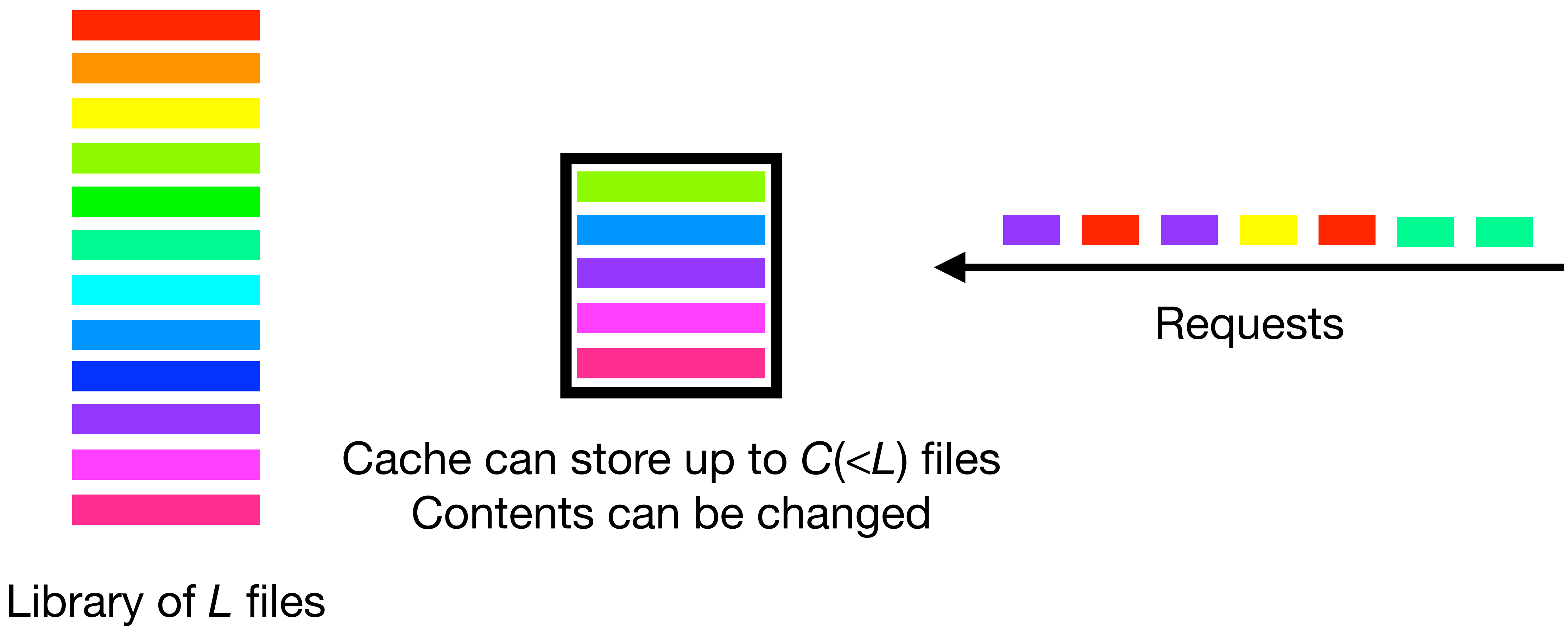
Regret-Optimal Online Caching for Adversarial and Stochastic Arrivals

Sharayu Moharir
IIT Bombay

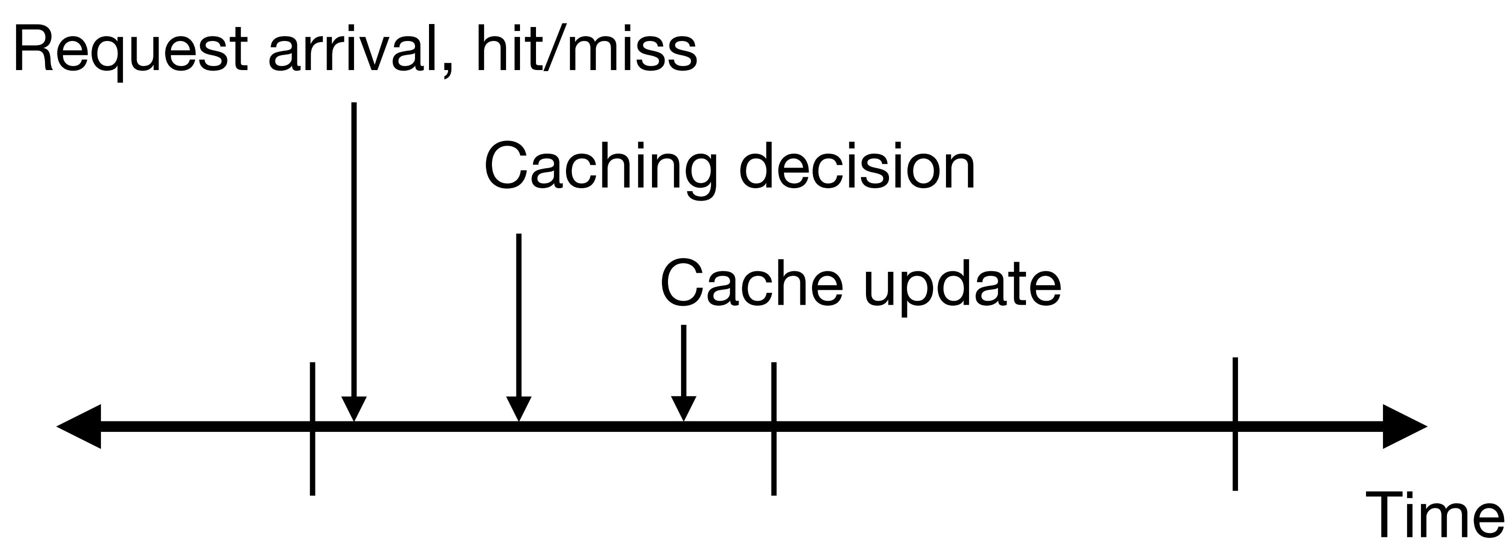
Joint work with Fathima Faizal, Priya Singh, and Nikhil Karamchandani

Caching



- *Hit*: requested file present in cache
- *Miss*: requested file not present in cache
- Algorithmic challenge: determine which files to cache over time
- Goal: maximize the number of hits/minimize the number of misses

