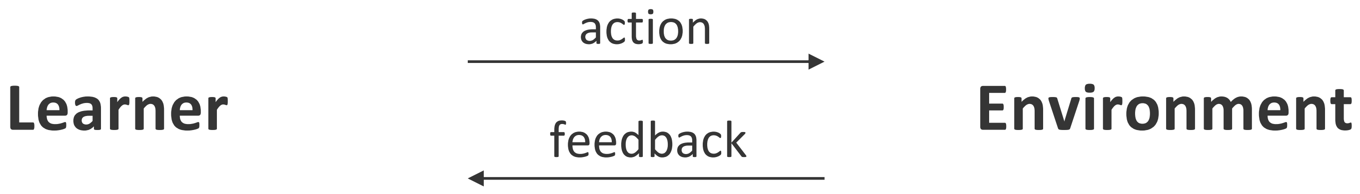
Optimal Dynamic Regret for Bandits without Prior Knowledge

Chen-Yu Wei
Research Fellow @ Simons Institute

Online Learning



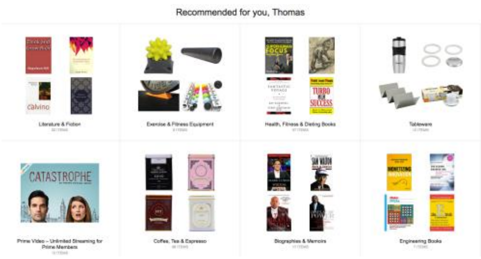
Online Learning with Bandit Feedback

Learner — action → **Environment**

← feedback

=

reward of the chosen action



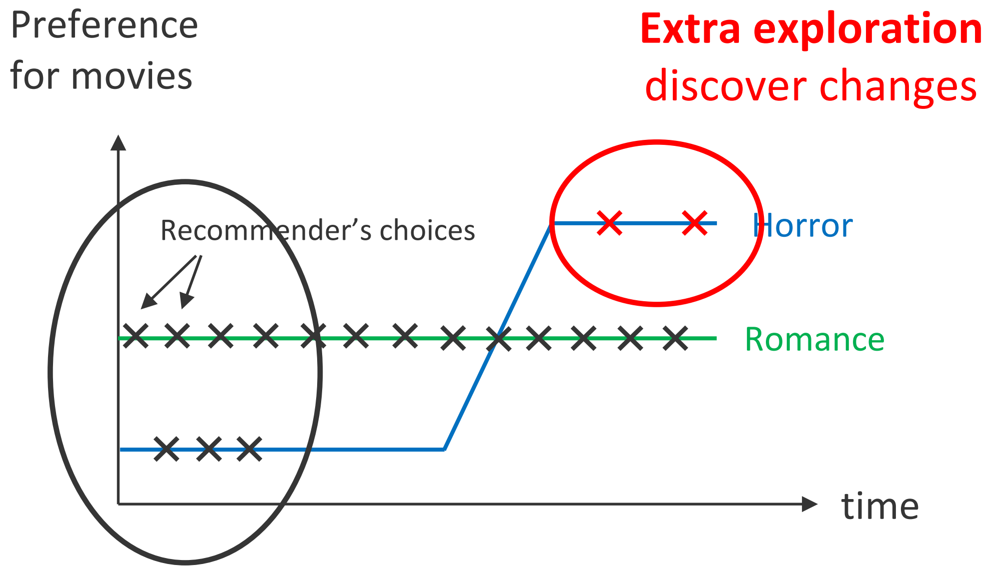
Recommender

-1 (Don't like it)

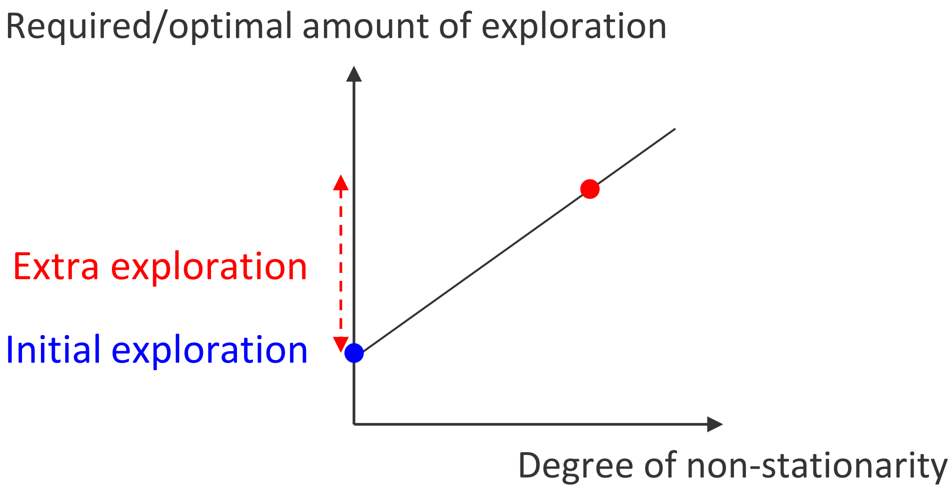
 ?

User

Bandit Feedback + Non-Stationarity



Initial exploration
resolve uncertainty
(e.g., optimism in the face of uncertainty)



Challenge:

How to use the right amount of exploration without prior knowledge on the degree of non-stationarity?

