Gamora: Learning-based Buffer-aware Preloading for Adaptive Short Video Streaming

Biao Hou, Song Yang, *Senior Member, IEEE,* Fan Li, *Member, IEEE,* Liehuang Zhu, *Senior Member, IEEE,* Lei Jiao, *Member, IEEE,* Xu Chen, *Senior Member, IEEE,* and Xiaoming Fu, *Fellow, IEEE*

Abstract—Nowadays, the emerging short video streaming applications have gained substantial attention. With the rapidly burgeoning demand for short video streaming services, maximizing their Quality of Experience (QoE) is an onerous challenge. Current video preloading algorithms cannot determine video preloading sequence decisions appropriately due to the impact of users' swipes and bandwidth fluctuations. As a result, it is still ambiguous how to improve the overall QoE while mitigating bandwidth wastage to optimize short video streaming services. In this paper, we devise Gamora, a buffer-aware short video streaming system to provide a high QoE of users. In Gamora, we first propose an unordered preloading algorithm that utilizes a Deep Reinforcement Learning (DRL) algorithm to make video preloading decisions. Then, we further devise an Asymmetric Imitation Learning (AIL) algorithm to guide the DRL-based preloading algorithm, which enables the agent to learn from expert demonstrations for fast convergence. Finally, we implement our proposed short video streaming system prototype and evaluate the performance of Gamora on various real-world network datasets. Our results demonstrate that Gamora significantly achieves QoE improvement by 28.7%-51.4% compared to state-of-the-art algorithms, while mitigating bandwidth wastage by 40.7%-83.2% without sacrificing video quality.

Index Terms—Short video streaming, Preloading, Buffer management, Asymmetric imitation learning.

1 Introduction

In recent years, short video streaming service has become increasingly popular among users. Commercial short video applications such as TikTok, Vine, and Instagram Reels attract billions of active users monthly and consistently top popularity lists for mobile apps [1]. According to a recent report, TikTok application attains 1.5 billion monthly active users in 2023, with projections indicating an anticipated surge to over two billion by the end of 2024 [2]. Unlike traditional long-form videos, short videos typically have a median video duration of around 100 seconds [3]. Specifically, users may swipe at any position in short videos and expect seamless playback for subsequent videos upon a swipe [4]. To provide immersive and interactive Quality of Experience (QoE) for users, short video service providers have striven to optimize video quality and reduce the rebuffering ratio [5], [6]. Nevertheless, arbitrary user swipes may cause discernible startup latency and bandwidth wastage, especially when short videos remain undownloaded or downloaded but never viewed. Consequently, it is indeed challenging for short video service platforms to improve the overall QoE while not concurrently raising bandwidth wastage.

Considering about the unique characteristics of short videos, it is ill-suited to directly apply the conventional long-form video prefetching algorithm to tailor the preloading sequences for short videos [7], [8], [9], [10]. Since current prefetching algorithms usually focus on obtaining high video bitrates and low rebuffering ratios without considering the impact of users' swipe events and bandwidth wastage on system performance. Moreover, existing prefetching algorithms assume that users would watch short video content sequentially to completion and hence download short videos in order. However, users' swipes are complicated, and swipe times determine the video chunks that will be viewed or skipped during a video session [11]. The video download sequence is not exactly the same as the video playback order due to user-specific swipe behaviors. Short video streaming service usually adopts a preloading algorithm [12] to determine the video buffering order and corresponding bitrate simultaneously. To this end, we need to appropriately preload short videos into the client buffer under dynamic network conditions. Preloading as many short videos as possible into the buffer under its limit ensures the desired QoE during playback but it may lead to substantial bandwidth wastage. Nevertheless, inadequate buffering may cause tremendous startup latency [13] when swiping to the subsequent video that is not yet downloaded.

The existing efforts on solving the video preloading problem in short video streaming can be broadly categorized into rule-based heuristic algorithms and learning-based optimization algorithms. Specifically, rule-based heuristic algorithms typically make short video preloading decisions based on pre-programmed rules using client observations such as network conditions, buffer status, etc.

- *Biao Hou, Song Yang and Fan Li are with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China. E-mail: {houbiao, S.Yang, fli}@bit.edu.cn. (Corresponding author: Song Yang).*
- *Liehuang Zhu is with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China. Email: liehuangz@bit.edu.cn.*
- *Lei Jiao is with the Department of Computer Science, University of Oregon, Eugene, OR 97403, USA. Email: jiao@cs.uoregon.edu.*
- *Xu Chen is with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China. E-mail: chenxu35@mail.sysu.edu.cn.*
- *Xiaoming Fu is with Institute of Computer Science, University of Göttingen, 37077 Göttingen, Germany. Email: Fu@cs.uni-goettingen.de.*

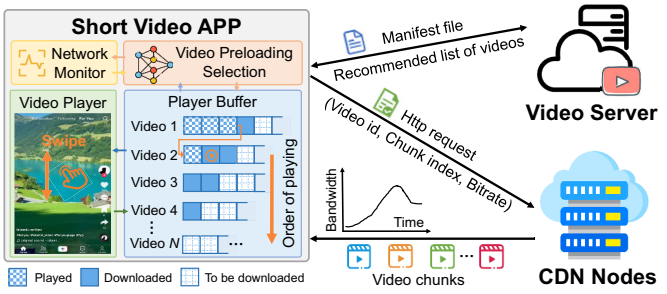


Fig. 1. A brief illustration of short video streaming system.

[14], [15], [16]. Although these algorithms are easy to implement, fixed rules hardly adapt to the complex fluctuations caused by heterogeneous networks and swipe behaviors, and inappropriate settings perhaps significantly impede their efficacies [17]. Learning-based algorithms apply deep Neural Networks (NNs) to directly connect environmental observations and actions to determine an optimal control policy for a given user-specific QoE objective [18], [19], [20]. Their performances heavily depend on training data, making it difficult to achieve fast convergence [21]. Moreover, once video preloading decisions are made, they cannot be rolled back. In this sense, one premature preloading mismatch decision may exacerbate the QoE of the entire short video streaming [22]. Unfortunately, off-the-shelf preloading algorithms usually fall short of considering users' swipe estimation according to video content and historical viewing behaviors. Consequently, there is a need for a tailor-made short video preloading algorithm to find an appropriate video preloading sequence that maximize QoE improvement by simultaneously reducing bandwidth wastage.

Driven by these practical concerns, we attempt to answer a pivot question: *Can short video streaming quickly adapt to frequent swipes and dynamic network conditions to propel the boundary of QoE?* To that end, we analyze the KuaiRand dataset [23] and conduct measurement experiments on short video streaming transmissions in real-world network traces. The empirical analysis elucidates that despite the drastic fluctuations in network bandwidth, there exists discernible short-term continuity in network sequences. This presents a unique opportunity to utilize learning-based methods to learn and adapt to dynamic environments and make sequential decisions in short video preloading scenarios. Inspired by this, we propose Gamora, a novel fine-grained video streaming system for short video applications that embraces joint optimization of QoE enhancement and bandwidth efficiency. In this paper, we first propose an unordered video preloading algorithm that utilizes a Deep Reinforcement Learning (DRL) algorithm [24] to automatically learn the correlations between network conditions, swipe behaviors, and the corresponding optimal potential video preloading decisions. Moreover, we present an Age-of-Information (AoI)-based [25] buffer management method, which can achieve high throughput and video timeliness of continuous viewing sequences by seamlessly compromising the online feedback of total buffer occupancy and buffer queue drain time [26]. Then, Gamora further applies the Asymmetric Imitation Learning (AIL) algorithm to guide the DRL-based preloading algorithm, which enables the trainee policy to learn enough knowledge to maximize expected rewards under repeated demonstration/supervision by experts [27]. Finally, we implement a holistic prototype of our proposed short video streaming system and conduct extensive experiments under different network conditions. We find that Gamora improves QoE by 28.7%-51.4% on average compared to existing algorithms, while reducing bandwidth wastage by 40.7%-83.2% without sacrificing video quality. To our knowledge, Gamora is a novel DRL-based buffer-aware system with AIL guidance for determining optimal unordered preloading sequences to satisfy users' QoE requirements and mitigate bandwidth wastage for short video sessions. In particular, this paper makes the following contributions:

- We design a buffer-aware short video streaming system by incorporating network prediction and swipe estimation to steer unordered video preloading algorithm to enhance the user's QoE during playback.
- We further propose an asymmetric imitation learning algorithm to guide DRL-based short video preloading decisions under experts' demonstration for fast convergence.
- We implement a holistic prototype of our proposed short video streaming system. Extensive results demonstrate that our proposed solution outperforms state-of-the-art algorithms in terms of overall QoE and bandwidth wastage.

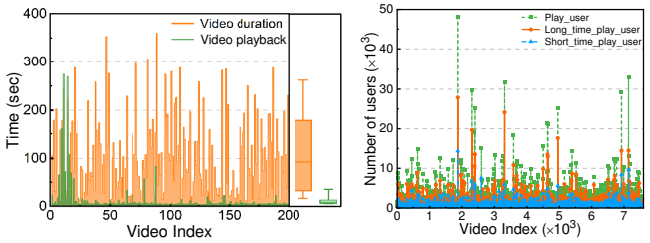
The remainder of this paper is organized as follows. In Section 2, we introduce the background and motivation of this paper. Section 3 elaborates on the system design. Section 4 presents the short video preloading algorithm. Section 5 demonstrates the prototype implementation. Section 6 evaluates and analyzes the performance of Gamora via extensive experiments. We provide the related works in Section 7 and conclude this paper in Section 8.

2 Background And Motivation

This section first presents a typical architecture of short video streaming. Then we analyze and perform a scaling experiment with real-world datasets to demonstrate the associated challenges of short video streaming.

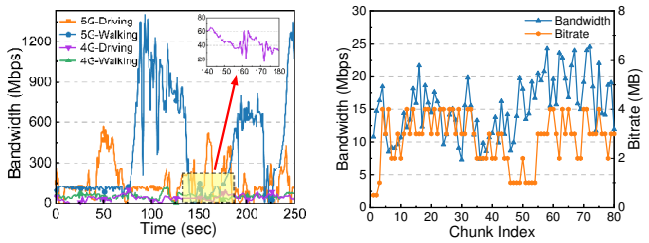
2.1 Short Video Streaming Primer

In short video applications, user swipes dictate the playing order of videos to cater for their needs. Fig. 1 shows a system overview for the short video streaming application. Upon receiving a client session request, the server inherently generates a list of Top-K short video recommendations [28] based on user characteristics and preferences derived from their previous access session records [29]. Subsequently, the server ships the client player with a manifest file containing the URL and relevant information. Like traditional video streaming, the client maintains a playback buffer and utilizes a preloading algorithm [30] to determine appropriate preload video sizes in the manifest file through Content Distribution Networks (CDN) nodes at any time to improve overall QoE for future short video sessions. Moreover, short videos are played sequentially within each logical buffer



(a) Duration and playback. (b) Distribution of users.

Fig. 2. User interaction statistics on the KuaiRand dataset [23].



(a) Dynamic bandwidth. (b) Bitrate adaptation.

Fig. 3. Bandwidth statistics across various network traces.

