# A UAV-Centric Improved Soft Actor-Critic Algorithm for QoE-focused Aerial Video Streaming

Abid Yaqoob, *Member, IEEE*, Zhenhui Yuan, *Member, IEEE*, Gabriel-Miro Muntean, *Fellow, IEEE*

***Abstract*—The increasing demand for uninterrupted connectivity emphasises the pivotal role of Unmanned Aerial Vehicles (UAVs) in facilitating real-time video streaming, despite the challenges associated with highly dynamic air-to-ground communications. Deep Reinforcement Learning (DRL)-based solutions (on-policy) are designed to optimize specific quality of experience (QoE) objectives, such as video quality and smoothness when networks fluctuate. However, they are vulnerable to different hyperparameters and have poor sample efficiency. To overcome this problem, we propose an improved off-policy soft actor-critic (SAC) solution, named I-SAC, which provides an exceptional exploration-exploitation trade-off for UAV-based aerial video streaming. I-SAC trains a neural network by jointly considering the video playback status, UAV flight metrics like altitude, velocity, and acceleration, as well as prior network conditions with the goal of maximizing the overall QoE. We design a new QoE metric that considers video quality, video quality oscillations, re-buffering, latency, and bandwidth utilization. We evaluate I-SAC with extensive real-world bandwidth settings, UAV flights, and multi-duration segment datasets. The trace-driven simulation results demonstrate that I-SAC significantly outperforms the closest on-policy and off-policy DRL-based alternative solutions in terms of QoE. Specifically, I-SAC achieves average QoE improvements of up to 54.32% under different testing scenarios.**

***Index terms*— Unmanned aerial vehicle, Deep reinforcement learning, Adaptive bitrate streaming, Soft actor-critic, End-user QoE.**

## I. Introduction

RECENTLY, unmanned aerial vehicles (UAVs) have shown great potential in the next generation communication services due to the lower capital and operational cost, spectral efficiency, improved coverage, ability to capture high-resolution images/videos, and real-time seamless transmission to the ground stations for detailed monitoring and analysis. The UAV market is expected to reach USD 45.8 billion by 2025 [1], making them a popular host with little or no human intervention for more practical applications such as real-time surveillance, emergency rescue, network coverage management, traffic monitoring, delivery services, computation offloading and multimedia streaming [2], [3] (Fig. 1). Compared with terrestrial base stations (BSs), cellular-connected UAVs can support real-time multimedia streaming owing to their flexible 3D mobility and improved wireless connectivity [4], [5]. However, there are several research challenges in delivering high-resolution video content because of the variable network capacity of air-to-ground communication channels and mobility patterns [6], [7].

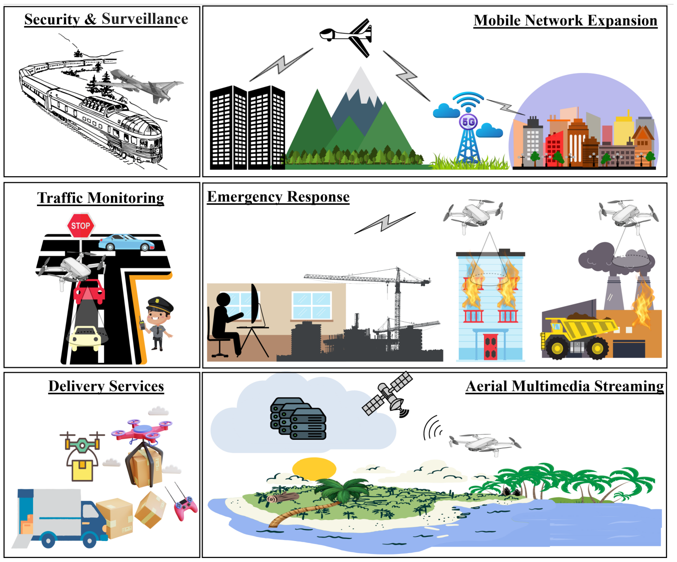


Fig. 1: UAV applications across a broad range of industries.

Deep Reinforcement Learning (DRL) models hold significant promise in addressing the complex and dynamic nature of UAV communication challenges. By using experience-driven bitrate selection in time-varying contexts, DRL-based approaches can adapt to current experiences and interact with the environment in a manner that leads to improved user quality of experience (QoE). Although recent studies have demonstrated the effectiveness of DRL-based approaches in enhancing QoE under traditional streaming architectures [8]–[10], these techniques face unique and significant challenges when applied to a UAV streaming environment.

Maintaining a clear line of sight between the ground station and the UAV is critical for connection stability, but obstacles, terrain, or distance can impede this line of sight, resulting in a lower quality transmission [11]. Therefore, the unpredictable nature of the UAVs' location and altitude, as well as the varying weather conditions and network coverage, pose significant

A. Yaqoob and G.-M. Muntean are with the Performance Engineering Lab and Insight SFI Centre for Data Analytics, School of Electronic Engineering, Dublin City University, Ireland, D09 DD7R, (e-mails: abid.yaqoob@dcu.ie and gabriel.muntean@dcu.ie). Z. Yuan is with the School of Engineering, University of Warwick, UK (e-mail: zhenhui.yuan@warwick.ac.uk). (Corresponding author: Gabriel-Miro Muntean.)

This work was funded in part by the Science Foundation Ireland (SFI) grant numbers 21/FFP-P/10244 (FRADIS) and 12/RC/2289_P2 (INSIGHT) and in part by EU Horizon Europe via grant agreement 101135637 (HEAT).

challenges for even well-trained agents in optimizing air-to-ground communication [12], [13]. The existing DRL models are inadequate in adapting to the exceedingly dynamic and unpredictable communication systems that are characteristic of UAV-based environments, resulting in undesirable consequences. One such consequence is ***Headstrong Policy Optimization***, which is observed in on-policy training methods like Advantage Actor-Critic [9], [14] and Asynchronous Advantage Actor-Critic [8], [15], [16]. These methods have poor training efficiency and are highly sensitive to hyperparameters, leading to a higher number of samples required for each gradient step update. Moreover, the existing models [8], [9], [17] that solely rely on bandwidth conditions and video playback states can result in unpredictable performance. Another adverse consequence is the development of ***Sub-Optimal QoE***. Existing QoE models fail to capture all factors that influence user perception and satisfaction within a UAV-based streaming scenario. QoE is a multifaceted concept that depends on various factors such as latency, bandwidth utilization, and playback smoothness, among others, [14], [18], [19]. Additionally, different users may react differently to streaming issues, with some users being more sensitive to rebuffering, whereas others may be more affected by low video quality. Therefore, to improve the performance of UAV-based adaptive video transmission, an algorithm must effectively model QoE, account for the dynamic UAV-based communication environment, include flight information (e.g., velocity, acceleration, and distance), and employ robust and sample-efficient off-policy training methods.

Driven by the aforementioned challenges, this paper presents a novel DRL-based Improved Soft Actor-Critic solution (I-SAC), for adapting real-time video transmission from UAVs to ground stations in order to improve viewer QoE levels. I-SAC is an improved variant of the off-policy soft actor-critic (SAC) algorithm [20], which is trained by reusing the past samples from an experience replay buffer to carry out an optimal bitrate allocation. One of the key advantages of the proposed I-SAC solution is its low-complexity and sample-efficient design, which greatly reduces the number of samples required for learning. Unlike other approaches that rely on multiple independent agents, our model uses a single learning agent, which reduces complexity and improves overall performance. Additionally, I-SAC is able to quickly learn by adapting to a wide range of flying situations to deliver optimal performance for various QoE goals. The following is a summary of the primary contributions of this work:

1) **Inherent Sensor Data-Based Aerial Bitrate Adaptation:** A Markov Decision Process (MDP) is used to model the problem at hand and a novel DRL model based on the maximum entropy framework is introduced to handle the inherent uncertainty and dynamic nature of the UAV-based communication environment. The sample-efficient design promotes exploration with stabilized training and circumvents the model's overcommitment to any single action during the sequential decision-making process. Unlike existing DRL-based solutions, the I-SAC model is able to perform more informed bitrate allocation decisions in continuous state and action space by utilizing a performance-oriented neural network architecture in order to process transmitter (UAV), receiver (player), and network data.
2) **Objective QoE Model for UAV Video Delivery:** A novel QoE metric is introduced that simultaneously considers both end-to-end metrics (latency and bandwidth utilization), as well as video streaming metrics (video quality, positive and negative quality oscillations, and playback rebuffering) within the context of adaptive bitrate streaming. Current QoE models, such as *Linear* QoE [21], *Log* QoE [22], *HD* QoE [13], and *VMAF* QoE [23], often overlook the significance of essential factors such as bandwidth utilization and latency. Consequently, their effectiveness may be limited to certain scenarios or QoE objectives. Furthermore, the behavior of these models is highly sensitive to minor adjustments in the weight coefficient. In contrast, the QoE model proposed in I-SAC offers a more holistic and precise evaluation of the viewer's experience in a UAV-aided streaming environment.
3) **The Impact of I-SAC on Aerial Video Streaming — Performance Analysis and Findings:** I-SAC is modelled using a Python-based simulator, set up to stream segments of varying durations (i.e., 2s, 3s, and 4s) from a flying UAV to a ground station, employing real-world communication settings. Our comprehensive experimental results highlight the effectiveness and superiority of I-SAC compared to other commonly used state-of-the-art DRL solutions. The results illustrate that an inherent sensor-data-oriented bitrate selection strategy is the optimal approach for managing real-time streaming, particularly in the context of experience-driven learning in complex environments. I-SAC consistently exhibits a reduction in convergence indeterminacy and stands out because of its low-complexity and sample-efficient design. These features highlight I-SAC as a compelling solution for improving QoE by 18.04% to 54.32% in UAV-based streaming applications.

This article is organized as follows: Section II reviews recent research on UAV-centric adaptive video streaming. Section III introduces the proposed system design, including the QoE model and problem formulation. Section IV elaborates on the I-SAC method and QoE optimization. Section V details the experimental setup and performance comparison of several streaming methods. Section VI concludes with future research directions.

## II. Related Works

This section provides a detailed overview of recent developments in the field of adaptive video streaming, which can be broadly classified into two categories: i) UAV-Centric Adaptive Video Streaming and ii) Learning-based Adaptive Video Streaming.