(avoid **over-exploration** or **under-exploration**)

Multi-Armed Bandits with Non-Stationarity

Given: K arms

For $t = 1, \ldots, T$:

Environment chooses a *mean reward vector* $\mu_t \in [0,1]^K$

Learner chooses an arm $a_t \in [K]$

Learner observes the reward r_t with $\mathbb{E}[r_t] = \mu_t(a_t)$

$$\text{Dynamic-Regret} = \sum_{t=1}^{T} \left(\max_{a \in [K]} \mu_t(a) - r_t \right)$$

$$S = 1 + \sum_{t=2}^{T} \mathbf{1}\{\mu_t \neq \mu_{t-1}\}$$

$$V = 1 + \sum_{t=2}^{T} \|\mu_t - \mu_{t-1}\|_\infty$$

$$\text{Dynamic-regret lower bound} = \Omega\left(\min\left\{\sqrt{ST}, V^{\frac{1}{3}} T^{\frac{2}{3}}\right\}\right)$$

Related works with $\tilde{O}(S^{\alpha}T^{1-\alpha})$ or $\tilde{O}(V^{\alpha}T^{1-\alpha})$ upper bounds

	Multi-armed bandits	Multi-armed contextual bandits	Linear bandits	Generalized linear bandits	MDP	Realizable contextual bandits
Auer et al., 2002	$\sqrt{ST}$ **(known S)**					
Besbes et al., 2014	$V^{1/3}T^{2/3}$ **(known V)**					
Karnin and Anava, 2016	$V^{0.18}T^{0.82}$					
Luo et al., 2018	$\min\{S^{1/4}T^{3/4}, V^{1/5}T^{4/5}\}$					
Cheung et al., 2018/2019	$V^{1/3}T^{2/3} + T^{3/4}$		$V^{1/4}T^{3/4}$			
Auer et al., 2018/2019	$\sqrt{ST}$					
Chen et al., 2019	$\min\{\sqrt{ST}, V^{1/3}T^{2/3}\}$					
W and Luo, 2021	$\min\{\sqrt{ST}, V^{1/3}T^{2/3}\}$					

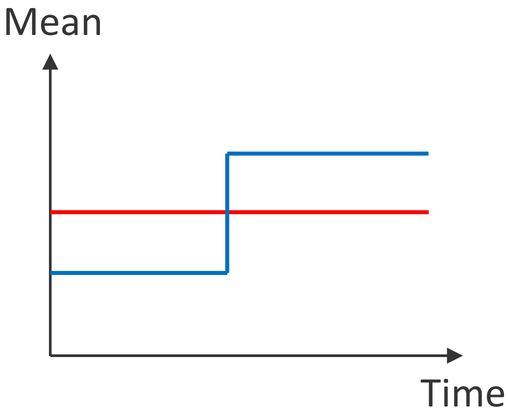
Papers We Will Discuss Today

Auer, Gajane, Ortner: multi-scale change detection
[EWRL 2018]: 2-armed bandits
[COLT 2019]: K-armed bandits

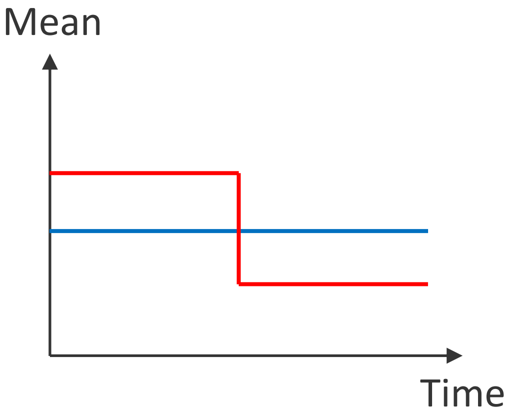
W and Luo [COLT 2021]: generalizing (Auer et al.) to a wide range of problems

Wang [arXiv, 2022]: further generalization

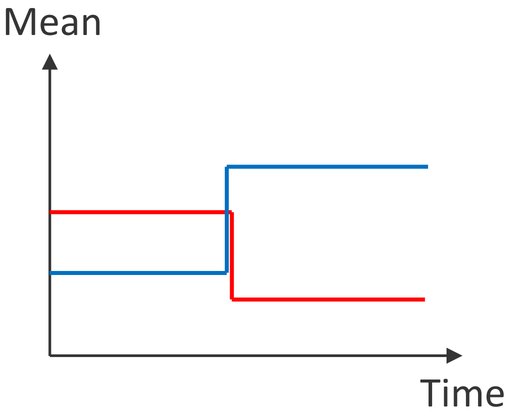
Which one is the most difficult to detect?



Bad arm → Good
(need extra exploration)