Request Models



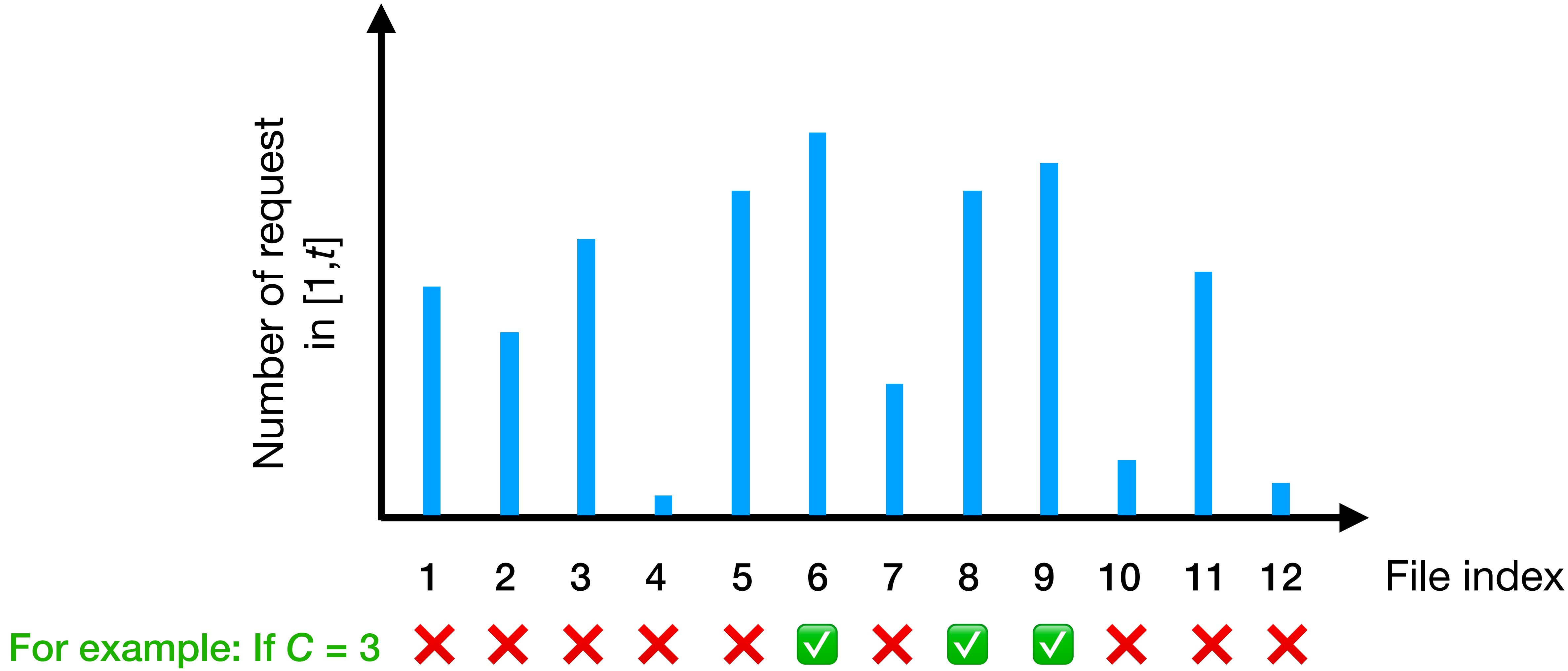
i.i.d. Stochastic Requests	Adversarial Requests
• Requests are i.i.d. random variables	• No assumptions on arrival sequence
• Online: Distribution unknown, caching decisions based on past arrivals	• Online: Caching decisions based on past arrivals

Performance Metric: Regret

i.i.d. Stochastic Requests	Adversarial Requests
• OPT: caches the C most popular files • Popularity of File i = $\mathbb{P}$(Request for file i) • Static policy, knows popularity of files	• OPT: static cache configuration which maximizes number of hits in $[1,T]$ • Offline: knows arrival sequence apriori
• D: distribution of request arrivals • Candidate policy $\mathcal{P}$ • $M_{\mathcal{P}}(T)$: number of misses in $[1,T]$ under $\mathcal{P}$ Regret: $R_{\mathcal{P}}(T) = \mathbb{E}_{D,\mathcal{P}}[M_{\mathcal{P}}(\mathrm{T})] - \mathbb{E}_{D}[M_{\mathrm{OPT}}(\mathrm{T})]$ • Guarantee on expected performance	• A: arrival sequence, candidate policy $\mathcal{P}$ • $M_{\mathcal{P}}(A,T)$: number of misses in $[1, T]$ for A under policy $\mathcal{P}$ Regret: $R_{\mathcal{P}}(T) = \max_{A}(\mathbb{E}_{\mathcal{P}}[M_{\mathcal{P}}(A,T)] - M_{\mathrm{OPT}}(A,T))$ • Worst-case performance guarantee

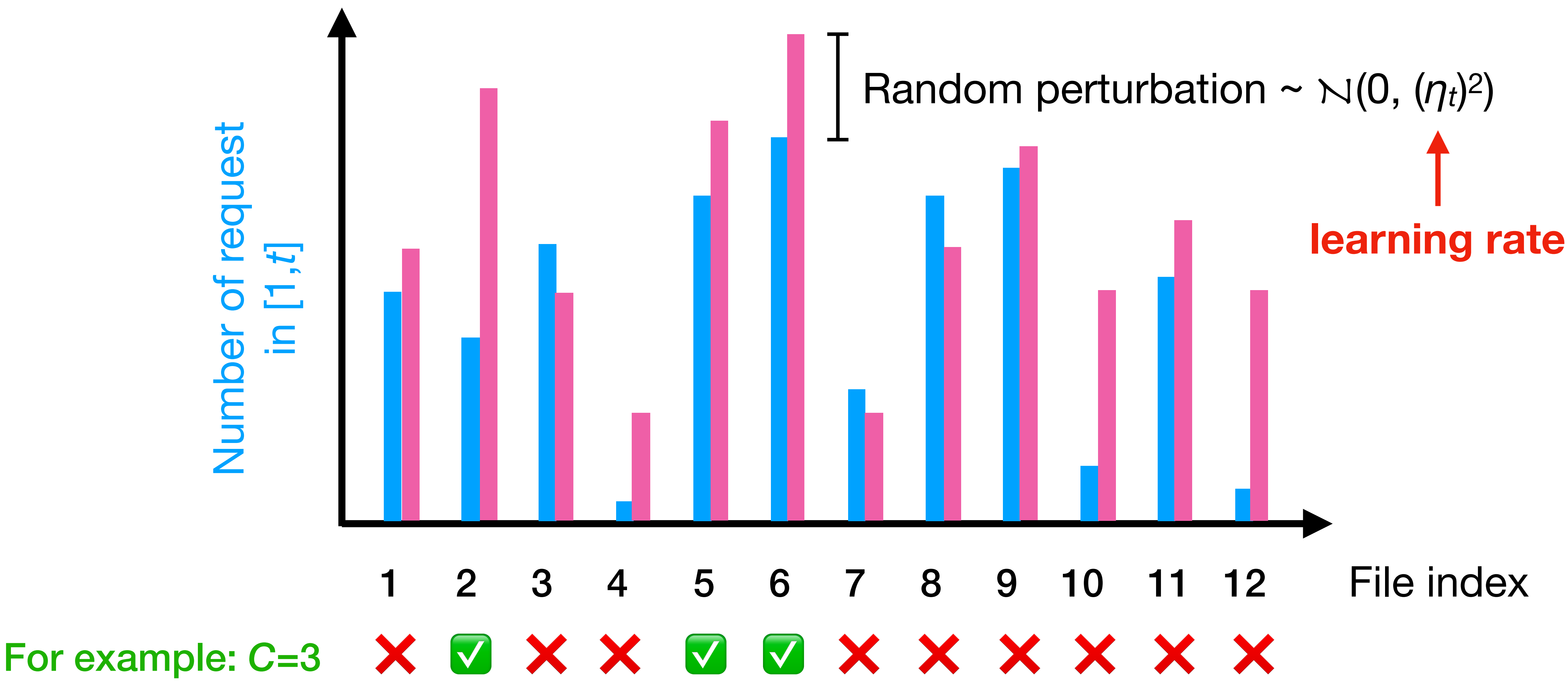
Is there a policy with order-optimal (w.r.t. time) regret
for both stochastic & adversarial arrivals?