### A. Learning-based Adaptive Video Streaming

Learning-based frameworks are frequently utilized in MPEG-DASH systems to improve QoE for users. These

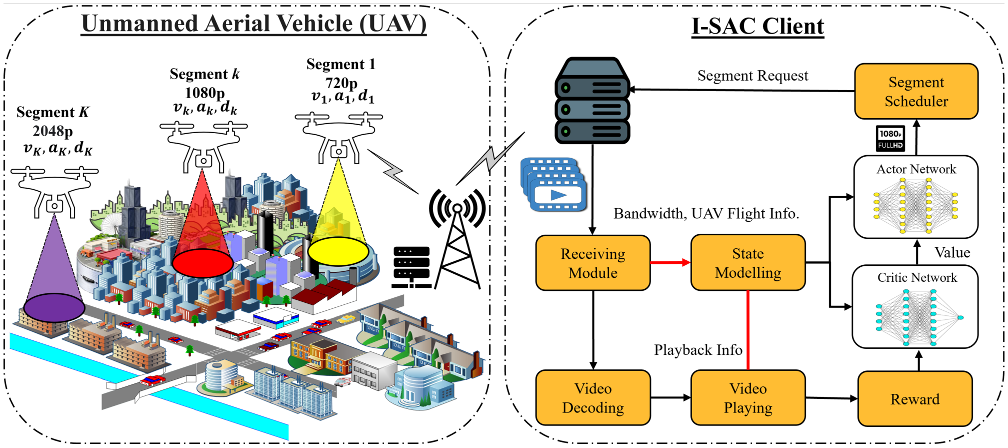


Fig. 2: The I-SAC system design with its major components.

frameworks utilize various reinforcement learning techniques to optimize the video quality and bitrate selection in real-time based on the user's QoE. A collaborative optimization approach for live transcoding and streaming in fog-enabled vehicular networks was proposed by Fu et al. [24]. In a live streaming model, a soft actor-critic DRL technique is employed to optimize the video quality and reduce the latency and quality changes by considering resource allocation for fog-computing, vehicle scheduling, and bitrate adaption. Cui et al. [25] proposed a DRL-based model to accelerate the learning process of optimizing the video parameters, i.e., streaming quality and the target buffer level. Experimental results reveal that the proposed solution can greatly improve the video quality and buffer stability in comparison to other approaches, i.e., Double DQN, MPC, and Buffer-based algorithms. Ma et al. [26] proposed QAVA, an HTTP-based QoE-aware bitrate aggregation solution based on DRL intelligence. QAVA aggregates client requests and adjusts bitrates based on client states, network circumstances, and content characteristics by monitoring network states and the storage and computational resources of a smart edge computing server. For ABR algorithms, online learning is essential since the state of the network is always altering. Meng et al. [27] proposed Fastconv, a DRL-based approach to enhance the convergence of adaptation algorithms. The proposed approach stabilizes input data through a network adaptation technique and employs a multiplexed convolution kernel to streamline the neural network design. Huang et al. [28] presented the Quality Aware Rate Control (QARC) solution for optimizing perceptual levels with reduced data and delay. QARC uses content facets for dynamic bitrate adaptation and incorporates a video quality prediction network (VQPN) to estimate future frame quality. Subsequently, video quality reinforcement learning (VQRL) models leverage VQPN outputs and network states to forecast the ideal future bitrate.

### B. *UAV-Centric Adaptive Video Streaming*

End-to-end video streaming over UAV communication networks is increasingly recognized for its potential in delivering high-quality video streams in contexts like remote areas, emergencies, and live events. However, realizing the full potential of UAV-aided streaming poses significant challenges. Burhanuddin et al. [10] employed Deep Q Network and Actor-Critic techniques for low-latency video transmission in UAV-to-UAV setups, focusing on optimizing base station coverage, bitrate, and transmission rate. The authors utilized UAVs in fire monitoring scenarios, with these UAVs sending video to a high-altitude flying mobile base station (UAV-BS).

Zhan et al. [29] introduced a rotary-wing UAV that serves as a mobile base station to stream Dynamic Adaptive Streaming over HTTP (DASH) video content to multiple ground users. The authors examined the integrated design of transmit power and bandwidth allocation to each ground user, as well as UAV trajectory planning to maximize the long-term QoE reward, which takes into account the perceived quality levels, playout buffer, and video transmission rate for all users. Chen et al. [30] considered a similar problem in UAV relay networks to serve multiple ground users. However, the proposed QoE model is limited, as it only takes into account video bitrate and freezing duration. Xiao et al. [14] proposed an SA-ABR solution that employs salient flying status information from UAVs to generate adaptive bitrate decisions. SA-ABR was trained to adapt to the unstable and highly dynamic UAV environment using the on-policy advantage actor-critic [35] method. The LSTM and CNN variants of SA-ABR outperformed Pensieve [13] and MPC [36] designs in terms of achieving higher video bitrates, lower quality fluctuations, and rebuffering penalties. Introducing a flock of UAVs, or swarm drones, is becoming more appealing due to the enormous potential in public safety, distribution, and surveillance applications, and it is essential for satisfying contemporary communication requirements.

Comsa et al. [31] proposed an actor-critic RL method for streaming live ultra HD video to mobile users. However, their approach did not optimize UAV parameters, focusing solely on the video streaming aspect. Conversely, Wu et al. [32] proposed a SAC solution for streaming scalable video coding (SVC) chunks to ground mobile users. This solution jointly optimizes UAV trajectory, video layer selection, and bandwidth resource allocation. Yuan et al. [33] developed EasySwarm, an open-source UAV swarming platform that forms a real-time IoT network for reliable and secure information transmission to the destination. Naveed et al. [34] proposed mobile UAVs acting as fully automated and independent servers for real-time

TABLE I: Comparative analysis of UAV-Assisted Adaptive Streaming Solutions

| Works | Performance Criteria | Model/Techniques | Streaming Scenario | UAV Role or Parameters | UAV Transmission | Extended QoE Formulation | Extended Segment Duration |
|---|---|---|---|---|---|---|---|
| [10] | Base station coverage, bitrate | Deep Q Network | Live streaming | UAV trajectory | A2A, A2G | No | No |
| [29] | Energy efficiency | Successive convex approximation | MPEG-DASH | UAV propulsion energy | A2G | No | No |
| [30] | Transmission power, Bandwidth allocation | Lyapunov optimization | Multi-user streaming | Transmission power | A2G | No | No |
| [14] | QoE Optimization | Actor-Critic | MPEG-DASH | Sensory data | A2G | No | No |
| [31] | PSNR, Throughput, Packet loss rate | Actor-Critic | Live UHD | Minimal | A2G | No | No |
| [32] | Bitrate, Delay, Bitrate switching | Soft Actor-Critic | SVC | UAV trajectory, UAV energy | A2V | No | No |
| [33] | Reliability, Delay | LoRa and MAC layer | IOT traffic | GPS, Speed, IP address, Battery | A2G | No | No |
| [34] | PSNR, SSIM | Fixed rules heuristic | Live streaming | Fixed, Random flights | A2G | No | No |
| **This work** | **QoE optimization** | **Improved Soft Actor-Critic** | **MPEG-DASH** | **UAV flight information** | **A2G** | **Yes** | **Yes** |

surveillance applications. The mobile UAVs leveraging cross-layer protocols are responsible for content capturing, bitrate computation, video conversion, and transmission over 4G LTE networks to the remote user.

Table I illustrates a comparison of recently proposed UAV-assisted adaptive streaming solutions. Deep Q Network-based transmission rate optimization method [10] faces limitations in continuous control problems common to Air-to-Air (A2A) and Air-to-Ground (A2G) environments. Optimization problems solved by successive convex approximation (SCA) [29] or Lyapunov optimization [30] risk sub-optimal convergence due to potential noise disturbances. On-policy actor-critic solutions [14], [31] utilize temporal difference methods for single-step value function updates, but may struggle to identify the best actions within continuous state and action spaces common in adaptive streaming scenarios. Prior efforts [32]–[34] do not fully address the complexities of the UAV streaming space, leading to sub-optimal performance. In contrast, this work presents a novel approach that achieves stable and sample-efficient exploration-exploitation procedures, resulting in enhanced QoE within the UAV streaming space. This work employs an extended QoE formulation and is tested under extended segment durations (i.e., 2s, 3s, and 4s) to provide a more robust adaptive streaming solution.

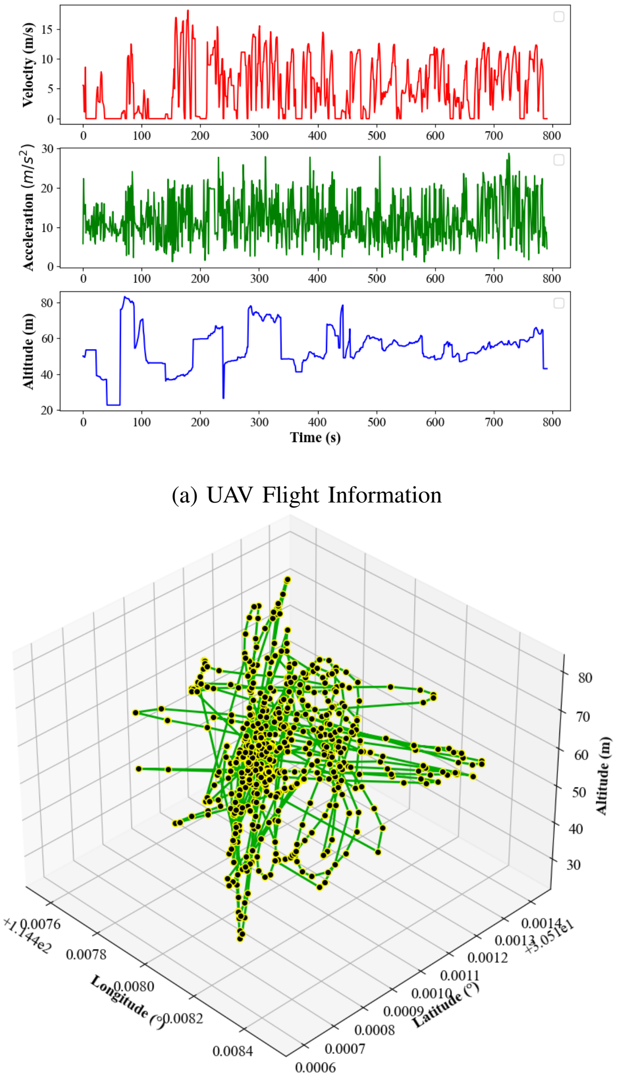
## III. Proposed System Architecture

### A. System Overview

Fig. 2 illustrates the proposed UAV-centric adaptive bitrate streaming architecture. The multi-rotor UAV equipped with a camera is responsible for capturing video while in flight and transmitting the data in real-time over a a persistent cellular network. In a surveillance scenario, UAV follows a navigation trajectory with orientation information $\mathcal{V}(U_x, U_y, U_z)$ to record and transmit the video during the full playback of $K$ segments. Fig. 3 displays the flight data of the UAV during a flight that lasted approximately 800 seconds from [14]. Fig. 3a shows the velocity, acceleration, and altitude values, while Fig. 3b represents the flight path of the UAV. The UAV velocity spans from 0.0 m/s to 18.11 m/s, with an average velocity of 5.14 m/s. The acceleration varies from 1.13 $m/s^2$ to 28.76 $m/s^2$, and the altitude ranges from 22.8 m to 83.2 m, with an average altitude of 54.28 m. The average longitude and latitude values for the UAV's flight path are 114.4° and 30.51°, respectively. These specifications pertain to a highly dynamic transmission environment in order to establish a well-defined UAV streaming setup.

