

# Improved Sleeping Bandits with Stochastic Actions Sets and Adversarial Rewards

Aadirupa Saha<sup>1</sup>, Pierre Gaillard<sup>2</sup>, Michal Valko<sup>3</sup>

1. Indian Institute of Science, Bangalore

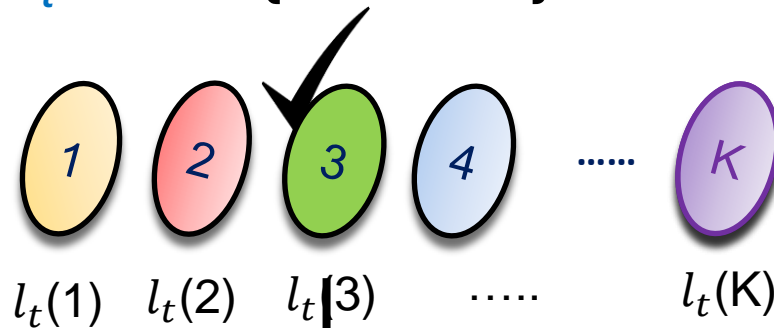
2. Inria, Paris

3. Deepmind, Paris

International Conference in Machine Learning, 2020

## More formally: Adversarial MAB

At round  $t$ ,  
Select an arm  $i_t$  from  $\{1, 2, \dots, K\}$



repeat

Observe (noisy) loss  $\ell_t(it) \in [0, 1]$

Expected Regret in T rounds:

$$R_T = \sum_{t=1}^T \ell_t(it) - \ell_t(i^*)$$


State of the art:  $\theta(\sqrt{KT})$

- EXP3 Algorithm

---

## Many Applications:

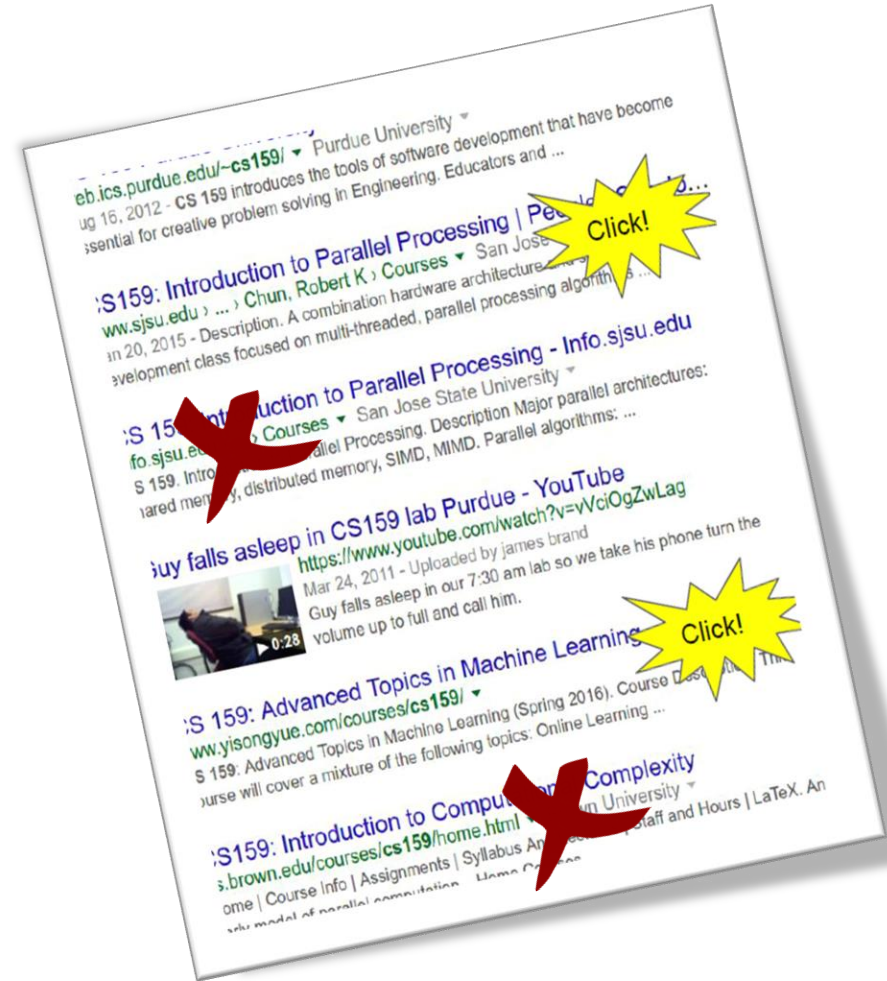
- Clinical Trials
- Wireless Communication
- Social Networks
- Search Engine Optimization
- Recommender Systems
- many more ...