Policy 1: Least Frequently Used



- Keep track of cumulative number of requests for each file
- Score(t) = cumulative number of requests in [1, t]
- Cache the C files with the C highest scores

Policy 2: Follow the Perturbed Leader



- Keep track of cumulative number of requests for each file
- Score(t) = cumulative number of requests in [1, t] + random perturbation
- Cache the C files with the C highest scores

Overview of Known Results

Policies	i.i.d. Stochastic Requests	Adversarial Requests
LFU	O(1) regret (order-optimal)[1]	$\Omega(T)$ regret, strictly sub-optimal[2]
FTPL	🤔	$O(\sqrt{T})$ regret for $\eta_t \propto \sqrt{T}$ (order-optimal)[2] $O(\sqrt{T})$ regret for $\eta_t \propto \sqrt{t}$ (order-optimal)[3]

1 A. Bura et al., Learning to Cache and Caching to Learn: Regret Analysis of Caching Algorithms, *IEEE/ACM ToN*
2 R. Bhattacharjee et al., Fundamental Limits of Online Network-Caching, *ACM SIGMETRICS* 2020
3 S. Mukhopadhyay et al., Online Caching with Optimal Switching Regret, *ISIT* 2021

FTPL with Constant Learning Rate

Recall:

- Random perturbation in time-slot $t \sim \mathcal{N}(0, (\eta_t)^2)$
- For i.i.d. stochastic arrivals, $R_{\text{LFU}}(T) = O(1)$

Theorem: For i.i.d. stochastic arrivals and $\eta_t \propto \sqrt{T}$:

$$R_{\text{FTPL}}(T) = \Omega(\sqrt{T}).$$

FTPL with $\eta_t \propto \sqrt{T}$ is strictly sub-optimal for i.i.d. stochastic arrivals

FTPL with Time-Varying Learning Rate

Recall:

- Random perturbation in time-slot $t \sim \mathcal{N}(0, (\eta_t)^2)$
- L = library size, C = cache size

Let:

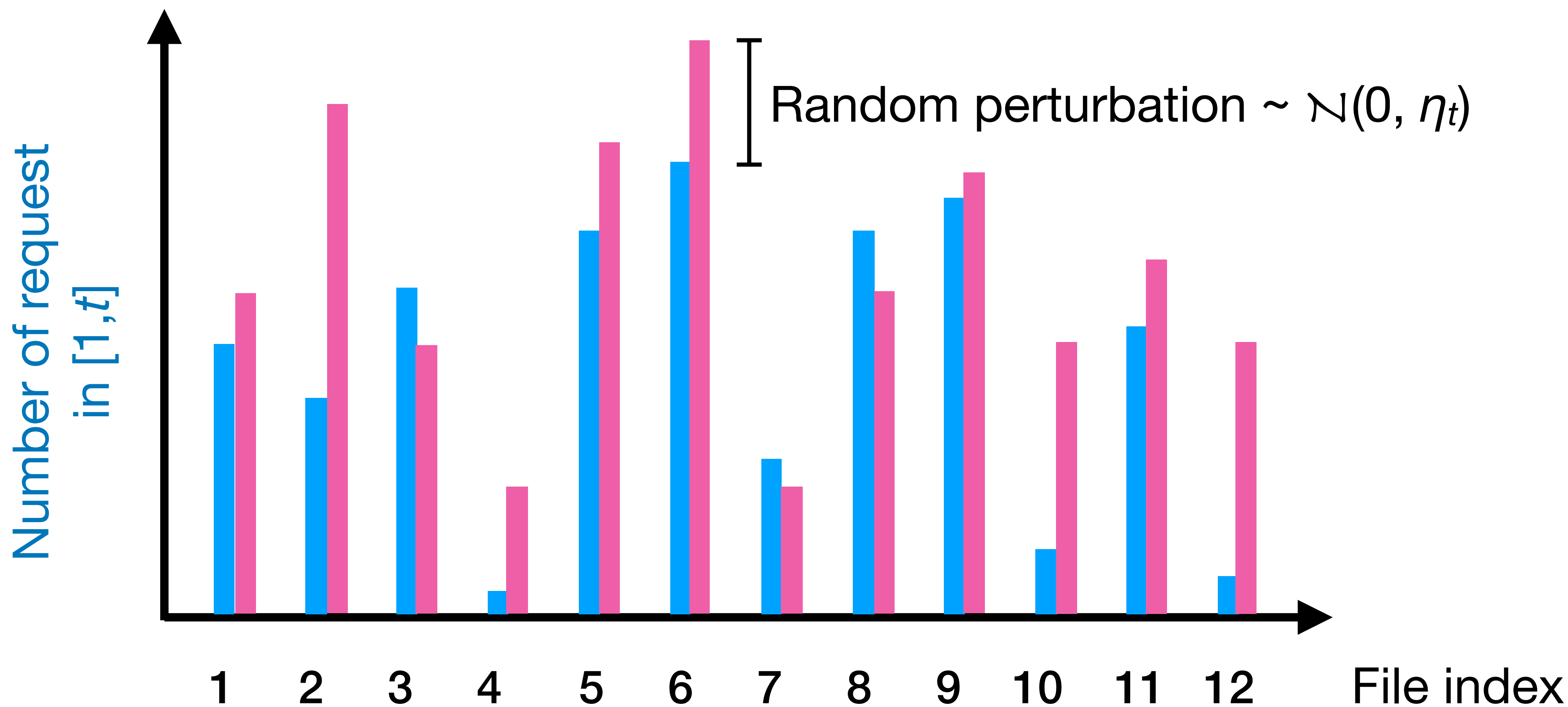
- $\mu_i = \mathbb{P}$(an incoming request is for file i)
- WLOG, files indexed in decreasing order of μ_is
- $\Delta = \mu_C - \mu_{C+1}$

