# Alfie: Neural-Reinforced Adaptive Prefetching for Short Videos

**Jingzong Li**, Hong Xu *, Ming Ma, Haopeng Yan, Chun Jason Xue

香港城市大學
City University of Hong Kong

香港中文大學
The Chinese University of Hong Kong

KUAISHOU

# Short videos become more and more pervasive



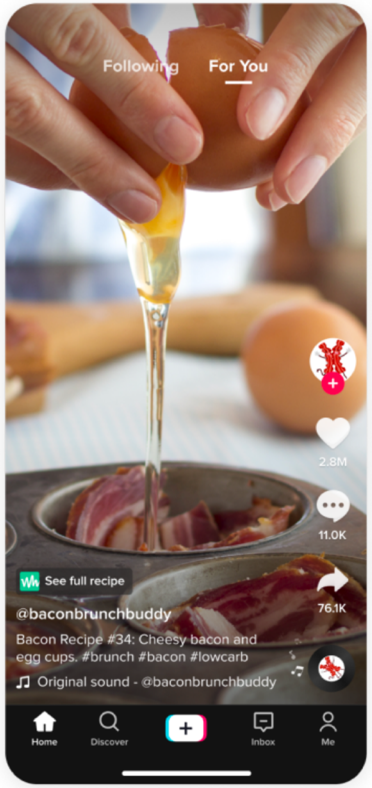
**The number of short video users has reached 873 million in China, representing 88.3% of its total netizens [1].**

