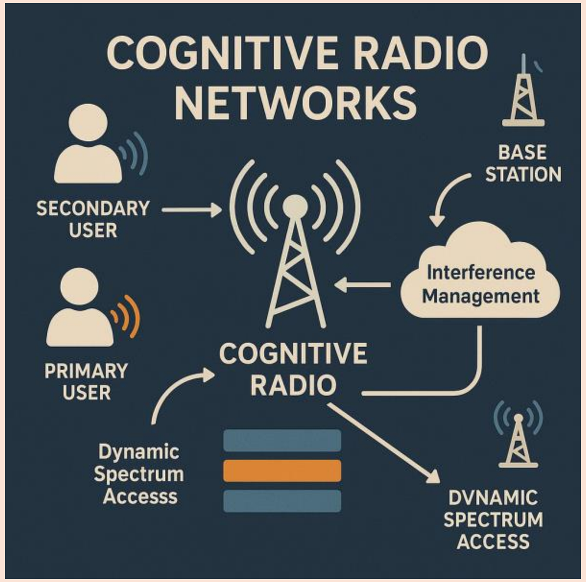
Asynchronous Multi-Agent Bandits: Fully Distributed vs. Leader-Coordinated Algorithms

Xuchuang Wang[1*], Janice Yu-Zhen Chen[1*], Xutong Liu[2], Lin Yang[3], Mohammad Hajiesmaili[1], Don Towsley[1], John C.S. Lui[4]

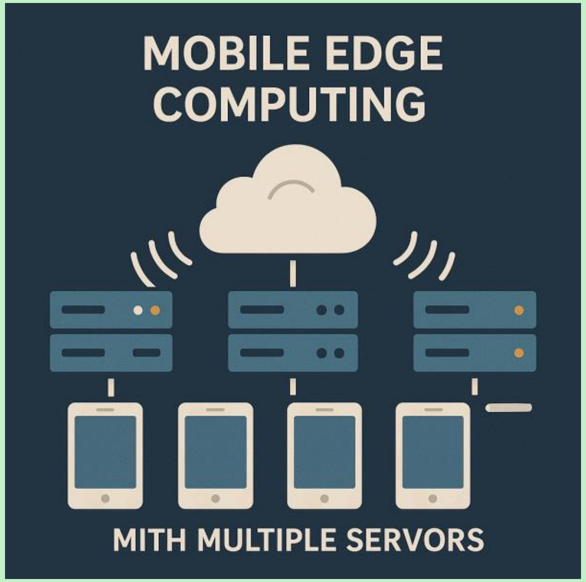
[1]University of Massachusetts Amherst, [2]Carnegie Mellon University, [3]Nanjing University, [4]Chinese University of Hong Kong

SIGMETRICS 2025

Asynchronous Multi-Agent Decision-Making Systems



Fully Distributed Systems



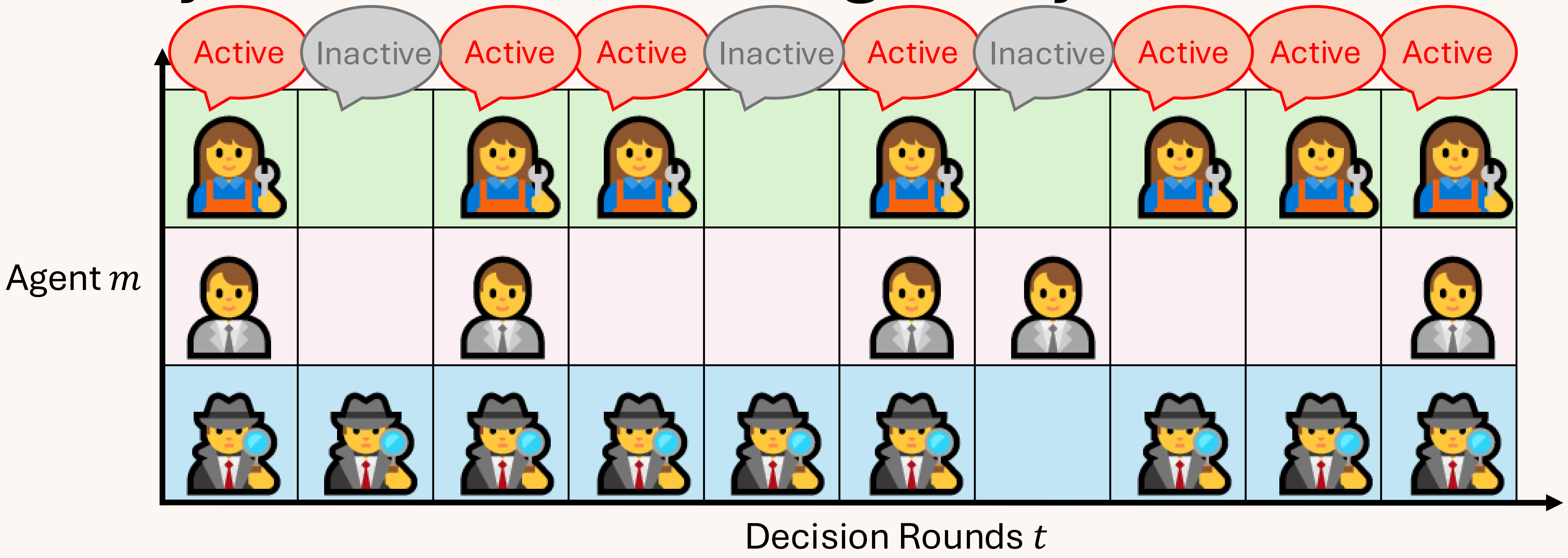
Leader-Coordinated Systems

Question: Can asynchronous agents cooperate efficiently ?