But fixed decision  
(arm) space is often  
**Unrealistic**

**EXP3 fails!**

# Search Engine Optimization:



# Recommender Systems

The screenshot shows a mobile application interface for "New York Restaurants". At the top left, a user profile for "SAM MICHAELS" is visible, with 46 messages and 132 likes, and a "Recompute Your Finds" button. Below the profile is a "FINDS [50]" section with the text "What Nara found for you". A "Filter:" section includes price levels (\$, \$\$, \$\$\$, \$\$\$\$), restaurant reservation services (OpenTable, grubHub), neighborhood selection, cuisine types (10), and friends (2). The main content area displays a grid of restaurant cards. Each card features a photo of the restaurant or food, the restaurant name, location, cuisine, price range, phone number, and an "Order Online" button. Some cards also have a "Reserve" button. The cards shown include: "SIDEWALK CAFE" (East Village, Downtown, Manhattan), "NUMERO 28 PIZZERIA NAPOLETANA" (Downtown, East Village, Manhattan), "JG MELON" (Uptown, Upper East Side, Manhattan), "JOHN'S OF 12TH STREET" (Downtown, East Village, Manhattan), "NICK'S PIZZA" (Queens), and "KESTE PIZZA & VINO". A large red diagonal watermark reading "Best restaurant family for dinner?" is overlaid across the center of the screen, with red 'X' marks over several restaurant cards.