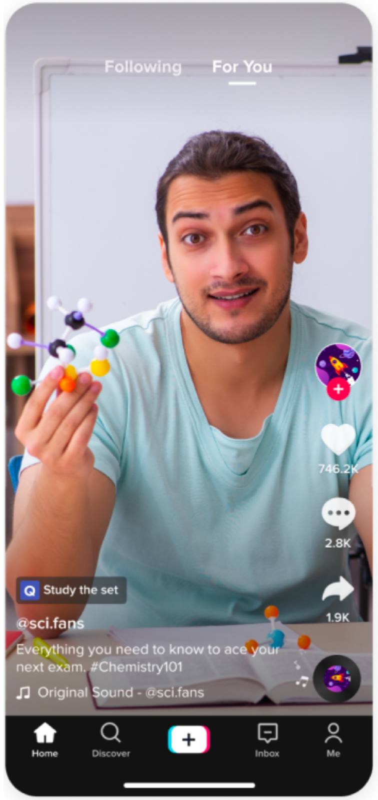
TikTok

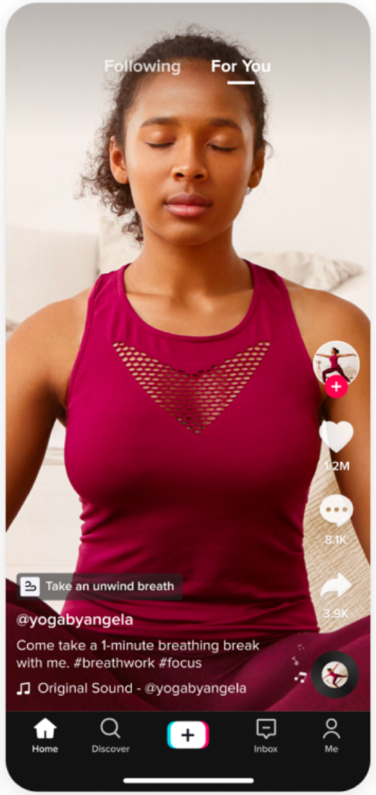
Kuaishou

Instagram Reels

YouTube Shorts



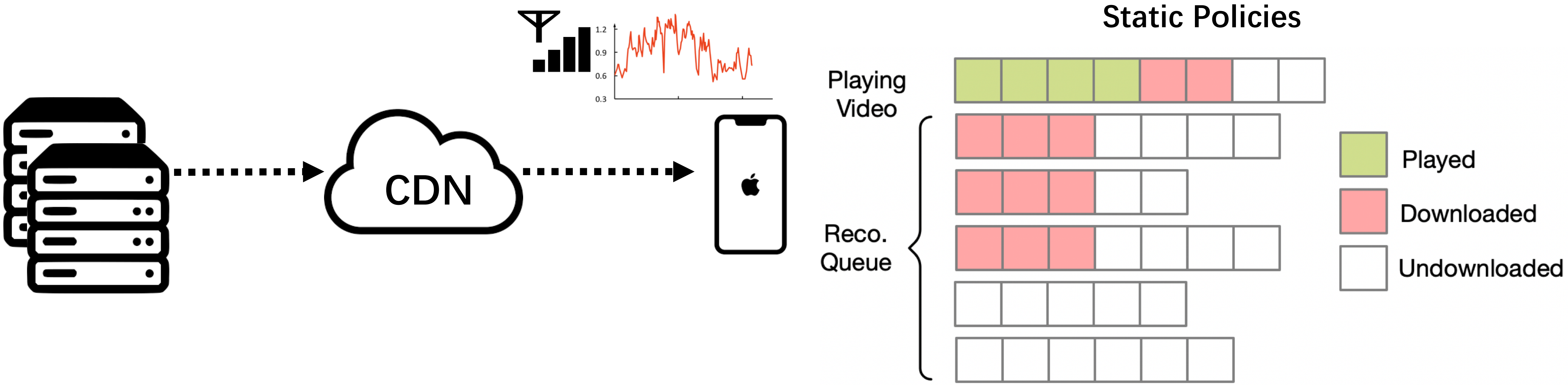




[1] Statistical Report on China's Internet Development in 2021, CNNIC

# Prefetching Policy



To provide **smooth playback** and avoid rebuffering delay, **prefetching** upcoming videos is commonly used in short videos.




However, **static policies** used in production lead to substantial **bandwidth overhead**, especially the **exit overhead**.

# Motivation Measurement



**Goal**: To quantify the bandwidth overhead of static prefetching.

**Data Preparation**: We collect a production trace of over **400 million** sessions of short video viewing for a **24-hour** period starting on March 1st, 2021 from a large short video company.

**Schemes:** we consider two representative instances of static policy:

- **S-3-3**: downloads the first 3 videos with first 3 chunks sequentially.
- **S-5-6**: downloads the first 5 videos with first 6 chunks sequentially.

**Metrics**:

- $T$: Rebuffering time (ms)
- $D$: Startup delay (ms): the lag between the user swiping and playing
- $W_s$: Swiping overhead (KB): downloaded but unwatched content due to user swiping
- $W_e$: Exit Overhead (KB): downloaded but unwatched content due to user exiting

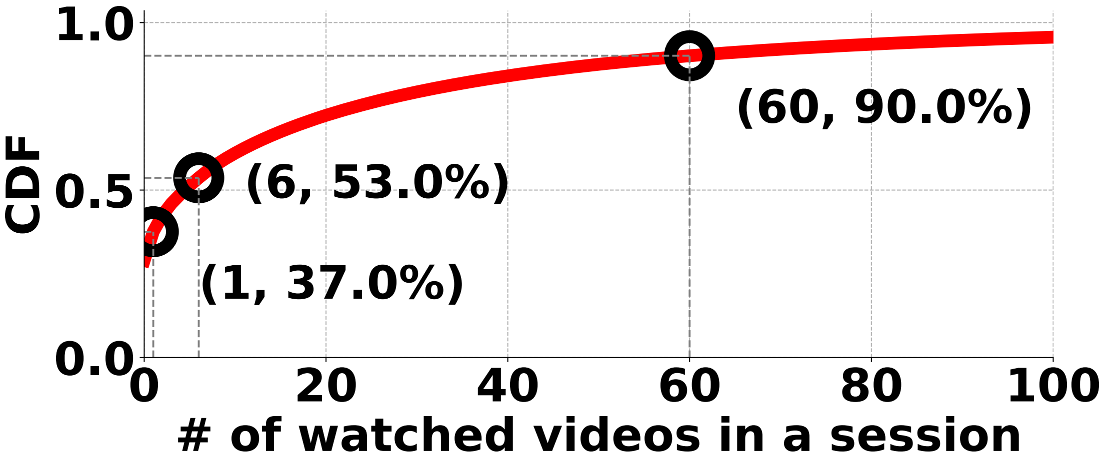
## Key Finding 1:

Static prefetching results in **significant bandwidth overhead**, including **exit overhead**.