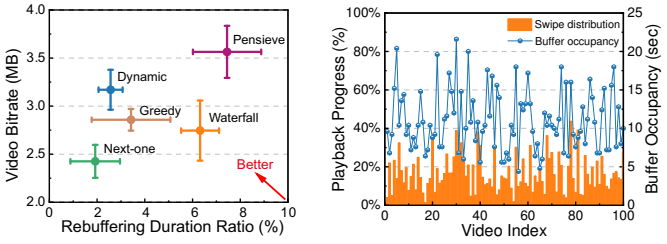
intra-queue but across inter-queues in a specific order. Playback is triggered at any time by user actions (e.g., swipe), or video completion, moving to the head of buffer to play the subsequent video. Once all the previous videos in the current queue are downloaded, the client player requests a new short video manifest file to continue playback. It facilitates swift and seamless adaptation of short videos to enhance viewer engagement [31], [32].

2.2 Short Video Session Characterization

We analyze short video streaming requests to investigate the distribution of video requests and uncover hidden characteristics in both the temporal and spatial dimensions. To that end, we utilize an unbiased sequential recommendation dataset called KuaiRand [23], which is collected from the Kuaishou application in one month. This dataset contains affluent features of items and users' behavioral history, encompassing millions of interactions on randomly exposed videos. We analyze a random sample of short video data from the dataset. Fig. 2a illustrates the internal relationship between video duration and user viewing time for short videos. The majority of video sessions have an average duration of less than 100 seconds. Moreover, nearly 93% of short videos are viewed for no more than 8 seconds, and the average video playback progress (defined as video playback duration divided by video duration) is 17%. This data explicitly suggests a long-tail effect in short videos, indicating that users lack the patience to watch videos in their entirety [29]. Fig. 2b provides further insights into user behavior by exploring users' proportion who engage in long and short video playback. Notably, approximately one-third of users fall into the category of short-time users with a playback duration of less than 3 seconds. Therefore, we can conclude that arbitrary user swipe is a pivot factor contributing to prohibitive bandwidth wastage. Since the user may swipe at any given moment, subsequently downloaded chunks in the current download buffer queue might remain unviewed.

2.3 Challenges of Short Video Streaming

We investigate the impacts of user swipes, network conditions, and buffer occupancy in depth for short video streaming. To study the user's QoE in a controlled and systematic manner, we develop a high-fidelity simulator that is capable of replaying the entire trajectory and chunk download strategy. It enables us to obtain detailed information about the video playback process swiftly. We leverage the `Mahimahi` tool [33] to control network conditions, which are collected from previous bandwidth measurements from commercial



(a) QoE impact. (b) Buffer occupancy.

Fig. 4. QoE performance of various short video preloading algorithms under lumos network conditions.

networks [34] [35]. Overall, we identify that these factors affect users' QoE. Specifically, short video streaming suffers from poor performance for the following reasons:

Network and video adaptation. To measure the impact of heterogeneous networks on the video chunk download strategy for short video streaming, we conduct a series of empirical experiments using the Lumos dataset [35]. As shown in Fig. 3a, different network bandwidth trajectories change frequently and fluctuate violently up to several hundred Mbps. Dramatically fluctuating networks face challenges for video preloading while delivering highly available throughput. Therein, short video streaming employs a vanilla waterfall strategy [20] for downloading video chunks sequentially from the top of the recommended queue. We describe the instantaneous network throughput and chunk adaptation decisions for short video streaming in Fig. 3b. We identify that the chunk decisions positively correlate with the network throughput [36].

Swipe estimation and wastage. Current short video streaming systems employ a fundamentally fixed-size static heuristic adaptation approach [14] to cope with dynamic network conditions. However, this approach often needs more adaptability to adjust optimally for different videos or users, leading to either extremely cautious or aggressive behavior. To address this thorny problem, we show the performance of five existing short-video streaming algorithms, as depicted in Fig. 4. Both the Next-one [37] and Waterfall strategies [38] select the next video once the current video has finished downloading. The primary distinction lies in the concurrent buffering capacity for short videos simultaneously. The Dynamic algorithm adopts a fixed buffer to download video sequentially, while the Greedy approach [1] dynamically selects the target bitrate based on the network bandwidth. Moreover, Pensieve [9] is a typical RL-based algorithm for long-form video streaming delivery. From Fig. 4a, we demonstrate that video bitrate and rebuffering ratio are two conflicting factors. Pensieve exhibits the

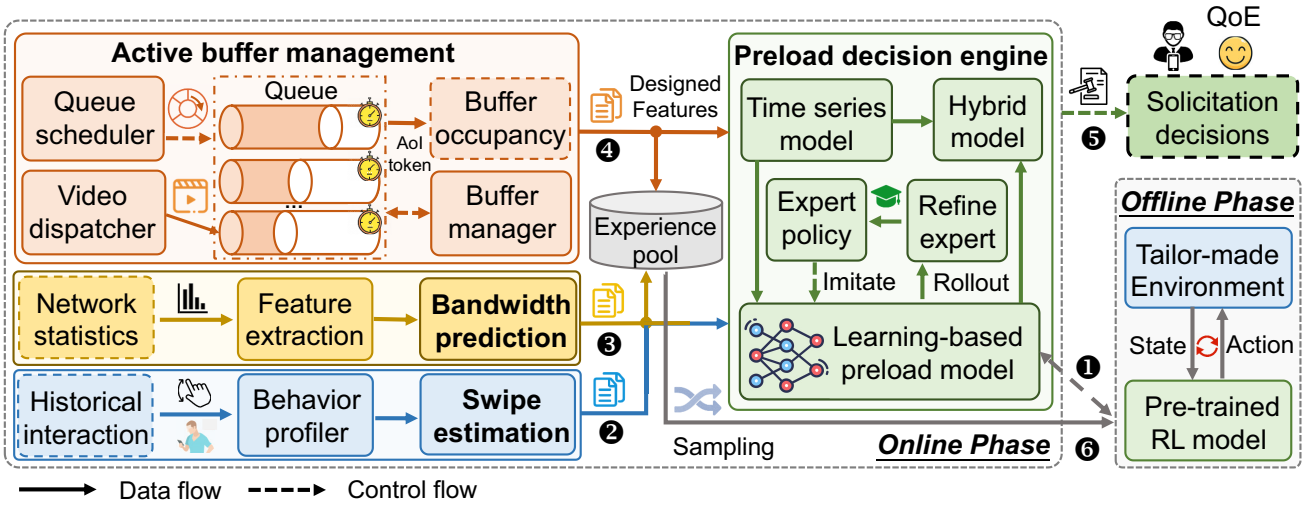


Fig. 5. A system overview of Gamora.

highest video quality, but its rebuffering ratio displays a significant upward trend with an average increase by 74.2% compared to Next-one. The coarse-grained approach leads to potential bitrate oscillations when there are changes in bandwidth during this interim period, attributable to the premature and blind binding of short video sequences.

Short video streaming systems often prioritize downloading the first chunk of content to address the uncertainty of user swipes to prevent rebuffering. However, especially when users swipe to the subsequent video, the remaining downloaded chunks of the preceding video evolve redundant, resulting in non-negligible wastage of bandwidth. As shown in Fig. 4b, the urgency of short video downloads leads to a significant impact on the average buffer occupancy, while the video playback process remains below 40%. As swipe bursts become more bursty, a meticulous active buffer management approach becomes essential to minimize bandwidth wastage without sacrificing video quality according to various situations.

3 System Design

This section first presents the system design of the short video streaming system called Gamora. Subsequently, we elaborate on core components, including bandwidth prediction and swipe estimation, AoI-based buffer management, and preload decision engine for short video streaming.

3.1 System Overview

Gamora's core concept is to use historical data from short video streaming to accurately predict future short-term bandwidth and analyze video swipe probability distributions to generate appropriate decisions for video preloading sequences. We utilize an efficient asymmetric imitation learning algorithm to optimize the short video preloading selection algorithm that decides the sequence of each short video to improve the overall QoE. Additionally, Gamora leverages online feedback from an accurate active buffer management method to orchestrate buffer queue download sequences in real-time and effectively reduce bandwidth wastage. Fig. 5 shows a system overview of Gamora's design. It consists of two primary phases, online and offline. In the online phase, it leverages the network bandwidth from the most recent session traces to train a time series model for generating network predictions. The extracted user swipe characteristics, network statistics, and buffer occupancy are comprehensively combined and fed into an online learning-based model that uses asymmetric imitation learning to guide chunk request decisions and accelerate the training process. The preloading decision results from a fusion of predictions from two models. We utilize these predictions to determine the ultimate video preloading decision and incorporate it into the video preloading algorithm. In the offline phase, a DRL model for short video preloading decisions is well-trained using historical trajectories.

Workflow. The implementation of short video preloading decision is as follows. ❶ In the offline phase, Gamora utilizes the previously collected trajectories to automatically train the RL model by interacting with specific environments in order to improve user QoE. ❷ Considering users' swipe features, we leverage previous research findings that suggest short videos with similar content often exhibit the same swipe pattern [3]. Consequently, we could obtain the coarse-to-fine knowledge of the swipe probability associated with various short videos. ❸ We then analyze short video sessions to exploit bandwidth trajectories and employ a canonical Long Short-Term Memory (LSTM)-based model [39] to learn the network characteristics for predicting bandwidth fluctuations. ❹ As for the buffer state, the buffer manager continually monitors the current player buffer occupancy. In handling short video streaming requests, the queue scheduler maintains concurrent buffer queues to synchronize the download rhythm of short videos. To tackle the buffer management challenges arising from user swipe, we introduce an Age of Information (AoI) token [25], representing the time elapsed since the playback of the current queue. Once the AoI token surpasses a predefined threshold, it is promptly triggered to release the already played buffer queue occupancy. ❺ Meanwhile, the preload decision engine employs asymmetric imitation learning to navigate the DRL model to promptly decide on the short

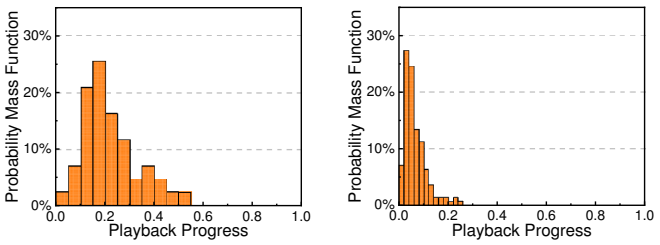


Fig. 6. Distribution of video viewing percentage for two sample videos.

video response (video ID, chunk index, chunk bitrate). The final preloading decision is adjusted according to buffer occupancy feedback. ❻ Subsequently, the pre-trained DRL model [40] undergoes periodic updates during the online phase to achieve fine-tuning and optimization.

3.2 Swipe and Bandwidth Prediction

Swipe distribution estimation. The overall QoE and bandwidth wastage in short video streaming closely correlate to user swipe. However, in existing short video streaming systems, swipe behavior is often deemed entirely random, although not in reality. To investigate this, we randomly select two highly-viewed short video types, including Kmovie and ShortCamera, from the KuaiRand dataset [23]. Fig. 6 exhibits a representative result of swipe probabilities in short videos. Approximately 80% of swipe events occur within the initial 20% of the short video's duration. Users are more inclined to swipe after video playback starts or automatically switch to another video after completing playback, consistent with previous studies on user swipe [11], [41]. Meanwhile, we observe the remarkable similarity in the distribution of swipe events in each video under diverse video datasets, suggesting that different users watching similar videos are likely to exhibit similar swiping behavior. With this information, we can gain insight into the likelihood of swiping for each short video and its potential location within a given video through machine learning methods during video playback. Therefore, if the current buffer length surpasses the user's anticipated swipe point, the short video system preemptively halts the current video request for new content (i.e., enters a sleep state) and proceeds to download the subsequent short video to conserve bandwidth consumption explicitly.