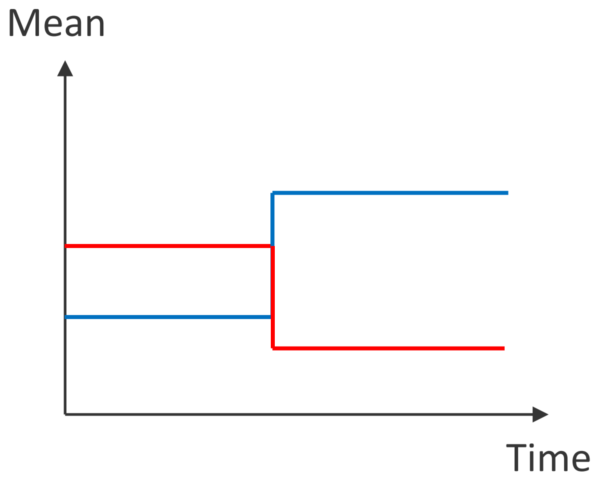
Good arm → Bad



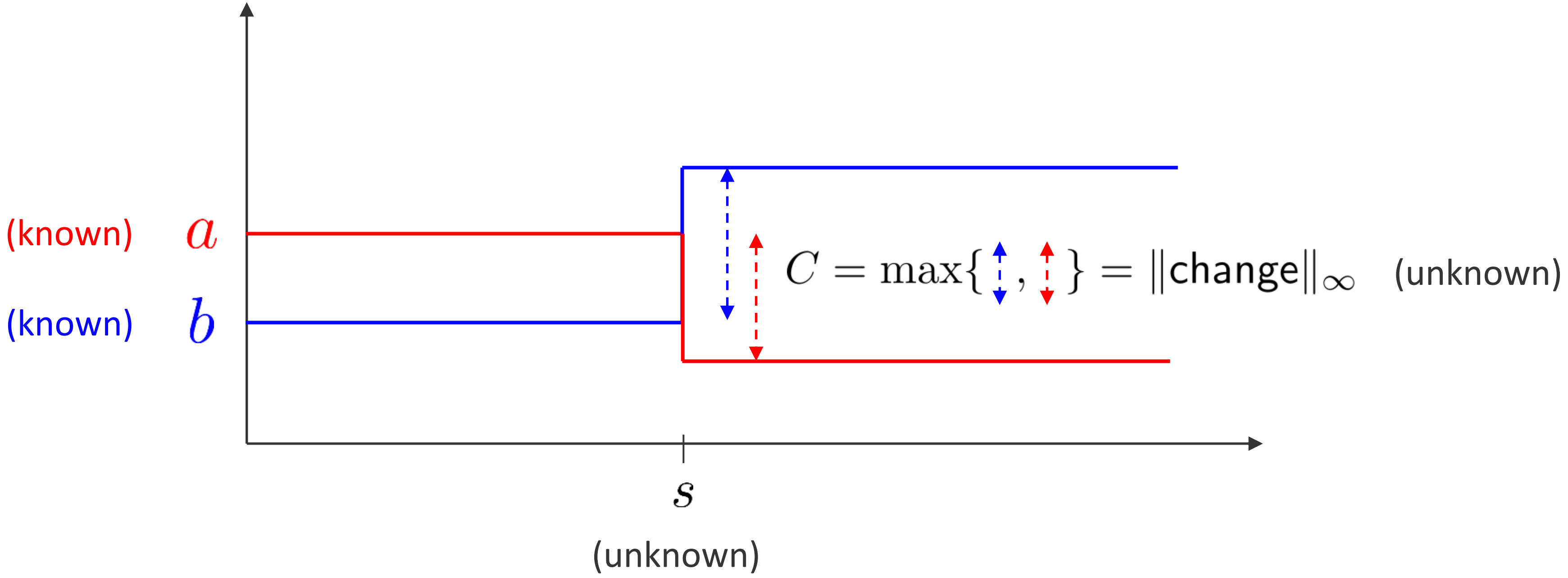
Both

Simplification

- Two arms
- Means change at most once
- The initial means are known



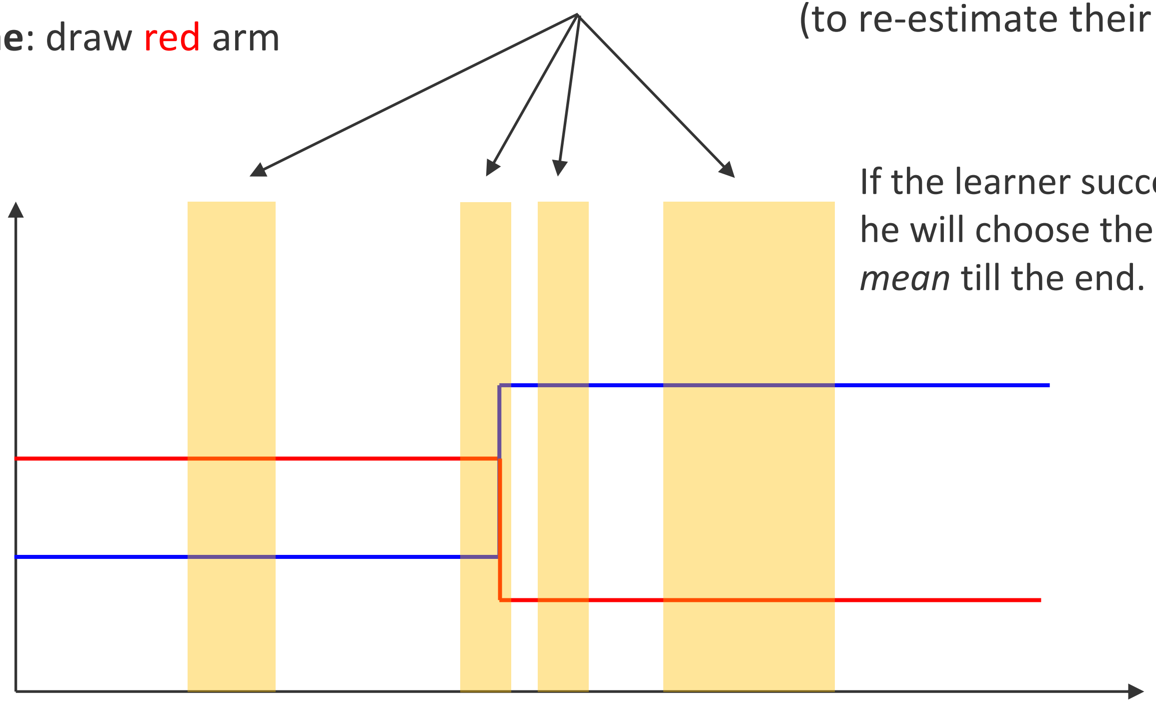
Simplification



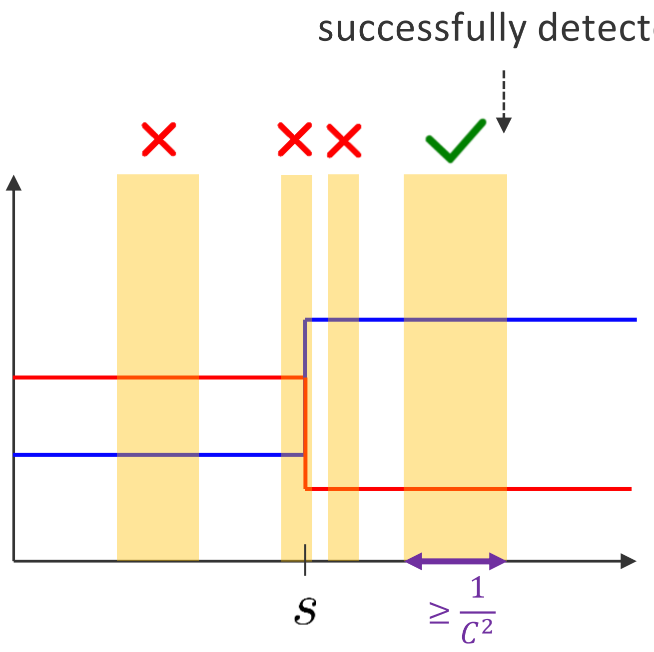
Algorithm Template

Detection blocks (DB): uniformly randomly draw two arms
(to re-estimate their means)

Other time: draw red arm



If the learner successfully detects the change, he will choose the arm with better *re-estimated mean* till the end.



