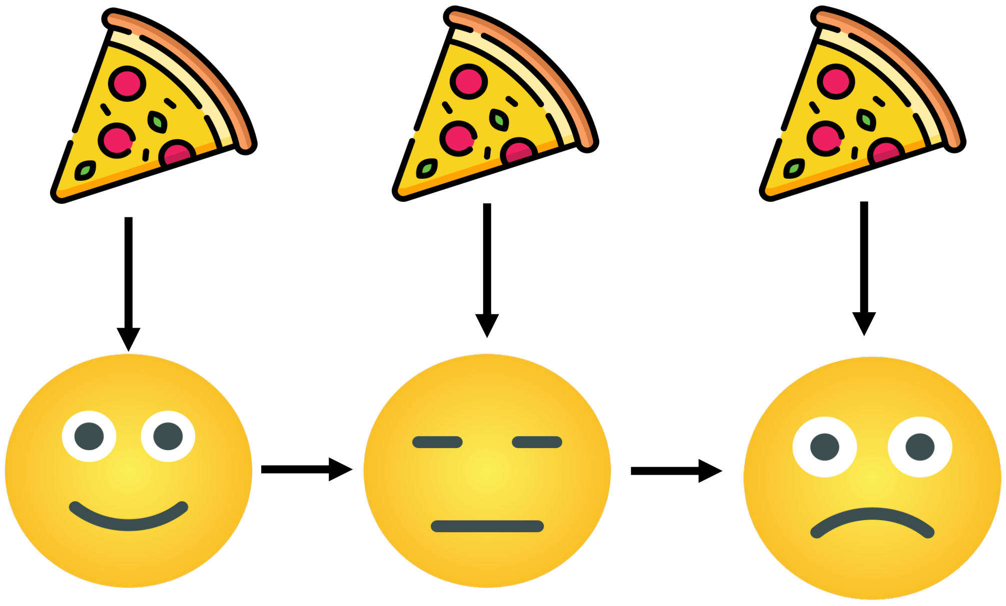


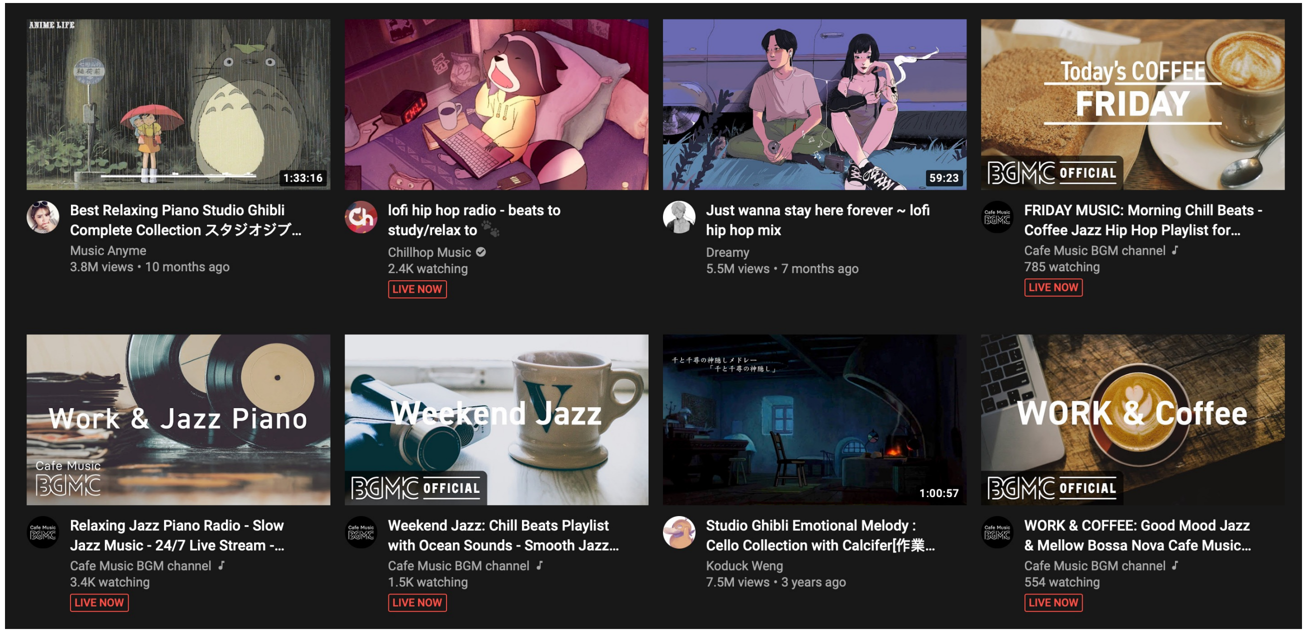


Rebounding bandits for modeling satiation effects. **Leqi L**, Kılınç-Karzan F, Lipton Z, Montgomery A. NeurIPS '21.

Satiation

Short-term enjoyment declines due to repetitive exposure to the same item.