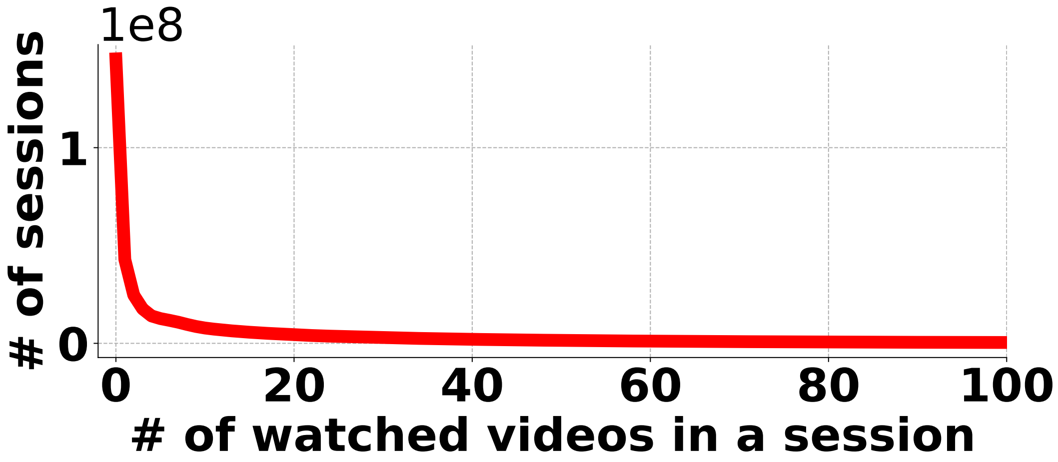
| Scheme | Daily Bandwidth Overhead (TB) | Daily Exit Overhead (TB) & Ratio | Annual Bandwidth Overhead Cost Range ($1M USD) | Annual Exit Overhead Cost Range ($1M USD) |
|---|---|---|---|---|
| *S-5-6* | 2795 | 1216 (43.5%) | [~41, ~122] | [~18, ~53] |
| *S-3-3* | 1738 | 365 (21.0%) | [~25, ~76] | [~5, ~16] |

## Key Finding 2:

User watching behavior is **long-tailed**.




(a) CDF of number of short videos viewed in each session

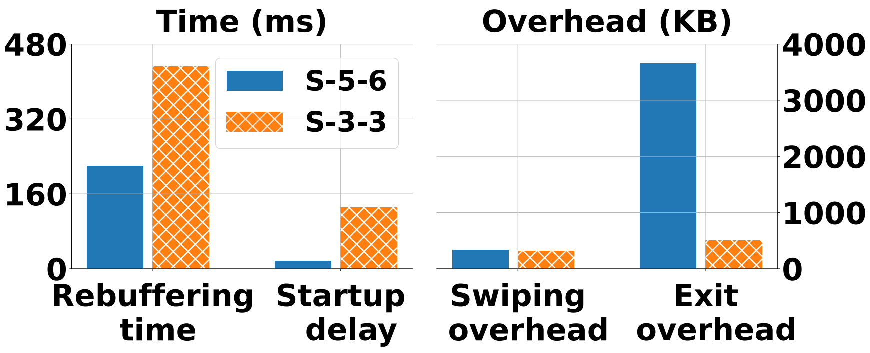



(b) Histogram of number of videos viewed in each session.