The UAV captures sensory data using various built-in sensor technologies, such as GPS and an inertial measurement unit (IMU). The flying UAV transmits both high-quality DASH video segments and sensory data over a cellular network. For lightweight sensory data transmission, it employs the Micro Air Vehicle Link (MAVLink) protocol, which can carry sensor data, flight parameters, and other control commands. At the client side in Fig. 2, the state modelling module is employed in order to effectively model the state space in a DRL framework. In this context, the past bandwidth samples, UAV flight information, and playback information are passed to the I-SAC agent. The UAV flight information, i.e., distance, velocity, acceleration, etc., is essential data that can broadly impact the future bandwidth estimation and ultimately the adaptive bitrate selection. For instance, in [14], the researchers found that the throughput drops to more than 50% when increasing the height of the UAV from 20 m to 60 m from the ground station while maintaining the same speed. Similarly, by doubling the velocity of the UAV from 4 m/s to 8 m/s, the throughput drops by about three times under the fixed distance. The variations in the acceleration data also hurt the available throughput. This highlights the importance of incorporating UAV sensory information in the adaptive bitrate selection process to ensure that the video quality is optimized for the changing network conditions.

In a complex UAV environment, the bitrate selection problem is formulated as an optimization problem to maximize the overall reward while taking into account the previous bandwidth samples, UAV sensory information, and the playback information, such as the number of remaining segments, current buffer level, bitrate of the last segment, prior reward, and the last segment download time. The off-policy maximum



(a) UAV Flight Information

(b) UAV Flight Path

Fig. 3: UAV flight information (velocity, acceleration, altitude) and flight path for a flight duration of about 800s.

entropy I-SAC client with an encouraging exploration generates the appropriate bitrate decisions for each timestamp in an adaptive fashion. Let $J_k^q$ represents the $q$th selected bitrate for the $k$th segment, where $q \epsilon [1, Q]$ and $k \epsilon [1, K]$. The segment request is then sent through a segment scheduler module, and upon receiving the segments, the client decodes and plays the requested views.

### *B. QoE Model*

Linear QoE models have been extensively employed to evaluate the streaming performance of various adaptive bitrate solutions [37]–[40]. These QoE models predominantly involve objectives related to video quality, rebuffering, and quality oscillations. Nonetheless, several other parameters, such as latency, bandwidth utilization, and positive and negative quality variations, substantially influence the overall performance assessment [41], [42].

- **Video Quality:** By mapping the bitrate of each segment, the relevant perceived quality can be computed as:

$$f_k^1 = Q(J_k^q) \tag{1}$$

There are different methods to model the bitrate utility. Following [21], we set $Q(J_k^q) = \log(J_k^q / J_k^1)$, which demonstrates that the slight quality gain declines with increasing bitrates.

- **Quality Oscillations:** The difference in quality levels between two successive segments potentially reduces the streaming performance in HTTP Adaptive Streaming (HAS). As a result, the quality fluctuations should be minimal [23]. Positive quality oscillations refer to the situation where the video quality temporarily increases above the target quality level. Positive quality oscillations can improve the overall viewing experience, but can also cause the buffer to fill up too quickly, and the video may stall or buffering may occur. The positive quality oscillation metric can be defined as follows:

$$f_k^2 = [f_k^1 - f_{k-1}^1]_+ \tag{2}$$

Negative quality oscillations refer to the situation where the video quality temporarily decreases below the target quality level. This can happen for a variety of reasons, such as when the user's network conditions degrade or when the video encoder is configured to under-deliver quality. Negative quality oscillations can negatively impact the overall viewing experience and lead to lower perceived video quality. The negative quality oscillation function is given as follows:

$$f_k^3 = [f_k^1 - f_{k-1}^1]_- \tag{3}$$

Both positive and negative quality oscillations are unwanted effects and the adaptive client should be able to smoothly play the highest quality content.

- **Rebuffering:** A user experiences playback interruptions when the playback buffer is empty and is highly sensitive to the video viewer. For instance, the authors in [43] showed that a single rebuffering event leads to three times higher risks of playback abandonment compared to a single-quality oscillation event. When downloading segment $k$, rebuffering duration can be estimated as:

$$f_k^4 = \begin{cases} t_k - b_k, & \textbf{if } (b_k < t_k) \\ 0, & Otherwise \end{cases} \tag{4}$$

where $t_k$ and $b_k$ are the segment download time and the buffer capacity for the $k$th interval.

- **Latency:** Latency is an important factor to consider in order to assess the playback performance of adaptive bitrate streaming algorithms. Ideally, the client should be able to buffer the upcoming segments in advance before the previous segments consume. When downloading segment $k$, the latency can be measured as:

$$f_k^5 = t_k = \frac{\nu \times J_k^q}{x_k} \tag{5}$$

where $v$ is the segment duration and $x_k$ is the actual recorded bandwidth.

- **Bandwidth Utilization:** Ideally, the adaptive client should select the highest bitrates supported by the available bandwidth budget [44]. However, over- or under-estimating the bandwidth leads to wrong quality segments, impacting the end-user expectation levels.

  The bandwidth utilization function highly correlated with the performance of adaptation algorithms is defined as follows [45]:

$$f_k^6 = \begin{cases} \frac{J_k^q}{x_k}, & \textbf{if } (J_k^q \leq x_k) \\ \left(1 - \frac{J_k^q}{x_k}\right), & Otherwise \end{cases} \tag{6}$$

This work considers a weighted combination of video quality, bandwidth utilization, positive and negative quality oscillations, and rebuffering functions to define an improved QoE metric for adaptive video streaming as follows:

$$QoE_k = \beta_1 \times f_k^1 + \beta_2 \times f_k^2 - \beta_3 \times f_k^3 - \beta_4 \times f_k^4 - \beta_5 \times f_k^5 + \beta_6 \times f_k^6 \tag{7}$$

where $\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6$ are the weight coefficients representing the importance of video quality, positive and negative quality oscillations, rebuffering, latency, and bandwidth utilization, respectively. As users want to maximize $f_k^1$, $f_k^2$, and $f_k^6$, their contributions are positive in the novel QoE model. As viewers want to minimize $f_k^3$, $f_k^4$, and $f_k^5$, their contributions are negative in the model.

### C. Problem Formulation

In a client-centric end-to-end HTTP adaptive streaming architecture, the ultimate goal of the adaptive client is to continuously access optimal bitrates during each adaptation interval, thus maximising the aggregated QoE of all video segments. The optimization problem in our case can be expressed mathematically as follows:

**Problem P1:**

$$\textbf{arg } max \sum_{k \in [1,K]} QoE_k \tag{8}$$

The flying status of the UAV, the client's behaviors, and the available network capacity are all difficult to anticipate in advance, and as a result, it is challenging to maximize the QoE using deterministic one-shot optimization techniques [46]. Here, we take into account a DRL framework in which an agent (i.e., the DASH client) learns via interaction with the environment the optimum course of action (i.e., the best-fit bitrate) to attain the anticipated reward (i.e., optimizing the long-term QoE).

In the DRL paradigm, the agent takes future decisions with little or no awareness of the environment with a trial-and-error strategy to achieve a maximum reward. The MDP procedure that represents an agent-environment interaction can be described using a tuple, i.e., $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r)$. $\mathcal{S}$ and $\mathcal{A}$ represent the continuous state and action space, $\mathcal{P}$ is the state transition probability, which represents the probability density of the future state given the current state and action, and $r$ is the returned reward. In the context of adaptive bitrate streaming in a UAV environment, the DRL-based adaptive client receives a reward $r_k$ after executing an action $a_k \in \mathcal{A}$ with state $s_k \in \mathcal{S}$.

**State:** State records the most relevant observations of the environment. We consider the maximum possible states to better explore the environment. When downloading the $k$th segment, the current state $s_k$ about the environment is passed to the agent, which is defined as:

$$s_k = (\vec{y_k}, v_k, a_k, d_k, b_k, J_{k-1}^q, f_{k-1}, r_{k-1}, \bar{k}) \tag{9}$$

where $\vec{y_k}$ represents the bandwidth vector and $v_k$, $a_k$, $d_k$ represents UAV flight information, i.e., velocity, acceleration, and distance from the ground station. $b_k$ is the occupied buffer level to download segment $k$, $J_{k-1}^q$ is the bitrate of the last segment, $f_{k-1}$ is the download time, $r_{k-1}$ is the previously observed reward, and $\bar{k}$ is the number of remaining segments.

**Action:** The agent acts to transform the environment, resulting in a state transition from $s_{k-1}$ to $s_k$. When $(k-1)$th segment is completely downloaded, the agent determines the bitrate for the $k$th segment, i.e., $a_k = J_k^q$, based on the observed state $s_k$.

**Reward:** After the execution of each action, the environment returns instant feedback, i.e., QoE, to the agent, which is defined as:

$$r_k = QoE_k \tag{10}$$

Concerning problem P1, the goal has been adjusted to discover an optimal bitrate selection strategy $\pi^* : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ so that to maximize the expected long-term discounted QoE.

Consequently, the client-side optimal bitrate selection problem is remodeled as:

**Problem P2:**

$$arg \max_{\pi \epsilon \prod} \mathbb{E}_\pi [\sum_{k \epsilon K} \gamma^k r_k] \tag{11}$$

where $\prod$ is a collection of all potential policies and $\gamma \epsilon [0, 1]$ is the discount factor. The goal of optimum representation selection or maximizing anticipated reward may be converted further to identify a policy that maximizes the maximum entropy objective, i.e, $\mathcal{H}(\pi(a|s)) = -\log \pi(a|s)$ [47]. Entropy maximization can improve exploratory abilities, speed up learning, and discourage the policy from converging to a problematic local optimum. Thus the overarching objective is rewritten as:

$$\pi^* = arg \max_{\pi \epsilon \prod} \mathbb{E}_\pi [\sum_{k \epsilon K} \gamma^k (r_k - \log \pi(a_k|s_k)] \tag{12}$$

We employ an improved off-policy method to solve the above MDP problem.

