13ème Atelier en Évaluation des Performances

Deep Q-Learning with Whittle Index for Contextual Restless Multi-Armed Bandits

Presented by

PhD student Ibtihal EL MIMOUNI

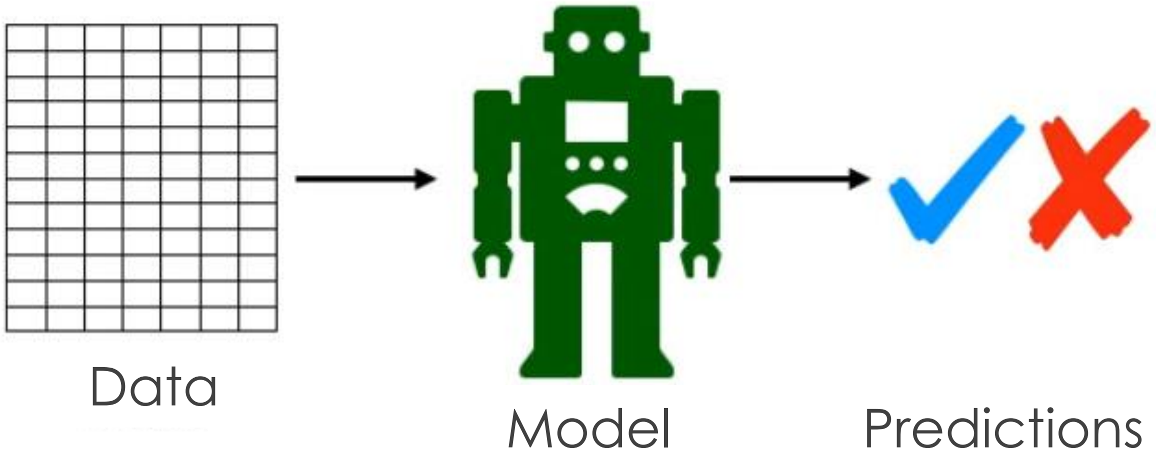
PhD supervised by

Dr Konstantin AVRACHENKOV & Mr Hervé BAILE

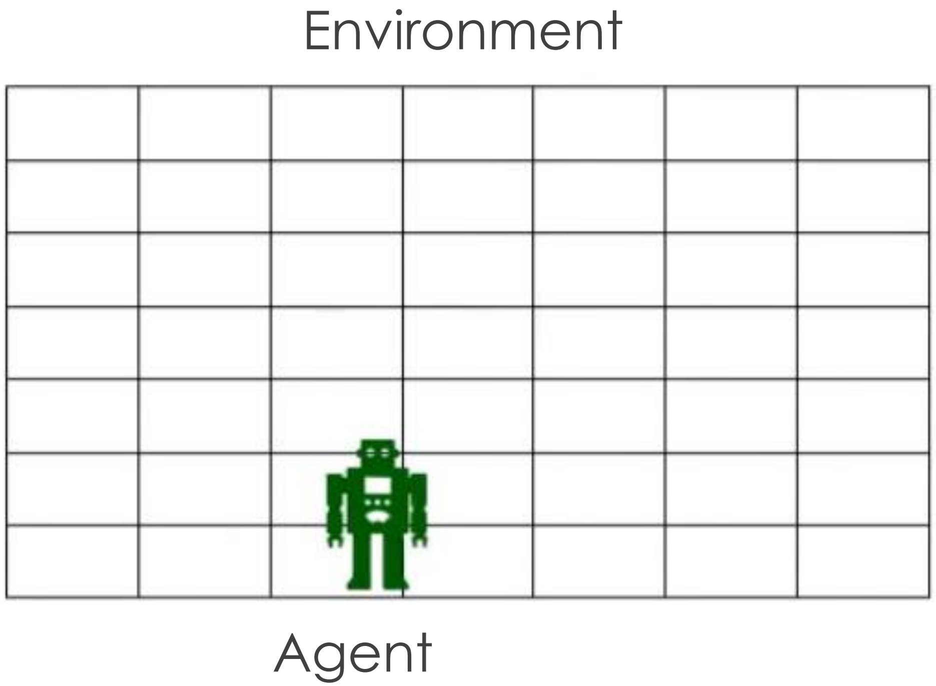
Tuesday, December 3rd 2024

INTRODUCTION
MACHINE LEARNING

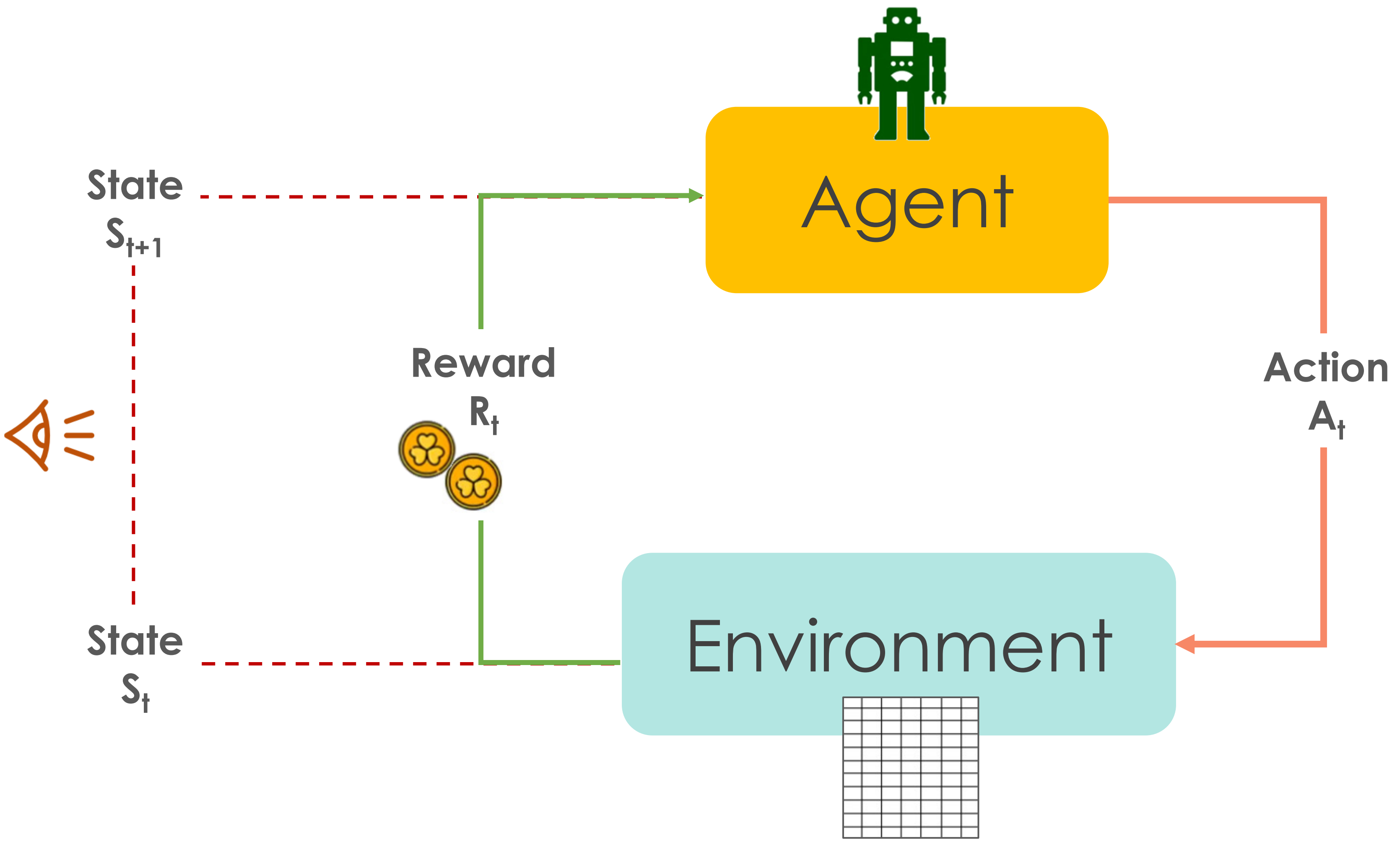
Predictive Machine Learning (ML)



Reinforcement Learning (RL)

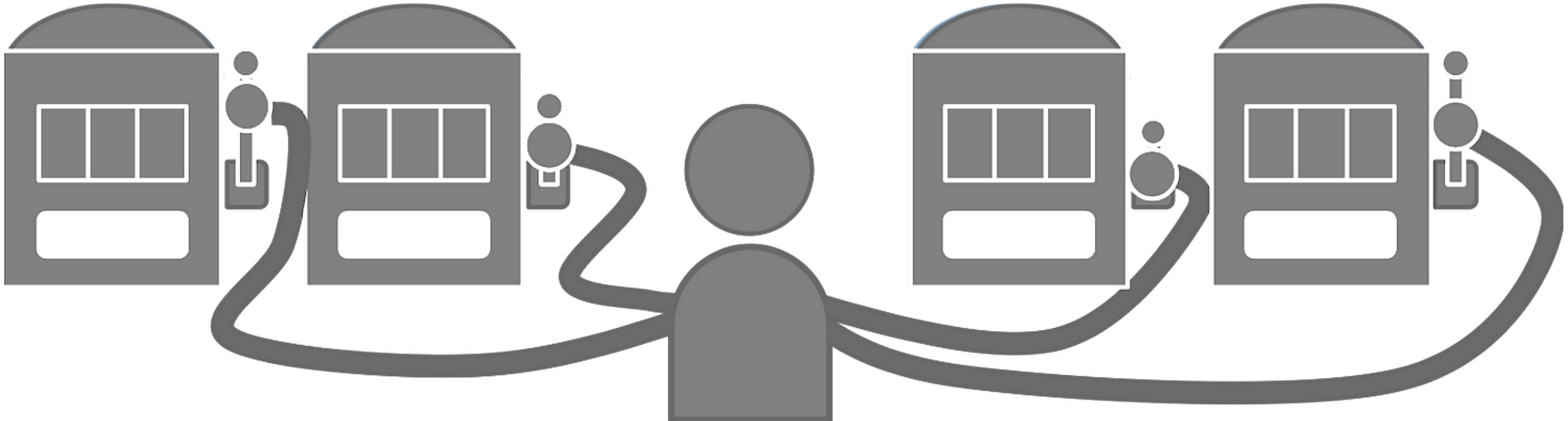


INTRODUCTION
REINFORCEMENT LEARNING



INTRODUCTION
MULTI-ARMED BANDIT (MAB)

- **Multi-Armed:** Multiple options that an algorithm can choose from. Each option is like a **lever** or **arm** that can be pulled.
- **Bandit:** Analogy of casinos where there are multiple slot machines and the gambler has to decide which machine to play.



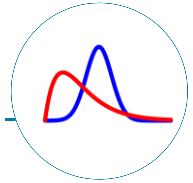
MULTI-ARMED BANDIT
EXPLORATION-EXPLOITATION TRADE-OFF

The agent **explores** different arms to see which ones yield the best reward.

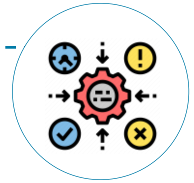
The agent **exploits** the arms that have yielded the best rewards.

