



# Neural Contextual Bandits for Personalized Recommendation



**Yikun Ban**



**Yunzhe Qi**



**Jingrui He**

**University of Illinois Urbana-Champaign**

{Yikunb2, Yunzheq2, Jingrui}@illinois.edu

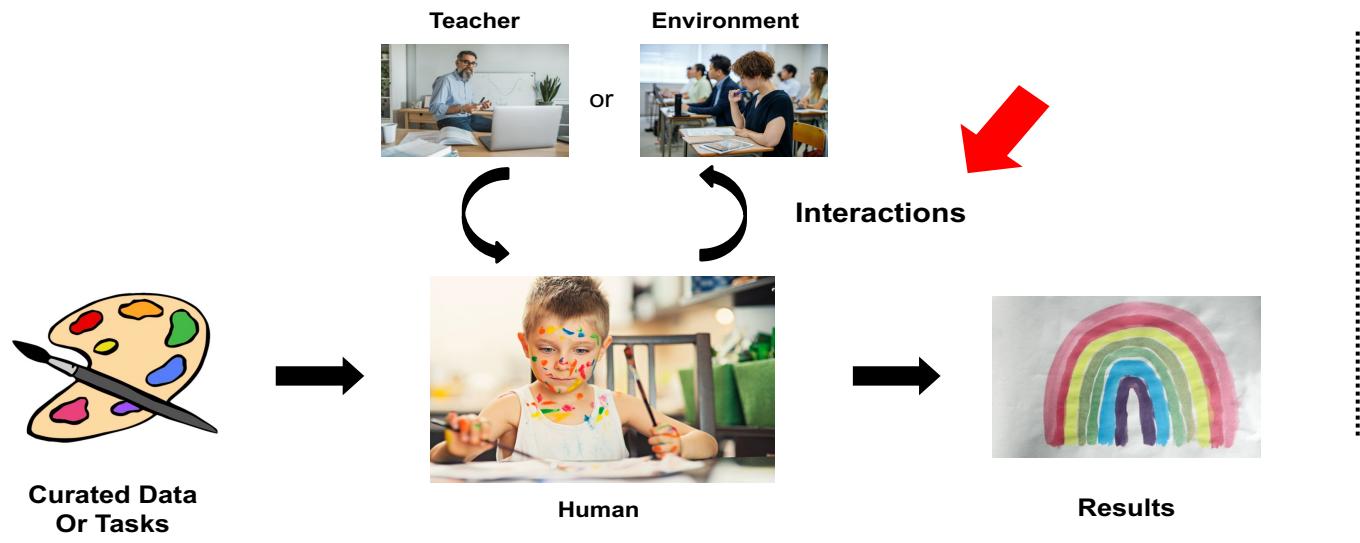
Time: 9:00 AM – 12:30 PM, 13 May 2024

Location: Virgo 1, Resorts World Sentosa Convention Centre, Singapore

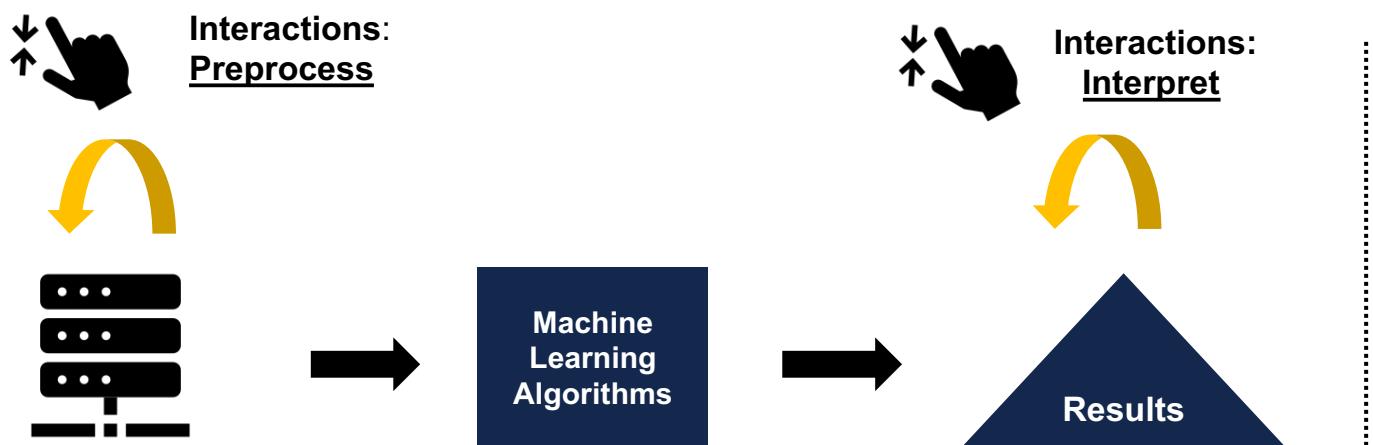
Website: [www.banyikun.com/wwwtutorial/](http://www.banyikun.com/wwwtutorial/)



# Interactions in Machine Learning



## Interactive Learning (Human)



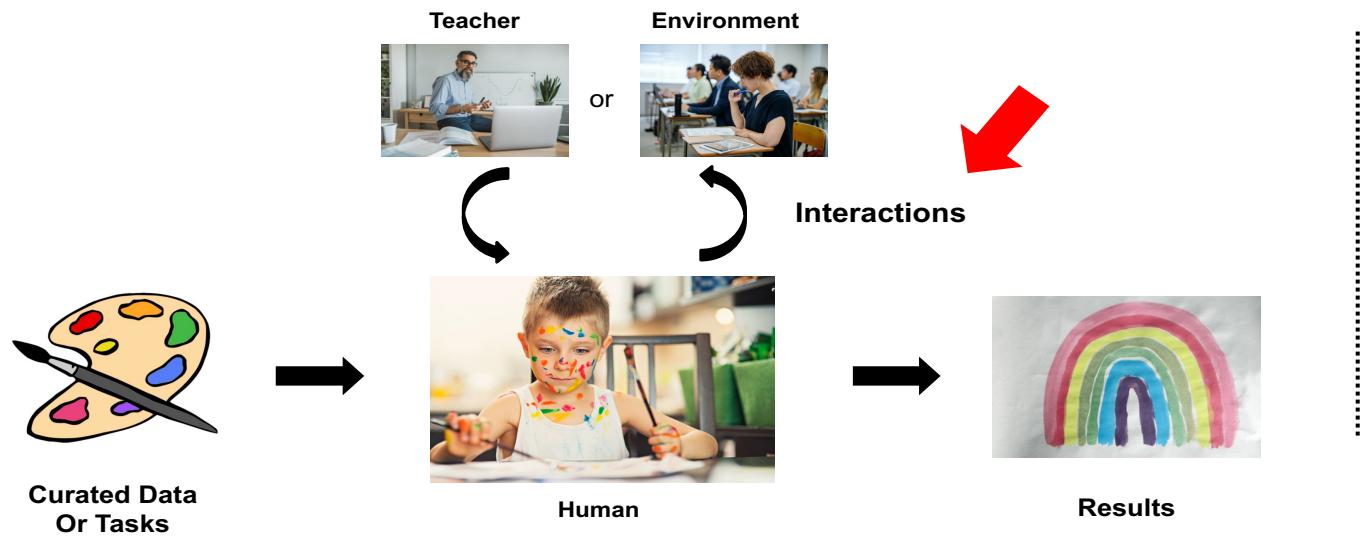
## Machine Learning (Conventional)



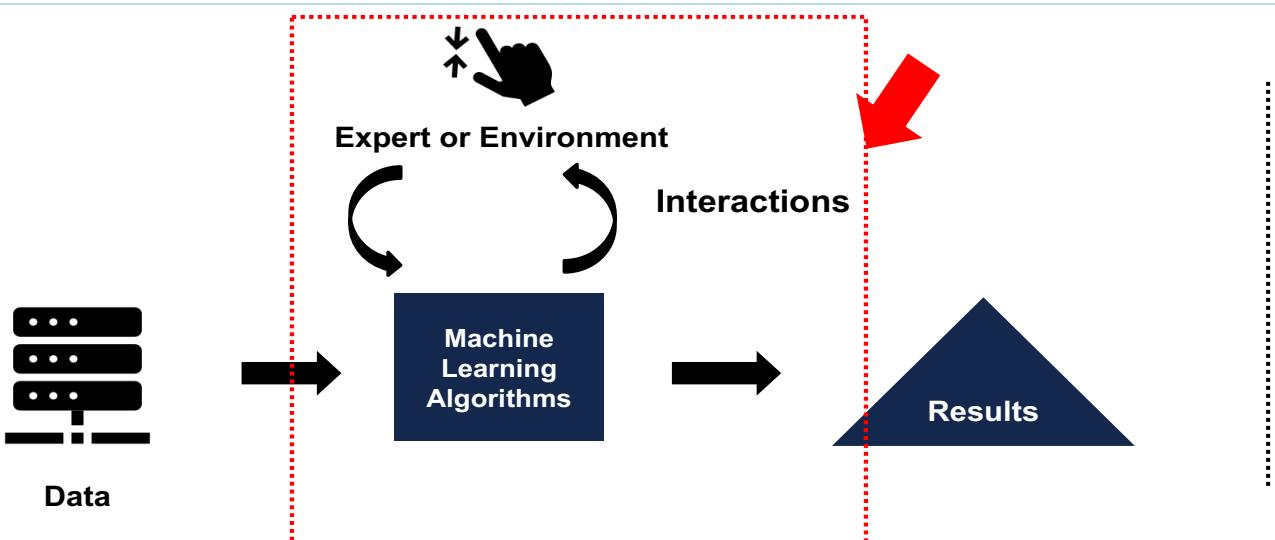
### Data

1. Ernst, Damien, and Arthur Louette. "Introduction to reinforcement learning." 2024.
2. Fails, Jerry Alan, and Dan R. Olsen Jr. "Interactive machine learning." *Proceedings of the 8th international conference on Intelligent user interfaces*. 2003.
3. Teso, Stefano, and Kristian Kersting. "Explanatory interactive machine learning." *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 2019.

# Interactions in Machine Learning



**Interactive Learning  
(Human)**



**Interactive Machine Learning**



1. Ernst, Damien, and Arthur Louette. "Introduction to reinforcement learning." 2024.
2. Fails, Jerry Alan, and Dan R. Olsen Jr. "Interactive machine learning." *Proceedings of the 8th international conference on Intelligent user interfaces*. 2003.
3. Teso, Stefano, and Kristian Kersting. "Explanatory interactive machine learning." *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 2019.

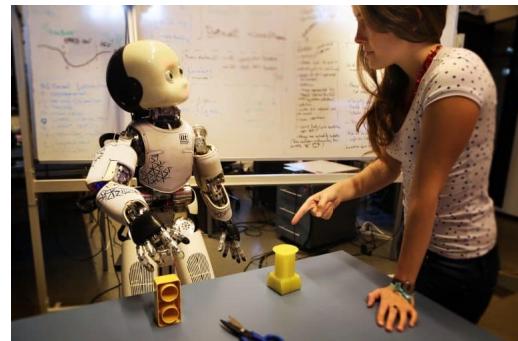
# Interactive Machine Learning and Applications



➤ Interactive Machine Learning (IML) is the core of Artificial Intelligence (AI).



(1) Recommender Systems



(2) Robot Learning

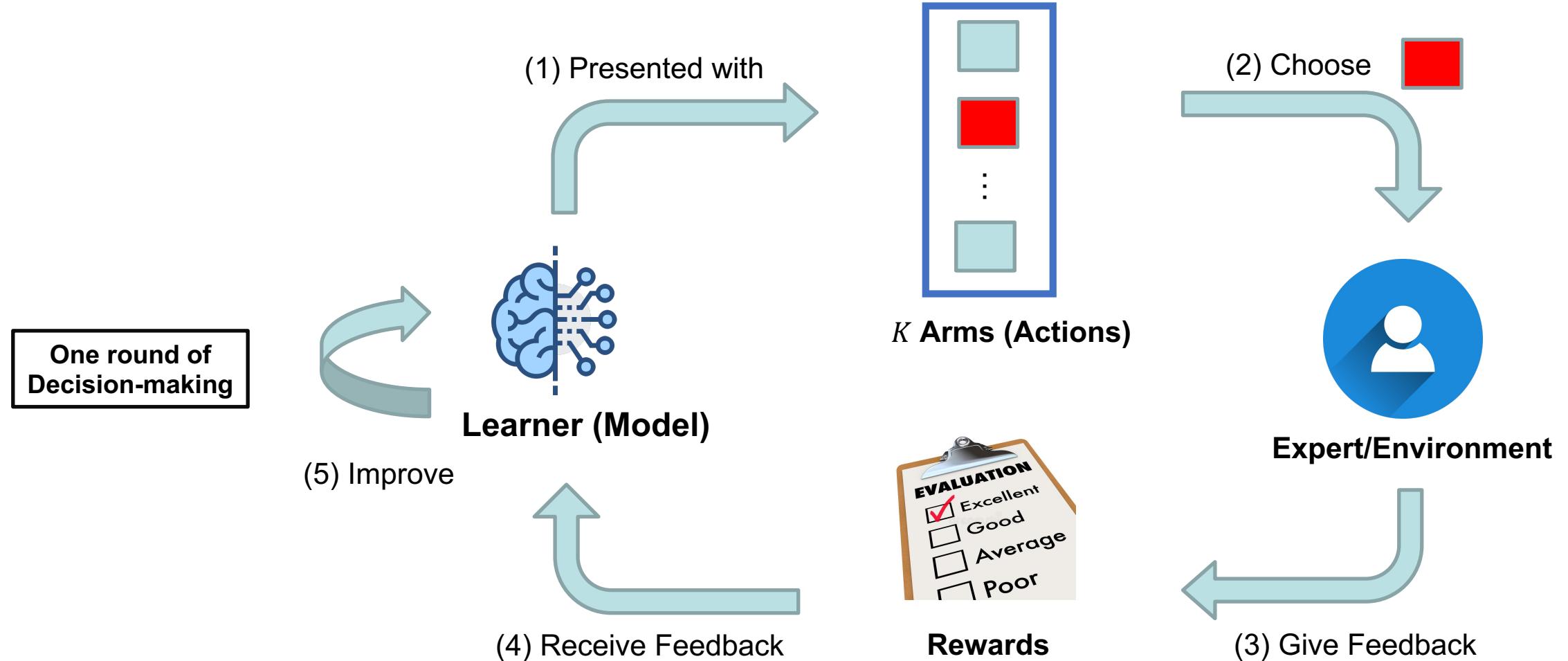


(3) Language Model

1. Ernst, Damien, and Arthur Louette. "Introduction to reinforcement learning." 2024.
2. Fails, Jerry Alan, and Dan R. Olsen Jr. "Interactive machine learning." *Proceedings of the 8th international conference on Intelligent user interfaces*. 2003.
3. Teso, Stefano, and Kristian Kersting. "Explanatory interactive machine learning." *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 2019.

# Sequential Decision-Making: Bandits Formulation

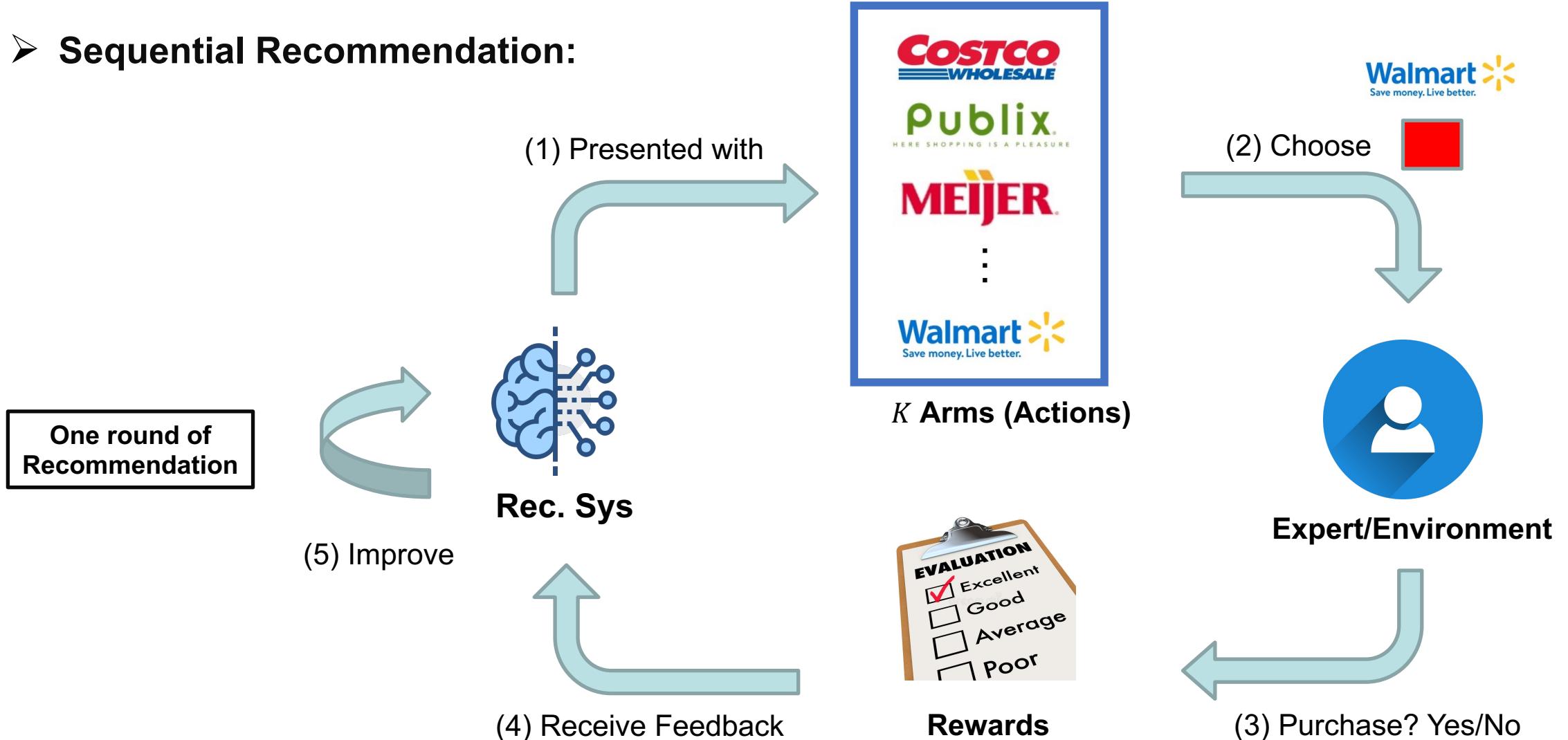
- Many IML scenarios can be formulated as **sequential decision-making**.



1. Ernst, Damien, and Arthur Louette. "Introduction to reinforcement learning." 2024.
2. Fails, Jerry Alan, and Dan R. Olsen Jr. "Interactive machine learning." *Proceedings of the 8th international conference on Intelligent user interfaces*. 2003.
3. Teso, Stefano, and Kristian Kersting. "Explanatory interactive machine learning." *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 2019.

# Sequential Recommendation: Bandits Formulation

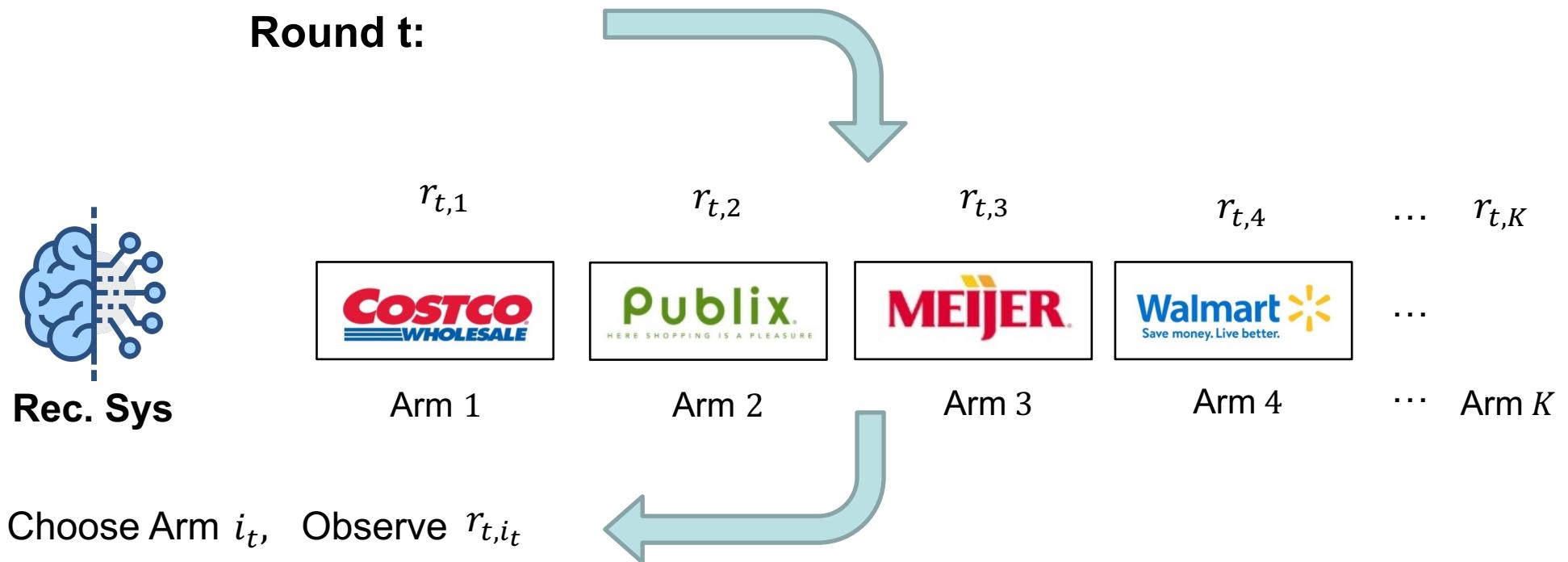
## ➤ Sequential Recommendation:



1. Ernst, Damien, and Arthur Louette. "Introduction to reinforcement learning." 2024.
2. Fails, Jerry Alan, and Dan R. Olsen Jr. "Interactive machine learning." *Proceedings of the 8th international conference on Intelligent user interfaces*. 2003.
3. Teso, Stefano, and Kristian Kersting. "Explanatory interactive machine learning." *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 2019.

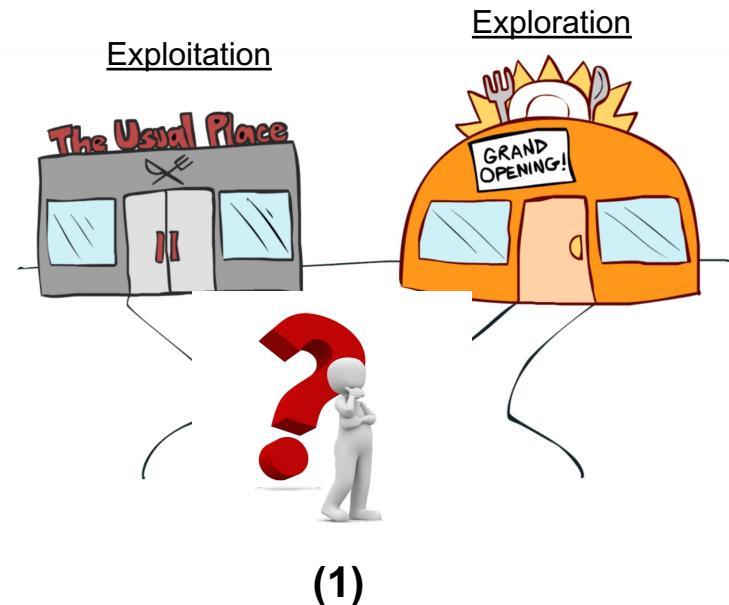
# Sequential Recommendation: Objective

**Goal:** Maximize  $\sum_{t=1}^T \mathbb{E}[r_{t,i_t}]$       Or      Minimize  $\sum_{t=1}^T (\mathbb{E}[r_{t,i_t^*}] - \mathbb{E}[r_{t,i_t}])$ , where  $i_t^* = \arg \max_{i \in [K]} \mathbb{E}[r_{t,i}]$ .



# Exploitation VS Exploration in Sequential Decision-Making

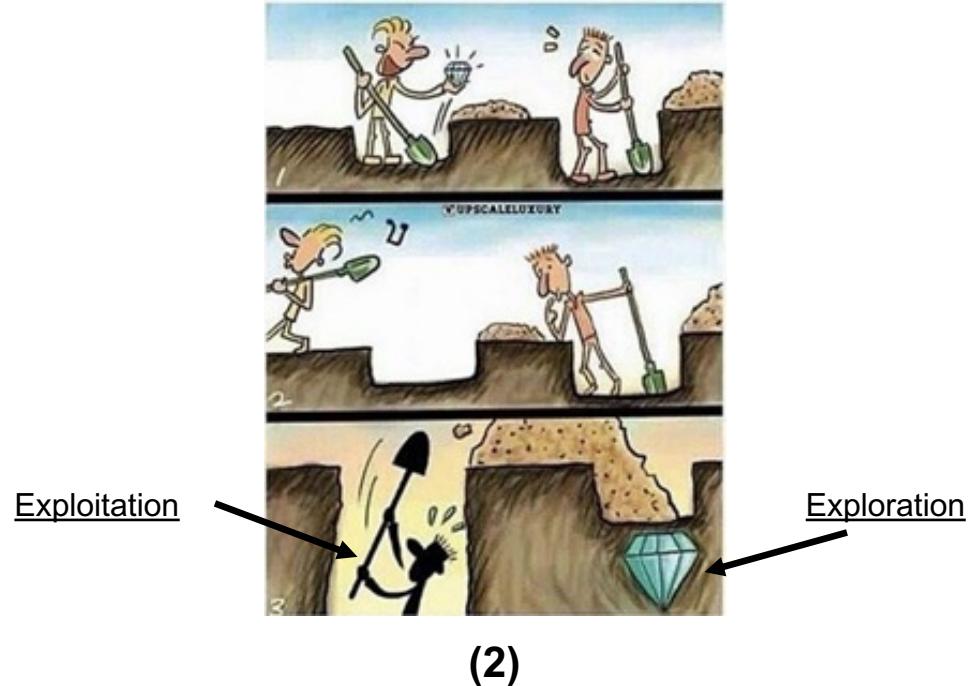
- Dilemma of **exploitation** and **exploration** is ubiquitous in **human decision-making**.



## Exploitation:

**Exploit** past data or observations.  
E.g., estimation by a greedy model

**VS**



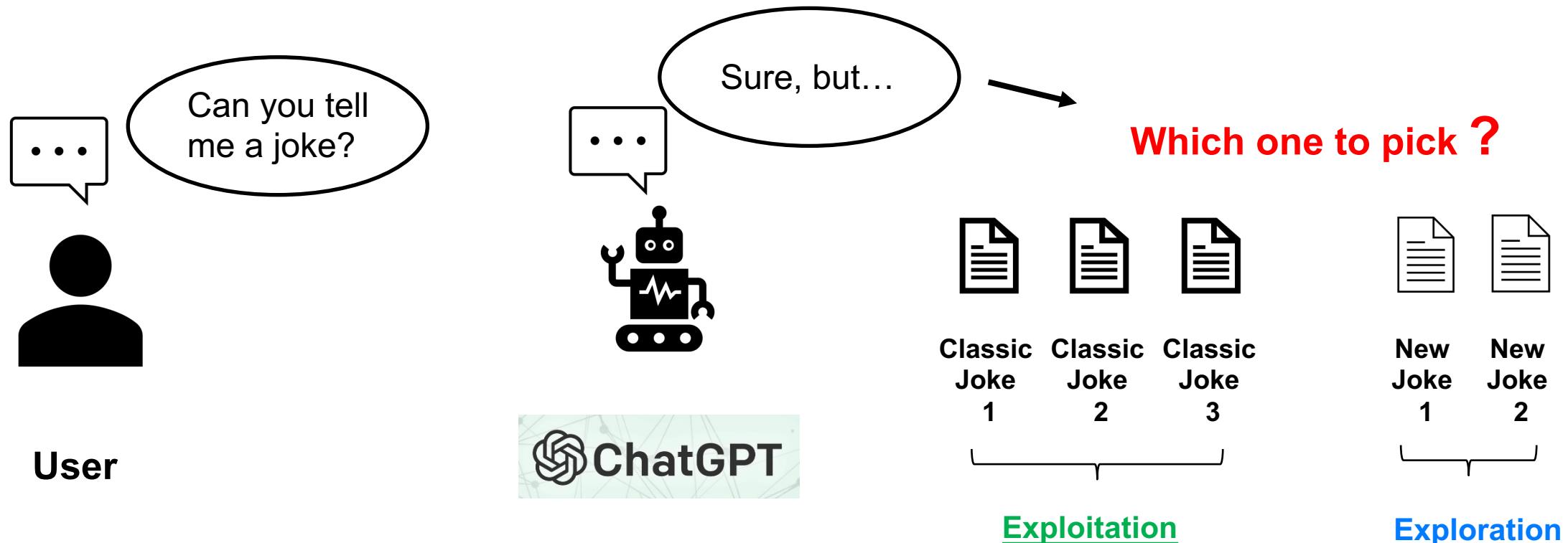
## Exploration:

**Explore** new knowledge for long-term benefit.  
E.g., take uncertain actions

# Exploitation VS Exploration in Sequential Recommendation



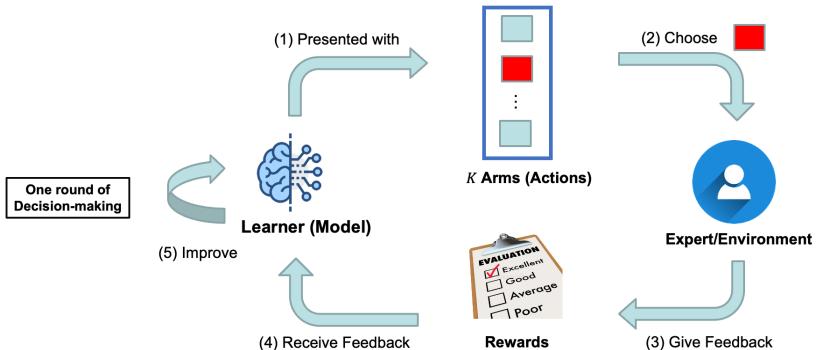
- Dilemma of **exploitation** and **exploration** is a fundamental problem in sequential decision-making.



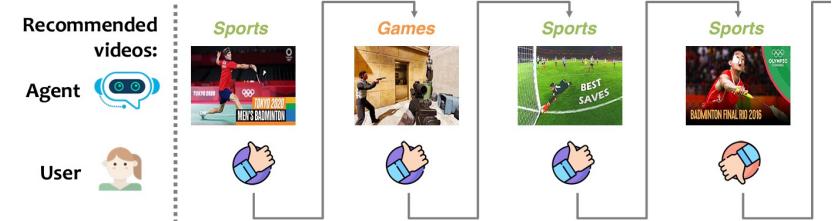
# Advantages of Bandit-based Methods

➤ No Requirement for Large Collected Data

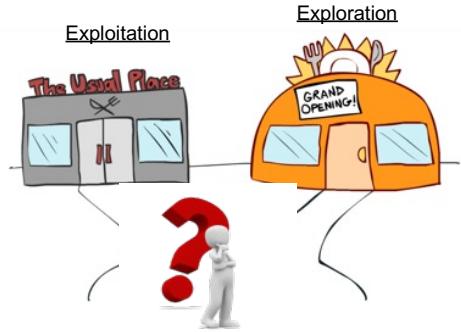
	Book	Bag	Headphones	Game Controller
A	✓	✗	✓	✓
B	✓	✓	✗	✗
C	✓	✓	✗	
D	✗		✓	
E	✓	✓	?	✗



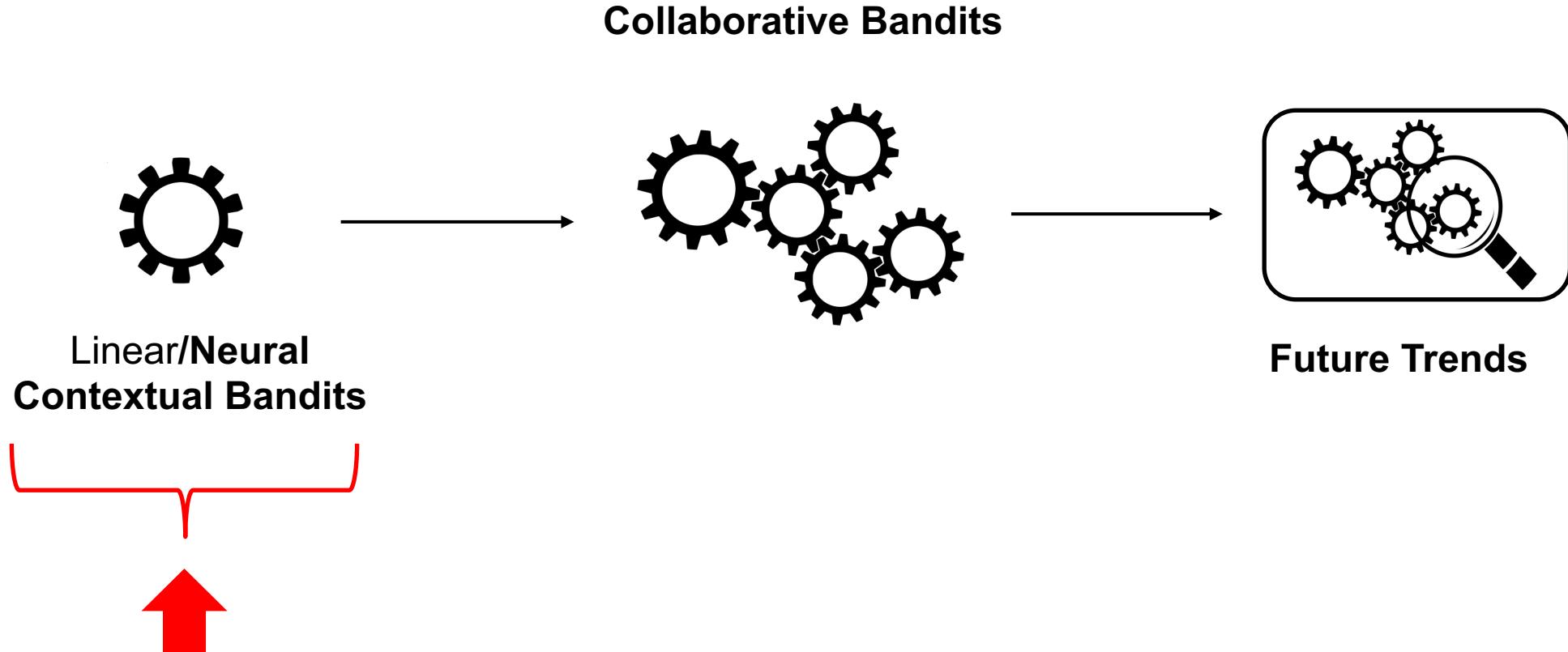
➤ Adapt over Time

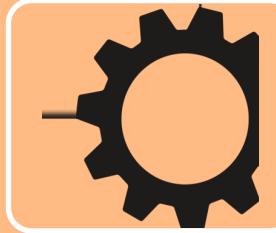


➤ Explicit Exploration



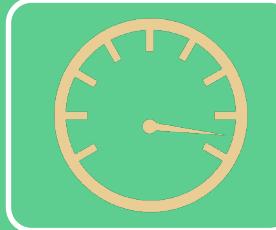
# Roadmap





## Fundamental Exploration

- Upper Confidence Bound
- Thompson Sampling
- Exploration Network



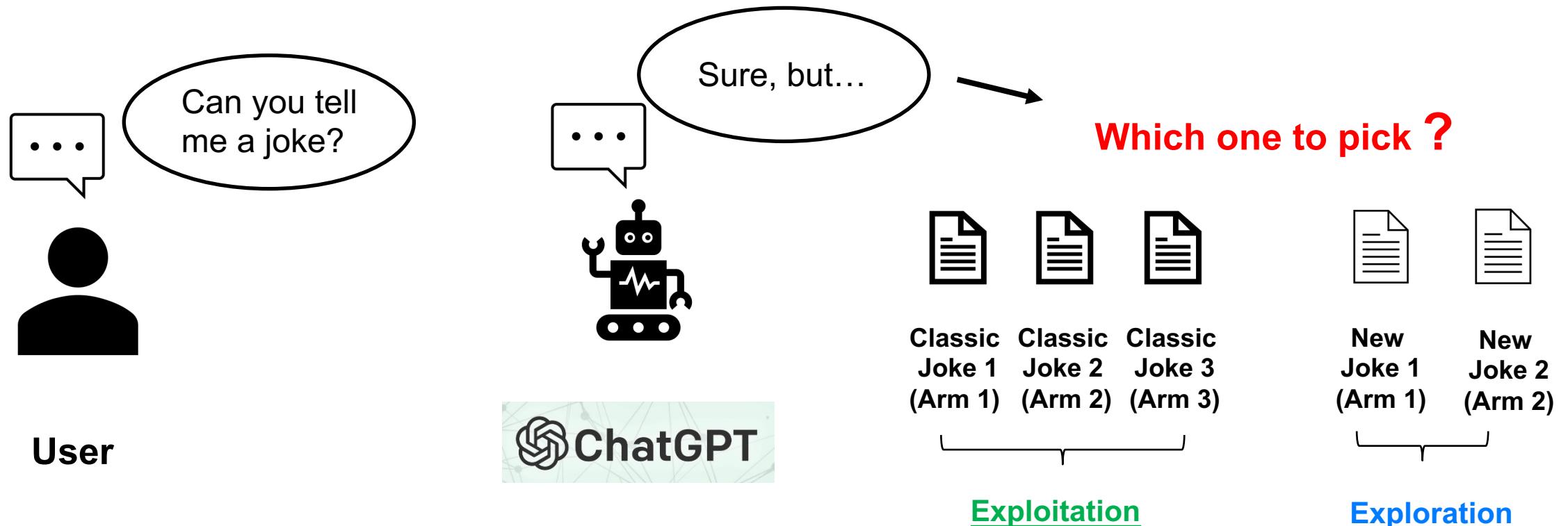
## Efficient Exploration

- Neural Linear UCB
- Neural Network with Perturbed Reward
- Inverse Weight Gap Strategy

# Background



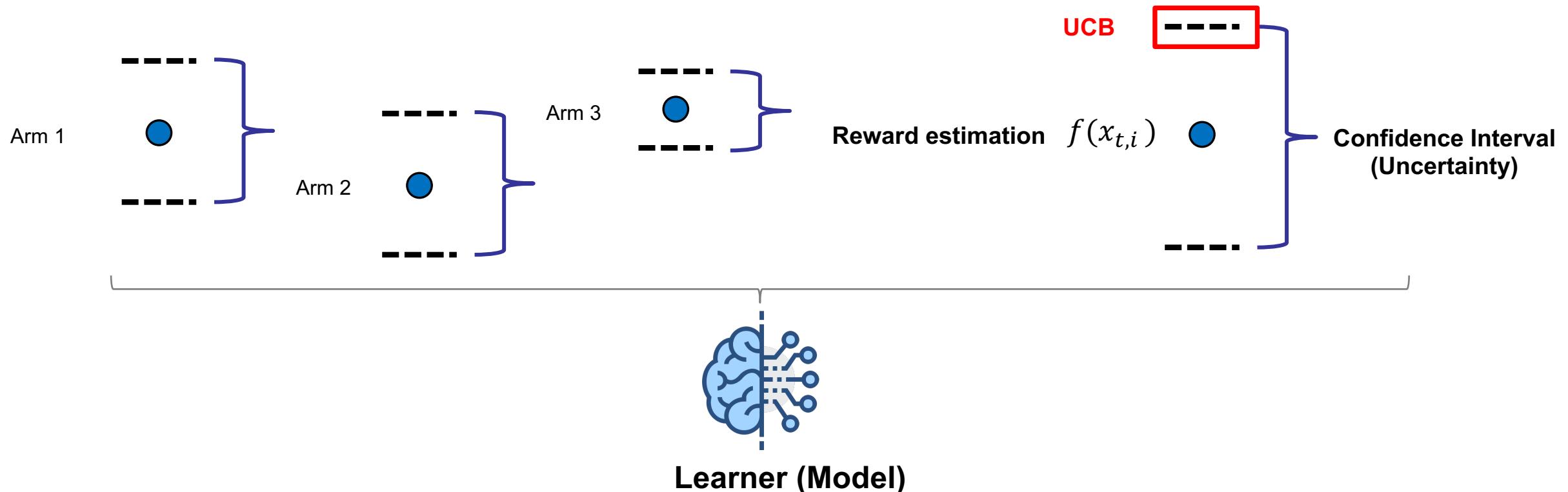
- Popular existing exploration strategies.
  - **$\epsilon$ -greedy:** With probability  $1 - \epsilon$ , greedily choose one arm according to history;  
Otherwise, choose an arm randomly.