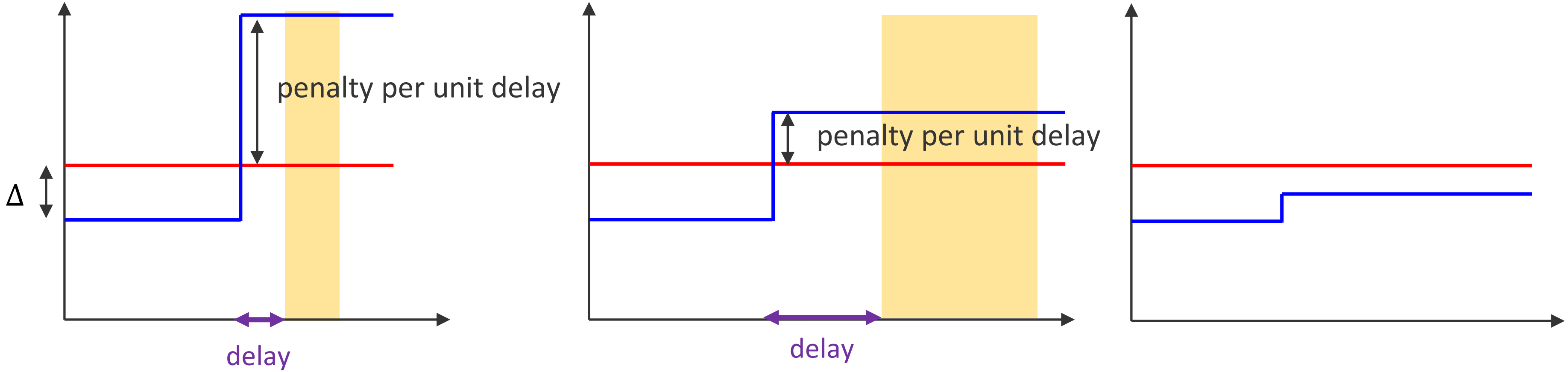
What makes a successful detection?

- DB starts after s
- DB length $\geq 1/C^2$

(To estimate the mean up to an accuracy of C, we need/only need $\approx \frac{1}{C^2}$ samples)

Regret =	**Detection overhead** +	**Non-detection penalty**
	(random draws in DB)	(detection delay)

Key: smartly schedule DBs to balance the two terms large $C > \Delta$ | small $C > \Delta$ | $C < \Delta$