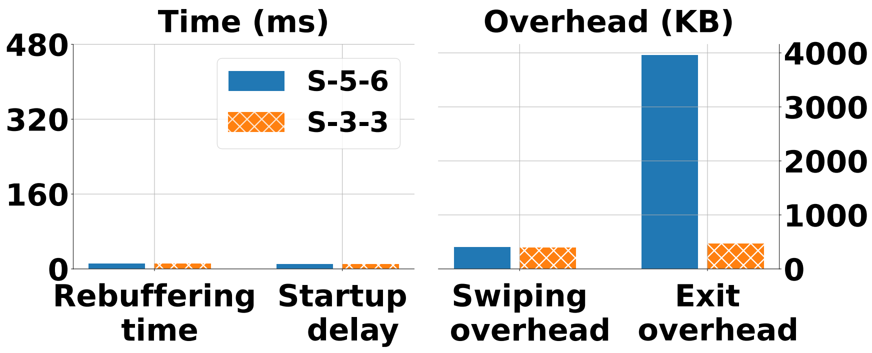
# Motivation Measurement



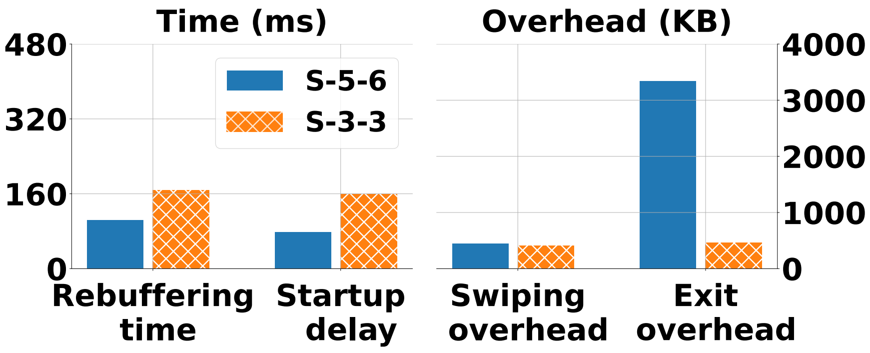
## **Key Finding 3**: Static policies **do not adapt** well.