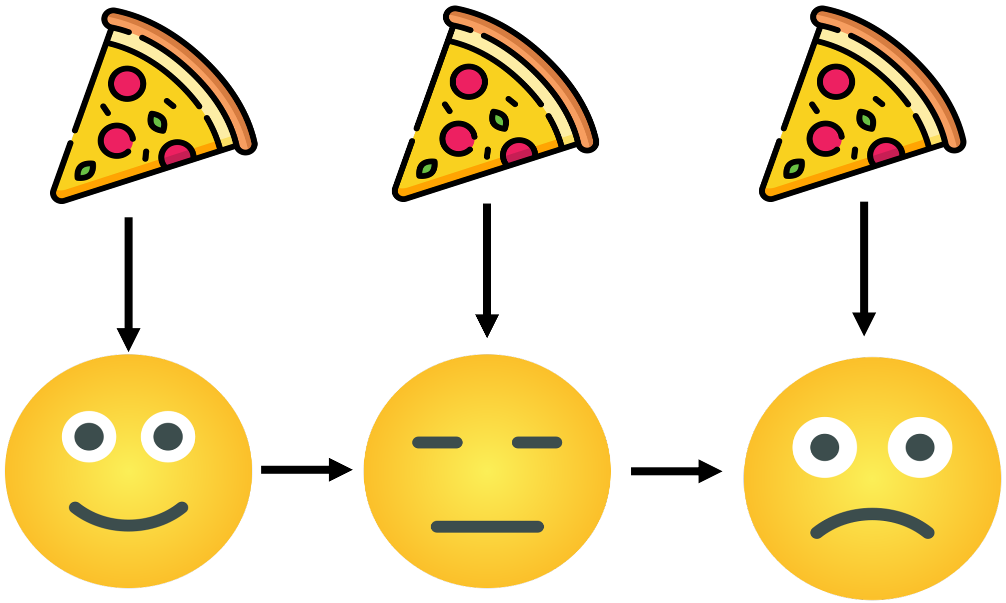
Observation

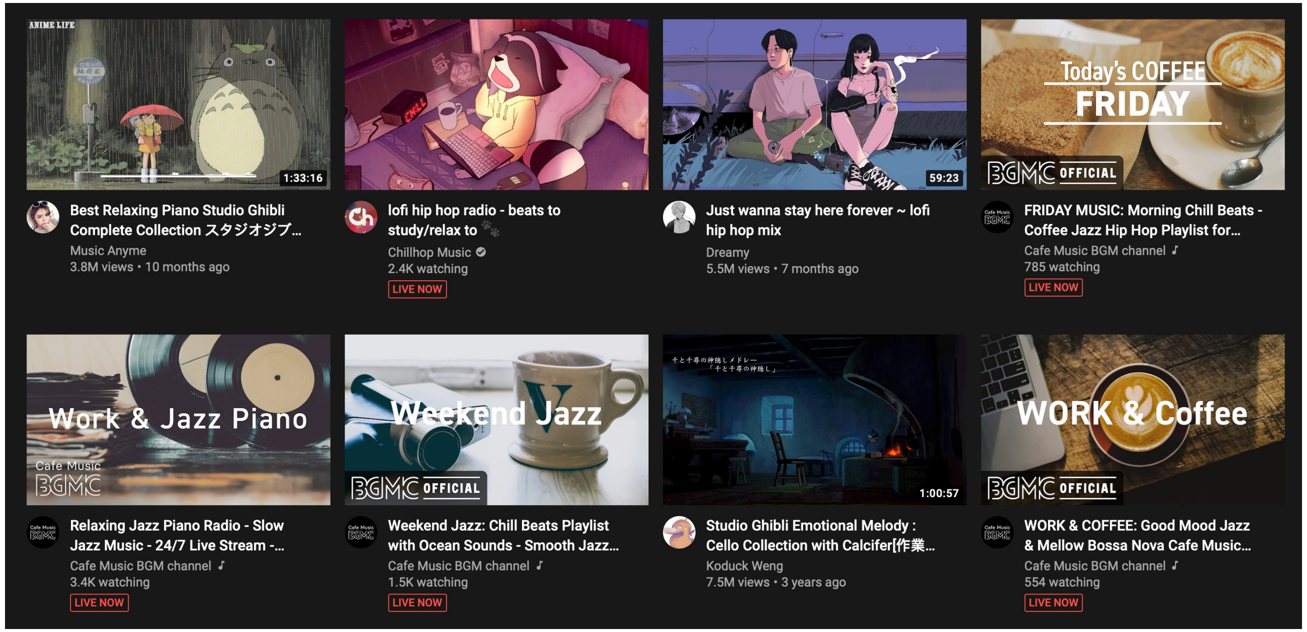
Under the traditional **fixed preference assumption** in MAB, this may happen.

Rebounding bandits for modeling satiation effects. **Leqi L**, Kılınç-Karzan F, Lipton Z, Montgomery A. NeurIPS '21.

Satiation

Short-term enjoyment declines due to repetitive exposure to the same item.