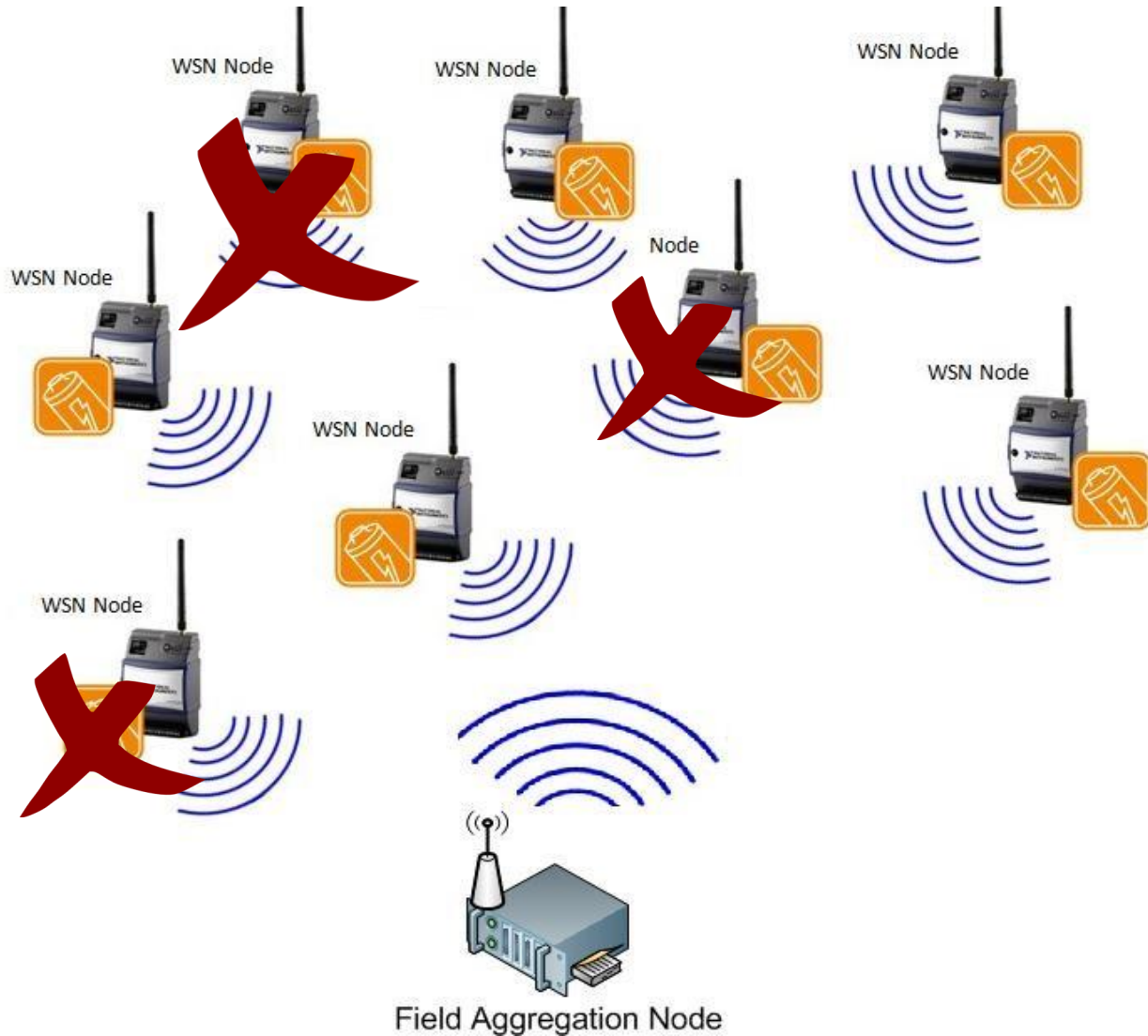
Retail Chain



**Guess the most liked flavour?**



# Wireless Communication



Identify the Sensor with  
highest **transmission rate**?

---

# *Sleeping* Bandits



# *Sleeping* Bandits

At round  $t = 1$



Arm-1



Arm-2



Arm-3

.....



Arm-K

Unavailable (Cannot be picked !!)



---

# *Sleeping* Bandits

At round  $t = 2$



Arm-1



Arm-2



Arm-3

.....



Arm-K

# *Sleeping* Bandits

Type of Availabilities:

1. Stochastic

2. Adversarial

At round  $t = 3$ , and so on.....



Arm-1



Arm-2



Arm-3

.....

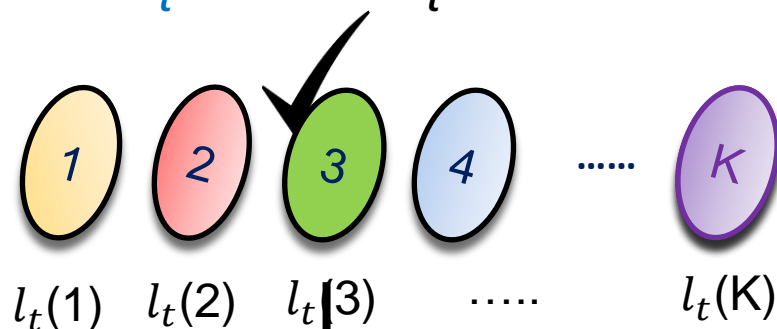


Arm-K

# Formally: Adversarial MAB + Stochastic Availabilities

At round  $t$ ,

Select an arm  $i_t$  from  $S_t$



repeat

Observe (noisy) loss  $\ell_t(i_t) \in [0,1]$

Regret in  $T$  rounds:

$$\max_{\pi: 2^{[K]} \mapsto [K]} \mathbf{E} \left[ \sum_{t=1}^T \ell_t(i_t) - \sum_{t=1}^T \ell_t(\pi(S_t)) \right]$$

---

# Existing Results

EXP4 algorithm:  $O(T^{1/2})$

Computationally Inefficient

Kanade et al (2009):  $O(T^{4/5})$

Neu et al (2014):  $O(T^{2/3})$

We achieved optimal dependence on  $T$ :  $O(\sqrt{T})$  and computationally efficient

Kanade et al. Sleeping experts and bandits with stochastic action availability and adversarial rewards. AISTATS 2009

G. Neu, M. Valko. Improved Sleeping Bandits with Stochastic Actions Sets and Adversarial Rewards. NIPS 2014



---

## Problem: With Independent Availabilities

Availability Vectors:  $\{a_i\}_{i \in [K]}$

At any time  $t$ :  $\mathbf{1}(i \in S_t) \sim \text{Ber}(a_i)$

---

# Our Algorithm (Independent Availabilities)

## Sleeping-EXP3

# Sleeping-EXP3 (Independent Availabilities)

$$p_1(i) = \frac{1}{K}, \forall i \in [K]$$

**while**  $t = 1, 2, \dots$  **do**

Define  $q_t^S(i) := \frac{p_t(i)\mathbf{1}(i \in S)}{\sum_{j \in S} p_t(j)}$ ,  $\forall i \in [K], S \subseteq [K]$