Outline

- 📷 **Model:**
  - Asynchronous Multi-Agent Bandits
  - Group Regret & Communication
- 🤖 Algorithm:
  - Fully Distributed: `SE-AAC-ODC`
  - Leader-Coordinated: `LF-Relay`
- ⚖️ Comparison:
  - Theoretical
  - Empirical

Asynchronous Multi-Agent System

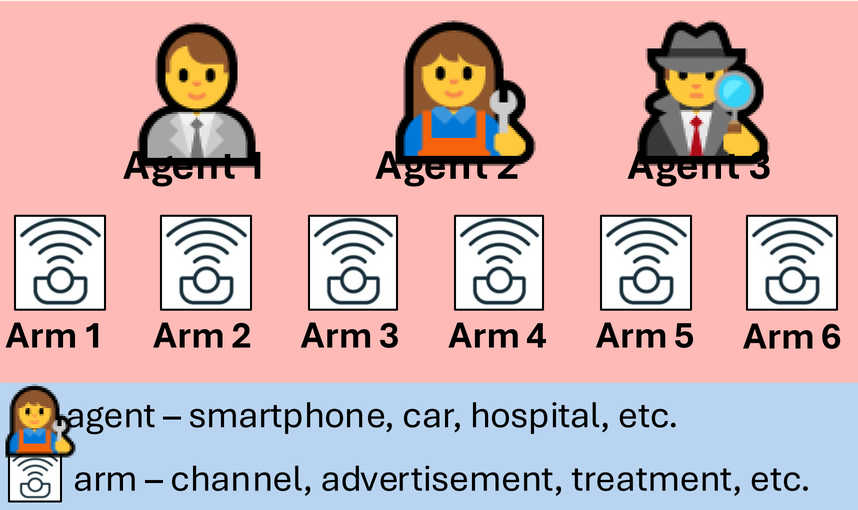


For each agent $m \in \mathcal{M}$: Set of active decision rounds $\mathcal{T}^{(m)}$

is **Unknown and Arbitrary** (or even *oblivious adversary*)

Chen, Yu-Zhen Janice, et al. "On-demand communication for asynchronous multi-agent bandits." *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023.

Asynchronous Multi-Agent Bandits: Setting



- K Arms -- each arm k has a reward *dist*. with **unknown** mean μ_k
  - Optimal arm: $k^* = \mathrm{argmax}_{k \in \mathcal{K}}\, \mu_k$
- For each agent $m \in \mathcal{M}$ & each **active** decision round $t \in \mathcal{T}^{(m)}$,
  - Select one arm $I_t^{(m)} \in \mathcal{K}$ to pull;
  - Observe the reward realization $X_{I,t}^{(m)}$ of the pulled arm.

Asynchronous Multi-Agent Bandits: Objective

- Group Regret:

Optimal Reward

Multi-agent's Total rewards

$$R_T \coloneqq \sum_{m\in\mathcal{M}} \left(T^{(m)} \mu_{k^*} - \sum_{t\in\mathcal{T}^{(m)}} \mu_{I_t^{(m)}} \right)$$

- Communication:

$$C_T \coloneqq \sum_{m\in\mathcal{M}} \sum_{t\in\mathcal{T}^{(m)}} \mathbb{I}[\text{Agent } m \text{ send/receive message at round } t]$$

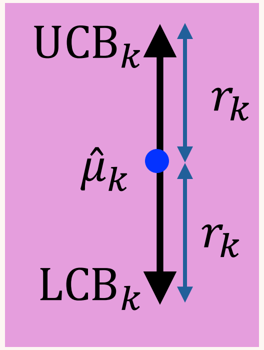
- Aim to minimize both objectives.

Outline

- Model:
  - Asynchronous Multi-Agent Bandits
  - Group Regret & Communication
- **Algorithm:**
  - **Fully Distributed: `SE-AAC-ODC`**
  - Leader-Coordinated: `LF-Relay`
- Comparison:
  - Theoretical
  - Empirical

- Arm-Pulling Policy (**`SE`**)
- Communication Policy
  - When to Communicate (**`AAC`**)
  - Who to Communicate with (**`ODC`**)

When to Comm: **AAC**–Accuracy Adaptive Comm.



Communication trigger:

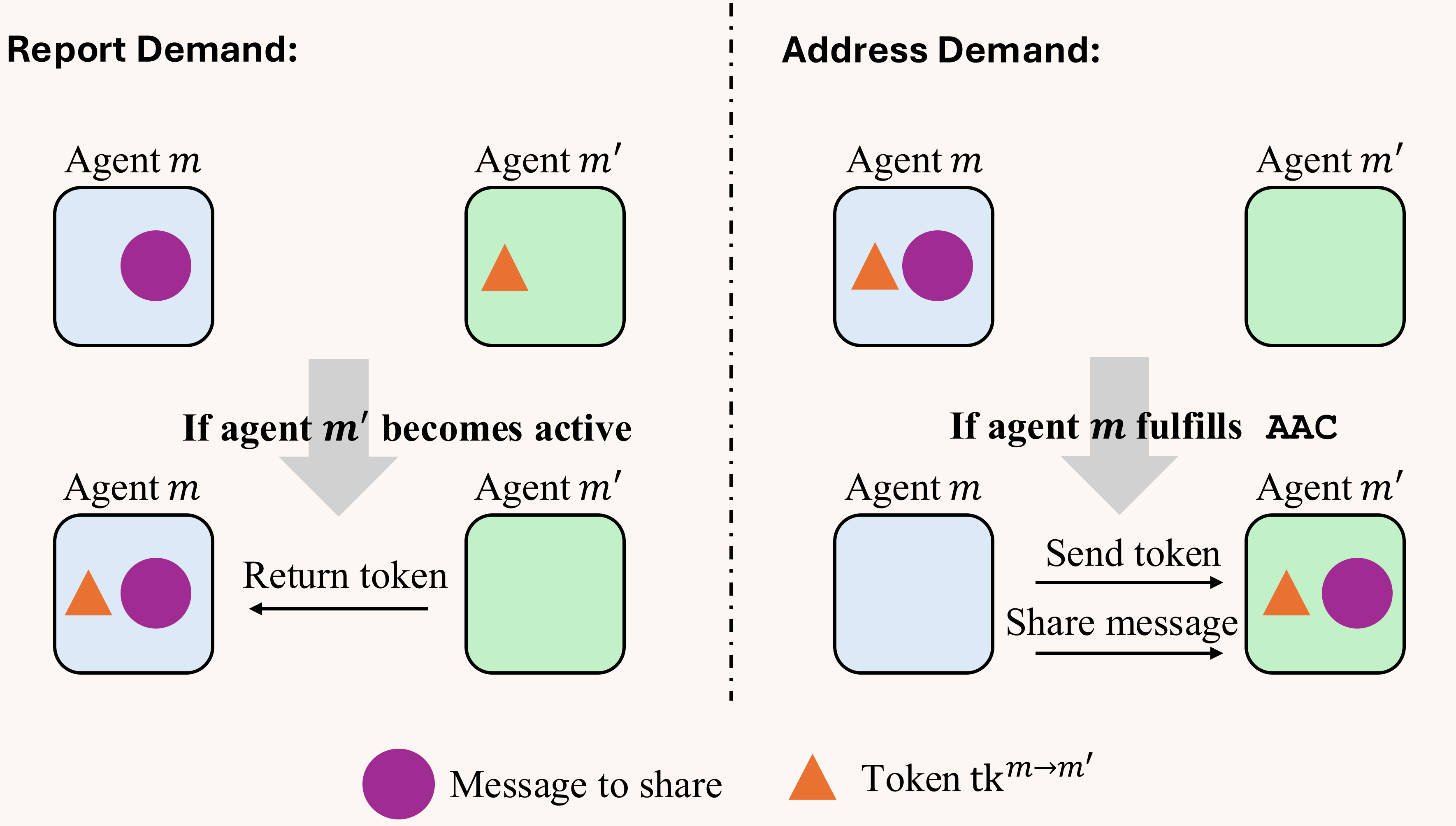
- Confidence radius r_k as accuracy measure.
- When the accuracy improves $\alpha > 1$ times, communicate.

If $\alpha r_{I_t^{(m)}} \le r_{\text{last}, I_t^{(m)}}$ **then** communicate info. of arm $I_t^{(m)}$

- The fastest agent (since last comm.) communicates more frequently.
- Communicate to ALL agents? $\Longrightarrow$ Redundant / Inefficient

Yang, Lin, et al. "Cooperative multi-agent bandits: Distributed algorithms with optimal individual regret and constant communication costs." *arXiv preprint arXiv:2308.04314* (2023).

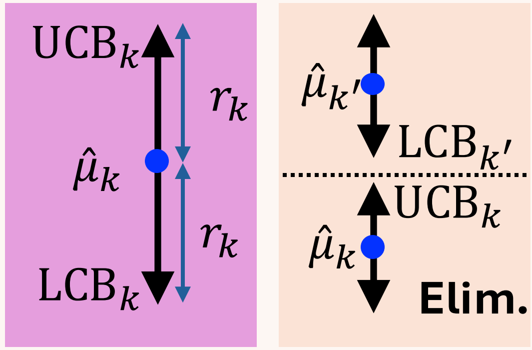
Who to Comm. with: **ODC**–On-Demand Comm.



Chen, Yu-Zhen Janice, et al. "On-demand communication for asynchronous multi-agent bandits." *International Conference on Artificial Intelligence and Statistics*. PMLR, 2023.

Fully Distributed Algorithm (**SE-AAC-ODC**)

- Arm-Pulling Policy (**SE**: Successive Elimination)



Maintain candidate arm set $\mathcal{C}$ (potential good arms)

For each active decision round:

- **Decision**: Pull arms from the candidate arm set in a round-robin manner
- **Elimination**: Remove arm k from $\mathcal{C}$ whose $\mathrm{UCB}_k < \max_{k' \in \mathcal{C}} \mathrm{LCB}_{k'}$

If remove arm k from candidate set $\mathcal{C}$ **then** notify the elimination.