MULTI-ARMED BANDIT
OVERVIEW

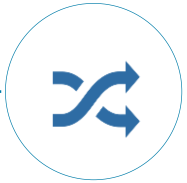
Types of MAB



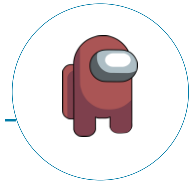
- Stochastic Bandits



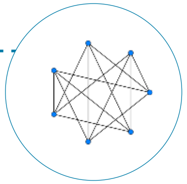
- Contextual Bandits



- Restless Bandits



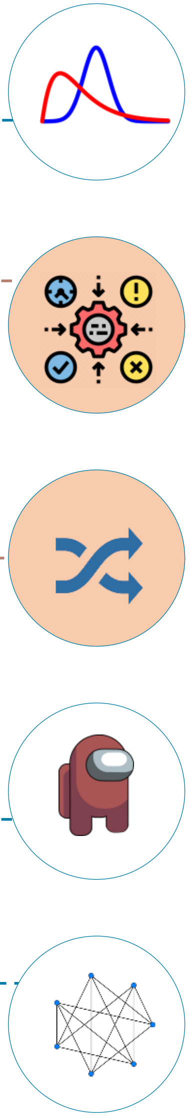
- Adversarial Bandits



- Combinatorial Bandits

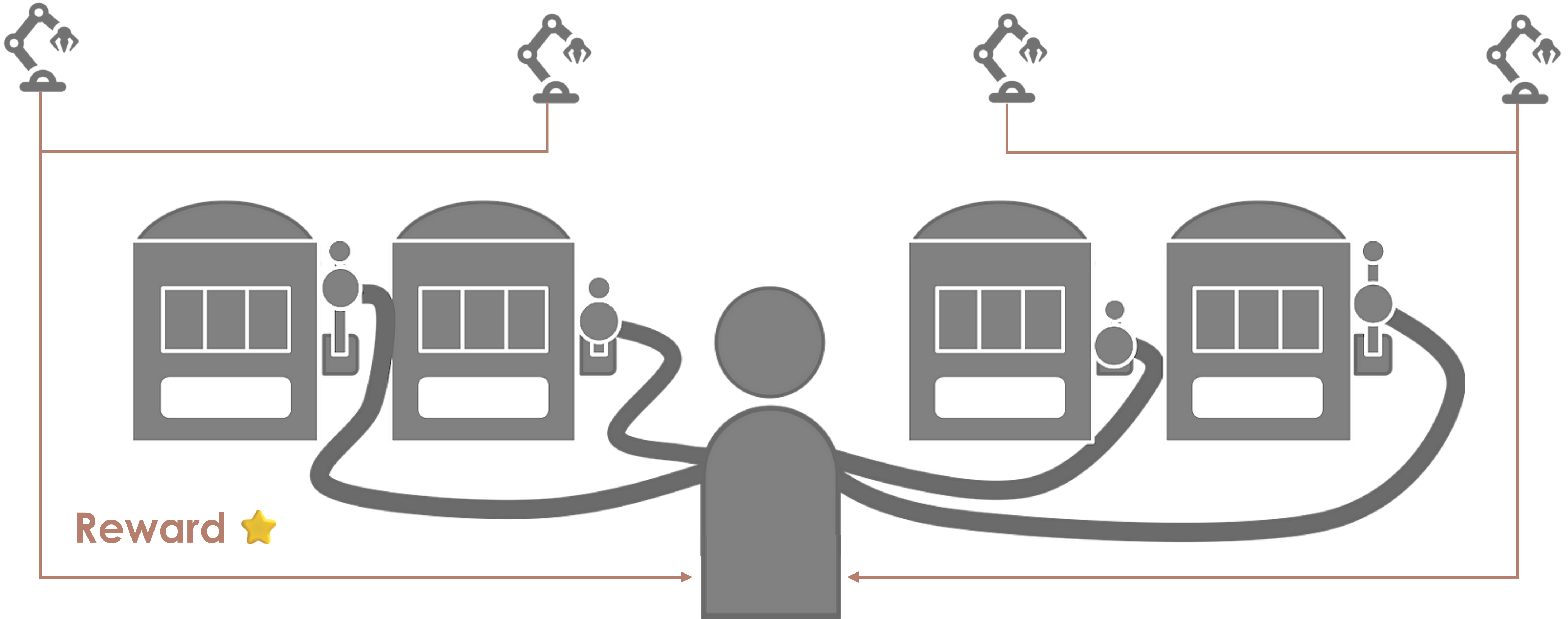
MULTI-ARMED BANDIT
OVERVIEW

Types of MAB



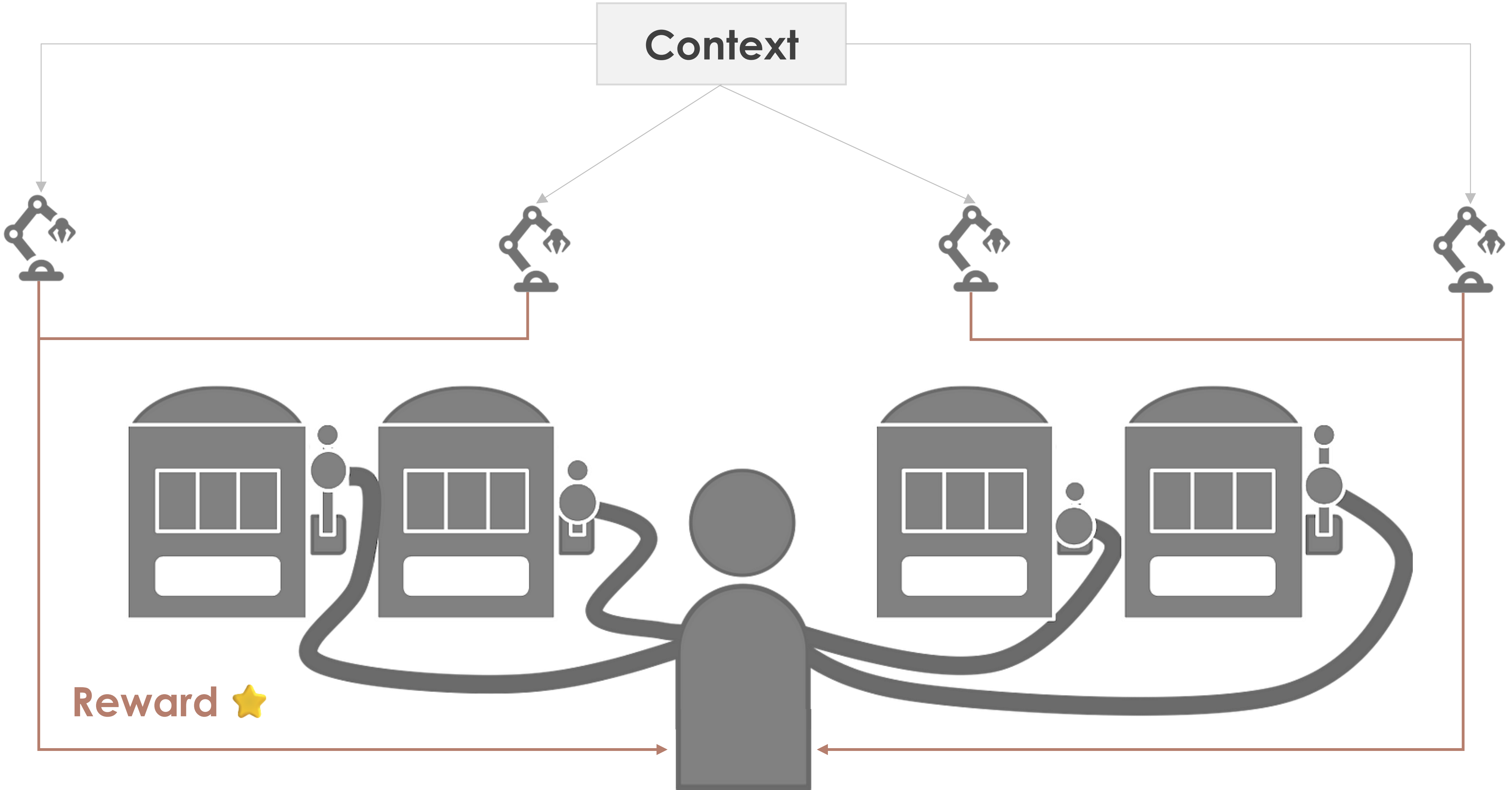
- Stochastic Bandits
- **Contextual Bandits**
- **Restless Bandits**
- Adversarial Bandits
- Combinatorial Bandits

MULTI-ARMED BANDITS
OVERVIEW



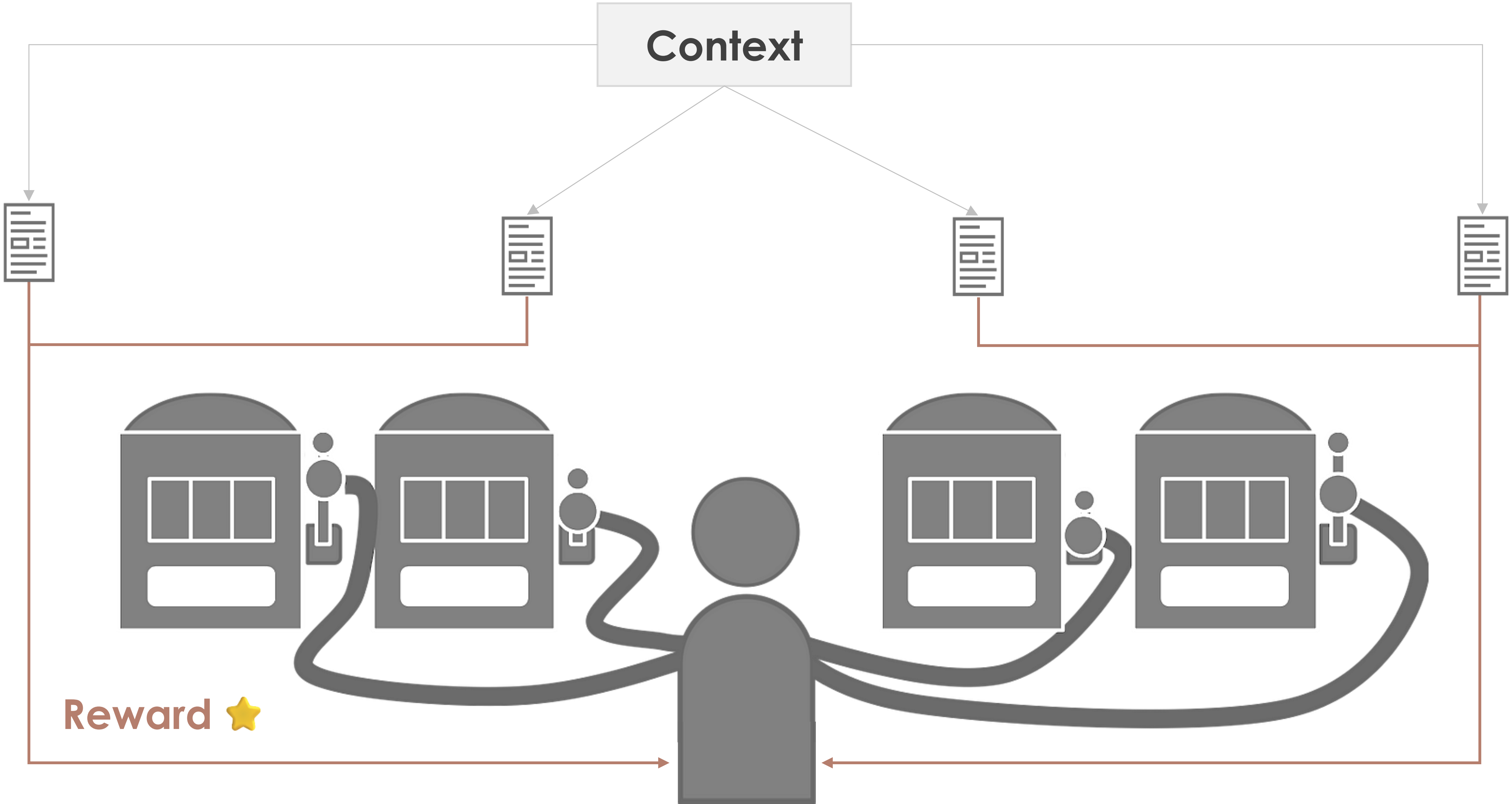
CONTEXTUAL MULTI-ARMED BANDITS

ADDING CONTEXT



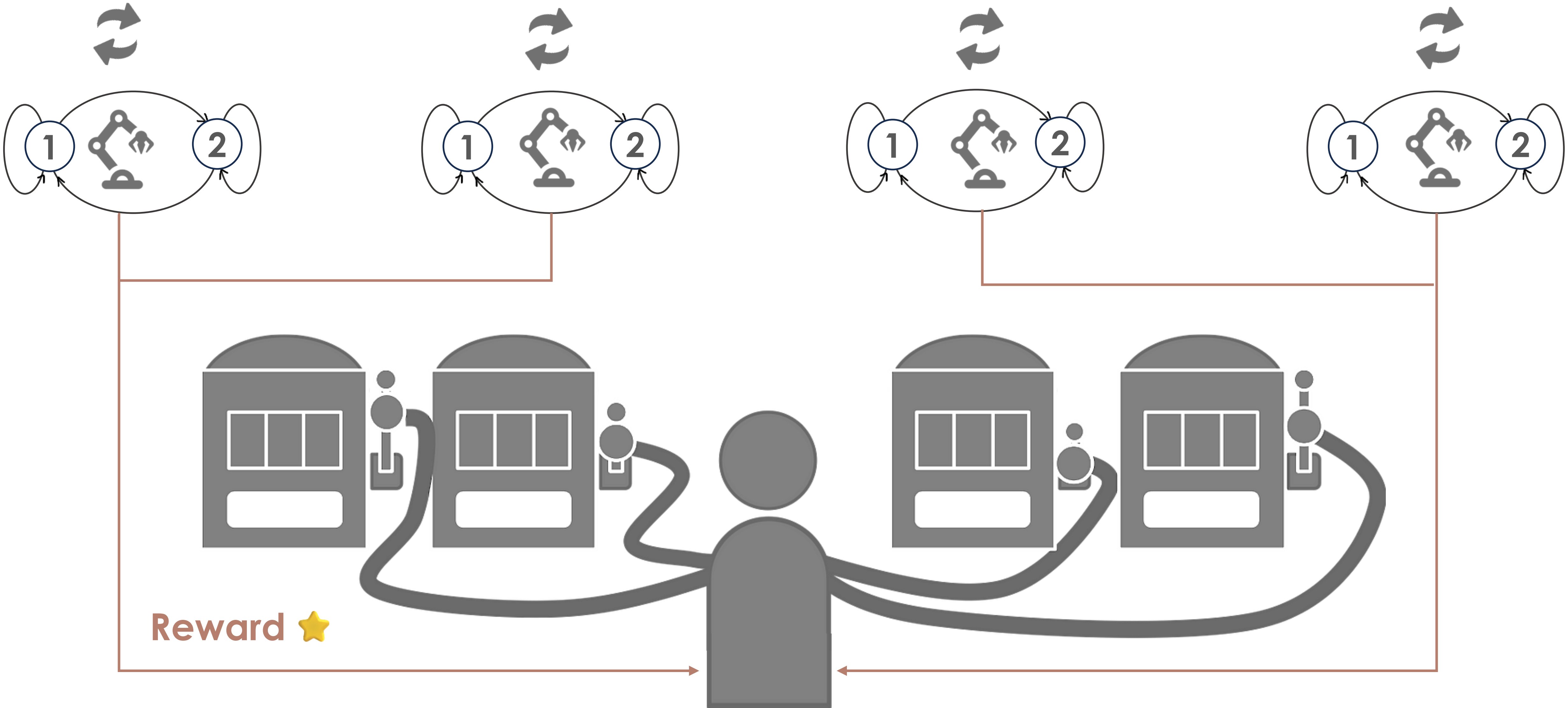
CONTEXTUAL MULTI-ARMED BANDITS

EXAMPLE: LinUCB



RESTLESS MULTI-ARMED BANDITS

RESTLESS ARMS



RESTLESS MULTI-ARMED BANDITS

MARKOVIAN BANDITS

- Restless Multi-Armed Bandits (or Restless Bandits) evolve independently in a controlled Markov manner, i.e. we have:

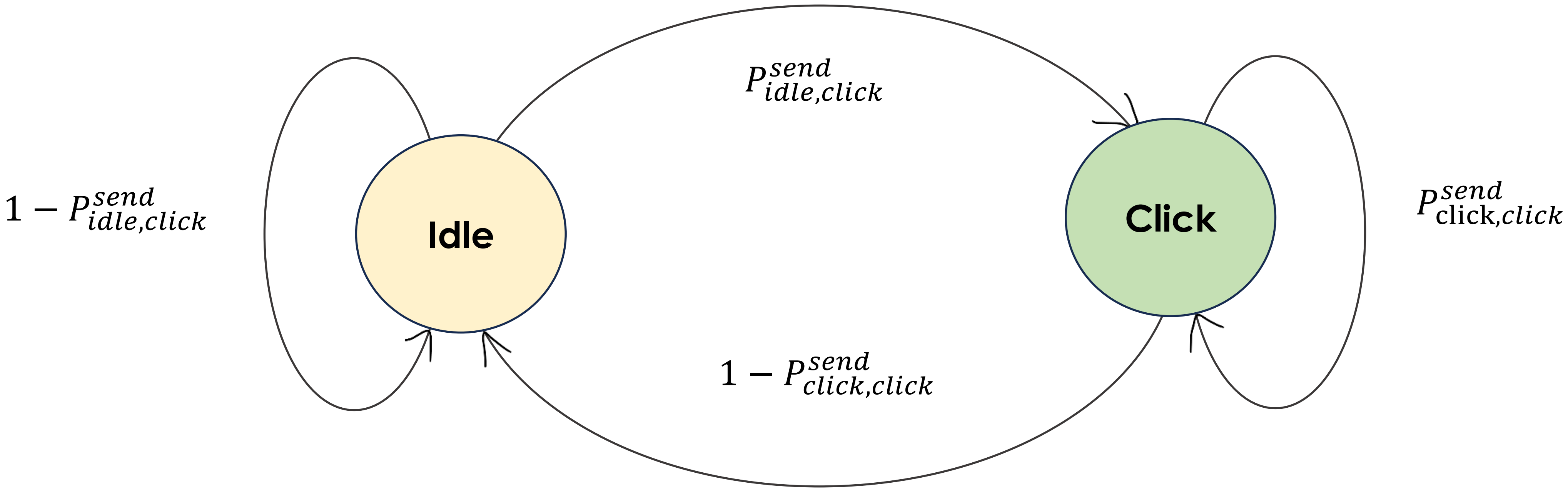
$$\boldsymbol{P}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t) := \mathbb{P}\left(\boldsymbol{S}_{t+1} = \boldsymbol{s}_{t+1}|\boldsymbol{S}_{0:t} = \boldsymbol{s}_{0:t}, \boldsymbol{A}_{0:t} = \boldsymbol{a}_{0:t}\right)$$

$$= \prod_{i=1}^{n} \mathbb{P}\left(S^i_{t+1} = s^i_{t+1}|S^i_t = s^i_t, A^i_t = a^i_t\right)$$

- $S_t := (S^1_t, \ldots S^n_t)$ and $A_t = (A^1_t, \ldots, A^n_t)$ denote the joint state and actions at time t.