## IV. Design of I-SAC to Meet QoE Objectives

This section presents the details of our proposed algorithm that improves the adaptive video streaming experience based on the SAC learning framework. We need accurate measurements in order to use the network conditions and bitrate choices more effectively. The I-SAC solution utilizes an advanced neural network architecture, which is depicted in Fig. 4. This architecture employs a number of states to generate

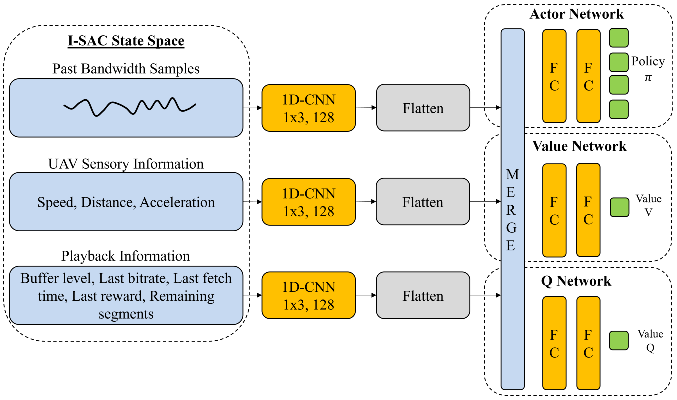


Fig. 4: The I-SAC model with two fully connected layers.

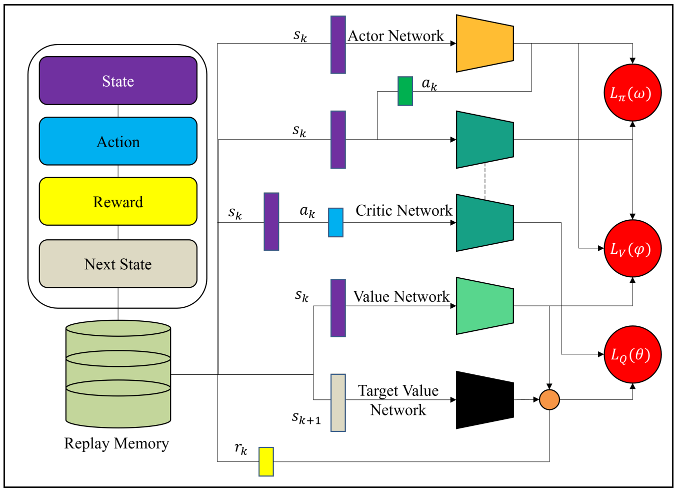


Fig. 5: Training methodology of the proposed I-SAC model.

robust bitrate selection decisions. The bitrate selection process takes into account metrics that are beneficial for both the transmitter (UAV) and the receiver (Player) as per the feedback received during an adaptive interval. The neural network architecture of I-SAC comprises a convolutional neural network (CNN) that is fed by the I-SAC learning agent with inputs such as bandwidth, UAV flight data, and playback information including buffer level, last received video bitrate, segment download time, prior reward, and the number of remaining segments. A focused and appropriate strategy is implemented to extract the underlying features for each type of input. For instance, a single 1D-CNN layer with a 1x3 kernel and 128 channels is utilized to extract network characteristics. This is followed by two identical 1D-CNN layers used to extract pertinent information from the UAV and the video-playing client. The processed features are then flattened and combined before passing into two fully connected (FC) layers. Finally, the actor outputs the action probabilities for each bitrate. ReLU (Rectified Linear Unit) is used as the active function for each feature extraction layer. The architecture is designed such that it considers both the UAV and the client-side information, thus making the decision of bitrate selection more robust and reliable.

## A. I-SAC Training Mechanism

I-SAC is designed to optimize the long-term reward in sequential decision-making tasks. The architecture consists of two main components: the actor network and the critic network. The actor network or policy improvement network is responsible for generating actions based on the current state of the environment. The critic network, also known as the policy evaluation network, is responsible for evaluating the quality of the actions generated by the actor network. The overall training procedure of the I-SAC model is shown in Fig. 5, where the actor network executes a specific action and the critic network calculates the soft state-value $V_\phi(s_k)$ and soft Q-values $Q_{\theta_i}(s_k, a_k)$. In training, a replay buffer is employed to store the fresh experiences at each timestamp, i.e., $\mathcal{M} \leftarrow (s_k, a_k, r_k, s_{k+1}) \cup \mathcal{M}$. The use of a replay buffer allows the model to learn from previously experienced data, rather than only the current data. This can help to improve the consistency of the training process and reduce the impact of any data dispersion. The neural network provides the mean and covariance, which are then used to represent the policy as a Gaussian distribution [20].

*1) Critic Network:* Critic network consisting of soft state-value $V_\phi$ and soft Q-value $Q_{\theta_i}$ networks aims to approximate the value functions. These networks are parameterized by $\phi$ and $\theta$. The soft Q-value network's performance is determined by the mean squared error (MSE), as shown below:

$$\mathcal{L}_Q(\theta) = \mathbb{E}_{(s_k, a_k)}\left[\frac{1}{2}\left(Q_\theta(s_k, a_k) - \hat{Q}_\theta(s_k, a_k)\right)^2\right] \tag{13}$$

with

$$\hat{Q}_\theta(s_k, a_k) = r_k + \gamma \times \mathbb{E}_{(s_{k+1})}[\hat{V}_{\bar{\phi}}(s_{k+1})] \tag{14}$$

$\hat{V}_{\bar{\phi}}(s_{k+1})$ represents a weighted combination of the soft and target values with parameters $\phi$ and $\bar{\phi}$, and is given as:

$$\hat{V}_{\bar{\phi}}(s_{k+1}) = \rho . V_\phi(s_k) + (1 - \rho) . V_{\bar{\phi}}(s_{k+1}) \tag{15}$$

where $\rho$ is a coefficient to balance the relative importance of $V_\phi$ and $V_{\bar{\phi}}$, which can be calculated by:

$$\rho = \frac{\exp(\mu)}{1 + \exp(\mu)} \tag{16}$$

where $\mu$ is the normalised difference between $V_\phi$ and $V_{\bar{\phi}}$:

$$\mu = \frac{|V_\phi(s_k) - V_{\bar{\phi}}(s_{k+1})|}{\max\{V_\phi(s_k), V_{\bar{\phi}}(s_{k+1})\}} \tag{17}$$

In the I-SAC model, the Q-value network is updated with the soft Q value and the target value functions, which can stabilize training. The gradient of $L_Q(\theta)$ with respect to $\theta$ is used to update the soft Q-value network parameter:

$$\nabla_\theta \mathcal{L}_Q(\theta) = \nabla_\theta Q_\theta(s_k, a_k)\left[Q_\theta(s_k, a_k) - r_k - \gamma \times \hat{V}_{\bar{\phi}}(s_{k+1})\right] \tag{18}$$

The target value network enables training stability and shares a similar network structure as the value network. An exponentially moving average is employed to compute the parameter $\bar{\phi}$ with smoothing factor $\delta \epsilon (0, 1)$.

$$\bar{\phi} \leftarrow \delta . \phi + (1 - \delta) . \bar{\phi} \tag{19}$$

On the other hand, the soft value network $V_\phi$ for value estimation is trained to minimize the following MSE:

$$\mathcal{L}_V(\phi) = \mathbb{E}_{(s_k)}\left\{\frac{1}{2}(V_\phi(s_k) - \mathbb{E}_{(a_k)}[Q_\theta(s_k,a_k) - \log\pi_\omega(a_k|s_k)]^2\right\} \tag{20}$$

Correspondingly, the gradient of $\mathcal{L}_V(\phi)$ is calculated as

$$\nabla_\phi\mathcal{L}_V(\phi) = \nabla_\phi V_\phi(s_k)\big[V_\phi(s_k) - Q_\theta(s_k,a_k) + \log\pi_\omega(a_k|s_k)\big] \tag{21}$$

*2) Actor Network:* During the learning process, the agent accumulates experience through interaction with the environment. The actor approximates policy using a parameterized neural network and decides and implements an action based on the current state of the environment. The predicted soft Q-value is used by the actor network to change its policy $piomega(.|s)$. The actor network seeks to minimize the following objective function for policy improvement:

$$\mathcal{L}_\pi(\omega) = \mathbb{E}_{(s_k)}[\log\pi_\omega(a_k|s_k) - Q_\theta(s_k,a_k)] \tag{22}$$

The objective function in eq. (22) is re-parameterized using a neural network transformation, i.e., $a_k = \sigma_\omega(\tau; s_k)$, where $\tau$ is an input Gaussian noise vector [48].

$$\mathcal{L}_\pi(\omega) = \mathbb{E}_{(s_k)}[\log\pi_\omega(\sigma_\omega(\tau;s_k)|s_k) - Q_\theta(s_k,\sigma_\omega(\tau;s_k))] \tag{23}$$

The gradient of $\mathcal{L}_\pi(\omega)$ is given as:

$$\begin{aligned}\nabla_\omega\mathcal{L}_\pi(\omega) = \nabla_\omega \log\pi_\omega(a_k|s_k) + \big[\nabla_{a_k}\log\pi_\omega(a_k|s_k) \\ -\nabla_{a_k}Q_\theta(s_k,a_k)\big]\nabla_\omega\sigma\omega(\tau_k;s_k)\end{aligned} \tag{24}$$

A well-trained policy can eventually be implemented in real-world networks, achieving low computational complexity and real-time execution. Algorithm 1 presents the training procedure for the proposed I-SAC algorithm. The process starts with the initialization of the actor and critic networks, along with a replay buffer ($\mathcal{M}$) for storing the experience tuples. Within each training step of a data trajectory, the actor examines the current environment state ($s_k$) and selects an action ($a_k$) based on the current policy. This chosen action influences the environment, and the agent subsequently receives a reward ($r_k$) and updated observations ($s_{k+1}$). These experiences ($s_k, a_k, r_k, s_{k+1}$) are stored in the replay buffer. During each update step, the agent randomly samples experiences from the replay buffer and updates the parameters of the actor and critic networks via backpropagation of the loss functions to optimize the learning performance. Fig. 5 depicts the design of the two Q-networks utilized in updating the actor and critic network parameters, structured to prevent biases during the policy improvement phase. The value network is specifically employed to update the parameters of the critic network. The network parameters are updated using the stochastic gradient descent method, minimizing the loss functions. This process is repeated for a specified number of iterations until the agent is considered trained. The agent is then returned for use in decision-making.