Key Characteristics

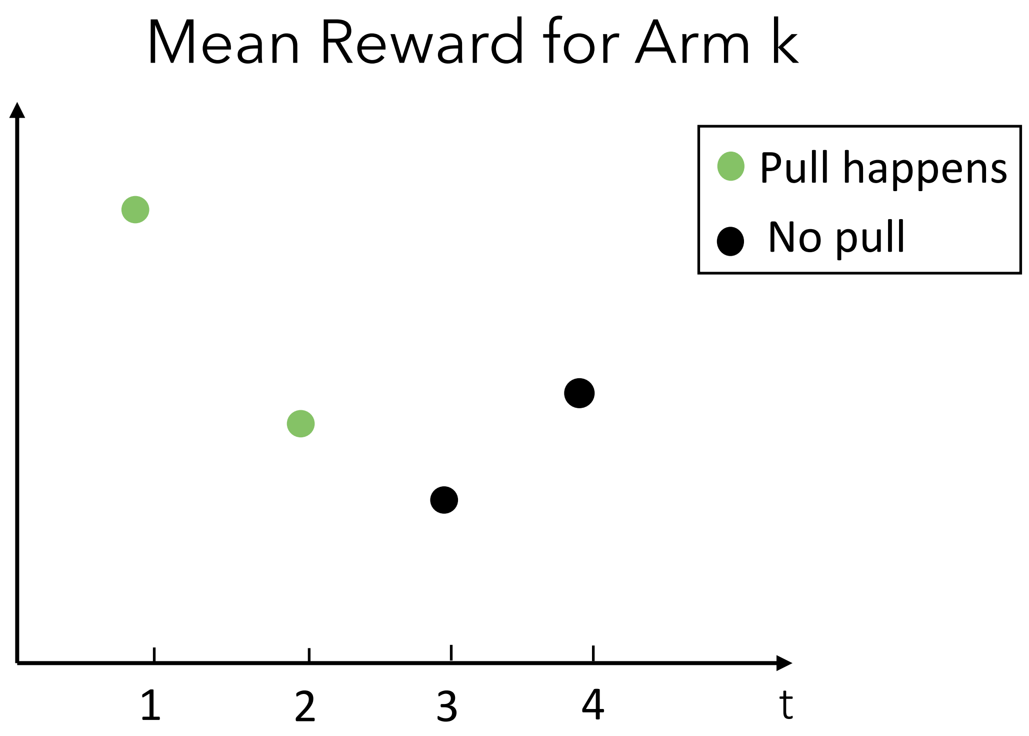
- ❑ Decline with repetitive exposure
- ❑ Rebound towards the baseline with no exposure

Rebounding bandits for modeling satiation effects. **Leqi L**, Kılınç-Karzan F, Lipton Z, Montgomery A. NeurIPS '21.

Rebounding bandits

Mean reward characteristics:

- Decline with consecutive pulls
- Rebound towards the baseline with disuse

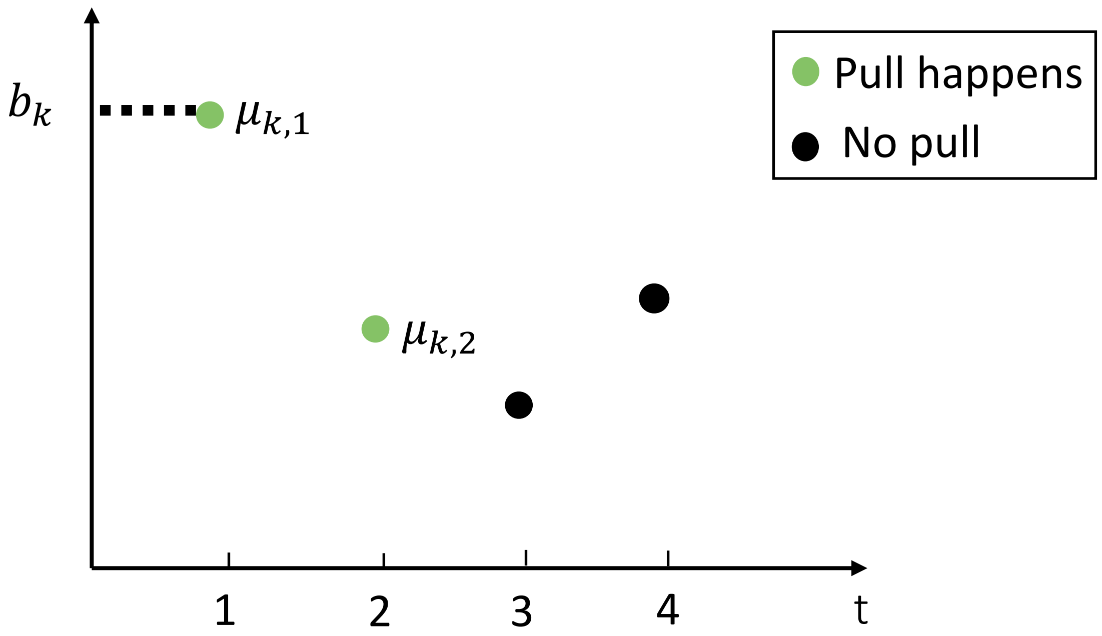


Rebounding bandits for modeling satiation effects. **Leqi L**, Kılınç-Karzan F, Lipton Z, Montgomery A. NeurIPS '21.