RESTLESS MULTI-ARMED BANDITS
A MARKOV DECISION PROCESS

- The state of **each arm** evolves according to a Markovian transition, even when the arm is not played.



RESTLESS BANDITS
FORMULATION

- A Restless Multi-Armed Bandit (RMAB) is defined by the controlled Markov processes $\mathbf{MDP} = (\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{R}, \boldsymbol{P})$, such that:
  - $\boldsymbol{S} = \{\mathbf{1}, \mathbf{2}, \dots, |\boldsymbol{S}|\}$ is the state space, where s_i is the state of arm i.
  - $\boldsymbol{A} = \{\mathbf{0}, \mathbf{1}\}$ denotes the actions space with two actions: $a_\mathrm{i} = 1$ to activate arm i, and $a_\mathrm{i} = 0$ to keep arm i passive.
  - $\boldsymbol{r}_\mathrm{i}{}^{\mathbf{t}} = \boldsymbol{R}(\boldsymbol{s}_\mathrm{i}{}^{\mathbf{t}}, \boldsymbol{a}_\mathrm{i}{}^{\mathbf{t}}, \boldsymbol{s}_\mathrm{i}{}^{\mathbf{t}})$ is the reward function.
  - $\mathbf{P}(\mathbf{s_i}^{\mathbf{t+1}} \,|\, \mathbf{s_i}^{\mathbf{t}}, \mathbf{a_i}^{\mathbf{t}})$ denotes the transition probability from state $\mathbf{s_i}^{\mathbf{t}}$ to state $\mathbf{s_i}^{\mathbf{t+1}}$.

RESTLESS BANDITS
OBJECTIVE

- Find a policy $\pi = (\pi_1, \ldots, \pi_N)$ which maximizes the expected discounted reward i.e. π such that:

$$\text{maximize} \ \ \mathbb{E}_{s_0^{(i)} \sim \mu^{(i)}, a_t^{(i)} \sim \pi^{(i)}(s_t^{(i)}), s_{t+1} \sim p^{(i)}(\cdot | s_t^{(i)}, a_t^{(i)})} \left[\sum_{t=0}^{\infty} \sum_{i=1}^{N} \gamma^t r^{(i)}(s_t^{(i)}, a_t^{(i)}) \right]$$

subject to the following hard constraint $\forall t$:

$$\sum_{i=1}^{N} a_t^{(i)} = M$$

RESTLESS BANDITS
OBJECTIVE

- Find a policy $\pi = (\pi_1, \dots, \pi_N)$ which maximizes the expected discounted reward i.e. π such that:

$$\text{maximize} \quad \mathbb{E}_{s_0^{(i)} \sim \mu^{(i)}, a_t^{(i)} \sim \pi^{(i)}(s_t^{(i)}), s_{t+1} \sim p^{(i)}(\cdot | s_t^{(i)}, a_t^{(i)})} \left[\sum_{t=0}^{\infty} \sum_{i=1}^{N} \gamma^t r^{(i)}(s_t^{(i)}, a_t^{(i)}) \right]$$

subject to the following relaxed constraint :

$$\mathbb{E}_{a_t^{(i)} \sim \pi^{(i)}(s_t^{(i)})} \left[\sum_{t=0}^{\infty} \sum_{i=1}^{N} \gamma^t a_t^{(i)} \right] = \frac{M}{1 - \gamma}$$

WHITTLE INDEX

THREE OPTIMIZATION PROBLEMS

- **[Original]** Original problem :

$$\text{maximize } \mathbb{E}[\sum_{t=0}^{\infty}\sum_{i=1}^{N}\gamma^t r^{(i)}(s_t^{(i)}, a_t^{(i)})]$$
$$\text{s.t. } \sum_{i=1}^{N} a_t^{(i)} = M, \ \forall t$$

- **[Relaxed]** Problem with Relaxed activation constraint :

$$\sum_{t=0}^{\infty}\sum_{i=1}^{N}\gamma^t a_t^{(i)} = \frac{M}{1-\gamma}$$

- **[Lagrange]** Introducing Lagrange Multiplier :

$$\text{maximize } \mathbb{E}\left[\sum_{t=0}^{\infty}\sum_{i=1}^{N}\gamma^t \left(r^{(i)}\left(s_t^{(i)}, a_t^{(i)}\right) + \tilde{\lambda}\left(1 - a_t^{(i)}\right)\right)\right]$$

WHITTLE INDEX

THREE OPTIMIZATION PROBLEMS

- **[Original]** Original problem :

$$\text{maximize } \mathbb{E}[\sum_{t=0}^{\infty}\sum_{i=1}^{N}\gamma^t r^{(i)}(s_t^{(i)}, a_t^{(i)})]$$

$$\text{s.t.} \sum_{i=1}^{N} a_t^{(i)} = M, \ \forall t$$

- **[Relaxed]** Problem with Relaxed activation constraint : $\tilde{\lambda}$ x $\sum_{t=0}^{\infty}\sum_{i=1}^{N}\gamma^t a_t^{(i)} = \frac{M}{1-\gamma}$ x $\tilde{\lambda}$

- **[Lagrange]** Introducing Lagrange Multiplier :

$$\text{maximize } \mathbb{E}\left[\sum_{t=0}^{\infty}\sum_{i=1}^{N}\gamma^t\left(r^{(i)}\left(s_t^{(i)}, a_t^{(i)}\right) + \tilde{\lambda}\left(1 - a_t^{(i)}\right)\right)\right]$$

WHITTLE INDEX
DECOUPLED PROBLEM

- **[Lagrange]** The function is given by :

$$\text{maximize } \mathbb{E}\left[\sum_{t=0}^{\infty}\sum_{i=1}^{N}\gamma^t\left(r^{(i)}(s_t^{(i)}, a_t^{(i)}) - \tilde{\lambda}\, a_t^{(i)}\right)\right] + \tilde{\lambda}(M/(1-\gamma)) \quad s.t.\ \ a_t^{(i)} \in \{0,1\}, \forall t$$

- **[Decoupled Problem]** Solving the above for $\tilde{\lambda}$ is equivalent to solving the following for each i ∊ [N]. We can decouple this problem and neglect the last term.

$$\text{maximize } \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t\left(r^{(i)}\left(s_t^{(i)}, a_t^{(i)}\right) - \tilde{\lambda}\, a_t^{(i)}\right)\right] \quad s.t.\ a_t^{(i)} \in \{0,1\}, \forall t$$

WHITTLE INDEX

DECOUPLED PROBLEM

- **[Lagrange]** The Lagrange dual function is given by :

$$\text{maximize } \mathbb{E}\left[\sum_{t=0}^{\infty}\sum_{i=1}^{N}\gamma^t\left(r^{(i)}(s_t^{(i)}, a_t^{(i)}) - \tilde{\lambda}\, a_t^{(i)}\right)\right] + \underbrace{\cancel{\tilde{\lambda}(M/(1-\gamma))}}_{\boldsymbol{cte}} \quad s.t.\;\; a_t^{(i)} \in \{0,1\}, \forall t$$

- **[Decoupled Problem]** Solving the above for $\tilde{\lambda}$ is equivalent to solving the following for each i ∊ [N]. We can decouple this problem and neglect the last term.

$$\text{maximize } \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t\left(r^{(i)}\left(s_t^{(i)}, a_t^{(i)}\right) - \tilde{\lambda}\, a_t^{(i)}\right)\right] \quad s.t.\; a_t^{(i)} \in \{0,1\}, \forall t$$

WHITTLE INDEX
THREE OPTIMIZATION PROBLEMS

- **[Original]** Original problem :

$$\text{maximize } \mathbb{E}[\sum_{t=0}^{\infty}\sum_{i=1}^{N}\gamma^t r^{(i)}(s_t^{(i)}, a_t^{(i)})]$$

$$\text{s.t.} \sum_{i=1}^{N} a_t^{(i)} = M, \ \forall t$$

- **[Relaxed]** Problem with Relaxed activation constraint : $\tilde{\lambda}$ x $\sum_{t=0}^{\infty}\sum_{i=1}^{N}\gamma^t a_t^{(i)} = \frac{M}{1-\gamma}$ x $\tilde{\lambda}$

- **[Lagrange]** Introducing Lagrange Multiplier :

$$\text{maximize } \mathbb{E}\left[\sum_{t=0}^{\infty}\sum_{i=1}^{N}\gamma^t \left(r^{(i)}\left(s_t^{(i)}, a_t^{(i)}\right) + \tilde{\lambda}\left(1 - a_t^{(i)}\right)\right)\right]$$

WHITTLE INDEX
SOLVING THE DECOUPLED PROBLEM

$$\mathbb{E}\left[\sum_{t=0}^{\infty}\sum_{i=1}^{N}\gamma^{t}\left(r^{(i)}\left(s_t^{(i)},a_t^{(i)}\right)+\tilde{\lambda}\left(1-a_t^{(i)}\right)\right)\right]$$

- The solution of the equation above is obtained by combining the solution to N independent problems.

$$V^i(s) = \max_{a\in\{0,1\}}\left[a\left(r^i(s,1)+\gamma\cdot\sum_j p^i(j|s,1)V^i(j)\right)+\left(1-a\right)\left(r^i(s,0)+\lambda+\gamma\cdot\sum_j p(j,|s,0)V^i(j)\right)\right.$$

- We can rewrite the expression of the equation above as a function of the state-action *$Q^i(s,a)$*.

$$Q^i(s,a) = a\left(r^i(s,a)+\gamma\cdot\sum_j p^i(j|s,a)V^i(j)\right)+(1-a)\left(r^i(s,a)+\lambda+\gamma\cdot\sum_j p(j|s,a)V^i(j)\right)$$

SOLVING THE DECOUPLED PROBLEM
BELLMAN OPTIMALITY EQUATION

- We have to find Q for each i ∈ [N] such that the above is satisfied for some $\tilde{\lambda}$.

[Bellman Optimality Equation]

$$Q^*(s,a) = (1-a)(r(s,a)+\tilde{\lambda}) + r(s,a) + \sum_{s' \in \mathcal{S}} p(s'|s,a) \max_{a' \in \mathcal{A}} Q^*(s',a') \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}$$

[Q-learning]

$$Q_{n+1}(s,a) = \quad Q_n(s,a) + \alpha \, \mathbb{I}\{s_n = s, a_n = a\}\Big((1-a)(r(s,a)+\tilde{\lambda}) + r(s,a) + \max_{a' \in \mathcal{A}} Q_n(s',a')\Big)$$

SOLVING THE DECOUPLED PROBLEM
MINIMUM SUBSIDY

- Rewriting the Bellman optimality equation for active actions 1 and 0 gives:

$$Q^*(s,1) = r(s,1) + \sum_{s' \in \mathcal{S}} p(s'|s,a) \max_{a' \in \mathcal{A}} Q^*(s',a')$$