# Background

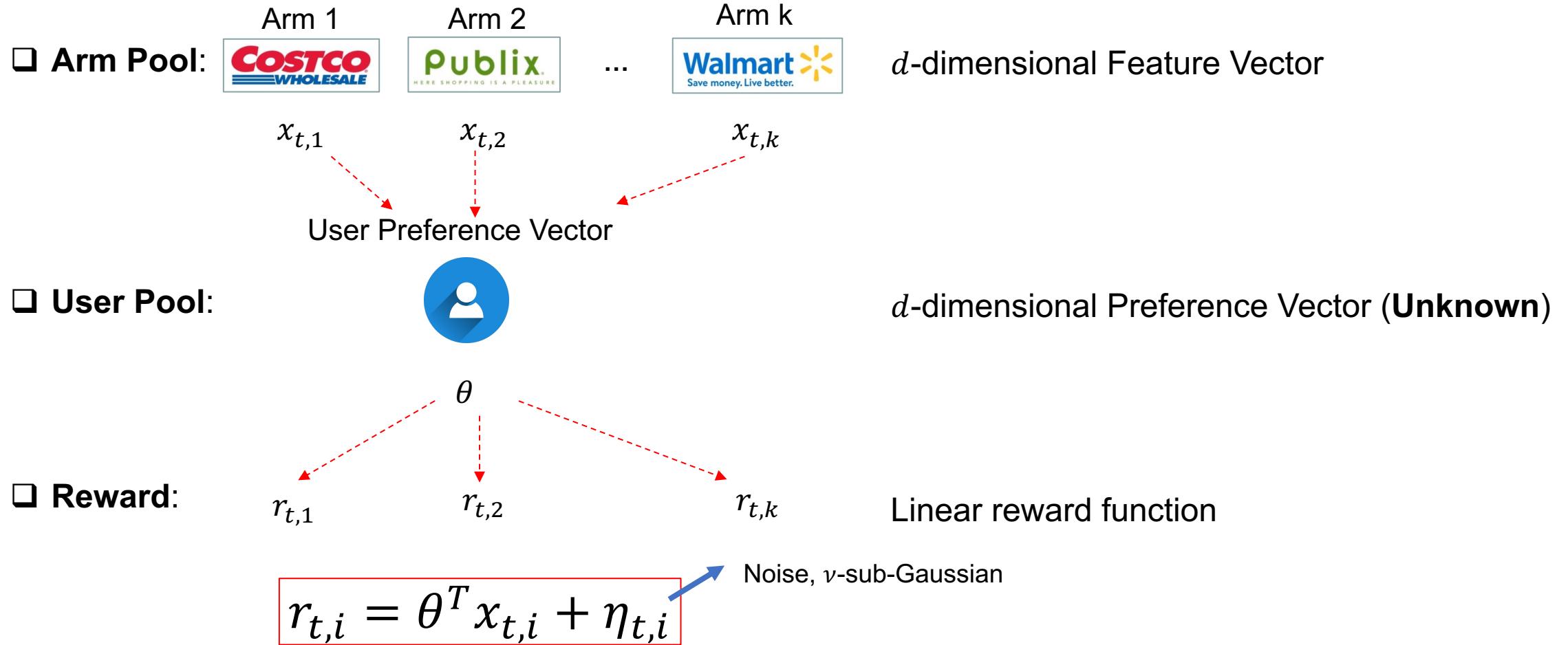


- Popular existing exploration strategies.
  - **$\epsilon$ -greedy**: With probability  $1 - \epsilon$ , greedily choose one arm according to history;  
Otherwise, choose an arm randomly.
  - **Upper Confidence Bound [1]**:



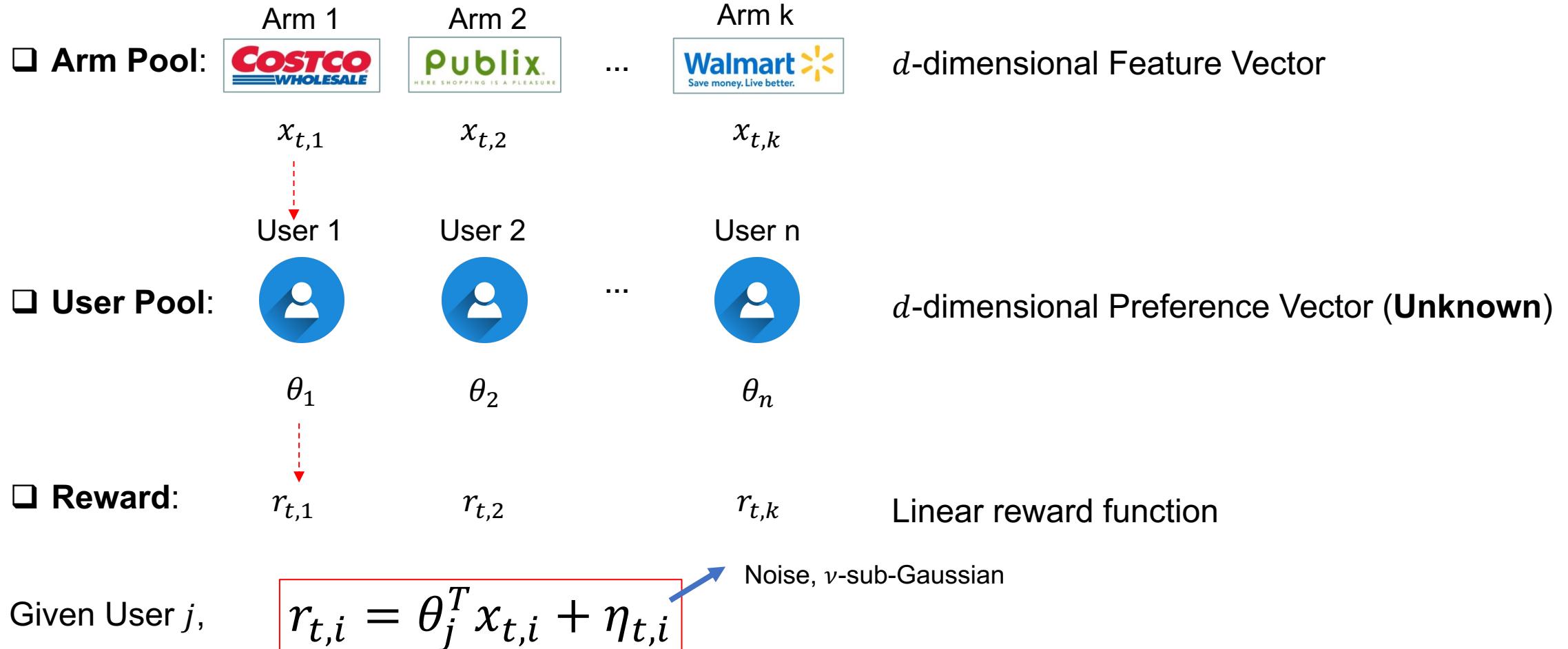
# Linear UCB: Joint Problem Definition

In round  $t$ : A user is serving

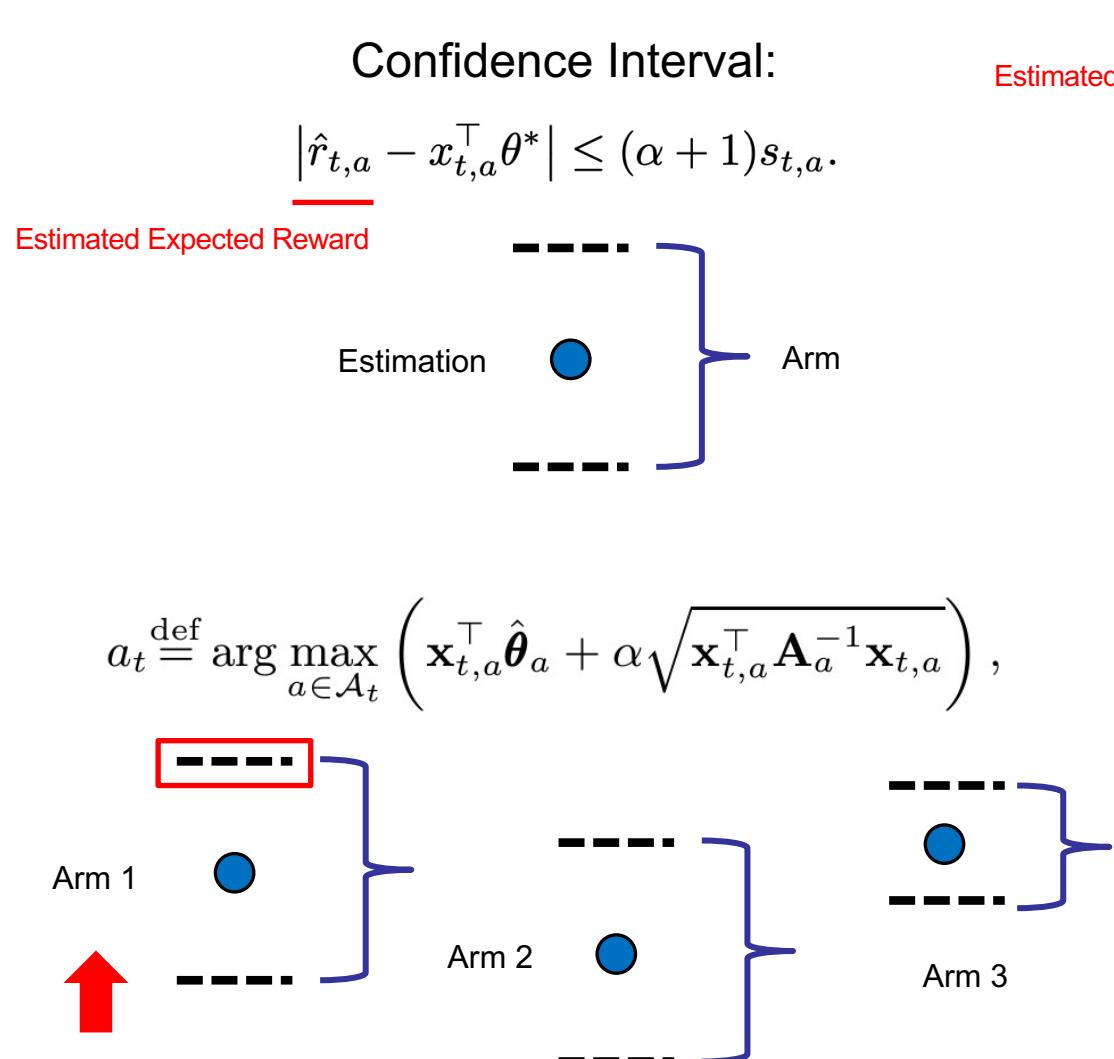


# Linear UCB: Disjoint Problem Definition

In round  $t$ : A user is serving



# Linear UCB



$$a_t \stackrel{\text{def}}{=} \arg \max_{a \in \mathcal{A}_t} \left( \mathbf{x}_{t,a}^\top \hat{\theta}_a + \alpha \sqrt{\mathbf{x}_{t,a}^\top \mathbf{A}_a^{-1} \mathbf{x}_{t,a}} \right),$$

## Joint Linear Models

Estimated by Ridge Regression

```

for  $t = 1, 2, 3, \dots, T$  do -- A user is serving in each round
     $\theta_t \leftarrow A^{-1}b$  (Item 1)(Item 2) ...
    Observe  $K$  features,  $x_{t,1}, x_{t,2}, \dots, x_{t,K} \in \mathbb{R}^d$ 
    for  $a = 1, 2, \dots, K$  do
         $p_{t,a} \leftarrow \theta_t^\top x_{t,a} + \alpha \sqrt{x_{t,a}^\top A^{-1} x_{t,a}}$  {Computes
            upper confidence bound}
    end for Exploitation Exploration
    Choose action  $a_t = \arg \max_a p_{t,a}$  with ties broken arbitrarily
    Observe payoff  $r_t \in \{0, 1\}$ 
     $A \leftarrow A + x_{t,a_t} x_{t,a_t}^\top$ 
     $b \leftarrow b + x_{t,a_t} r_t$ 
end for

```

Estimated by Ridge Regression

# Linear UCB: Regret Analysis

## ➤ Confidence Interval:

Estimated by Ridge Regression

With high probability,

$$|\hat{r}_{t,a} - x_{t,a}^\top \theta^*| \leq (\alpha + 1)s_{t,a}.$$

where

$$s_{t,a} = \sqrt{x_{t,a}^\top A_t^{-1} x_{t,a}} \in \mathbb{R}_+$$

$$\begin{aligned} \hat{r}_{t,a} - x_{t,a}^\top \theta^* &= x_{t,a}^\top \theta_t - x_{t,a}^\top \theta^* \\ &= x_{t,a}^\top A_t^{-1} b_t - x_{t,a}^\top A_t^{-1} (I_d + D_t^\top D_t) \theta^* \\ &= x_{t,a}^\top A_t^{-1} D_t^\top y_t - x_{t,a}^\top A_t^{-1} (\theta^* + D_t^\top D_t \theta^*) \\ &= x_{t,a}^\top A_t^{-1} D_t^\top (y_t - D_t \theta^*) - x_{t,a}^\top A_t^{-1} \theta^*, \end{aligned}$$

## ➤ Regret Upper Bound

$$O\left(\sqrt{Td \ln^3(KT \ln(T)/\delta)}\right).$$

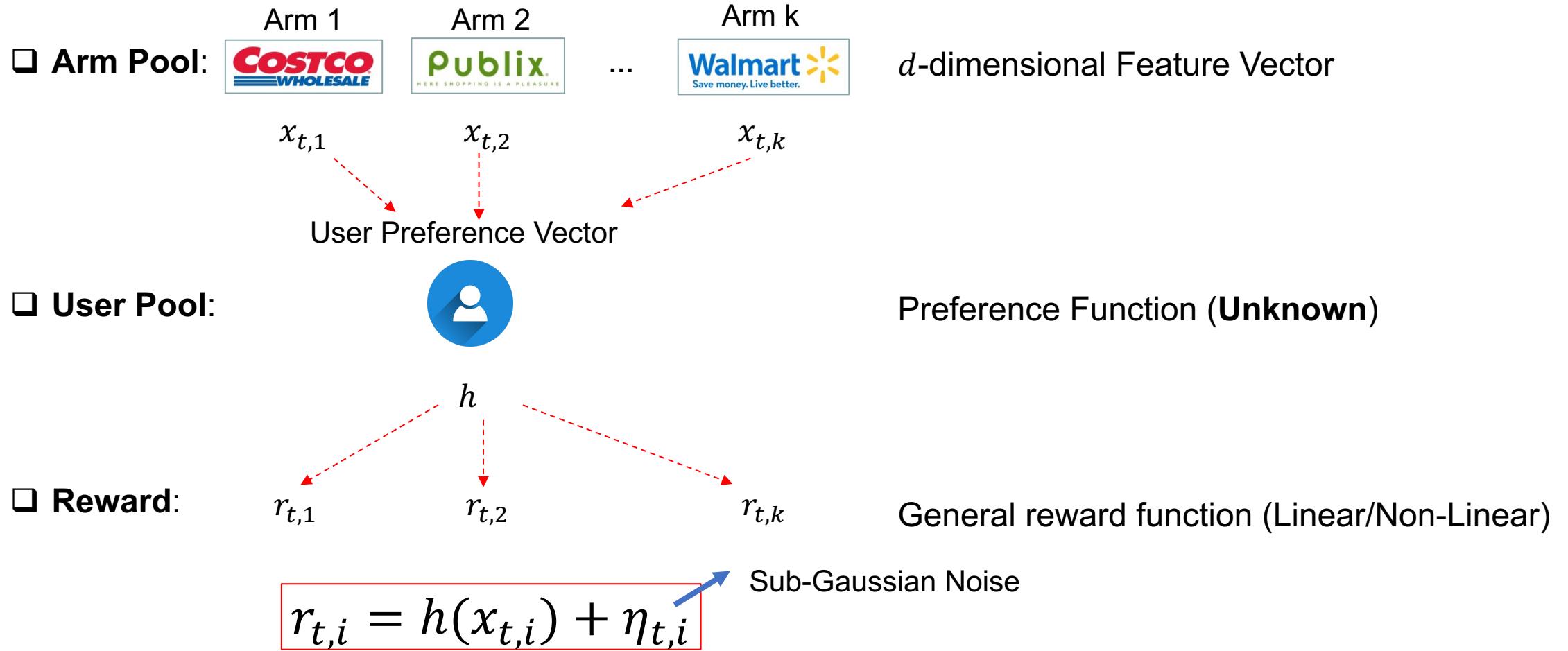
The Number of Rounds

The Number of Items

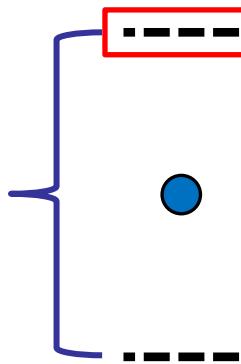
Dimensionality of Item Context Vector

# Neural Bandits: Problem Formulation

In round  $t$ :

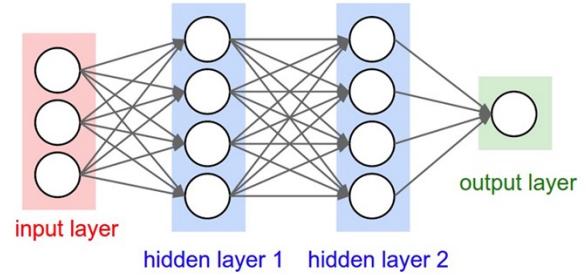


# Neural UCB: Method



$$U_{t,a}$$

$$f(\mathbf{x}; \boldsymbol{\theta}) = \sqrt{m} \mathbf{W}_L \sigma\left( \mathbf{W}_{L-1} \sigma\left( \dots \sigma(\mathbf{W}_1 \mathbf{x}) \right) \right)$$



Gradient of  $f$

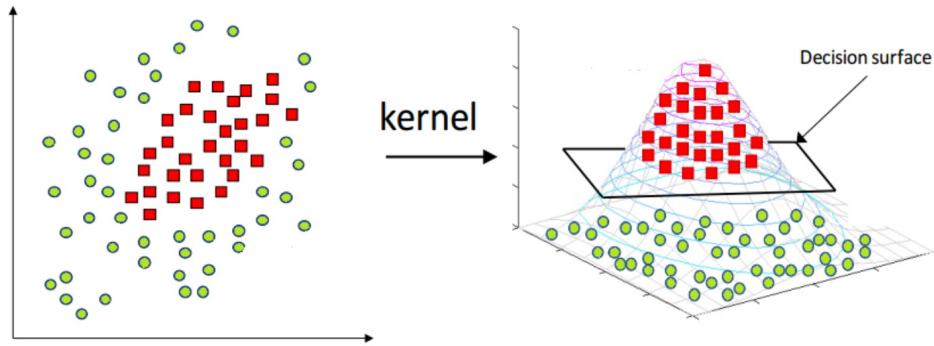
$$U_{t,a} = \underbrace{f(\mathbf{x}_{t,a}; \boldsymbol{\theta}_{t-1})}_{\text{mean}} + \gamma_{t-1} \sqrt{\underbrace{\mathbf{g}(\mathbf{x}_{t,a}; \boldsymbol{\theta}_{t-1})^\top \mathbf{Z}_{t-1}^{-1} \mathbf{g}(\mathbf{x}_{t,a}; \boldsymbol{\theta}_{t-1}) / m}_{\text{variance}}}$$

Compared with LinUCB (Li et al. 2010)

$$U_{t,a} = \underbrace{\langle \mathbf{x}_{t,a}, \boldsymbol{\theta}_{t-1} \rangle}_{\text{mean}} + \gamma_{t-1} \sqrt{\underbrace{\mathbf{x}_{t,a}^\top \mathbf{Z}_{t-1}^{-1} \mathbf{x}_{t,a}}_{\text{variance}}}$$

# Neural Tangent Kernel

- A **sufficiently wide neural network** behaves like a **linearized model** governed by the derivative of network with respect to its parameters (**Gradient**).

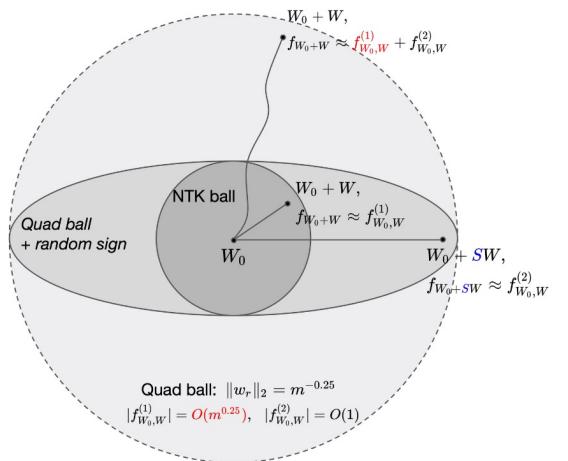


<https://www.geeksforgeeks.org/major-kernel-functions-in-support-vector-machine-svm/>

## Neural Tangent Kernel

$$\Theta(x, x'; \theta) = \nabla_\theta f(x; \theta) \cdot \nabla_\theta f(x'; \theta).$$

- With **near-infinite width**, Neural network behave like a **kernel predictor** with Neural Tangent Kernel (NTK)



1. Jacot, Arthur, Franck Gabriel, and Clément Hongler. "Neural tangent kernel: Convergence and generalization in neural networks." NeurIPS 2018.
2. Allen-Zhu, Zeyuan, Yuanzhi Li, and Zhao Song. "A convergence theory for deep learning via over-parameterization." ICML 2019.

# Neural UCB: Workflow

- In each round, a user is serving

```

for  $t = 1, \dots, T$  do K arms (Items)
    Observe  $\{\mathbf{x}_{t,a}\}_{a=1}^K$  Exploration
    for  $a = 1, \dots, K$  do Exploitation
        Compute  $U_{t,a} = f(\mathbf{x}_{t,a}; \boldsymbol{\theta}_{t-1}) + \gamma_{t-1} \sqrt{\mathbf{g}(\mathbf{x}_{t,a}; \boldsymbol{\theta}_{t-1})^\top \mathbf{Z}_{t-1}^{-1} \mathbf{g}(\mathbf{x}_{t,a}; \boldsymbol{\theta}_{t-1})/m}$ 
        Let  $a_t = \text{argmax}_{a \in [K]} U_{t,a}$ 
    end for
    Play  $a_t$  and observe reward  $r_{t,a_t}$  Similar to Linear Regression
    Compute  $\mathbf{Z}_t = \mathbf{Z}_{t-1} + \mathbf{g}(\mathbf{x}_{t,a_t}; \boldsymbol{\theta}_{t-1})\mathbf{g}(\mathbf{x}_{t,a_t}; \boldsymbol{\theta}_{t-1})^\top/m$ 
    Let  $\boldsymbol{\theta}_t = \text{TrainNN}(\lambda, \eta, J, m, \{\mathbf{x}_{i,a_i}\}_{i=1}^t, \{r_{i,a_i}\}_{i=1}^t, \boldsymbol{\theta}_0)$  Train Neural Networks
    Compute Confidence Radius
        
$$\gamma_t = \sqrt{1 + C_1 m^{-1/6} \sqrt{\log m} L^4 t^{7/6} \lambda^{-7/6}} \cdot \left( \nu \sqrt{\log \frac{\det \mathbf{Z}_t}{\det \lambda \mathbf{I}}} + C_2 m^{-1/6} \sqrt{\log m} L^4 t^{5/3} \lambda^{-1/6} - 2 \log \delta + \sqrt{\lambda} S \right)$$

        
$$+ (\lambda + C_3 t L) \left[ (1 - \eta m \lambda)^{J/2} \sqrt{t/\lambda} + m^{-1/6} \sqrt{\log m} L^{7/2} t^{5/3} \lambda^{-5/3} (1 + \sqrt{t/\lambda}) \right].$$

end for
Neural Function Approximation Error

```

# Neural UCB: Regret Analysis

- Definition of **NTK Matrix** on all observed contexts of T rounds.

$$\begin{aligned}\tilde{\mathbf{H}}_{i,j}^{(1)} &= \Sigma_{i,j}^{(1)} = \langle \mathbf{x}^i, \mathbf{x}^j \rangle, & \mathbf{A}_{i,j}^{(l)} &= \begin{pmatrix} \Sigma_{i,i}^{(l)} & \Sigma_{i,j}^{(l)} \\ \Sigma_{i,j}^{(l)} & \Sigma_{j,j}^{(l)} \end{pmatrix}, \\ \Sigma_{i,j}^{(l+1)} &= 2\mathbb{E}_{(u,v) \sim N(\mathbf{0}, \mathbf{A}_{i,j}^{(l)})} [\sigma(u)\sigma(v)], \\ \tilde{\mathbf{H}}_{i,j}^{(l+1)} &= 2\tilde{\mathbf{H}}_{i,j}^{(l)}\mathbb{E}_{(u,v) \sim N(\mathbf{0}, \mathbf{A}_{i,j}^{(l)})} [\sigma'(u)\sigma'(v)] + \Sigma_{i,j}^{(l+1)}.\end{aligned}$$

Then,  $\mathbf{H} = (\tilde{\mathbf{H}}^{(L)} + \Sigma^{(L)})/2$  is called the *neural tangent kernel (NTK)* matrix on the context set.

**Assumption:**  $\mathbf{H} \succeq \lambda_0 \mathbf{I}$ .

- Satisfied if no two observed arm contexts are parallel.

- Analyze dynamics of gradient and NTK regression.

**Lemma:** When neural network is wide enough,

$$\begin{aligned}h(\mathbf{x}^i) &= \langle \mathbf{g}(\mathbf{x}^i; \boldsymbol{\theta}_0), \boldsymbol{\theta}^* - \boldsymbol{\theta}_0 \rangle, & \text{Linear function w.r.t. Gradient} \\ \sqrt{m} \|\boldsymbol{\theta}^* - \boldsymbol{\theta}_0\|_2 &\leq \sqrt{2\mathbf{h}^\top \mathbf{H}^{-1} \mathbf{h}}, & (5.1)\end{aligned}$$

for all  $i \in [TK]$ .

# Neural UCB: Regret Analysis

**Assumption:**  $\mathbf{H} \succeq \lambda_0 \mathbf{I}$ .

$$\sqrt{m} \|\boldsymbol{\theta}^* - \boldsymbol{\theta}_0\|_2 \leq \sqrt{2 \mathbf{h}^\top \mathbf{H}^{-1} \mathbf{h}},$$

$$S = \sqrt{2 \mathbf{h}^\top \mathbf{H}^{-1} \mathbf{h}}$$

Satisfied if *no* two contexts in  $\{\mathbf{x}^i\}_{i=1}^{TK}$  are parallel.

$$h(\mathbf{x}^i) = \langle \mathbf{g}(\mathbf{x}^i; \boldsymbol{\theta}_0), \boldsymbol{\theta}^* - \boldsymbol{\theta}_0 \rangle,$$

$$\tilde{d} = \frac{\log \det(\mathbf{I} + \mathbf{H}/\lambda)}{\log(1 + TK/\lambda)}$$

## Theorem

LinUCB:

$\tilde{O}(d\sqrt{T})$

Let  $\mathbf{h} = [h(\mathbf{x}^i)]_{i=1}^{TK} \in \mathbb{R}^{TK}$ . Set  $J = \tilde{\Theta}(TL/\lambda)$ ,  $\eta = \Theta((mTL + m\lambda)^{-1})$  and  $S = 2\sqrt{\mathbf{h}^\top \mathbf{H}^{-1} \mathbf{h}}$ . Under the overparameterized setting ( $m \gg 1$ ), with probability at least  $1 - \delta$ ,

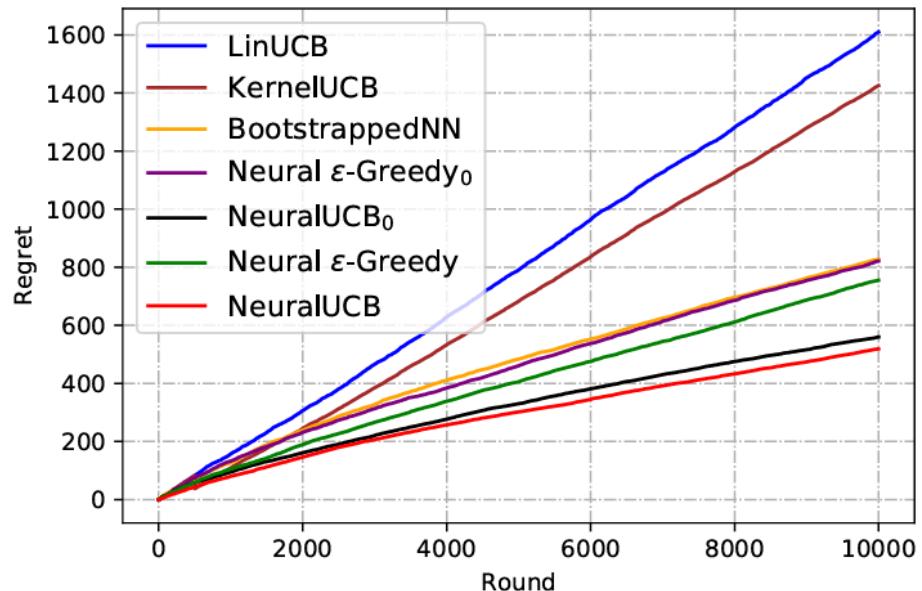
$$R_T = \tilde{O}\left(\sqrt{\tilde{d}T} \sqrt{\max\{\tilde{d}, S^2\}}\right).$$

Upper Bound of Neural Parameters

Effective dimension in NTK Space

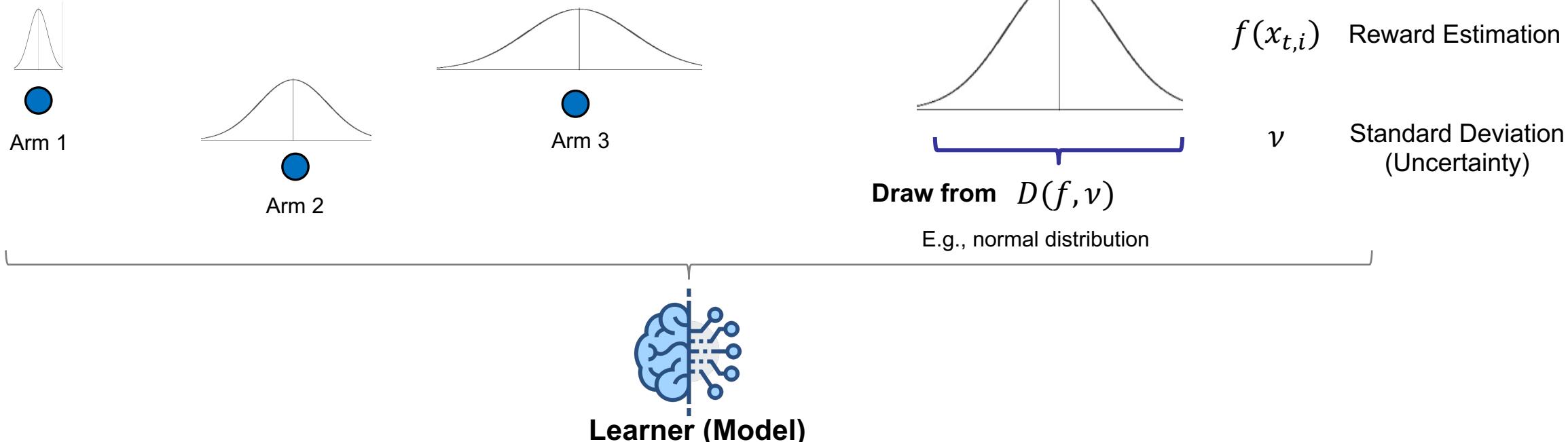
# Neural UCB: Empirical Evaluation

- NeuralUCB uses neural networks for exploitation, and gradient to explore.
- NeuralUCB achieve  $\tilde{O}(\sqrt{T})$  regret upper bound, similar to LinearUCB.
- NeuralUCB generally outperforms linear contextual bandits.