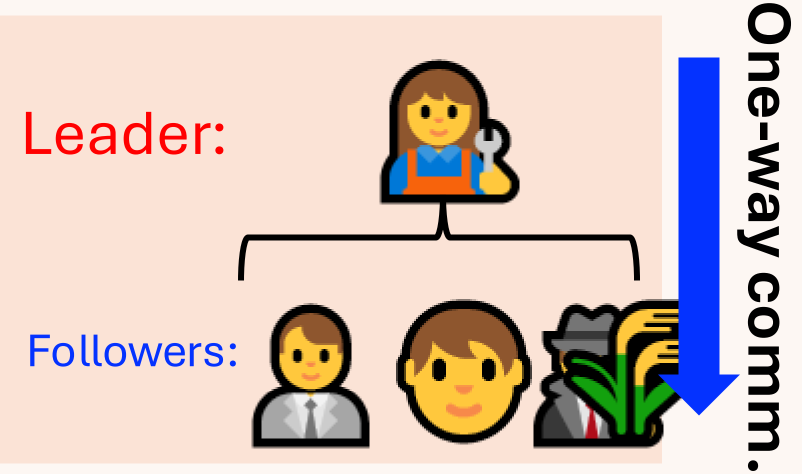
- Communication Policy
  - When to communicate (**AAC**) + Who to communicate with (**ODC**)

Outline

- 📷 Model:
  - Asynchronous Multi-Agent Bandits
  - Group Regret & Communication
- 🤖 **Algorithm:**
  - Fully Distributed: `SE-AAC-ODC`
  - **Leader-Coordinated: `LF-Relay`**
- ⚖️ Comparison:
  - Theoretical
  - Empirical

Leader-Follower Scheme

- One agent as the Leader
    1. **Explore** when needed
    2. **Recommend** arm for followers to pull
- Other $M - 1$ agents as Followers
    3. **Exploit**: just pull the recommended arm.

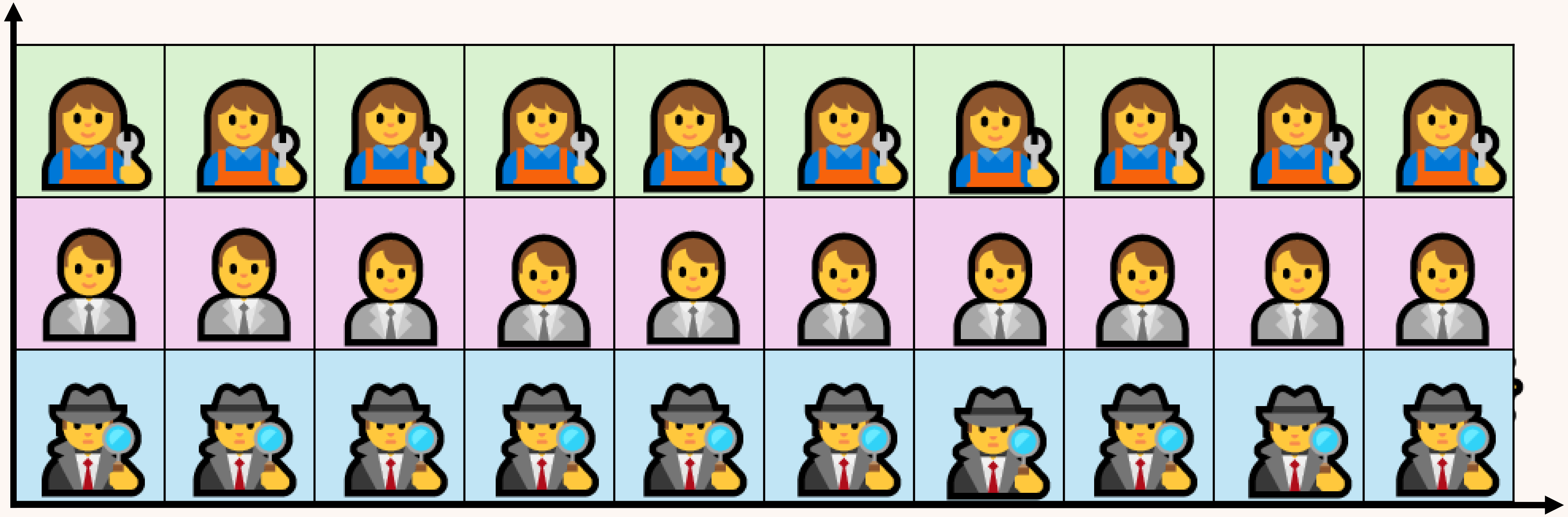


The leader needs to be active *frequently* for exploration.

Wang, Po-An, et al. "Optimal algorithms for multiplayer multi-armed bandits." *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020.

The leader needs to be active *frequently* for exploration.

Leader-Follower Scheme



- For synchronous, any agent could be the leader.
- For asynchronous, hard to find a single competent leader.