Each short video comprises several chunks with uniform duration. Let v represent a short video, and $\mathcal{V}$ denote the set of short videos. Each short video v is divided into N_c ($c \in \mathcal{C}$) chunks, where L signifies the video chunk length. Due to variations in user swipe patterns across different videos, we represent the probability $p_{v,c}$ of a user swiping after viewing the current chunk. The probability of user swipes can be inferred from the historical interaction sessions associated with videos of the same type. However, the user swipe is not limited to occurring at chunk boundaries, and they may swipe at any time during playback in reality. Behavior profiler extrapolates the swipe probability to all video chunks, relying on the empirical value obtained from existing video chunks. Suppose the cumulative buffering time of video v from the initial chunk to the chunk c is $N_c \times L$. Then, assuming the current playback progress is Γ, so the swipe probability $p_{v,c}$ until the chunk N_{c+1} of the next video v is calculated as:

$$p_{v,c}(N_{c+1}) = \left| \frac{N_c \times L - \Gamma}{N_c \times L} \right| \times p_{v,c}(N_c) \tag{1}$$

where $p_{v,c}(N_{c+1})$ represents the probability of subsequent video chunk. We use a random forest model, to build a predictive model for swipe behavior probability. The swipe prediction model is integrated into the short video streaming system. The system monitors real-time playback and user interaction data to continuously update swipe probability estimations. More precise decisions regarding the preloading of short video chunks can be achieved through a rough estimation of the user's swipe probability for a given short video. It is noteworthy that, in practical short video streaming applications, the real-time playback progress of short video can be readily tracked a priori [29].

Short-term bandwidth prediction. We learn the dynamics of the network state by extracting short video session characteristics from historical bandwidth traces to predict the instantaneous bandwidth of the next interval in the future. The performance metrics that characterize the network links during short video transmission are generally described by latency, packet loss and bandwidth. Therefore, we leverage the feature extraction module to extract these features from historical network statistics. Nevertheless, the intricate variations and uncertain fluctuations in network bandwidth pose a significant challenge to accurately predicting bandwidth. Hence, there is an urgent need to devise an appropriate bandwidth prediction model for short video streaming systems to deliver suitable video responses for diverse user requests during playback.

To this end, we utilize an online bandwidth prediction model, specifically the Autoregressive Integrated Moving Average (ARIMA)-based LSTM model [42], [43], known for delivering more robust predictions than simple linear models such as the harmonic mean method. Short-term bandwidth features are extracted using the ARIMA model to identify contextual information in the network trajectories, while the LSTM model captures long-term temporal dependencies in these trajectories to eliminate biases for bandwidth prediction. Specifically, during the transmission of short video streaming, we utilize the historical network state data of the previous T period to extract the corresponding network features. For each tuple of input, we consider a three-part network feature: $\mathcal{X}_t = (R_t^\sigma, B_t^\sigma, P_t^\sigma)$, where R_t^σ denotes the Round-trip time (RTT), B_t^σ is the bandwidth measurement, and P_t^σ is the network packet loss rate. These metrics are derived from the real-time internal state of TCP during the transmission of short videos. In the LSTM model, we incorporate 64 nodes in hidden layers and employ the Adam optimizer [44] for training. The final output Y_t is the instantaneous bandwidth prediction over a short-term period. Therefore, the ARIMA-based LSTM model enables rapid estimation of network variation trends for video preloading adaptation in the subsequent decision models.

3.3 AoI-based Buffer Management

A pragmatic approach to conserve bandwidth involves regulating the download rhythm of short videos when

the buffered content surpasses the maximum buffering threshold. Our proposed solution is an **A**oI-based **B**uffer **M**anagement (ABM) module that dynamically orchestrates short video buffering sequences and releases via online feedback to achieve smooth playback without sacrificing QoE during short video playback. Concretely, Gamora dynamically adjusts the number and quality of preloaded short video chunks for subsequent time intervals based on the current resource availability by alternatively constructing a video download queue in advance. The ABM algorithm is designed to accommodate incoming video playback requests, while timely suspension of the preceding buffer queue to conserve bandwidth resources. It then releases the buffer queue viewed by users to guarantee valid content service according to the AoI value. Furthermore, we incorporate considerations for total buffer occupancy and queue drain time to ensure that Gamora exhibits specific properties such as mutual isolation, swipe tolerance, and bounded bandwidth wastage.

Fine-grained buffer management. Our proposed ABM module endeavors to optimize buffer utilization through fine-grained management across the entire buffer space at the queue level. Specifically, ABM assigns AoI thresholds to each short video queue by considering the statistics of the entire buffer in the spatial dimension and the characteristics of each buffer downloading queue in the temporal dimension. The threshold dynamically adjusts based on variations in the total remaining buffers and alterations in network bandwidth. In case the download queue length approaches the designated threshold, ABM temporarily suspends the ongoing download queue to initiate the download of the subsequent video with inferior priority. Specifically, we assign the specific threshold $\Psi_p^v(t)$ to the buffer queues of short video v with priority p at time t according to Eq. (2), which primarily takes into account four dynamic factors: (1) the number of video buffer queues with priority N_p; (2) the drain time of the video occupied queue μ_p^v, i.e., the short video playback time; (3) the total buffer space $\mathbb{B}$ of client; and (4) the current occupancy queue space $\mathbb{Q}$. Therefore, the per-queue threshold $\Psi_v^p(t)$ is defined as:

$$\Psi_v^p(t) = \alpha_p \cdot \frac{1}{N_p} \cdot (\mathbb{B} - \mathbb{Q}) \cdot \frac{\mu_p^v}{b_t} \tag{2}$$

where parameter α_p defines the buffer queue available to each priority level of short video. N_p indicates the number of video queues with priority p and μ_p^v represents the normalized playback speed of the short video client. The variables b_t is the current bandwidth at time t and $\mathbb{B} - \mathbb{Q}$ denotes the remaining buffer occupancy, respectively. To ensure practicality, we use empirical values [23] and periodic measurements for threshold calculation in our evaluations.

To maximize buffer efficiency, we utilize a simple priority buffering queue sequence to determine the maximum buffer occupancy. Given the potential for users to swipe away the video at any moment, we regulate the maximum queue buffer size contingent on the likelihood of the video being viewed at the present playback progress. In effect, it is essential to prevent any priority buffer queue from monopolizing the buffer, leading to starvation of other priorities. ABM achieves this equilibrium by taking into account the number of download queues per priority during the calculation of each queue AoI threshold. To accomplish this, the calculation of queue thresholds involves considering the number of download queues for each priority level. Moreover, we set a maximum buffer size of B^{max} to prevent bandwidth wastage. The amount of buffer queue available for any priority p of short video v is given by:

$$B_p^v = \frac{B^{max} \cdot \alpha_p}{1 + \sum_{p \in \mathcal{P}} N_p \cdot \alpha_p} \tag{3}$$

where $\mathcal{P}$ represents the set of priorities for short videos within the buffer queue. ABM allocates buffer space in proportion to the drain time of each downloading queue, thereby improving buffer efficiency while mitigating the adverse effects of queuing delays. Intuitively, the bounded allocation strategy of ABM enables it to absorb small video bursts, while the stable exhaustion time property further enhances its swipe tolerance as the unoccupied buffer queue can promptly respond to incoming short video requests. In particular, if the buffering duration exceeds the buffer size, the client would sleep mode for τ seconds to halt the download of video chunks [41].

AoI-based buffer release. The buffer manager relies heavily on playback information from various short video queues for the buffer release process. As users may choose to review previous short videos at any point, it is essential to determine the appropriate time to release the buffer queue to guarantee the valid content service before the AoI expires. To that end, we use the AoI metric that measures the elapsed time between the completion of buffer release and video playback from the user's perspective. The AoI metric is predominantly influenced by inter-arrival time, thereby effectively capturing the short-term playback information of short video system. Specifically, the AoI value progressively increases over time and until the corresponding buffer queue is released from the client buffer space, indicating that short video information played from an earlier request becomes outdated and necessitates its release to free up buffer occupancy. Consequently, we utilize $Z_n^v(t)$ to indicate the AoI received by the video queue in a time slot t. If the AoI exceeds the threshold, it signals the need to release and update the buffer with new content, thereby preventing the playback of outdated short videos. The queue scheduler does not receive an AoI token about video playback completion during the time interval of video playback, so its AoI value is reset to 1. On the other hand, if the queue scheduler receives the associated information, the AoI value increases by 1. Consequently, $Z_n^v(t)$ is defined as:

$$Z_n^v(t+1) = \begin{cases} Z_n^v(t) + 1, & \text{if } Q_n^v = 1 \\ 1, & \text{if } Q_n^v = 0 \end{cases} \tag{4}$$

where Q_n^v indicates the completion of playback for short video v at time slot t. To guarantee the desired information freshness of each queue, we need to constrict the AoI value of playback queue are not larger than a certain threshold [25], subject to minimize the response delay of requests.

3.4 Preload Decision Engine

The cornerstone of Gamora is its preloading decision engine, which systematically employs a short video preloading algorithm (Sec. 4) to make rational decisions based on the

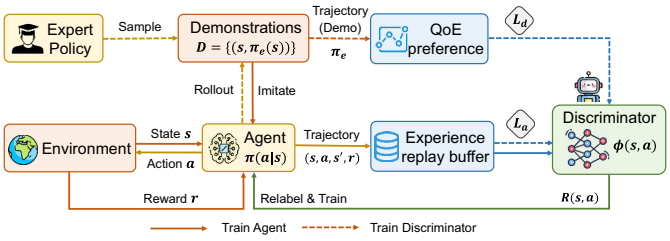


Fig. 7. The proposed asymmetric imitation learning algorithm.

previous information from the aforementioned components. Concretely, the preloading algorithm contains two phases: the online phase and the offline phase, respectively. During the offline phase, we pre-train the RL agent to interact with the environment, which utilizes them through massive explorations to derive the preloading model. Subsequently, Gamora's policy is further fine-tuned using AIL with preference as the IL framework until converging to a global optimum in the online phase. The video preload decision engine exploits an AIL algorithm to generate appropriate short video preloading decisions guided by mimicking the expert policy [45].

4 Short Video Preloading Algorithm

This section provides a comprehensive description of the short video preloading algorithm.