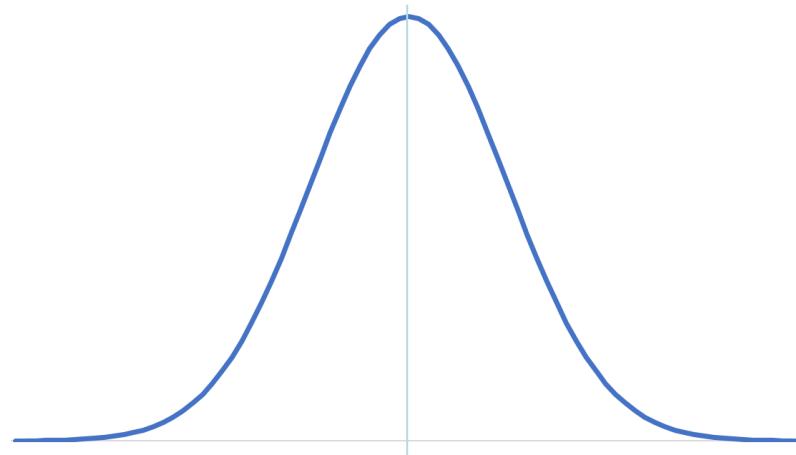
# Thompson Sampling

- Popular existing exploration strategies.
  - **$\epsilon$ -greedy:** With probability  $1 - \epsilon$ , greedily choose one arm according to history;  
Otherwise, choose an arm randomly.
  - **Upper Confidence Bound.**
  - **Thompson Sampling:**

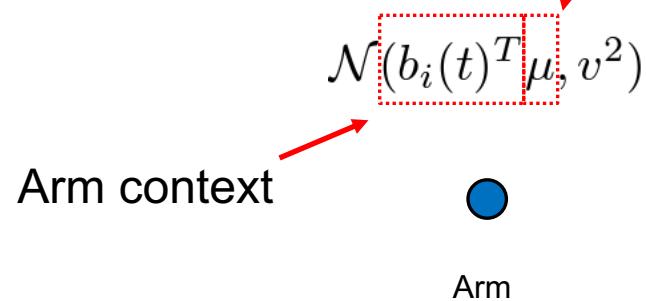


# Linear Thompson Sampling

Reward Distribution (Gaussian Prior)



User Preference Parameter (Unknown)



**for all**  $t = 1, 2, \dots$ , **do**

Sample  $\tilde{\mu}(t)$  from distribution  $\mathcal{N}(\hat{\mu}, v^2 B^{-1})$ .

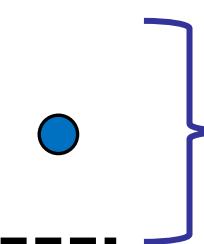
Play arm  $a(t) := \arg \max_i b_i(t)^T \tilde{\mu}(t)$ , and observe reward  $r_t$ .

Update  $B = B + b_{a(t)}(t)b_{a(t)}(t)^T$ ,  $f = f + b_{a(t)}(t)r_t$ ,  $\hat{\mu} = B^{-1}f$ .

**end for**

Estimated User Preference

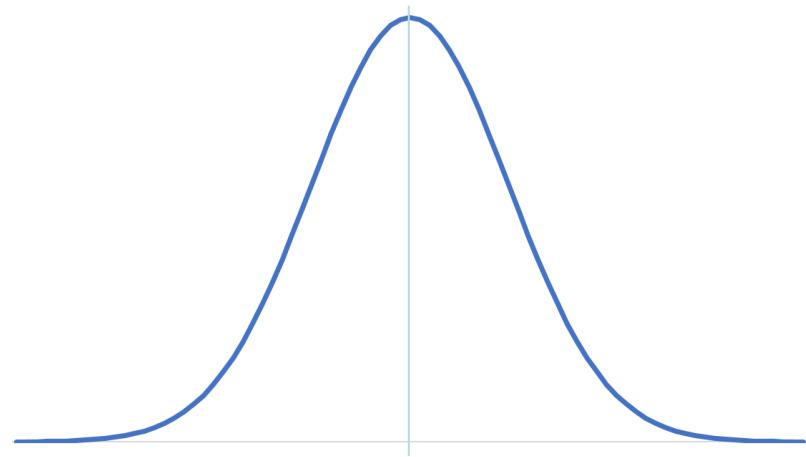
----



Confidence Interval

# Neural Thompson Sampling

Reward Distribution (Gaussian Prior)



$$N(h(x_{t,k}), \nu^2)$$

Expected Reward and Variance

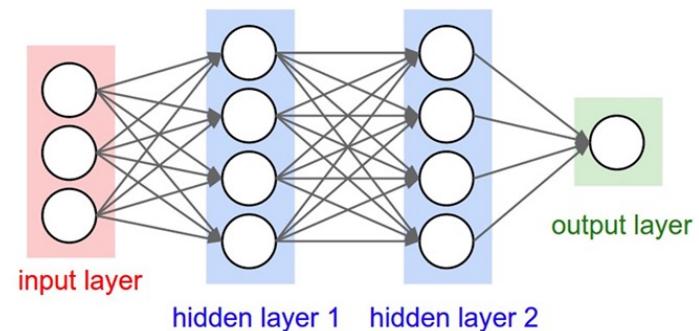


Arm

Estimated Distribution:

$$\mathcal{N}(f(\mathbf{x}_{t,k}; \boldsymbol{\theta}_{t-1}), \nu^2 \sigma_{t,k}^2)$$

$$f(\mathbf{x}; \boldsymbol{\theta}) = \sqrt{m} \mathbf{W}_L \sigma \left( \mathbf{W}_{L-1} \sigma \left( \cdots \sigma(\mathbf{W}_1 \mathbf{x}) \right) \right)$$





# Neural Thompson Sampling

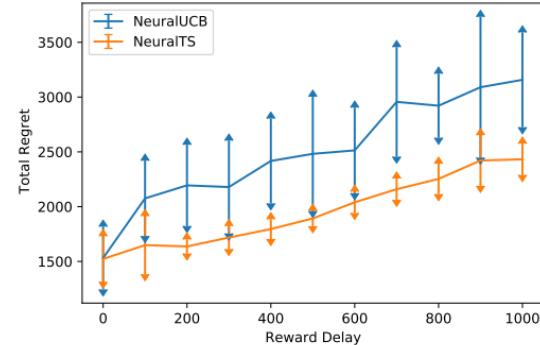
- In each round, a user is serving

```
for  $t = 1, \dots, T$  do K arms
  for  $k = 1, \dots, K$  do
     $\sigma_{t,k}^2 = \lambda \mathbf{g}^\top(\mathbf{x}_{t,k}; \boldsymbol{\theta}_{t-1}) \mathbf{U}_{t-1}^{-1} \mathbf{g}(\mathbf{x}_{t,k}; \boldsymbol{\theta}_{t-1})/m$  Similar to Linear Regression
    Sample estimated reward  $\tilde{r}_{t,k} \sim \mathcal{N}(f(\mathbf{x}_{t,k}; \boldsymbol{\theta}_{t-1}), \nu^2 \sigma_{t,k}^2)$ 
  end for
  Pull arm  $a_t$  and receive reward  $r_{t,a_t}$ , where  $a_t = \text{argmax}_a \tilde{r}_{t,a}$ 
  Set  $\boldsymbol{\theta}_t$  to be the output of gradient descent for solving (2.3)
   $\mathbf{U}_t = \mathbf{U}_{t-1} + \mathbf{g}(\mathbf{x}_{t,a_t}; \boldsymbol{\theta}_t) \mathbf{g}(\mathbf{x}_{t,a_t}; \boldsymbol{\theta}_t)^\top/m$ 
end for
```

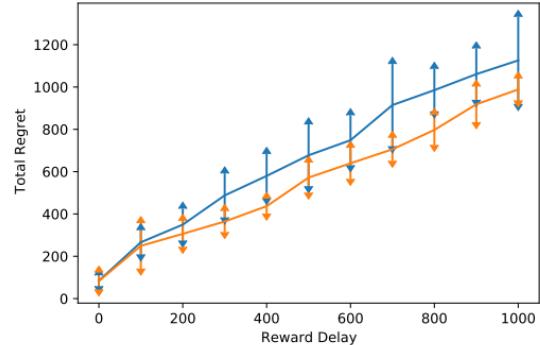
Compared to NeuralUCB:

$$U_{t,a} = f(\mathbf{x}_{t,a}; \boldsymbol{\theta}_{t-1}) + \gamma_{t-1} \sqrt{\mathbf{g}(\mathbf{x}_{t,a}; \boldsymbol{\theta}_{t-1})^\top \mathbf{Z}_{t-1}^{-1} \mathbf{g}(\mathbf{x}_{t,a}; \boldsymbol{\theta}_{t-1})/m}$$

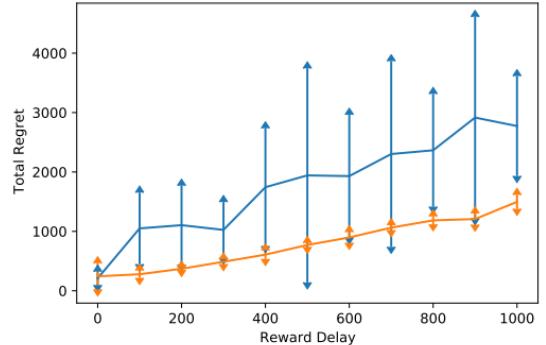
# Neural Thompson Sampling



(a) MNIST



(b) Mushroom



(c) Shuttle

- NeuralTS and NeuralUCB have **similar performance** when network is trained every iteration.
- NeuralTS is more robust than NeuralUCB when network is trained **in batch**.
- NeuralTS introduces more **robustness** in exploration.



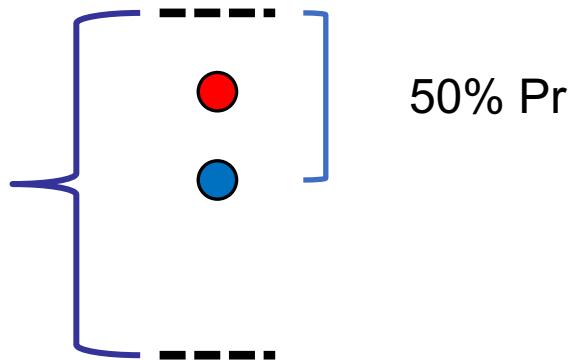
# EE-Net: Background

➤ UCB-based and TS-based exploration highly rely on large-deviation-based **statistical confidence interval**.

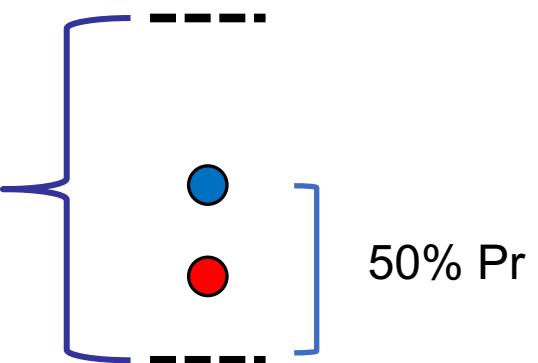
□ Ideal scenario:

● Expected reward

● Estimated Reward



And



Symmetric

# EE-Net: Motivation



- UCB-based and TS-based exploration highly rely on large-deviation-based **statistical confidence bound**.

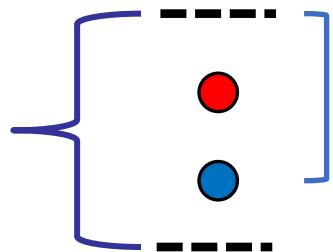
□ In practice, may be:



Expected reward

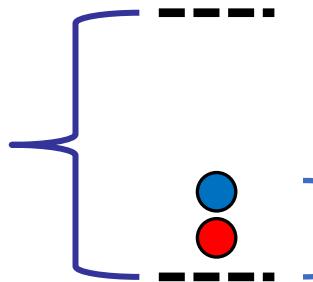


Estimated Reward



80% Pr

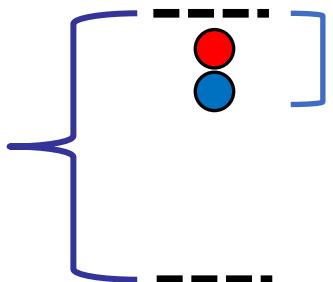
And



20% Pr

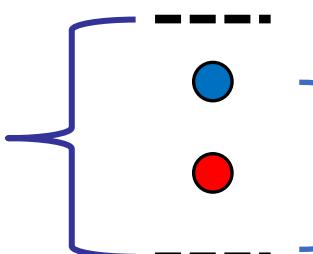
---

**Asymmetric**



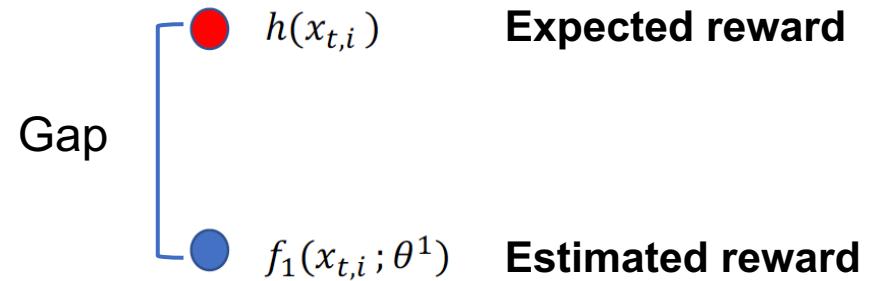
10% Pr

And



90% Pr

- **Why making exploration?**
  - Because we cannot make accurate prediction on a subset of data.
- **Goal of exploration:** Fill the **gap** between **expected reward** and **estimated reward**.



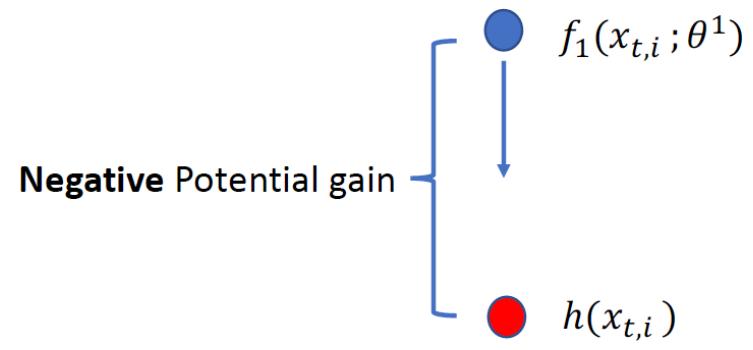
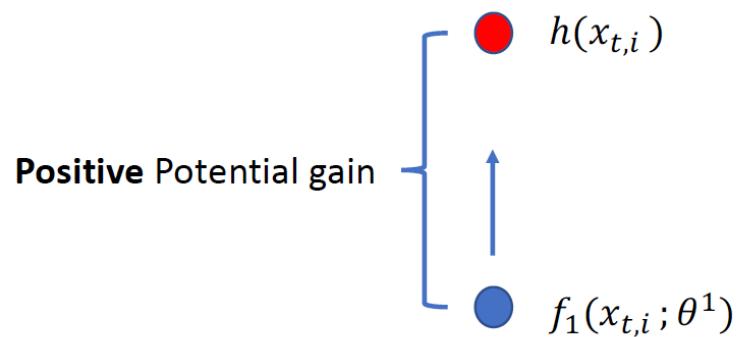
# EE-Net: Exploration Direction



➤ Two types of exploration: “Upward” exploration and “downward” Exploration.

●  $h(x_{t,i})$  Expected reward

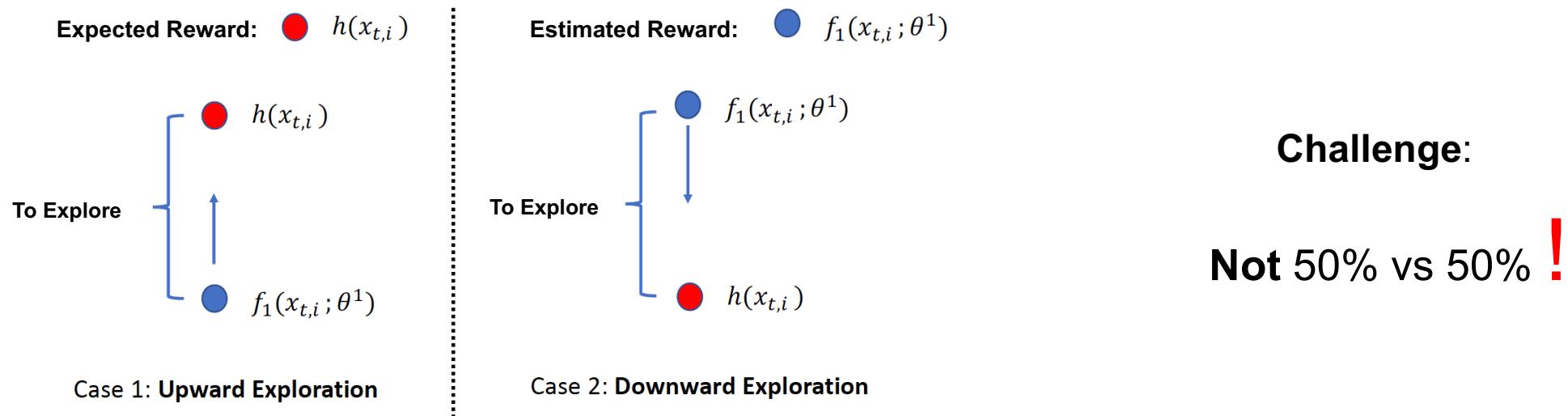
●  $f_1(x_{t,i}; \theta^1)$  Estimation



Underestimation

Overestimation

# Adapt to Exploration Direction is Challenging



Datasets	Upward Exploration	Downward Exploration
Mnist	<b>76.3%</b>	23.7%
Disin	29.1%	<b>70.9%</b>
MovieLens	<b>58.6%</b>	41.4%
Yelp	<b>55.3%</b>	44.7%



**Pessimistic Model (Human)**



**Optimistic Model (Human)**

# EE-Net: Solution



- Motivation: Can we have an **adaptive** exploration strategy for both “upward” and “downward” exploration?
- **Proposed solution:** We propose to use **another neural network to learn** the gap between expected reward and estimated reward (**potential gain**) **incorporating exploration direction.**

# EE-Net: Motivation and Solution



- Motivation: Can we have an **adaptive** exploration strategy for both “upward” and “downward” exploration?
- **Proposed solution:** We propose to use **another neural network to learn** the gap between expected reward and estimated reward (**potential gain**) **incorporating exploration direction.**
- **Exploitation neural network**  $f_1$  to estimate reward:
  - Given an arm  $x_{t,i}$ ,
  - $f_1(x_{t,i}; \theta^1) = \mathbf{W}_L \sigma(\mathbf{W}_{L-1} \sigma(\dots \sigma(\mathbf{W}_1 \cdot)))$
  - $f_1(x_{t,i}; \theta^1)$  is to **estimate expected reward** represented by some unknown function  $h(x_{t,i})$ .
  - In round  $t$ ,  $\theta^1$  is **trained on data of past  $t - 1$  rounds**, using **gradient descent**.

# EE-Net: Exploration Neural Networks



- **Exploration neural network  $f_2$**  (novel component) to estimate potential gain:
  - Given an arm  $x_{t,i}$  and its estimation  $f_1(x_{t,i}; \theta^1)$ , **expected potential gain** is defined as:

$$h(x_{t,i}) - f_1(x_{t,i}; \theta^1),$$

where  $h(x_{t,i})$  is the expected reward.

- Thus, given the received reward  $r_{t,i}$ , **potential gain** is defined as:

$$r_{t,i} - f_1(x_{t,i}; \theta^1),$$

where  $\mathbb{E}[r_{t,i}] = h(x_{t,i})$ .

- Potential gain has a good property: **Indicating exploration direction**.

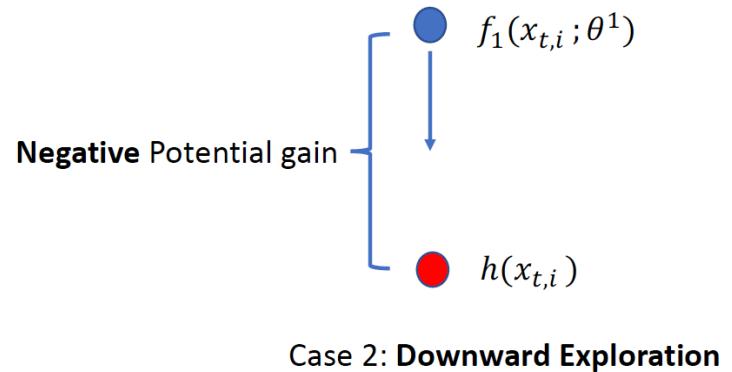
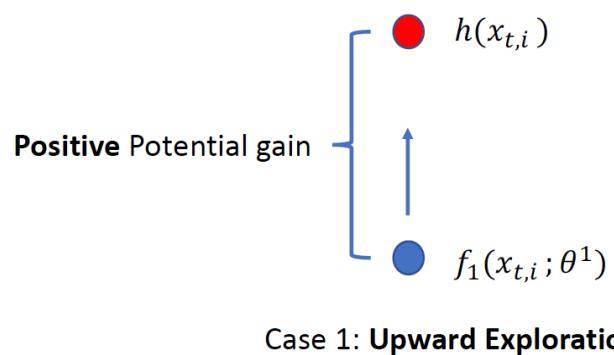
# EE-Net: Exploration Neural Networks



- **Exploration neural network  $f_2$**  (novel component) to estimate potential gain:
  - Potential gain has good property: **indicating exploration direction.**

$$h(x_{t,i}) - f_1(x_{t,i}; \theta^1) > 0$$

$$h(x_{t,i}) - f_1(x_{t,i}; \theta^1) < 0$$



# EE-Net: Exploration Neural Networks



- **Exploration neural network  $f_2$** (novel component) to estimate potential gain:
  - Label of  $f_2$ :  $r_{t,i} - f_1(x_{t,i}; \cdot)$

$$f_2(x_{t,i}; \theta^2) = \mathbf{W}_L \sigma(\mathbf{W}_{L-1} \sigma(\dots \sigma(\mathbf{W}_1 \cdot)))$$

- What is input of  $f_2$ ?

# EE-Net: Exploration Neural Networks in Bandits

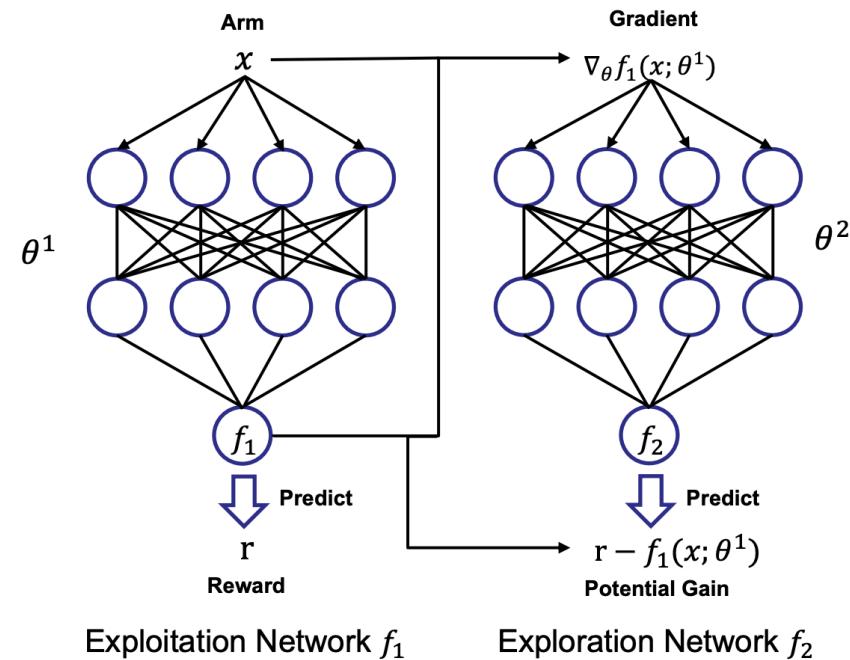


- Exploration neural network  $f_2$  (novel component) to estimate potential gain:
  - Input of  $f_2$ : Gradient of  $f_1$  with respect to  $\theta^1$ :

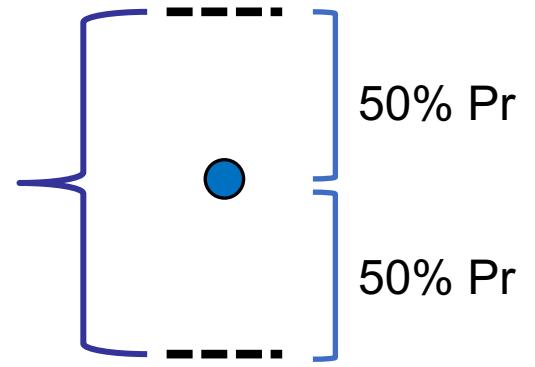
$$\nabla_{\theta^1} f(x_{t,i}; \theta^1)$$

- Rational:
  - Incorporate both feature of input and discriminative information of  $f_1$ .
  - Based on [2,3],  $f_1$  has the following confidence bound:
$$|h(\mathbf{x}_{t,i}) - f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}_{t-1}^1)| \leq \Psi(\nabla_{\boldsymbol{\theta}_{t-1}^1} f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}_{t-1}^1)),$$
Here, instead of choosing a fixed form  $\Psi$ , we use  $f_2$  to learn it.
  - In this way,  $\theta^2$  is trained on  $\{\nabla_{\theta^1} f(x_{t,i}; \theta_{\tau-1}^1)\}_{\tau=1}^t$  to store historical information.

# EE-Net: Overview

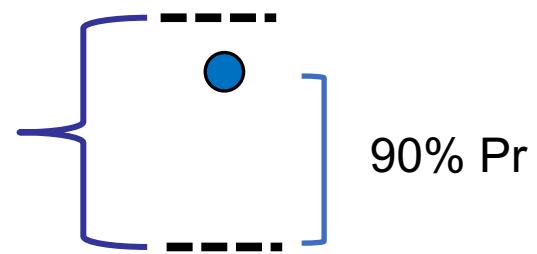


**Statistical**  
Confidence Interval



Symmetric and Fixed

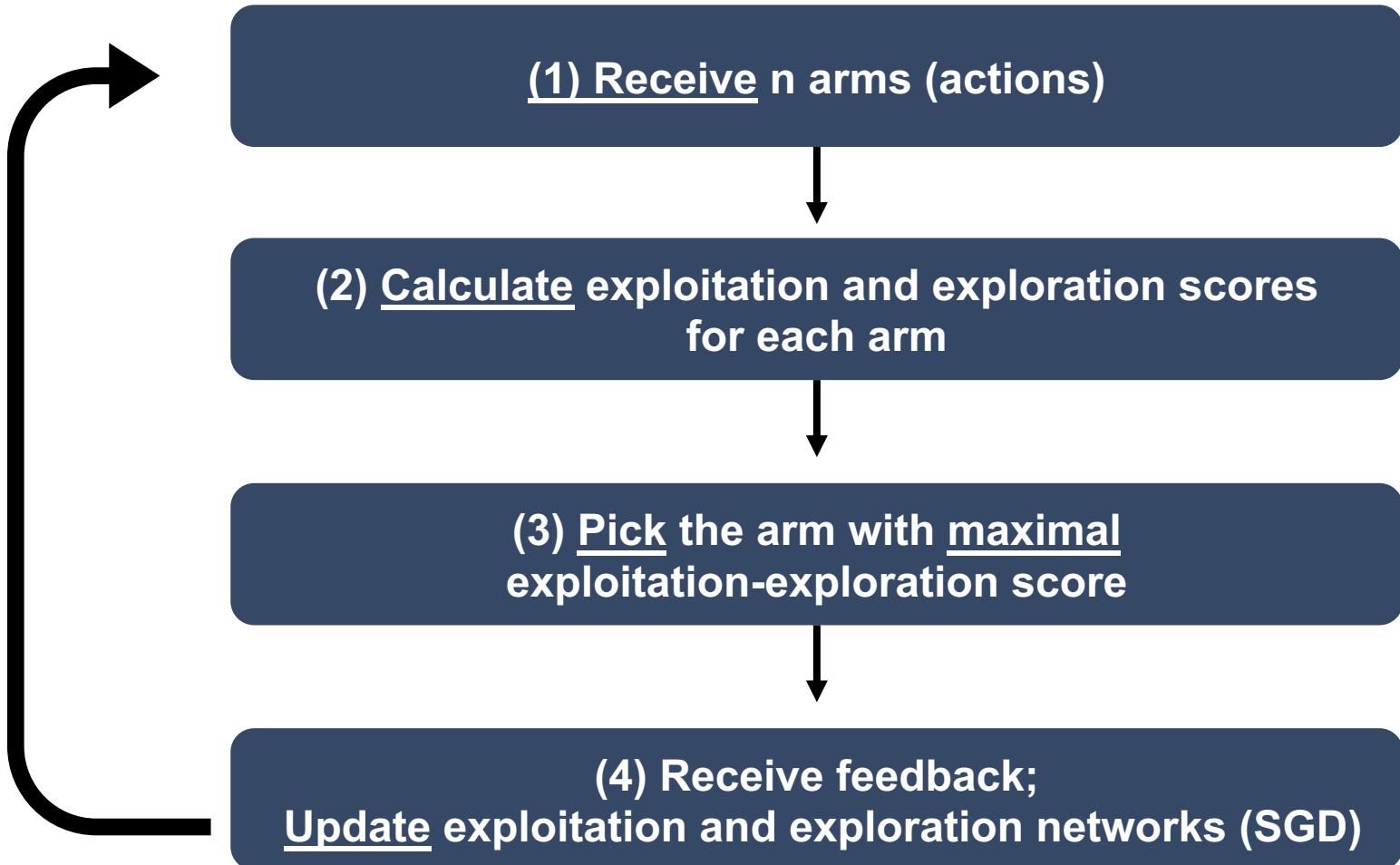
Confidence Interval  
**learned by neural network**  
**(Our approach)**



Asymmetric and Adaptive

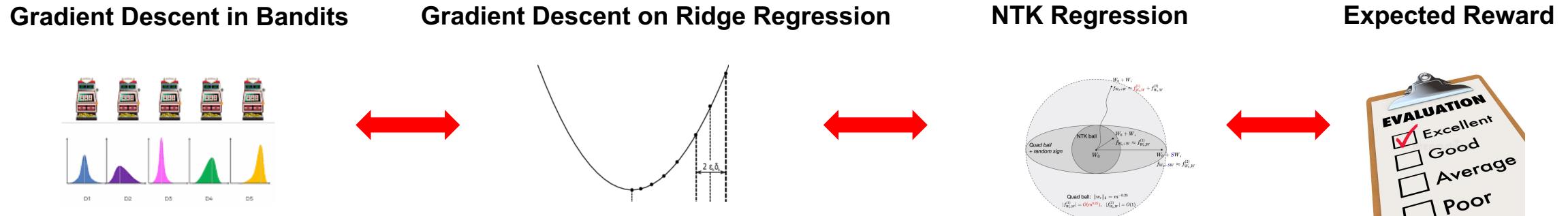
Methods	"Upward" Exploration	"Downward" Exploration
$\epsilon$ -Greedy	✗	✗
NeuralUCB	✓	✗
NeuralTS	Randomly	Randomly
EE-Net	✓	✓

# EE-Net: Workflow

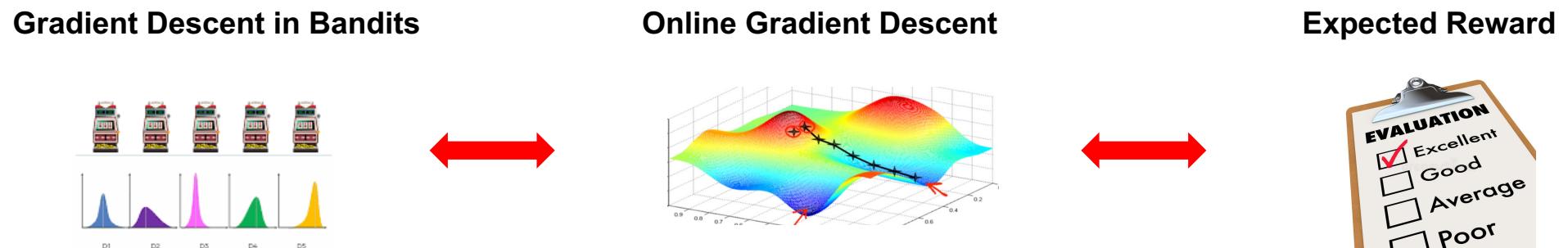


# EE-Net: Theoretical Analysis

➤ Proof Workflow of NeuralUCB [1] and NeuralTS [2]:



➤ Proof Workflow of EE-Net [3,4]:



- Zhou, Dongruo, Lihong Li, and Quanquan Gu. "Neural contextual bandits with ucb-based exploration." ICML, 2020.
- Zhang, Weitong, et al. "Neural thompson sampling." ICLR 2021.
- Ban, Yikun, et al. "EE-Net: Exploitation-Exploration Neural Networks in Contextual Bandits." ICLR 2022.
- Ban, Yikun, et al. "Neural Exploitation and Exploration of Contextual Bandits." JMLR 2024.