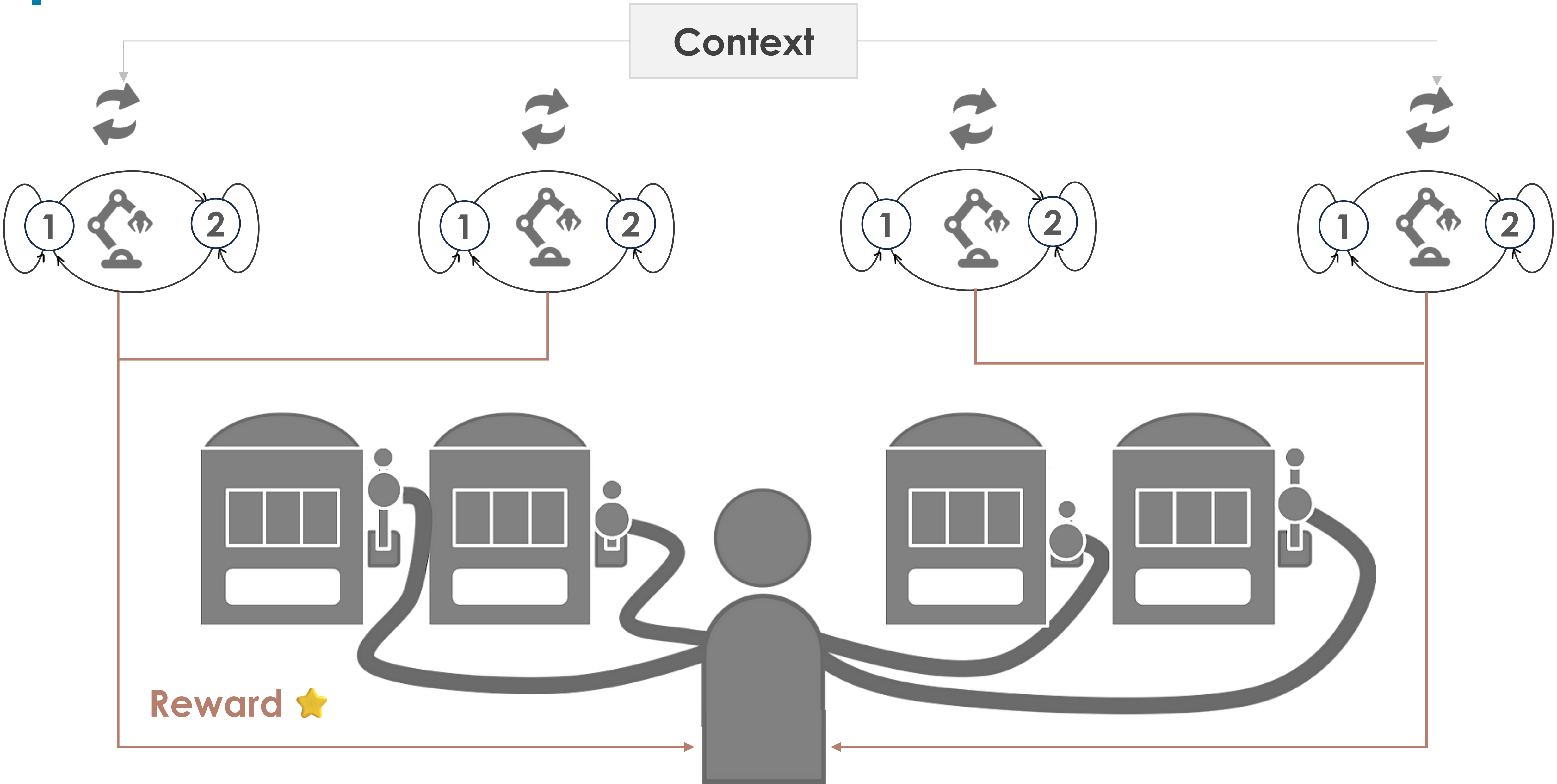
$$Q^*(s,0) = r(s,0) + \tilde{\lambda} + \sum_{s' \in \mathcal{S}} p(s'|s,a) \max_{a' \in \mathcal{A}} Q^*(s',a')$$

- Whittle defines the Whittle Index for state $\hat{k}$ to be the value $\lambda(\hat{k})$ of $\tilde{\lambda}$ for which both active and passive actions are equally preferred in state $\hat{k}$ i.e:

$$Q(\hat{k},1) = Q(\hat{k},0) \ \text{ for } \ \tilde{\lambda} = \lambda(\hat{k})$$

CONTEXTUAL RESTLESS BANDITS

OVERVIEW



RESTLESS BANDITS
RECALL

- A Restless Multi-Armed Bandit (RMAB) is defined by the controlled Markov processes $\mathbf{MDP} = (\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{R}, \boldsymbol{P})$, such that:
  - $\boldsymbol{S} = \{\mathbf{1}, \mathbf{2}, \dots, |\boldsymbol{S}|\}$ is the state space, where s_{i} is the state of arm i.
  - $\boldsymbol{A} = \{\mathbf{0}, \mathbf{1}\}$ denotes the actions space with two actions: $a_{\mathrm{i}} = 1$ to activate arm i, and $a_{\mathrm{i}} = 0$ to keep arm i passive.
  - $\boldsymbol{r}_{\mathrm{i}}{}^{\mathbf{t}} = \boldsymbol{R}(\boldsymbol{s}_{\mathrm{i}}{}^{\mathbf{t}}, \boldsymbol{a}_{\mathrm{i}}{}^{\mathbf{t}}, \boldsymbol{s}_{\mathrm{i}}{}^{\mathbf{t}})$ is the reward function.
  - $\mathbf{P}(\mathbf{s_i}^{\mathbf{t+1}} \,|\, \mathbf{s_i}^{\mathbf{t}}, \mathbf{a_i}^{\mathbf{t}})$ denotes the transition probability from state $\mathbf{s_i}^{\mathbf{t}}$ to state $\mathbf{s_i}^{\mathbf{t+1}}$.

CONTEXTUAL RESTLESS BANDITS

FORMULATION

- A **Contextual** Restless Multi-Armed Bandit **(CRMAB)** is defined by the controlled Markov processes $\mathbf{MDP^c} = (\boldsymbol{S}, \boldsymbol{C}, \boldsymbol{A}, \boldsymbol{R}, \boldsymbol{P})$, such that:
  - $\boldsymbol{S} = \{\mathbf{1}, \mathbf{2}, \dots, |\boldsymbol{S}|\}$ is the state space, where s_i is the state of arm i.
  - $\boldsymbol{C} = \{\mathbf{1}, \mathbf{2}, \dots, |\boldsymbol{C}|\}$, is the context space, where c_i is the context of arm i.
  - $\boldsymbol{A} = \{\mathbf{0}, \mathbf{1}\}$ denotes the actions space with two actions: $a_i = 1$ to activate arm i, and $a_i = 0$ to keep arm i passive.
  - $\boldsymbol{r}_i^{\mathbf{t}} = \boldsymbol{R}(\boldsymbol{s}_i^{\mathbf{t}}, \boldsymbol{c}_i^{\mathbf{t}}, \boldsymbol{a}_i^{\mathbf{t}})$ is the reward function.
  - $\mathrm{P}(\mathbf{s_i^{t+1}} \,|\, \mathbf{s_i^t}, \mathbf{a_i^t}, \mathbf{c_i^t})$ denotes the transition probabilities.

CONTEXTUAL RESTLESS BANDITS

FORMULATION

- A Contextual Restless Multi-Armed Bandit (CRMAB) is defined by the controlled Markov processes $\mathbf{MDP^c} = (\boldsymbol{S}, \boldsymbol{C}, \boldsymbol{A}, \boldsymbol{R}, \boldsymbol{P})$, such that:
  - $\boldsymbol{S} = \{\mathbf{1}, \mathbf{2}, \dots, |\boldsymbol{S}|\}$ is the state space, where s_i is the state of arm i.
  - $\boldsymbol{C} = \{\mathbf{1}, \mathbf{2}, \dots, |\boldsymbol{C}|\}$, is the context space, where c_i is the context of arm i.
  - $\boldsymbol{A} = \{\mathbf{0}, \mathbf{1}\}$ denotes the actions space with two actions: $a_i = 1$ to activate arm i, and $a_i = 0$ to keep arm i passive.
  - $\boldsymbol{r}_i^{\mathbf{t}} = \boldsymbol{R}(\boldsymbol{s}_i^{\mathbf{t}}, \boldsymbol{c}_i^{\mathbf{t}}, \boldsymbol{a}_i^{\mathbf{t}})$ is the reward function.
  - $\mathbf{P}(\mathbf{s_i^{t+1}} \,|\, \mathbf{s_i^{t}}, \mathbf{a_i^{t}}, \mathbf{c_i^{t}})$ denotes the transition probabilities

CONTEXTUAL RESTLESS BANDITS

OBJECTIVE

- Find a policy $\pi = (\pi_1, \dots, \pi_N)$ which maximizes the expected discounted reward i.e. π such that:

$$\text{maximize } \mathbb{E}\left[\sum_{t=0}^{\infty}\sum_{i=1}^{N}\gamma^t r^{(i)}(s_t^{(i)}, c_t^{(i)}, a_t^{(i)})\right]$$

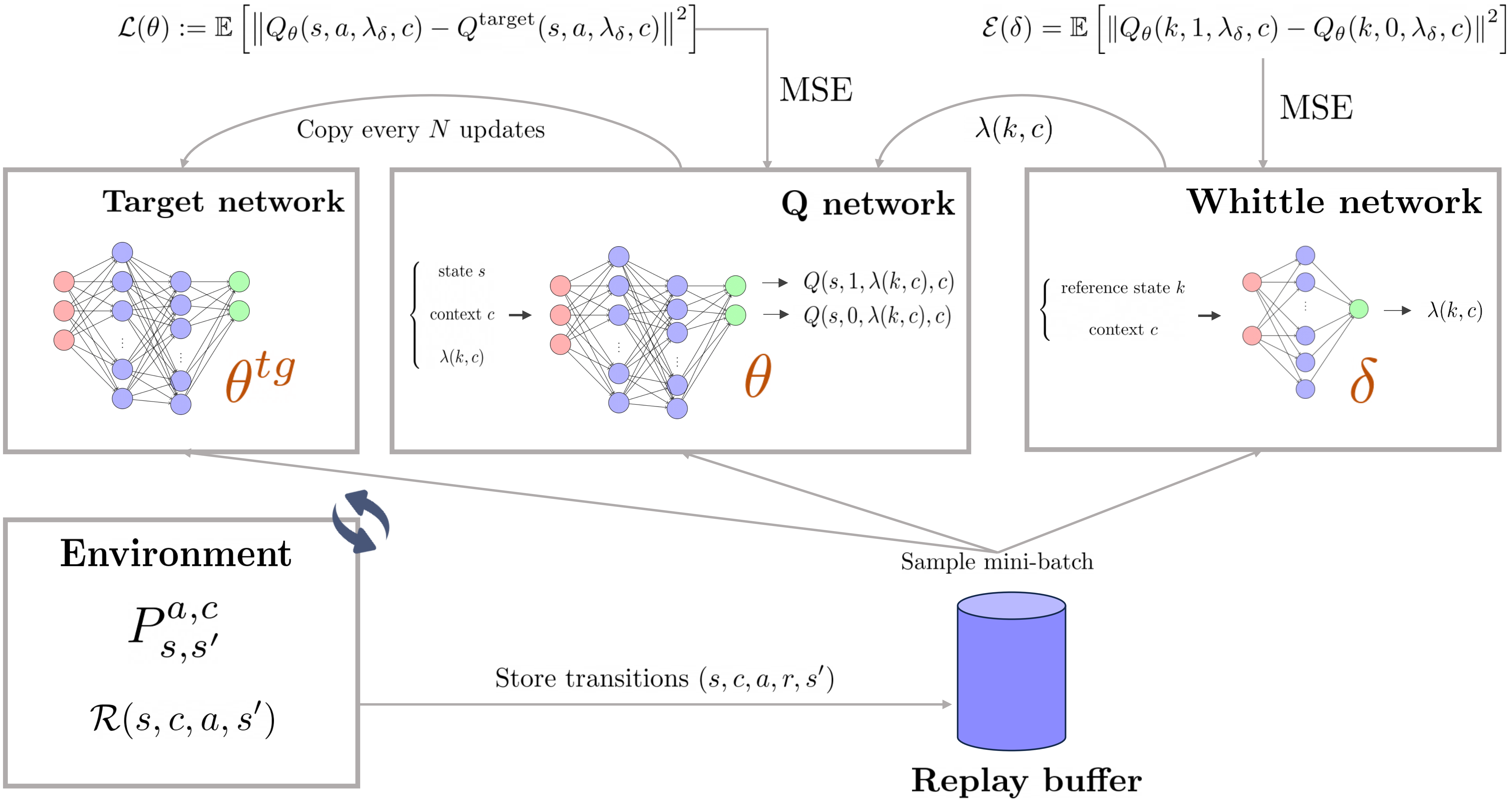
- Emerging topic, only two works on CRMAB:

[1] **X. Chen, I. Hou.** "Contextual Restless Multi-Armed Bandits with Application to Demand Response Decision-Making". In: arXiv preprint arXiv:2403.15640 **(2024)**

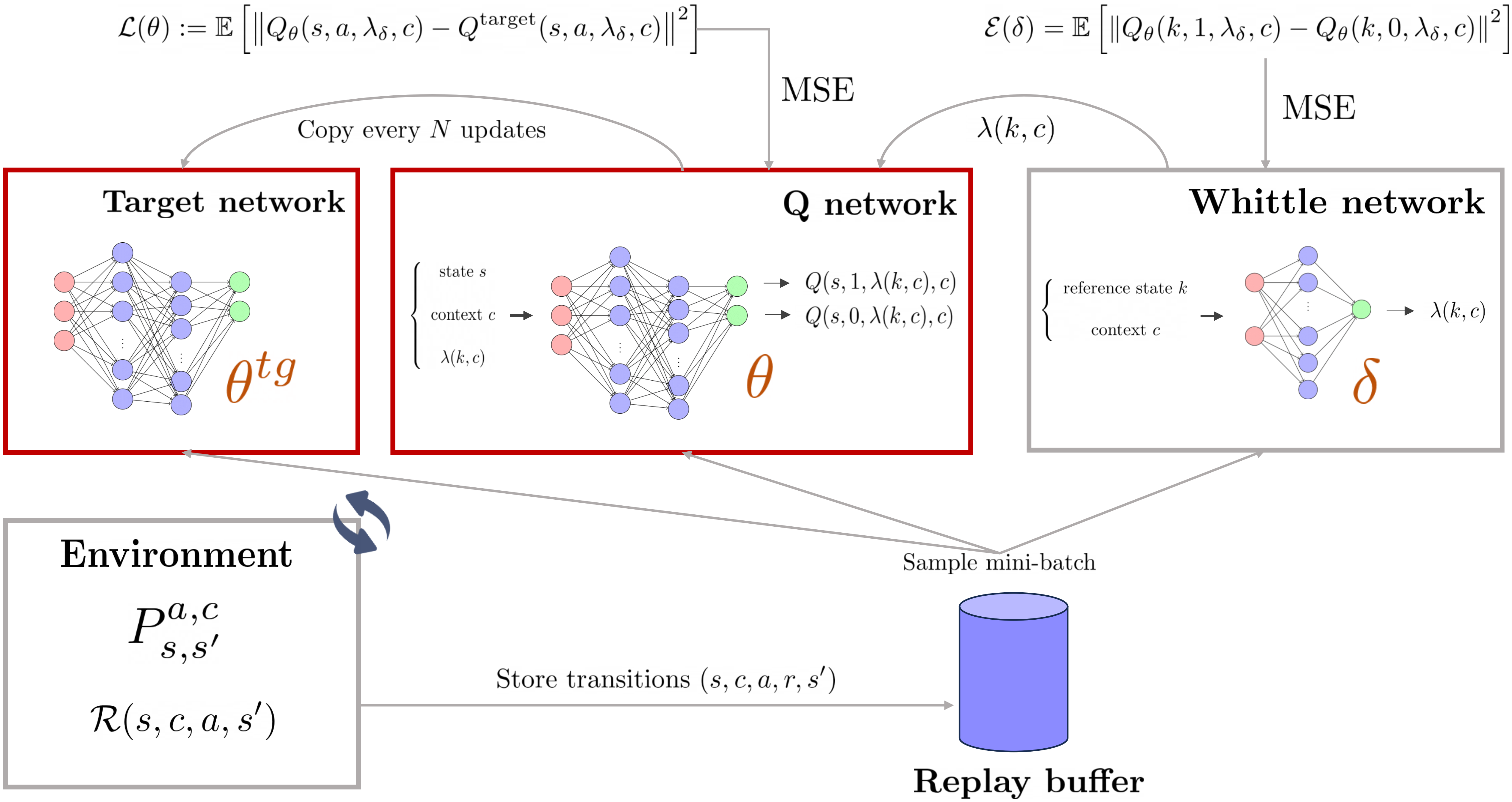
[2] **B. Liang, L. Xu, A. Taneja, M. Tambe, and L. Janson.** "A Bayesian Approach to Online Learning for Contextual Restless Bandits with Applications to Public Health". In: arXiv preprint arXiv:2402.04933 **(2024)**