Theorem: For i.i.d. stochastic arrivals and $\eta_t \propto \sqrt{t}$:

$$R_{\mathrm{FTPL}}(T) = O(\log L/\Delta^2).$$

FTPL with $\eta_t \propto \sqrt{t}$ has order-optimal regret (w.r.t. time) for i.i.d. stochastic and adversarial arrivals

Recall: Follow the Perturbed Leader



- Keep track of cumulative number of requests for each file
- Score(t) = cumulative number of requests in $[0, t]$ + random perturbation
- Cache the C files with the C highest scores

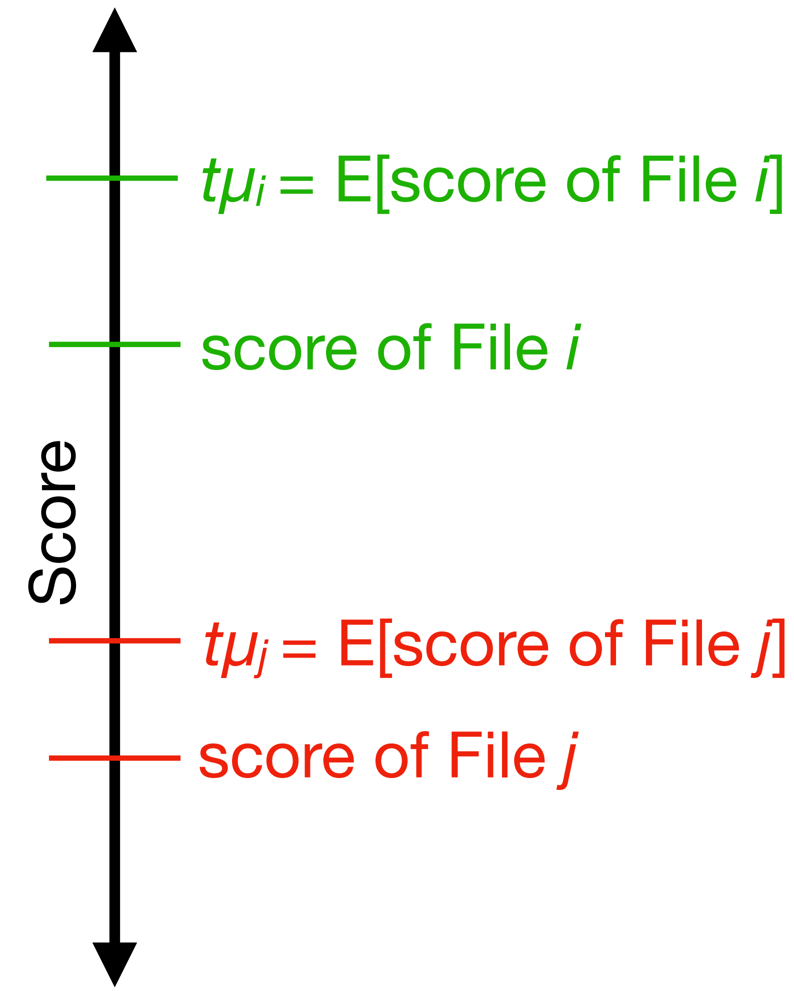
Proof Outline (Part 1 of 2)

Let

- $\mu_i = \mathbb{P}$(an incoming request is for file i)
- WLOG, $i < j \implies \mu_i > \mu_j$

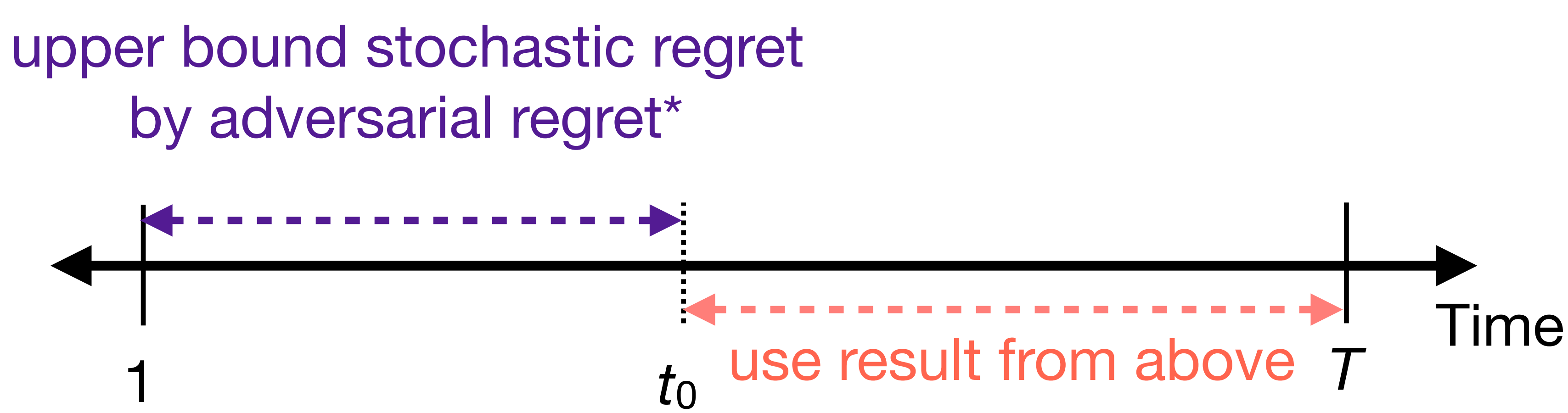
Key idea: Low regret if FTPL mimics OPT w.h.p.

- OPT caches Files 1 to C
- Consider $i \leq C$ and $j > C \implies \mu_i > \mu_j$
- Event E: score of File i > score of File j
- Lower bound $\mathbb{P}$(E)
- Account for all possible pairs of $i \leq C$ & $j > C$ and all time



Proof Outline (Part 2 of 2)

- Event E: score of File i > score of File j
- Account for **all possible pairs** of $i \leq C$ and $j > C$ and **all time**