# EE-Net: Theoretical Analysis

**Assumption 1:** For any  $t \in [T], i \in [n], \|\mathbf{x}_{t,i}\|_2 = 1$ , and  $r_{t,i} \in [0, 1]$ .

- Assumption 1 is standard and mild in analysis of over-parameterized neural networks.
- **No assumption** on distribution of arm contexts.
- Then, we have the following **average error bound for exploration network  $f_2$** :

**Lemma1.** For any  $\delta \in (0, 1), R > 0$ , suppose  $m$  satisfies the conditions in Theorem 6. In round  $t \in [T]$ , let

$$\hat{i} = \arg \max_{i \in [k]} \left( f_1(\mathbf{x}_{t,\hat{i}}; \boldsymbol{\theta}_{t-1}^1) / \sqrt{m} + f_2(\phi(\mathbf{x}_{t,\hat{i}}); \boldsymbol{\theta}_{t-1}^2) / \sqrt{m} \right).$$

Then, with probability at least  $1 - \delta$ , we have

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{r_{t,\hat{i}}} \left[ \min \left\{ \left| f_2(\phi(\mathbf{x}_{t,\hat{i}}); \boldsymbol{\theta}_{t-1}^2) / \sqrt{m} - (r_{t,\hat{i}} - f_1(\mathbf{x}_{t,\hat{i}}; \boldsymbol{\theta}_{t-1}^1) / \sqrt{m}) \right|, 1 \right\} \right] \\ \leq \underbrace{\sqrt{\frac{\Psi(\boldsymbol{\theta}_0^2, R)}{T}}}_{(1)} + \underbrace{\mathcal{O} \left( \frac{3LR}{\sqrt{2T}} \right)}_{(2)} + \underbrace{\sqrt{\frac{2 \log(\mathcal{O}(1)/\delta)}{T}}}_{(3)}. \end{aligned} \tag{5.3}$$

# EE-Net: Theoretical Analysis

**Lemma 1.** For any  $\delta \in (0, 1)$ ,  $R > 0$ , suppose  $m$  satisfies the conditions in Theorem 6. In round  $t \in [T]$ , let

$$\hat{i} = \arg \max_{i \in [k]} \left( f_1(\mathbf{x}_{t,i}; \boldsymbol{\theta}_{t-1}^1) / \sqrt{m} + f_2(\phi(\mathbf{x}_{t,i}); \boldsymbol{\theta}_{t-1}^2) / \sqrt{m} \right).$$

Then, with probability at least  $1 - \delta$ , we have

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{r_{t,\hat{i}}} \left[ \min \left\{ \left| f_2(\phi(\mathbf{x}_{t,\hat{i}}); \boldsymbol{\theta}_{t-1}^2) / \sqrt{m} - (r_{t,\hat{i}} - f_1(\mathbf{x}_{t,\hat{i}}; \boldsymbol{\theta}_{t-1}^1) / \sqrt{m}) \right|, 1 \right\} \right] \\ \leq \underbrace{\sqrt{\frac{\Psi(\boldsymbol{\theta}_0^2, R)}{T}}}_{(1)} + \underbrace{\mathcal{O}\left(\frac{3LR}{\sqrt{2T}}\right)}_{(2)} + \underbrace{\sqrt{\frac{2 \log(\mathcal{O}(1)/\delta)}{T}}}_{(3)}. \end{aligned} \quad (5.3)$$

➤ (1) Complexity term  $\Psi$ : **Infimum of regression error** caused by function class  $B(\boldsymbol{\theta}^2, R)$ :

$$\mathcal{B}(\boldsymbol{\theta}_0^2, R) = \{ \tilde{\boldsymbol{\theta}}^2 \in \mathbb{R}^p : \| \tilde{\boldsymbol{\theta}}^2 - \boldsymbol{\theta}_0^2 \|_2 \leq \mathcal{O}\left(\frac{R}{\sqrt{m}}\right) \}. \quad \Psi(\boldsymbol{\theta}_0^2, R) = \inf_{\tilde{\boldsymbol{\theta}}^2 \in \mathcal{B}(\boldsymbol{\theta}_0^2, R)} \sum_{t=1}^T (f^2(\mathbf{x}_{t,\hat{i}}; \tilde{\boldsymbol{\theta}}^2) - r_{t,\hat{i}}^2)^2$$

➤ (2) **Price** of picking function class  $B(\boldsymbol{\theta}^2, R)$  controlled by radius  $R$ .

➤ (3) **Confidence bound** for predictions of  $f_2$ .

# EE-Net: Regret Upper Bound

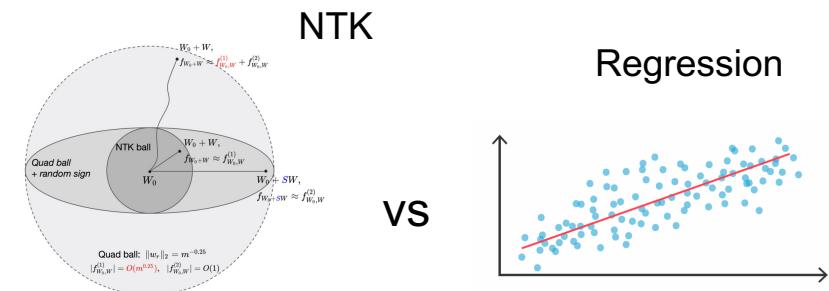
- Then, we have following regret upper bound  $\tilde{O}(\sqrt{T})$  for EE-Net:

**Theorem 1.** Let  $f_1, f_2$  follow the setting of  $f$  (Eq. (5.1)) with the same width  $m$  and depth  $L$ . Suppose  $m \geq \Omega(\text{poly}(T, L, R, \log(1/\delta)))$ ,  $\eta_1 = \eta_2 = \frac{\sqrt{\nu}R}{m\sqrt{T}}$  and  $\Psi(\boldsymbol{\theta}_0^2, R) \& \Psi^*(\boldsymbol{\theta}_0^2, R) \leq \Psi$ . Then, for any  $\delta \in (0, 1)$ ,  $R > 0$ , with probability at least  $1 - \delta$  over the initialization, there exists a constant  $\nu$ , such that the pseudo regret of Algorithm 1 in  $T$  rounds satisfies

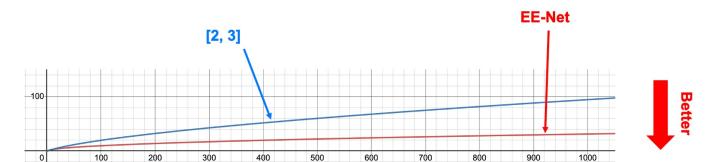
$$\mathbf{R}_T \leq \sqrt{T} \cdot \mathcal{O}\left(RL + \sqrt{\Psi} + 2\sqrt{2\log(\mathcal{O}(1)/\delta)}\right) + \mathcal{O}(1) \quad (5.2)$$

- Compared to existing works NeuralUCB [3] and NeuralTS [4]:

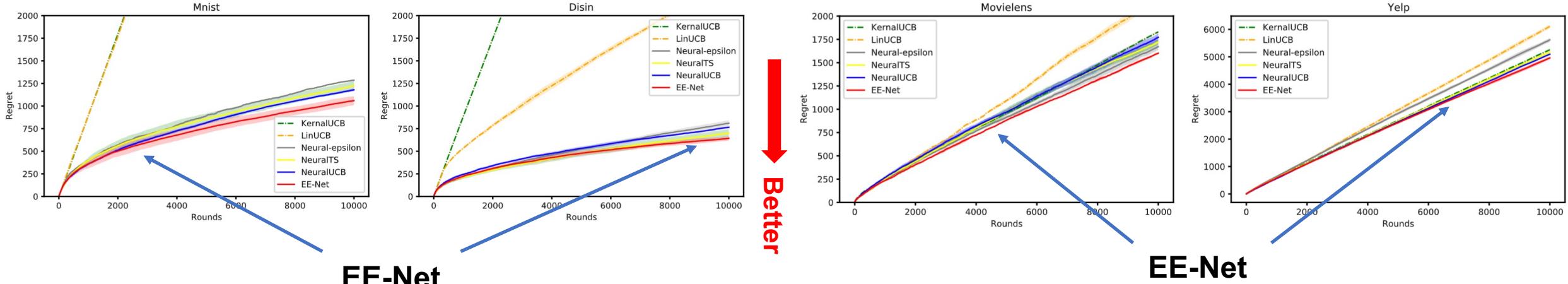
$$\mathbf{R}_T \leq \mathcal{O}\left(\sqrt{\tilde{d}T \log T + S^2}\right) \cdot \mathcal{O}\left(\sqrt{\tilde{d} \log T}\right), \text{ and } \tilde{d} = \frac{\log \det(\mathbf{I} + \mathbf{H}/\lambda)}{\log(1 + Tn/\lambda)}$$



- [Better Interpretability]: Have the similar complexity term but  $\Psi$  easier to interpret.
- [Contexts]: Allow arm contexts to be repeatedly observed.
- [Tighter Bound]: EE-Net improves by a multiplicative factor  $\log T$ .



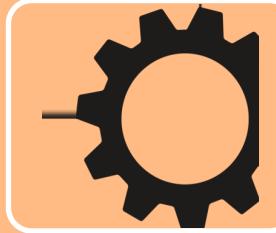
# EE-Net: Empirical Experiments



## ➤ Setup:

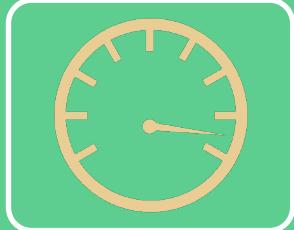
- Classification and recommendation dataset.
- 5 state-of-the-art baselines including  $\epsilon$ -greedy, UCB, TS exploration strategy.
- All methods have the same exploitation network  $f_1$ .

➤ **EE-Net achieves substantial improvements, because all improvements purely come from exploration!**



## Fundamental Exploration

- Upper Confidence Bound
- Thompson Sampling
- Exploration Network



## Efficient Exploration

- Neural Linear UCB
- Neural Network with Perturbed Reward
- Inverse Weight Gap Strategy



# Neural Linear UCB

- In each round, a user is serving

```

for  $t = 1, \dots, T$  do
    receive feature vectors  $\{\mathbf{x}_{t,1}, \dots, \mathbf{x}_{t,K}\}$  K arms
    choose arm  $a_t = \operatorname{argmax}_{k \in [K]} \theta_{t-1}^\top \phi(\mathbf{x}_{t,k}; \mathbf{w}_{t-1}) + \alpha_t \|\phi(\mathbf{x}_{t,k}; \mathbf{w}_{t-1})\|_{\mathbf{A}_{t-1}^{-1}}$ , and obtain
    reward  $\hat{r}_t$  Representation by Neural Network
    update  $\mathbf{A}_t$  and  $\mathbf{b}_t$  as follows:
         $\mathbf{A}_t = \mathbf{A}_{t-1} + \phi(\mathbf{x}_{t,a_t}; \mathbf{w}_{t-1}) \phi(\mathbf{x}_{t,a_t}; \mathbf{w}_{t-1})^\top$ ,  $\mathbf{b}_t = \mathbf{b}_{t-1} + \hat{r}_t \phi(\mathbf{x}_{t,a_t}; \mathbf{w}_{t-1})$ ,
        update  $\theta_t = \mathbf{A}_t^{-1} \mathbf{b}_t$  Exploitation Exploration
        if  $\operatorname{mod}(t, H) = 0$  then Estimated by Ridge Regression
             $\mathbf{w}_t \leftarrow$  output of Algorithm 2
             $q = q + 1$ 
        else
             $\mathbf{w}_t = \mathbf{w}_{t-1}$ 
        end if
    end for
Output  $\mathbf{w}_T$ 

```

Compared with LinUCB (Li et al. 2010)

$$U_{t,a} = \underbrace{\langle \mathbf{x}_{t,a}, \theta_{t-1} \rangle}_{\text{mean}} + \gamma_{t-1} \sqrt{\underbrace{\mathbf{x}_{t,a}^\top \mathbf{Z}_{t-1}^{-1} \mathbf{x}_{t,a}}_{\text{variance}}}$$

$$\phi(\mathbf{x}; \mathbf{w}) = \sqrt{m} \sigma(\mathbf{W}_L \sigma(\mathbf{W}_{L-1} \cdots \sigma(\mathbf{W}_1 \mathbf{x}) \cdots)).$$

- Update Neural Network Parameter:

Loss function:

$$\mathcal{L}_q(\mathbf{w}) = \sum_{i=1}^{qH} (\theta_i^\top \phi(\mathbf{x}_{i,a_i}; \mathbf{w}) - \hat{r}_i)^2.$$

- Gradient Descent.

# Neural Bandit With Perturbed Reward

- In each round, a user is serving

```

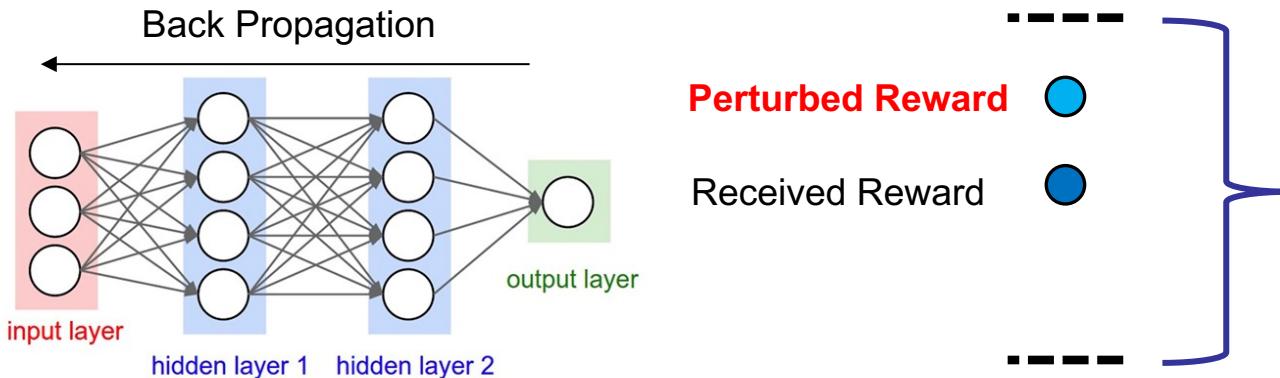
for  $t = 1, \dots, T$  do
    if  $t > K$  then Initialization: Pull each arm once
        Pull arm  $a_t$  and receive reward  $r_{t,a_t}$ , where  $a_t = \text{argmax}_{i \in [K]} f(\mathbf{x}_i, \boldsymbol{\theta}_{t-1})$ . Selection Criterion
        Generate  $\{\gamma_s^t\}_{s \in [t]} \sim \mathcal{N}(0, \nu^2)$ . Perturbed Reward
        Set  $\boldsymbol{\theta}_t$  by the output of gradient descent for solving Eq (3.2).
    else
        Pull arm  $a_k$ .
    end if
end for

```

Reward Perturbation (Noise)

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \sum_{s=1}^t (f(\mathbf{x}_{a_s}; \boldsymbol{\theta}) - (r_{s,a_s} + \gamma_s^t))^2 / 2 + m\lambda \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2^2 / 2$$

Implicit Exploration:



# Neural SquareCB: Inverse Gap Strategy

- In each round, a user is serving

Special Case:  $y = 1 - r$

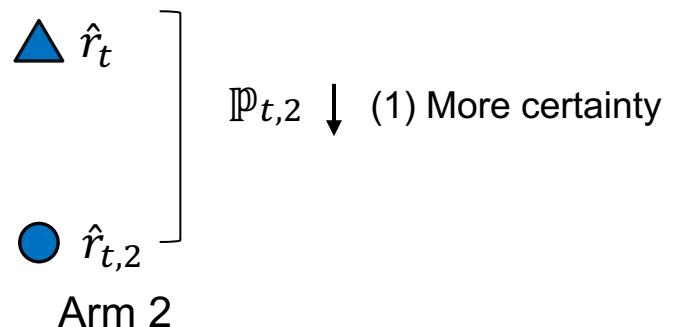
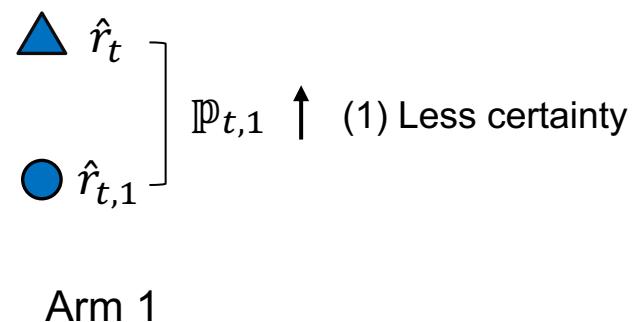
```

for  $t = 1, 2, \dots, T$  do           K arms
    Receive contexts  $\mathbf{x}_{t,1}, \dots, \mathbf{x}_{t,K}$ , and compute  $\hat{y}_{t,a} = \tilde{f}^{(S)}(\theta; \mathbf{x}_{t,a}, \varepsilon^{(1:S)})$ ,  $\forall a \in [K]$ 
    Let  $b = \arg \min_a \hat{y}_{t,a}$ ,  $p_{t,a} = \frac{1}{K + \gamma(\hat{y}_{t,b} - \hat{y}_{t,a})}$ , and  $p_{t,b} = 1 - \sum_{a \neq b} p_{t,a}$ 
    Sample arm  $a_t \sim p_t$  and observe output  $y_{t,a_t}$  →
    Update  $\theta_{t+1} = \prod_{B_{\rho, \rho_1}^{\text{Frob}}(\theta_0)} \left( \theta_t - \eta_t \nabla \mathcal{L}_{\text{Sq}}^{(S)}(y_{t,a_t}, \{\tilde{f}(\theta; \mathbf{x}_{t,a_t}, \varepsilon_s)\}_{s=1}^S) \right)$ .
end for

```

Equal to the arm with maximal reward Inverse Weight Gap to form distribution for Selection

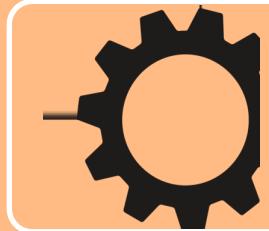
$$\hat{r}_t = \arg \max f(x_{t,i}; \theta_t) \quad \mathbb{P}_{t,i} \propto \frac{1}{\hat{r}_t - \hat{r}_{t,i}} \quad \text{Selection Probability}$$



# Neural SquareCB: Inverse Gap Strategy

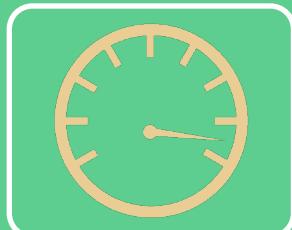
Algorithm	Regret	Remarks
Neural UCB [Zhou et al., 2020]	$\tilde{\mathcal{O}}(\tilde{d}\sqrt{T})$	Bound depends on $\tilde{d}$ and could be $\Omega(T)$ in worst case.
Neural TS Zhang et al. [2021]	$\tilde{\mathcal{O}}(\tilde{d}\sqrt{T})$	Bound depends on $\tilde{d}$ and could be $\Omega(T)$ in worst case.
EE-Net [Ban et al., 2022b]	$\tilde{\mathcal{O}}(\sqrt{T})$	Assumes that the contexts at every round are drawn i.i.d and needs to store all the previous networks.
NeuSquareCB (This work)	$\tilde{\mathcal{O}}(\sqrt{KT})$	No dependence on $\tilde{d}$ and holds even when the contexts are chosen adversarially.
NeuFastCB (This work)	$\tilde{\mathcal{O}}(\sqrt{L^*K} + K)$	No dependence on $\tilde{d}$ and holds even when the contexts are chosen adversarially. Further, this is the first data-dependent regret bound for neural bandits.

- Remove dependence of effective dimension.
- Minimize dependence on Neural Tangent Kernel.



## Fundamental Exploration

- Neural UCB [1] -- An Extension of LinUCB to NTK Space
- Neural TS [2] -- An Extension of LinTS to NTK Space
- EE-Net [3] -- Another Neural Network for Exploration

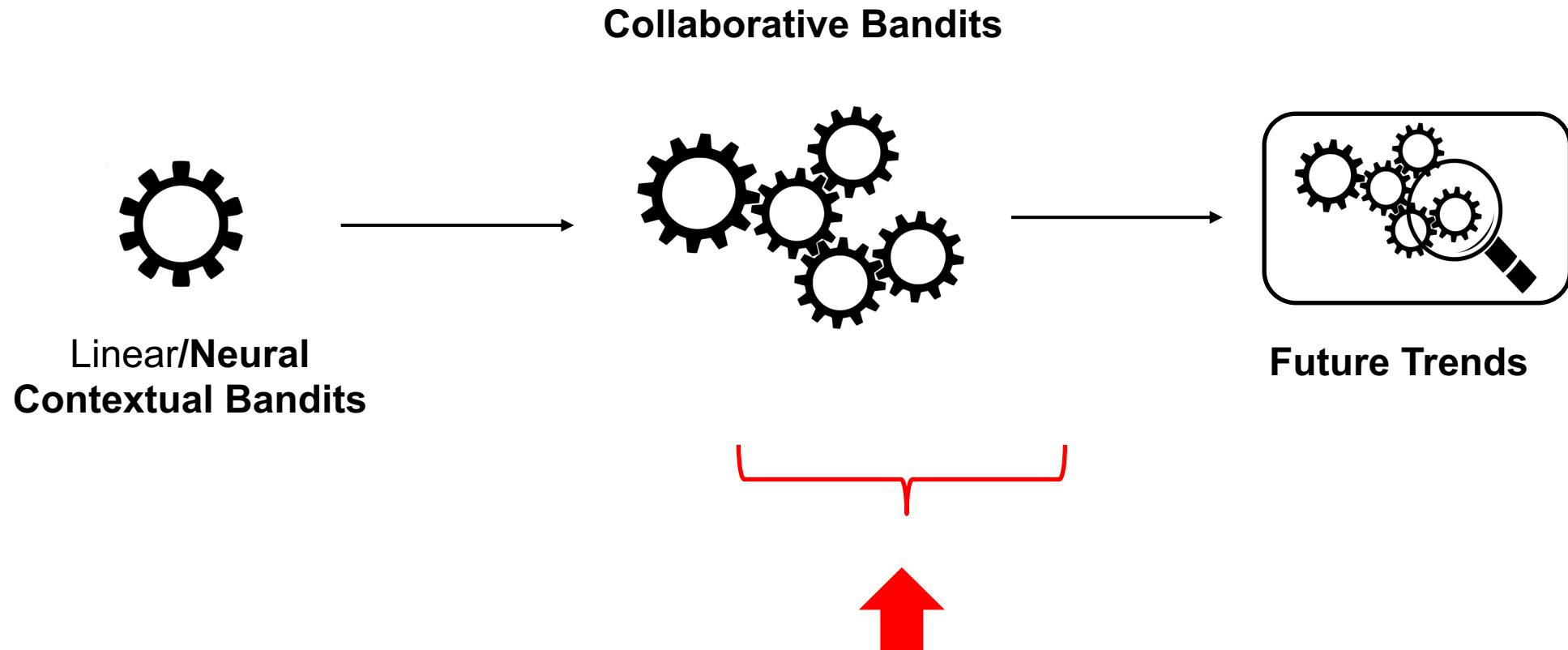


## Efficient Exploration

- Neural Linear UCB [4] -- LinUCB with Neural Representation
- Neural Network with Perturbed Reward [5] -- Implicit Exploration by Perturbing Rewards
- Neural Square CB[6] -- Exploration using Inverse Weight Gap Strategy

1. Zhou, Dongruo, Lihong Li, and Quanquan Gu. "Neural contextual bandits with ucb-based exploration." ICML 2020.
2. Zhang, Weitong, Dongruo Zhou, Lihong Li, and Quanquan Gu. "Neural thompson sampling." ICLR 2021.
3. Ban, Yikun, Yuchen Yan, Arindam Banerjee, and Jingrui He. "Ee-net: Exploitation-exploration neural networks in contextual bandits." ICLR 2022.
4. Xu, Pan, et al. "Neural contextual bandits with deep representation and shallow exploration." ICLR 2022.
5. Jia, Yiling, Weitong Zhang, Dongruo Zhou, Quanquan Gu, and Hongning Wang. "Learning neural contextual bandits through perturbed rewards." ICLR 2022.
6. Deb, Rohan, Yikun Ban, Shiliang Zuo, Jingrui He, and Arindam Banerjee. "Contextual bandits with online neural regression." ICLR 2024.

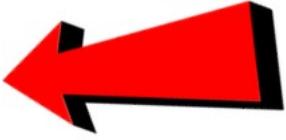
# Roadmap





## Introduction

- Background & Motivations
- Challenges



## Online Clustering of Bandits

- Clustering of Linear Bandits
- Clustering of Neural Bandits



## Graph Bandit Learning with Collaboration

- User side: Graph Neural Bandits
- Arm side: Neural Bandit with Arm Group Graph
- Other Scenarios: Bandit Learning with Graph Feedback & Online Graph Classification with Neural Bandit



## Bandits for Combo Recommendation

- Multi-facet Contextual Bandits

# Collaborative Contextual Bandits: Background & Motivation



- ❑ Conventional approaches, e.g., **collaborative and content-based filtering**:

	Book	Bag	Headphones	Game Controller
A	✓	✗	✓	✓
B	✓	✓	✗	✗
C	✓	✓	✗	
D	✗		✓	
E	✓	✓	?	✗

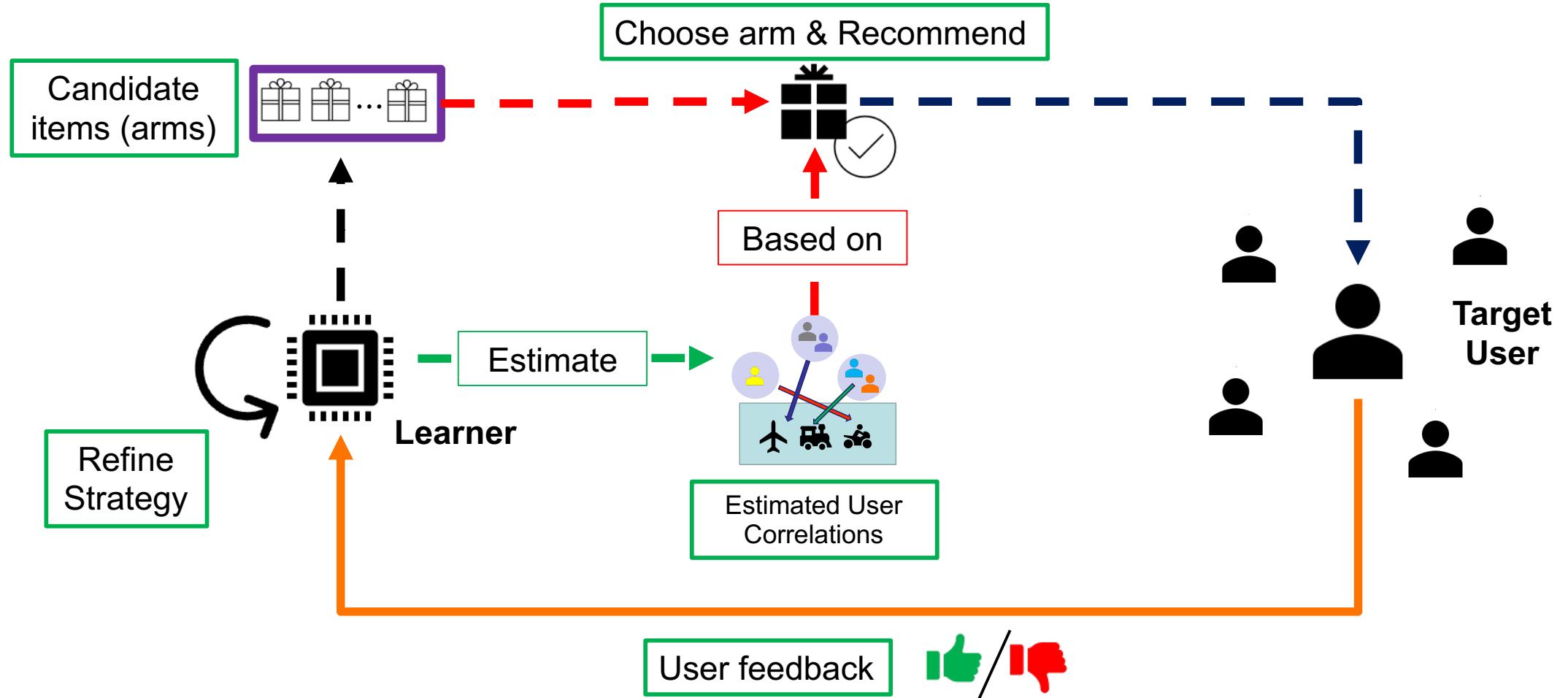
## Challenges:

(InCube Group)

- ❑ **Cold-start** problem (Lack of history data);
- ❑ **Rapid change** of recommendation content and user interests.
- ❑ Dilemma of **Exploitation and Exploration**.

# Collaborative Contextual Bandits: Background & Motivation

## □ Online recommendation scenario (in each round):



1. Lihong Li, et al. 2010. A contextual-bandit approach to personalized news article recommendation. In WWW. 661–670.
2. Claudio Gentile, et al. 2014. Online clustering of bandits. In ICML. 757–765.

# Collaborative Contextual Bandits: Background & Motivation

## □ The dilemma of exploitation and exploration:

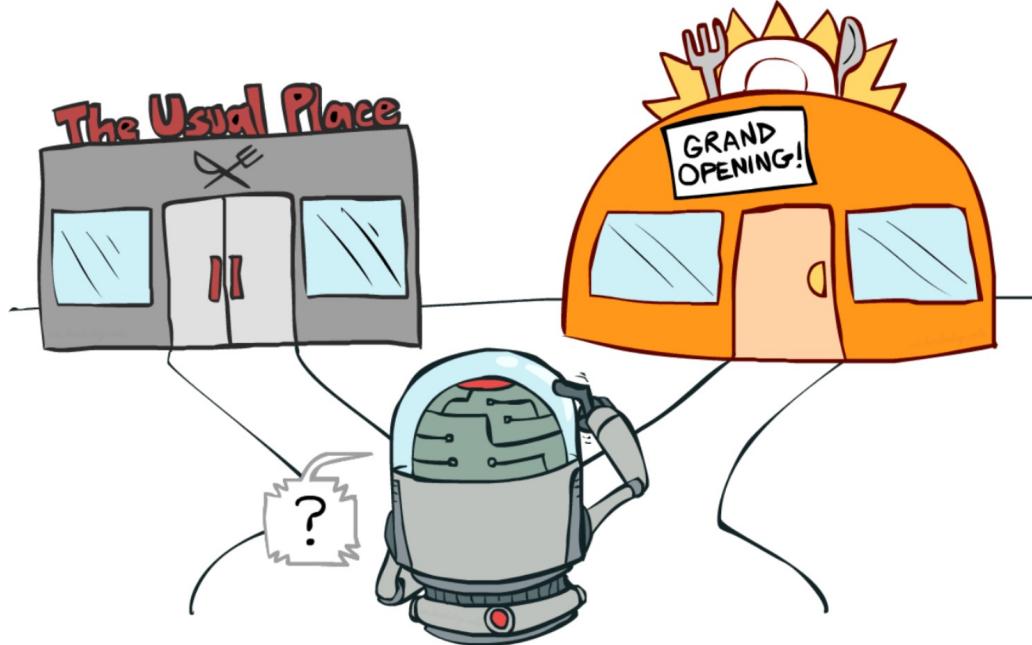
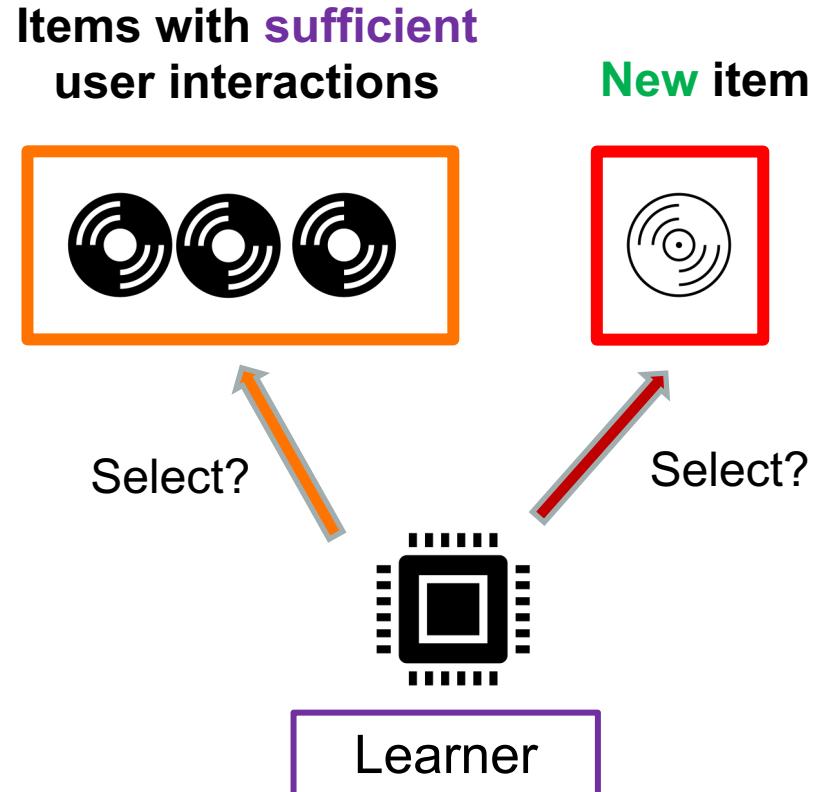


Figure: UC Berkeley CS 188, Introduction to Artificial Intelligence



1. Lihong Li, et al. 2010. A contextual-bandit approach to personalized news article recommendation. In WWW. 661–670.
2. Claudio Gentile, et al. 2014. Online clustering of bandits. In ICML. 757–765.

# Collaborative Contextual Bandits: Background & Motivation

- One user's decision is affected by other users.



- **Motivations:** Utilizing the **mutual influence / user collaborative effects** can
  - Improve **recommendation quality**.
  - Alleviate the **interaction scarcity** issue in terms of individual users.
  - Rapidly adapt to **new users / items** based on interactions with other users.

1. Lihong Li, et al. 2010. A contextual-bandit approach to personalized news article recommendation. In WWW. 661–670.  
2. Claudio Gentile, et al. 2014. Online clustering of bandits. In ICML. 757–765.

# Collaborative Contextual Bandits: Challenges

## □ **Challenge #1:** How to formally model user collaborations?

- User clusters [1, 2, 3, 4, 5, 6, 7], graphs with user nodes [10], etc.

## □ **Challenge #2:** How to discover user correlations?

- Leveraging the **known** user correlation information from the environment [8, 9];
- User clustering based on their past interactions [2,3,4,5,7], exploitation-exploration graph construction [10].

## □ **Challenge #3:** How to utilize user correlation to improve recommendation quality?

- Combination of linear estimations [1, 2, 3, 4, 5, 6], gradient-based meta-learning [7], graph neural networks [10], etc.

1. Gentile et. al., Online clustering of bandits. ICML 2014.

2. Li et. al., Improved algorithm on online clustering of bandits. IJCAI 2019.

3. Nguyen et. al., Dynamic clustering of contextual multi-armed bandits. CIKM 2014.

4. Gentile et. al., On context-dependent clustering of bandits. ICML 2017.

5. Ban et. al., Local clustering in contextual multi-armed bandits. WWW 2021.

6. Li et. al., Collaborative filtering bandits. SIGIR 2016.

7. Ban et. al., Meta clustering of neural bandits. In submission.

8. Nicolo Cesa-Bianchi et. al., A gang of bandits. NIPS 2013.

9. Wu et. al., Contextual bandits in a collaborative environment. SIGIR 2016.

10. Qi et. al., Graph neural bandits. KDD 2023.



## Introduction

- Background & Motivations
- Challenges



## Online Clustering of Bandits

- Clustering of Linear Bandits
- Clustering of Neural Bandits



## Graph Bandit Learning with Collaboration

- User side: Graph Neural Bandits
- Arm side: Neural Bandit with Arm Group Graph
- Other Scenarios: Bandit Learning with Graph Feedback & Online Graph Classification with Neural Bandit

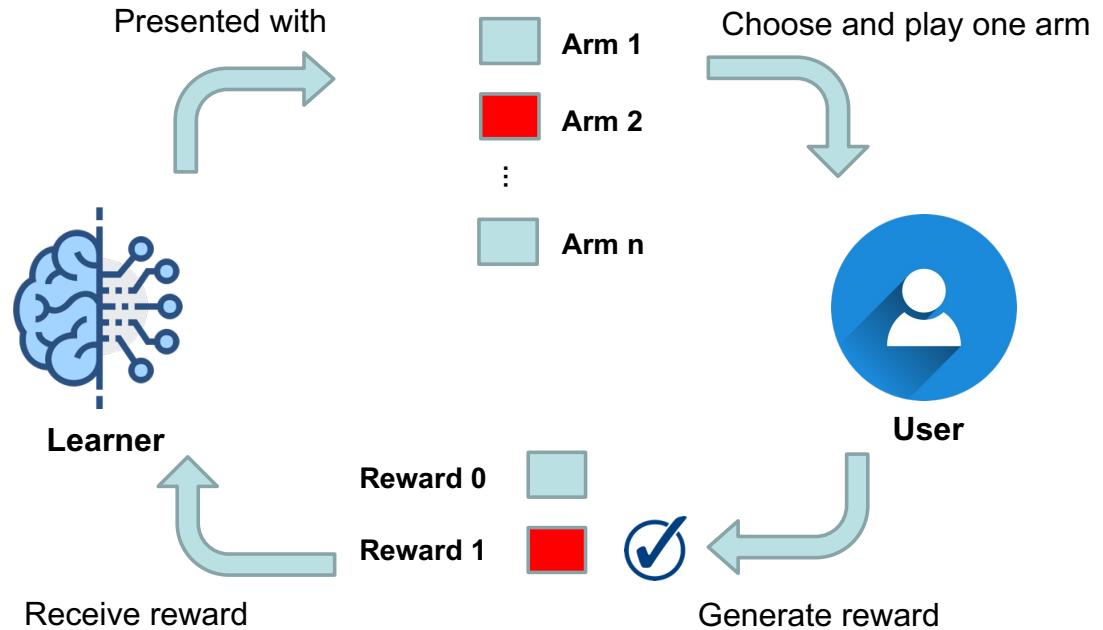


## Bandits for Combo Recommendation

- Multi-facet Contextual Bandits

# Online Clustering of Bandits

- Standard MAB algorithms view each user as an individual, without **user dependency**.