Rebounding bandits

Mean reward characteristics:
- Decline with consecutive pulls
- Rebound towards the baseline with disuse

Mean Reward for Arm k



Stochastic Linear Dynamical System

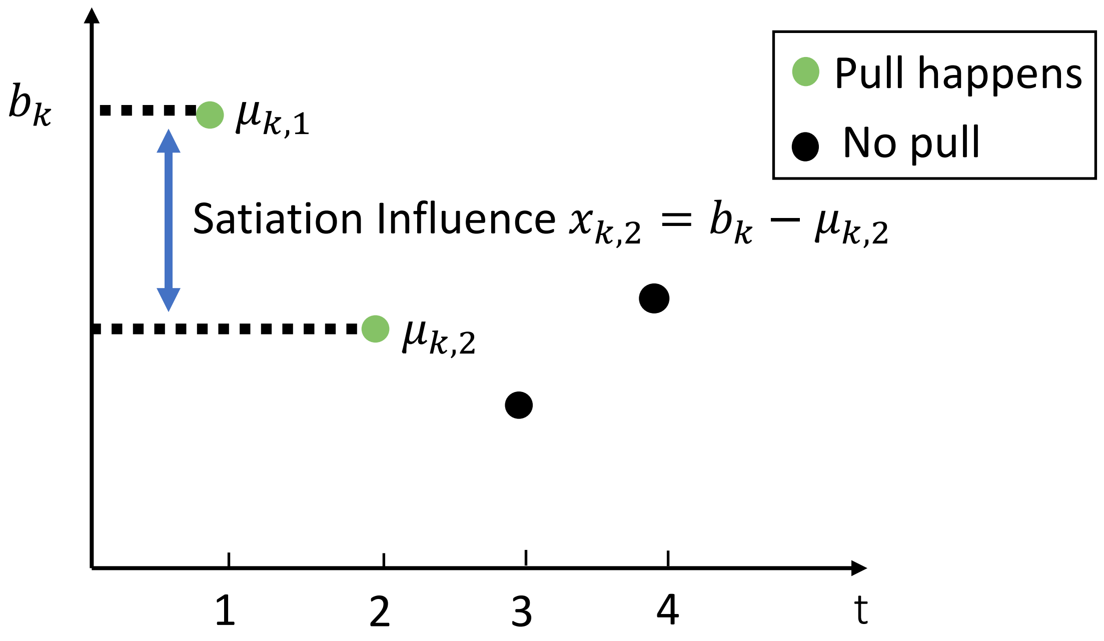
Rebounding bandits for modeling satiation effects. **Leqi L**, Kılınç-Karzan F, Lipton Z, Montgomery A. NeurIPS '21.

Rebounding bandits

Mean reward characteristics:

- Decline with consecutive pulls
- Rebound towards the baseline with disuse

Mean Reward for Arm k



Stochastic Linear Dynamical System

Reward model

$$\mu_{k,t} = b_k - x_{k,t}$$

Base Reward · Satiation Influence

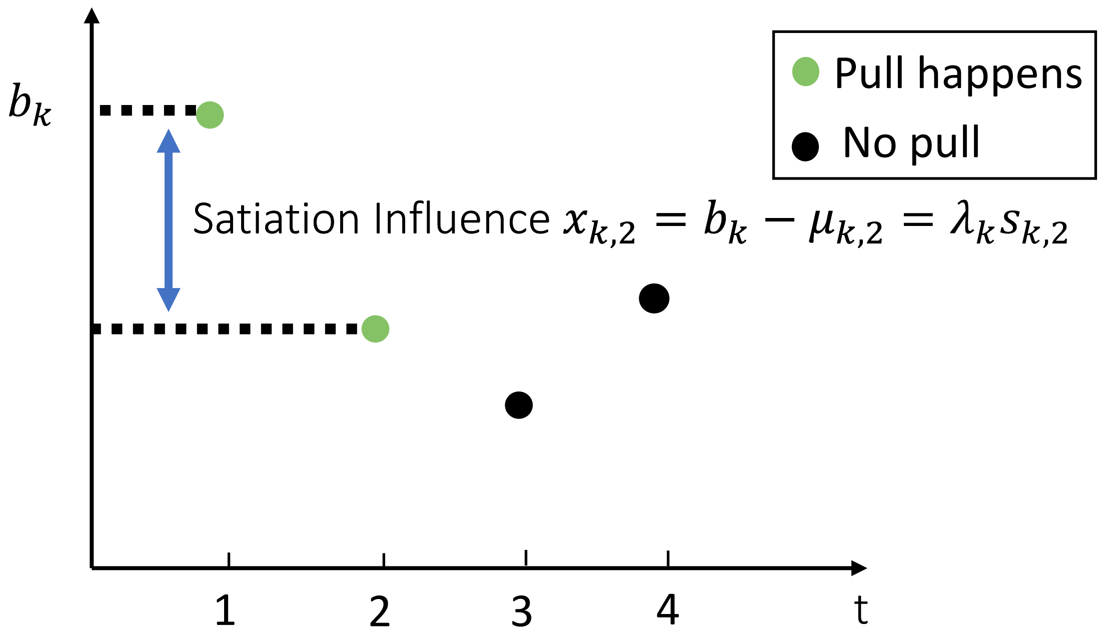
Rebounding bandits for modeling satiation effects. **Leqi L**, Kılınç-Karzan F, Lipton Z, Montgomery A. NeurIPS '21.