To expedite convergence and enhance learning efficiency, we adopt an AIL algorithm with preferences to refine the short video preloading model for maximizing overall QoE. Formally, an expert's policy or sampled trajectories are acquired to address the short video preloading task. The trainee learns and adjusts through direct imitation of the expert's actions, accessing demonstrations based on their observations. However, this algorithm may lead to poor performance when confronted with partial information not encompassed in the training set. A crucial challenge lies in ensuring that the expert is cognizant of the trainee's knowledge gaps [46], a determination that may be intricate. This limitation restricts the expert's capacity to provide suitable supervision. To tackle this concern, we employ an AIL algorithm, refining the agent's policy $\pi(a|s)$ by incorporating the trainee's imitation of the expert. As shown in Fig. 7, the AIL algorithm narrows the policy distance between the trainee's and policies to refine the expert's policy π_e by leveraging additional QoE preference feedback. Utilizing a pre-trained RL agent, we leverage expert demonstrations $\mathcal{D}$ and the agent's experience replay buffer to generate trajectories for updating the discriminator. Subsequently, a discriminator $\phi(s, a)$ is trained to differentiate between the generated samples, consisting of state-action pairs, and the expert demonstrations L_d and buffer L_e. The discriminator selects the superior sample, and this iterative process continues until convergence, signifying that the generated samples closely resemble those of the expert demonstration. Facilitated by the discriminator, the agent struggles to emulate the expert's behaviors by executing appropriate actions, such as selecting short video sequences. In particular, the AIL approach seamlessly integrates with the RL algorithm, eliminating the need for manual intervention.

Our adopted AIL with preferences mitigates state uncertainty by iteratively enhancing the policy through regression learning of expert behaviors, thereby improving the expected surrogate reward of the asymmetric policy. Specifically, preferences are acquired by randomly extracting trajectories from the demonstrations or an existing trajectory buffer. This updating process aims to reconstruct the optimal partially observed policy in conjunction with the AIL method. To derive the update of the expert policy π_θ, we analyze the cumulative RL objective $J(\theta)$ under the asymmetric policy π_θ^* as follows:

$$J(\theta) = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{d^{\pi_\theta^*}(s_t)\pi_\theta^*(a_t|s_t)} \left[Q^{\pi_\theta^*}(s_t, a_t) \right] \tag{5}$$

where $d^{\pi_\theta}(s_t)$ is the state occupancy, and Q^{π_θ} is defined as:

$$Q^{\pi_\theta}(s_t, a_t) = \mathbb{E}_{p(s_{t+1}|s_t, a_t)} \left[\mathcal{R}(s_t, a_t) + \gamma V^{\pi_\theta}(s_{t+1}, a_{t+1}) \right] \tag{6}$$

where the asymmetric value function is defined as $V^{\pi_\phi}(s_t) = \mathbb{E}_{\pi_\phi(a_t|s_t)} \left[Q^{\pi_\phi}(s_t, a_t) \right]$. This objective defines the cumulative discounted rewards that the trainee receives for various impacts on the expert's policy, given the current state-action pairs. The aim is to maximize $J(\theta)$ and ensure that the trainee receives guidance from the sampled expert demonstrations to update the RL parameters [47].

Our AIL algorithm comprises two layers of optimization to derive partially observed expert policy based on previous trajectories. Initially, we utilize preference feedback to refine the parameters of the expert's policy, aiming to maximize the expected reward of the implicit strategy given the current trainee policy. Subsequently, we use the samples from the replay buffer to calculate the demonstration loss and preference loss, optimizing them with respect to φ by projecting them onto the updated implicit policy defined by the refined expert. Consequently, we optimize a surrogate reward $J_\varphi(\theta)$ as:

$$\nabla_\theta J_\psi(\theta) = \mathbb{E}_{\pi_\theta^*(a_t|s_t) d^{\pi_\psi}(s_t)} \left[Q^{\pi_\psi}(s_t, a_t) \right] + H\left(\pi\left(\cdot \mid s_t \right) \right) \tag{7}$$

where $Q^{\pi_\psi}(s_t, a_t)$ represents the expected cumulative reward under the variational trainee policy, and $H\left(\pi\left(\cdot \mid s_t \right) \right)$ is the conditional entropy of policy π. The AIL algorithm gathers samples by rolling out under the mixture policy and then updates expert policy with an importance-weighted policy gradient to fit the trainee policy parameters.

Offline pre-training with RL. Gamora's agent aims to enhance the user-defined QoE while minimizing the amount of bandwidth wastage through a DRL-based algorithm in short video streaming. This involves interacting with the environment via a trial-and-error learning mechanism to achieve multi-objective optimization properties. Gamora incorporates two significant changes compared to standard RL-based algorithms [40]. Firstly, a bandwidth prediction sub-network is integrated into the policy network, enhancing the tight feedback loops. Secondly, historical short video requests are employed as state inputs alongside with dynamic reward functions. In this way, Gamora can associate short video requests with the optimal video preloading policy for maximizing cumulative rewards. Fig. 8 illustrates Gamora's neural network (NN) architecture. Further details about the NN, involving inputs, outputs, and model structure, are provided below.

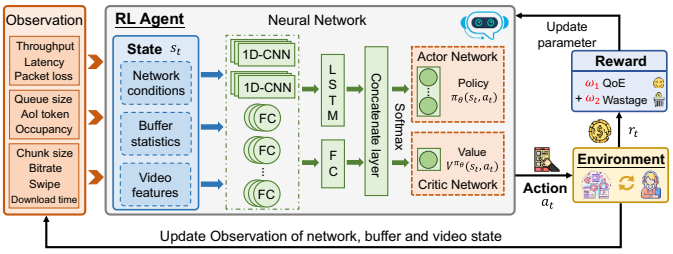


Fig. 8. Architecture of RL model with an actor-critic approach for estimating optimal preloading decision in a short video instance.

(1) *Input state*. We logically categorize the agent NN input into three individual portions, network conditions, buffer statistics, and video features, respectively. Concretely, we expose a finite subset of states ($s_t \in \mathcal{S}$) as the observations $\mathcal{O}$ from dynamic environment, where the agent c observes current state input $\mathcal{O}_t^c = \{Y_t^c, L_t^c, Pl_t^c, Bc_t^c, Qs_t^c, AoI_t^c, Cs_t^c, Dt_t^c, Vb_t^c, Sw_t^c\}$ at each time epoch t. These inputs are: predicted bandwidth Y_t^c, latency L_t^c, packet loss Pl_t^c, current playback buffer occupancy Bc_t^c, queue size Qs_t^c, AoI token AoI_t^c, chunk size Cs_t^c, download time Dt_t^c, chunk quality Vb_t^c, and swipe distribution Sw_t^c. At each time epoch t, the RL agent typically takes the current state s_t as input, the neural network as an optimal policy π_{θ_t}, and outputs a target video preloading decision a_t to interact with the customized short video system.

(2) *Action space*. Upon observing current state s_t, the RL agent performs an action ($a_t \in \mathcal{A}$) to maximize the expected cumulative reward ($r_t \in \mathcal{R}$) based on past state-action pairs. Specifically, the action space $\mathcal{A}$ is denoted as the available three decisions (i.e., 3-dimensional vector) for a given short video. Namely, the information required for download includes the video ID, chunk index, and bitrate. In each time t, the policy π_{θ_t} of agent maps observation $\mathcal{O}_t^c$ to compact discrete action space $\mathcal{A}$ and select an action a_t to enhance QoE performance.

(3) *Reward function*. After determining actions a_t, the RL agent interacts with the short video environment by adjusting video preloading decisions and then receives a cumulative discounted reward $r_t \in \mathcal{R}$ to update policy π_{θ_t}. The ultimate goal of RL model is to learn to select the optimal action that maximizes the QoE while minimizing the wastage. Therefore, we use a well-known reward function [9] that linearly combines four metrics: perceptual quality $\mathcal{Q}_i$, rebuffering duration $\mathcal{T}_i$, startup latency $\mathcal{L}_i$ and bandwidth wastage $\mathcal{W}_i$ as:

$$\begin{aligned} r_t &= \omega_1 \cdot QoE + \omega_2 \cdot Wastage \\ &= \omega_1 \cdot \left(\sum_{i=1}^{N_c} q(\mathcal{Q}_i) - \mu \sum_{i=1}^{N_c} \mathcal{T}_i - \delta \sum_{i=1}^{N_c} \mathcal{L}_i \right) - \omega_2 \cdot \sum_{i=1}^{N_c} \mathcal{W}_i \end{aligned} \tag{8}$$

where N_c represent the total number of chunks for each short video session. $q(Q_i)$ maps the bitrate Q_i to a utility value in Mbps, which signifies the quality perceived by the user. μ and δ are the coefficients. Referring to recent studies [17], [48], we empirically set these two weights to 1.85 and 0.5, respectively.

(4) *NN Architecture*. Gamora exploits the Proximal Policy Optimization (PPO) algorithm [21], a fundamental actor-critic framework for training NN policies to maximize the specific reward. In this paper, the architecture of the NN consists of two networks: the actor and the critic [24]. Before being inputted into the networks, state sequences are flattened. Each network structure utilizes two 1D-CNN layers and eight linear fully connected (FC) layers to extract temporal features. We use the Softmax activation function with the L2-norm as the last FC layer for both networks, resulting in optimal video preloading adaptation policy $\pi_\theta : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$. The actor model produces a $1 \times n$ dimensional vector representing video id, chunk index, and bitrate levels with their associated probabilities. The critic network generates a single scalar to indicate the value function for the current state. In addition, we instantiate multiple surrogate agents (default is 8) concurrently to expedite training using collected tuples in parallel.

Online fine-tuning with AIL. Gamora employs the AIL algorithm to fine-tune the NN parameters of short video preloading strategy, minimizing the number of trial-and-error iterations required for the agent to acquire the task, consequently expediting convergence. The AIL algorithm enables the agent to receive guidance from an expert through supervised learning by observing trajectories. The proficiency of the expert significantly influences the agent's performance within the AIL framework. To this end, we utilize PDAS [41], a rule-based hand-crafted algorithm, as the expert to guide the fine-tuning process. The selection of action combinations with superior QoE preferences serves as the basis for expert policies. The labels accurately reflect the real-time optimal preloading decisions available for video network paths. Furthermore, we continually enhance the rollout of expert's policies through iterative refinement, thus improving learning efficiency.

In each i-th round of imitation learning in Gamora, our goal is to use sampled expert trajectories to formulate a short video preloading policy π_{θ_i} that minimizes the imitation loss of the visited state induced by the previous round's policy $\pi_{\theta_{t-1}}$. The concrete policy π_{θ^*} is defined as:

$$\pi_i = \pi_{\theta^*} = \arg\min_{\theta \in \Pi_\Theta} \mathbb{E}_{d^{\pi_{i-1}}(s)}[\ell(\pi_\theta(s), \pi^*(s))] \tag{9}$$

where $\ell(\pi_\theta(s), \pi^*(s))$ represents the imitation loss for every state s. To cater to the distinctive features of short video streaming, Gamora incorporates two customized fine-tuning strategies: (1) an imitation loss mechanism that derives the update applied to the RL parameters through the expert policy, and (2) a dedicated method to smooth swipe transition between short video sequences, which we elaborate on below.