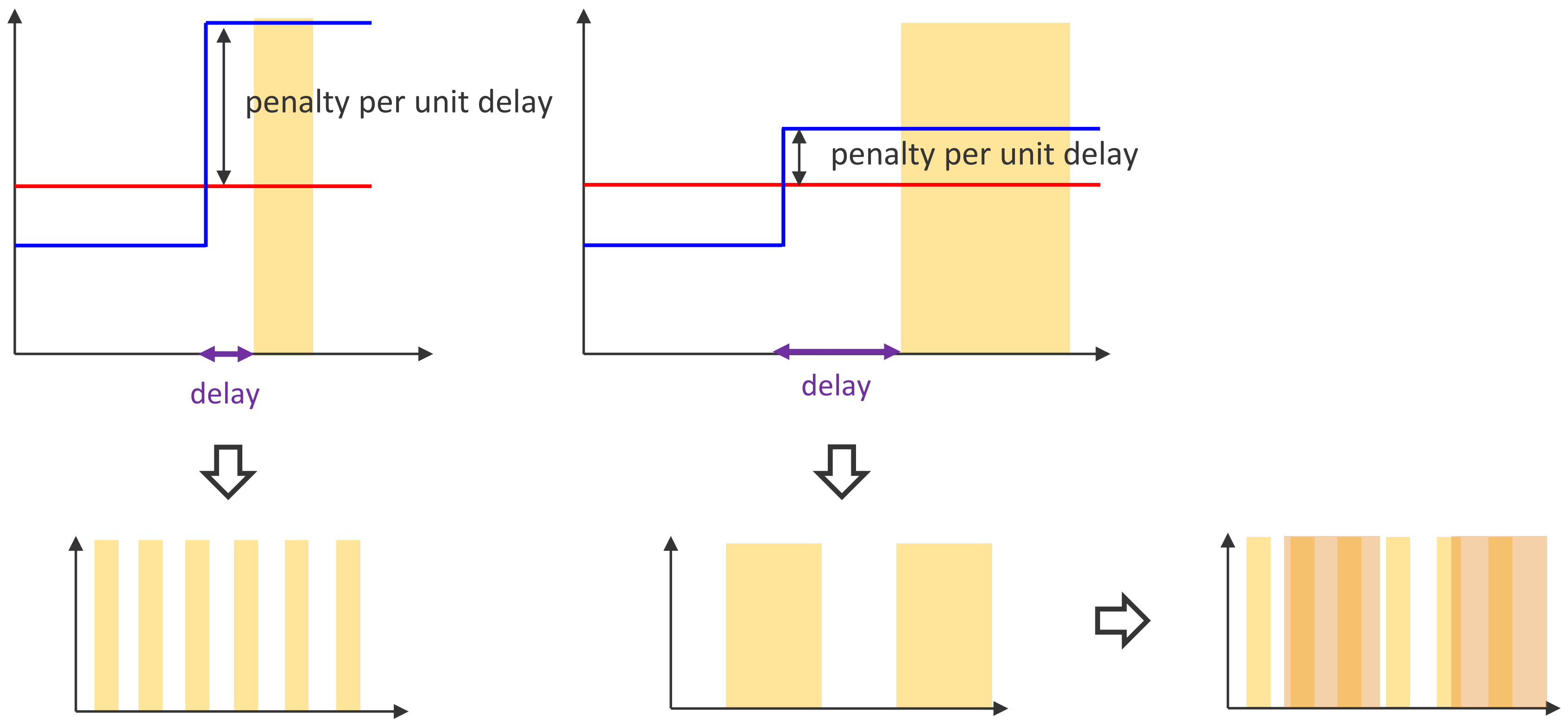


Shorter DB is enough ☺
Higher penalty per unit delay ☹

Need longer DB ☹
Lower penalty per unit delay ☺

No need to detect ☺

Delay := time from the change point to the beginning of a successful detection


penalty per unit delay
delay
penalty per unit delay
delay

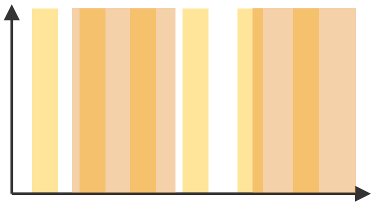
Algorithm (just one change point)

Draw two arms uniformly at random, until $t \gtrsim \frac{1}{|a-b|^2}$.
(Also, perform some non-stationarity detection)

For $t = 1, 2, \ldots$:

For $\epsilon = 1, \frac{1}{2}, \frac{1}{4}, \ldots, \frac{\Delta}{2}$:

w.p. $p_\epsilon = \frac{\epsilon}{\sqrt{t}}$, initiate a DB of length $\approx \frac{1}{\epsilon^2}$. (allow overlap)



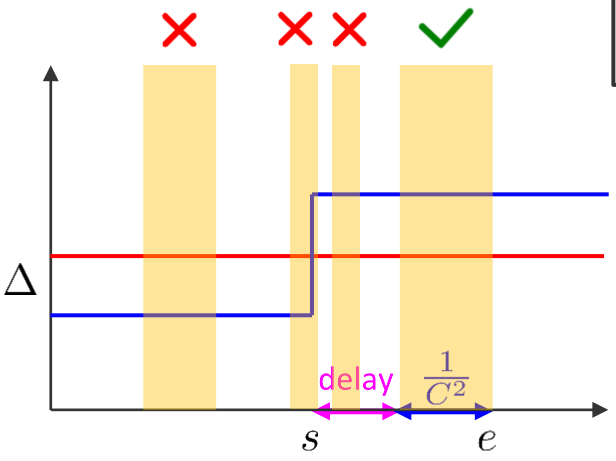
Uniformly randomly choose arms if t lies in any DB; otherwise choose argmax$\{a, b\}$

Detection:

At the end of every DB with length $\frac{1}{\epsilon^2}$, check if

$$|a - a'| > \epsilon \text{ or } |b - b'| > \epsilon?$$

where a', b' are mean estimations in DB.
If so, choose argmax$\{a', b'\}$ in the remaining rounds.



Proof sketch: Regret $\leq O(\sqrt{T})$

change amount $= C$ (assume $C \geq \Delta$)

dynamic-regret $\leq$ **detection overhead** $+$ **non-detection penalty**

$\Delta \times$ (total DB length in $[0, s]$)

$C \times (e - s)$

$$\Delta \times \sum_{t=0}^{s} \sum_{\epsilon \in \{1, \frac{1}{2}, \ldots, \Delta\}} p_\epsilon \times \frac{1}{\epsilon^2}$$

$C \times \left(\text{delay} + \frac{1}{C^2}\right)$

$C \times \frac{1}{p_C} + \frac{1}{C}$

$p_\epsilon = \frac{\epsilon}{\sqrt{t}}$ ⇨

$\leq \sqrt{s}$

$\leq \sqrt{e}$ $= \sqrt{\text{DB length}}$

Algorithm (Multiple change points)

Draw two arms uniformly at random, until $t \gtrsim \frac{1}{|a-b|^2}$.
(Also, perform some non-stationarity detection)

For $t = 1, 2, \ldots$:

For $\epsilon = 1, \frac{1}{2}, \frac{1}{4}, \ldots, \frac{\Delta}{2}$:

w.p. $p_\epsilon = \frac{\epsilon}{\sqrt{t}}$, initiate a DB of length $\approx \frac{1}{\epsilon^2}$.