Receive  $S_t \subseteq [K]$

Sample  $i_t \sim \mathbf{q}_t^{S_t}$

Receive loss  $\ell_t(i_t)$

Compute:  $\hat{a}_{ti} = \frac{\sum_{\tau=1}^t \mathbf{1}(i \in S_\tau)}{t}$

← Estimated availability

$$P_{\hat{\mathbf{a}}}(S) = \prod_{i=1}^K \hat{a}_{ti}^{\mathbf{1}(i \in S)} (1 - \hat{a}_{ti})^{1 - \mathbf{1}(i \in S)}$$

$$\bar{q}_t(i) = \sum_{S \in 2^{[K]}} P_{\hat{\mathbf{a}}}(S) q_t^S(i)$$

← Item Probability

Estimate loss:  $\hat{\ell}_t(i) = \frac{\ell_t(i)\mathbf{1}(i=i_t)}{\bar{q}_t(i) + \lambda_t}$ ,  $\forall i \in [K]$

← Loss Estimate

Update  $p_{t+1}(i) = \frac{p_t(i)e^{-\eta \hat{\ell}_t(i)}}{\sum_{j=1}^K p_t(j)e^{-\eta \hat{\ell}_t(j)}}$ ,  $\forall i \in [K]$

← EXP3 Update

**end while**

$$\text{Regret: } \tilde{O}(K^2 \sqrt{T})$$

---

# Our Algorithm (General Availabilities)

## Sleeping-EXP3G

# Sleeping-EXP3G (General Availabilities)

**while**  $t = 1, 2, \dots$  **do**

Receive  $S_t$

Compute  $q_t(i) = \frac{p_t(i)\mathbf{1}(i \in S_t)}{\sum_{j \in S_t} p_t(j)}$ ,  $\forall i \in [K]$

Sample  $i_t \sim \mathbf{q}_t$

Receive loss  $\ell_t(i_t)$

Compute  $\bar{q}_t(i) := \frac{1}{t} \sum_{\tau=1}^t q_t^{S_\tau}(i)$

Estimate loss bound  $\hat{\ell}_t(i) = \frac{\ell_t(i)\mathbf{1}(i=i_t)}{\bar{q}_t(i) + \lambda_t}$

Update  $p_{t+1}(i) = \frac{p_t(i)e^{-\eta\hat{\ell}_t(i)}}{\sum_{j=1}^K p_t(j)e^{-\eta\hat{\ell}_t(j)}}$ ,  $\forall i \in [K]$

**end while**

Regret:  $\tilde{O}(\sqrt{2^K T})$



Item Probability



Loss Estimate



EXP3 Update

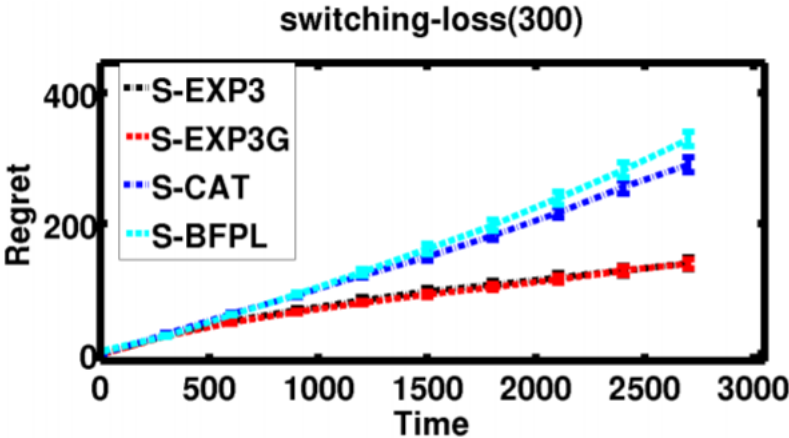
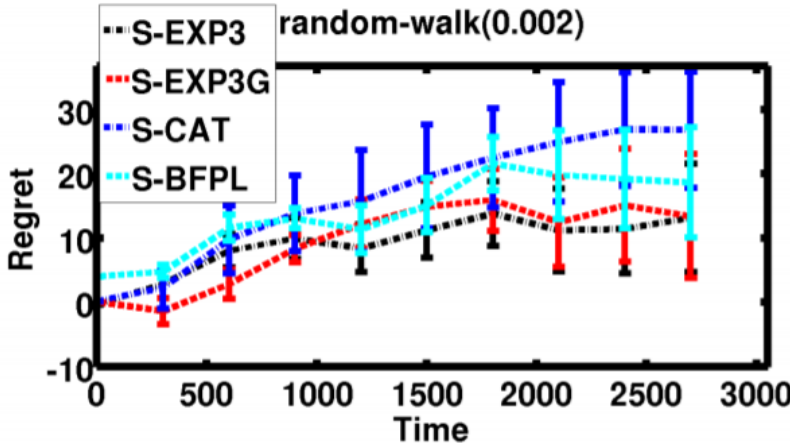