Rebounding bandits

Mean reward characteristics:

- Decline with consecutive pulls
- Rebound towards the baseline with disuse

Mean Reward for Arm k



Stochastic Linear Dynamical System

Reward model

$$\mu_{k,t} = b_k - \underbrace{x_{k,t}}_{\text{Satiation Influence}}$$

$$= b_k - \underbrace{\lambda_k}_{\text{Exposure Influence Factor}} \underbrace{s_{k,t}}_{\text{Satiation}}$$

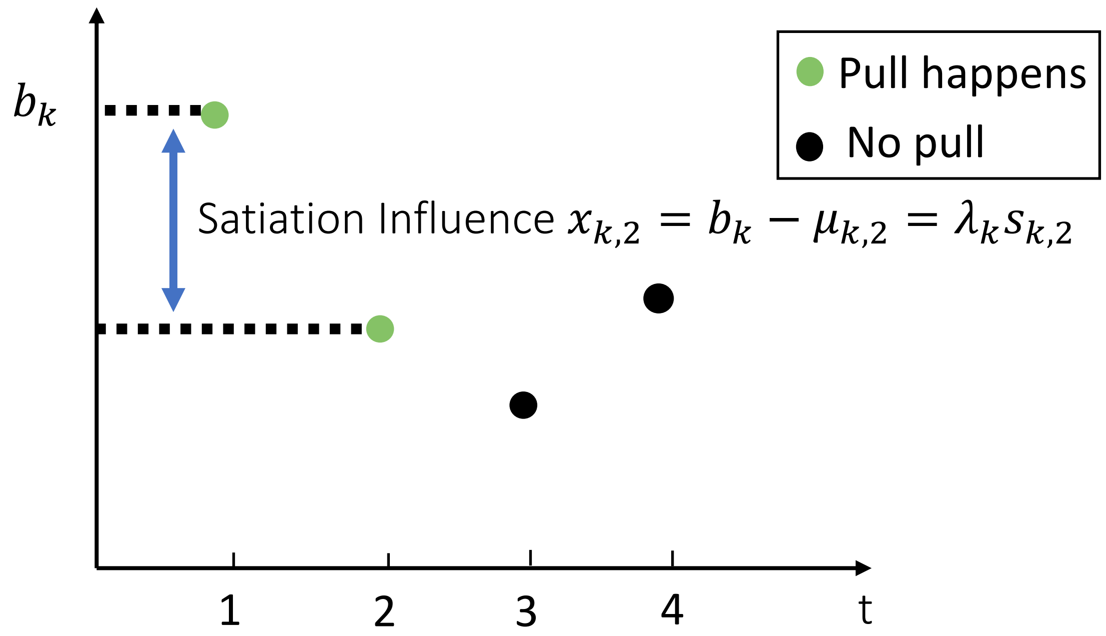
Rebounding bandits for modeling satiation effects. **Leqi L**, Kılınç-Karzan F, Lipton Z, Montgomery A. NeurIPS '21.

Rebounding bandits

Mean reward characteristics:

- Decline with consecutive pulls
- Rebound towards the baseline with disuse

Mean Reward for Arm k



Stochastic Linear Dynamical System

Reward model

$$\mu_{k,t} = b_k - x_{k,t} \quad \text{Satiation Influence}$$
$$= b_k - \lambda_k \, s_{k,t}$$

Exposure Influence Factor Satiation

Satiation Dynamics

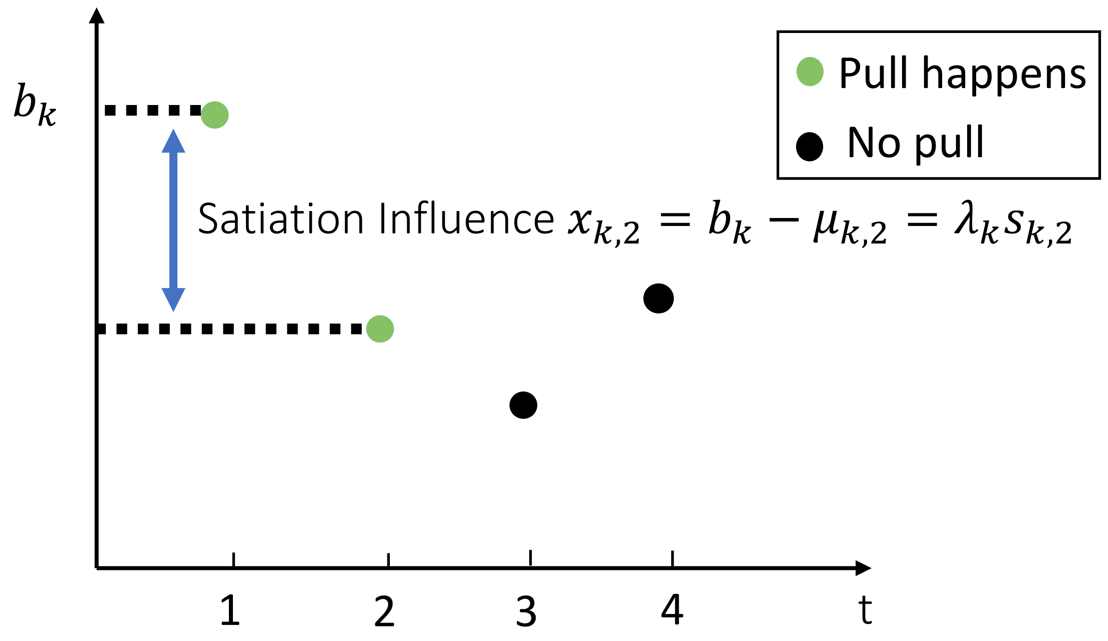
- ❑ Increase with consecutive pulls
- ❑ decrease towards zero with disuse