(1) *Imitation loss function*. The AIL process involves iteratively demonstrating pairs of state-action trajectories instead of feature-label pairs, which is similar to conventional approach for supervised learning. Therefore, we leverage the cross-entropy function for evaluation. For Gamora, the loss function of the AIL process is described as follows:

$$L_{AIL} = -\sum_{t=1}^{T} \hat{\mathcal{A}}_t^e \log \pi_\theta(s_t, a_t) \tag{10}$$

where $\pi_\theta(s_t, a_t)$ is the actor policy of the agent. The expert policy generates a list of action probability distributions denoted by $\hat{\mathcal{A}}_t^e$, where the value of the expert action is 1 while the others are 0. As for Gamora, even occasional burst

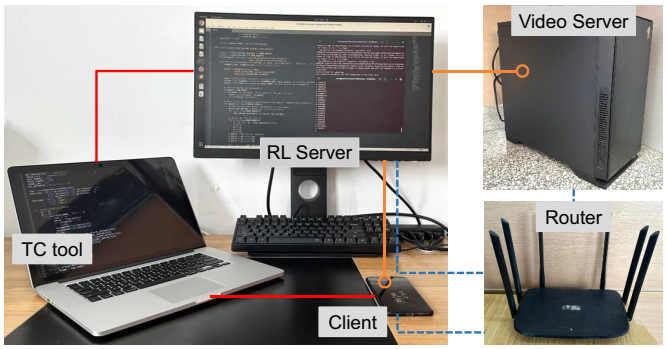


Fig. 9. Implementation of our short video streaming system prototype.

prediction deviations can negatively affect short video QoE. To mitigate this, we introduce a cross-entropy approach [48] to penalize such occurrences. The weights $\kappa(s)$ are defined as follows:

$$\kappa(s) = \begin{cases} \frac{1}{2}\|\pi_\theta(s) - \pi^*(s)\|^2, & \text{if } \pi_\theta(s) \leq \pi^*(s) \\ \xi\|\pi_\theta(s) - \pi^*(s)\| + \frac{1}{2}\xi^2, & \textit{otherwise} \end{cases} \tag{11}$$

where the parameter ξ generally indicates an auxiliary penalty for uncertainty over the expert state during training. Subsequently, the imitation loss is modified as $L_{AIL} \times \kappa(s)$.

(2) *Smoothing swipe switch*. During the fine-tuning phase, Gamora strategically integrates the swipe smoothness requirement as a regularization term into the AIL algorithm to mitigate rebuffering events and video bitrate switches. Swipe smoothness refers to the seamless video adaptation of the short video streaming to dynamic network conditions and user interactions, resulting in a more fluent viewing experience. Let's consider the expert policy π_θ, which yields optimal preloading decisions for the previous k time periods (up to t) as $[\pi_\theta(s_{t-k}), \pi_\theta(s_{t-k+1}), ..., \pi_\theta(s_{t-1})]$ using samples $\mathcal{D}$ from β proportion of the demonstrator's observation–action tuples. Gamora attempts to identify a saddle point where the predicted value at time interval t aligns closely with the historical decision-weighted average $\phi(t,k)$, denoted as:

$$\phi(t,k) = \sum_{i=1}^{k} 2^{-i} \times \pi_\theta(s_{t-i}) \tag{12}$$

where in $\phi(t,k)$, the closer trajectories are to historical expert demonstrations, the better the performance yield. We integrate the unbiased regularization term into the optimal policy derivation objective with a weight λ to estimate π_θ as follows:

$$\pi_{\theta^*} = \arg\min_{\theta \in \Pi_\Theta} \mathbb{E}_{s_t \sim d^{\pi_{i-1}}} [\ell(\pi_\theta(s_t), \pi^*(s_t)) + \lambda\|\pi_\theta(s_t) - \phi(t,k)\|] \tag{13}$$

The AIL algorithm iteratively updates the model parameters θ with the stochastic gradient descent method [49] to improve the convergence efficiency, which provides a swipe smoothness guarantee simultaneously.

5 Implementation

In this section, our implementation of Gamora comprises two primary components: real-world testbed and trace-driven simulator. We then present the corresponding network/video datasets and existing baselines.

TABLE 1. Gamora training/testing parameter settings.

Parameter	Notation	Value
Actor & Critic learning rates	lr	10^{-4}, 10^{-3}
PPO Clip parameter & steps	ϵ, t	0.2, 20
GAE parameter & AIL weight	σ, λ	0.95, 0.5
Discount factor	γ	0.99
Number of workers	N	16
Expert policy proportion	β	0.5

5.1 Experimental Setup

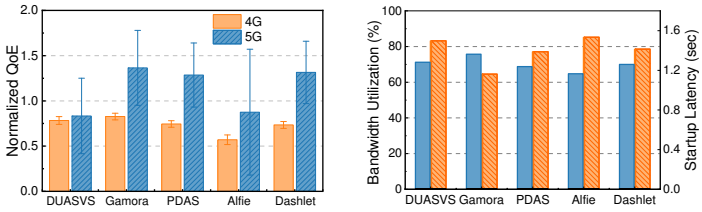
Testbed implementation. We establish an end-to-end short video streaming prototypical testbed and utilize public real-world network datasets to validate the performance of Gamora. As shown in Fig. 9, the system predominantly consists of three PCs and a mobile phone, interconnected via a network router. Specifically, the server runs Ubuntu 20.04 LTS with Intel Xeon Gold 6226 CPU @2.90GHz and Nvidia RTX 3090 GPU. The PCs act as the HTTP content server, running `WebRTC` [21] in the Nginx server to host short video contents and using Linux `TC` tool [10] to throttle network traffic. Gamora operates as a Python daemon. It integrates with an asynchronous RL server using `gRPC` to execute HTTP requests. When receiving HTTP requests to predict short video preloading workflow, the HTTP RPC is invoked for execution. Moreover, the client is a Dash.js-based media player [50] deployed on a rooted mobile phone (Android 12), which extends the Node.js configuration to facilitate short video playback.

Trace-driven simulator. We use TensorFlow 2.4.0 [51] and Ray 2.6.0 [52] to train Gamora's model architecture and construct the multiple parallel training workflows in the RL server. To hasten the model training process, we also utilize a trajectory-driven chunk-level simulator, derived from the existing simulator provided by Grand Challenges [53], to simulate a typical short video application. Furthermore, the simulator automatically generates the user's viewing behavior during video playback, taking the users' swipe patterns into account. It emulates the video player under various network conditions and faithfully records the crucial video chunk characteristics for efficient evaluation, such as the encapsulation of experience tuples and bandwidth overhead. The performance of Gamora significantly relies on the training parameter settings, so we empirically set the parameters as outlined in Table (1).

5.2 Methodology

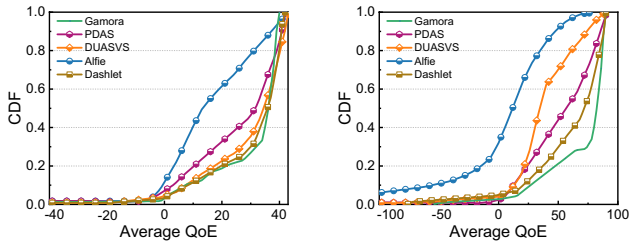
Network traces. We use diverse users' mobility network traces to simulate the bandwidth throttling between servers and clients. These traces are sourced from the real-world commercial network measurements [34] and Lumos network dataset [35], which we used for evaluation. We randomly select synthetic network traces from individual datasets, where the inter-variation duration maintains a consistent interval of 1 second between bandwidth values.

Video datasets. We utilize the H.264 codec in `FFmpeg` (version 4.3.6) to encode the public Big Buck Bunny (BBB) dataset [50] into various video segments, each lasting a fixed



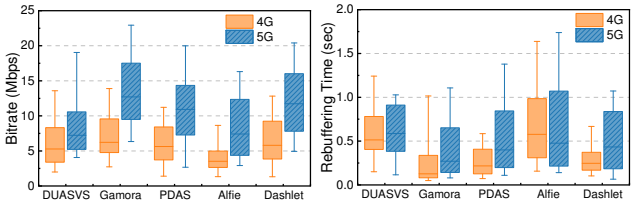
(a) QoE performance. (b) Bandwidth utilization.

Fig. 10. The performance of various algorithms in terms of QoE and bandwidth utilization across diverse network traces.



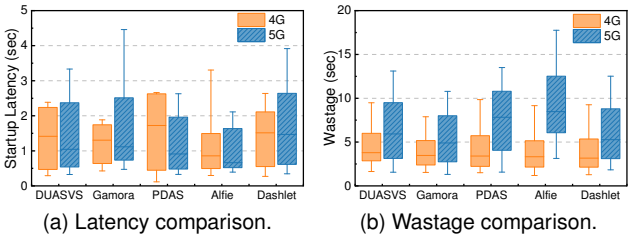
(a) Diverse algorithms for 4G trace. (b) Diverse algorithms for 5G trace.

Fig. 11. The CDF of average QoE by various algorithms under Lumos dataset.



(a) Bitrate comparison. (b) Rebuffering comparison.

Fig. 12. The performance of different preloading algorithms under various network conditions in term of bitrate and rebuffering.



(a) Latency comparison. (b) Wastage comparison.

Fig. 13. The performance of different preloading algorithms under various network condition in term of latency and bandwidth wastage.

number of seconds at 30 fps and available at five representative bitrates/resolutions (e.g., 1080P). For simplicity, we randomly selected a subset of the Kuairand dataset [23] to create a representative distribution of user swipe behaviors. Specifically, we split the video datasets, assigning 80% for training usage and the remaining 20% for testing.

Baseline algorithms. We compare four representative state-of-the-art baselines to evaluate the performance of our proposed Gamora solution. All algorithms are implemented in the same environment:

- Alfie [1]: It utilizes a vanilla DRL-based algorithm to optimize the user's QoE for short videos w.r.t. the current throughput-bound to improve the long-term streaming performance.
- DUASVS [15]: It develops an Integrated Learning algorithm to predict throughput using historical network conditions, and then dynamically determines short video bitrates and prefetch threshold to mitigate data loss during playback.
- PDAS [41]: It leverages a probability-driven buffer-based adaptive bitrate approach to balance the trade-off between QoE and bandwidth wastage in short video streaming.
- Dashlet [11]: It accounts for the complexity of user's swipe pattern and proposes a greedy algorithm to strategically determine the potential short video pre-buffering sequences and bitrate selection for maximizing overall QoE.

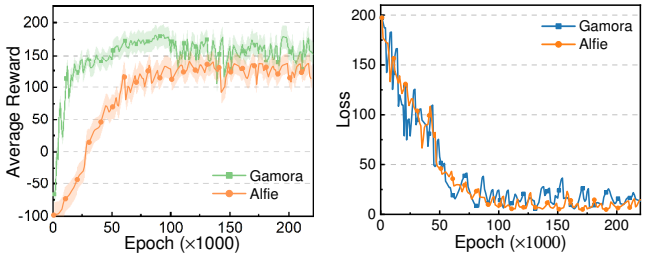
6 Performance Evaluation

This section evaluates the performance of Gamora by incrementally setting up different components under various network conditions and measures the influence of such configurations on the performance of short video streaming.