- For **refined recommendation strategies** when user correlations are unknown:
  - Objective #1: **Identify user clusters** in MAB;
  - Objective #2: **Exploit the user clusters** to improve the recommendation.



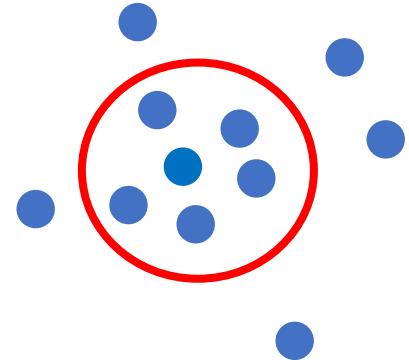
1. Gentile et. al., Online clustering of bandits. ICML 2014.
2. Li et. al., Improved algorithm on online clustering of bandits. IJCAI 2019.
3. Nguyen et. al., Dynamic clustering of contextual multi-armed bandits. CIKM 2014.
4. Gentile et. al., On context-dependent clustering of bandits. ICML 2017.
5. Ban et. al., Local clustering in contextual multi-armed bandits. WWW 2021.
6. Li et. al., Collaborative filtering bandits. SIGIR 2016.

# Online Clustering of Linear Bandits



## □ Clustering of Linear Bandits:

- Under **linear** stochastic contextual bandit settings:  $r = \langle \theta_u, x \rangle + \eta$ .
- User **correlation intensity** between  $u, u'$  is measured by  $\|\theta_u - \theta_{u'}\|_2$ .
  1. User clusters with identical preferences [1, 2, 3, 4, 6] ( $\forall u, u' \in \mathcal{N}: \theta_u = \theta_{u'}$ ).
  2. **A generalized formulation:**  $\gamma$ -cluster of users [5] ( $\forall u, u' \in \mathcal{N}: \|\theta_u - \theta_{u'}\|_2 \leq \gamma$ ).



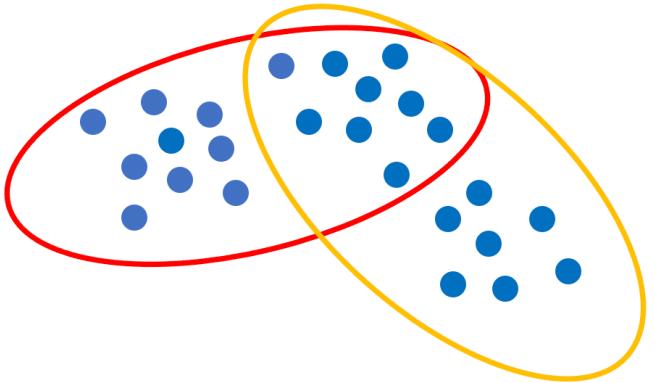
- 
1. Gentile et. al., Online clustering of bandits. ICML 2014.
  2. Li et. al., Improved algorithm on online clustering of bandits. IJCAI 2019.
  3. Nguyen et. al., Dynamic clustering of contextual multi-armed bandits. CIKM 2014.
  4. Gentile et. al., On context-dependent clustering of bandits. ICML 2017.

5. Ban et. al., Local clustering in contextual multi-armed bandits. WWW 2021.
6. Li et. al., Collaborative filtering bandits. SIGIR 2016.

# LOCB: Motivation and Challenges

## ➤ Challenge 1: When to ensure a set of identified users is a true cluster?

- Cluster: A set of users with similar expected rewards.
- Expected rewards of users are unknown.



## ➤ Challenge 2: Can we further reduce the clustering complexity?

- Previous works have clustering complexity  $O(n)$ .
- $n$  is the number of users.

## ➤ Challenge 3: Can we consider and address soft clustering?

- Consider overlapping clusters.
- A user is allowed to belong to multiple clusters.

# LOCB: Local Clustering of Linear bandits

- Characterizing similar users' behaviors:
  - **Definition ( $\gamma$ -Cluster)**: Given a subset of users  $\mathcal{N} \subseteq N$  and a threshold  $\gamma > 0$ ,  $\mathcal{N}$  is considered a  $\gamma$ -Cluster if it satisfies:  $\forall i, j \in \mathcal{N}, \|\theta^i - \theta^j\| < \gamma$ .
  
- Objectives:
  - **Objective #1**: Identify clusters among users, such that the clusters returned by the proposed algorithm are true  $\gamma$ -Clusters with probability at least  $1-\delta$ .
  - **Objective #2**: Leverage user clusters to improve the quality of recommendation, evaluated by **Regret**.

$$R_T = \mathbb{E}\left[\sum_{t=1}^T R_t\right] = \sum_{t=1}^T (\boxed{\theta_{i_t}^\top \mathbf{x}_t^*} - \boxed{\theta_{i_t}^\top \mathbf{x}_t})$$

Optimal Reward                          Received Reward

- Clustering Module + Pulling Module

# LOCB: Clustering Module

- Identify  $k$  clusters, given  $k$  seeds in each round:

- **Seed selection**: Randomly choose  $k$  users.
- **Neighbors**: Two users are neighbors if they belong to the same  $\gamma$ -cluster.
- **Potential neighbors**: User  $i$  is considered as the potential neighbor of seed user  $s$ , when:

$$\|\hat{\theta}_{i,t} - \hat{\theta}_{s,t}\| \leq B_{\theta,i}(m_{i,t}, \delta') + B_{\theta,s}(m_{s,t}, \delta').$$

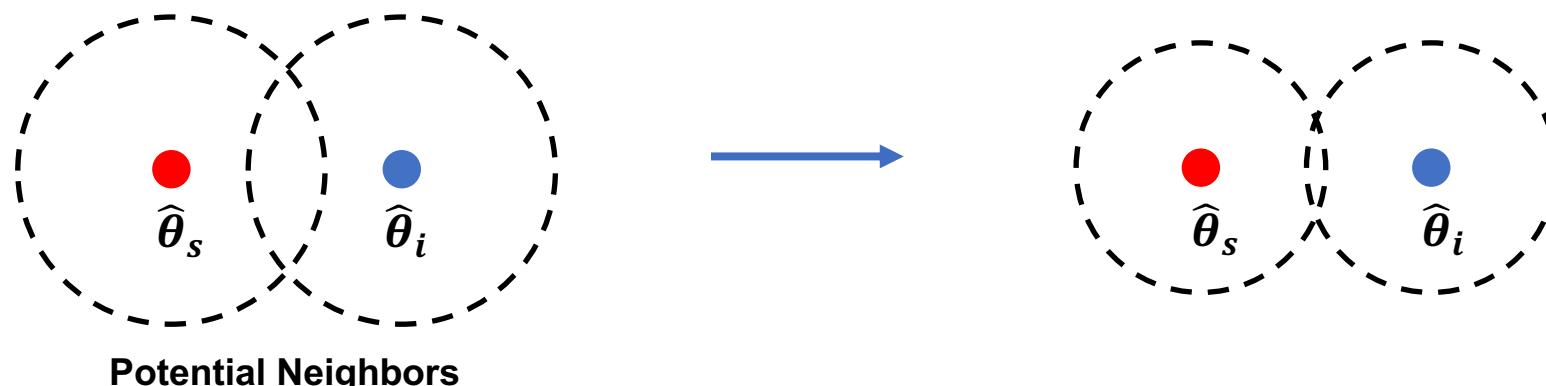
Seed-user parameter    User-specific bound    Seed-specific bound

$$B_{\theta,i}(m_{i,t}, \delta') = \frac{\sigma \sqrt{2d \log t + 2 \log(2/\delta')} + 1}{\sqrt{1 + h(m_{i,t}, H)}},$$

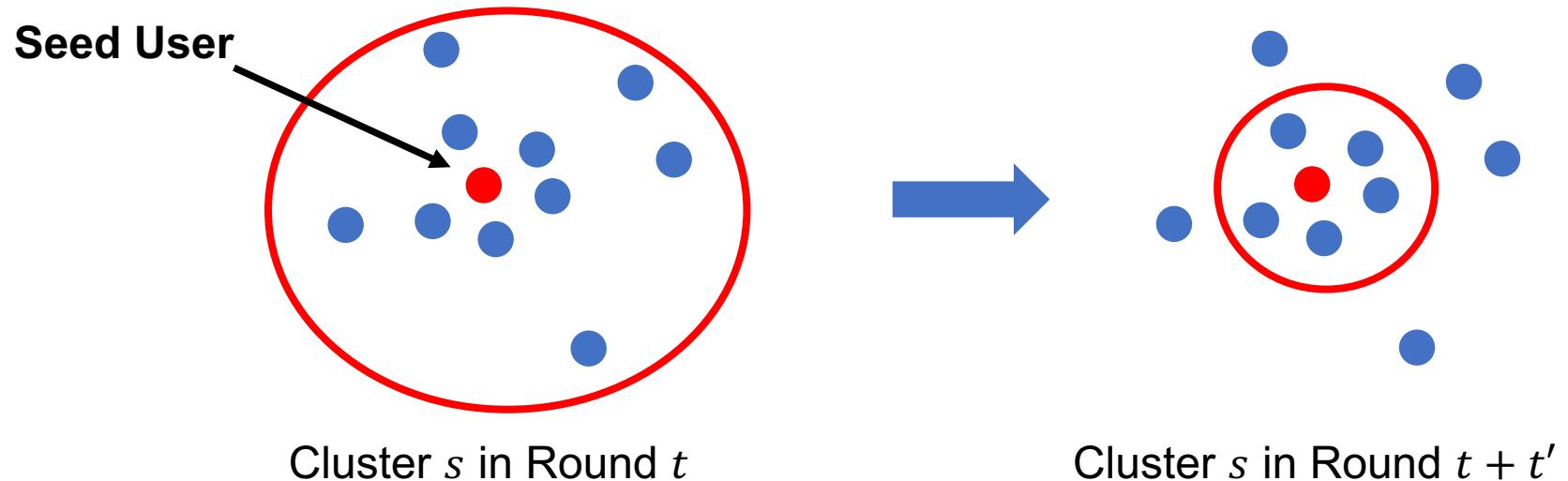
$$h(m_{i,t}, H) = \left( \frac{\lambda m_{i,t}}{4} - 8 \log\left(\frac{m_{i,t}+3}{H}\right) - 2\sqrt{m_{i,t} \log\left(\frac{m_{i,t}+3}{H}\right)} \right)$$

- **Cluster**: Seed user + Its potential neighbors.

- **User specific bound**: with a high probability,  $\|\hat{\theta}_{i,t} - \theta_i\| \leq B_{\theta,i}(m_{i,t}, \delta')$



# LOCB: Evolution of Clusters



- **Evolution of neighbors:**  $\|\hat{\theta}_{i,t} - \hat{\theta}_{s,t}\| \leq B_{\theta,i}(m_{i,t}, \delta') + B_{\theta,s}(m_{s,t}, \delta')$ .



User/seed specific bound is shrinking as more rounds are played for these users.

- **Termination criterion**
  - Given cluster  $\mathcal{N}_{s,t}$ , Clustering Module outputs this cluster when

$$\sup\{B_{\theta,i}(m_{i,t}, \delta') : i \in \mathcal{N}_{s,t}\} < \frac{\gamma}{8}$$

# LOCB: Pulling Module

## ➤ Individual CB vs. Cluster CB

□ Confidence interval for each **cluster**

$$\mathbb{P} \left( \forall t \in [T], |\hat{\theta}_{\mathcal{N}_{s,t}}^T \mathbf{x}_{a,t} - \theta_{\mathcal{N}_{s,t}}^T \mathbf{x}_{a,t}| > CB_{r,\mathcal{N}_{s,t}} \right) < \delta'$$

Cluster CB

$$CB_{r,\mathcal{N}_{s,t}} = \frac{1}{|\mathcal{N}_{s,t}|} \sum_{i \in \mathcal{N}_{s,t}} CB_{r,i}$$

□ Confidence interval for each **user**

$$\mathbb{P} \left( \forall t \in [T], |\hat{\theta}_{i,t}^T \mathbf{x}_{a,t} - \theta_i^T \mathbf{x}_{a,t}| > CB_{r,i} \right) < \delta'$$

Individual CB

## ➤ Pulling Module selects one arm by Cluster UCB:

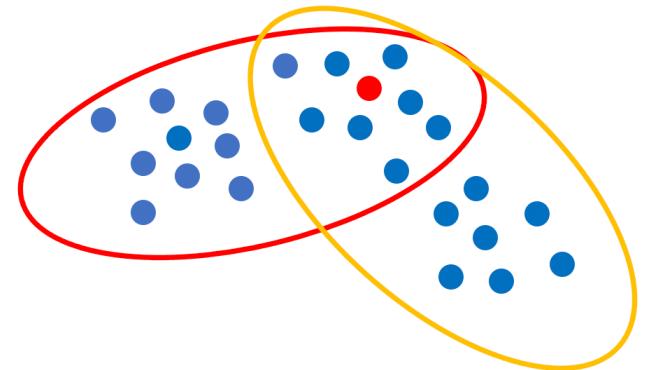
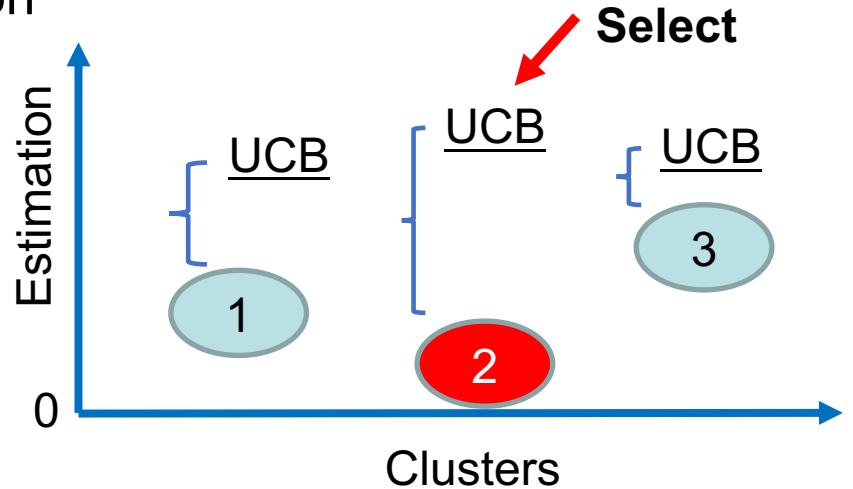
$$\mathbf{x}_t = \arg \max_{\mathbf{x}_{a,t} \in \mathbf{X}_t} \hat{\theta}_{\mathcal{N}_{s,t}}^T \mathbf{x}_{a,t} + CB_{r,\mathcal{N}_{s,t}}$$

Cluster-level exploration

$$\text{Cluster behavior } \hat{\theta}_{\mathcal{N}_{s,t}} = \frac{1}{|\mathcal{N}_{s,t}|} \sum_{i \in \mathcal{N}_{s,t}} \hat{\theta}_{i,t}.$$

# LOCB: Overlapping Clusters

- A user may belong to multiple overlapping clusters:
  - Cluster selection



- Pulling Module selects the **cluster with the maximum potential**:

$$\mathbf{x}_t = \arg \max_{\mathbf{x}_{a,t} \in \mathcal{X}_t} \max_{s \in S_t(i_t)} \left( \hat{\theta}_{N_{s,t}}^T \mathbf{x}_{a,t} + CB_{r,N_{s,t}} \right)$$

↑                      ↑  
 Arm set          Cluster set ( $O(k)$ )

# LOCB: Results

## ➤ Theoretical analysis:

### □ Correctness ✓

**THEOREM 5.1 (CORRECTNESS).** Given a threshold  $\gamma$  and a set of seeds  $S \subseteq N$ , for each  $s \in S$ , let  $N_s$  represent the cluster output by LOCB with respect to  $s$ . The terminate criterion of Clustering module is defined as:

$$\sup\{B_{\theta_i}(m_{i,t}, \delta') : i \in N_{s,t}\} < \frac{\gamma}{8}.$$

Then, with probability at least  $1 - \delta$ , after the Clustering module terminates, for each  $s \in S$ , it has

$$\forall i, j \in N_s, \|\theta_i - \theta_j\| < \gamma.$$

### □ Efficiency ✓

**THEOREM 5.2.** Suppose each user is evenly served and  $m_{i,t} \geq \frac{2 \times 32^2}{\lambda^2} \log\left(\frac{2nd}{\delta'}\right) \log\left(\frac{32^2}{\lambda^2} \log\left(\frac{2nd}{\delta'}\right)\right)$  for any  $i \in N$ . Then, with probability at least  $1 - \delta$ , the number of rounds  $\hat{T}$  needed for the Clustering module to terminate is upper bounded by

$$\hat{T} < \frac{2nd}{C} \log \frac{nd}{C} + \frac{2n}{C} \left( \log\left(\frac{2^{(d+1)} n}{\delta}\right) - \frac{\gamma^2 - 256}{512\sigma^2} \right) + n.$$

$$\text{where } C = \frac{\lambda\gamma^2}{16^3\sigma^2}.$$

### □ Effectiveness ✓

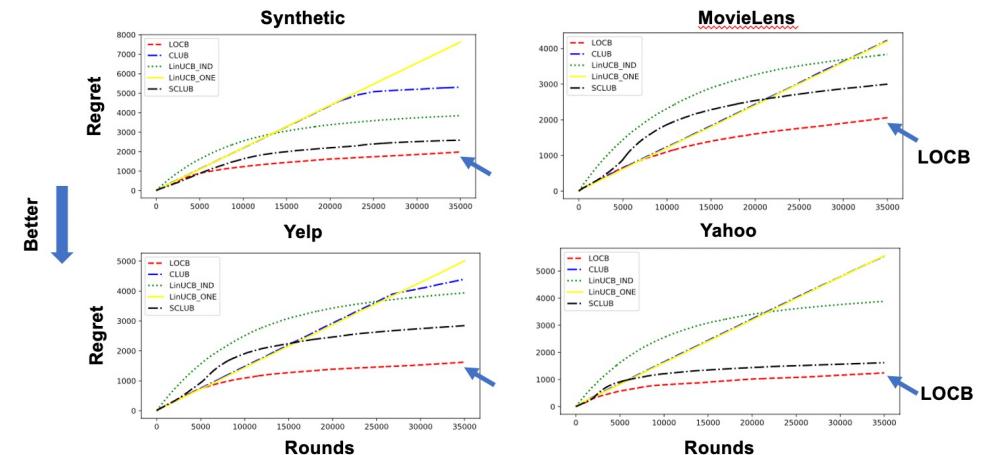
**THEOREM 5.3.** Suppose that each user is evenly served. Given  $\gamma$  and a set of seeds  $S$ , after  $T > \hat{T}$  rounds, the accumulated regret of LOCB can be upper bounded as follows:

$$R_T \leq \left[ \sqrt{nT} \cdot \sqrt{2d \log(1 + T/dn)} \cdot O\left(\sqrt{d \log(T/\delta)}\right) \right] \\ + \left( T - O(nd \log nd) \right) \gamma + O(nd \log nd) \cdot O\left(\sqrt{d \log(Tn/\delta)}\right).$$

## ➤ Evaluations:

### □ Improve performance up to 12.4%.

	Synthetic			Yelp				MovieLens			Yahoo			
	F1	Pre	Recall	F1	Pre	Recall		F1	Pre	Recall	F1	Pre	Recall	
N-CLUB	0.390	0.246	0.943	0.484	0.334	0.884		0.417	0.286	0.773	0.454	0.334	0.709	
ST-CLUB	0.578	0.549	0.612	0.626	0.593	0.663		0.520	0.429	0.663	0.528	0.385	0.841	
ST-SCLUB	0.714	0.745	0.687	0.768	0.863	0.693		0.538	0.739	0.424	0.632	0.781	0.532	
N-LOCB	0.662	0.618	0.714	0.675	0.620	0.743		0.472	0.432	0.524	0.615	0.553	0.692	
LOCB	<b>0.880</b>	<b>0.913</b>	<b>0.856</b>	<b>0.879</b>	<b>0.908</b>	<b>0.853</b>		<b>LOCB</b>	<b>0.814</b>	<b>0.892</b>	<b>0.749</b>	<b>0.869</b>	<b>0.935</b>	<b>0.813</b>





## Introduction

- Background & Motivations
- Challenges



## Online Clustering of Bandits

- Clustering of Linear Bandits
- Clustering of Neural Bandits



## Graph Bandit Learning with Collaboration

- User side: Graph Neural Bandits
- Arm side: Neural Bandit with Arm Group Graph
- Other Scenarios: Bandit Learning with Graph Feedback & Online Graph Classification with Neural Bandit



## Bandits for Combo Recommendation

- Multi-facet Contextual Bandits

# Online Clustering of Neural Bandits



## ➤ Challenge 1: How to efficiently determining a user's relative group?

- User relative group: A set of users with **same expected rewards on a specific item (arm)**.
- Expected rewards of users are unknown. The mapping function  $h(x)$  can be linear or non-linear.

## ➤ Challenge 2: Effective parametric representation of dynamic clusters?

- Introducing **meta-learner** capable of representing and swiftly adapting to evolving user clusters.
- Enabling the rapid acquisition of nonlinear cluster representations.

## ➤ Challenge 3: Balancing exploitation and exploration?

- A novel UCB-type exploration strategy.
- Taking both user-side and meta-side information into account.



# M-CNB: Meta Clustering of Neural Bandits

## ➤ Characterizing user clusters without linear assumptions:

**Definition 3.1** (Relative Cluster). In round  $t$ , given an arm  $\mathbf{x}_{t,i} \in \mathbf{X}_t$ , a relative cluster  $\mathcal{N}(\mathbf{x}_{t,i}) \subseteq N$  with respect to  $\mathbf{x}_{t,i}$  satisfies

- (1)  $\forall u, u' \in \mathcal{N}(\mathbf{x}_{t,i}), \mathbb{E}[r_{t,i}|u] = \mathbb{E}[r_{t,i}|u']$
- (2)  $\nexists \mathcal{N}' \subseteq N$ , s.t.  $\mathcal{N}'$  satisfies (1) and  $\mathcal{N}(\mathbf{x}_{t,i}) \subset \mathcal{N}'$ .

**Definition 3.2** ( $\gamma$ -gap). Given two different cluster  $\mathcal{N}(\mathbf{x}_{t,i}), \mathcal{N}'(\mathbf{x}_{t,i})$ , there exists a constant  $\gamma > 0$ , such that

$$\forall u \in \mathcal{N}(\mathbf{x}_{t,i}), u' \in \mathcal{N}'(\mathbf{x}_{t,i}), |\mathbb{E}[r_{t,i}|u] - \mathbb{E}[r_{t,i}|u']| \geq \gamma.$$

## ➤ Objectives:

- **Objective #1:** Identify clusters among users, such that the clusters returned by the proposed algorithm are accurate user clusters.
- **Objective #2:** Leverage user correlations to improve the quality of recommendation, evaluated by Pseudo Regret.

$$R_T = \sum_{t=1}^T \mathbb{E}[r_t^* - r_t | u_t, \mathbf{X}_t], \quad \mathbb{E}[r_t^* | u_t, \mathbf{X}_t] = \max_{\mathbf{x}_{t,i} \in \mathbf{X}_t} h_{u_t}(\mathbf{x}_{t,i})$$

Optimal Reward                          Received Reward

General reward function

# M-CNB: Clustering Module



- Identify relative cluster for target user  $u_t \in N$ :

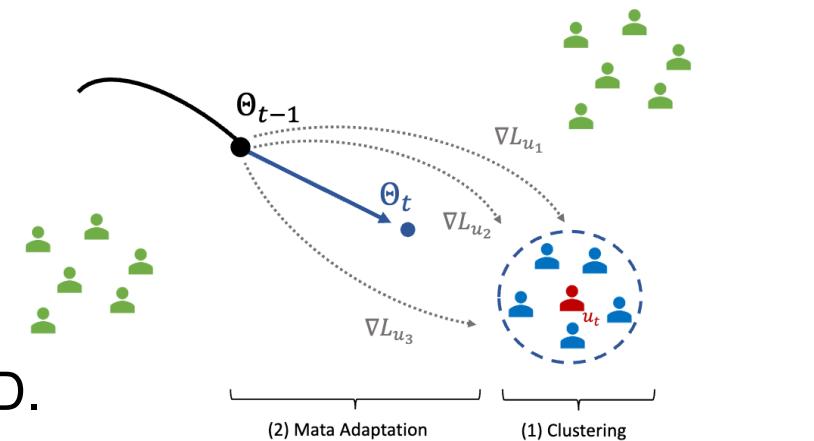
- **Arm-specific**: Different arms can induce distinct user clusters.
- **User models**: Each user  $u \in N$  is assigned with their own user models  $f(\cdot; \theta^u)$ .
- **Potential neighbors**: User  $u$  is the potential neighbor of target user  $u_t$ , when:

$$\widehat{\mathcal{N}}_{u_t}(\mathbf{x}_{t,i}) = \left\{ u \in N \mid |f(\mathbf{x}_{t,i}; \theta_{t-1}^u) - f(\mathbf{x}_{t,i}; \theta_{t-1}^{u_t})| \leq \frac{\nu - 1}{\nu} \gamma \right\}.$$

Preference est. for other users      Preference est. for target user      Tunable distance threshold

- **Meta-adaptation**: Adapting to estimated user clusters.

- Randomly draw a few samples from the historical data of detected cluster  $\{\mathcal{T}_{t-1}^u\}_{u \in \widehat{\mathcal{N}}_{u_t}(\mathbf{x}_{t,i})}$ .
- The meta-model  $f(\cdot; \theta)$  is adapted through a few steps of SGD.



# M-CNB: Pulling Module

- Informed UCB for reward estimation:

Meta-Model Error

$$\sum_{t=1}^T \mathbb{E}_{r_t | \mathbf{x}_t} \left[ |f(\mathbf{x}_t; \Theta_t) - r_t| \mid u_t \right] \leq \sum_{t=1}^T \underbrace{\frac{O(\|\nabla_\Theta f(\mathbf{x}_t; \Theta_t) - \nabla_\theta f(\mathbf{x}_t; \theta_0^{u_t})\|_2)}{m^{1/4}}}_{\text{Meta-side info}} + \sum_{u \in N} \mu_T^u \left[ O\left(\sqrt{\frac{S+1}{2\mu_T^u}}\right) + \sqrt{\frac{2\log(1/\delta)}{\mu_T^u}} \right],$$

Gradient Discrepancy between User Model and the Meta-Model

User-side Upper Bound based on Service Frequency

- Pulling Module selects one arm by Cluster UCB:

$$\mathbf{x}_t = \arg_{\mathbf{x}_{t,i} \in \mathcal{X}_t} \max \mathbf{U}_{t,i}$$

$$\mathbf{U}_{t,i} = f(\mathbf{x}_{t,i}; \Theta_{t,i}) + \frac{\|\nabla_\Theta f(\mathbf{x}_{t,i}; \Theta_{t,i}) - \nabla_\theta f(\mathbf{x}_{t,i}; \theta_0^{u_t})\|_2}{m^{1/4}} + \sqrt{\frac{S+1}{2\mu_t^u}} + \sqrt{\frac{2\log(1/\delta)}{\mu_t^u}}$$

Meta-Model Reward Estimation

UCB

# M-CNB: Theoretical and Empirical Results

## ➤ Theoretical analysis from two aspects:

### □ Instance-dependent Regret Bound ✓

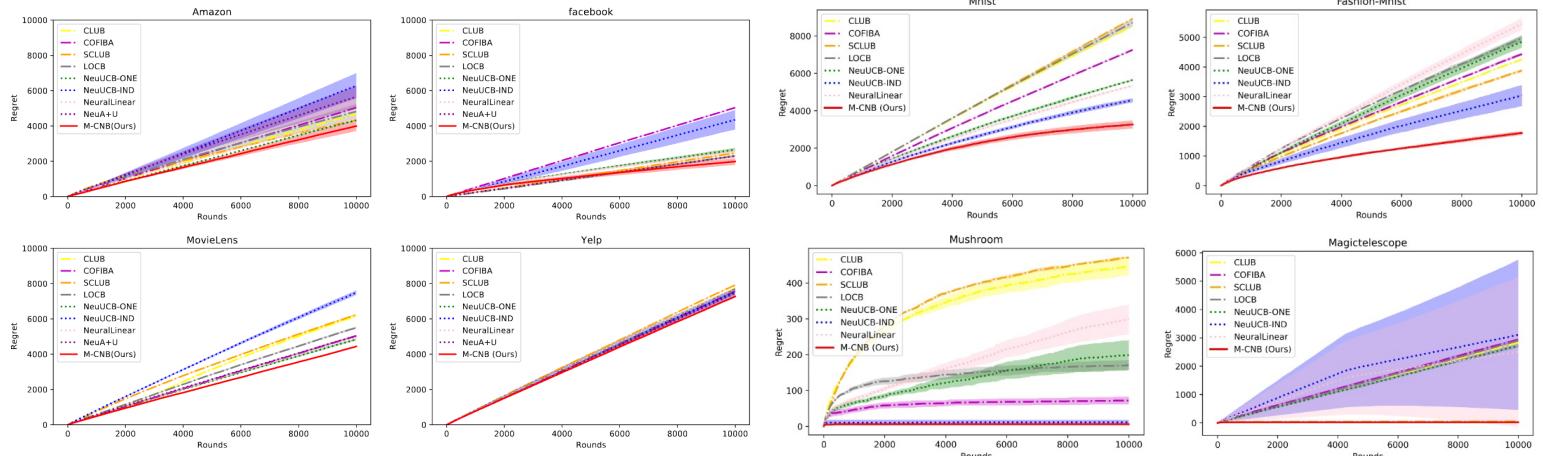
**Theorem 5.1.** Given the number of rounds  $T$  and  $\gamma$ , for any  $\delta \in (0, 1)$ ,  $R > 0$ , suppose  $m \geq \tilde{\Omega}(\text{poly}(T, L, R) \cdot Kn \log(1/\delta))$ ,  $\eta_1 = \eta_2 = \frac{R^2}{\sqrt{m}}$ , and  $\mathbb{E}[|\mathcal{N}_{u_t}(\mathbf{x}_t)|] = \frac{n}{q}$ ,  $t \in [T]$ . Then, with probability at least  $1 - \delta$  over the initialization, Algorithm 1 achieves the following regret upper bound:

$$R_T \leq \sqrt{qT \cdot S_{TK}^* + O(1)} + O(\sqrt{2qT \log(O(1)/\delta)}).$$

where  $S_{TK}^* = \inf_{\theta \in B(\theta_0, R)} \sum_{t=1}^{TK} \mathcal{L}_t(\theta)$ .

## ➤ Evaluations:

- M-CNB (red curve) outperforms baselines, for both recommendation and classification data sets.



### □ NTK-regression based Regret Bound ✓

**Lemma 5.3.** Suppose Assumption 5.1 and conditions in Theorem 5.1 holds where  $m \geq \tilde{\Omega}(\text{poly}(T, L) \cdot Kn \lambda_0^{-1} \log(1/\delta))$ . With probability at least  $1 - \delta$  over the initialization, there exists  $\theta' \in B(\theta_0, \tilde{\Omega}(T^{3/2}))$ , such that

$$\mathbb{E}[S_{TK}^*] \leq \mathbb{E}\left[\sum_{t=1}^{TK} \mathcal{L}_t(\theta')\right] \leq \tilde{\mathcal{O}}\left(\sqrt{\tilde{d}} + S\right)^2 \cdot \tilde{d}.$$

# Online Clustering of Bandits

□ Motivations: We need to **estimate user correlations on the fly**, during online recommendation.

□ **Clustering of Linear Bandits** [1, 2, 3, 4, 5, 6]:

- Under **linear** stochastic contextual bandit settings:  $r = \langle \theta_u, x \rangle + \eta$ .
- User **correlation intensity** between  $u, u'$  is measured by  $\|\theta_u - \theta_{u'}\|_2$ .
- Adopt *combination of linear estimators* for reward estimation & exploration.

□ **Clustering of Neural Bandits** [7]:

- Under **neural** stochastic contextual bandit settings:  $r = h_u(x) + \eta$ .
- User clusters with identical preferences ( $\forall u, u' \in \mathcal{N}, x \in \mathbb{R}^d: h_u(x) = h_{u'}(x)$ ).
- Utilizing *gradient-based Meta-Learning* for reward estimation & exploration.

- 
1. Gentile et. al., Online clustering of bandits. ICML 2014.
  2. Li et. al., Improved algorithm on online clustering of bandits. IJCAI 2019.
  3. Nguyen et. al., Dynamic clustering of contextual multi-armed bandits. CIKM 2014.
  4. Gentile et. al., On context-dependent clustering of bandits. ICML 2017.

5. Ban et. al., Local clustering in contextual multi-armed bandits. WWW 2021.
6. Li et. al., Collaborative filtering bandits. SIGIR 2016.
7. Ban et. al., Meta clustering of neural bandits. In submission.



## Introduction

- Background & Motivations
- Challenges



## Online Clustering of Bandits

- Clustering of Linear Bandits
- Clustering of Neural Bandits



## Graph Bandit Learning with Collaboration

- User side: Graph Neural Bandits
- Arm side: Neural Bandit with Arm Group Graph
- Other Scenarios: Bandit Learning with Graph Feedback & Online Graph Classification with Neural Bandit



## Application in Recommender Systems

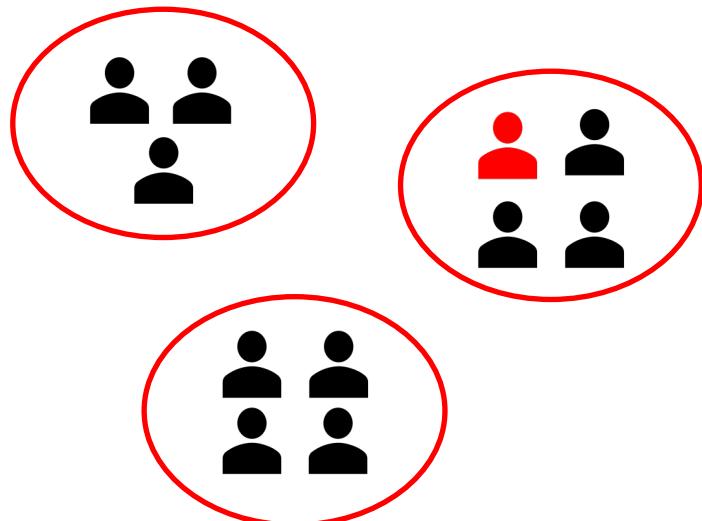
- Multi-facet Personalized Recommendation

# Collaborative Exploration: Graph Bandits Learning



## Clustering of Bandits [1,2]

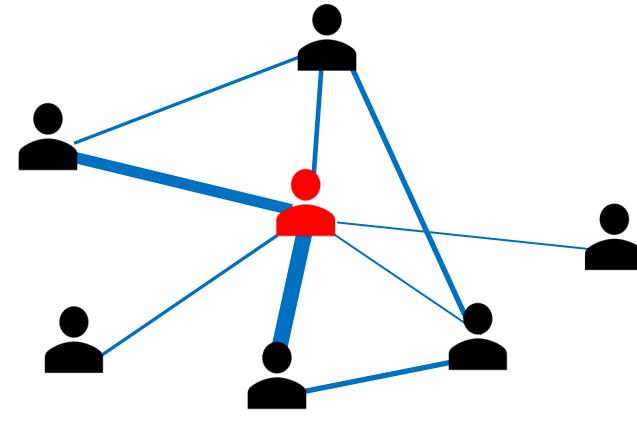
- **Coarse-grained user correlations:**
  - Users within the same cluster share **identical preferences**.
  - **Contribute equally** to serving user.