**Algorithm 1:** Proposed Overall Training Procedure in the I-SAC Agent

```
Input : $\theta_1$, $\theta_2$, $\phi$, $\omega$ ← Initial network parameters; $\bar{\phi}$
        ← $\phi$; $\lambda_\theta$, $\lambda_\phi$, $\lambda_\omega$ ← Learning rates of
        network; $\mathcal{T}$ ← Training dataset; $\mathcal{M}$ ←
        Replay memory.
Procedure TRAINING begin
    for each data trajectory in $\mathcal{T}$ do
        for each environment step do
            Select action $a_k$ with policy $\pi_\omega(.|s_k)$;
            Perform action $a_k$ in the environment;
            Observe next state $s_{k+1}$ and reward $r_k$;
            Save transitions in replay memory $\mathcal{M}$ ←
              $\mathcal{M}$ ∪ $(s_k, a_k, r_k, s_{k+1})$;
        end
        for each update step do
            Sample a random experience batch from
              $\mathcal{M}$;
            Update $\theta_i$ ← $\theta_i$ - $\lambda_\theta$ $\nabla_\theta\mathcal{L}_Q(\theta_i)$ for
              $i\epsilon\{1,2\}$, with $\nabla_\theta\mathcal{L}_Q(\theta_i)$ defined in eq.
              (18);
            Update $\phi$ ← $\phi$ - $\lambda\phi$ $\nabla_\phi\mathcal{L}_V(\phi)$, (Eq. 21);
            Update $\omega$ ← $\omega$ - $\lambda\omega$ $\nabla_\omega\mathcal{L}_\pi(\omega)$, eq. (24);
            Update $\bar{\phi}$ ← $\delta$ . $\phi$ + $(1-\delta)$ . $\bar{\phi}$;
        end
    end
end
end procedure
```

TABLE II: Hyperparameters employed in I-SAC

| Parameter | Value |
|---|---|
| Optimizer | Adam [49] |
| Replay buffer size | $10^6$ |
| Discount factor | 0.99 |
| Learning rate actor | $3x10^{-5}$ |
| Learning rate critic | $5x10^{-3}$ |
| $\delta$ | $10^{-2}$ |
| Training steps | 8000 |
| Update interval | 41 segments |
| Hidden units FC1 | 128 |
| Hidden units FC2 | 64 |

## V. Performance Evaluation

This section introduces the proposed solution trace-driven performance assessment in comparison to other on-policy and off-policy DRL-based adaptive streaming solutions. The experimental setup, datasets, comparative approaches, and QoE assessment metrics are presented in detail. The experimental findings and analysis for each of the streaming solutions are then presented.

### A. Experimental Setup

We developed a Python-based simulator using PyTorch framework [50] to evaluate the performance of the proposed I-SAC algorithm. The simulator was implemented on an Ubuntu 22.04 machine, equipped with a 64-bit Intel Core i7-7500U CPU 2.7 GHz quad-core, and 16 GB memory. The actor and critic networks in the I-SAC architecture are composed of two fully connected layers, each with 128 and 64 neurons, respectively. In the I-SAC training process, the actor and critic learning rates are set to $3x10^{-5}$ and $5x10^{-3}$. The replay

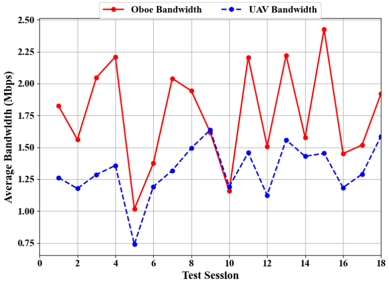


Fig. 6: Average bandwidth in each testing step of UAV and Oboe network traces.

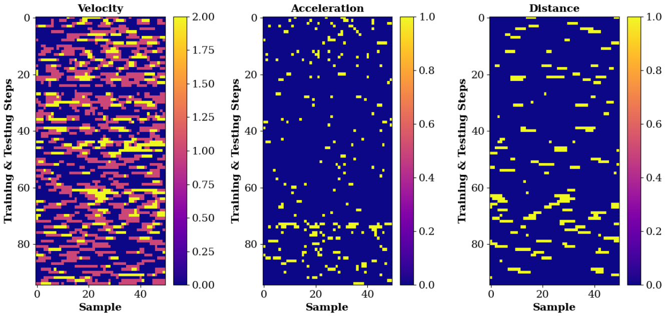


Fig. 7: Distribution of UAV velocity, acceleration, and distance values.

buffer size, discount factor $\gamma$, and smoothing coefficient are set to $10^6$, 0.99, and 0.01 respectively. The total number of training steps was 8000 with 41 segments in each step. The hyperparameters utilized for the training procedure are listed in Table II. These values were chosen based on the results reported in previous works [14], [20], [24]. The I-SAC algorithm was trained using these hyperparameters, and the performance was evaluated by comparing the results with the state-of-the-art algorithms.

### B. Experimental Dataset

We evaluated the performance of the proposed I-SAC algorithm using extensive trace-driven simulations. We considered three types of datasets, namely, bandwidth dataset, UAV dataset, and video bitrates dataset. These datasets are described below:

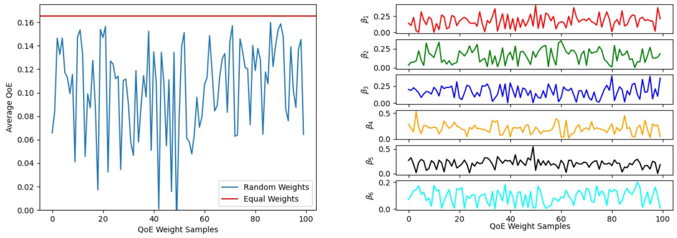
1) **Bandwidth Dataset:** We employed two bandwidth datasets, namely, UAV dataset [14] and Oboe dataset [51]. These datasets contain collected wireless and cellular network traces. The UAV dataset consists of real-world network traces collected by flying the UAV over three urban experimental sites. We employed 4750 bandwidth samples which range from 0 to 16.67 Mbps. The training set contains the first 80% samples of the data, and the testing set is comprised of the remaining 20%. The 428 traces in the Oboe dataset were gathered across 500 video streaming sessions. The network throughput ranges from 0 to 3 Mbps. To increase the evaluation efficiency, we used 900 bandwidth samples from the Oboe dataset. It is important to note that the UAV dataset and Oboe dataset are representatives of real-world scenarios and provide a good evaluation of the performance of the proposed I-SAC algorithm in different network conditions. The normalization of the bandwidth samples also allows us to evaluate the algorithm's performance in constrained streaming sessions, which is an important aspect of practical applications. Fig. 6 shows the average bandwidth samples employed for 18 testing steps from both datasets.
2) **UAV Dataset**: We employed the UAV flying information (i.e., velocity, acceleration, distance) from the UAV dataset [14]. The data is preprocessed according to the original settings. Fig. 7 provides a visual representation of the distribution of UAV velocity, acceleration, and distance values. The heatmap is color-coded to show the frequency of these values, which are normalized as 0, 1, or 2 based on their respective ranges. For UAV velocity, values below 8 m/s are marked as "0", values between 8 m/s and 12 m/s are marked as "1", and values over 12 m/s are marked as "2". The heatmap in Fig. 7 shows that the UAV velocity values are distributed over a wide range, with the majority of values falling between 5 m/s and 15 m/s. For UAV acceleration, values above 18 $m/s^2$ are marked as "1", while values below 18 m/$s^2$ are marked as "0". The majority of UAV acceleration values fall below 18 m/$s^2$. Regarding UAV distance, values exceeding 50 m are marked as "1", while the other values are marked as "0". Fig. 7 indicates that the majority of UAV distance values fall below 50 m.
3) **Video Bitrates:** We employed four video bitrates 0.396 Mbps (480x360), 1.033 Mbps (1280x720), 1.547 Mbps (1280x720), and 2.484 Mbps (1920x1080) from the DASH dataset[1]. The video is temporally divided into 2s segment duration, and each streaming session consists of 41 segments.

### C. Comparative Solutions:

In order to assess the proposed solution's performance, I-SAC is compared to the following state-of-the-art ABR algorithms.

1) **A2C-LSTM:** We used SA-ABR [14] solution in conjunction with an underlying LSTM architecture and the Advantage Actor Critic (A2C) approach to compare the training performance. The neural network in A2C-LSTM involves two LSTM layers each with 64 hidden cells and two FC layers, each with 30 and 10 hidden units.
2) **A2C-CNN:** The A2C-CNN architecture [14] replaces the base LSTM structure with two 1-D convolutional layers each with 64 neurons. The CNN output is combined with other features and fed into three fully connected layers with 128, 30, and 8 hidden units.
3) **SAC-LSTM:** The off-policy original SAC model [20] is adopted to evaluate and analyze the fine-tuning methods critically. The SAC-LSTM algorithm uses the same network architecture and input states as employed in A2C-LSTM. The hyperparameters employed in this architecture are as follows: actor and critic learning rates

[1] http://www-itec.uni-klu.ac.at/ftp/data-sets/mmsys12/ BigBuckBunny/



(a) Average QoE Vs. Weights

(b) QoE Weight Coefficient

Fig. 8: Average QoE values with 100 random samples of QoE weight coefficients.

are $2\times10^{-5}$ and $10^{-2}$; $\delta$ is $10^{-2}$; experience replay buffer is set to $10^6$.

4) **SAC-CNN:** It is a combination of SAC-LSTM and A2C-CNN, where the SAC-LSTM network architecture is replaced with A2C-CNN settings.

### *D. QoE Coefficients*

Our main objective is to propose a viable UAV ABR solution that improves viewer QoE. The proposed QoE metric uses weight parameters that can be realistically chosen to best reflect various quality aspects. We normalized the values of the individual components to reduce bias caused by the division of data. The values of the functions $f_k^1$, $f_k^2$, $f_k^4$, and $f_k^5$ are in the range of 0 to 1, whereas the values of the functions $f_k^3$ and $f_k^6$ range between -1 and 0, and -1 and 1, respectively.

Following [52], approximately 27000 streaming samples were collected to mitigate potential bias in the computation of QoE weights. The samples were assessed by using 100 randomly applied QoE weight coefficients. The average QoE outcomes for randomly weighted training samples are shown in Figure 8a. The associated weight coefficients denoted by $\beta_s$ are shown in Fig. 8b. For each randomly weighted sample, the sum of QoE weights was set to 1. A maximum average QoE value of 0.159 was observed at sample index 86, and the corresponding weight coefficient values at this index were $\beta_1 = 0.117$, $\beta_2 = 0.224$, $\beta_3 = 0.181$, $\beta_4 = 0.202$, $\beta_5 = 0.118$, and $\beta_6 = 0.154$. Furthermore, a scenario was considered in which all QoE components were deemed equally important by video consumers. The average QoE value calculated with equal weights was 0.165, which was higher than that achieved with random weights, as shown in Fig. 8a. As a result, the proposed QoE metric employs equal weights for all of its components, i.e., $\beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = \beta_6 = 1/6$.

### *E. Results and Analysis*

This subsection presents comprehensive comparative evaluations and analyses of the proposed and alternative solutions. We consider the overall QoE assessment, individual components assessment, and segment duration assessment to analyze the performance and streaming behavior of the different streaming algorithms.