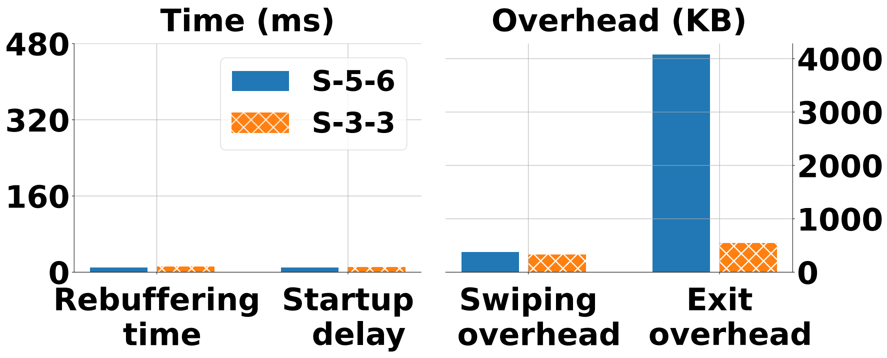

(a) 2Mbps bandwidth




(b) 10Mbps bandwidth



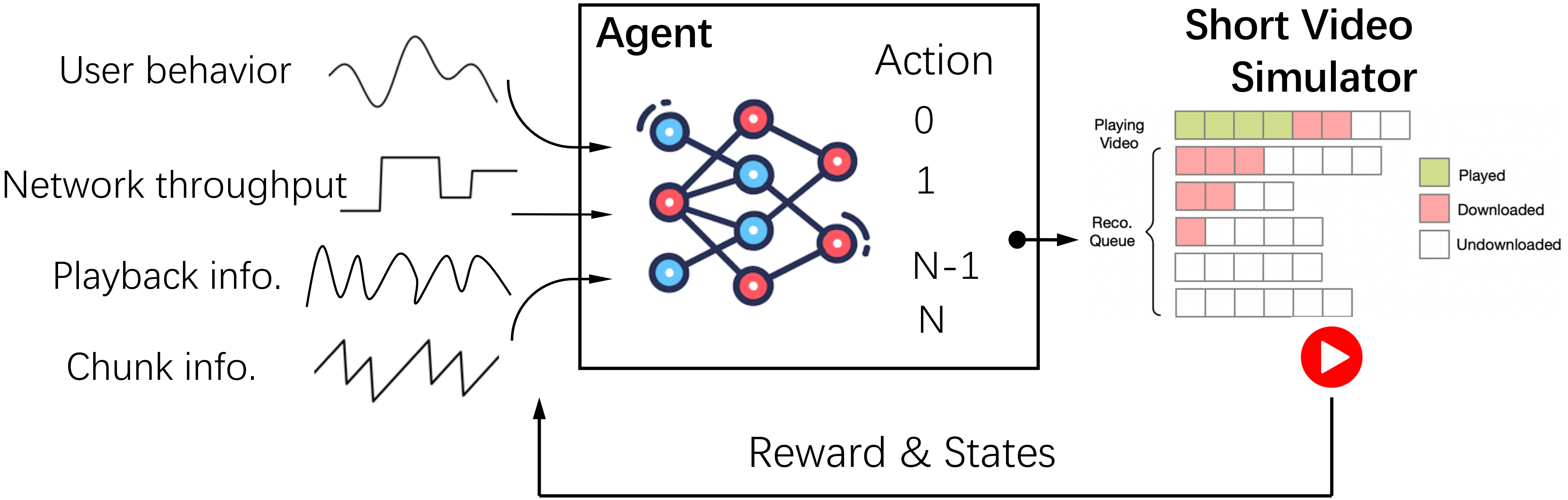

(c) Fast swiping




(d) Slow swiping

# Alfie Design

**Alfie**:  a bandwidth-efficient short video prefetching algorithm that can dynamically adjust the prefetching strategy via **reinforcement learning**.



# Alfie Design

**Reward Function Shaping**: We design a reward function specialized for short video streaming.

$$R(S_i, A_i, S_{i+1}) = \begin{cases} R_{\text{idle}}(S_i, S_{i+1}), & \text{if } A_i = 0, \\ R_{\text{prefetch}}(S_i, A_i, S_{i+1}), & \text{otherwise.} \end{cases}$$

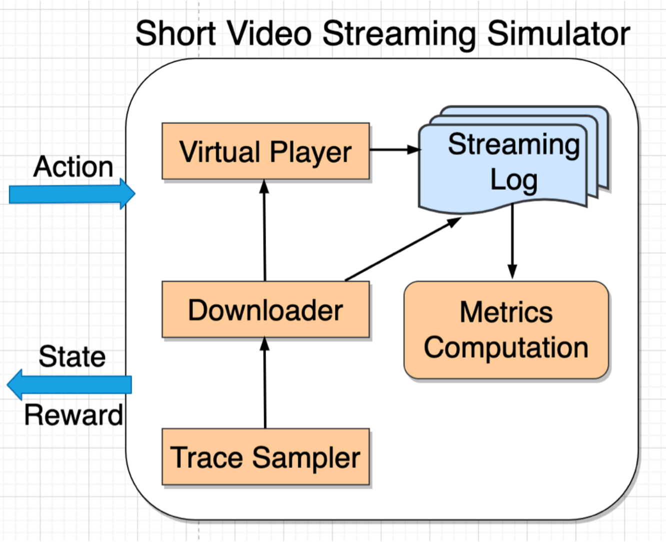
**Algorithm 1** Calculation of reward $R_{\text{prefetch}}$

**Input:** $A_i$: video selected for prefetching; $j$: position of the chunk to be prefetched in $A_i$; $T(S_i, A_i, S_{i+1})$: rebuffering time during downloading chunk $j$

$h \leftarrow$ GetVideoNum(VideoId, SessionTrace) ▷ Get the number of remaining videos from the session trace; VideoId is the current video's ID
**if** $T(S_i, A_i, S_{i+1}) > 0$ **then**
  $R_{\text{prefetch}} \leftarrow -1$ ▷ punishment for rebuffering
**else if** $A_i > h$ **then**
  $R_{\text{prefetch}} \leftarrow -1$ ▷ punishment for prefetching a video that will not be watched due to user exit
**else**
  $s \leftarrow$ GetStayingTime($A_i$, SessionTrace) ▷ Get the time user spent on video $A_i$ from the trace
  **if** $j > s$ **then**
    $R_{\text{prefetch}} \leftarrow -1$ ▷ punishment for downloading a chunk that will not be viewed due to user swiping
  **else**
    $R_{\text{prefetch}} \leftarrow \beta * (s - j)/s$
**return** $R_{\text{prefetch}}$

**Slow Start Mechanism**

**Short Video Streaming Simulator**




Please refer to the paper for details.