- Optimize for t_0, improves dependence of regret bound on library size L

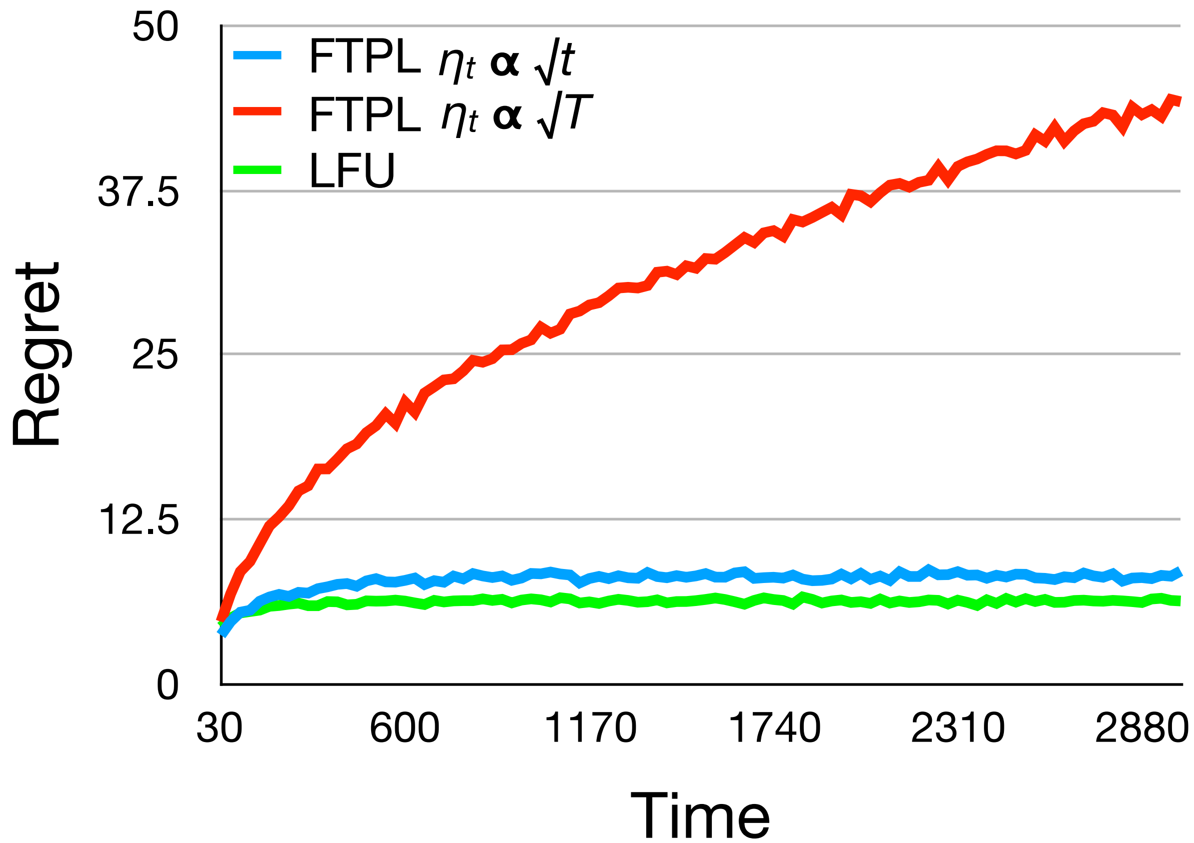
---

*J. Mourtada et al., On the optimality of the Hedge algorithm in the stochastic regime, *JMLR* 2019