*1) Overall QoE Assessment:* Fig. 9 shows the benefits of using the proposed **I-SAC** algorithm for adaptive bitrate streaming in a UAV environment. The streaming performance in terms of average QoE under Oboe and UAV bandwidth traces achieved by **I-SAC**, SAC-CNN, SAC-LSTM, A2C-CNN, and A2C-LSTM solutions is illustrated. Fig. 9 results indicate that **I-SAC** can achieve optimal performance when tested using dynamically changing UAV network traces. The average reward value of **I-SAC** is 0.59, while the average score of other SAC and A2C-based algorithms is 0.55, and 0.43, respectively. Similarly, **I-SAC** outperforms comparative solutions for the Oboe traces and achieves on average a reward of 0.71 compared to 0.66 and 0.55 for SAC-LSTM and A2C-CNN solutions, respectively. **I-SAC** increases the QoE scores by about 7.47%-7.96% (over SAC-CNN), 7.33%-7.6% (over SAC-LSTM), 29.22%-39.38% (over A2C-CNN) and 11.17%-34.18% (over A2C-LSTM) for UAV and Oboe bandwidth settings, respectively. The higher performance of A2C-LSTM over A2C-CNN is due to the higher mapped visual quality levels; however, this could result in unwanted higher rebuffering and latency penalties. The results reveal that **I-SAC** can deal with variable networks and flying conditions in the context of experience-driven adaptive streaming. In addition, it can be seen from Fig. 9 that regardless of the network configurations, **I-SAC** could successfully adapt to the new environment and steadily obtains the best performance with the assistance of both previous learning and fresh experience.

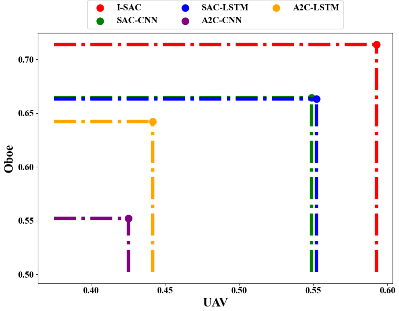
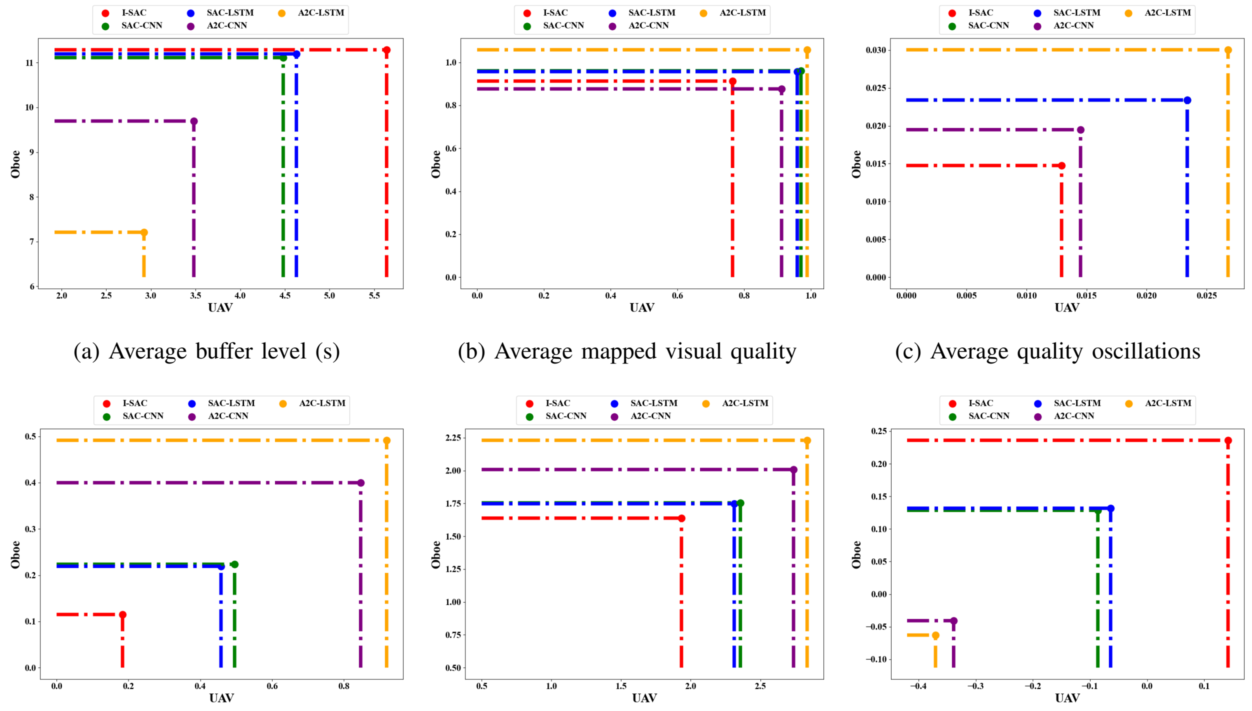


Fig. 9: Average QoE achieved by **I-SAC** and four other comparative algorithms under UAV and Oboe network traces.

*2) E2E and Playback Components Assessment:* Next, we will separately assess the performance against individual QoE components, i.e. buffer, visual quality, quality oscillation, rebuffering, latency, and bandwidth utilization. These results are depicted in Fig. 10. According to Fig. 10a, we can observe that **I-SAC** performs better than SAC and A2C solutions in terms of buffer value. The main factor enabling **I-SAC** to improve DRL agent training and performance is the effective learning of new experiences to adapt to the different network environments. The average buffer level of all solutions with Oboe traces is higher than with the UAV traces. The reason is that the average throughput of Oboe traces is 1.75 Mbps compared to 1.31 Mbps of UAV traces. In general, **I-SAC** results in 1.18-19.19% and 25.12-43.23% higher buffer levels compared to LSTM and CNN versions of SAC and A2C agents under Oboe and UAV testing datasets.

The **I-SAC** agent outputs actions corresponding to each bitrate. The bitrates are mapped into visual quality levels (i.e., $Q(J_k^q) = \log(J_k^q/J_k^1)$). Fig. 10b shows the average mapped visual quality for five different ABR algorithms. The on-policy



(a) Average buffer level (s)

(b) Average mapped visual quality

(c) Average quality oscillations

(d) Average rebuffering (s)

(e) Average latency (s)

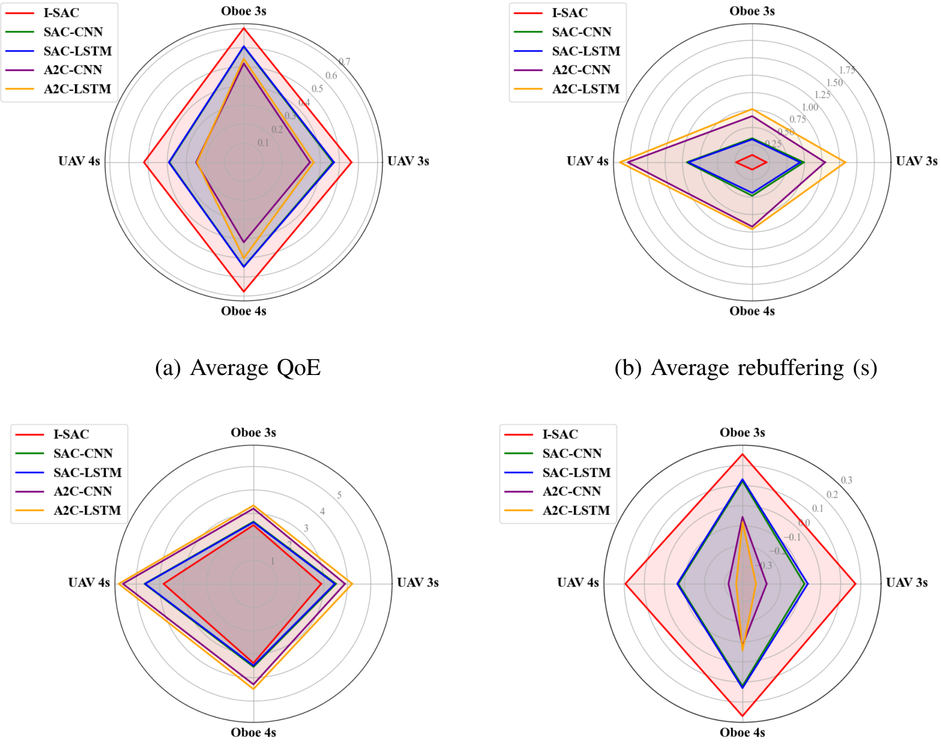
(f) Average bandwidth utilization

Fig. 10: Average E2E and playback components assessment in **I-SAC** and four other comparative algorithms under UAV and Oboe network traces.

A2C-LSTM achieves the highest mapped visual quality scores compared to other methods over all the network settings. However, the aggressive quality switch-up decisions lead to underperformance in several other quality objectives defined in our QoE model. In contrast, the **I-SAC** solution finds an optimal trade-off between different quality objectives to achieve the higher (6.83%-28.32%) learning goal (reward) and results in acceptable visual quality. Fig. 10c illustrates that **I-SAC** solution results in the lowest quality oscillations by avoiding unwanted quality switches. On the other hand, A2C-LSTM which switches to the highest visual quality leads to the highest quality oscillations. Although, both the SAC and A2C solutions were designed for UAV streaming environments and they can be best trained by applying any combination of QoE objectives. However, it is difficult for on-policy (A2C-CNN, A2C-LSTM) and off-policy (SAC-CNN, SAC-LSTM) to satisfy interruption-free streaming in a UAV streaming environment. **I-SAC** solution on the other hand performs the best and observes 0.18s and 0.11s average interruptions during the streaming sessions (Fig. 10d). The underperformance of A2C methods is due to the fact that the objective function of A2C leads to higher bitrate utility; therefore, A2Cs are unable to come up with workable solutions that can balance the performance of bitrate and rebuffering objectives.

We compare the average values of the end-to-end latency measured in terms of segment fetch time and observed that **I-SAC** always results in the lowest latency values, independent of the bandwidth traces (Fig. 10e). The average latency in **I-SAC** is 1.78s which is less than the segment duration, i.e., 2s. Whereas, the average latency in other solutions is greater than the segment duration which leads to higher rebuffering. Comparatively, **I-SAC** intelligently focuses on ensuring more balanced segment availability by making better utilization of the available network throughput. The experimental results under UAV and Oboe network traces in Fig. 10f reflect that **I-SAC** results in the highest bandwidth utilization compared with the baseline algorithms. It is noteworthy that under 1.31 Mbps and 1.75 Mbps of UAV and Oboe traces, the average bandwidth utilization in **I-SAC** implementation is always positive and highest. The other solutions in particular A2C-CNN and A2C-LSTM result in negative/over-estimation of the bandwidth. Our experiments reveal the limitations of existing solutions (i.e. A2C-CNN, A2C-LSTM, SAC-CNN, and SAC-LSTM) in optimizing the overall QoE objective in UAV-based streaming environments. The on-policy control strategies used in A2C-CNN and A2C-LSTM result in suboptimal performance on metrics such as quality oscillations, rebuffering, latency, and bandwidth utilization, despite being adaptable to distinct conditions with a lot of tuning. Similarly, the off-policy LSTM and CNN versions also have limitations, with QoE levels being irrelevant to optimal levels, particularly under low bandwidth settings. These limitations highlight the difficulty of achieving optimal QoE with existing solutions in UAV-based streaming environments. However, our proposed **I-SAC** solution, which employs a separate CNN layer for bandwidth, UAV sensory, and playback features processing, achieves better streaming performance, highlighting its effectiveness in improving the QoE for UAV-based streaming.