User (Bandit)

## Graph Bandits Learning [3]

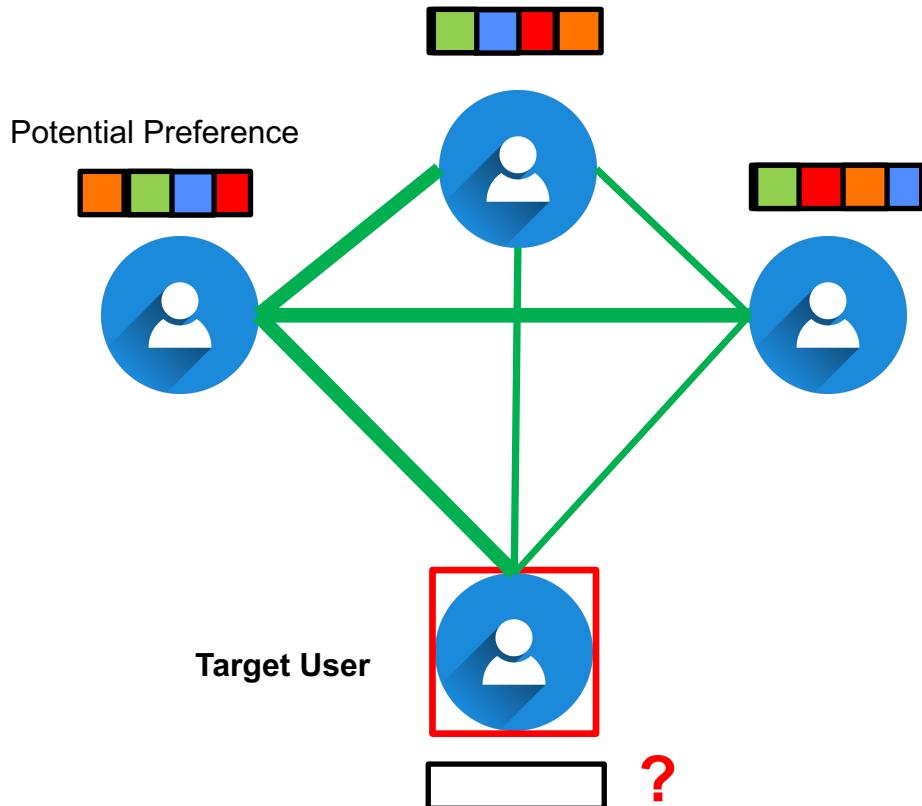
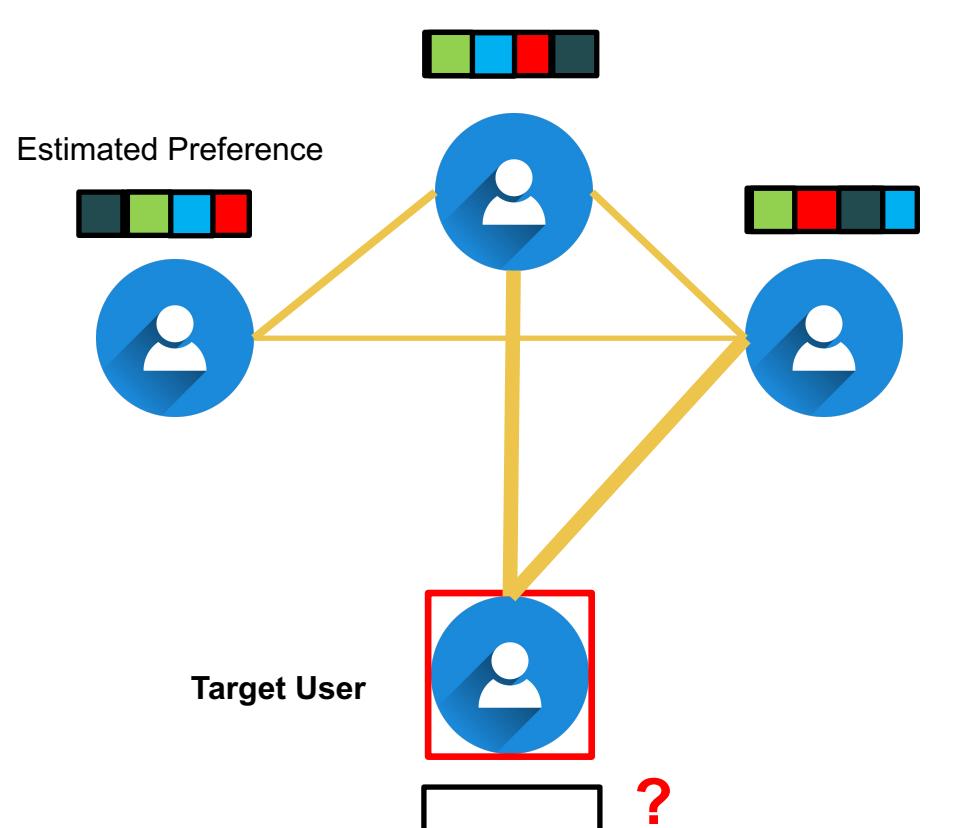
- **Fine-grained user correlations:**
  - Heterogeneity of users is preserved.
  - **Contribute differently** to serving user.



— User correlation with strength  
**(Unknown)**

1. Claudio Gentile, et al. 2014. Online clustering of bandits. In ICML. 757–765.
2. Shuai Li, et al. 2019. Improved Algorithm on Online Clustering of Bandits. In IJCAI. 2923–2929.
3. Y. Qi, Y. Ban\*, and J. He. Graph neural bandits. KDD 2023.

# GNB: Exploitation and Exploration Graphs



User Exploitation Graph

— Correlation w.r.t. Exploitation

User Exploration Graph

— Correlation w.r.t. Exploration

# GNB: Problem Definition



- For each round  $t \in [T]$  :
  - Receive a target user  $u_t \in \mathcal{U}$ , and candidate arms (items)  $\mathcal{X}_t$ .
    - $\mathcal{X}_t = \{\mathbf{x}_{i,t} \in \mathbb{R}^d, \text{ (e.g., } \odot \text{)}\}_{i \in [a]}$
  - Reward  $r_{i,t} = h(\mathcal{G}_{i,t}^{(1), *}, u_t, x_{i,t}) + \underline{\text{noise}}$ .  
Zero-mean  
noise
  - Learner **selects** arm  $x_t \in \mathcal{X}_t$  as the recommendation.

# GNB: Problem Definition



- For each round  $t \in [T]$  :
  - Receive a target user  $u_t \in \mathcal{U}$ , and candidate arms (items)  $\mathcal{X}_t$ .
    - $\mathcal{X}_t = \{\mathbf{x}_{i,t} \in \mathbb{R}^d, \text{ (e.g., } \text{)}\}_{i \in [a]}$
  - Reward  $r_{i,t} = h(\underline{\mathcal{G}_{i,t}^{(1), *}}, u_t, x_{i,t}) + \epsilon_{i,t}$ .
  - Learner **selects** arm  $x_t \in \mathcal{X}_t$  as the recommendation.

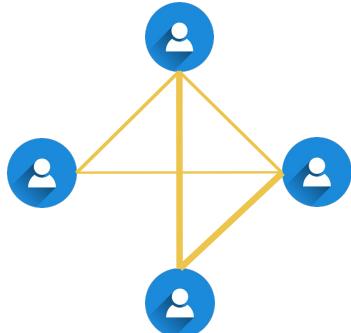
## ➤ Definition: User Correlation (Exploitation) Graph

- Given arm  $\mathbf{x}_{i,t}$ , unknown user exploitation graph

$$\mathcal{G}_{i,t}^{(1), *} = (\mathcal{U}, E, W_{i,t}^{(1), *})$$

- $\mathcal{U}$ : set of nodes (**users**)
- $E = \{e(u, u')\}_{u, u' \in \mathcal{U}}$ : set of edges
- $W_{i,t}^{(1), *}$ : set of **edge weights**

$$W_{i,t}^{(1), *} = \Psi^{(1)}(\mathbb{E}[r_{i,t} | \mathbf{u}_1, x_{i,t}], \mathbb{E}[r_{i,t} | \mathbf{u}_2, x_{i,t}])$$



User correlations w.r.t. the **expected reward**  
(**Exploitation Graph**)

# GNB: User Exploration Graph

## ➤ Definition: User Exploration Graph

- For arm  $x_{i,t}$ , unknown user exploration graph

$$\mathcal{G}_{i,t}^{(2),*} = (\mathcal{U}, E, \underline{W}_{i,t}^{(2),*})$$

Set of edge weights

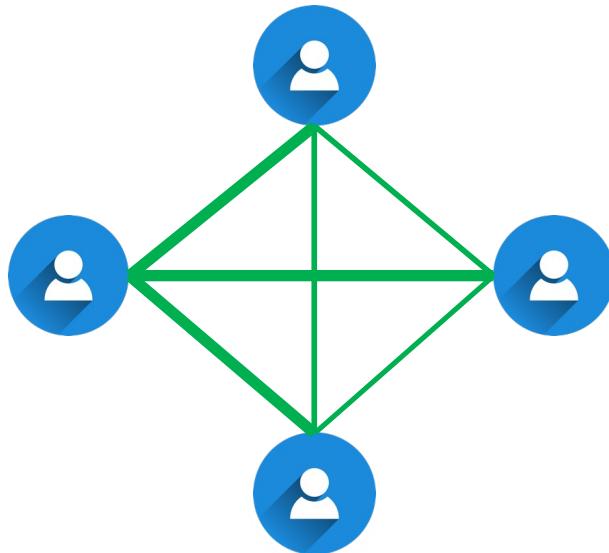
- For users  $u_1, u_2 \in \mathcal{U}$ , the corresponding edge weight:

- $w_{i,t}^{(2),*}(u_1, u_2) = \underline{\Psi}^{(2)} \left( \underbrace{\mathbb{E}[r_{i,t} | u_1, x_{i,t}] - f_{u_1}^{(1)}(x_{i,t}), \mathbb{E}[r_{i,t} | u_2, x_{i,t}] - f_{u_2}^{(1)}(x_{i,t})}_{\text{Pre-defined mapping}} \right)$

Potential Gain

## Potential Gain:

- $\mathbb{E}[r | u, x] - f_u^{(1)}(x)$
- Measures the uncertainty for the reward estimation



User correlations w.r.t. the Potential Gain  
(Exploration Graph)

# GNB: Problem Definition

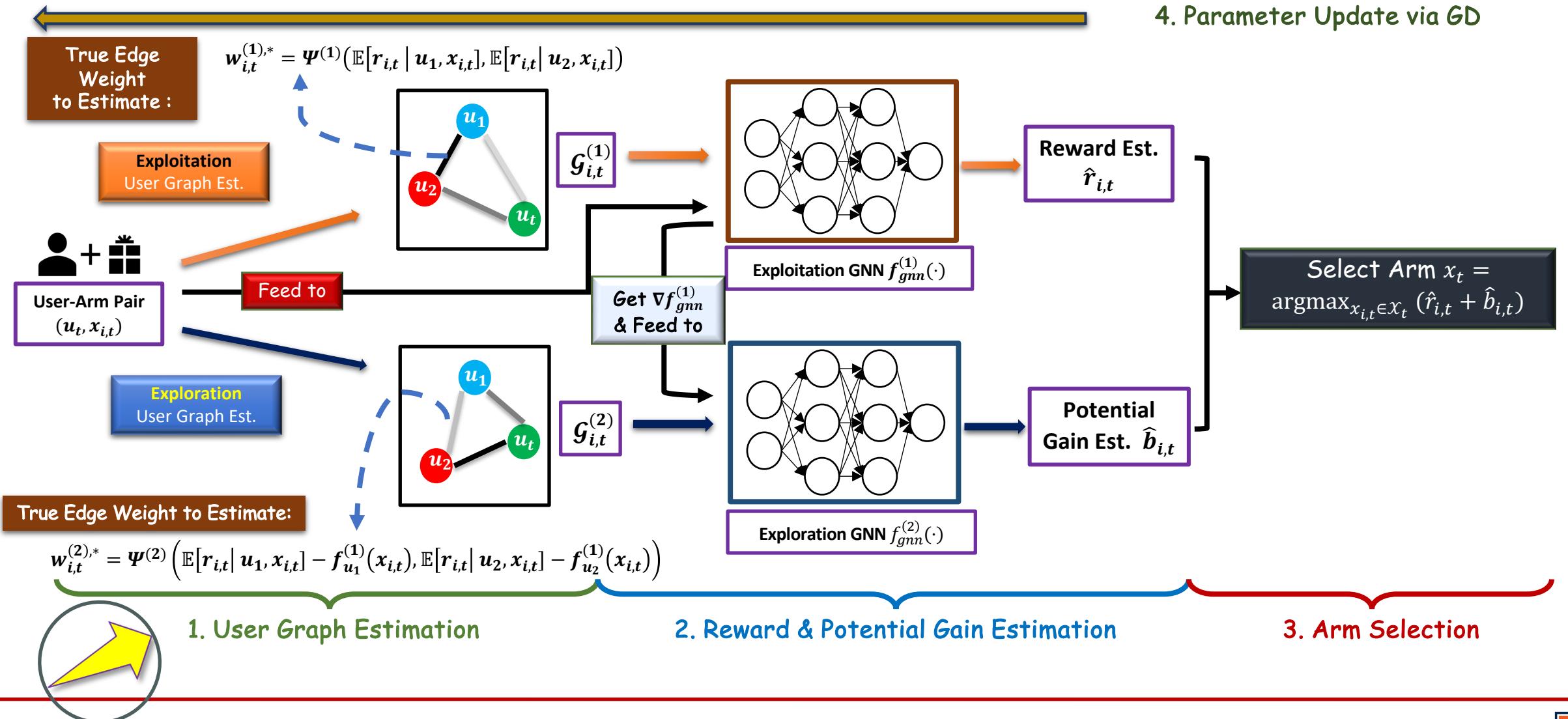
- For each round  $t \in [T]$  :
  - Receive a target user  $u_t \in \mathcal{U}$ , and candidate arms  $\mathcal{X}_t$ .
    - $\mathcal{X}_t = \{\mathbf{x}_{i,t} \in \mathbb{R}^d, \text{ (e.g., } \text{)}\}_{i \in [a]}$
  - Reward  $r_{i,t} = h(\mathcal{G}_{i,t}^{(1),*}, u_t, x_{i,t}) + \epsilon_{i,t}$ .
  - Learner **selects** arm  $x_t \in \mathcal{X}_t$  as the recommendation.
  
- Definition: User Correlation (Exploitation) Graph
  - Given arm  $x_{i,t}$ , **unknown** user exploitation graph
  $\mathcal{G}_{i,t}^{(1),*} = (\mathcal{U}, E, W_{i,t}^{(1),*})$ 
    - $W_{i,t}^{(1),*}$ : set of **edge weights**
  - For users  $u_1, u_2 \in \mathcal{U}$ , the corresponding **edge weight**:
    - $w_{i,t}^{(1),*}(u_1, u_2) = \Psi^{(1)}(\mathbb{E}[r_{i,t} | u_1, x_{i,t}], \mathbb{E}[r_{i,t} | u_2, x_{i,t}])$

## ➤ Objective: Minimizing Pseudo Regret

$$R(T) = \sum_{t=1}^T \mathbb{E}[r_t^* - r_t]$$

Chosen arm reward  
 Optimal arm reward  
 $\mathbb{E}[r_t^*] = \max_{i \in [a]} \mathbb{E}[r_{i,t}]$

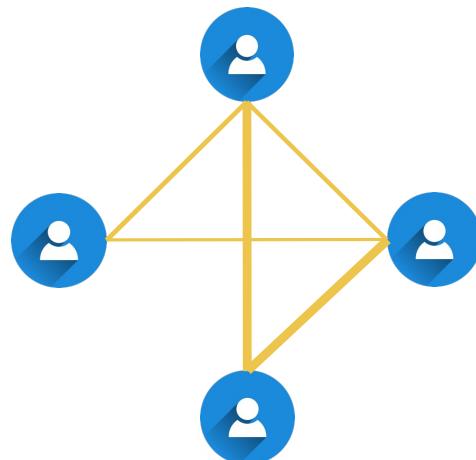
# GNB: Framework Overview



# User Exploitation Graph Estimation

## User Preference (expected reward) Estimation:

- Estimated by **user exploitation networks**  
 $\{f_u^{(1)}\}_{u \in \mathcal{U}}$
- Approximating  $\mathbb{E}[r | u, x]$
- Input:  $x$       Label:  $r$



User correlations w.r.t. the **expected reward**  
**(Exploitation Graph)**

## ➤ User Exploitation Graph Estimation:

- Given arm  $x_{i,t}$ , **estimated** user exploitation graph  
 $\mathcal{G}_{i,t}^{(1)} = (\mathcal{U}, E, W_{i,t}^{(1)})$ 
  - $W_{i,t}^{(1)}$ : set of **estimated** edge weights

- For users  $u_1, u_2 \in \mathcal{U}$ , **estimated** edge weight
  - $w_{i,t}^{(1)}(u_1, u_2) = \Psi^{(1)} \left( f_{u_1}^{(1)}(x_{i,t}), f_{u_2}^{(1)}(x_{i,t}) \right)$

**Estimated User  
Preference**

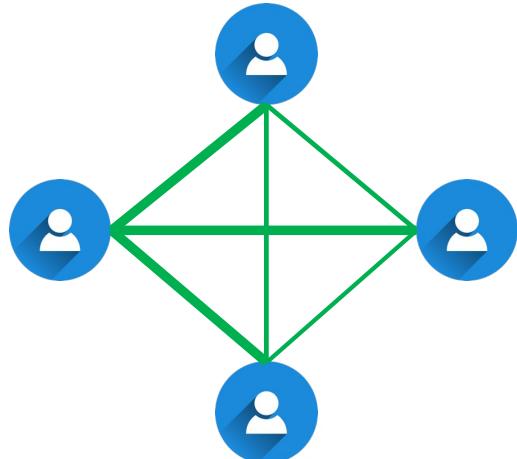
# User Exploration Graph Estimation

## Potential Gain:

- Estimated by user **exploration** networks

$$\{f_u^{(2)}\}_{u \in \mathcal{U}}$$

- Input:**  $\nabla f_u^{(1)}(x)$  -- the **gradients** of  $f_u^{(1)}$ .
- Label:**  $r_u - f_u^{(1)}(x)$ .



User correlations w.r.t. the Potential Gain  
**(Exploration Graph)**

## ➤ User **Exploration Graph Estimation**:

- Given arm  $x_{i,t}$ , **estimated** user exploration graph

$$\mathcal{G}_{i,t}^{(2)} = (\mathcal{U}, E, W_{i,t}^{(2)})$$

Edge weight **estimations**

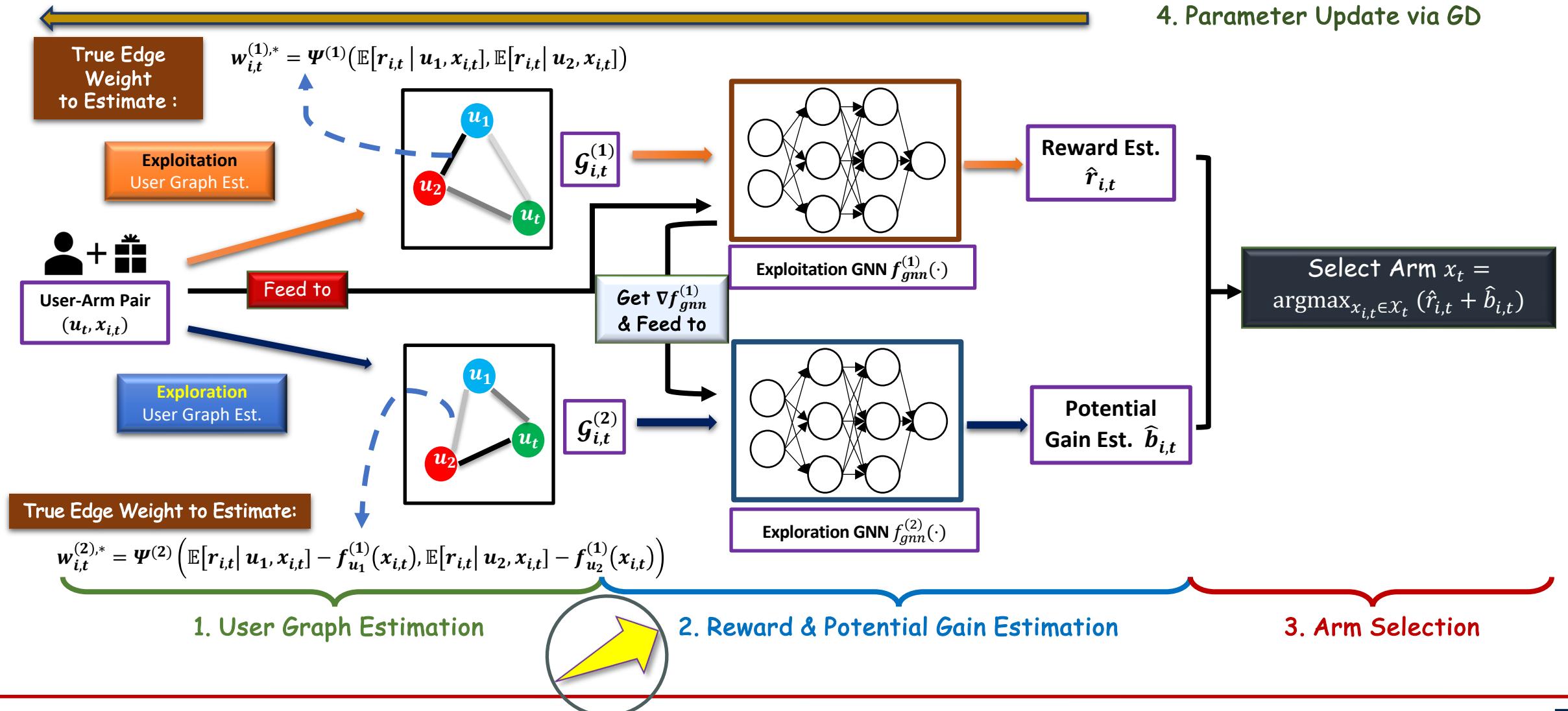
- For users  $u_1, u_2 \in \mathcal{U}$ , **estimated** edge weight

$$w_{i,t}^{(2)}(u_1, u_2) =$$

$$\Psi^{(2)} \left( f_{u_1}^{(2)}(\nabla f_{u_1}^{(1)}(x_{i,t})), f_{u_2}^{(2)}(\nabla f_{u_2}^{(1)}(x_{i,t})) \right)$$

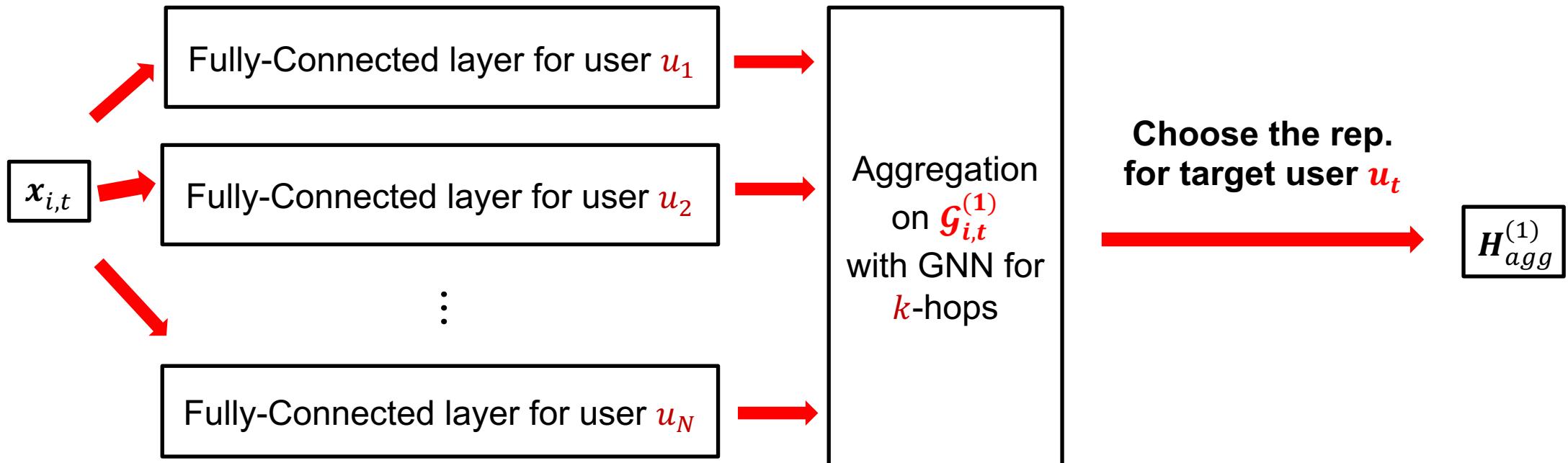
**Estimated Potential Gain**

# GNB: Framework Overview



# GNB: Aggregation on User Exploitation Graph

- ❑ For each arm  $x_{i,t} \in \mathcal{X}_t$ , reward estimation with estimated user exploitation graph  $\mathcal{G}_{i,t}^{(1)}$ .
- Given target user  $u_t$ , obtain **User-specific Arm Representation**  $H_{agg}$ :

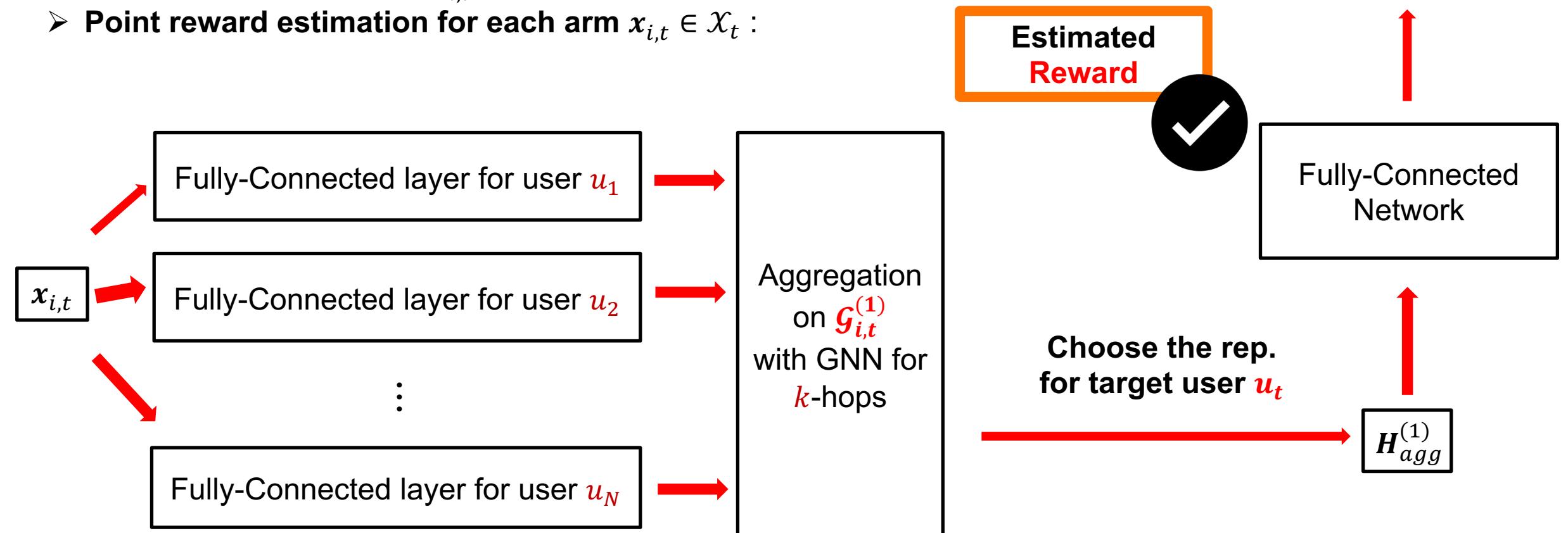


# GNB: Arm Reward Estimation

- For each arm  $x_{i,t} \in \mathcal{X}_t$ , reward estimation with estimated user exploitation graph  $\mathcal{G}_{i,t}^{(1)}$ .

➤ Point reward estimation for each arm  $x_{i,t} \in \mathcal{X}_t$ :

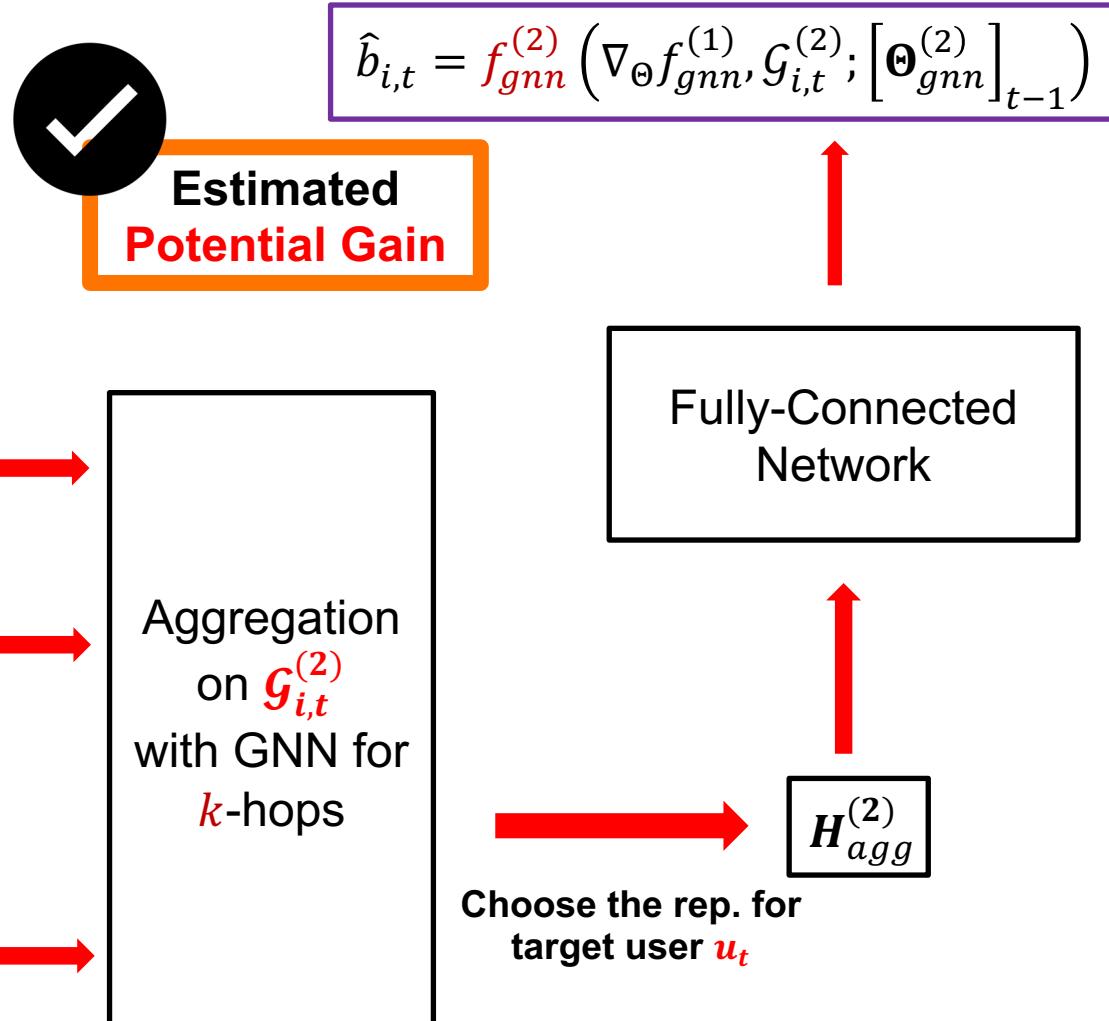
$$\hat{r}_{i,t} = f_{gnn}^{(1)}(x_{i,t}, \mathcal{G}_{i,t}^{(1)}; [\Theta_{gnn}^{(1)}]_{t-1})$$



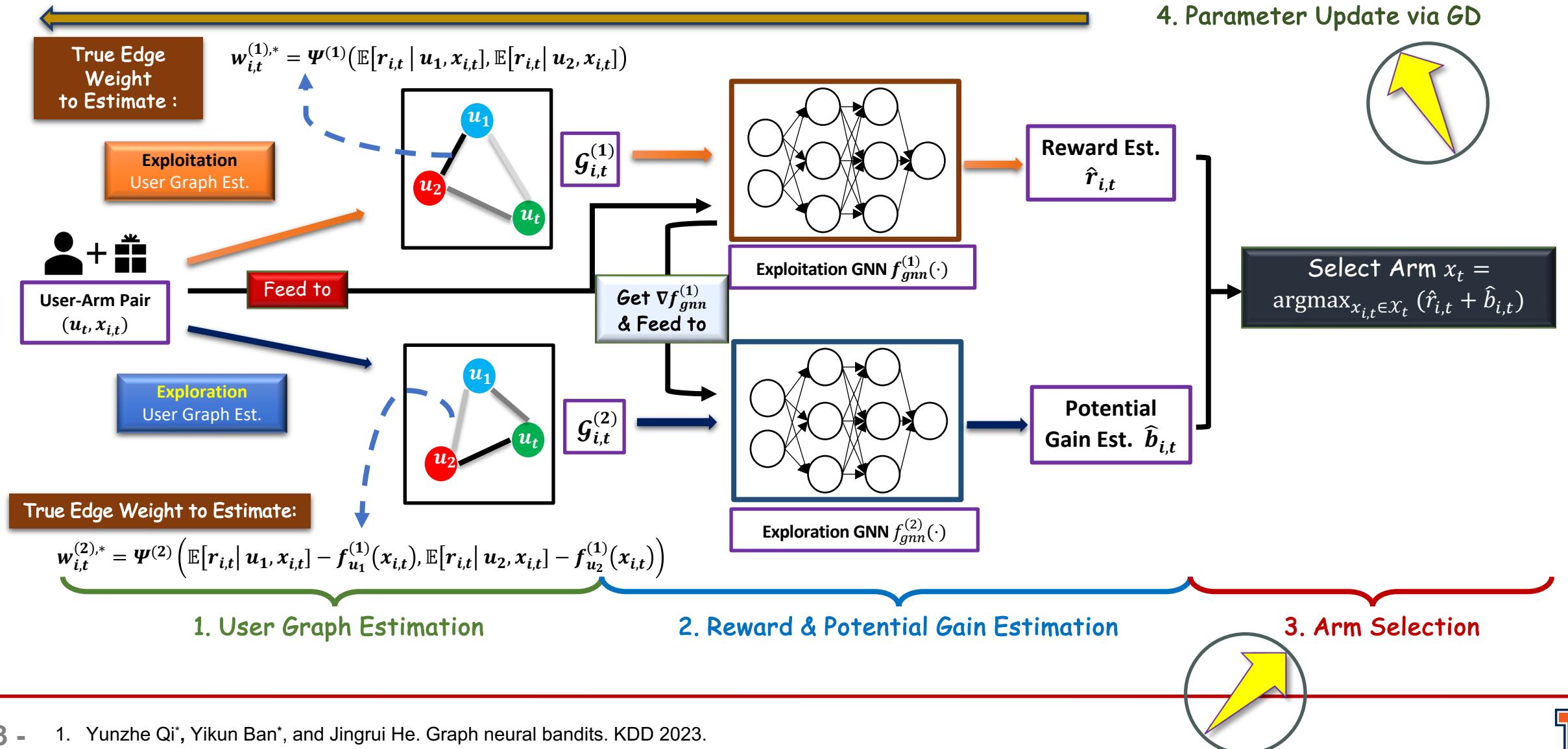
# GNB: Potential Gain Estimation

- For each arm  $x_{i,t} \in \mathcal{X}_t$ , reward estimation with estimated user exploration graph  $\mathcal{G}_{i,t}^{(2)}$ .