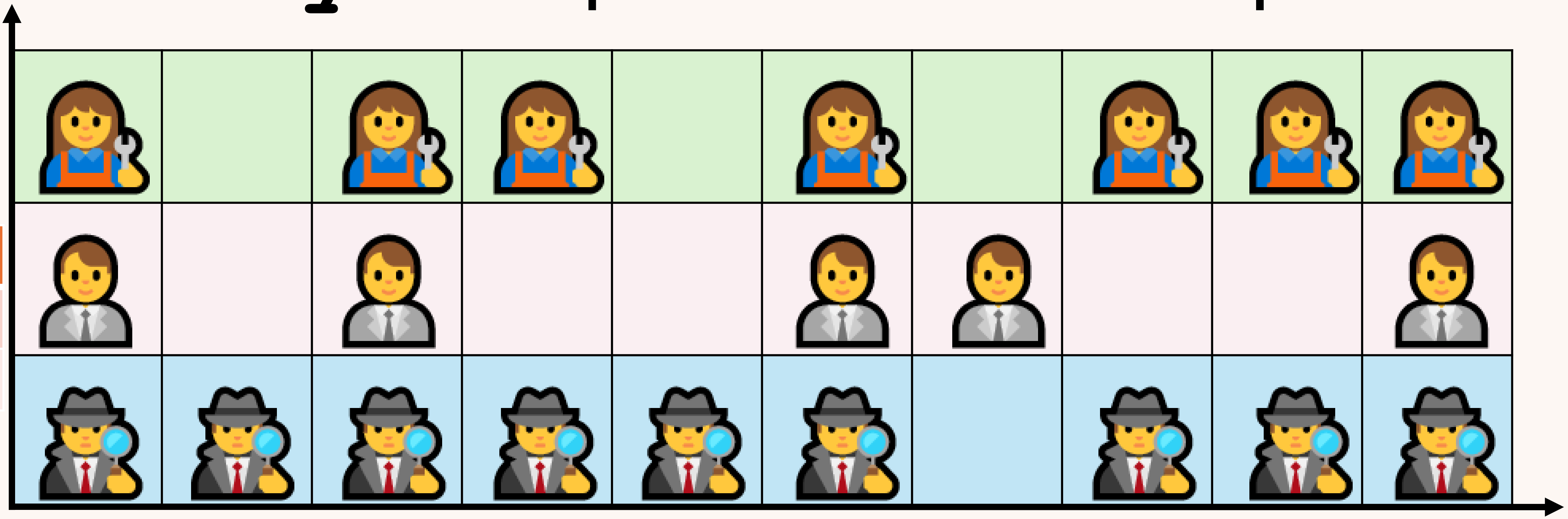
A sequence of agents as the Leader! But active rounds are unknown!

A sequence of agents as the Leader!

Leader `Relay`: Competent Leader Sequence

"Arm" m Agent m

Reward	Status
1	Active
0	Inactive



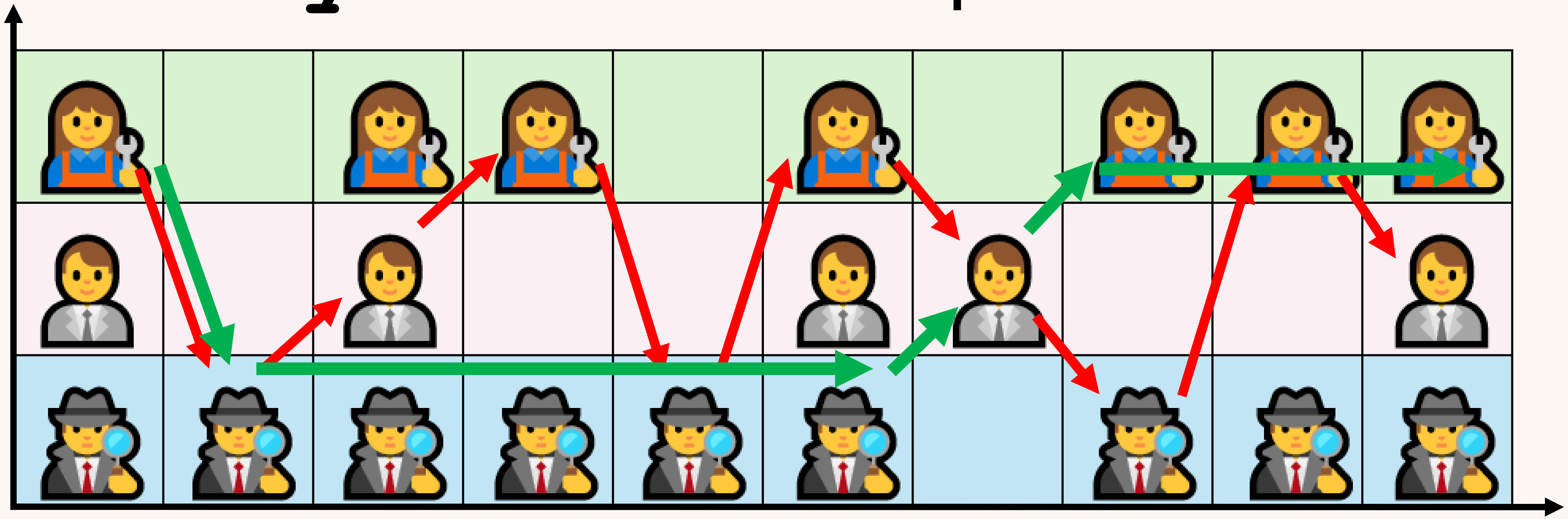
Adversary Bandit Problem

- Adversarial reward sequence: Binary active/inactive status
- Maximize total reward: Maximize #active status
- Arm-pulling sequence: Leader Sequence

Auer, Peter, et al. "The nonstochastic multiarmed bandit problem." SIAM journal on computing 32.1 (2002): 48-77.

Changing leadership frequently can incur linear communication costs.