---

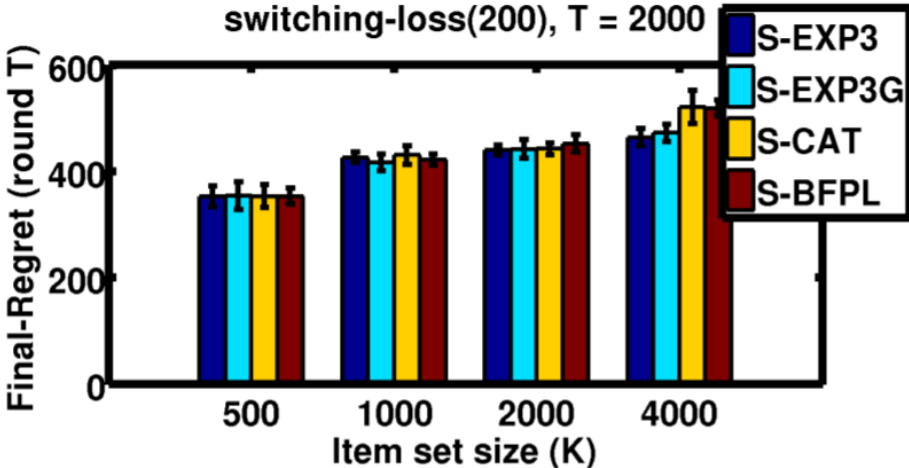
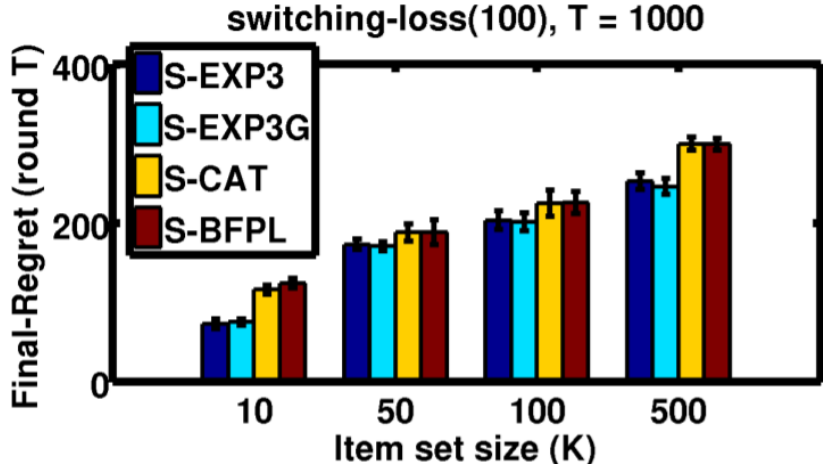
# Empirical Evaluations

# Independent Availabilities

Regret vs Time:

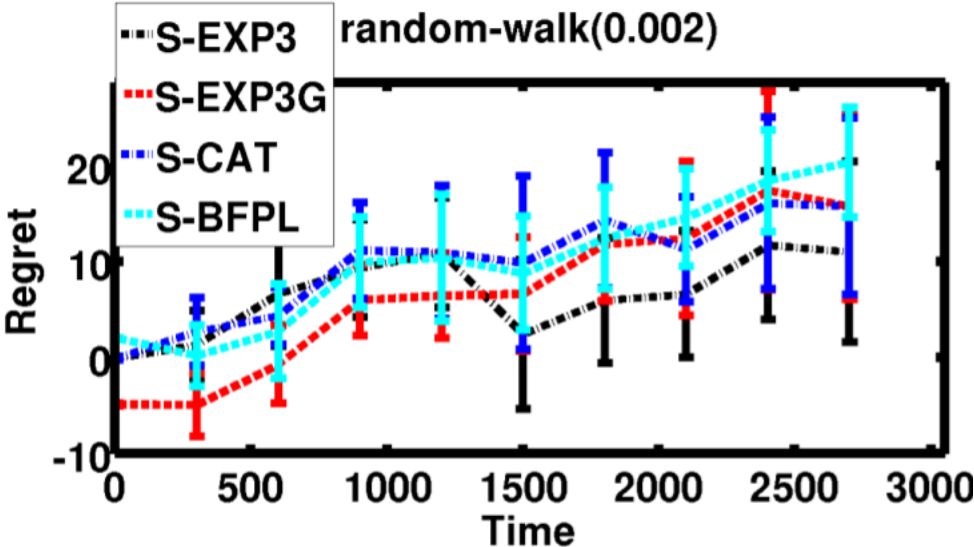
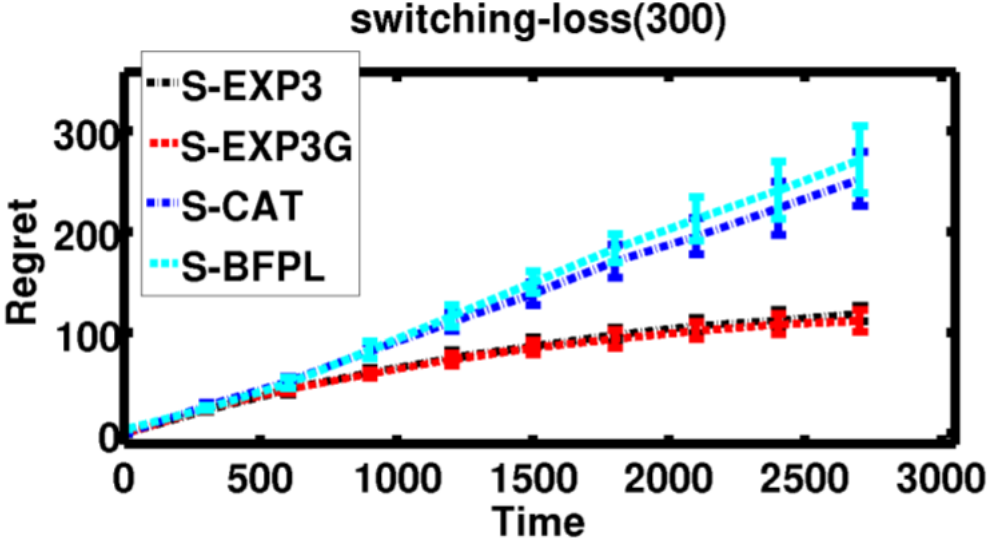
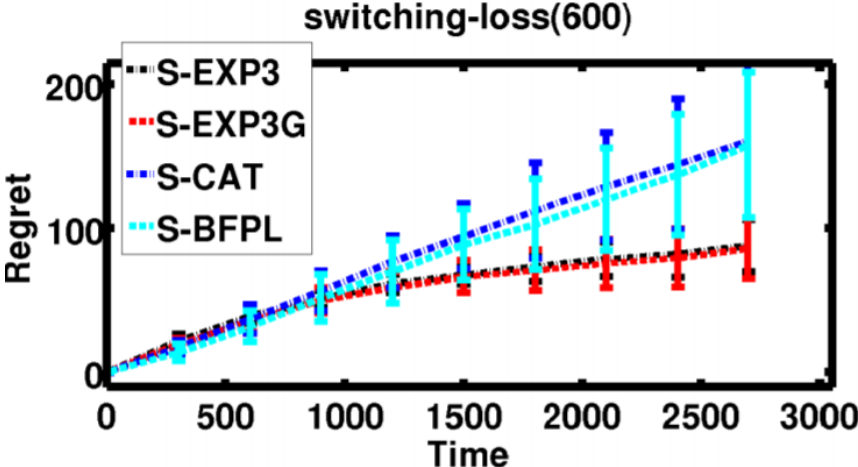


Final Regret vs K:



# General Availabilities

Regret vs Time:



---

# Results Summary:

Independent Availabilities:  $\tilde{O}(K^2\sqrt{T})$

We achieved optimal dependence in T

Existing result:  $O(T^{2/3})$

General Availabilities:  $\tilde{O}(\sqrt{2^K T})$

Computationally Efficient

Suboptimal in K

---

## Several Future Directions:

- i. Exact lower bound? Is  $\Omega(\sqrt{KT})$  really tight or it is  $\Omega(K\sqrt{T})$ ?
- ii. Improved algorithms with optimal dependency in  $K$ .
- iii. Extending similar ideas to related setups: Rotting or Dying bandits
- iv. Regret vs Effective-dimension: Extension to large arm-space (potentially infinite)?



---

# Thanks

## Acknowledgement

- French National Research Agency project BOLD (ANR19-CE23-0026-04)
- European CHIST-ERA project DELTA
- Antoine Chambaz, Marie-Helene Gbaguidi and Inria, Paris