*3) Ablation Study: Impact of Segment Duration in UAV Streaming:* Segment duration is a crucial parameter in UAV streaming that affects the playback experience for viewers [53]. To evaluate the performance of our proposed **I-SAC**



(a) Average QoE

(b) Average rebuffering (s)

(c) Average latency (s)

(d) Average bandwidth utilization

Fig. 11: Average QoE, rebuffering, latency, and bandwidth utilization values achieved by **I-SAC** and four other comparative algorithms for segment durations of 3 and 4 seconds.

algorithm in the context of UAV streaming, we further divided the video clip into 3s and 4s segment durations, chosen based on the experimental findings of [54], [55]. Fig. 11 results demonstrate the effectiveness of the proposed **I-SAC** algorithm in improving the playback performance in a UAV-based streaming environment. The QoE values in Fig. 11a are obtained by comparing **I-SAC** against state-of-the-art DRL-based solutions, including SAC-CNN, SAC-LSTM, A2C-CNN, and A2C-LSTM, under both UAV and Oboe network traces and with segment durations of 3s and 4s. It can be observed that the QoE values for Oboe network traces are higher than for UAV network traces due to the higher average bandwidth.

The proposed **I-SAC** solution consistently outperforms the other methods in all tested scenarios. Specifically, **I-SAC** demonstrates an average improvement of 20.21% for 3s-long video segments and 33.74% for segments of 4s compared to SAC-CNN and SAC-LSTM when the UAV traces are used, respectively. Similar observations can be made for the Oboe network traces, where **I-SAC** surpasses SAC-CNN and SAC-LSTM by 15.83% (for 3s) and 23.81% (for 4s). In a similar vein, **I-SAC** displays improvements ranging from 29.61% to 35.78% (for 3s) and 34.9% to 61.72% (for 4s) over the A2C-LSTM and A2C-CNN solutions within Oboe traces. Note that with an increase in segment duration, the overall QoE decreases for all solutions. However, **I-SAC** results in almost similar QoE levels for different segment durations.

Fig. 11b compares the performance of various solutions in terms of rebuffering rate, a crucial streaming performance indicator that represents the frequency of video playback disruptions due to buffering. A lower rebuffering rate corresponds to an enhanced viewer's QoE. Our proposed solution, **I-SAC**, consistently exhibits superior performance by maintaining the lowest rebuffering rates across all tested scenarios, indicating its effective handling of playback interruptions. In contrast, the other solutions, namely SAC-CNN, SAC-LSTM, A2C-CNN, and A2C-LSTM, yield less favorable results, characterized by significantly higher rebuffering rates. Among these, the A2C-LSTM and A2C-CNN solutions specifically obtain the highest rebuffering rates in most situations. Additionally, noteworthy is that the rebuffering rate is influenced by video segment duration. Particularly, a segment duration of 3s results in a lower rebuffering rate than when segments of 4s are used, implying a smoother playback experience.

Fig. 11c depicts the average latency values, illustrating a notable correlation between longer segment durations and increased communication delays. Within the UAV environment, the **I-SAC** solution remarkably achieves latency values of 2.8s for 3s and 3.8s for 4s segment duration. Similarly, within the Oboe environment, the **I-SAC** exhibits superior performance, obtaining latency values of 2.5s and 3.38s for 3s and 4s segment durations, respectively. These latency metrics not only highlight the effective functionality of the **I-SAC**, but also underscore its capability to adapt to varying testing conditions. In contrast, A2C-LSTM and A2C-CNN exhibit the highest communication delays. The ultimate result is the provision of an optimal streaming experience characterized by the lowest latency values. Fig. 11d displays bandwidth utilization results. Here, the **I-SAC** consistently demonstrates higher and more positive bandwidth utilization compared to the other solutions under all testing datasets. This outcome reveals the **I-SAC**'s superior performance in managing network resources effectively to deliver an improved playback performance. An important consideration in these analyses is the significant impact of the chosen segment duration on the results obtained. The segment duration influences how the video content is fragmented and subsequently transmitted, which in turn affects the overall video quality experienced by the end user.

In summary, **I-SAC** is a robust solution for improving QoE during UAV-related streaming. **I-SAC** was trained offline; however, its lightweight design allows for easy online implementation and updates. The off-policy maximum entropy characteristics in **I-SAC** make it less sensitive to non-linear data variations and therefore it can adapt well under changing environments (e.g., Oboe network traces), or playback conditions (e.g., 2s, 3s, and 4s segment duration). With approximately 8000 training steps, taking less than two hours, **I-SAC** consistently outperforms state-of-the-art DRL-based solutions, achieving higher QoE values and lower rebuffering rates across numerous tested scenarios and segment durations. In general, compared to closely-related works, **I-SAC** achieves up to 13.86%. 39.31%, and 72.58% higher QoE under UAV traces, and up to 22.21%, 24.26%, and 36.06% higher QoE under Oboe traces for 2s, 3s, and 4s segment durations, respectively. Our results also highlight the importance of carefully selecting the segment duration in a UAV network environment, as longer segment durations can result in higher latency values and rebuffering rates. **I-SAC** is capable of effectively adapting to varying flying conditions, resulting in the lowest latency values and highest bandwidth utilization. Our results demonstrate a promising potential of **I-SAC** in enhancing the user experience in a UAV-based streaming environment. By successfully addressing core challenges and capitalizing on adaptive strategies, **I-SAC** sets a new benchmark in QoE-focused solutions for UAV streaming.

## VI. Conclusions and Future Works

This paper introduces I-SAC, an improved off-policy DRL solution that allows the optimal adaptation of video content based on sample-efficient exploration and maximum entropy in a UAV-based streaming environment. The adaptive bitrate allocation problem is expressed as an MDP and is addressed by the I-SAC method, taking into account the dynamic characteristics of network resources and UAV flight details. I-SAC is a straightforward and theoretically sound solution which improves QoE of UAV-based video streaming. I-SAC captures and processes environmental features, including network bandwidth, UAV flying status, and playback information, via convolution layers to maximize the long-term QoE reward, which is based on numerous important E2E and playback components. Extensive experimental outcomes based on real-world datasets demonstrate that the I-SAC algorithm significantly outperforms the closest on-policy and off-policy DRL-based solutions in terms of superior learning efficiency and QoE. Specifically, testing results show that I-SAC improves the overall video streaming QoE by up to 18.04%, 31.79%, and 54.32% for 2s, 3s, and 4s-long video segment durations, respectively.

Future work will explore the integration of meta-learning techniques with the joint optimization of computational resources and bitrate adaptation in a mobile edge computing-driven UAV streaming environment. It will also consider more diverse performance testing including hardware-in-the-loop experiments.

## References

[1] R. Zhang, M. Wang, L. X. Cai, and X. Shen, "Learning to be Proactive: Self-Regulation of UAV based Networks with UAV and User Dynamics," *IEEE Transactions on Wireless Communications*, vol. 20, no. 7, pp. 4406–4419, 2021.

[2] L. Gupta, R. Jain, and G. Vaszkun, "Survey of Important Issues in UAV Communication Networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2015.

[3] Z. Yuan, J. Jin, J. Chen, L. Sun, and G.-M. Muntean, "ComProSe: Shaping Future Public Safety Communities with ProSe-Based UAVs," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 165–171, 2017.

[4] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. G. Giordano, A. Garcia-Rodriguez, and J. Yuan, "Survey on UAV Cellular Communications: Practical Aspects, Standardization Advancements, Regulation, and Security Challenges," *IEEE Communications surveys & tutorials*, vol. 21, no. 4, pp. 3417–3442, 2019.

[5] Z. Yuan and G.-M. Muntean, "AirSlice: A Network Slicing Framework for UAV Communications," *IEEE Communications Magazine*, vol. 58, no. 11, pp. 62–68, 2020.

[6] C. Zhan, H. Hu, X. Sui, Z. Liu, J. Wang, and H. Wang, "Joint Resource Allocation and 3D Aerial Trajectory Design for Video Streaming in UAV Communication Systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 8, pp. 3227–3241, 2021.

[7] Z. Liu and Y. Jiang, "Cross-Layer Design for UAV-Based Streaming Media Transmission," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 7, pp. 4710–4723, 2022.

[8] I. M. Ozcelik and C. Ersoy, "ALVS: Adaptive Live Video Streaming using Deep Reinforcement Learning," *Journal of Network and Computer Applications*, vol. 205, p. 103451, 2022.

[9] J. Yang, Y. Jiang, and S. Wang, "Enhancement or super-resolution: Learning-based adaptive video streaming with client-side video processing," in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 739–744.

[10] L. A. binti Burhanuddin, X. Liu, Y. Deng, U. Challita, and A. Zahemszky, "QoE Optimization for Live Video Streaming in UAV-to-UAV Communications via Deep Reinforcement Learning," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 5358–5370, 2022.

[11] M. Zhang, Y. Xiong, S. X. Ng, and M. El-Hajjar, "Content-Aware Transmission in UAV-Assisted Multicast Communication," *IEEE Transactions on Wireless Communications*, 2023.

[12] J. Luo, F. R. Yu, Q. Chen, and L. Tang, "Adaptive Video Streaming with Edge Caching and Video Transcoding over Software-Defined Mobile Networks: A Deep Reinforcement Learning Approach," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 1577–1592, 2019.

[13] H. Mao, R. Netravali, and M. Alizadeh, "Neural Adaptive Video Streaming with Pensieve," in *Proceedings of the conference of the ACM special interest group on data communication*, 2017, pp. 197–210.

[14] X. Xiao, W. Wang, T. Chen, Y. Cao, T. Jiang, and Q. Zhang, "Sensor-Augmented Neural Adaptive Bitrate Video Streaming on UAVs," *IEEE Transactions on Multimedia*, vol. 22, no. 6, pp. 1567–1576, 2019.

[15] X. Jiang, Y.-H. Chiang, Y. Zhao, and Y. Ji, "Plato: Learning-based Adaptive Streaming of 360-degree Videos," in *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*. IEEE, 2018, pp. 393–400.

[16] J. Fu, Z. Chen, X. Chen, and W. Li, "Sequential Reinforced 360-degree Video Adaptive Streaming with Cross-User Attentive Network," *IEEE Transactions on Broadcasting*, vol. 67, no. 2, pp. 383–394, 2020.

[17] F. Chiariotti, S. D'Aronco, L. Toni, and P. Frossard, "Online Learning Adaptation Strategy for DASH Clients," in *Proceedings of the 7th International Conference on Multimedia Systems*, 2016, pp. 1–12.