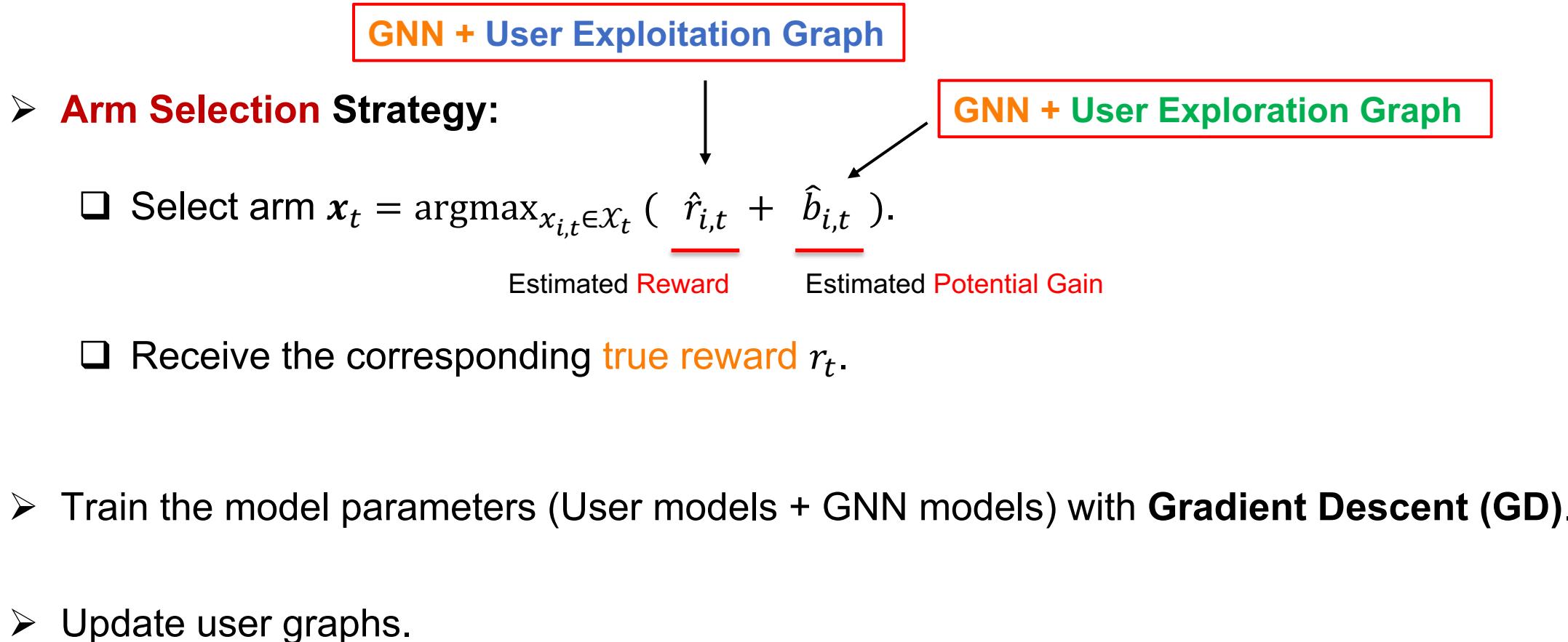
➤ Potential gain estimation for each arm  $x_{i,t} \in \mathcal{X}_t$ :



# GNB: Framework Overview



# GNB: Arm Selection & Training



# GNB: Theoretical Analysis

➤ **Pseudo regret for  $T$  rounds:**

$$R(T) = \sum_{t=1}^T \mathbb{E}[(r_t^* - r_t)]$$

➤ Given **sufficiently large network width  $m$**  (over-parameterization), under **mild assumptions**, with the probability at least  $1 - \delta$ :

$$R(T) \leq \sqrt{T} \cdot \left( O(L\xi_L) \cdot \sqrt{2 \log\left(\frac{Tn \cdot a}{\delta}\right)} \right) + \sqrt{T} \cdot O(L) + O(\xi_L) + O(1).$$

where  $n$  is the number of users,  $a$  is the number of arms in each round, and  $T$  is the number of rounds.

**Remarks:**

- Achieves the regret bound of  $\mathcal{O}(\sqrt{T \log(nT)})$ .
  - Existing works with **user clustering** need  $\mathcal{O}(\sqrt{nT \log(T)})$  for user collaboration modeling.
- Free of the terms  $d$ 
  - $d$  (arm context dimension, common in **linear bandit** works)

1. Shuai Li, et al. 2019. Improved Algorithm on Online Clustering of Bandits. In IJCAI. 2923–2929.  
 2. Shuai Li, et al. 2016. Collaborative filtering bandits. In SIGIR. 539–548.  
 3. Dongruo Zhou, et al. 2020. Neural contextual bandits with ucb-based exploration. In ICML. 11492–11502.  
 4. Yikun Ban, et al. 2022. Neural Collaborative Filtering Bandits via Meta Learning. arXiv preprint arXiv:2201.13395 (2022).

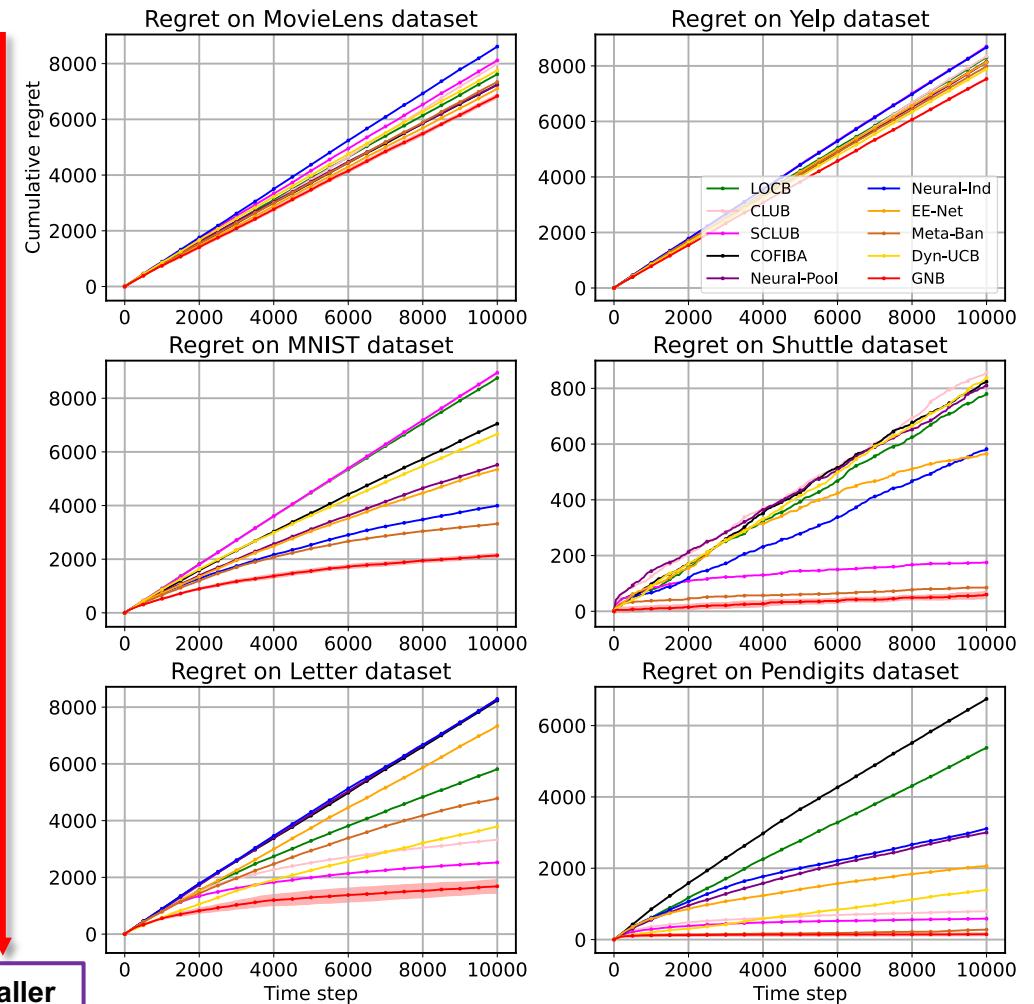
# Experiments: Real Data Sets

## ➤ Experiment settings:

- ❑ Under **online recommendation** settings, we evaluate the proposed GNB framework on **six** real data sets with different specifications.
- ❑ We include **nine** state-of-the-art related algorithms as the baselines, including both linear and neural algorithms.

## ➤ Summary of experiment results:

- ❑ **Neural algorithms** generally perform better than **linear ones**, with the representation power of neural networks.
- ❑ GNB can generally achieve the **best performance** against the strong baselines.



Smaller  
= Better

1. Shuai Li, et al. 2019. Improved Algorithm on Online Clustering of Bandits. In IJCAI. 2923–2929.
2. Shuai Li, et al. 2016. Collaborative filtering bandits. In SIGIR. 539–548.
3. Dongruo Zhou, et al. 2020. Neural contextual bandits with ucb-based exploration. In ICML. 11492–11502.
4. Yikun Ban, et al. 2022. Neural Collaborative Filtering Bandits via Meta Learning. arXiv preprint arXiv:2201.13395 (2022).
5. Trong T Nguyen and Hady W Lauw. 2014. Dynamic clustering of contextual multi-armed bandits. In CIKM. 1959–1962.
6. Claudio Gentile, et al. 2014. Online clustering of bandits. In ICML. 757–765.
7. Yikun Ban and Jingrui He. 2021. Local clustering in contextual multi-armed bandits. 2021. In WWW. 2335–2346.
8. Yikun Ban, et al. 2022. EE-Net: Exploitation-Exploration Neural Networks in Contextual Bandits. In ICLR.



## Introduction

- Background & Motivations
- Challenges



## Online Clustering of Bandits

- Clustering of Linear Bandits
- Clustering of Neural Bandits



## Graph Bandit Learning with Collaboration

- User side: Graph Neural Bandits
- Arm side: Neural Bandit with Arm Group Graph
- Other Scenarios: Bandit Learning with Graph Feedback & Online Graph Classification with Neural Bandit

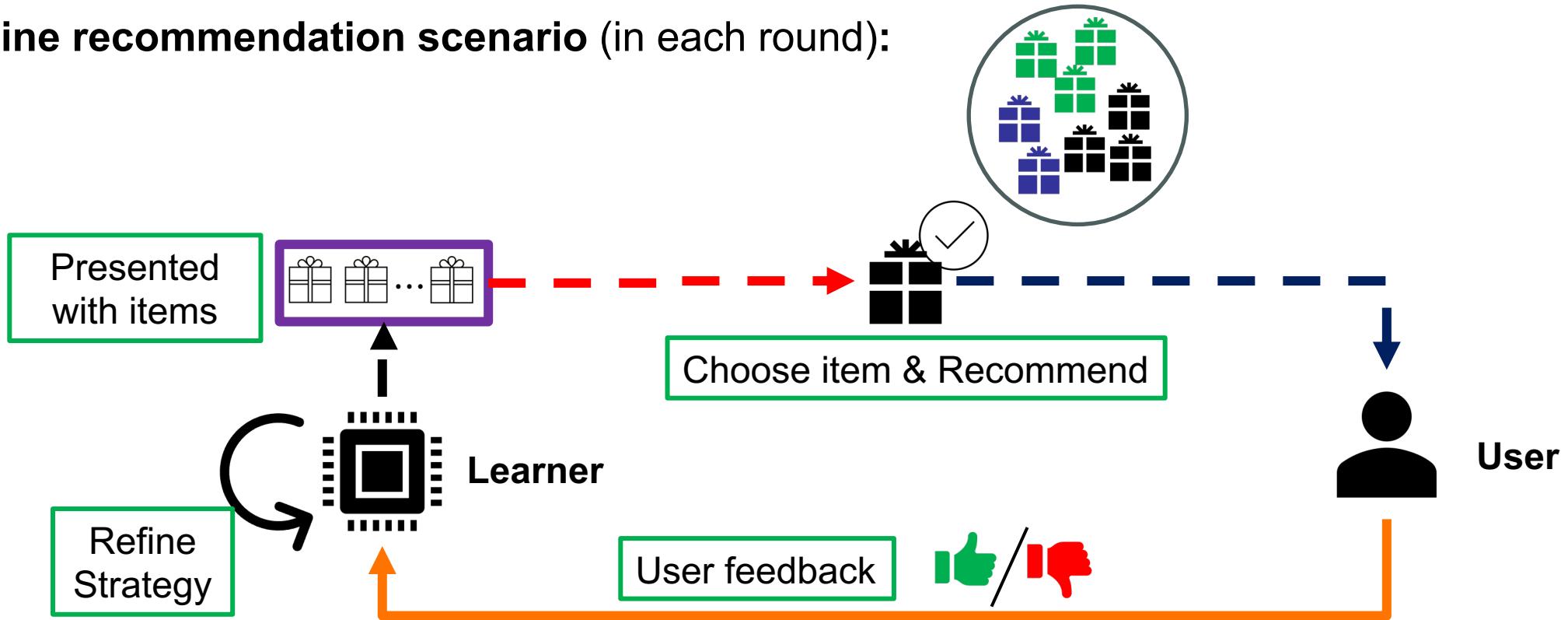


## Bandits for Combo Recommendation

- Multi-facet Contextual Bandits

# Online Recommendation with Arm Group Information

- Online recommendation scenario (in each round):



Leverage the available arm group information can help improve recommendation quality.

# Arm Group Information



➤ The **group (category) information** for arms (items) is commonly accessible:

□ **Media contents:**

- Music, Movies (grouped by **genres**)

□ **Text contents:**

- Articles (grouped by **literary styles**)

□ **E-commerce:**

- Restaurants (grouped by **cuisine types**)

□ **Etc.**

No existing **MAB** method trying to directly leverage the **available arm group information**.

# Formal Problem Definition

## ➤ Arm Groups:

- Assume a fixed pool  $\mathcal{C}$  of  $|\mathcal{C}| = N_c$  **arm groups**.
- Each **arm group**  $c \in \mathcal{C}$  (e.g., movie genre) relates to an arm distribution  $\mathcal{D}_c$ .

## ➤ For each round $t \in [T]$ :

- Receive a set of arms  $\mathcal{X}_t$ , and the corresponding set of **arm groups**  $\mathcal{C}_t \subseteq \mathcal{C}$ .

- $\mathcal{X}_t = \left\{ \mathbf{x}_{c,t}^{(i)} \in \mathbb{R}^{d_x}, \text{ (e.g., } \text{[ ])} \right\}_{c \in \mathcal{C}_t, i \in [n_{c,t}]}$
- $\mathbf{x}_{c,t}^{(i)} \sim \mathcal{D}_c$

- Reward  $r_{c,t}^{(i)} = h(W^*, \mathbf{x}_{c,t}^{(i)}) + \epsilon_{c,t}^{(i)}$ .
- Unknown affinity matrix for arm groups:  
 $W^* \in \mathbb{R}^{N_c \times N_c}$

- Learner **chooses** arm  $x_t \in \mathcal{X}_t$ .

## ➤ Objective: Minimizing Pseudo Regret

$$\begin{aligned}
 R(T) &= \sum_{t=1}^T \mathbb{E}[(r_t^* - r_t)] \\
 &= \sum_{t=1}^T \underbrace{|h(W^*, x_t^*) - h(W^*, x_t)|}_{\text{Optimal arm}} \quad \underbrace{|h(W^*, x_t^*) - h(W^*, x_t)|}_{\text{Chosen arm}}
 \end{aligned}$$

# Modeling with Arm Group Graph (AGG)

➤ Apply Arm Group Graph (AGG) to model **arm group correlations**:

➤ In round  $t \in [T]$  :

- Undirected graph  $\mathcal{G}_t = (V, E, W_t)$ 
  - $V$ : set of nodes
    - **Each node is an arm group**  $c \in \mathcal{C}$ ,
    - $N_c$  nodes in total
  - $E = \{e(c, c')\}_{c, c' \in \mathcal{C}}$ : set of edges
  - $W_t$ : Set of **edge weights**
- Arm group correlations are modeled by the edge weights from set  $W_t$ .

➤ True reward:  $r_{c,t}^{(i)} = h(W^*, x_{c,t}^{(i)}) + \epsilon_{c,t}^{(i)}$ .

- Unknown affinity matrix:  $W^* \in \mathbb{R}^{N_c \times N_c}$

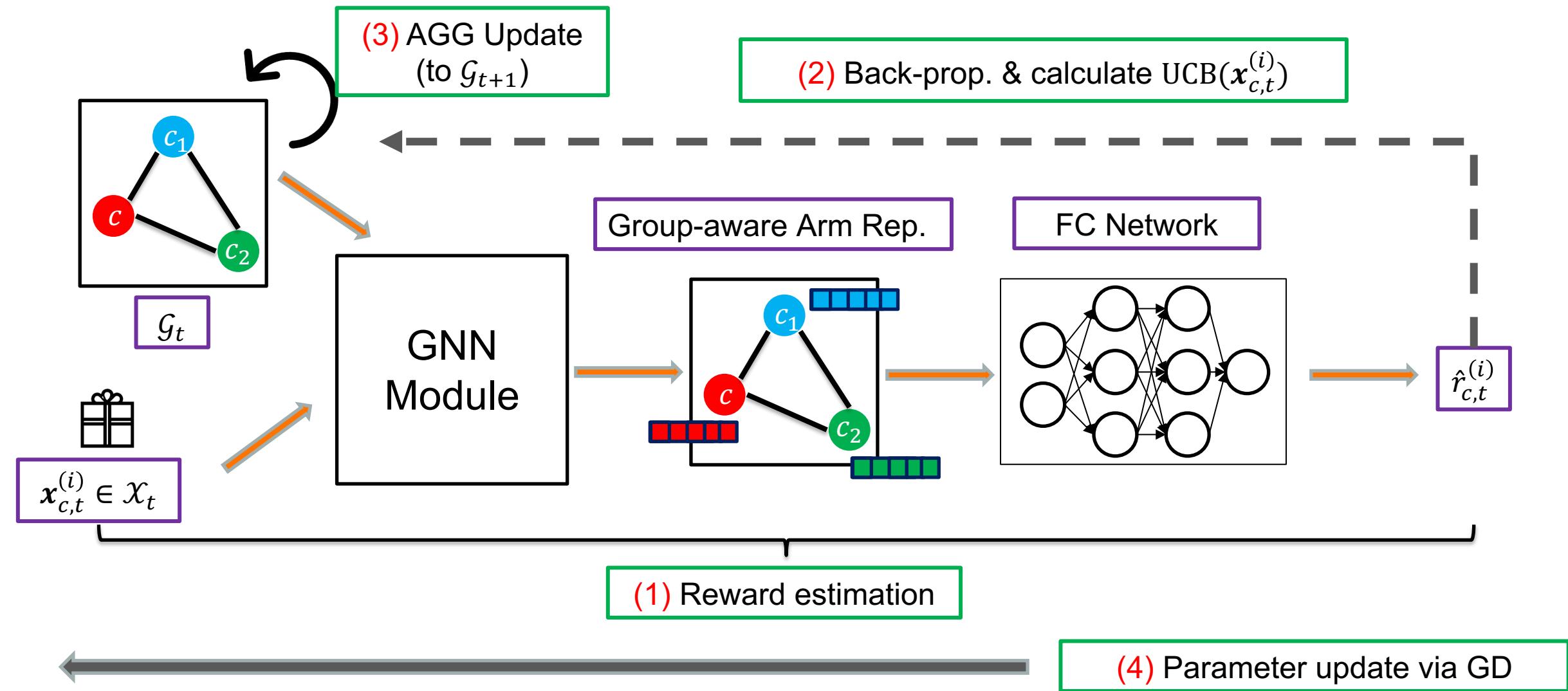
➤ Unknown true graph:  $\mathcal{G}^*$

- Encoding the true arm group correlations.

○ In substitution for  $W^*$ :

$$r_{c,t}^{(i)} = h(\mathcal{G}^*, x_{c,t}^{(i)}) + \epsilon_{c,t}^{(i)}.$$

# Proposed Framework: AGG-UCB



# AGG-UCB: Arm Group Graph Estimation



- **Recall for Arm Groups:**

- Assume a fix pool  $\mathcal{C}$  of  $|\mathcal{C}| = N_c$  arm groups.
- Each group  $c \in \mathcal{C}$  has a context distribution  $\mathcal{D}_c$ .

- **Definition: True edge weights**

- For  $c, c' \in \mathcal{C}$ , **true** edge weight in  $\mathcal{G}^*$ :

- $w^*(c, c') = \exp\left(\frac{-\left\|\mathbb{E}_{x \sim \mathcal{D}_c}[\phi(x)] - \mathbb{E}_{x' \sim \mathcal{D}_{c'}}[\phi(x')]\right\|^2}{\sigma_s}\right)$

- $\phi(\cdot)$ : kernel mapping function

- **Arm Group Graph estimation:**

- Estimated edge weight in round  $t \in [T]$ :

- $w_t(c, c') = \exp\left(\frac{-\|\Psi_t(\mathcal{D}_c) - \Psi_t(\mathcal{D}_{c'})\|^2}{\sigma_s}\right)$

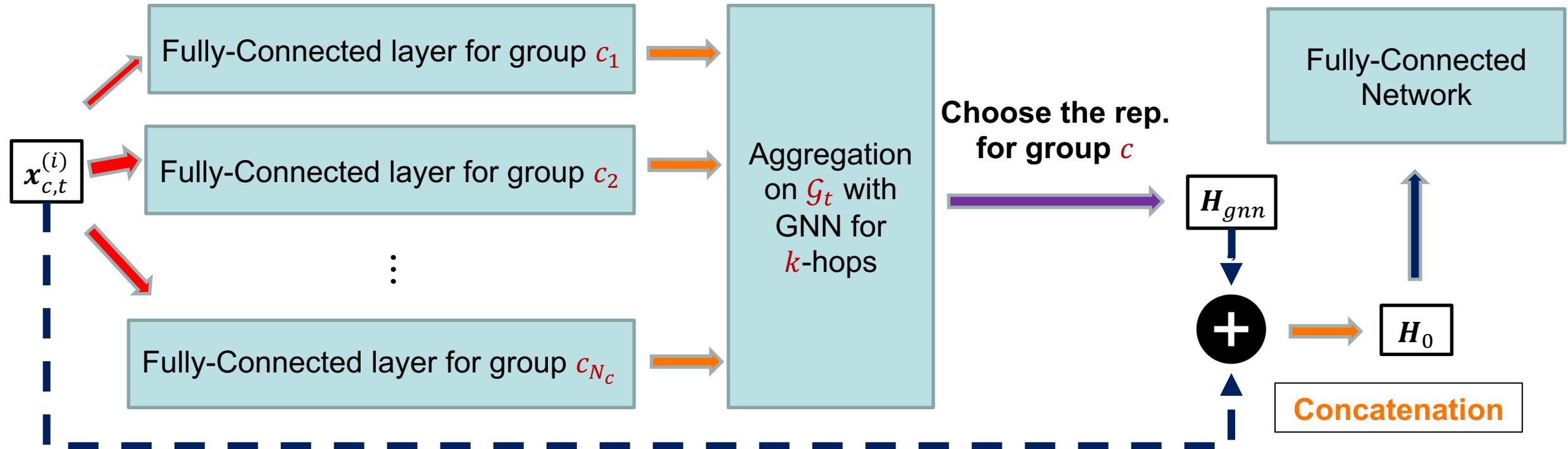
- **Kernel Mean Embedding** [1]:  $\Psi_t(\mathcal{D}_c)$

- $w_t(c, c') \in W_t$ : **weight** for edge  $e(c, c') \in E$  in graph  $\mathcal{G}_t$

# AGG-UCB: Arm Reward Estimation

☐ Reward estimation with estimated  $\mathcal{G}_t$ .

➤ Point reward estimation for each arm  $x_{c,t}^{(i)} \in \mathcal{X}_t$  :



➤ **Exploration with Upper Confidence Bound (UCB):**

- The UCB( $\cdot$ ) satisfies :

$$\mathbb{P}\left(\left|\underbrace{f\left(\mathcal{G}_t, \mathbf{x}_{c,t}^{(i)}; \boldsymbol{\Theta}_{t-1}\right)}_{\text{Reward Est.}} - \underbrace{h\left(\mathcal{G}^*, \mathbf{x}_{c,t}^{(i)}\right)}_{\text{Exp. Reward}}\right| > \mathbf{UCB}\left(\mathbf{x}_{c,t}^{(i)}\right)\right) \leq \delta$$

➤ **Arm Selection Strategy:**

- Select arm  $\mathbf{x}_t = \operatorname{argmax}_{\mathbf{x}_{c,t}^{(i)} \in \mathcal{X}_t} \left( \hat{r}_{c,t}^{(i)} + \gamma \cdot \mathbf{UCB}\left(\mathbf{x}_{c,t}^{(i)}\right) \right)$
- Receive the corresponding true reward  $r_t$

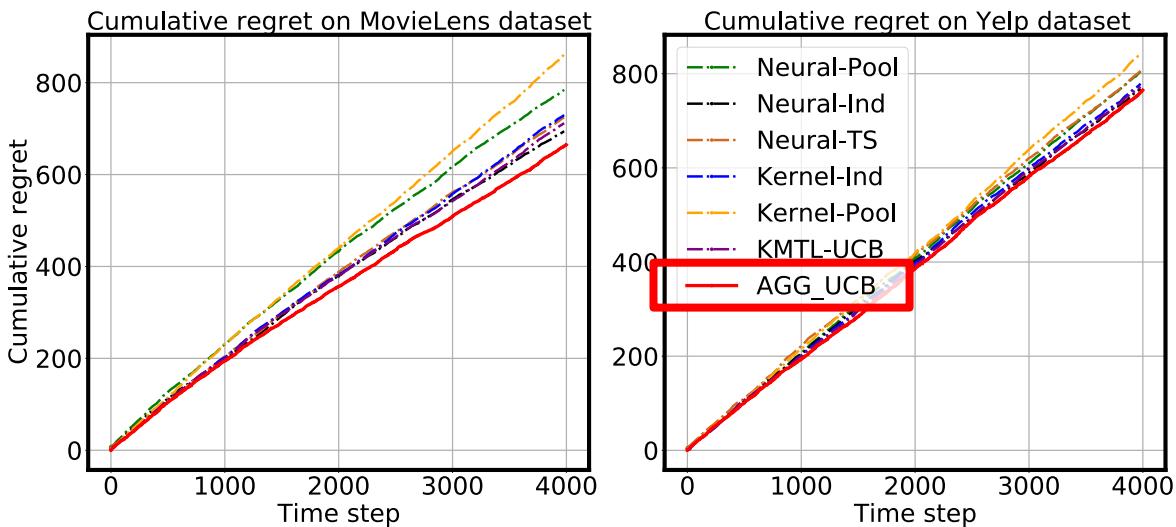
# Theoretical and Empirical Results

- **Theoretical:** Given sufficiently large network width  $m$ , with the probability at least  $1 - \delta$ :

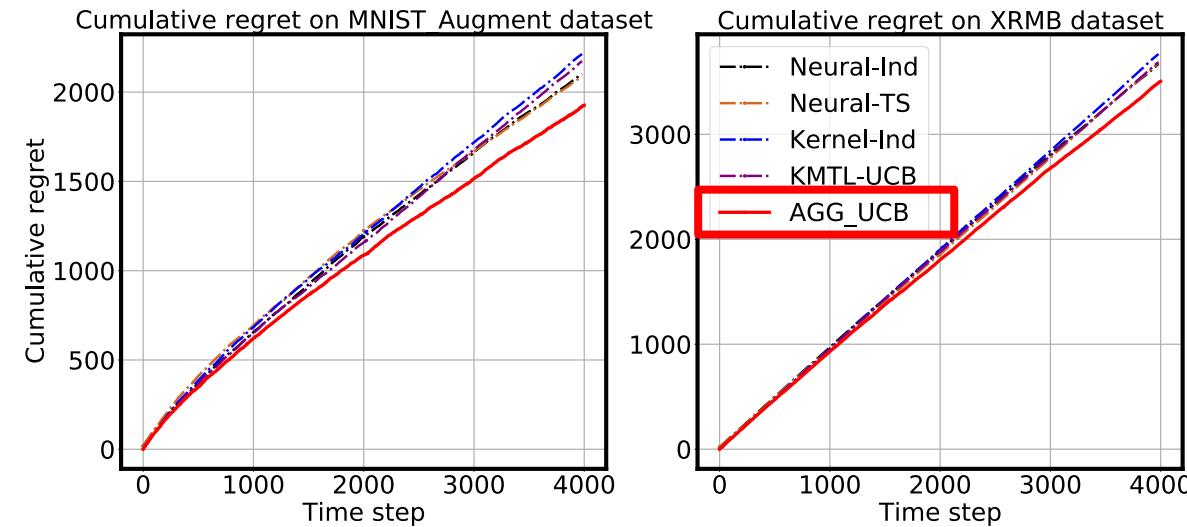
$$R(T) \leq 2 \cdot (2B_4\sqrt{T} + 2 - B_4) + 2\sqrt{2\tilde{d}T \log(1 + T/\lambda)} + 2T \\ \cdot (\sqrt{\lambda}S + \sqrt{1 - 2\log(\delta/2)} + (\tilde{d}\log(1 + T/\lambda)))$$

Achieves the regret bound of  
 $\mathcal{O}(\tilde{d}\sqrt{T\log^2(T) \cdot \log(N_c)})$

- **Empirical:** Leveraging arm group information with AGG-UCB can improve good performances.



**Figure 1:** Cumulative regrets on recommendation data sets



**Figure 2:** Cumulative regrets on Classification data sets



# Graph Bandit Learning: Other Scenarios

## 1. Bandit Learning with Graph Feedback [1]:

- Arms are nodes on a graph  $G = (V, E)$ . In each round  $t \in [T]$ , the learner chooses one node  $I_t \in V$ .
- Learner observes **reward for chosen arm**  $I_t$ , and **neighbor rewards** (e.g., out-neighbors in a directed graph).
- **Objective:** minimizing the cumulative pseudo regret over  $T$  rounds.

## 2. Optimal Graph Search with Bandit [2]:

- In each round  $t \in [T]$ , the learner aims to choose **one graph**  $G_t \in \mathcal{G}$ , from a **fixed** graph domain  $\mathcal{G}$ . Reward generated by  $r_t = h(G_t) + \epsilon_t$ .
- **Objective:** minimizing the cumulative pseudo regret over  $T$  rounds.
- **Application example:** material designing, drug search.