Rebounding bandits for modeling satiation effects. **Leqi L**, Kılınç-Karzan F, Lipton Z, Montgomery A. NeurIPS '21.

Rebounding bandits

Mean reward characteristics:
- Decline with consecutive pulls
- Rebound towards the baseline with disuse

Mean Reward for Arm k



Stochastic Linear Dynamical System

Reward model

$$\mu_{k,t} = b_k - \lambda_k s_{k,t}$$

Satiation

Satiation Dynamics

$$s_{k,t} = \gamma_k (s_{k,t-1} + u_{k,t-1})$$

Satiation Retention Factor

Pulled vs Not Last-step consumption

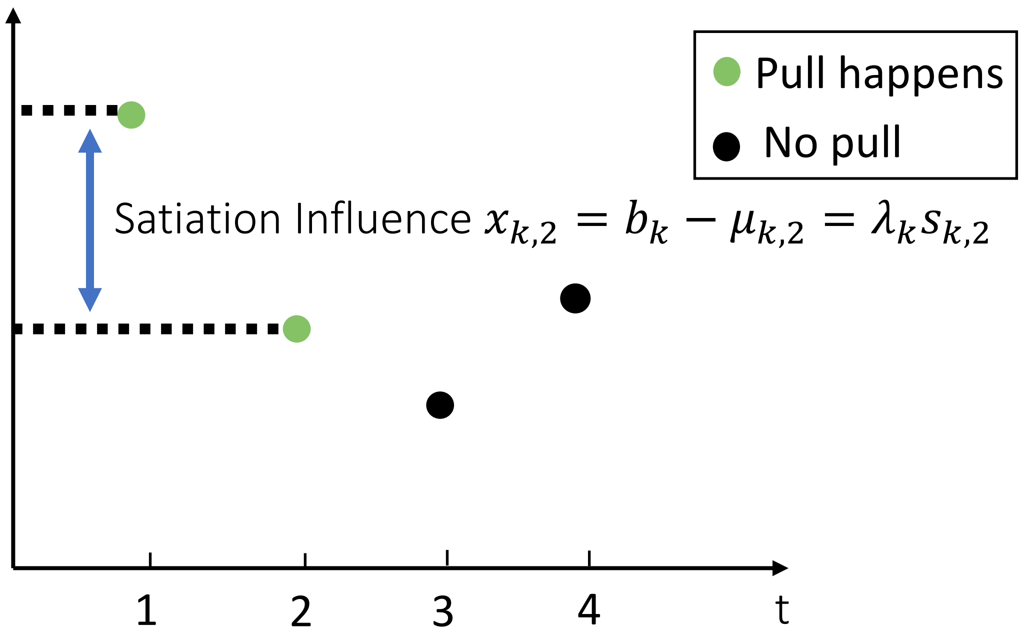
Rebounding bandits for modeling satiation effects. **Leqi L**, Kılınç-Karzan F, Lipton Z, Montgomery A. NeurIPS '21.

Rebounding bandits

Mean reward characteristics:

- Decline with consecutive pulls
- Rebound towards the baseline with disuse

Mean Reward for Arm k



Stochastic Linear Dynamical System

Reward model

$$\mu_{k,t} = b_k - \lambda_k s_{k,t}$$

Satiation

Satiation Dynamics

$$s_{k,t} = \gamma_k (s_{k,t-1} + u_{k,t-1}) + z_{k,t-1}$$

Satiation Retention Factor

Pulled vs Not Last-step consumption

Noise

Rebounding bandits for modeling satiation effects. **Leqi L**, Kılınç-Karzan F, Lipton Z, Montgomery A. NeurIPS '21.

Rebounding bandits

Mean reward characteristics:

- Decline with consecutive pulls
- Rebound towards the baseline with disuse

Stochastic Linear Dynamical System

Reward model

$$\mu_{k,t} = b_k - \lambda_k \boxed{s_{k,t}}$$

Satiation

Satiation Dynamics

$$s_{k,t} = \boxed{\gamma_k}(s_{k,t-1} + \boxed{u_{k,t-1}}) + \boxed{z_{k,t-1}}$$

Satiation Retention Factor | Pulled vs Not | Noise

Learner's Goal

Maximize the expected cumulative reward:

$$\max_{\pi_1,\ldots,\pi_T} \mathbb{E}\left[\sum_{t=1}^{T} \mu_{\pi_t,t}\right]$$

Rebounding bandits for modeling satiation effects. **Leqi L**, Kılınç-Karzan F, Lipton Z, Montgomery A. NeurIPS '21.