Simulations

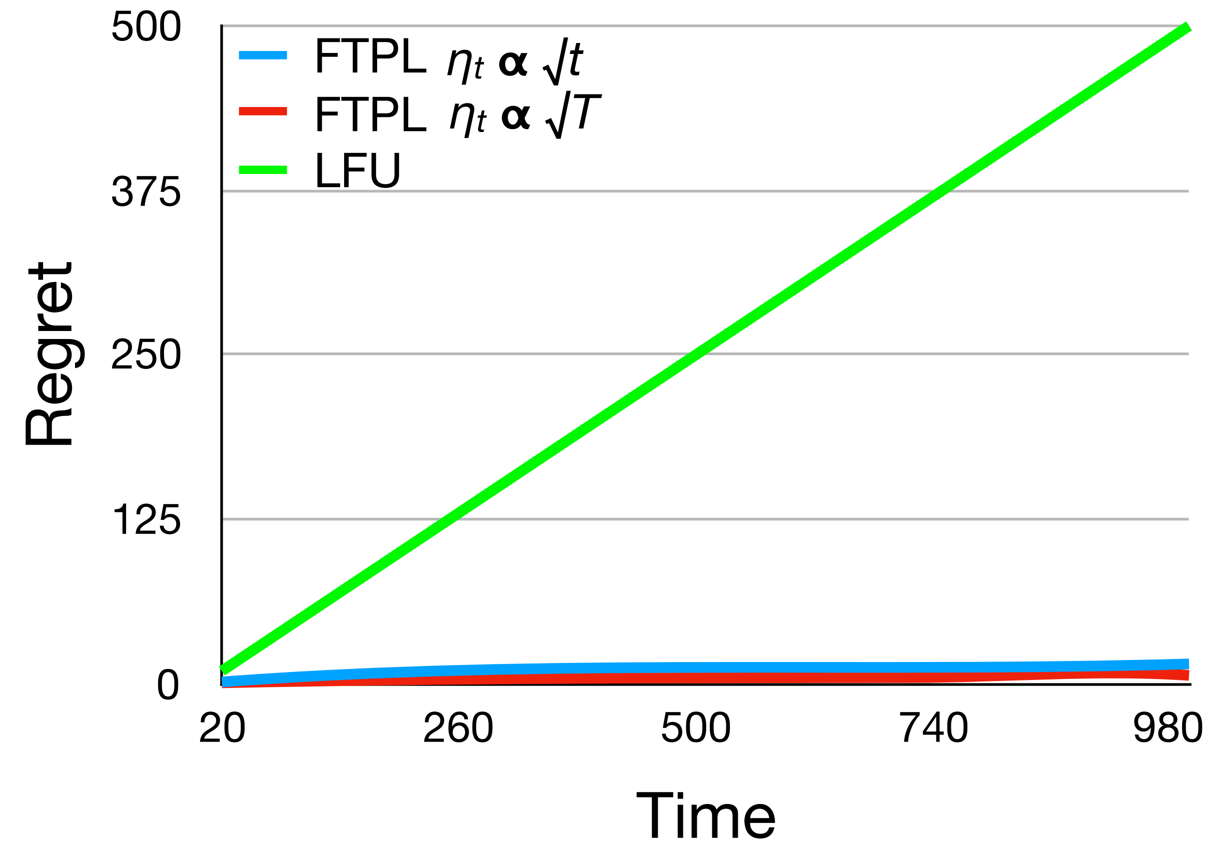
i.i.d. stochastic arrivals

C = 4, L = 10, $\mu_i = 2^{-i}$ for $i<L$, $\mu_L = 2^{-L+1}$



round robin arrivals

C = 1, L = 2



Summary

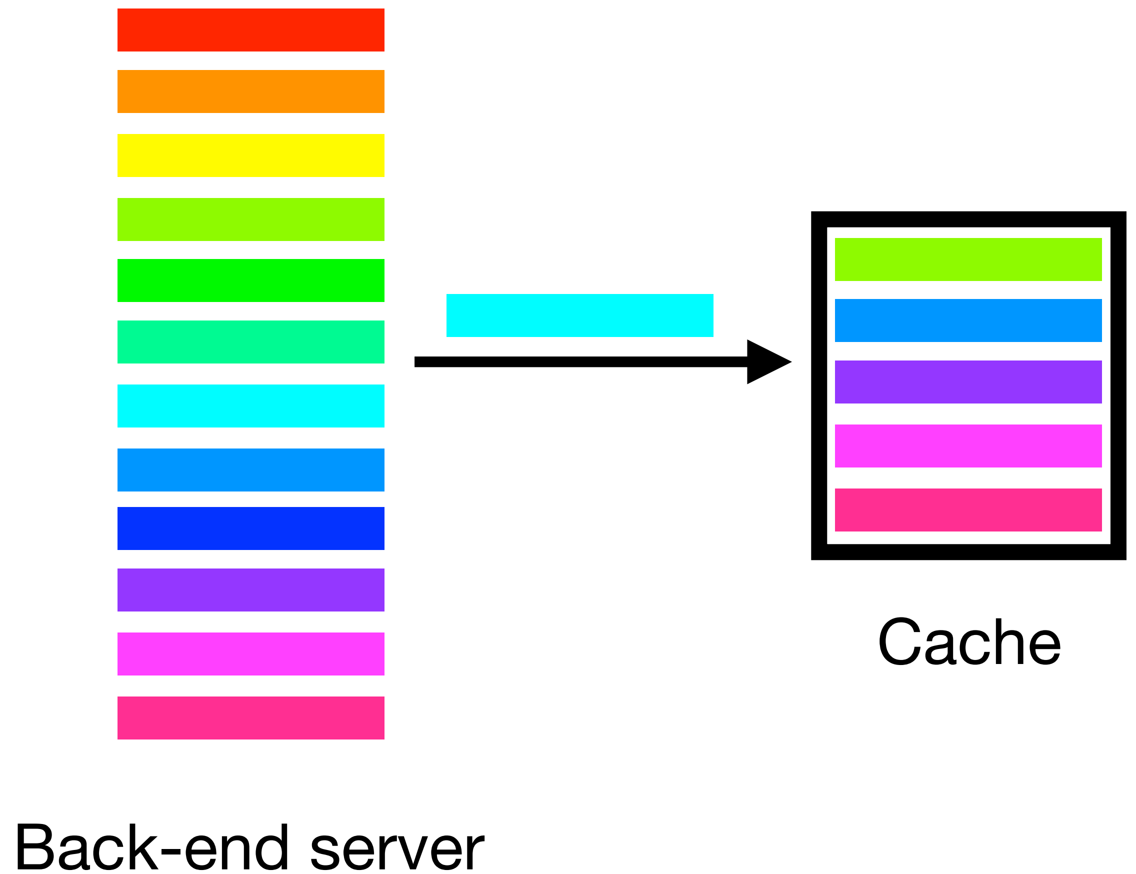
Policies	i.i.d. Stochastic Requests	Adversarial Requests
LFU	O(1) regret (order-optimal)[1]	$\Omega(T)$ regret, strictly sub-optimal[2]
FTPL	$\Omega(\sqrt{T})$ regret for $\eta_t \propto \sqrt{T}$ (sub-optimal) **O(1) regret for $\eta_t \propto \sqrt{t}$ (order-optimal)**	$O(\sqrt{T})$ regret for $\eta_t \propto \sqrt{T}$ (order-optimal)[2] **$O(\sqrt{T})$ regret for $\eta_t \propto \sqrt{t}$ (order-optimal)[3]**

FTPL with $\eta_t = \sqrt{t}$ has order-optimal regret for
both stochastic & adversarial arrivals

---

1 A. Bura et al., Learning to Cache and Caching to Learn: Regret Analysis of Caching Algorithms, *IEEE/ACM ToN*
2 R. Bhattacharjee et al., Fundamental Limits of Online Network-Caching, *ACM SIGMETRICS* 2020
3 S. Mukhopadhyay et al., Online Caching with Optimal Switching Regret, *ISIT* 2021

Generalizations



- So far, no penalty for changing cache contents
- Generalizations: restricted switching and switching at a cost

Is there a policy that has order-optimal (w.r.t. time) regret
for both stochastic & adversarial arrivals?

Restricted Switching

Setting: cache contents can only be changed every r time-slots

	i.i.d. Stochastic Requests	Adversarial Requests
Lower bound	$\Omega(r)$	$\Omega(\sqrt{rT})$
FTPL	**O(r) regret for $\eta_t = \sqrt{t}$ (order-optimal)**	**O($\sqrt{rT}$) regret for $\eta_t \propto \sqrt{t}$ (order-optimal)**

FTPL with $\eta_t = \sqrt{rt}$ has order-optimal (w.r.t. time) regret
for both stochastic & adversarial arrivals

Extension: non-uniform gaps between changes to cache contents

Switching at a Cost

Setting: every change to cache contents costs D units
Updated regret definition takes into account the switching cost

Recall:

- Random perturbation in time-slot $t \sim \mathcal{N}(0, (\eta_t)^2)$
- L = library size
- C = cache size

Let:

- $\mu_i = \mathbb{P}$(an incoming request is for file i)
- WLOG, files indexed in decreasing order of μ_is
- $\Delta = \mu_C - \mu_{C+1}$

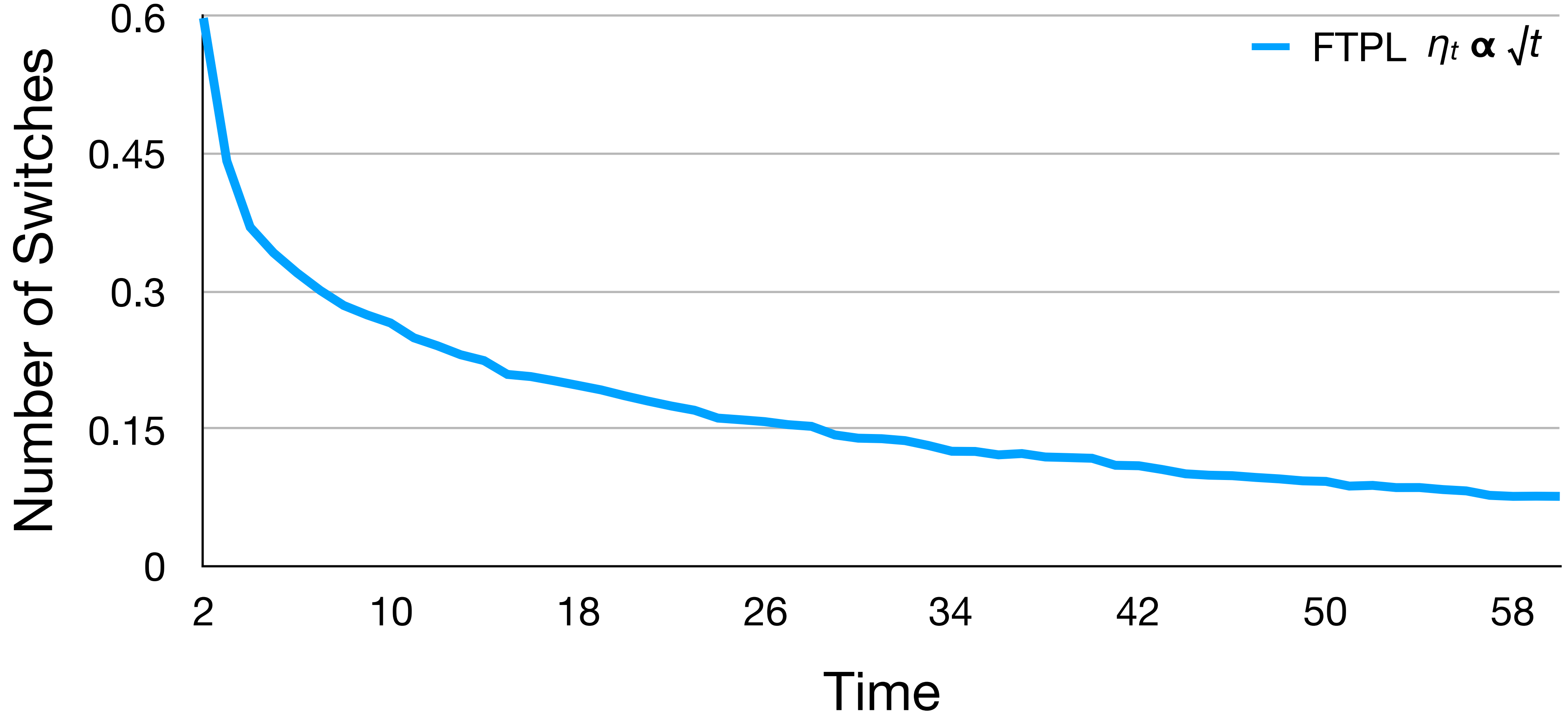
Theorem: For i.i.d. stochastic arrivals and $\eta_t \propto \sqrt{t}$:

$$R_{\mathrm{FTPL}}(T) = O(D \log L/\Delta^2).$$

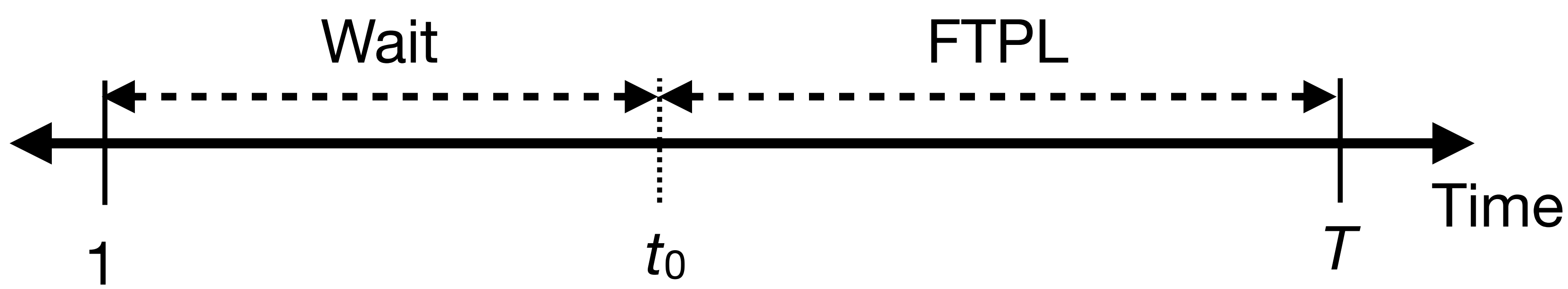
Switches under FTPL

i.i.d. stochastic arrivals

C = 2, L = 5, $\boldsymbol{\mu}$ = [0.5 , 0.25 , 0.125 , 0.0625, 0.0625]



Our Policy: Wait-then-FTPL



Wait-then-FTPL: If $t < t_0$, do nothing, else mimic FTPL

- Number of misses increases with the duration of the wait period
- Switch cost decreases with the duration of the wait period
- To balance this trade-off: $t_0 = u(\log D)^{1+\alpha}$ for $u, \alpha \geq 0$

Performance of Wait-then-FTPL

Setting: every change to cache contents costs D units
Updated regret definition takes into account the switch cost

Recall:
- Random perturbation in time-slot $t \sim \mathcal{N}(0, (\eta_t)^2)$
- L = library size, C = cache size

Let:
- $\mu_i = \mathbb{P}$(an incoming request is for file i)
- WLOG, files indexed in decreasing order of μ_is
- $\Delta = \mu_C - \mu_{C+1}$

Theorem: For i.i.d. stochastic arrivals and $\eta_t \propto \sqrt{t}$:

$$R_{\text{Wait-then-FTPL}}(T) = O((\log D)^{1+\alpha} \log L/\Delta^2).$$

Recall: $R_{\text{FTPL}}(T) = O(D \log L/\Delta^2)$

Performance of Wait-then-FTPL

Setting: every change to cache contents costs D units
Updated regret definition takes into account the switch cost

Recall:

- Random perturbation in time-slot $t \sim \mathcal{N}(0, (\eta_t)^2)$
- L = library size
- C = cache size

Let:

- $\mu_i = \mathbb{P}$(an incoming request is for file i)
- WLOG, files indexed in decreasing order of μ_is
- $\Delta = \mu_C - \mu_{C+1}$

Theorem: For adversarial arrivals and $\eta_t \propto \sqrt{t}$:

$$R_{\text{Wait-then-FTPL}}(T) = O(D\sqrt{T}).$$

Summary (with Switching Cost)

Recall: D units of cost incurred for each switch

	i.i.d. Stochastic Requests	**Adversarial Requests**
FTPL	O(D) regret for $\eta_t \propto \sqrt{t}$ (order-optimal w.r.t. time)	O($D\sqrt{T}$) regret for $\eta_t \propto \sqrt{t}$ (order-optimal w.r.t. time)[1]
Wait-then-FTPL	O($(\log D)^{1+\alpha}$) regret for $\eta_t \propto \sqrt{t}$ (order-optimal w.r.t. time)	O($D\sqrt{T}$) regret for $\eta_t \propto \sqrt{t}$ (order-optimal w.r.t. time)