Uniformly randomly choose arms if t lies in any DB; otherwise choose argmax$\{a, b\}$

Detection:

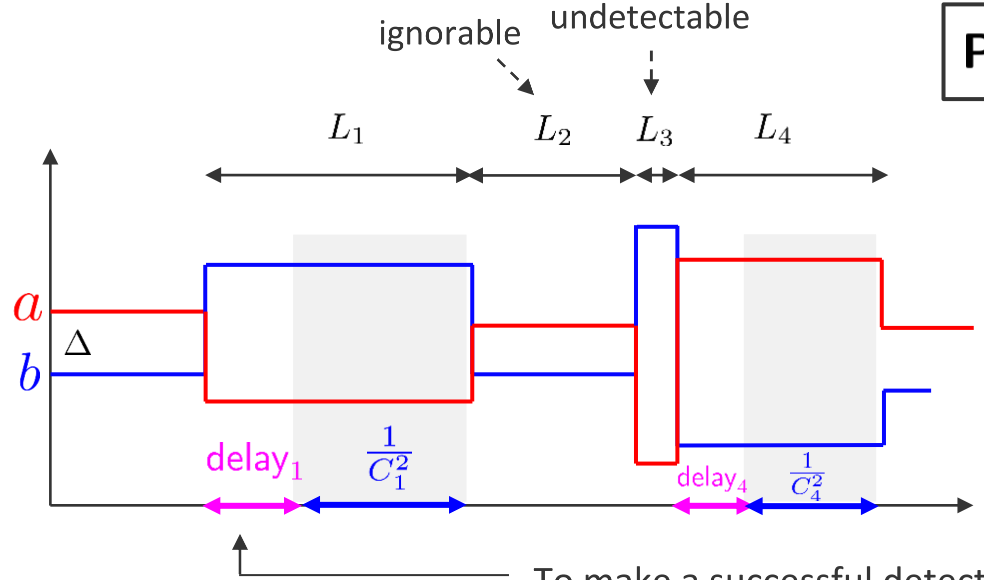
At the end of every DB with length $\frac{1}{\epsilon^2}$, check if

$$|a - a'| > \epsilon \text{ or } |b - b'| > \epsilon?$$

where a', b' are new estimations in DB.
If so, restart the algorithm.

Proof sketch: Regret $\leq O\left(\sum_i \sqrt{L_i}\right)$

Only need to show this before restart



To make a successful detection, a DB of length $1/C_1^2$ needs to start here.

C_i: change in interval i compared to the initial reward (i.e., at time 0)

$C_i < \frac{\Delta}{2}$	ignorable
$C_i > \frac{\Delta}{2}, L_i < \frac{1}{C_i^2}$	undetectable
$C_i > \frac{\Delta}{2}, L_i > \frac{1}{C_i^2}$	detectable

$e :=$ the time we terminate the algorithm and restart

detection overhead $\leq \sqrt{e}$

non-detection penalty in ignorable intervals $= 0$

non-detection penalty in undetectable intervals $= C_i L_i \leq \sqrt{L_i}$

non-detection penalty in detectable intervals $= C_i L_i \leq C_i \left(\text{delay}_i + \frac{1}{C_i^2}\right) \leq C_i \text{delay}_i + \sqrt{L_i}$

regret $\lesssim \sqrt{e} + \sum_i \sqrt{L_i} + \sum_{i:\text{detectable}} C_i \text{delay}_i \lesssim \sum_i \sqrt{L_i}$

$\leq \sqrt{e}$

General Decision Making with Non-Stationarity

Given: policy set Π

For $t = 1, \ldots, T$:

- Environment chooses a mapping $f_t\colon \Pi \to [0, 1]$
- Learner chooses a policy $\pi_t \in \Pi$
- Learner observes the reward r_t with $\mathbb{E}[r_t] = f_t(\pi_t)$

$$\text{Dynamic-Regret} = \sum_{t=1}^{T} \left(\max_{\pi \in \Pi} f_t(\pi) - r_t \right)$$

Extensions to Other Settings

K-armed bandit
(Auer, Gajane, Ortner, 2019)

Contextual K-armed bandit
(Chen, Lee, Luo, **W**, 2019)

Combinatorial semi-bandit
(Chen, Wang, Zhao, Zheng, 2021)

Algorithm 1 AdSwitch

Input: Time horizon T.
Initialization $\ell \leftarrow 0, t \leftarrow 0$.
Start a new episode:
$\ell \leftarrow \ell + 1$.
Set start of the episode $t_\ell \leftarrow t + 1$.
$\text{GOOD}_{t+1} = \{1, \ldots, K\}$, $\text{BAD}_{t+1} = \{\}$.
Next time step:
$t \leftarrow t + 1$.
Add checks for bad arms:
For all $a \in \text{BAD}_t$, and all $i \geq 1$ with $2^{-i} \geq \tilde{\Delta}_\ell(a)/16$,
with probability $2^{-i}\sqrt{\ell/(KT \log T)}$ add $\mathcal{S}_t(a) \leftarrow \mathcal{S}_t(a) \cup (2^{-i}, \lceil 2^{2i+1} \log T \rceil, t)$.
Select an arm:
Select $a_t = \arg\min_a \{\tau : a \notin \{a_\tau, \ldots, a_{t-1}\}, a \in \text{GOOD}_t \vee \mathcal{S}_t(a) \neq \{\}\}$.
Receive reward r_t.
Check for changes of good arms:
If there is $a \in \text{GOOD}_t$ and $t_\ell \leq s_1 \leq s_2 \leq t$ and $t_\ell \leq s \leq t$ such that condition (3)
holds, then start a new episode.
Check for changes of bad arms:
If there is $a \in \text{BAD}_t$ and $t_\ell \leq s \leq t$ such that condition (4) holds,
then start a new episode.
For $a \in \text{BAD}_t$, $\mathcal{S}_{t+1}(a) \leftarrow \{(\epsilon, n, s) \in \mathcal{S}_t(a) : n_{[s,t]} < n\}$.
Evict arms from GOOD_t***:***
$\text{BAD}_{t+1} = \text{BAD}_t \cup \{a \in \text{GOOD}_t | \exists s \geq t_\ell \text{ for which (1) holds}\}$.
For evicted arms $a \in \text{BAD}_{t+1} \setminus \text{BAD}_t$, calculate $\tilde{\mu}_\ell(a)$ and $\tilde{\Delta}_\ell(a)$ according to (2), and
set $\mathcal{S}_{t+1}(a) \leftarrow \{\}$.
$\text{GOOD}_{t+1} = \{1, \ldots, K\} \setminus \text{BAD}_{t+1}$.
Continue with the next time step.

based on ILOVETOCONBANDITS
(Agarwal et al., 2014)

Maintain a **distribution over policies**, and control the variance of the reward estimator for all policies.