Rebounding bandits

Mean reward characteristics:

- Decline with consecutive pulls
- Rebound towards the baseline with disuse

Stochastic Linear Dynamical System

Reward model

$$\mu_{k,t} = b_k - \lambda_k \underbrace{s_{k,t}}_{\text{Satiation}}$$

Satiation Dynamics

$$s_{k,t} = \underbrace{\gamma_k}_{\text{Satiation Retention Factor}}(s_{k,t-1} + \underbrace{u_{k,t-1}}_{\text{Pulled vs Not}}) + \underbrace{z_{k,t-1}}_{\text{Noise}}$$

Learner's Goal

Maximize the expected cumulative reward:

$$\max_{\pi_1,\dots,\pi_T} \mathbb{E}\left[\sum_{t=1}^{T} \mu_{\pi_t,t}\right]$$

Key Challenge:

- ❑ Unknown satiation dynamics
- ❑ Requires planning

Rebounding bandits for modeling satiation effects. **Leqi L**, Kılınç-Karzan F, Lipton Z, Montgomery A. NeurIPS '21.

Rebounding bandits

Mean reward characteristics:

- Decline with consecutive pulls
- Rebound towards the baseline with disuse

Learner's Goal

Maximize the expected cumulative reward:

$$\max_{\pi_1,\ldots,\pi_T} \mathbb{E}\left[\sum_{t=1}^{T} \mu_{\pi_t,t}\right]$$

Key Challenge:

- ❑ Unknown satiation dynamics
- ❑ Requires planning

Regret

REGRET = Cumulative reward for oracle policy − Cumulative reward collected

Best fixed arm

Ignores planning

Rebounding bandits for modeling satiation effects. **Leqi L**, Kılınç-Karzan F, Lipton Z, Montgomery A. NeurIPS '21.

Rebounding bandits

Mean reward characteristics:

- Decline with consecutive pulls
- Rebound towards the baseline with disuse

Learner's Goal

Maximize the expected cumulative reward:

$$\max_{\pi_1,\dots,\pi_T} \mathbb{E}\left[\sum_{t=1}^{T} \mu_{\pi_t,t}\right]$$

Key Challenge:

❑ Unknown satiation dynamics

❑ Requires planning

Regret

Pike-Burke et. al. NeurIPS 2019

w-step lookahead REGRET = Cumulative reward for oracle policy − Cumulative reward collected

Best policy given known dynamics & plans every w steps.

No regret algorithm: regret of learner grows sublinearly in T.

Rebounding bandits for modeling satiation effects. **Leqi L**, Kılınç-Karzan F, Lipton Z, Montgomery A. NeurIPS '21.

Algorithm: Explore-Estimate-Plan

1. **Explore:**
play CYCLE or REPEAT for $T^{2/3}$ steps

2. **Estimate:**
Least squares estimation of the satiation dynamics (γ_k, λ_k)

3. **Plan:**
Solve for w-lookahead policy using estimated dynamics

Rebounding bandits for modeling satiation effects. **Leqi L**, Kılınç-Karzan F, Lipton Z, Montgomery A. NeurIPS '21.

Algorithm: Explore-Estimate-Plan

1. Explore:
play CYCLE or REPEAT for $T^{2/3}$ steps

2. Estimate:
Least squares estimation of the satiation dynamics (γ_k, λ_k)

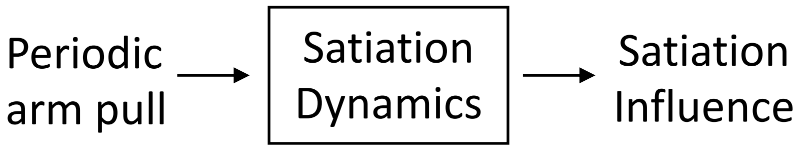
3. Plan:
Solve for w-lookahead policy using estimated dynamics

Periodic arm pull $\rightarrow$ $\boxed{\text{Satiation Dynamics}}$ $\rightarrow$ Satiation Influence

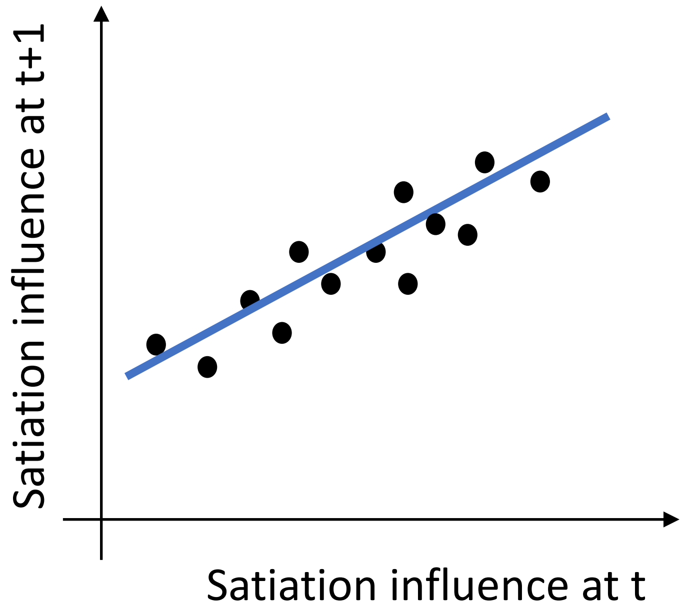
Rebounding bandits for modeling satiation effects. **Leqi L**, Kılınç-Karzan F, Lipton Z, Montgomery A. NeurIPS ’21.