## Introduction

- Background & Motivations
- Challenges



## Online Clustering of Bandits

- Clustering of Linear Bandits
- Clustering of Neural Bandits



## Graph Bandit Learning with Collaboration

- User side: Graph Neural Bandits
- Arm side: Neural Bandit with Arm Group Graph
- Other Scenarios: Bandit Learning with Graph Feedback & Online Graph Classification with Neural Bandit

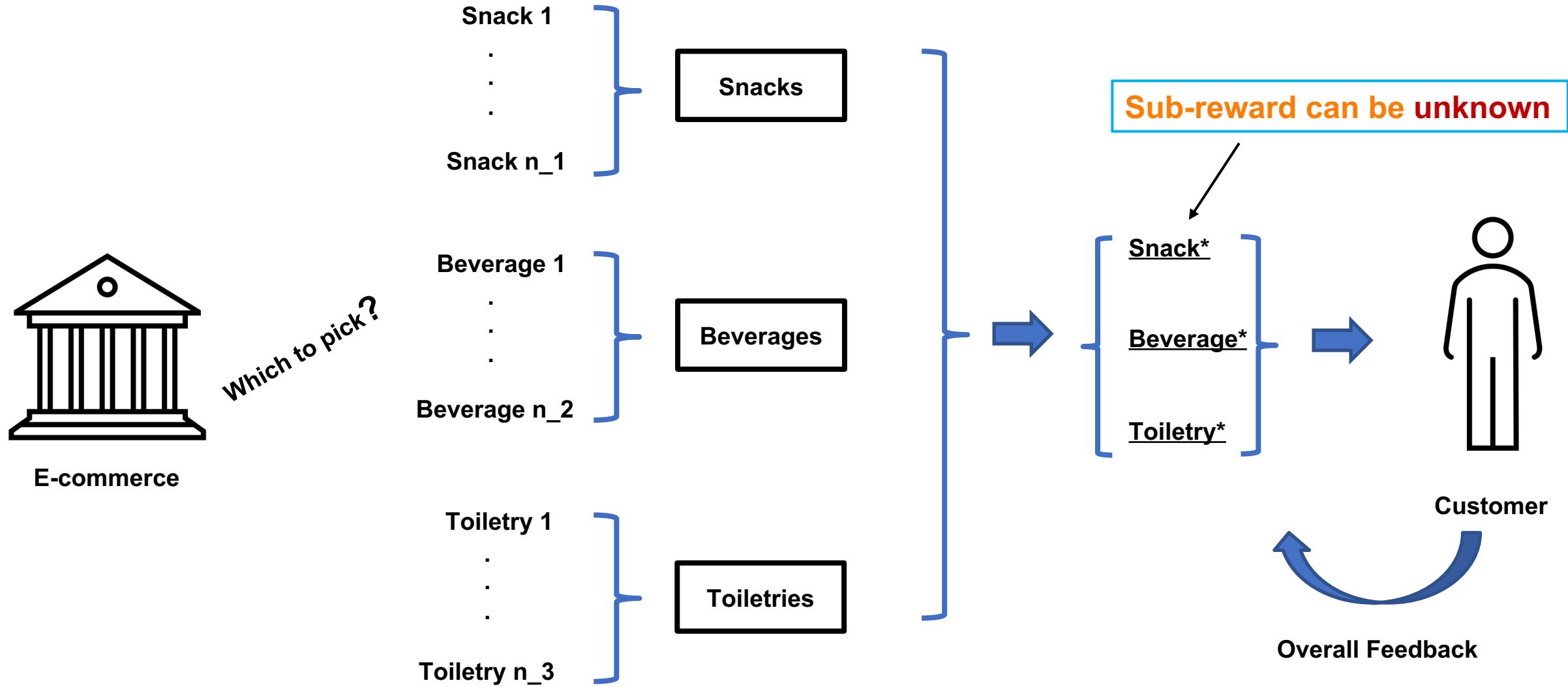


## Bandits for Combo Recommendation

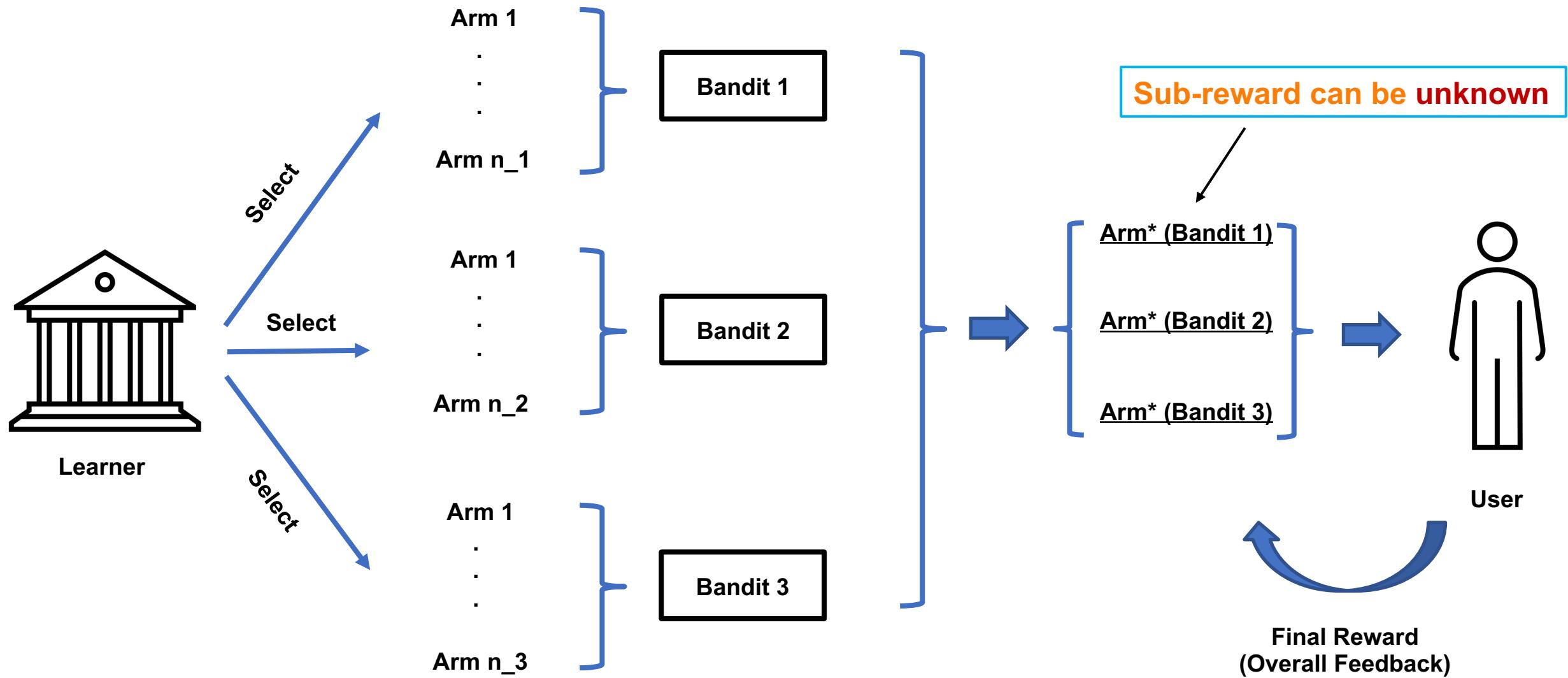
- Multi-facet Contextual Bandits



# Motivated Case: Promotion Campaign



# Application: Multi-facet Recommendation with Neural Bandits



# Formal Definition of Multi-facet Contextual Bandit : In Round $t$

- Sub-reward Functions (unknown):

$$r_t^1 = h_1(x_t^1) \text{ (Linear or Non-linear)}$$

$$r_t^2 = h_2(x_t^2)$$

⋮

$$r_t^K = h_K(x_t^K)$$

**Assumption1:**  $h_k(\mathbf{0}) = 0, \forall k$

- Final Reward Function (unknown):

$$R_t = H(r_t^1, r_t^2, \dots, r_t^K) + \epsilon_t \quad \leftarrow \text{Noise}$$

**Expectation:**  $H(X_t) = E[R_t|X_t] = H(r_t^1, r_t^2, \dots, r_t^K)$

**Assumption2:**  $H$  is  $\bar{C}$ - Lipschitz continuous.

- Evaluation Measure: Regret

$$\text{Reg} = E \left[ \sum_t (R_t^* - R_t) \right]$$

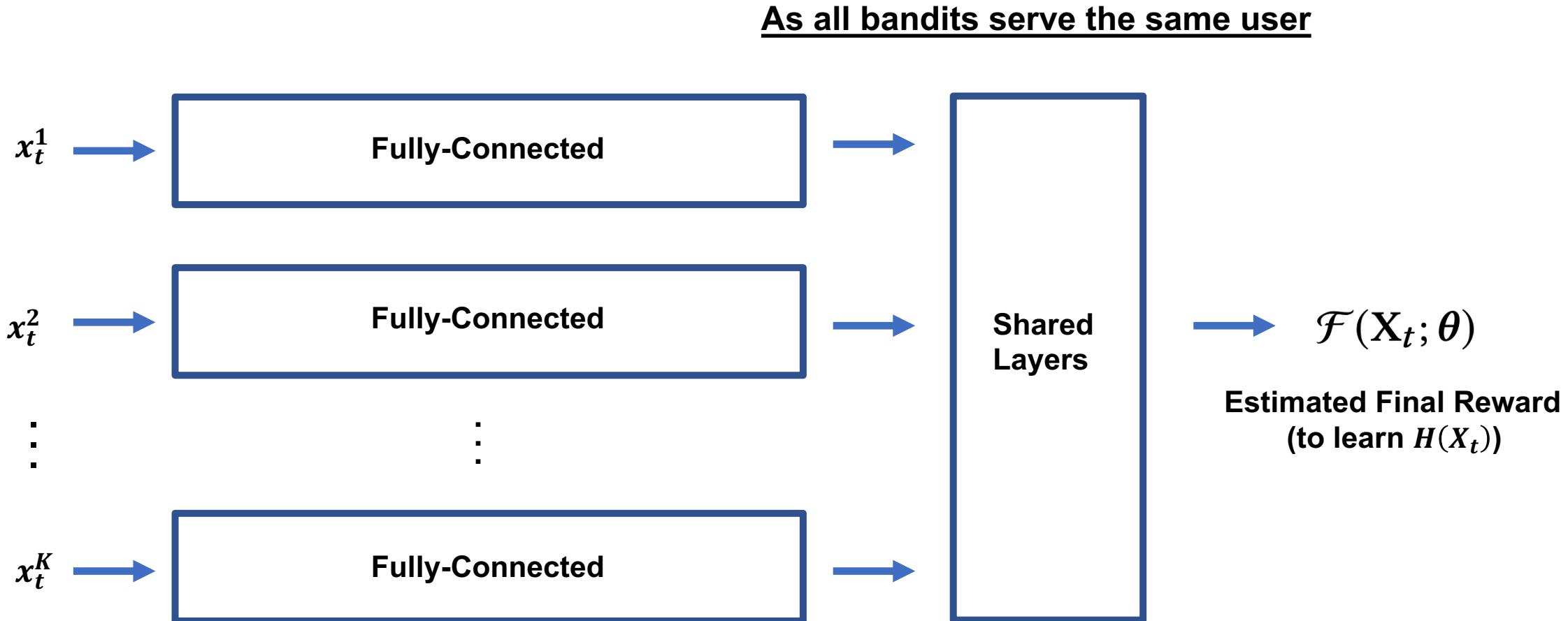
$$= \sum_t [H(X_t^*) - H(X_t)]$$

Optimal Final Reward

Received Final Reward

➤ Goal: Minimize the regret of T rounds.

# MuFasa: Exploitation (Neural Network Model)



# MuFasa: Exploration (Upper Confidence Bound)



➤ **UCB:**  $\mathbb{P}(|\mathcal{F}(\mathbf{X}_t; \theta_t) - \mathcal{H}(\mathbf{X}_t)| > \text{UCB}(\mathbf{X}_t)) \leq \delta,$

➤ **K selected arms are determined by:**

$$\mathbf{X}_t = \arg \max_{\mathbf{X}'_t \in S_t} (\mathcal{F}(\mathbf{X}'; \theta_t) + \text{UCB}(\mathbf{X}'_t)).$$

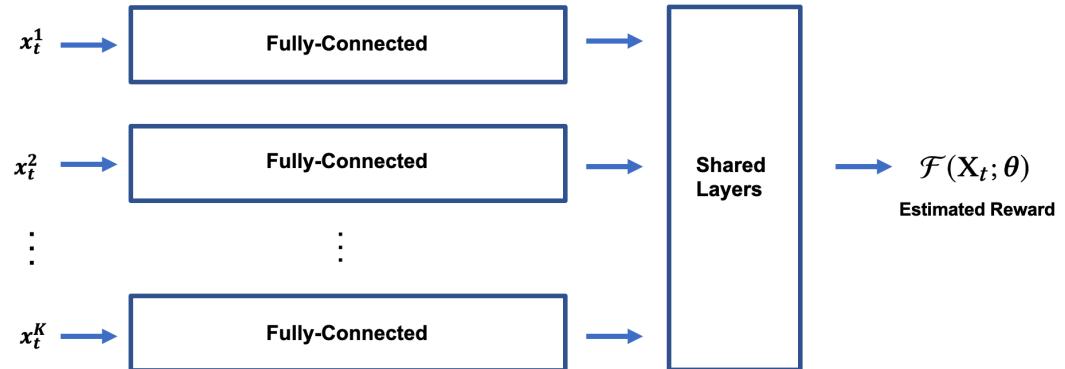
**Where**

$$S_t = \{(\mathbf{x}_t^1, \dots, \mathbf{x}_t^k, \dots, \mathbf{x}_t^K) \mid \mathbf{x}_t^k \in X_t^k, k \in [K]\},$$

**(all possible combinations of K arms)**

# MuFasa: Novel Upper Confidence Bound (UCB)

- With the assembled neural framework (MuFasa):



- With probability at least  $1 - \delta$ , the UCB holds

$$|\mathcal{F}(\mathbf{X}_t; \theta_t) - \mathcal{H}(\mathbf{X}_t)| \leq \bar{C} \sum_{k=1}^K \mathcal{B}^k + \mathcal{B}^F = UCB(\mathbf{X}_t), \text{ where}$$

$$\mathcal{B}^k = \gamma_1 \|g_k(\mathbf{x}_t^k; \theta_t^k)/\sqrt{m_1}\|_{A_t^{k-1}} + \gamma_2 \left(\frac{\delta}{k+1}\right) \|g_k(\mathbf{x}_t^k; \theta_0^k)/\sqrt{m_1}\|_{A_t^{k'-1}} + \gamma_1 \gamma_3 + \gamma_4$$

Error of facet-specific networks

$$\mathcal{B}^F = \gamma_1 \|G(\mathbf{f}_t; \theta_t^\Sigma)/\sqrt{m_2}\|_{A_t^{F-1}} + \gamma_2 \left(\frac{\delta}{k+1}\right) \|G(\mathbf{f}_t; \theta_0^\Sigma)/\sqrt{m_2}\|_{A_t^{F'-1}} + \gamma_1 \gamma_3 + \gamma_4$$

Error of shared network

# Regret Analysis

$$\begin{aligned}
 \mathbf{Reg} &= E \left[ \sum_t (\mathbf{R}_t^* - \mathbf{R}_t) \right] \\
 &= \sum_t [\mathbf{H}(\mathbf{X}_t^*) - \mathbf{H}(\mathbf{X}_t)]
 \end{aligned}$$

- After T rounds, with probability at least  $1 - \delta$ ,

$$\begin{aligned}
 \mathbf{Reg} \leq & (\bar{C}K + 1) \sqrt{T} 2 \sqrt{\tilde{P} \log(1 + T/\lambda) + 1/\lambda + 1} \\
 & \cdot \left( \sqrt{(\tilde{P} - 2) \log \left( \frac{(\lambda + T)(1 + K)}{\lambda\delta} \right)} + 1/\lambda + \lambda^{1/2}S + 2 \right) + 2(\bar{C}K + 1),
 \end{aligned}$$

- Achieve near-optimal regret bound  $\tilde{O}\left((K + 1)\sqrt{T}\right)$ , same as a single linear bandit  $\tilde{O}(\sqrt{T})$

# All Sub-rewards Available (Different Final Reward Function)

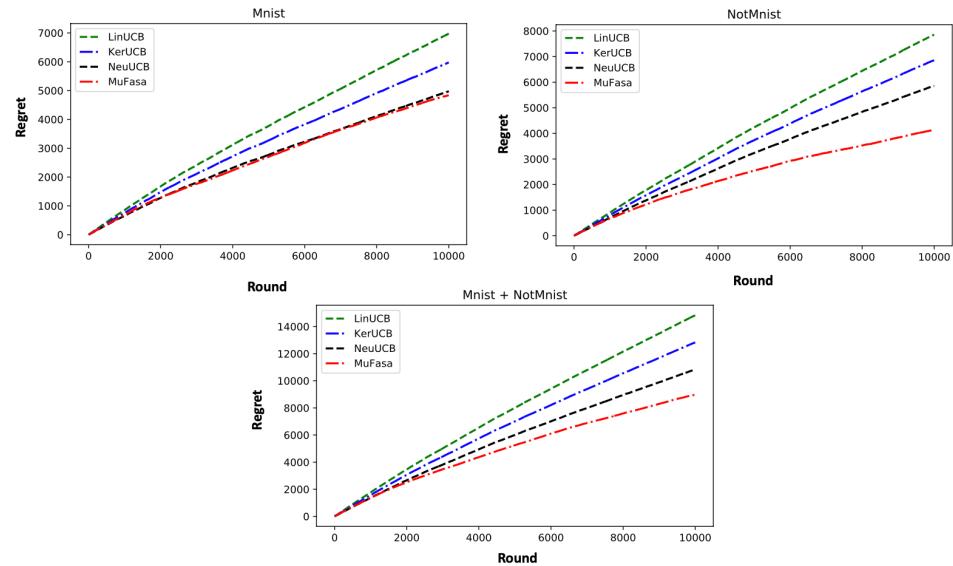


Figure: Regret comparison on Mnist+NotMnist with  $H_1$ .

$$H_1(\text{vec}(\mathbf{r}_t)) = r_t^1 + r_t^2$$

## Observation:

- Superiority of MuFasa is slightly higher on  $H_2$ , compared to  $H_1$ .

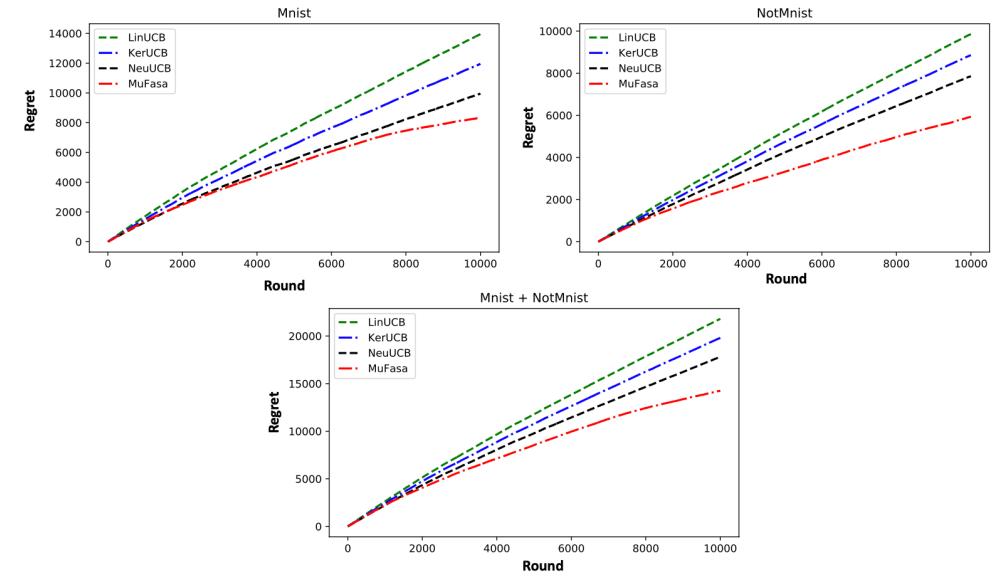


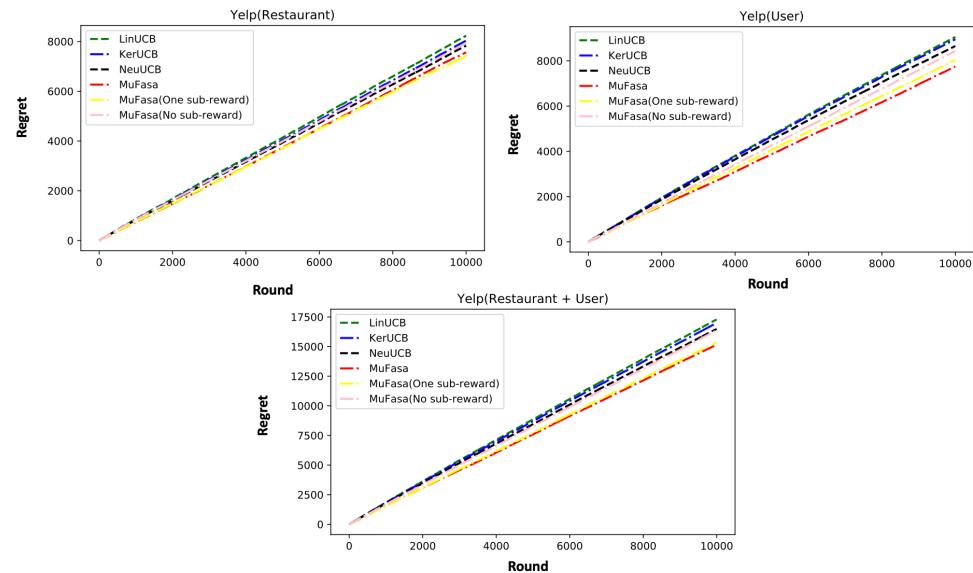
Figure: Regret comparison on Mnist+NotMnist with  $H_2$ .

$$H_2(\text{vec}(\mathbf{r}_t)) = 2r_t^1 + r_t^2.$$

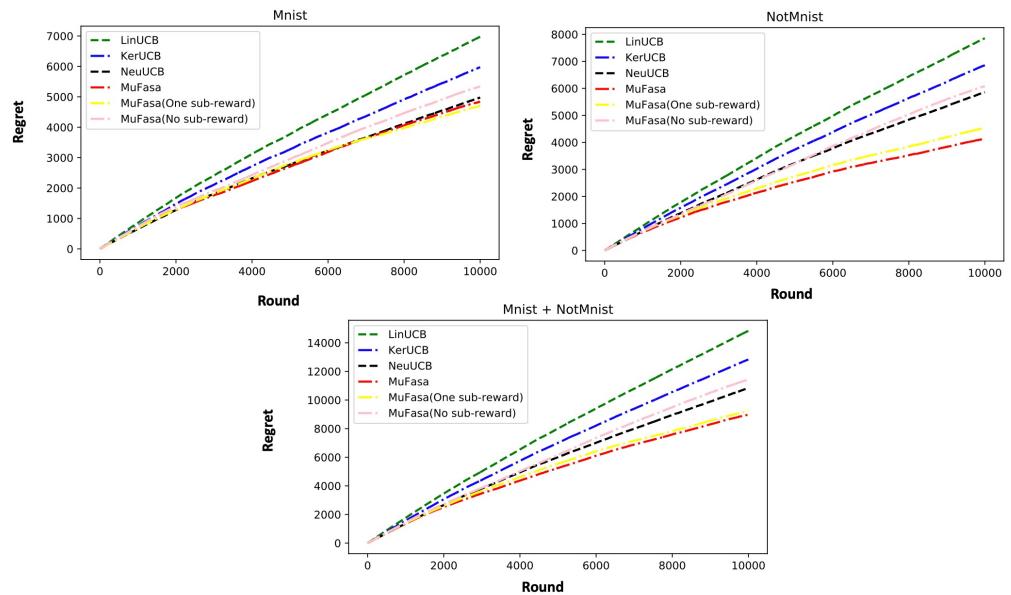
## Insights:

- MuFasa can select arms according to different weights of bandits  
(Bandit 1 has higher weight in  $H_2$ ).

# Partial Sub-rewards Available



**Figure: Regret comparison on Yelp with different reward availability.**

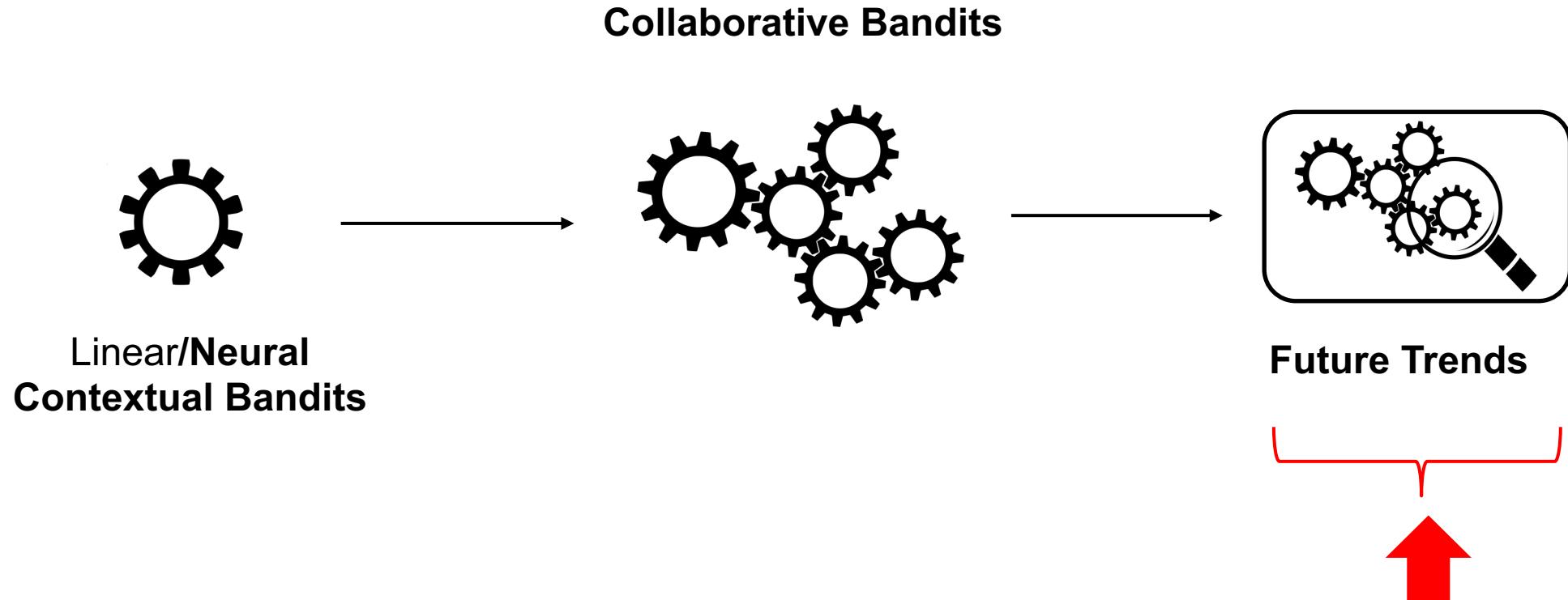


**Figure: Regret comparison on Mnist+NotMnist with different reward availability.**

## Observation:

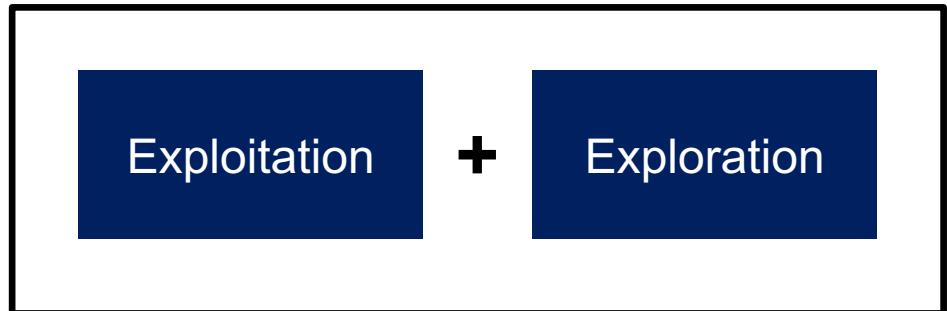
- With one sub-reward, MuFasa still outperforms all baselines.
- Without any sub-reward, MuFasa's performance is close to the best baseline.

# Roadmap



# Trustworthy Exploration: Transparency

**Q:** Can we have a **transparent** exploration with clear rationales and explanations?



➤ **Challenges:**

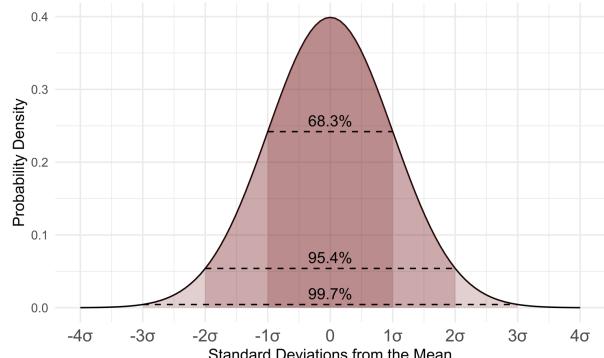
- More exploration models based on neural networks (**Black Box**).

**Black Box !**

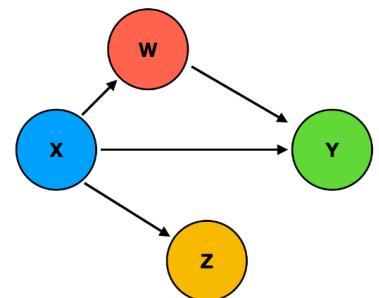
E.g. [Ban et al. ICLR 2022]

➤ **Future Directions:**

- Bayesian Bandits/RL.
- Causal Bandits/RL.



Statistics



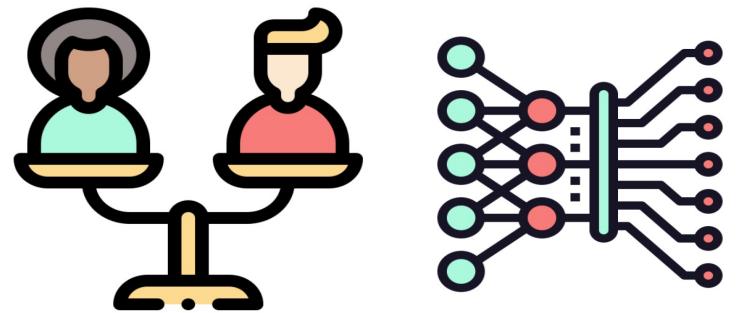
Causal Inference

# Trustworthy Exploration: Fairness

Q: How to ensure **fairness** in the context of exploration?

➤ **Challenges:**

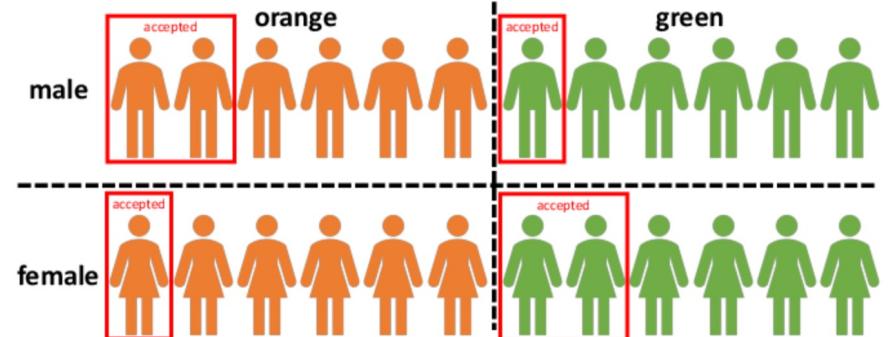
- Non-IID data.
- balance required between **exploration power** and **fairness**.



Group Fairness

➤ **Future Directions:**

- Derive fairness confidence interval for exploration.
- Fairness Regularization.



Group Fairness [1]

# Trustworthy Exploration: Privacy

**Q:** Can we have an exploration strategy preserving privacy?

➤ **Challenges:**

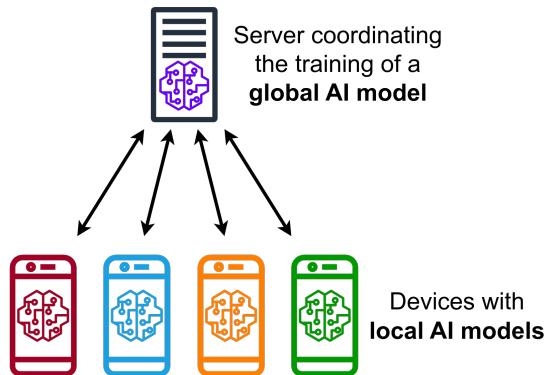
- ❑ Privacy-preserving exploration methods.



User Privacy

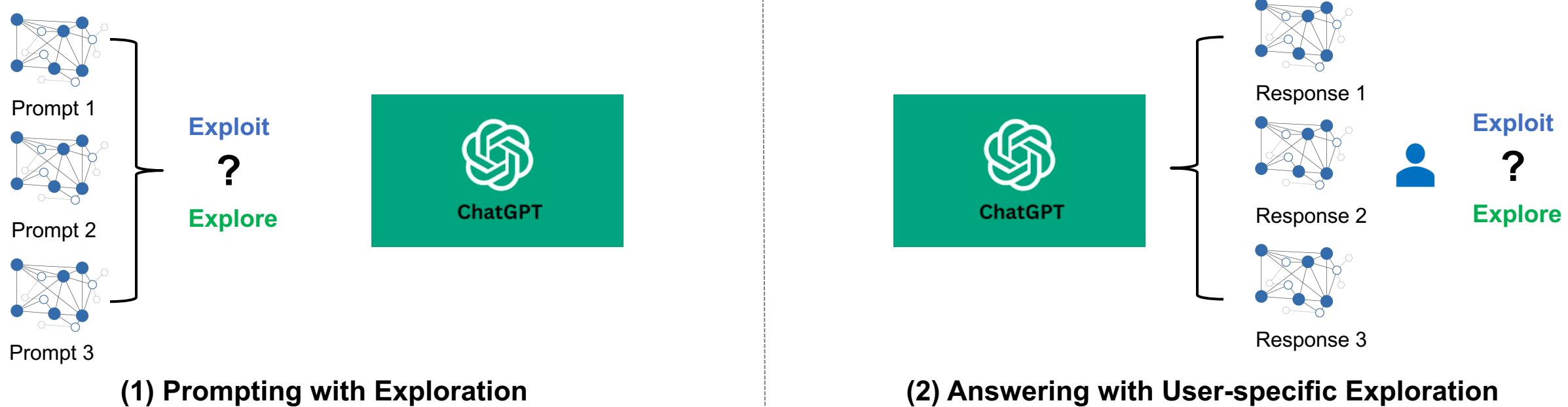
➤ **Future Directions:**

- ❑ Federated Bandits/RL.



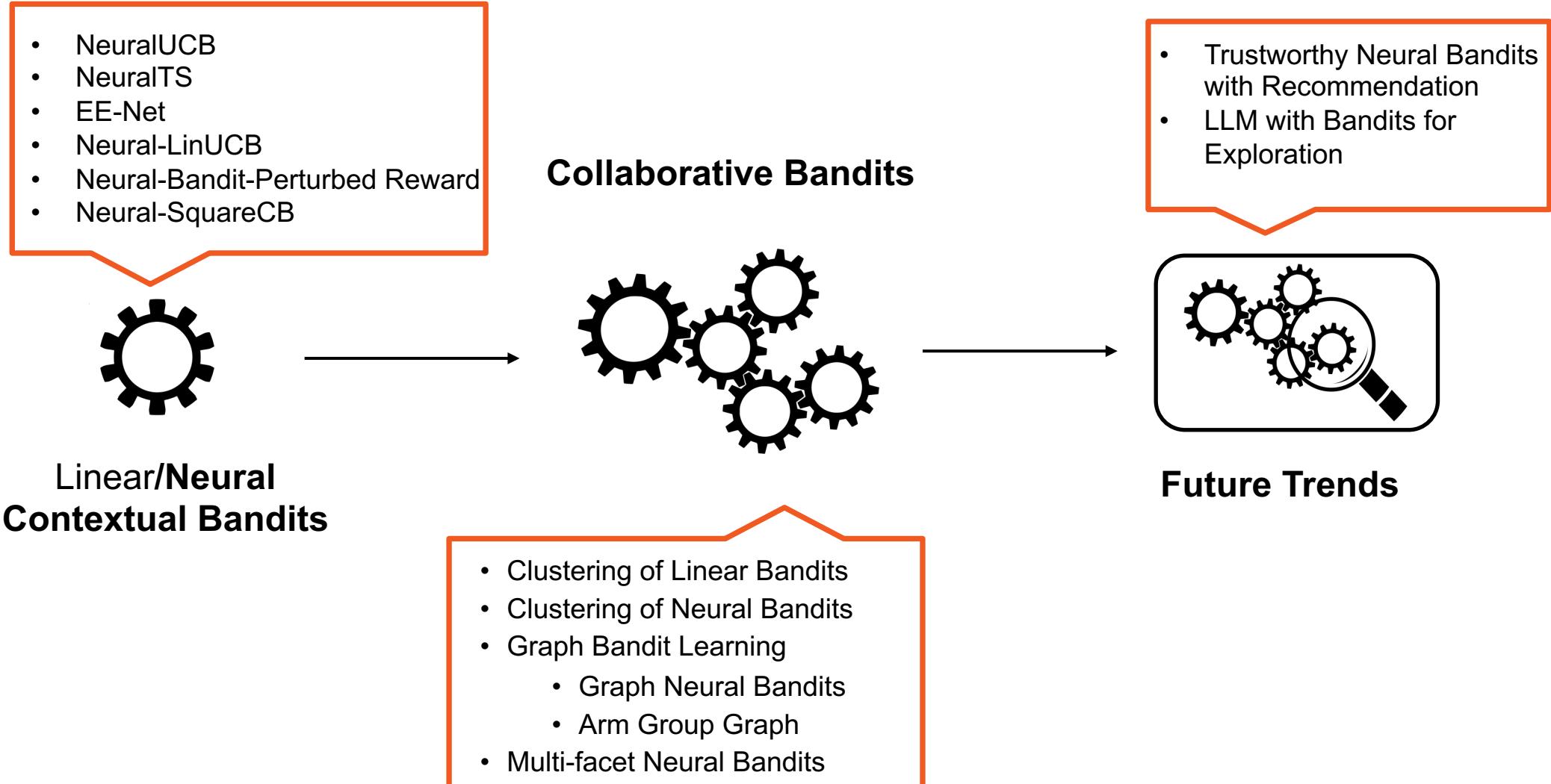
Federated Learning

# Customized Exploration: Large Language Model



1. Lin, Xiaoqiang, et al. "Use your INSTINCT: instruction optimization using neural bandits coupled with transformers." ICML 2024.
2. Chen, Zekai, et al. "Online Personalizing White-box LLMs Generation with Neural Bandits." *arXiv preprint arXiv:2404.16115* (2024).
3. Köpf, Andreas, et al. "Openassistant conversations-democratizing large language model alignment." NeurIPS 2023.
4. Rafailov, Rafael, et al. "Direct preference optimization: Your language model is secretly a reward model." NeurIPS 2023.
5. Zhang, Qingru, et al. "Platon: Pruning large transformer models with upper confidence bound of weight importance." ICML 2022.

# Roadmap





# Neural Contextual Bandits for Personalized Recommendation



**Yikun Ban**



**Yunzhe Qi**



**Jingrui He**

**University of Illinois Urbana-Champaign**

{Yikunb2, Yunzheq2, Jingrui}@illinois.edu

Time: 9:00 AM – 12:30 PM, 13 May 2024

Location: Virgo 1, Resorts World Sentosa Convention Centre, Singapore

Website: [www.banyikun.com/wwwtutorial/](http://www.banyikun.com/wwwtutorial/)