ALGORITHM
OVERVIEW

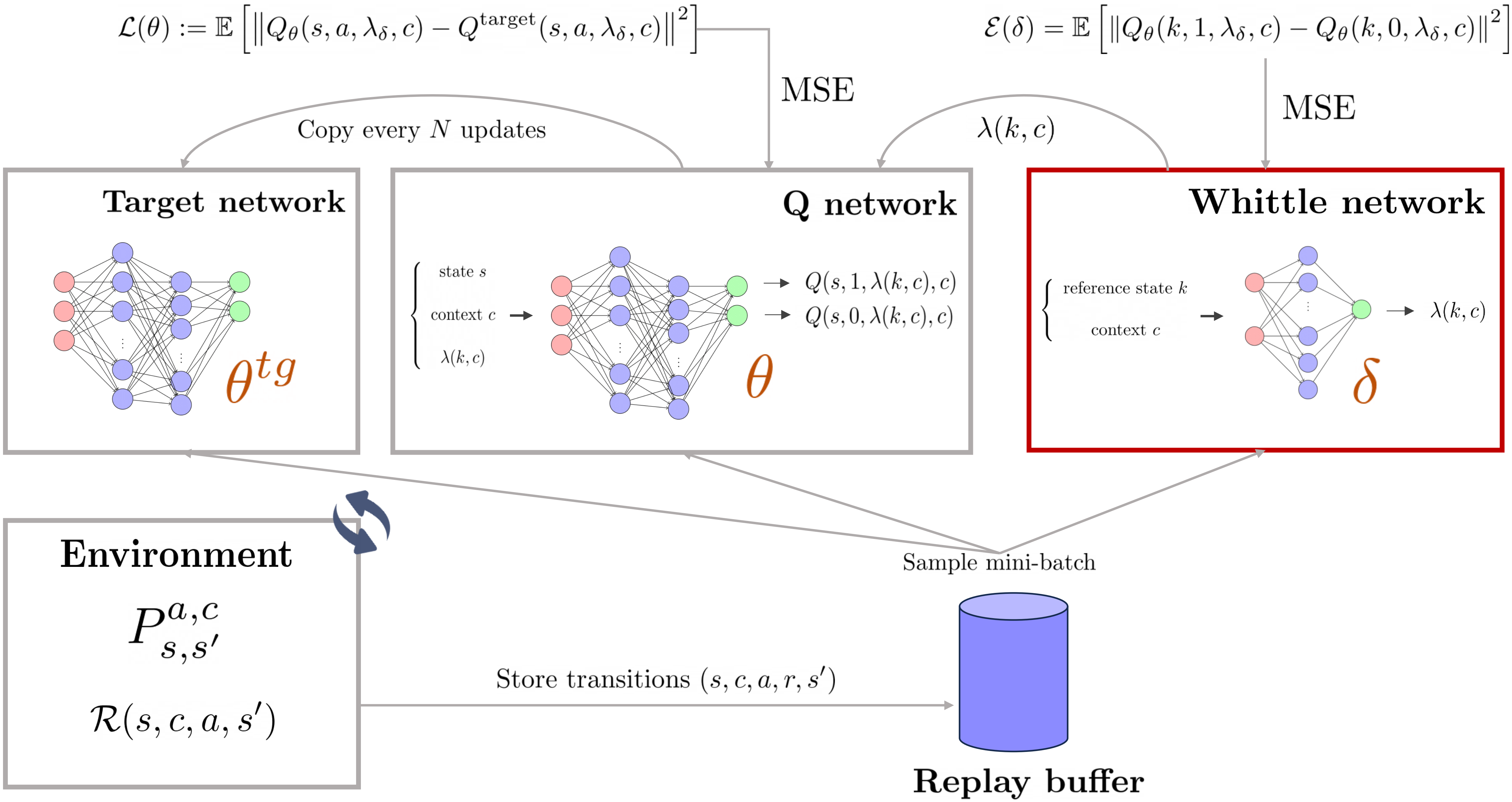
- Deep Q-learning the Whittle Index with Context (DQWIC).
- Combines **deep RL** and **Whittle index** theory within the CRMAB framework.
- Leverages **two Neural Networks (NN).**
  - Q-network for approximating action-value functions,
  - Whittle-network for estimating Whittle indices.
- The learning process occurs through a **two time scale stochastic approximation.**



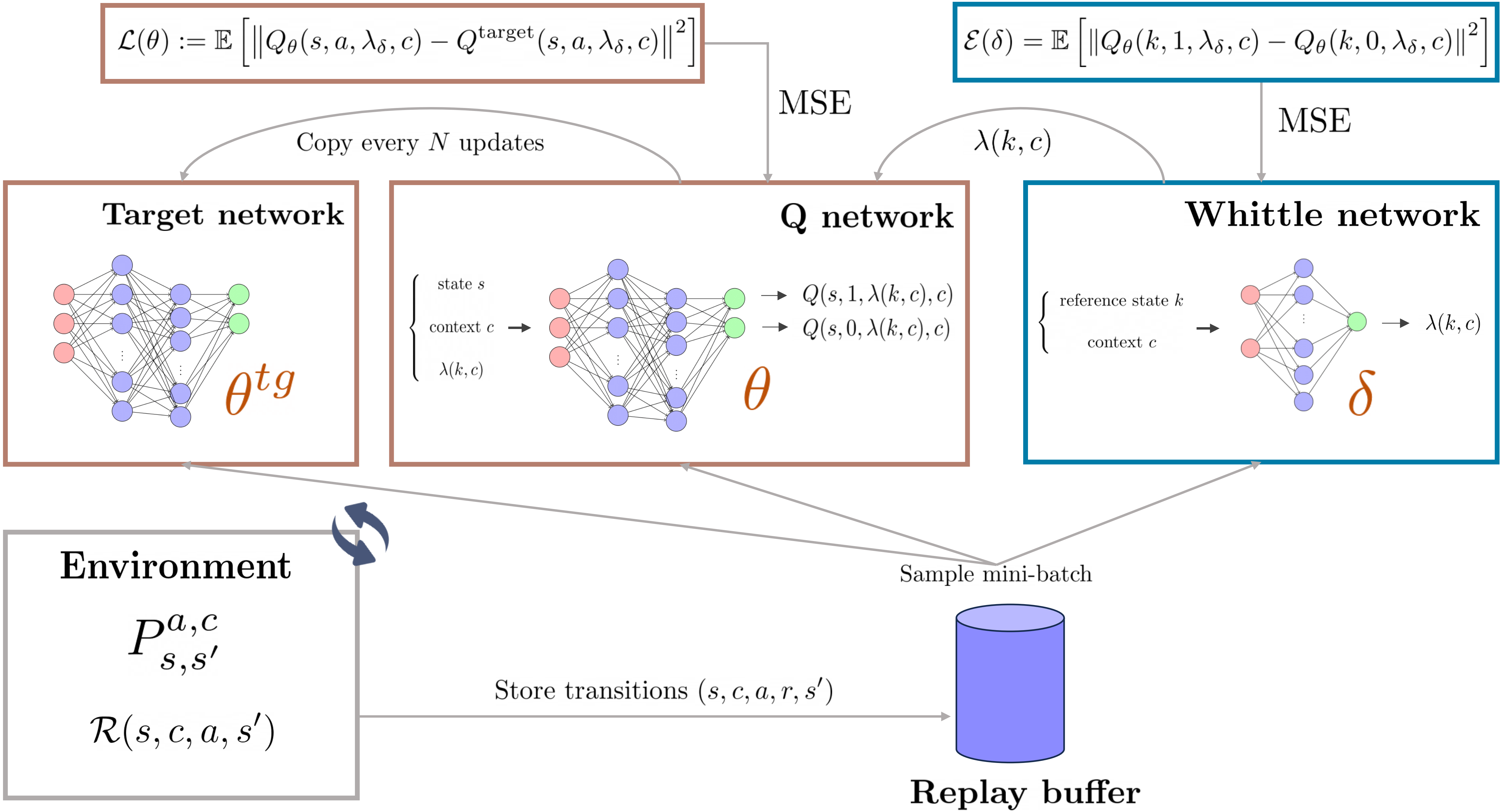
$\mathcal{L}(\theta) := \mathbb{E}\left[\left\|Q_\theta(s,a,\lambda_\delta,c) - Q^{\text{target}}(s,a,\lambda_\delta,c)\right\|^2\right]$
$\mathcal{E}(\delta) = \mathbb{E}\left[\|Q_\theta(k,1,\lambda_\delta,c) - Q_\theta(k,0,\lambda_\delta,c)\|^2\right]$
MSE
MSE
Copy every N updates
$\lambda(k,c)$
Target network
Q network
Whittle network
θ^{tg}
state s
context c
$\lambda(k,c)$
$Q(s,1,\lambda(k,c),c)$
$Q(s,0,\lambda(k,c),c)$
θ
reference state k
context c
$\lambda(k,c)$
δ
Environment
$P^{a,c}_{s,s'}$
$\mathcal{R}(s,c,a,s')$
Sample mini-batch
Store transitions (s,c,a,r,s')
Replay buffer



$\mathcal{L}(\theta) := \mathbb{E}\left[\left\|Q_\theta(s,a,\lambda_\delta,c) - Q^{\text{target}}(s,a,\lambda_\delta,c)\right\|^2\right]$
$\mathcal{E}(\delta) = \mathbb{E}\left[\left\|Q_\theta(k,1,\lambda_\delta,c) - Q_\theta(k,0,\lambda_\delta,c)\right\|^2\right]$
MSE
MSE
Copy every N updates
$\lambda(k,c)$
Target network
θ^{tg}
Q network
state s
context c
$\lambda(k,c)$
$Q(s,1,\lambda(k,c),c)$
$Q(s,0,\lambda(k,c),c)$
θ
Whittle network
reference state k
context c
$\lambda(k,c)$
δ
Environment
$P^{a,c}_{s,s'}$
$\mathcal{R}(s,c,a,s')$
Store transitions (s,c,a,r,s')
Sample mini-batch
Replay buffer



$\mathcal{L}(\theta) := \mathbb{E}\left[\left\|Q_\theta(s,a,\lambda_\delta,c) - Q^{\text{target}}(s,a,\lambda_\delta,c)\right\|^2\right]$
$\mathcal{E}(\delta) = \mathbb{E}\left[\left\|Q_\theta(k,1,\lambda_\delta,c) - Q_\theta(k,0,\lambda_\delta,c)\right\|^2\right]$
MSE
MSE
Copy every N updates
$\lambda(k,c)$
Target network
θ^{tg}
Q network
state s
context c
$\lambda(k,c)$
$Q(s,1,\lambda(k,c),c)$
$Q(s,0,\lambda(k,c),c)$
θ
Whittle network
reference state k
context c
$\lambda(k,c)$
δ
Environment
$P^{a,c}_{s,s'}$
$\mathcal{R}(s,c,a,s')$
Store transitions (s,c,a,r,s')
Sample mini-batch
Replay buffer



$\mathcal{L}(\theta) := \mathbb{E}\left[\left\|Q_\theta(s,a,\lambda_\delta,c) - Q^{\text{target}}(s,a,\lambda_\delta,c)\right\|^2\right]$
$\mathcal{E}(\delta) = \mathbb{E}\left[\left\|Q_\theta(k,1,\lambda_\delta,c) - Q_\theta(k,0,\lambda_\delta,c)\right\|^2\right]$
MSE
MSE
Copy every N updates
$\lambda(k,c)$
Target network
θ^{tg}
Q network
state s
context c
$\lambda(k,c)$
$Q(s,1,\lambda(k,c),c)$
$Q(s,0,\lambda(k,c),c)$
θ
Whittle network
reference state k
context c
$\lambda(k,c)$
δ
Environment
$P^{a,c}_{s,s'}$
$\mathcal{R}(s,c,a,s')$
Sample mini-batch
Store transitions (s,c,a,r,s')
Replay buffer

DQWIC
PSEUDOCODE

Algorithm: Deep Q-learning Whittle Index with Context (DQWIC)

Initialize: Q-network parameters θ, Whittle network parameters δ, target network parameters θ^{tg}, replay memory $\mathcal{D}$, and hyperparameters ϵ, γ.

Get initial state s and context c

for each time step t **do**

    if Uniform$[0,1] < \epsilon$ **then**
        Explore by selecting M random arms
    else
        Exploit by selecting top M arms with the highest Whittle indices
    end if

    Execute action a, observe context c, next state s' and reward r
    Store transitions (s, c, a, r, s') in replay memory $\mathcal{D}$

    if $|\mathcal{D}| >$ batch size **then**

        /* *On a faster time scale* */
        Sample mini-batch of transitions from $\mathcal{D}$
        Compute Q^{target}
        Compute loss: $\mathcal{L}(\theta)$
        Update Q-network parameters θ

        /* *On a slower time scale* */
        Sample batch of (k, c) from $\mathcal{D}$
        Compute loss: $\mathcal{E}(\delta)$
        Update Whittle network parameters δ

    end if

    Update target network parameters $\theta^{tg} \leftarrow \theta$

end for

DQWIC
PSEUDOCODE

Algorithm: Deep Q-learning Whittle Index with Context (DQWIC)

1

Initialize: Q-network parameters θ, Whittle network parameters δ, target network parameters θ^{tg}, replay memory $\mathcal{D}$, and hyperparameters ϵ, γ.