Algorithm: Explore-Estimate-Plan

1. Explore:
play CYCLE or REPEAT for $T^{2/3}$ steps



2. Estimate:
Least squares estimation of the satiation dynamics (γ_k, λ_k)

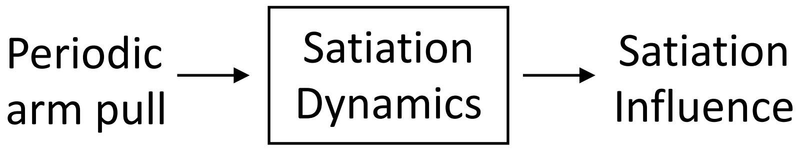


3. Plan:
Solve for w-lookahead policy using estimated dynamics

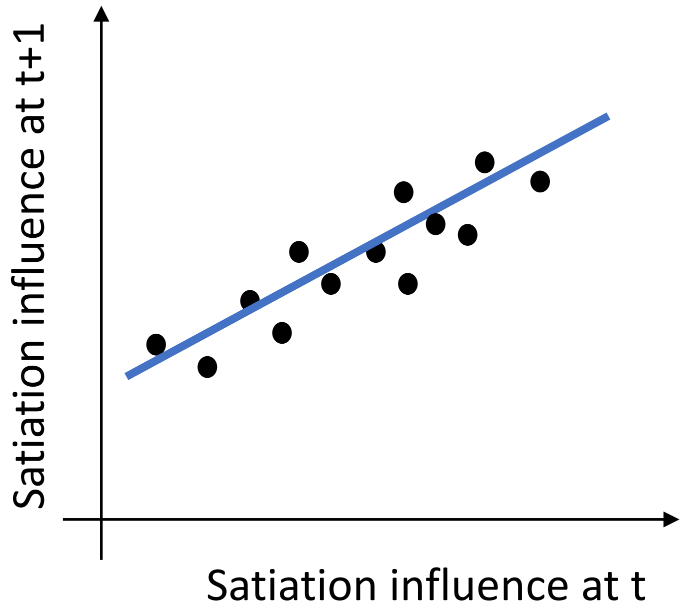
Rebounding bandits for modeling satiation effects. **Leqi L**, Kılınç-Karzan F, Lipton Z, Montgomery A. NeurIPS ’21.

Algorithm: Explore-Estimate-Plan

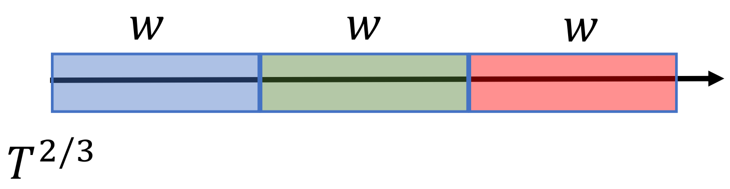
1. Explore:
play CYCLE or REPEAT for $T^{2/3}$ steps



2. Estimate:
Least squares estimation of the satiation dynamics (γ_k, λ_k)



3. Plan:
Solve for w-lookahead policy using estimated dynamics



Rebounding bandits for modeling satiation effects. **Leqi L**, Kılınç-Karzan F, Lipton Z, Montgomery A. NeurIPS ’21.

Algorithm: Explore-Estimate-Plan

1. Explore:
play CYCLE or REPEAT for $T^{2/3}$ steps

2. Estimate:
Least squares estimation of the satiation dynamics (γ_k, λ_k)

3. Plan:
Solve for w-lookahead policy using estimated dynamics

Theorem (Informal)

For T large enough, Explore-Estimate-Plan incurs $O(\sqrt{K}T^{2/3}\log(T))$ w-step lookahead regret.

Rebounding bandits for modeling satiation effects. **Leqi L**, Kılınç-Karzan F, Lipton Z, Montgomery A. NeurIPS '21.

Message

Designing bandit algorithms that interact with people requires more realistic assumptions on **human preferences**.

Rebounding bandits: use dynamical systems to model evolving preferences.

Outline

- Broader landscape of Learning from Human Feedback
- Two case studies:
  - Language Modeling: **RLHF** for aligning model with **user intent**
  - Recommender systems: **Multi-armed bandits** accounting for **evolving** human preferences
- What's next?

What’s next?

Learning from human feedback is a rich research area!

Open research questions on:

- What are the forms of human feedback?
- How to collect “good” human feedback?
- How to use human feedback? (Modeling human feedback and improving the ML system of interest.)

Depending on the application context, we will develop different solutions. RL will be a central theme, as ML systems’ interactions with human users are almost never one-shot!

Reinforcement Learning from Human Feedback

Leqi Liu
Princeton Language & Intelligence
University of Texas, Austin
April 16th, 2024

PRINCETON
Language + Intelligence