Leader `Relay`: Low Leadership Switch



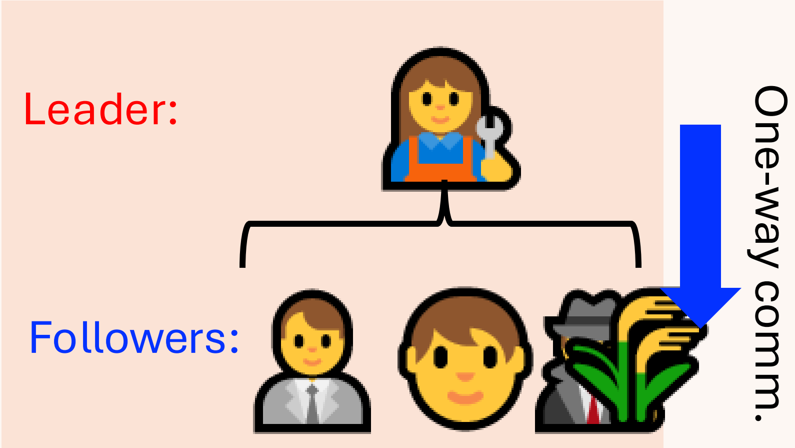
Adversary Bandits with Low Switch Cost

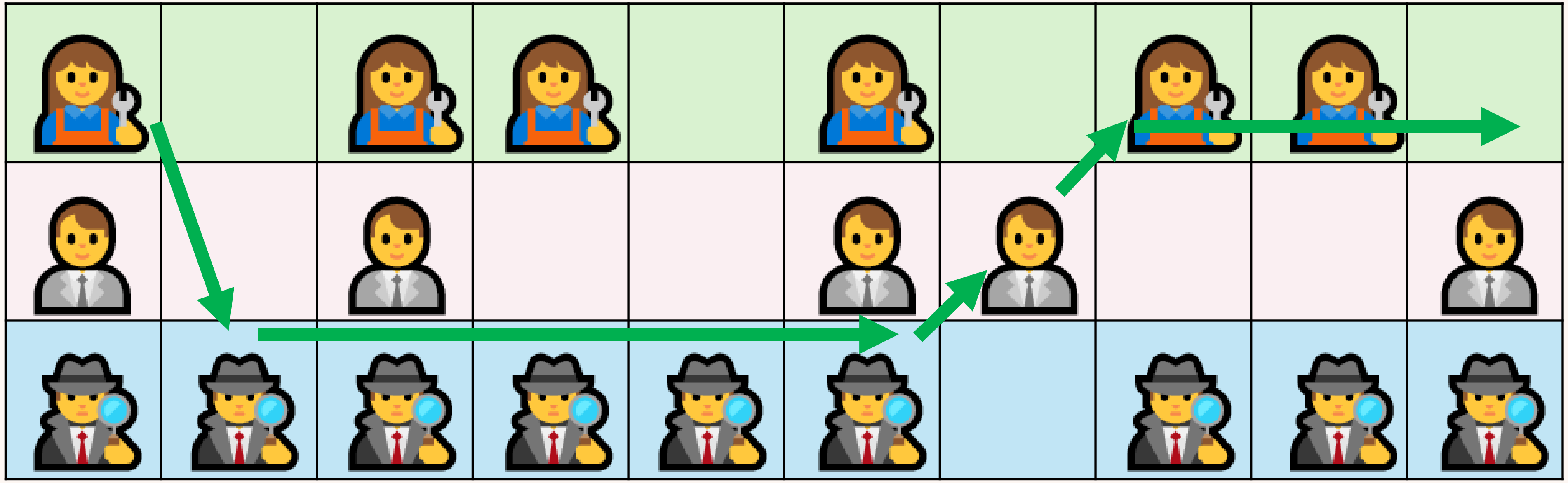
- **Mini-Batch**: Split into S batches, fix leader for each $\frac{T}{S}$-batch
- **`Tsallis-INF`** ($4\sqrt{MT}$ regret): Switch $S = 64M^3$ is enough!

- Zimmert, Julian, and Yevgeny Seldin. "Tsallis-inf: An optimal algorithm for stochastic and adversarial bandits." *Journal of Machine Learning Research* 22.28 (2021): 1-49.
- Altschuler, Jason M., and Kunal Talwar. "Online Learning over a Finite Action Set with Limited Switching." *Mathematics of Operations Research* 46.1 (2021): 179-203.

Leader-Coordinated Algorithm: Leader-Follower + Leader-Relay

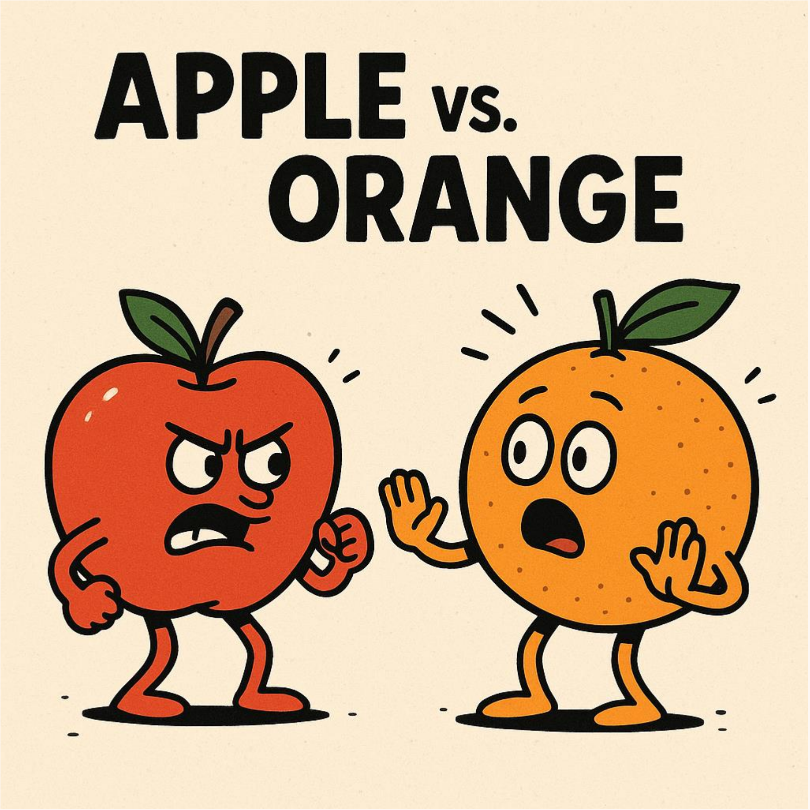
- One agent as the Leader
    1. **Explore** when needed
    2. Recommend arm for followers to pull
- Other $M-1$ agents as Followers
    - Exploit: just pull the recommended arm.