N/A to MDPs, linear contextual bandits, generalized linear bandits, convex bandits, etc.

Rethink about the solution

Do we really need to track every policy’s changes?

We only need to track

- whether the best policy’s reward becomes high
- whether the learner’s reward becomes low

No-Regret Algorithm

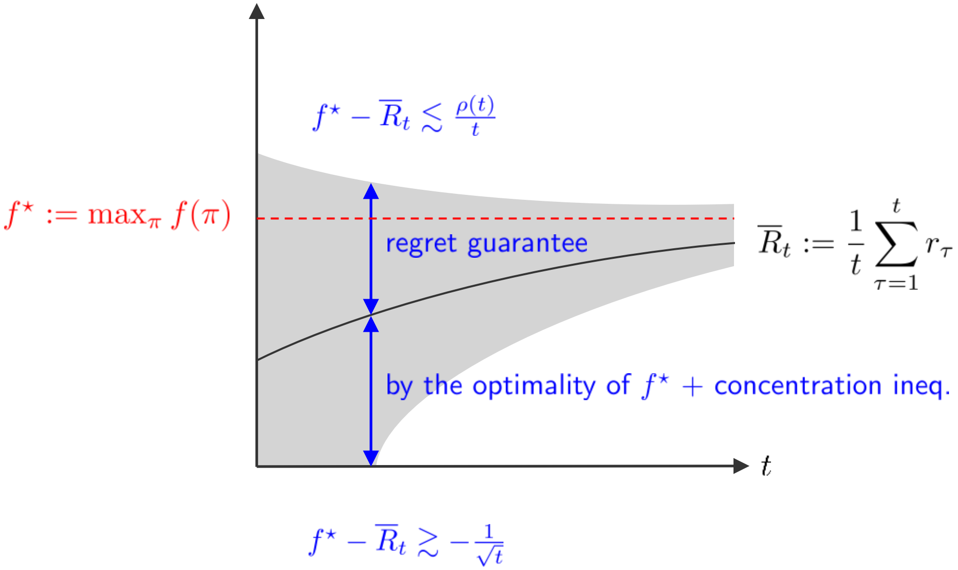
No-Regret Algorithm for the Stationary Environment

In the *stationary* environment ($f_t = f$), the algorithm ensures

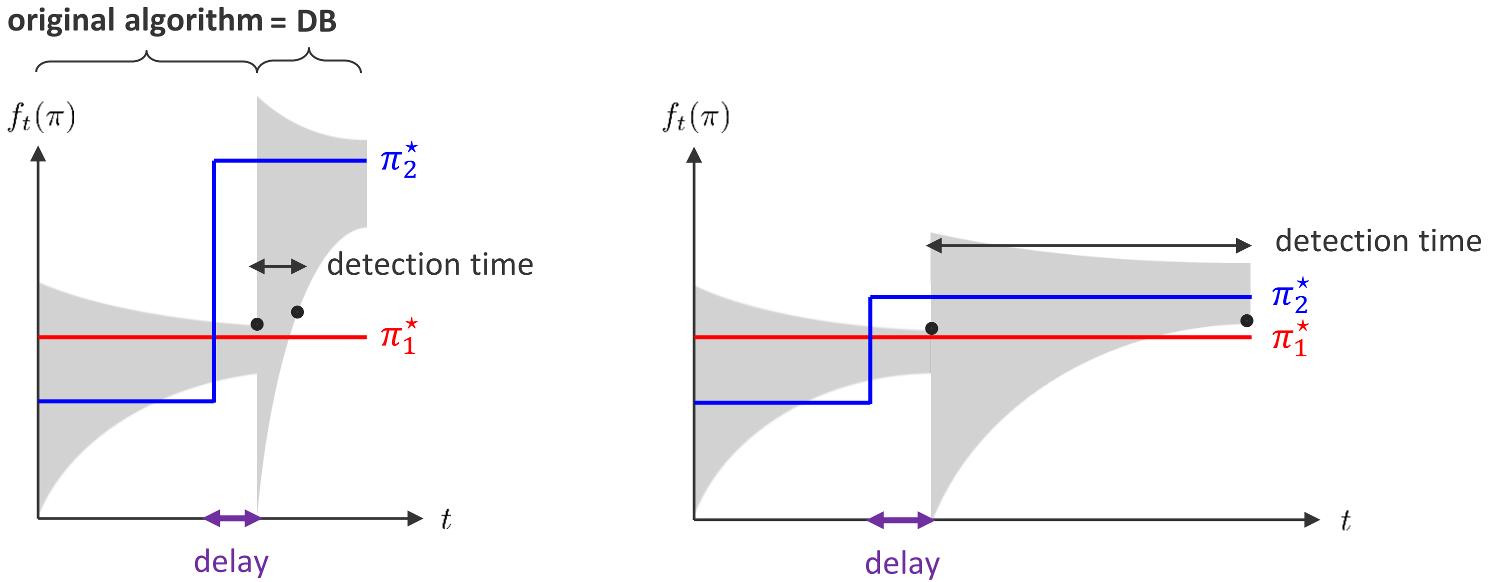
$$\max_{\pi} \sum_{\tau=1}^{t} \left(f(\pi) - r_\tau\right) \lesssim \rho(t) \quad \text{for some } \rho(t) \text{ sublinear in } t$$

No-Regret Algorithm: Track the Optimal Policy

In a stationary environment:



No-Regret Algorithm as a Detection Block



Algorithm [Auer, Gajane, Ortner, 2018]

Draw two arms uniformly at random, until $t \gtrsim \frac{1}{|a-b|^2}$.
(Also, perform some non-stationarity detection)

Initial exploration

For $t = 0, 1, 2 \ldots$:

For $\epsilon = 1, \frac{1}{2}, \frac{1}{4}, \ldots, \frac{\Delta}{2}$:

w.p. $p_\epsilon = \frac{\epsilon}{\sqrt{t}}$, initiate a DB of length $\approx \frac{1}{\epsilon^2}$.

randomly choose arms if t lies in DB; otherwise choose argmax$\{a, b\}$

Extra exploration

Detection:

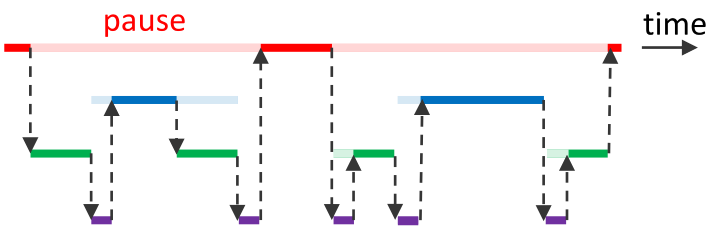
At the end of every DB with length $\frac{1}{\epsilon^2}$, check if

$$|a - a'| > \epsilon \text{ or } |b - b'| > \epsilon?$$

where a', b' are new estimations in DB.
If so, restart the algorithm.

Check if any arm changes

Algorithm (combining [**W** and Luo, 2021] and [Wang, 2022])



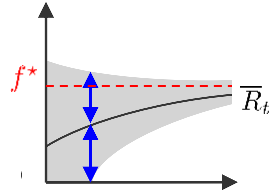
For $t = 0, 1, 2, \ldots$:

For $\epsilon = 1, \frac{1}{2}, \frac{1}{4}, \ldots, \frac{1}{\sqrt{t}}$:

If $\frac{1}{\epsilon^2}$ divides t, w.p. $p_\epsilon = \frac{1}{\epsilon\sqrt{t}}$, initiate a Base Algorithm of length $\approx \frac{1}{\epsilon^2}$

Execute the Base Algorithm with the smallest length among overlapping ones.

Detection:



For the Base Algorithm $\mathcal{A}$ executed at round t,

$$U_t \leftarrow \min\left(U_{t-1}, \overline{R}_t^{\mathcal{A}} + \text{confidence}_t^{\mathcal{A}}\right) \quad L_t \leftarrow \max\left(L_{t-1}, \overline{R}_t^{\mathcal{A}} - \text{confidence}_t^{\mathcal{A}}\right)$$

If $U_t < L_t$, restart. (detect whether $f_t^\star$ changes)

If $\sum_{\tau=1}^{t}(U_\tau - r_\tau) > \Omega(\rho(t))$, restart. (detect whether learner's performance drops)

Remarks on [W and Luo, 2021] and [Wang, 2022]

Actual assumption in [W and Luo, 2021]: UCB condition

In the *stationary* environment ($f_t = f$), the algorithm can output $\tilde{f}_t$ at time t and ensure

$$\tilde{f}_t \geq \max_{\pi} f(\pi)$$

$$\sum_{\tau=1}^{t} (\tilde{f}_t - r_\tau) \lesssim \rho(t) \quad \text{for some } \rho(t) \text{ sublinear in } t$$

[Wang, 2022]: no-regret condition **implies** UCB condition

$$\tilde{f}_t = \sum_{\tau=1}^{t} r_\tau + \frac{\rho(t)}{t}$$

In the *stationary* environment ($f_t = f$), the algorithm ensures

$$\max_{\pi} \sum_{\tau=1}^{t} (f(\pi) - r_\tau) \lesssim \rho(t) \quad \text{for some } \rho(t) \text{ sublinear in } t$$

Assumptions for handling gradual changes

In the *near-stationary* environment where

$$V_{[1,t]} \triangleq 1 + \sum_{\tau=2}^{t} \max_{\pi} |f_\tau(\pi) - f_{\tau-1}(\pi)| \lesssim \frac{\rho(t)}{t}$$

the algorithm ensures

$$\max_{\pi} \sum_{\tau=1}^{t} \big(f(\pi) - r_\tau\big) \lesssim \rho(t) + tV_{[1,t]}$$

Summary

(Near-)stationary algorithm ⇨ **Meta Algorithm** ⇨ Non-stationary algorithm

Auer, Gajane, Ortner, 2018

Multi-scale detection

W and Luo, 2021

Black-box usage of algorithms with UCB condition

Wang, 2022

No-regret cond. → UCB cond.

Recent Development and Open Problems

S_C: how many times the regret benchmark changes

$\sqrt{S_c T}$? $\longrightarrow$ $\sqrt{S_{\text{sig}} T}$ [Suk and Kpotufe, 2022]

$\sqrt{S_{\text{sig}} T} \longrightarrow \sqrt{S_{\text{best}} T} \longrightarrow \sqrt{ST}$ [Abbasi-Yadkori et al., 2022]

$\sqrt{S_{\text{sig}} T} \longrightarrow V^{\frac{1}{3}} T^{\frac{2}{3}}$

[Auer et al., 2002]	$O\left(\sqrt{S_c T}\right)$	known S_c
[Cheung et al., 2018]	$O\left(\sqrt{S_c T} + T^{\frac{3}{4}}\right)$	unknown S_c, oblivious adversary
[Marinov & Zimmert, 2021]	$\omega\left(S_c^{\alpha} T^{1-\alpha}\right)$, any α	unknown S_c, adaptive adversary
	$O\left(\sqrt{S_c T}\right)$	unknown S_c, oblivious adversary?

Thank you!