# Evaluation – Overall performance



## Baselines:

- Oracle (Upper bound)
- Next-one
- S-3-3
- S-5-6
- S-5-12
- LiveClip [NOSSDAV'20]

## Metrics:

- $T$: Rebuffering time (ms)
- $D$: Startup delay (ms)
- $W_s$: Swiping overhead (KB)
- $W_e$: Exit Overhead (KB)
- Negative utility:

$$U = T + D + 0.1 \times W_s + 0.1 \times W_e$$

## Dataset:

- Network traces: kuaishou trace and public trace
- Session traces

| Network | Scheme | Rebuffering time (ms) | Startup delay (ms) | Swiping overhead (KB) | Exit overhead (KB) | Negative Utility |
|---|---|---|---|---|---|---|
| Kuaishou trace | Oracle | 249 | 111 | 7 | 0 | 365 |
| | Next-One | 444 | 252 | 2116 | 21149 | 17616 |
| | S-3-3 | 594 | 141 | 194 | 475 | 1222 |
| | S-5-6 | 464 | 114 | 224 | 3873 | 3559 |
| | S-5-12 | 435 | 118 | 343 | 7228 | 6061 |
| | LiveClip | 488 | 129 | 1383 | 2965 | 3779 |
| | Alfie | 318 | 121 | 247 | 543 | 1014 |
| Public trace | Oracle | 119 | 75 | 4 | 0 | 197 |
| | Next-One | 232 | 170 | 2353 | 26078 | 21080 |
| | S-3-3 | 303 | 90 | 221 | 506 | 922 |
| | S-5-6 | 232 | 81 | 244 | 3993 | 3395 |
| | S-5-12 | 217 | 81 | 371 | 7586 | 6085 |
| | LiveClip | 246 | 86 | 1342 | 3146 | 3596 |
| | Alfie | 156 | 83 | 271 | 407 | 733 |

Upper bound

Upper bound

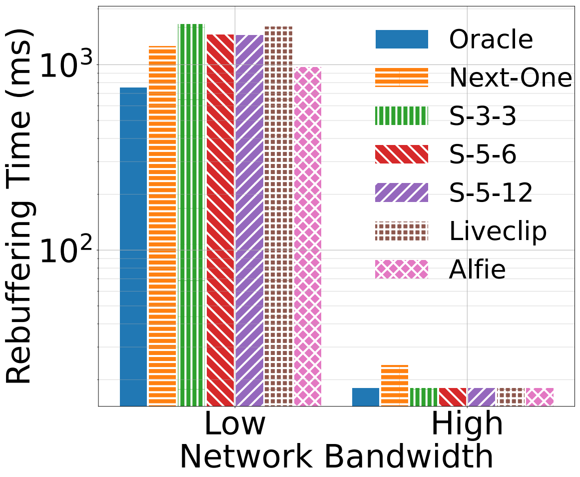
Table: Overall performance

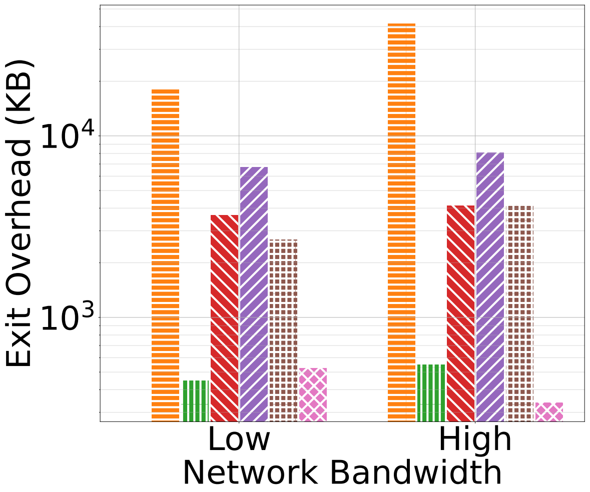
# Does Alfie generalize?