Get initial state s and context c

for each time step t **do**

2

    if Uniform$[0, 1] < \epsilon$ **then**
        Explore by selecting M random arms
    else
        Exploit by selecting top M arms with the highest Whittle indices
    end if

    Execute action a, observe context c, next state s' and reward r
    Store transitions (s, c, a, r, s') in replay memory $\mathcal{D}$

    if $|\mathcal{D}| >$ batch size **then**

3

        / On a faster time scale */*
        Sample mini-batch of transitions from $\mathcal{D}$
        Compute Q^{target}
        Compute loss: $\mathcal{L}(\theta)$
        Update Q-network parameters θ

4

        / On a slower time scale */*
        Sample batch of (k, c) from $\mathcal{D}$
        Compute loss: $\mathcal{E}(\delta)$
        Update Whittle network parameters δ

    end if

5

    Update target network parameters $\theta^{tg} \leftarrow \theta$

end for

1

Parameter initialization

- Q-network params θ
- Whittle network params δ
- Replay memory D
- Exploration param ϵ
- Discount factor γ
- State s and context c

Algorithm: Deep Q-learning Whittle Index with Context (DQWIC)

Initialize: Q-network parameters θ, Whittle network parameters δ, target network parameters θ^{tg}, replay memory $\mathcal{D}$, and hyperparameters ϵ, γ.

```
Get initial state s and context c
for each time step t do
    if Uniform[0, 1] < ϵ then
        Explore by selecting M random arms
    else
        Exploit by selecting top M arms with the highest Whittle indices
    end if
    Execute action a, observe context c, next state s′ and reward r
    Store transitions (s, c, a, r, s′) in replay memory D
    if |D| > batch size then
        /* On a faster time scale */
        Sample mini-batch of transitions from D
        Compute Q^target
        Compute loss: L(θ)
        Update Q-network parameters θ
        /* On a slower time scale */
        Sample batch of (k, c) from D
        Compute loss: E(δ)
        Update Whittle network parameters δ
    end if
    Update target network parameters θ^tg ← θ
end for
```

1 Parameter initialization

2 Selection of M arms

- Epsilon-greedy:
  - Explore with ϵ
  - Exploit with $1 - \epsilon$
- Observe transitions and store them in replay memory D

Algorithm: Deep Q-learning Whittle Index with Context (DQWIC)

Initialize: Q-network parameters θ, Whittle network parameters δ, target network parameters θ^{tg}, replay memory $\mathcal{D}$, and hyperparameters ϵ, γ.

Get initial state s and context c

for each time step t **do**

    if Uniform$[0, 1] < \epsilon$ **then**

        Explore by selecting M random arms

    else

        Exploit by selecting top M arms with the highest Whittle indices

    end if

    Execute action a, observe context c, next state s' and reward r

    Store transitions (s, c, a, r, s') in replay memory $\mathcal{D}$

    if $|\mathcal{D}| >$ batch size **then**

        / On a faster time scale */*

        Sample mini-batch of transitions from $\mathcal{D}$

        Compute Q^{target}

        Compute loss: $\mathcal{L}(\theta)$

        Update Q-network parameters θ

        / On a slower time scale */*

        Sample batch of (k, c) from $\mathcal{D}$

        Compute loss: $\mathcal{E}(\delta)$

        Update Whittle network parameters δ

    end if

    Update target network parameters $\theta^{tg} \leftarrow \theta$

end for

1 Parameter initialization

2 Selection of M arms

3 Run on fast timescale

Algorithm: Deep Q-learning Whittle Index with Context (DQWIC)

Initialize: Q-network parameters θ, Whittle network parameters δ, target network parameters θ^{tg}, replay memory $\mathcal{D}$, and hyperparameters ϵ, γ.

```
Get initial state s and context c
for each time step t do
    if Uniform[0, 1] < ε then
        Explore by selecting M random arms
    else
        Exploit by selecting top M arms with the highest Whittle indices
    end if
    Execute action a, observe context c, next state s′ and reward r
    Store transitions (s, c, a, r, s′) in replay memory 𝒟
    if |𝒟| > batch size then
        /* On a faster time scale */
        Sample mini-batch of transitions from 𝒟
        Compute Q^target
        Compute loss: ℒ(θ)
        Update Q-network parameters θ
        /* On a slower time scale */
        Sample batch of (k, c) from 𝒟
        Compute loss: ℰ(δ)
        Update Whittle network parameters δ
    end if
    Update target network parameters θ^tg ← θ
end for
```

1. Parameter initialization
2. Selection of M arms
3. Run on fast timescale
4. Run on slow timescale

Algorithm: Deep Q-learning Whittle Index with Context (DQWIC)

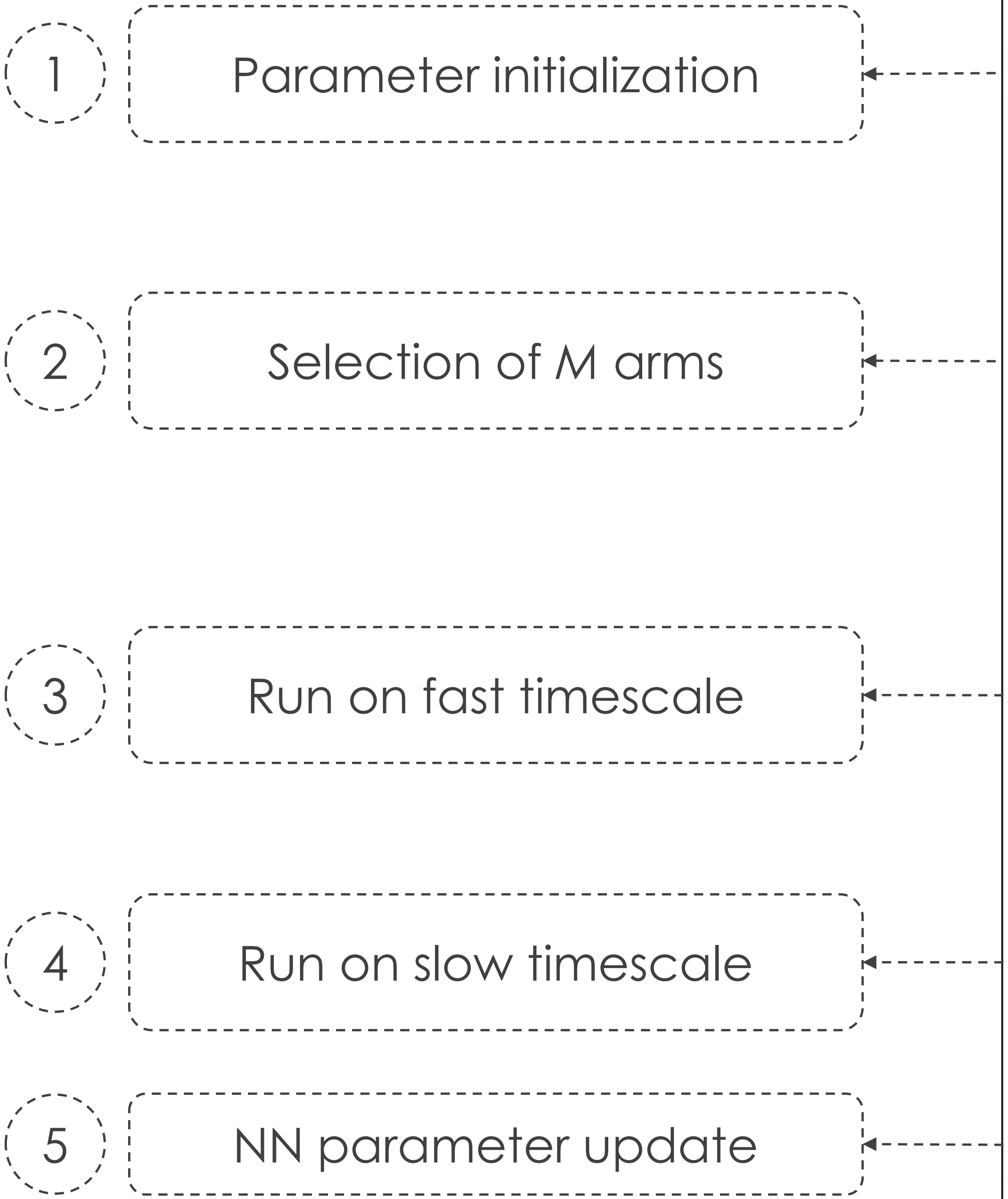
Initialize: Q-network parameters θ, Whittle network parameters δ, target network parameters θ^{tg}, replay memory $\mathcal{D}$, and hyperparameters ϵ, γ.

Get initial state s and context c
for each time step t **do**
    if Uniform$[0, 1] < \epsilon$ **then**
        Explore by selecting M random arms
    else
        Exploit by selecting top M arms with the highest Whittle indices
    end if
    Execute action a, observe context c, next state s' and reward r
    Store transitions (s, c, a, r, s') in replay memory $\mathcal{D}$
    if $|\mathcal{D}| >$ batch size **then**
        / On a faster time scale */*
        Sample mini-batch of transitions from $\mathcal{D}$
        Compute Q^{target}
        Compute loss: $\mathcal{L}(\theta)$
        Update Q-network parameters θ
        / On a slower time scale */*
        Sample batch of (k, c) from $\mathcal{D}$
        Compute loss: $\mathcal{E}(\delta)$
        Update Whittle network parameters δ
    end if
    Update target network parameters $\theta^{tg} \leftarrow \theta$
end for

Algorithm: Deep Q-learning Whittle Index with Context (DQWIC)

Initialize: Q-network parameters θ, Whittle network parameters δ, target network parameters θ^{tg}, replay memory $\mathcal{D}$, and hyperparameters ϵ, γ.

```
Get initial state s and context c
for each time step t do
    if Uniform[0, 1] < ε then
        Explore by selecting M random arms
    else
        Exploit by selecting top M arms with the highest Whittle indices
    end if
    Execute action a, observe context c, next state s' and reward r
    Store transitions (s, c, a, r, s') in replay memory D
    if |D| > batch size then
        /* On a faster time scale */
        Sample mini-batch of transitions from D
        Compute Q^target
        Compute loss: L(θ)
        Update Q-network parameters θ
        /* On a slower time scale */
        Sample batch of (k, c) from D
        Compute loss: E(δ)
        Update Whittle network parameters δ
    end if
    Update target network parameters θ^tg ← θ
end for
```



Algorithm: Deep Q-learning Whittle Index with Context (DQWIC)

Initialize: Q-network parameters θ, Whittle network parameters δ, target network parameters θ^{tg}, replay memory $\mathcal{D}$, and hyperparameters ϵ, γ.

Get initial state s and context c

for each time step t **do**

    if Uniform$[0, 1] < \epsilon$ **then**

        Explore by selecting M random arms

    else

        Exploit by selecting top M arms with the highest Whittle indices

    end if

    Execute action a, observe context c, next state s' and reward r

    Store transitions (s, c, a, r, s') in replay memory $\mathcal{D}$

    if $|\mathcal{D}| >$ batch size **then**

        / On a faster time scale */*

        Sample mini-batch of transitions from $\mathcal{D}$

        Compute Q^{target}

        Compute loss: $\mathcal{L}(\theta)$

        Update Q-network parameters θ

        / On a slower time scale */*

        Sample batch of (k, c) from $\mathcal{D}$

        Compute loss: $\mathcal{E}(\delta)$

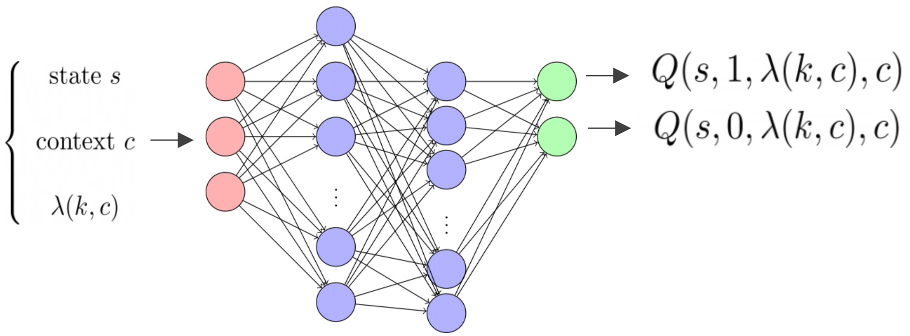
        Update Whittle network parameters δ

    end if

    Update target network parameters $\theta^{tg} \leftarrow \theta$

end for

DQWIC
Q-NETWORK



- On a fast time scale, sample a batch (which contains $[s, c, a, r, s']$) of size B from the memory D.
- The Q-network computes the predicted Q-values for all possible actions as follows:

$$Q^{\text{target}}(s_t, a_t, \lambda_\delta, c_t) = (1 - a_t)(r_0(s_t) + \lambda_\delta) + a_t r_1(s_t) + \gamma \max_{a \in \{0,1\}} Q^{\theta_{tg}}(s_t', a, \lambda_\delta, c_t)$$

- Target network is a copy of the Q-network.
- $\max_{a \in \{0,1\}} Q^{\theta_{\text{tg}}}(s_t', a, \lambda_\delta, c_t)$ is the maximum Q-value for the next state s'_t over all possible actions, predicted by the target network parameters θ^{tg}.

DQWIC

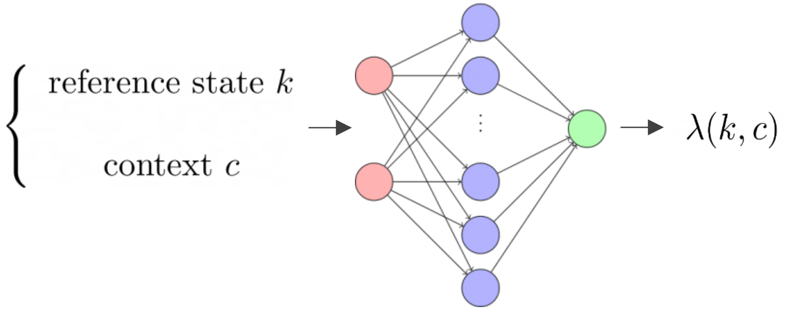
PARAMETER UPDATE

- The Q-network parameters θ are updated by minimizing the Mean Squared Error (MSE) between the predicted Q-values and the target Q-values.
- The loss function is defined as:

$$\mathcal{L}(\theta) := \mathbb{E}\left[\left\|Q_\theta(s, a, \lambda_s, c) - Q^{\text{target}}(s, a, \lambda_s, c)\right\|^2\right]$$

- The parameters θ of the Q-network are updated using backpropagation to minimize this loss.
- The parameters of the target network θ^{tg} are periodically synchronized with those of the Q-network θ, for instance every 100 iterations, to stabilize training.