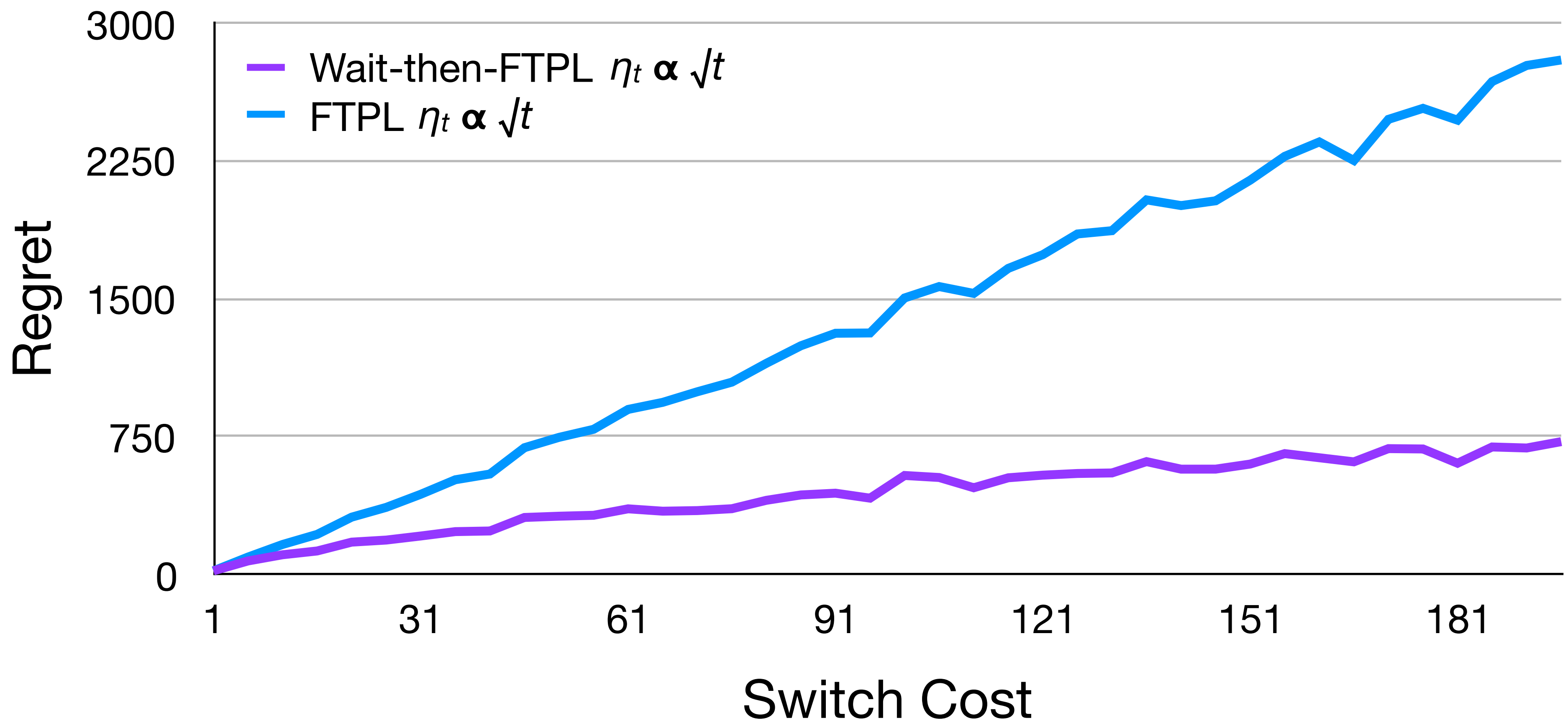
FTPL and W-FTPL with $\eta_t \propto \sqrt{t}$ have order-optimal (w.r.t. time) regret for both stochastic & adversarial arrivals

1 S. Mukhopadhyay et al., Online Caching with Optimal Switching Regret, *ISIT* 2021

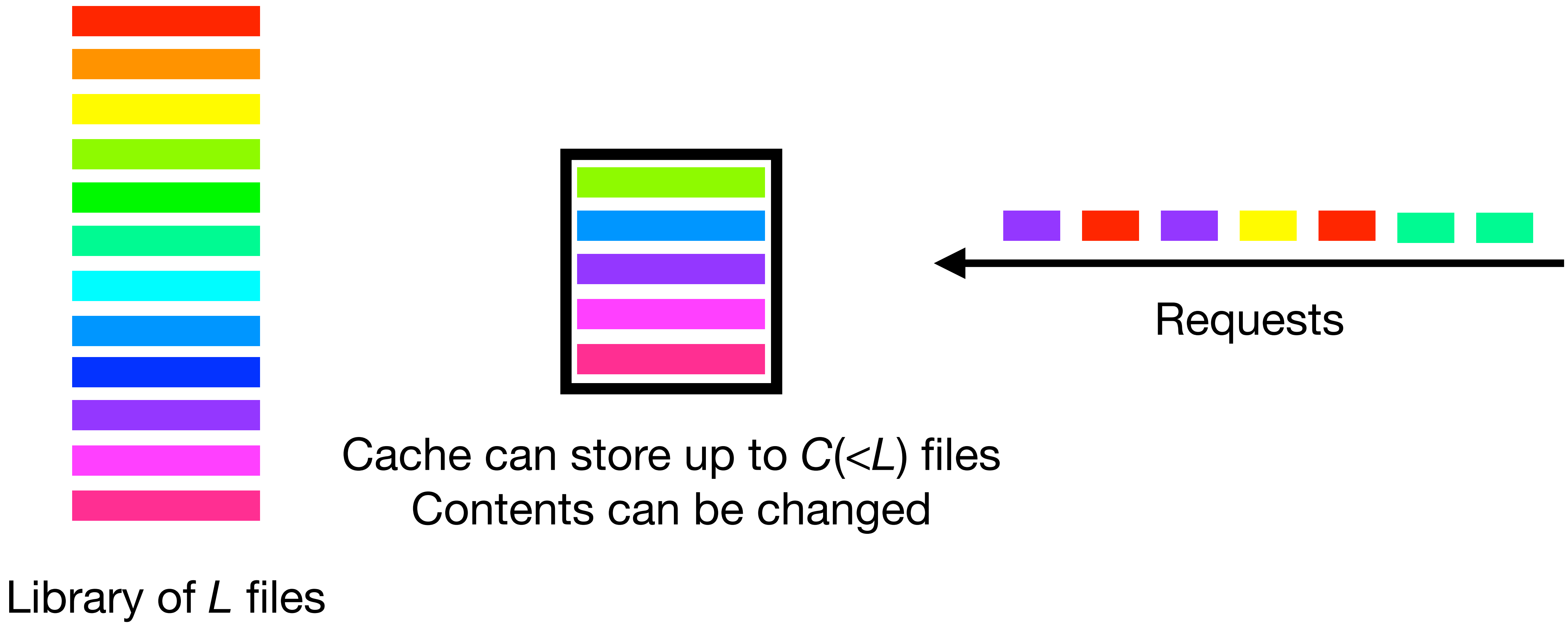
Simulations

i.i.d. stochastic arrivals

C = 2, L = 5, $\boldsymbol{\mu}$ = [0.5 , 0.25 , 0.125 , 0.0625, 0.0625], a = 0, u = 5, T= 200



Conclusions



- Studied the online caching problem, performance metric: regret
- FTPL has order-optimal regret for stochastic and adversarial arrivals
- FTPL can have poor performance in the presence of switching cost
- Our variant Wait-then-FTPL addresses this limitation of FTPL

Thanks!