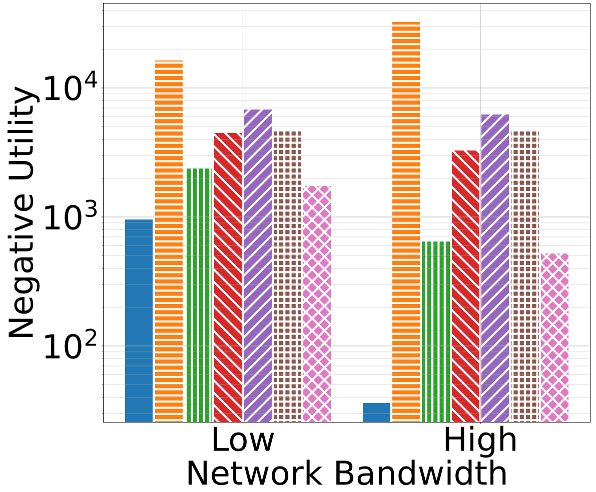


Rebuffering time | Exit Overhead | Negative utility

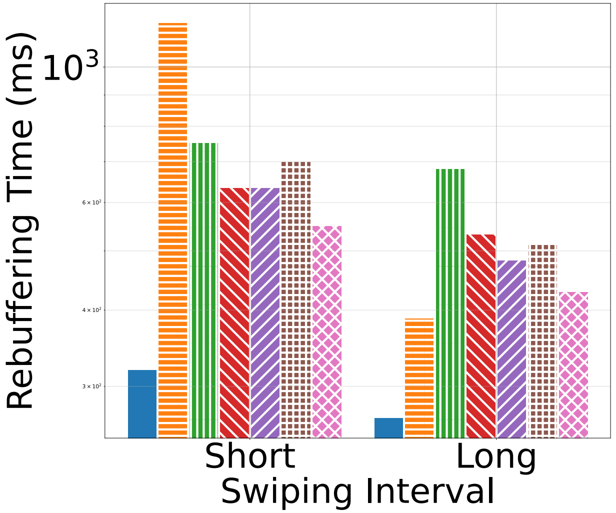
Different network conditions

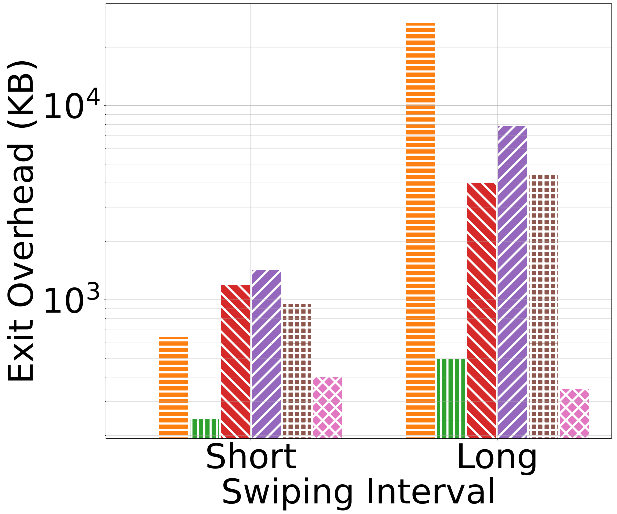


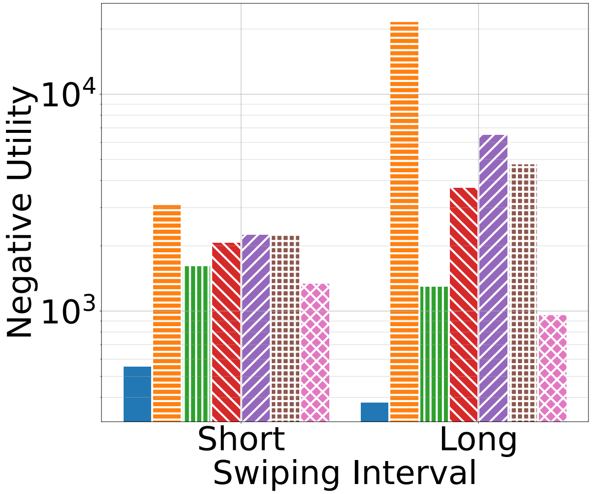




Different behavior patterns




Alfie delivers 18.9%–26.8% improvement in overall utility over existing methods

# Takeaway



- **Exit overhead is non-negligible** when designing a bandwidth-efficient prefetching policy.
- Prefetching is intrinsically a **sequential** and **far-sighted** process which perfectly fits for DRL.
- A **high-fidelity** short video streaming **simulator** is important to train the prefetching algorithm.
- Alfie is able to **adapt** to variable and unseen environments by learning from massive past experiences.

# Alfie: Neural-Reinforced Adaptive Prefetching for Short Videos

# Thank you

{ jingzong.li@my.cityu.edu.hk }