DQWIC
WHITTLE NETWORK



- On a slow time scale, a batch of reference states k and contexts c are sampled from the replay memory D. These samples are used to compute the MSE loss function E (δ) based on the difference between the Q-values for the active and passive actions.

$$\mathcal{E}(\delta) = \mathbb{E}\left[\left\|Q_\theta(k, 1, \lambda_s, c) - Q_\theta(k, 0, \lambda_s, c)\right\|^2\right]$$

- The parameters δ of the Whittle network are updated using gradient descent to minimize the loss E (δ).

APPLICATION
RECOMMENDER SYSTEMS

- A Recommender System (RS) is:
  - an information filtering system
  - suggests items to users based on their past preferences or behavior.
  - used in many domains such as in: e-commerce, entertainment, and social media, etc.
- Let $R(u, j)$ represent the interaction between user u and item j. This could be:
  - **Explicit:** Ratings (e.g., $R(u, j) \in \{1, 2, \ldots, 5\}$).
  - **Implicit:** Binary clicks or preferences (e.g., $R(u, j) \in \{0, 1\}$).

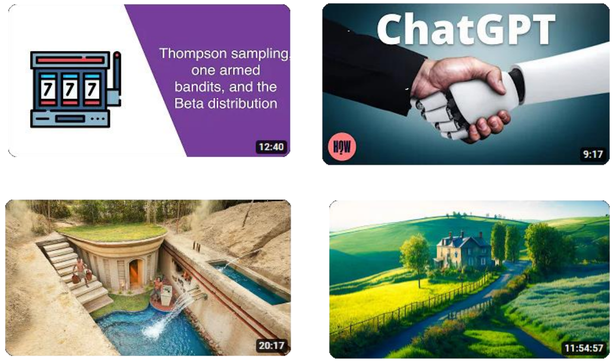
RECOMMENDER SYSTEM

EXAMPLES

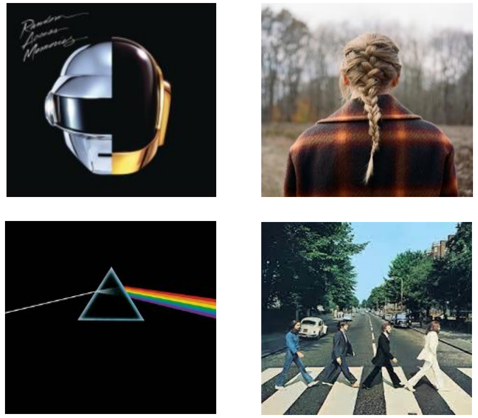
Because you watched this show …

Customers like you also bought …

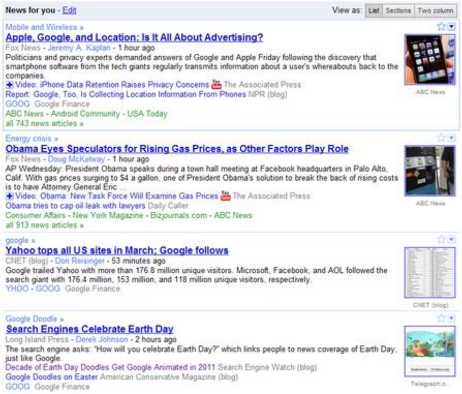
Videos recommended for you …



See what's popular now …



Recommended articles for you …



News for you - Edit

View as: List | Sections | Two column

Mobile and Wireless »

Apple, Google, and Location: Is It All About Advertising?

Fox News - Jeremy A. Kaplan - 1 hour ago

Politicians and privacy experts demanded answers of Google and Apple Friday following the discovery that smartphone software from the tech giants regularly transmits information about a user's whereabouts back to the companies.

Video: iPhone Data Retention Raises Privacy Concerns The Associated Press

Report: Google, Too, Is Collecting Location Information From Phones NPR (blog)

GOOG Google Finance

ABC News - Android Community - USA Today

all 743 news articles »

ABC News

Energy crisis »

Obama Eyes Speculators for Rising Gas Prices, as Other Factors Play Role

Fox News - Doug McKelway - 1 hour ago

AP Wednesday: President Obama speaks during a town hall meeting at Facebook headquarters in Palo Alto, Calif. With gas prices surging to $4 a gallon, one of President Obama's solution to break the back of rising costs is to have Attorney General Eric ...

Video: Obama: New Task Force Will Examine Gas Prices The Associated Press

Obama tries to cap oil leak with lawyers Daily Caller

Consumer Affairs - New York Magazine - Bizjournals.com - ABC News

all 913 news articles »

ABC News

google »

Yahoo tops all US sites in March; Google follows

CNET (blog) - Don Reisinger - 53 minutes ago

Google trailed Yahoo with more than 176.8 million unique visitors. Microsoft, Facebook, and AOL followed the search giant with 176.4 million, 153 million, and 118 million unique visitors, respectively.

YHOO - GOOG Google Finance

CNET (blog)

Google Doodle »

Search Engines Celebrate Earth Day

Long Island Press - Derek Johnson - 2 hours ago

The search engine asks: "How will you celebrate Earth Day?" which links people to news coverage of Earth Day, just like Google.

Decade of Earth Day Doodles Get Google Animated in 2011 Search Engine Watch (blog)

Google Doodles on Easter American Conservative Magazine (blog)

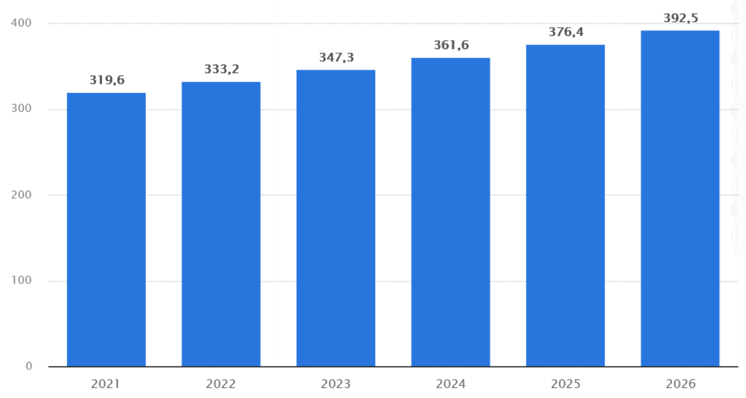
GOOG Google Finance

Telegraph.c...

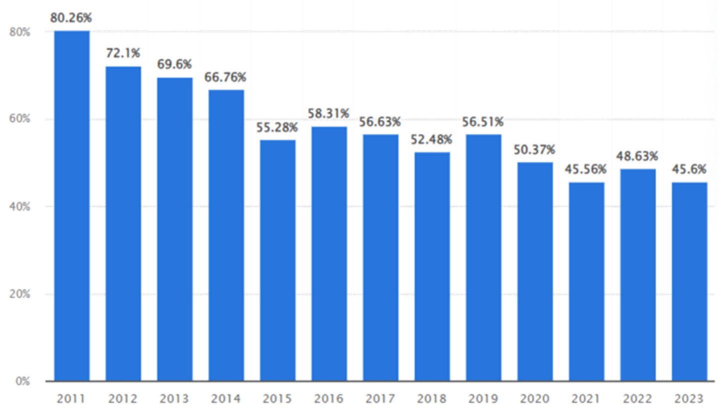
Items similar to your purchase …

EMAIL RECOMMENDER SYSTEM
MOTIVATION

- Sending mass emails results in suboptimal performance:
  - bad domain reputation with potential **spamming**,
  - overwhelming recipients with irrelevant content,
  - decreasing user engagement,
  - negative **user experience**.



~ 347 billion daily emails (2023)

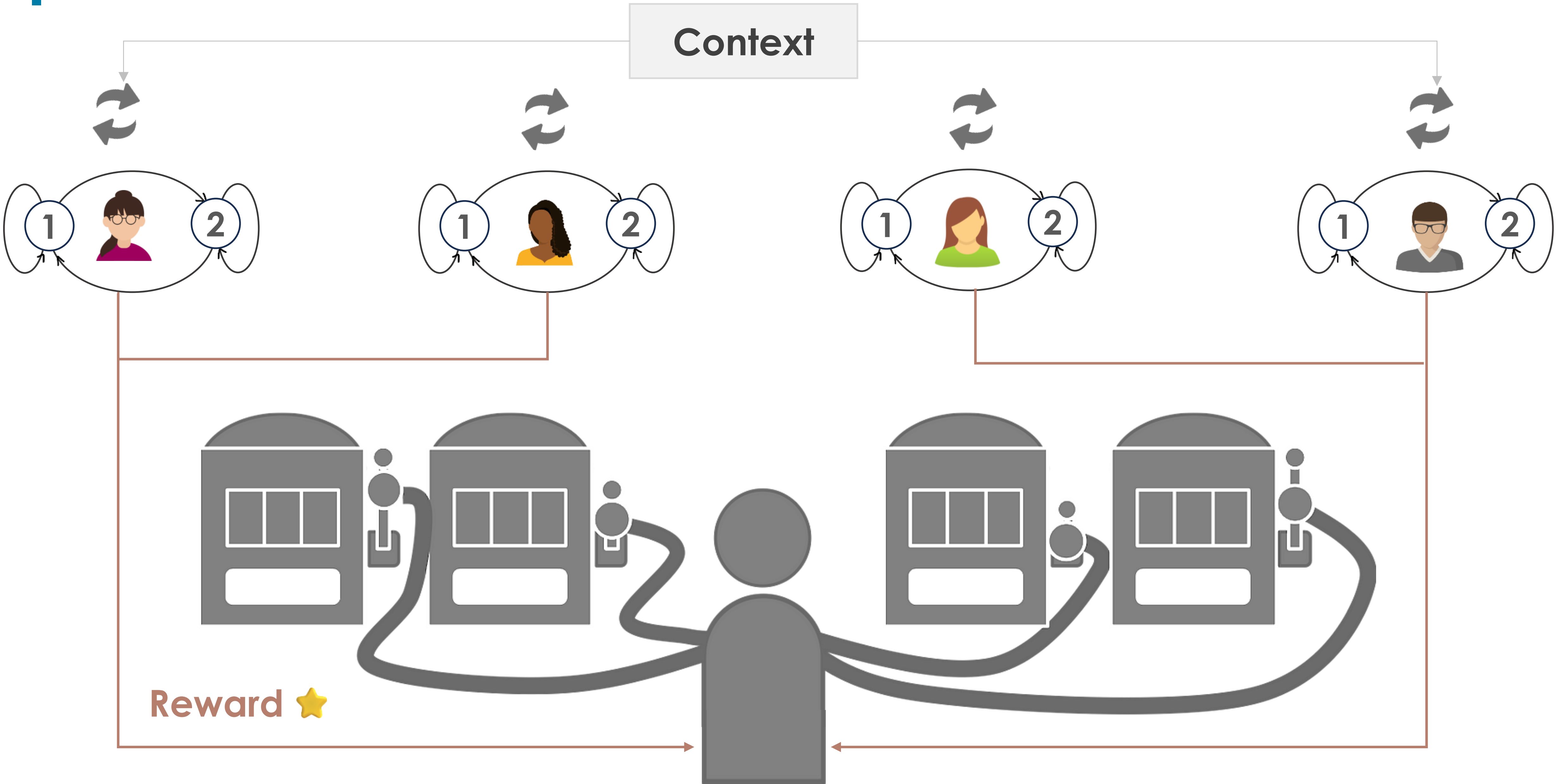


~ 46 % of emails are spams (2023)

statista

CONTEXTUAL RESTLESS BANDITS

EMAIL RS



CONTEXTUAL RESTLESS BANDITS

EMAIL RS

- Choose M users out of N.
- Actions: to send an email or not
- Items: promotional email.
- States: interactions of users:
  - Idle
  - Open
  - Click
  - Purchase