[18] X. Wei, M. Zhou, S. Kwong, H. Yuan, and T. Xiang, "QOE-Based Neural Live Streaming Method with Continuous Dynamic Adaptive Video Quality Control," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2021, pp. 1–6.

[19] M. Naresh, N. Gireesh, P. Saxena, and M. Gupta, "SAC-ABR: Soft Actor-Critic based Deep Reinforcement Learning for Adaptive BitRate Streaming," in *2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS)*. IEEE, 2022, pp. 353–361.

[20] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[21] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," in *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE*. IEEE, 2016, pp. 1–9.

[22] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, p. 325–338, aug 2015. [Online]. Available: https://doi.org/10.1145/2829988.2787486

[23] T. Huang, C. Zhou, R.-X. Zhang, C. Wu, X. Yao, and L. Sun, "Comyco: Quality-Aware Adaptive Video Streaming via Imitation Learning," in *Proceedings of the 27th ACM International Conference on Multimedia*, ser. MM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 429–437. [Online]. Available: https://doi.org/10.1145/3343031.3351014

[24] F. Fu, Y. Kang, Z. Zhang, F. R. Yu, and T. Wu, "Soft Actor–critic DRL for Live Transcoding and Streaming in Vehicular Fog-Computing-enabled IoV," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1308–1321, 2020.

[25] L. Cui, D. Su, S. Yang, Z. Wang, and Z. Ming, "TCLiVi: Transmission Control in Live Video Streaming based on Deep Reinforcement Learning," *IEEE Transactions on Multimedia*, vol. 23, pp. 651–663, 2020.

[26] X. Ma, Q. Li, L. Zou, J. Peng, J. Zhou, J. Chai, Y. Jiang, and G.-M. Muntean, "QAVA: QoE-Aware Adaptive Video Bitrate Aggregation for HTTP Live Streaming Based on Smart Edge Computing," *IEEE Transactions on Broadcasting*, 2022.

[27] L. Meng, F. Zhang, L. Bo, H. Lu, J. Qin, and J. Han, "FastConv: Fast Learning based Adaptive Bitrate Algorithm for Video Streaming," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[28] T. Huang, R.-X. Zhang, C. Zhou, and L. Sun, "QARC: Video Quality Aware Rate Control for Real-time Video Streaming based on Deep Reinforcement Learning," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1208–1216.

[29] C. Zhan and R. Huang, "Energy Efficient Adaptive Video Streaming With Rotary-Wing UAV," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 8040–8044, 2020.

[30] Y. Chen, H. Zhang, and Y. Hu, "Optimal Power and Bandwidth Allocation for Multiuser Video Streaming in UAV Relay Networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6644–6655, 2020.

[31] I.-S. Comşa, G.-M. Muntean, and R. Trestian, "An Innovative Machine-Learning-based Scheduling Solution for Improving Live UHD Video Streaming Quality in Highly Dynamic Network Environments," *IEEE Transactions on Broadcasting*, vol. 67, no. 1, pp. 212–224, 2020.

[32] D. Wu, L. Wang, M. Liang, Y. Kang, Q. Jiao, Y. Cheng, and J. Li, "UAV-Assisted Real-Time Video Transmission for Vehicles: A Soft Actor-Critic DRL Approach," *IEEE Internet of Things Journal*, 2023.

[33] Z. Yuan, J. Jin, L. Sun, K.-W. Chin, and G.-M. Muntean, "Ultra-reliable IoT Communications with UAVs: A Swarm Use Case," *IEEE Communications Magazine*, vol. 56, no. 12, pp. 90–96, 2018.

[34] M. Naveed, S. Qazi, S. M. Atif, B. A. Khawaja, and M. Mustaqim, "SCRAS Server-based Crosslayer Rate-Adaptive Video Streaming over 4G-LTE for UAV-based Surveillance Applications," *Electronics*, vol. 8, no. 8, p. 910, 2019.

[35] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous Methods for Deep Reinforcement Learning," in *International conference on machine learning*. PMLR, 2016, pp. 1928–1937.

[36] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015, pp. 325–338.

[37] L. Amour, S. Souihi, A. Mellouk, and S. Mushtaq, "Q2ABR: QoE-aware Adaptive Video Bit Rate Solution," *International Journal of Communication Systems*, vol. 33, no. 10, p. e4204, 2020.

[38] T. Hoßfeld, M. Seufert, C. Sieber, and T. Zinner, "Assessing Effect Sizes of Influence Factors Towards a QoE Model for HTTP Adaptive Streaming," in *2014 sixth international workshop on quality of multimedia experience (qomex)*. IEEE, 2014, pp. 111–116.

[39] A. Bentaleb, A. C. Begen, and R. Zimmermann, "SDNDASH: Improving QoE of HTTP Adaptive Streaming using Software Defined Networking," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 1296–1305.

[40] M. Zhao, X. Gong, J. Liang, W. Wang, X. Que, and S. Cheng, "QoE-Driven Cross-Layer Optimization for Wireless Dynamic Adaptive Streaming of Scalable Videos Over HTTP," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 451–465, 2015.

[41] L. Wang, A. Durning, and D. T. Delaney, "ML-based Video Streaming QoE Modeling with E2E and Link Metrics," in *2022 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2022, pp. 1–8.

[42] N. Eswara, S. Ashique, A. Panchbhai, S. Chakraborty, H. P. Sethuram, K. Kuchi, A. Kumar, and S. S. Channappayya, "Streaming Video QoE Modeling and Prediction: A Long Short-Term Memory Approach," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 3, pp. 661–673, 2020.

[43] H. Nam, K.-H. Kim, and H. Schulzrinne, "QoE matters more than QoS: Why people stop watching cat videos," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.

[44] C. Xu, S. Jia, M. Wang, L. Zhong, H. Zhang, and G.-M. Muntean, "Performance-Aware Mobile Community-Based VoD Streaming Over Vehicular Ad Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 3, pp. 1201–1217, 2015.

[45] T. Huang, R.-X. Zhang, and L. Sun, "Deep Reinforced Bitrate Ladders for Adaptive Video Streaming," in *Proceedings of the 31st ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, ser. NOSSDAV '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 66–73. [Online]. Available: https://doi.org/10.1145/3458306.3458873

[46] N. Promwongsa, A. Ebrahimzadeh, D. Naboulsi, S. Kianpisheh, F. Belqasmi, R. Glitho, N. Crespi, and O. Alfandi, "A Comprehensive Survey of the Tactile Internet: State-of-the-Art and Research Directions," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 472–523, 2021.

[47] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, "Reinforcement Learning with Deep Energy-based Policies," in *International conference on machine learning*. PMLR, 2017, pp. 1352–1361.

[48] X. Gao, J. Zeng, X. Zhou, T. Qiu, and K. Li, "Soft Actor-Critic Algorithm for 360-Degree Video Streaming with Long-Term Viewport Prediction," in *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE, 2021, pp. 462–469.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An Imperative Style, High-Performance Deep Learning Library," *Advances in neural information processing systems*, vol. 32, 2019.

[51] 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 *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, ser. SIGCOMM '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 44–58. [Online]. Available: https://doi.org/10.1145/3230543.3230558

[52] Z. Duanmu, A. Rehman, and Z. Wang, "A Quality-of-Experience Database for Adaptive Video Streaming," *IEEE Transactions on Broadcasting*, vol. 64, no. 2, pp. 474–487, 2018.

[53] D. M. Nguyen, L. B. Tran, H. T. Le, N. P. Ngoc, and T. C. Thang, "An Evaluation of Segment Duration Effects in HTTP Adaptive Streaming over Mobile Networks," in *2015 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS)*. IEEE, 2015, pp. 248–253.

[54] M. Chen, "AMVSC: A Framework of Adaptive Mobile Video Streaming in the Cloud," in *2012 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2012, pp. 2042–2047.

[55] M. B. Yahia, Y. L. Louedec, G. Simon, L. Nuaymi, and X. Corbillon, "HTTP/2-based Frame Discarding for Low-Latency Adaptive Video Streaming," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 15, no. 1, pp. 1–23, 2019.

**Abid Yaqoob** (M'20) completed his B.Sc. in computer systems engineering from the Islamia University of Bahawalpur, Pakistan, in 2014, and an M.Sc. in network and information security from Northwestern Polytechnical University, Xi'an, China in 2018. He completed his PhD with the Performance Engineering Laboratory and the Insight Centre for Data Analytics, School of Electronic Engineering, Dublin City University, Ireland, in 2022. His research focuses on 5G advancements, network slicing, priority-aware differentiated video streaming, video QoE enhancement, and immersive multimedia processing and delivery solutions.

**Zhenhui Yuan** is an Assistant Professor at the University of Warwick, UK. He received his B.degree (Software Engineering) at Wuhan University, China, in 2008 and PhD (Electronic Engineering) at Dublin City University (DCU), Ireland, in 2012. His research interests lie in communications and networking of sensors, robots and vehicles. He was a postdoc researcher (2012-2014) at DCU Ireland and a 3GPP delegate (2014-2015) on 5G at Huawei, Shanghai, China. During 2015-2019, he was an Associate Professor in the Key Lab of RF Circuits and Systems (Ministry of Education) at Hangzhou Dianzi Univeristy, China. He was also a co-founder and CTO (2015-2019) at RobSense (Hangzhou) Technology Co.Ltd, a startup that designs and produces UAV flight controllers and IoT gateway. He was with Northumbria University, UK as a Senior Lecturer between 2020 and 2023He won the best paper awards at IEEE ICCRE 2016 and IEEE BMSB 2014. He served as the Guest Editor in IEEE Network and IEEE IoT-Journal, and the Associate Editor at IEEE ACCESS, IEEE Transactions on Consumer Electronics and Pervasive Computing & Communications journal. He is a member of IEEE P1954 on UAV communications.

**Gabriel-Miro Muntean** (M'04, SM'17 and F'23) is a Professor with the School of Electronic Engineering, Dublin City University (DCU), Ireland, and Co-Director of the DCU Performance Engineering Lab. Prof. Muntean was awarded the Ph.D. degree by DCU for research on adaptive multimedia delivery in 2004. He has published over 500 papers in top-level international journals and conferences, authored four books and 29 book chapters, and edited six additional books. His research interests include quality, performance, and energy-saving issues related to multimedia and multiple sensorial media delivery, technology-enhanced learning, and other data communications over heterogeneous networks. Prof. Muntean is an Associate Editor of the IEEE Transactions on Broadcasting, the Multimedia Communications Area Editor of the IEEE Communications Surveys and Tutorials, and chair and reviewer for important international journals, conferences, and funding agencies. Prof. Muntean coordinated the EU project NEWTON and led the DCU team in the EU projects TRACTION and HEAT and other Irish large projects such as eStream and FRADIS.