6.1 Comparison with Baseline Algorithms

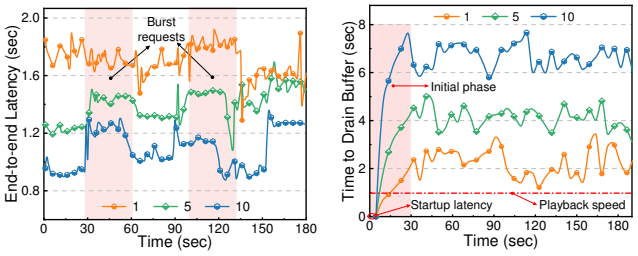
We conduct a comparative analysis between Gamora and existing baseline algorithms using the uniform Lumos network traces [35]. Fig. 10 illustrates the result of the average QoE value computed according to Eq. (8) and bandwidth utilization. Our findings indicate the following: (1) Gamora consistently outperforms the four comparing algorithms in terms of the QoE metric under all dynamic network conditions. On average, the QoE improvement between the closest competing solutions remains 8.3% and 44.7% for 4G and 5G networks, respectively. (2) The performance of the four benchmark algorithms is relatively similar. For example, PDAS exhibits a marginal improvement in 5G networks and worse in 4G networks. Moreover, Fig. 10b illustrates the bandwidth utilization and the average startup latency during short video playback. Our analysis reveals that Gamora achieves a throughput gain of 7.8%, 10.1%, 16.9%, and 3.1% compared to DUASVS, PDAS, Alfie, and Dashlet, respectively, while experiencing only a negligible increase in startup delay. Our investigation attributes Gamora's success to its ability to make accurate decisions regarding short video preloading adaptation and fine-grained buffer management during short video playback.

Fig. 11 illustrates the Cumulative Distribution Function (CDF) of average QoE for competitive baselines under heterogeneous network conditions. The CDF distribution curves, shown in Fig. 11a and Fig. 11b, provide evidence of the consistent performance of Gamora. Gamora also demonstrates a noteworthy improvement in average QoE, ranging from 28.7% to 51.4% when compared to other baseline algorithms. It is intriguing to note that Gamora exhibits varying performance levels in different QoE ranges. Under 5G network condition, 80% of the average QoE exceeds 50, whereas under 4G conditions, only about 80% of the average QoE value exceeds 20. For example, in Fig. 11a, in contrast to its competitors, Gamora exhibits superior performance in the high QoE range but significantly lags



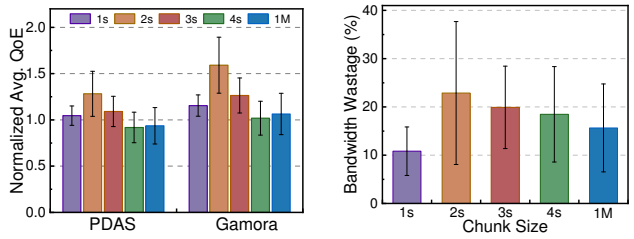
(a) Reward. (b) Loss.

Fig. 14. The training log of Gamora and Alfie.



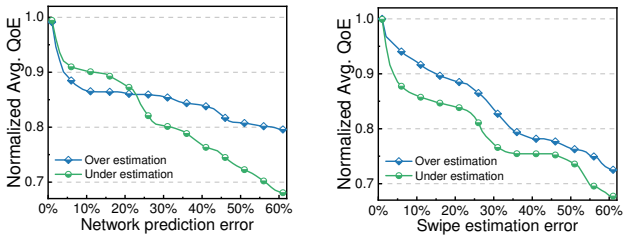
(a) Download time. (b) Bandwidth occupancy.

Fig. 16. Video performance of Gamora with different queue numbers.



(a) QoE performance. (b) Bandwidth wastage.

Fig. 15. QoE performance of Gamora with different chunk lengths.



(a) Network prediction error. (b) Swipe estimation error.

Fig. 17. Impact of network prediction and swipe estimation errors.

in the low to medium QoE range. This behavior is due to the multidimensional exploration of action space. With immediate guidance from expert demonstration, Gamora can effectively adapt to network bandwidth fluctuations and determine the appropriate short video preloading sequences for transmission during playback.

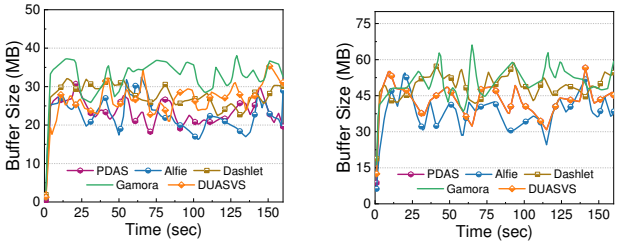
6.2 End-to-end QoE Improvement

Fig. 12 and Fig. 13 demonstrate the impact factors contributing to the overall QoE of short video streaming for Gamora vs. the baselines. Gamora consistently exhibits a prominent lead in both QoE metrics during video playback, containing bitrate, rebuffering, startup latency, and bandwidth wastage. Specifically, under the Lumos5G network conditions, Gamora outperforms the superior Dashlet by 6.1% in bandwidth wastage, because Gamora mitigates excessive buffering by strategically pausing video download sequences that surpass the buffering threshold and initiating the download of the subsequent video. Furthermore, Gamora achieves significant improvements by reducing the buffering time by 26.8%-73.5%, startup latency by 19.4%-29.7%, and 40.7%-83.2% bandwidth wastage while increasing video chunk bitrates by 54.1%-82.4% compared to other baselines. Gamora's superior performance is due to its precise and rapid video preloading approach to maximize the expected cumulative reward explicitly, thus mitigating the uncertainty associated with user swipes and network conditions during video playback. Moreover, relative metrics such as video bitrate exhibited superior performance under 5G conditions, aligning with real-world expectations. The high bandwidth and low latency of 5G contribute to an enhanced video viewing experience for mobile short video streaming. Nevertheless, the augmented bandwidth often leads to instant download completion of short videos, which tends to waste bandwidth because users may only watch a portion of the downloaded video before swiping away.

We demonstrate Gamora's prompt adaptation to new short video requests by average reward and loss, as indicated in Fig. 14. We compare Gamora's and Alfie's training logs on the Lumos5G dataset, aiming to evaluate the training efficiency of our asymmetric expert-guided algorithm and RL algorithm. Fig. 14a illustrates that Gamora consistently achieves higher rewards compared to Alfie within a low amount of interactions. Because Alfie initially adopts a suboptimal strategy, which requires a prolonged convergence period to reach the global policy. This challenge arises from the demand for an extensive exploration of future short video preloading sequences to unveil all possible state spaces that surrogate agents can generate. In contrast, Gamora efficiently achieves expert-level strategies in significantly lower 50k epochs through fine-tuning based on asymmetric imitation learning. Moreover, we also observe that Gamora converges at a higher pace than Alfie in Fig. 14b. Our proposed asymmetric imitation fine-tuning framework empowers Gamora to progress substantially with expert guidance during the RL fine-tuning phase.

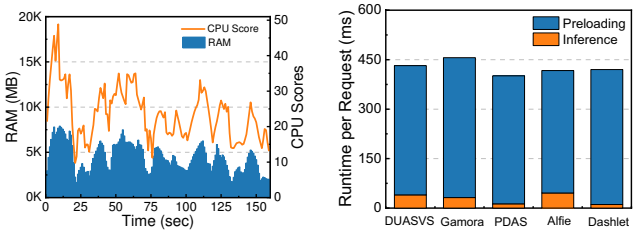
6.3 Understanding Gamora In-depth

Impact of chunk size. Similar to traditional long video streaming, Gamora divides the short videos into fixed-duration chunks along the temporal dimension. We explore the impact of dynamic video chunk lengths, such as 1s, 2s, 3s, and 4s, on Gamora's performance. We also Gamora's performance based on a fixed byte size (1M). Fig. 15 illustrates the average QoE and bandwidth wastage of video chunks for short video streaming. As depicted in Fig. 15a, the performance of Gamora exhibits a steady decline as the chunk size increases. For instance, when the video chunk length increases from 1s to 4s, the average QoE decreases by 35.9%. Fig. 15b reveals the relationship between chunk size and bandwidth wastage, with larger chunk size resulting in higher bandwidth wastage. The underlying reason is



(a) Diverse algorithms for 4G trace.(b) Diverse algorithms for 5G trace.

Fig. 18. The performance of different short video streaming algorithms under various network condition in term of buffer size.



(a) GPU utilization and memory usage.

(b) Runtime breakdown.

Fig. 19. Overhead of Gamora system during short video playback.

that substantial chunks of the download remain unviewed than requested whenever a user performs a swipe action by the client, leading to a higher volume of wasted bytes. Consequently, we advocate Gomora splits video into 2s chunks by default to enhance video QoE while reducing the bandwidth volume during playback.

Impact of queue number. Gamora downloads short videos by maintaining concurrent buffer queues. We conduct a quantitative study under the Lumos5G conditions to validate the impact of buffer queues. Fig. 16 demonstrates the buffer queue utilized by Gamora in regard to end-to-end latency and buffer occupancy. In Fig. 16a, we observe that an increase in the number of candidate videos for download leads to longer download times. However, this problem is mitigated by efficiently increasing the number of download queues. For instance, maintaining ten queues results in a 60.3% reduction in average latency compared to only having one queue. Simultaneously, employing multiple buffer queues helps alleviate spikes in short video requests, thereby reducing end-to-end latency. Furthermore, there is a rapid increase in buffer occupancy with the growing number of download queues, sustaining prolonged buffer consumption times, as illustrated in Fig. 16b. On average, Gamora experiences a buffer occupancy increase of approximately 7 seconds, enabling it to adeptly handle arbitrary swipes without incurring short video stall events.

Impact of prediction error. We conduct further analysis to evaluate the QoE influence of users' swipe distribution and network prediction errors, as depicted in Fig. 17. Specifically, we run an Oracle baseline to serve as a benchmark for normalized average QoE. The oracle is the Gamora with a priori knowledge of user swipe behaviors and network throughput traces, free from any prediction errors. Fig. 17a presents the outcomes under the influence of network prediction errors. We find that the average QoE declines rapidly as the prediction error increases. Gamora's QoE diminishes to 81% and 72% of its value when the network estimate is overestimated or underestimated by 50%, respectively. Because underestimating throughput consistently prompts the selection of lower-quality short videos or even the suspension of the download. Fig. 17b illustrates the results regarding the swipe estimation errors. Herein, overestimation denotes the case where users watch longer than the correct distribution, i.e., swipe short video later. Gamora exhibits significant resilience to prediction errors, maintaining a prediction error margin of no more than 20% compared to the benchmark (no error) while achieving a normalized QoE value of 80% or more.

Impact of buffer management. To thoroughly evaluate the available buffer occupancy, we investigate the Gamora performance under various network conditions. As shown in Fig. 18, we demonstrate the buffer occupancy size of the short video preloading algorithm in different network environments. We observe that as the short video playback starts, all algorithms download the video chunks quickly and maintain the current buffer within an appropriate range to avoid video lag and unnecessary data wastage during video sessions. Gamora outperforms the baselines, maintaining an average buffer size of 32.3MB. Additionally, the 5G network provides higher bandwidth compared to the 4G network, resulting in a higher available buffer for the Gamora algorithm under 5G conditions. It is worth noting that even at high bandwidth, the buffer occupancy does not continuously increase but stabilizes at a relatively high level.

System overhead. Fig. 19 illustrates the system overhead of Gamora during short video playback. Specifically, Fig. 19a demonstrates the CPU and memory utilization of the Gamora algorithm inference throughout the entire experiment. The average CPU and memory utilization rates of Gamora are 23.57% and 4.13GB, respectively. Considering the computational capabilities of the client (16 GB memory), the overhead introduced by the Gamora algorithm is within an acceptable range. We further compare the runtime of Gamora with other benchmark algorithms during short video playback. Fig. 19b illustrates the breakdown of runtime. Notably, the preloading time significantly exceeds the model inference time. Heuristic-based algorithms, Dashlet and PDAS, have shorter inference time compared to DRL-based algorithms. Among the DRL-driven algorithms, our proposed Gamora algorithm has a much shorter inference time than DUASVS and Alfie. This is primarily due to the low computational complexity of Gamora, achieved through AIL fine-tuning. The results indicate that while there is some overhead, it remains within acceptable limits for most practical video applications.