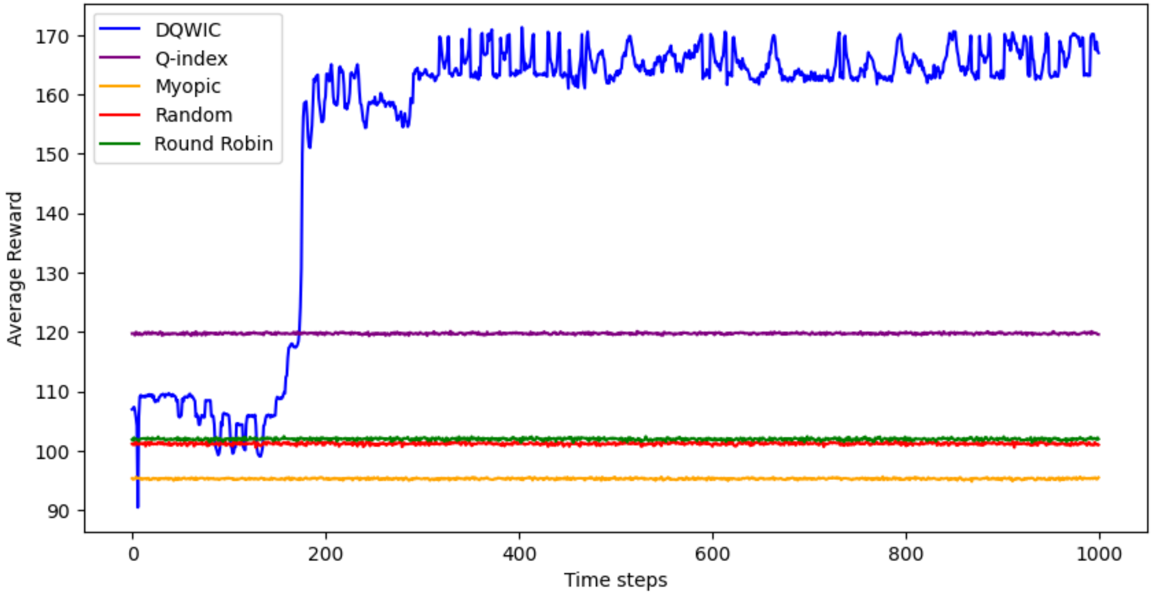
EMAIL RECOMMENDER SYSTEM

BASELINES

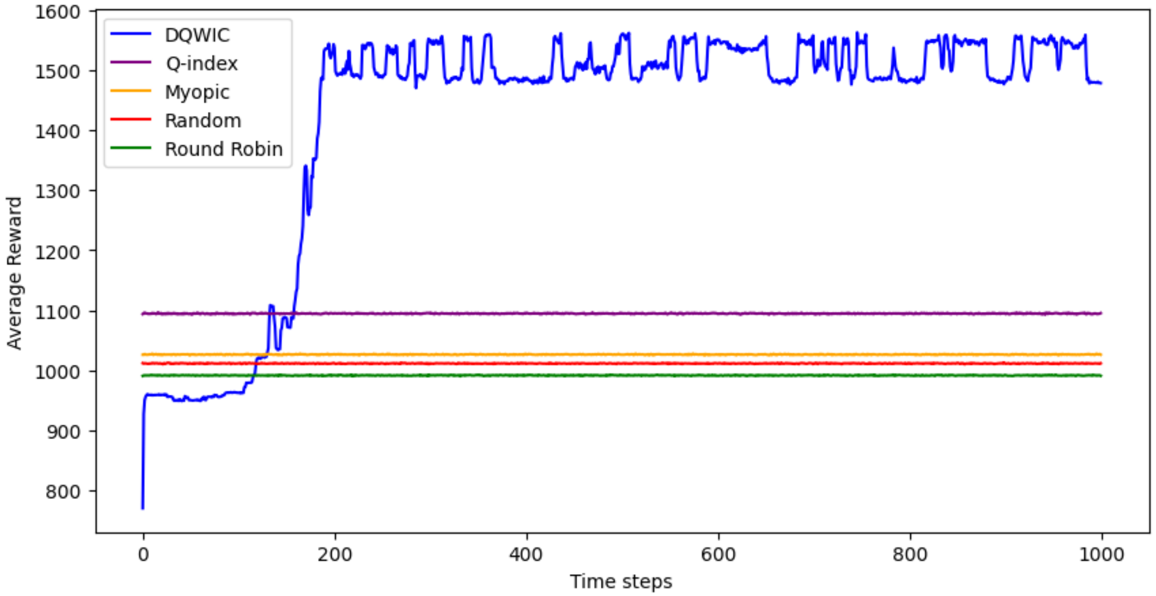
Policy	Definition
Random	Selects users randomly without considering engagement information.
Myopic	Selects users who are most likely to lead to immediate conversions, such as those who have recently opened emails.
Round-robin	Selects to contact users sends emails evenly in a cyclic order.
Q-index	The users are ranked according to the active Q-values approximated by a deep neural network.

EMAIL RECOMMENDER SYSTEM
RESULTS

- **Q-network:** 2 hidden layers with 130 and 50 neurons, respectively, connected by ReLU.
- **Whittle network:** 1 hidden layer with 250 neurons, connected by ReLU.



Real Data | N = 100, M = 10



Synthetic Data | N = 1000, M = 100

CONCLUSION

- We proposed deep Q-learning with Whittle index for **CRMAB**, and applied it to an **email recommender system.**
- One important feature of our algorithm is the usage of only three neural networks for many (potentially thousands) heterogeneous arms. This is possible thanks to the **addition of context to the RMAB model**.
- **Experiments** on both **synthetic and real-world data** show that DQWIC outperforms existing baselines.
- **Future work** will focus on incorporating multiple actions, and enhancing fairness.

THANK YOU

REFERENCES

[1] X. Chen, I. Hou, et al. “Contextual Restless Multi-Armed Bandits with Application to Demand Response Decision-Making”. In: arXiv preprint arXiv:2403.15640 (2024)

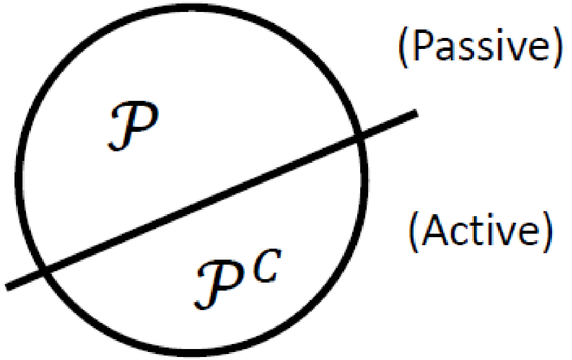
[2] B. Liang, L. Xu, A. Taneja, M. Tambe, and L. Janson. “A Bayesian Approach to Online Learning for Contextual Restless Bandits with Applications to Public Health”. In: arXiv preprint arXiv:2402.04933 (2024).

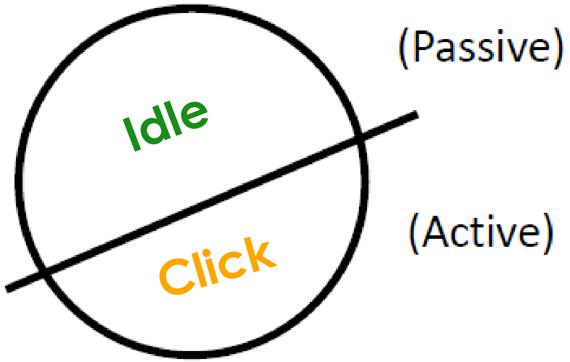
WHITTLE INDEX

SOLUTION TO THE DECOUPLED PROBLEM

- In general, the optimal policy divides the state space into two subsets:
    - Let $\mathcal{P}(\tilde{\lambda})$ be the set of ALL states for which it is **optimal to be idle** when the playing charge is $\tilde{\lambda}$.
    - Let $\mathcal{P}^c(\tilde{\lambda})$ be the set of ALL states for which it is **optimal to be in click** when the playing charge is $\tilde{\lambda}$.
- **Optimal policy:** play, if $s_t^{(i)} \in \mathcal{P}^C(\tilde{\lambda})$; stop, otherwise.

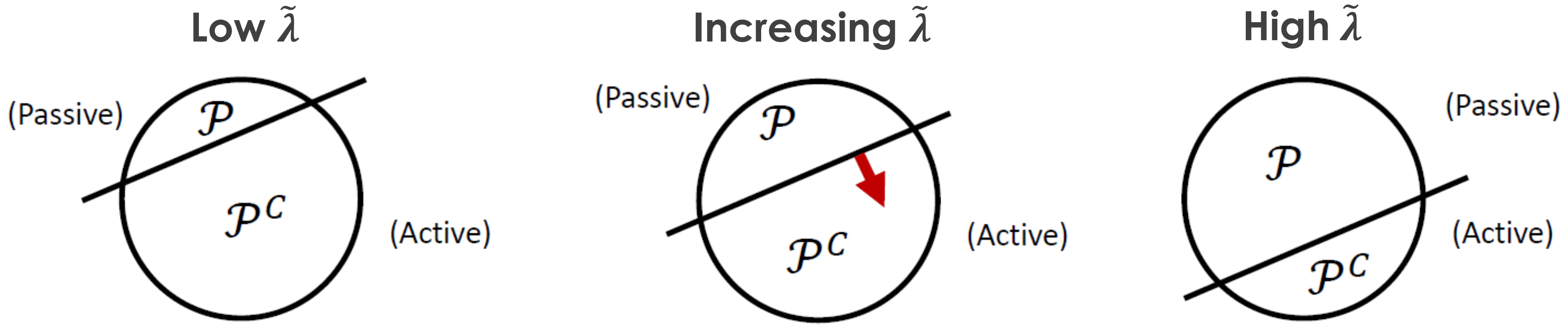
State space with $\tilde{\lambda}$





WHITTLE INDEX
INDEXABILITY

- An arm α is considered to be indexable if the set P increases monotonically from Φ to the entire state space as we increase $\tilde{\lambda}^{\alpha}$ (the subsidy for passive action).
- The RMAB problem is indexable if the Decoupled Problem is indexable for all bandits.

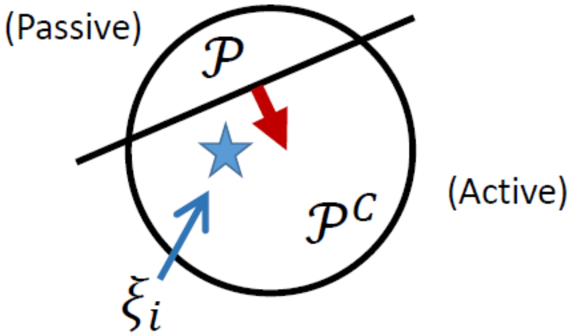


WHITTLE INDEX

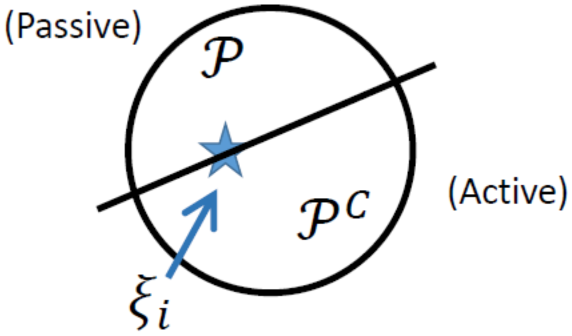
DEFINITION

- For an arm α , Whittle index is defined as the $\lambda^{\alpha} = \inf \{\lambda^{\alpha} : \alpha \in P\}$.
- It is the infimum subsidy λ^{α} one has to pay so that it is equally desirable to give an arm α active and passive action.

Increase $\tilde{\lambda}$

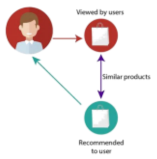


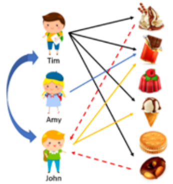
Until $\tilde{\lambda}$' is such that



RECOMMENDER SYSTEM
ALGORITHMS

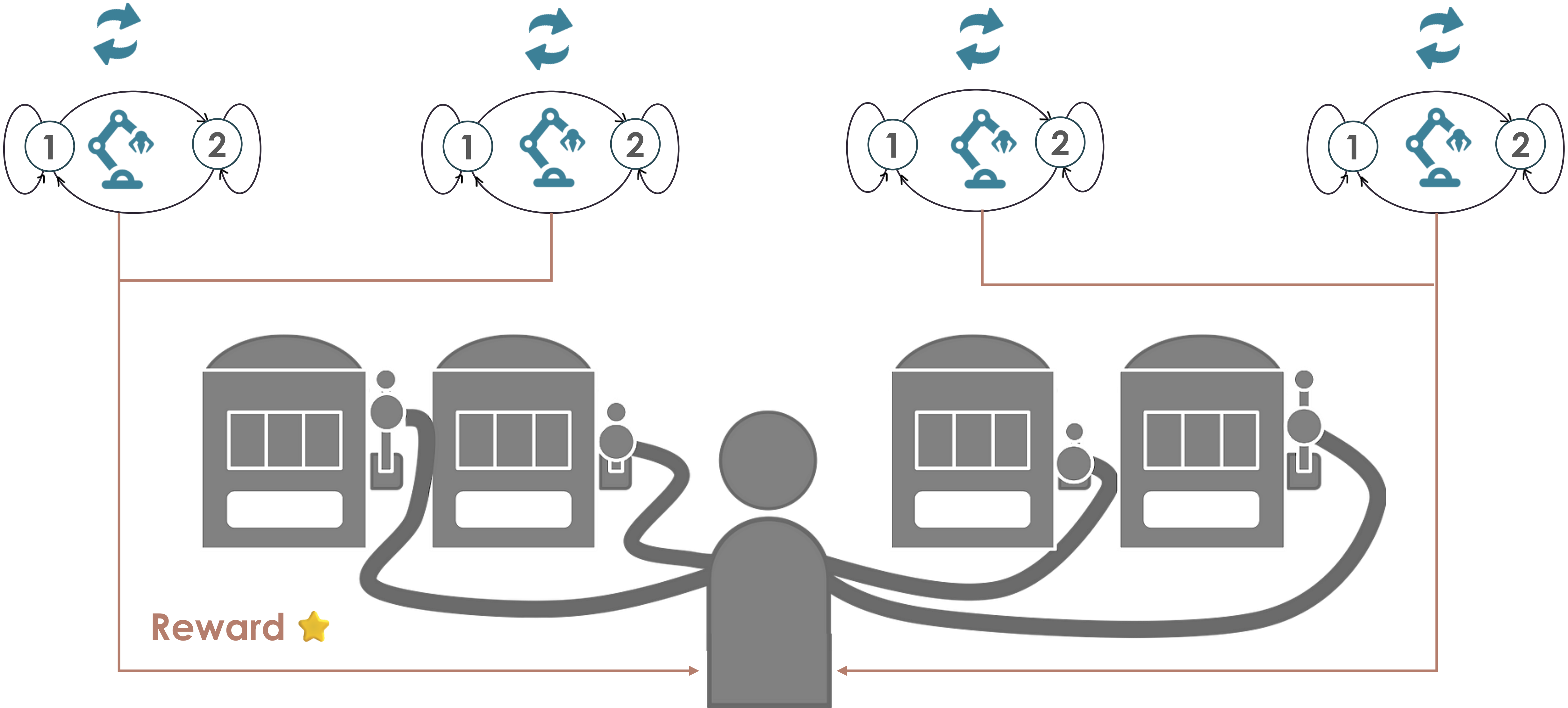
- Recommender System
  - Content-Based Filtering
  - Collaborative Filtering
    - Model-Based Filtering
      - Matrix Factorization
      - Clustering Approaches
      - Neural Networks
    - Memory-Based Filtering
      - Item-Based
      - User-Based
  - Hybrid Approaches





RESTLESS BANDITS

RECALL



APPLICATION
RECOMMENDER SYSTEMS

- A Recommender System (RS) is:
  - an information filtering system
  - suggests items to users based on their past preferences or behavior.
  - used in many domains such as in: e-commerce, entertainment, and social media, etc.
- Let $R(u, j)$ represent the interaction between user u and item j. This could be:
  - **Explicit:** Ratings (e.g., $R(u, j) \in \{1, 2, \ldots, 5\}$).
  - **Implicit: Binary clicks or preferences (e.g., $R(u, j) \in \{0, 1\}$).**