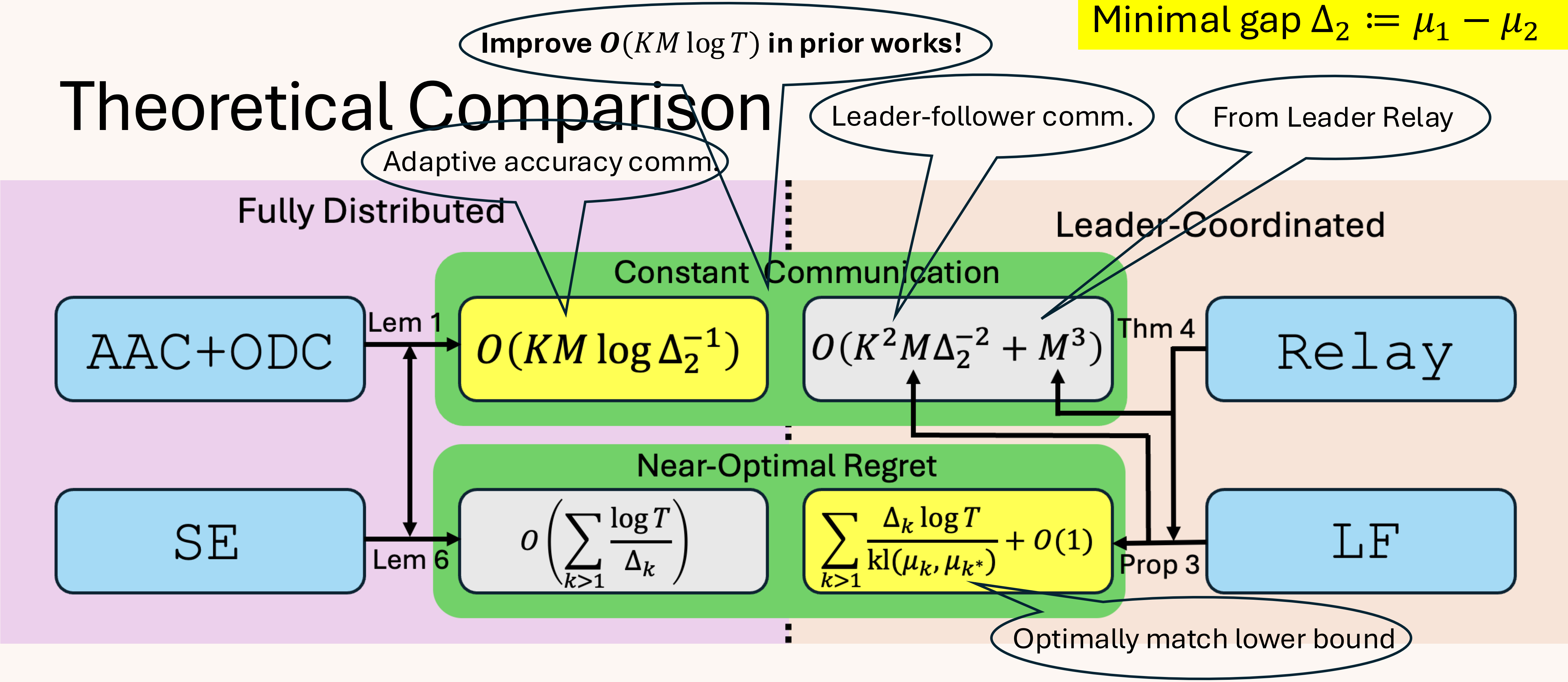
Outline

- 📷 Model:
  - Asynchronous Multi-Agent Bandits
  - Group Regret & Communication
- 🤖 Algorithm:
  - Fully Distributed: `SE-AAC-ODC`
  - Leader-Coordinated: `LF-Relay`
- ⚖️ **Comparison:**
  - Theoretical
  - Empirical