7 Related Work

This section briefly summarizes relevant literatures on optimizing short video streaming and imitation learning.

Short video streaming. Recent advancements in short video streaming show great potential for enhancing the video QoE by utilizing adaptive video algorithms. In general, it falls into two categories: (1) Heuristic-based algorithms [14], [15], [16] and (2) learning-based algorithms [18],

[19], [20]. Specifically, heuristic-based algorithms typically use estimated bandwidth [7], buffer status [8], or a combination [5], [37] to optimize the QoE for users. APL [14] utilizes Lyapunov optimization theory to provide near-optimal online decisions for the short video preloading problem. Dashlet [11] accounts for the complexity of user's swipe pattern and proposes a greedy algorithm to determine the potential short video prefetching sequences for maximizing overall QoE. However, such pre-defined generic rules are difficult to adapt to diverse heterogeneous networks and complicated interactive applications. Besides, learning-based algorithms apply DRL techniques [9], [18], [21], [50] to make appropriate decisions with observed features from the video streaming environment. DAM [19] incorporates user interactions and action masking into the RL agent for predicting the next requested short video segment. LiveClip [20] presents a Markov model to design a practical streaming policy for mobile short-form videos. Huang *et al.* propose DeepBuffer, a DRL-based algorithm that simultaneously determines appropriate bitrate ladders and controls buffer size to enhance the user's QoE during video streaming [54]. Nevertheless, these algorithms depend on extensive training data, which impedes rapid adaptation to dynamic environments. There are several studies to handle QoE optimization problem from diverse perspectives, including multi-path delivery [30] and short video recommendations [4], [28]. But all these works maintain a common prerequisite: the sequence of video playback matches the sequence of video downloads. In contrast, Gamora not only considers swipe complexity, but also overcomes buffer management under dynamic network conditions to effectively enhance QoE while mitigating bandwidth wastage.

Imitation learning. Imitation learning [27] has been applied in various research domains [18], [46], [55], given its robust learning capacity, such as task allocation and video streaming, etc. Wang *et al.* [46] devise an IL-driven online task scheduling algorithm in the vehicular edge computing environment. Comyco [17] stands out as the pioneering effort to integrate the IL algorithm into bitrate decision tasks, contributing to the efficient exploration of optimal streaming policies for on-demand video. Zhou *et al.* [48] propose an IL-enabled online algorithm to leverage the characteristics of codec and transport layers for optimal bitrate selection in video telephony. However, existing methods have a serious limitation [45]: the trainee usually concentrates on a single expert policy interaction whilst neglecting crucial feedback from training data, which could lead to the possibility of making sub-optimal choices under partially observed information. Therefore, we leverage a well-designed AIL-enabled approach to orchestrate the rollout of efficient expert demonstrations for video preloading optimization problem to accelerate learning convergence in the short video system environment.

8 Conclusion

In this paper, we have presented Gamora, a buffer-aware learning-based short video streaming system, to enhance the user's QoE. Gamora strategically determines and enforces appropriate short video preloading sequences through fine-grained network prediction, swipe estimation, and AoI-based buffer management online feedback techniques. Moreover, we further devise an asymmetric imitation learning algorithm to efficiently guide agents via expert demonstrations to explore and adapt to optimal policy to generate appropriate video preloading sequences. We develop a holistic prototype for short video streaming system to assess the effectiveness of Gamora. Extensive experimental results have demonstrated that Gamora significantly enhances video quality and mitigates bandwidth wastage compared to state-of-the-art baselines.

Acknowledgments

The work of Song Yang is partially supported by the National Natural Science Foundation of China (NSFC, No. 62172038, No. 62472028), in part by Beijing Natural Science Foundation (No. 4232033) and the National Key R&D Program of China (No. 2023YFB3107300). The work of Fan Li is supported by the NSFC (No. 62372045). The work of Liehuang Zhu is partially supported by the Yunnan Provincial Major Science and Technology Special Plan Projects (No. 202302AD080003). The work of Lei Jiao is supported in part by the U.S. National Science Foundation (CNS-2047719 and CNS-2225949). The work of Xu Chen is partially supported by Key Areas R&D Program of Guangdong Province (No. 2024B0101020004), and Guangdong Basic and Applied Basic Research Foundation (No. 2023B1515120058). The work of Xiaoming Fu is partially supported by Horizon Europe CODECO project (Grant No. 101092696) and Horizon Europe COVER project (No. 101086228). Song Yang is the corresponding author.

References

[1] J. Li, H. Xu, M. Ma, H. Yan, and C. J. Xue, "Alfie: Neural-reinforced adaptive prefetching for short videos," in *Proc. IEEE Int. Conf. Multimedia and Expo (ICME)*, 2022, pp. 1–6.

[2] B. of Apps, "Tiktok revenue and usage statistics 2023," 2023. [Online]. Available: https://www.businessofapps.com/data/tik-tok-statistics/

[3] Y. Zhang, Y. Liu, L. Guo, and J. Y. B. Lee, "Measurement of a large-scale short-video service over mobile and wireless networks," *IEEE Trans. Mobile Comput.*, vol. 22, no. 6, pp. 3472–3488, 2023.

[4] D. Ran, Y. Zhang, W. Zhang, and K. Bian, "SSR: Joint optimization of recommendation and adaptive bitrate streaming for short-form video feed," in *Proc. IEEE MSN*, 2020, pp. 418–426.

[5] Y. Yuan, W. Wang, Y. Wang, S. S. Adhatarao, B. Ren, K. Zheng, and X. Fu, "Joint optimization of QoE and fairness for adaptive video streaming in heterogeneous mobile environments," *IEEE/ACM Trans. Netw.*, vol. 32, no. 1, pp. 50–64, 2024.

[6] A. ParandehGheibi, M. Médard, A. Ozdaglar, and S. Shakkottai, "Avoiding interruptions—a QoE reliability function for streaming media applications," *IEEE J. Sel. Areas Comm.*, vol. 29, pp. 1064–1074, 2011.

[7] Z. Akhtar, Y. S. Nam, R. Govindan, S. Rao, J. Chen, E. Katz-Bassett, B. Ribeiro, J. Zhan, and H. Zhang, "Oboe: Auto-tuning video ABR algorithms to network conditions," in *Proc. Conf. ACM Special Interest Group Data Commun.*, 2018, pp. 44–58.

[8] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," in *Proc. ACM SIGCOMM*, vol. 44, 2014, pp. 187–198.

[9] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *ACM SIGCOMM*, 2017, pp. 197–210.

[10] M. Palmer, M. Appel, K. Spiteri, B. Chandrasekaran, A. Feldmann, and R. K. Sitaraman, "VOXEL: Cross-layer optimization for video streaming with imperfect transmission," in *Proc. ACM CoNEXT*, New York, NY, USA, 2021, pp. 359–374.

[11] Z. Li, Y. Xie, R. Netravali, and K. Jamieson, "Dashlet: Taming swipe uncertainty for robust short video streaming," in *Proc. USENIX NSDI*, 2023, pp. 1583–1599.

[12] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," *IEEE/ACM Trans. Netw.*, vol. 28, no. 4, pp. 1698–1711, 2020.

[13] A. Ahmed, Z. Shafiq, H. Bedi, and A. Khakpour, "Suffering from buffering? detecting QoE impairments in live video streams," in *Proc. IEEE Int. Conf. Netw. Protocols (ICNP)*, 2017, pp. 1–10.

[14] H. Zhang, Y. Ban, X. Zhang, Z. Guo, Z. Xu, S. Meng, J. Li, and Y. Wang, "APL: Adaptive preloading of short video with lyapunov optimization," in *Proc. IEEE VCIP*, 2020, pp. 13–16.

[15] G. Zhang, J. Zhang, K. Liu, J. Guo, J. Y. Lee, H. Hu, and V. Aggarwal, "DAUSVS: A mobile data saving strategy in short-form video streaming," *IEEE Trans. Serv. Comput.*, vol. 16, no. 2, pp. 1066–1078, 2022.

[16] C. Qiao, J. Wang, and Y. Liu, "Beyond QoE: Diversity adaptation in video streaming at the edge," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 289–302, 2020.

[17] T. Huang, C. Zhou, X. Yao, R.-X. Zhang, C. Wu, B. Yu, and L. Sun, "Quality-aware neural adaptive video streaming with lifelong imitation learning," *IEEE J. Sel. Areas Comm.*, vol. 38, pp. 2324–2342, 2020.

[18] Y. Li, Q. Zheng, Z. Zhang, H. Chen, and Z. Ma, "Improving ABR performance for short video streaming using multi-agent reinforcement learning with expert guidance," in *Proc. ACM NOSSDAV*, 2023, pp. 58–64.

[19] S.-Z. Qian, Y. Xie, Z. Pan, Y. Zhang, and T. Lin, "DAM: Deep reinforcement learning based preload algorithm with action masking for short video streaming," in *Proc. ACM Multimedia*, 2022.

[20] J. He, M. Hu, Y. Zhou, and D. Wu, "LiveClip: towards intelligent mobile short-form video streaming with deep reinforcement learning," in *Proc. ACM NOSSDAV*, 2020, pp. 54–59.

[21] X. Xiao, M. Yan, Y. Zuo, B. Liu, P. Ruan, Y. Cao, and W. Wang, "From ember to blaze: Swift interactive video adaptation via meta-reinforcement learning," in *Proc. IEEE INFOCOM*, 2023, pp. 1–10.

[22] X. Zuo, J. Yang, M. Wang, and Y. Cui, "Adaptive bitrate with user-level QoE preference for video streaming," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2022, pp. 1279–1288.

[23] C. Gao, S. Li, Y. Zhang, J. Chen, B. Li, W. Lei, P. Jiang, and X. He, "KuaiRand: An unbiased sequential recommendation dataset with randomly exposed videos," in *ACM CIKM*, 2022, pp. 3953–3957.

[24] H. Mossalam, Y. M. Assael, D. M. Roijers, and S. Whiteson, "Multi-objective deep reinforcement learning," *arXiv:1610.02707*, 2016.

[25] B. Li, "Efficient learning-based scheduling for information freshness in wireless networks," in *IEEE INFOCOM*, 2021, pp. 1–10.

[26] V. Addanki, M. Apostolaki, M. Ghobadi, S. Schmid, and L. Vanbever, "ABM: Active buffer management in datacenters," in *Proc. ACM SIGCOMM*, 2022, pp. 36–52.

[27] A. Li, B. Boots, and C.-A. Cheng, "MAHALO: Unifying offline reinforcement learning and imitation learning from observations," in *International Conference on Machine Learning*, 2023.

[28] Q. Cai, Z. Xue, C. Zhang, W. Xue, S. Liu, R. Zhan, X. Wang, T. Zuo, W. Xie, D. Zheng *et al.*, "Two-stage constrained actor-critic for short video recommendation," in *ACM WWW*, 2023, pp. 865–875.

[29] Z. Chen, Q. He, Z. Mao, H.-M. Chung, and S. Maharjan, "A study on the characteristics of douyin short videos and implications for edge caching," in *Proc. ACM TURC-China*, 2019, pp. 1–6.

