Alfie: Neural-Reinforced Adaptive Prefetching for Short Videos

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

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Daily Bandwidth Overhead (TB)</th>
<th>Daily Exit Overhead (TB) &amp; Ratio</th>
<th>Annual Bandwidth Overhead Cost Range ($1M USD)</th>
<th>Annual Exit Overhead Cost Range ($1M USD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-5-6</td>
<td>2795</td>
<td>1216 (43.5%)</td>
<td>[~41, ~122]</td>
<td>[~18, ~53]</td>
</tr>
<tr>
<td>S-3-3</td>
<td>1738</td>
<td>365 (21.0%)</td>
<td>[~25, ~76]</td>
<td>[~5, ~16]</td>
</tr>
</tbody>
</table>
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.
Key Finding 3: Static policies do not adapt well.

(a) 2Mbps bandwidth

(b) 10Mbps bandwidth

(c) Fast swiping

(d) Slow swiping
Alfie: a bandwidth-efficient short video prefetching algorithm that can dynamically adjust the prefetching strategy via reinforcement learning.
**Reward Function Shaping:** We design a reward function specialized for short video streaming.

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

**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

<table>
<thead>
<tr>
<th>Network</th>
<th>Scheme</th>
<th>Rebuffering time (ms)</th>
<th>Startup delay (ms)</th>
<th>Swiping overhead (KB)</th>
<th>Exit overhead (KB)</th>
<th>Negative Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kuaishou trace</td>
<td>Oracle</td>
<td>249</td>
<td>111</td>
<td>7</td>
<td>0</td>
<td>365</td>
</tr>
<tr>
<td></td>
<td>Next-One</td>
<td>444</td>
<td>252</td>
<td>2116</td>
<td>21149</td>
<td>17616</td>
</tr>
<tr>
<td></td>
<td>S-3-3</td>
<td>594</td>
<td>141</td>
<td>194</td>
<td>475</td>
<td>1222</td>
</tr>
<tr>
<td></td>
<td>S-5-6</td>
<td>464</td>
<td>114</td>
<td>224</td>
<td>3873</td>
<td>3559</td>
</tr>
<tr>
<td></td>
<td>S-5-12</td>
<td>435</td>
<td>118</td>
<td>343</td>
<td>7228</td>
<td>6061</td>
</tr>
<tr>
<td></td>
<td>LiveClip</td>
<td>488</td>
<td>129</td>
<td>1383</td>
<td>2965</td>
<td>3779</td>
</tr>
<tr>
<td></td>
<td>Alfie</td>
<td>318</td>
<td>121</td>
<td>247</td>
<td>543</td>
<td>1014</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network</th>
<th>Scheme</th>
<th>Rebuffering time (ms)</th>
<th>Startup delay (ms)</th>
<th>Swiping overhead (KB)</th>
<th>Exit overhead (KB)</th>
<th>Negative Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public trace</td>
<td>Oracle</td>
<td>119</td>
<td>75</td>
<td>4</td>
<td>0</td>
<td>197</td>
</tr>
<tr>
<td></td>
<td>Next-One</td>
<td>232</td>
<td>170</td>
<td>2353</td>
<td>26078</td>
<td>21080</td>
</tr>
<tr>
<td></td>
<td>S-3-3</td>
<td>303</td>
<td>90</td>
<td>221</td>
<td>506</td>
<td>922</td>
</tr>
<tr>
<td></td>
<td>S-5-6</td>
<td>232</td>
<td>81</td>
<td>244</td>
<td>3993</td>
<td>3395</td>
</tr>
<tr>
<td></td>
<td>S-5-12</td>
<td>217</td>
<td>81</td>
<td>371</td>
<td>7586</td>
<td>6085</td>
</tr>
<tr>
<td></td>
<td>LiveClip</td>
<td>246</td>
<td>86</td>
<td>1342</td>
<td>3146</td>
<td>3596</td>
</tr>
<tr>
<td></td>
<td>Alfie</td>
<td>156</td>
<td>83</td>
<td>271</td>
<td>407</td>
<td>733</td>
</tr>
</tbody>
</table>

Table: Overall performance
Does Alfie generalize?

Rebuffering time

Different network conditions

Different behavior patterns

Exit Overhead

Negative utility

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 }