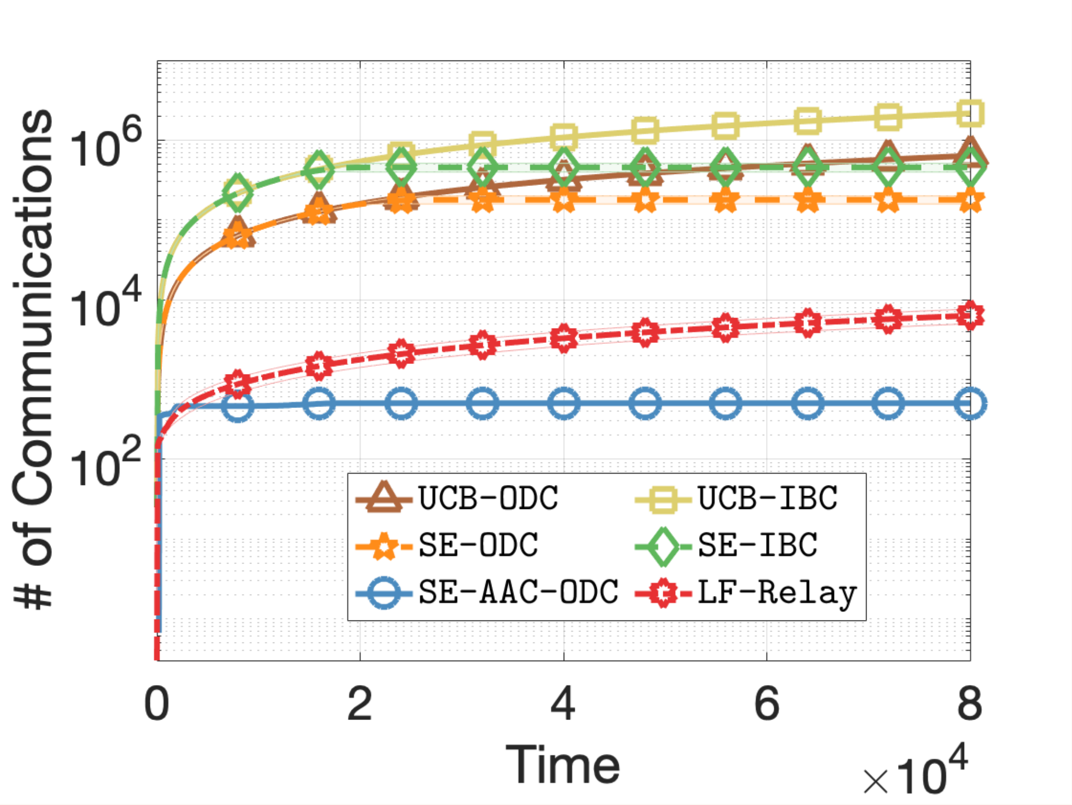
Minimal gap $\Delta_2 := \mu_1 - \mu_2$

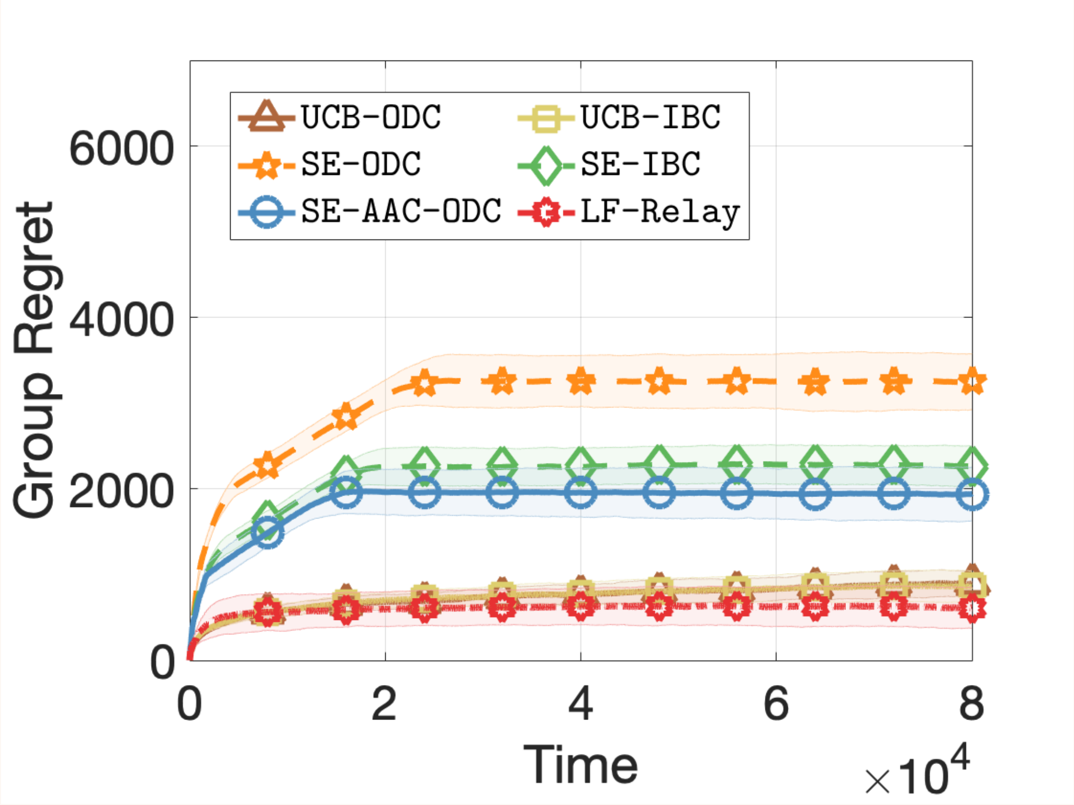
Theoretical Comparison



- The **leader-coordinated** algorithm achieves the **optimal regret**.
- The **fully distributed** algorithm enjoys lower **communication** costs.

Empirical Comparison





SE-AAC-ODC: lowest communication!

LF-Relay: lower regret!

Thank you!

Summary

- ⚔️ Fully distributed algorithm
  - Better Communication
  - Challenge: When to Comm. & Who to Comm. with
  - Technique: Accuracy Adaptive + On-Demand
- 🚀 Leader-coordinated algorithm
  - Better Group Regret
  - Challenge: Choose Competent Leaders
  - Technique: Leader Relay (Adversary Bandits) + Leader-Follower