[30] D. Wei, J. Zhang, H. Li, Z. Xue, Y. Peng, and R. Han, "Multipath smart preloading algorithms in short video peer-to-peer CDN transmission architecture," *IEEE Network*, vol. 38, no. 3, pp. 285–291, 2024.

[31] Y. Cheng, H. Zhang, and J. Jiang, "Online profiling and adaptation of quality sensitivity for internet video," in *Proceedings of the ACM Symposium on Cloud Computing (SoCC)*, 2023, pp. 520–527.

[32] S. S. Krishnan and R. K. Sitaraman, "Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs," *IEEE/ACM Trans. Netw.*, vol. 21, no. 6, pp. 2001–2014, 2013.

[33] R. Netravali, A. Sivaraman, S. Das, A. Goyal, K. Winstein, J. Mickens, and H. Balakrishnan, "Mahimahi: accurate Record-and-Replay for HTTP," in *Proc. USENIX ATC*, 2015, pp. 417–429.

[34] X. Yang, X. Wang, Z. Li, Y. Liu, F. Qian, L. Gong, R. Miao, and T. Xu, "Fast and light bandwidth testing for internet users," in *Proc. USENIX NSDI*, 2021, pp. 1011–1026.

[35] A. Narayanan, E. Ramadan, R. Mehta, X. Hu, Q. Liu, R. A. Fezeu, U. K. Dayalan, S. Verma, P. Ji, T. Li *et al.*, "Lumos5G: Mapping and predicting commercial mmWave 5G throughput," in *Proc. ACM IMC*, 2020, pp. 176–193.

[36] T. Huang, C. Zhou, R.-X. Zhang, C. Wu, and L. Sun, "Learning tailored adaptive bitrate algorithms to heterogeneous network conditions: A domain-specific priors and meta-reinforcement learning approach," *IEEE J. Sel. Areas Comm.*, vol. 40, pp. 2485–2503, 2022.

[37] Y. Qin, S. Hao, K. R. Pattipati, F. Qian, S. Sen, B. Wang, and C. Yue, "Quality-aware strategies for optimizing ABR video streaming QoE and reducing data usage," in *Proc. ACM MMSys*, 2019.

[38] D. Nguyen, P. Nguyen, V. Long, T. T. Huong, and P. N. Nam, "Network-aware prefetching method for short-form video streaming," in *Proc. IEEE MMSP*, 2022, pp. 1–5.

[39] S. Huang, H. Zhang, X. Wang, M. Chen, J. Li, and V. C. Leung, "Fine-grained spatio-temporal distribution prediction of mobile content delivery in 5G ultra-dense networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 1, pp. 469–482, 2022.

[40] R. S. Sutton, A. G. Barto *et al.*, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 135.

[41] C. Zhou, Y. Ban, Y. Zhao, L. Guo, and B. Yu, "PDAS: Probability-driven adaptive streaming for short video," in *Proc. 30th ACM Int. Conf. Multimedia (MM)*, 2022, pp. 7021–7025.

[42] K. Winstein, A. Sivaraman, and H. Balakrishnan, "Stochastic forecasts achieve high throughput and low delay over cellular networks," in *Proc. USENIX NSDI*, 2013, pp. 459–471.

[43] K. Chen, B. Wang, W. Wang, X. Li, and F. Ren, "DeeProphet: Improving http adaptive streaming for low latency live video by meticulous bandwidth prediction," in *Proc. ACM Web Conference (WWW)*, 2023, pp. 2991–3001.

[44] D. P. Kingma and J. Ba, "Adam: A method for chastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[45] A. Warrington, J. W. Lavington, A. Scibior, M. Schmidt, and F. Wood, "Robust asymmetric learning in POMDPs," in *Proc. Int. Conf. Machine Learning (ICML)*, 2021, pp. 11 013–11 023.

[46] X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing," *IEEE Trans. Mobile Comput.*, vol. 21, no. 2, pp. 598–611, 2022.

[47] S. Abbasloo, C.-Y. Yen, and H. J. Chao, "Classic meets modern: A pragmatic learning-based congestion control for the internet," in *Proc. ACM SIGCOMM*, 2020, pp. 632–647.

[48] A. Zhou, H. Zhang, G. Su, L. Wu, R. Ma, Z. Meng, X. Zhang, X. Xie, H. Ma, and X. Chen, "Learning to coordinate video codec with transport protocol for mobile video telephony," in *Proc. ACM MobiCom*, 2019, pp. 1–16.

[49] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," *Advances in neural information processing systems (NeurIPS)*, vol. 26, pp. 315–323, 2013.

[50] F. Y. Yan, H. Ayers, C. Zhu, S. Fouladi, J. Hong, K. Zhang, P. Levis, and K. Winstein, "Learning in situ: a randomized experiment in video streaming," in *Proc. USENIX NSDI*, 2020, pp. 495–511.

[51] "TensorFlow," 2022. [Online]. Available: https://www.tensorflow.org

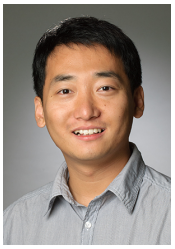
[52] "Ray," 2023. [Online]. Available: https://github.com/ray-project/ray/releases/tag/ray-2.6.0

[53] "ACM Multimedia 2022 Grand Challenge," 2022. [Online]. Available: https://github.com/AItransCompetition/Short-Video-Streaming-Challenge

[54] T. Huang, C. Zhou, R.-X. Zhang, C. Wu, and L. Sun, "Buffer awareness neural adaptive video streaming for avoiding extra buffer consumption," in *IEEE INFOCOM*, 2023, pp. 1–10.

[55] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Comput. Surv.*, vol. 50, no. 2, pp. 1–35, 2017.

Biao Hou received the B.S. degree in computer science and the M.S. degree in computer science from the Inner Mongolia University, Hohhot, China, in 2018 and 2021, respectively. He is currently the Ph.D. student with the School of Computer Science and Technology, Beijing Institute of Technology. His research interests include edge computing, serverless computing and video streaming.



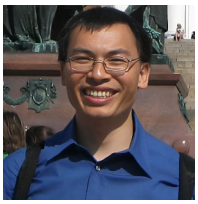
Song Yang (Senior Member, IEEE) received the BS degree in software engineering and MS degree in computer science from the Dalian University of Technology, Dalian, Liaoning, China, in 2008 and 2010, respectively, and the PhD degree from the Delft University of Technology, The Netherlands, in 2015. From August 2015 to July 2017, he worked as postdoc researcher with the EU FP7 Marie Curie Actions CleanSky Project in Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG), Göttingen, Germany. He is currently an associate professor with the School of Computer Science and Technology in Beijing Institute of Technology, China. His research interests include data communication networks, cloud/edge computing, and network function virtualization.

Fan Li (Member, IEEE) received the BEng and MEng degrees in communications and information system from the Huazhong University of Science and Technology, Wuhan, China, in 1998 and 2001, respectively, the MEng degree in electrical engineering from the University of Delaware, Newark, Delaware, in 2004, and the PhD degree in computer science from the University of North Carolina at Charlotte, Charlotte, North Carolina, in 2008. She is currently a professor with the School of Computer Science and Technology, Beijing Institute of Technology, China. Her current research focuses on wireless networks, ad hoc and sensor networks, and mobile computing. Her papers won best paper awards from IEEE MASS (2013), IEEE IPCCC (2013), ACM MobiHoc (2014), and Tsinghua Science and Technology (2015). She is a member of the ACM.

Liehuang Zhu (Senior Member, IEEE) received the BE and ME degrees from Wuhan University, Wuhan, China, in 1998 and 2001, respectively, and the PhD degree in computer science from the Beijing Institute of Technology, Beijing, China, in 2004. He is currently a professor with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China. He has published more than 150 peer-reviewed journal or conference papers. He has been granted a number of IEEE best paper awards, including IWQoS 17', TrustCom 18', and ICA3PP 20'. His research interests include security protocol analysis and design, blockchain, wireless sensor networks, and cloud computing.

Lei Jiao (Member, IEEE) received the Ph.D. degree in computer science from the University of Göttingen, Germany. He is currently an assistant professor at the Department of Computer Science, University of Oregon, USA. Previously he worked as a member of technical staff at Nokia Bell Labs in Dublin, Ireland and also as a researcher at IBM Research in Beijing, China. He is interested in the mathematics of optimization, control, learning, and mechanism design applied to computer and telecommunication systems, networks, and services. He is a recipient of the NSF CAREER award. He publishes papers in journals such as JSAC, ToN, TPDS, TMC, and TDSC, and in conferences such as INFOCOM, MOBIHOC, ICNP, ICDCS, SECON, and IPDPS. He also received the Best Paper Awards of IEEE LANMAN 2013 and IEEE CNS 2019, and the 2016 Alcatel-Lucent Bell Labs UK and Ireland Recognition Award. He has been on the program committees of INFOCOM, MOBIHOC, ICDCS, TheWebConf, and IWQoS, and served as the program chair of multiple workshops with INFOCOM and ICDCS.

Xu Chen (Senior Member, IEEE) received the PhD degree in information engineering from the Chinese University of Hong Kong in 2012. He is currently a full professor with Sun Yat-sen University, Guangzhou, China, and the vice-director of the National and Local Joint Engineering Laboratory of Digital Home Interactive Applications. He was a postdoctoral research associate with Arizona State University, Tempe, USA, from 2012 to 2014, and a Humboldt scholar fellow with the Institute of Computer Science of University of Goettingen, Germany, from 2014 to 2016. He was the recipient of prestigious Humboldt research fellowship awarded by Alexander von Humboldt Foundation of Germany, 2014 Hong Kong Young Scientist Runnerup Award, 2016 Thousand Talents Plan Award for Young Professionals of China, 2017 IEEE Communication Society Asia-Pacific Outstanding Young Researcher Award, 2017 IEEE ComSoc Young Professional Best Paper Award, Honorable Mention Award of 2010 IEEE International Conference on Intelligence and Security Informatics (ISI), Best Paper Runner-up Award of the 2014 IEEE International Conference on Computer Communications (INFOCOM), and Best Paper Award of 2017 IEEE International Conference on Communications (ICC). He is currently an Area Editor of the IEEE OPEN JOURNAL OF THE Communications Society, an Associate Editor of the IEEE Transactions Wireless Communications, IEEE Internet of Things Journal and IEEE journal on selected areas in communications (JSAC) Series on Network Softwarization and Enablers.



Xiaoming Fu (Fellow, IEEE) received the PhD degree in computer science from Tsinghua University, Beijing, China, in 2000. He was then a research staff with the Technical University Berlin until joining the University of Göttingen, Germany in 2002, where he has been a professor in computer science and heading the Computer Networks Group since 2007. He has spent research visits at universities of Cambridge, Uppsala, UPMC, Columbia, UCLA, Tsinghua, Nanjing, Fudan, and PolyU of Hong Kong. His research interests include network architectures, protocols, and applications. He is currently an editorial board member of the IEEE Network and IEEE Transactions on Network and Service Management, and has served on the organization or program committees of leading conferences such as INFOCOM, ICNP, ICDCS, SIGCOMM, MOBICOM, MOBIHOC, CoNEXT, ICN and Networking. He is an ACM Distinguished Member, a fellow of IET, and member of the Academia Europaea.