FReD-ViQ: Fuzzy Reinforcement Learning Driven Adaptive Streaming Solution for Improved Video Quality of Experience

Abid Yaqoob, *Student Member, IEEE*, Gabriel-Miro Muntean, *Fellow, IEEE*

***Abstract*—Next-generation cellular networks strive to offer ubiquitous connectivity, enhanced transmission rates with increased capacity, and superior network coverage. However, they face significant challenges due to the growing demand for multimedia services across diverse devices. Adaptive multimedia streaming services are essential for achieving good viewer Quality of Experience (QoE) levels amidst these challenges. Yet, the existing adaptive video streaming solutions do not consider diverse QoE preferences or are limited to meeting specific QoE objectives. This paper presents FReD-ViQ, a Fuzzy Reinforcement Learning-Driven Adaptive Streaming Solution for Improved Video QoE that combines the strengths of fuzzy logic and advanced Deep Reinforcement Learning (DRL) mechanisms to deliver exceptional, individually tailored user experiences. FReD-ViQ is a sophisticated streaming solution that leverages efficient membership function modelling to achieve a more finely-grained representation of both input and output spaces. This advanced representation is augmented by a set of fuzzy rules that govern the decision-making process. In addition to its fuzzy logic capabilities, FReD-ViQ incorporates a novel DRL algorithm based on Dueling Double Deep Q-Network (Dueling DDQN), noisy networks, and prioritized experience replay (PER) techniques. This innovative fusion enables effective modelling of uncertain network dynamics and high-dimensional state spaces while optimizing exploration-exploitation trade-offs in adaptive streaming environments. Extensive performance evaluations in real-world simulation settings demonstrate that FReD-ViQ effectively surpasses existing solutions across multiple QoE models, yielding average improvements of 23.10% (Linear QoE), 23.97% (Log QoE), and 33.42% (HD QoE).**

***Index terms*— Fuzzy logic, Deep reinforcement learning, MPEG-DASH, Adaptive video streaming, QoE**

I. Introduction

Adaptive video streaming has revolutionized the way users access multimedia content, delivering an optimal viewing experience by dynamically adjusting video quality based on network conditions and device capabilities. This process involves encoding each video file into multiple representations, allowing the MPEG-DASH [1] streaming client to switch between them, ensuring the most suitable video quality. The use of multiple representations, coupled with an adaptive quality switching algorithm, optimizes the user's Quality of Experience (QoE) for traditional [2] or immersive videos [3]–[5]. However, challenges persist in representation selection and achieving high perceived quality, which may hinder a user-friendly seamless streaming performance.

Recently, several Adaptive BitRate (ABR) solutions have been proposed to tackle the challenges of dynamic streaming environments, including heuristic-based [6]–[12], fuzzy-based [13]–[17], and reinforcement learning (RL) [18]–[28] approaches. The overall streaming process for these models is illustrated in Fig. 1, utilizing various techniques and decision variables to enhance bitrate selection. Although these solutions have made progress in improving end-user QoE, they still face challenges in delivering an optimal user experience. Heuristic-based methods, for instance, often suffer from a lack of flexibility due to their reliance on a fixed set of rules. The non-stationary nature of networks can cause abrupt changes in network conditions, which may negatively impact the performance of heuristic-based ABR algorithms [29]. Conversely, fuzzy-based approaches are characterized by intricate rule development and decision-making processes that may struggle to handle unexpected changes in the ABR environment. This becomes especially problematic in mobile networks, where users frequently transition between cells, leading to constant fluctuations in network conditions [30]. Deep Reinforcement Learning (DRL) models, on the other hand, hold significant promise for addressing the complex and dynamic nature of communication environments. However, the high dimensionality of the state space, which includes

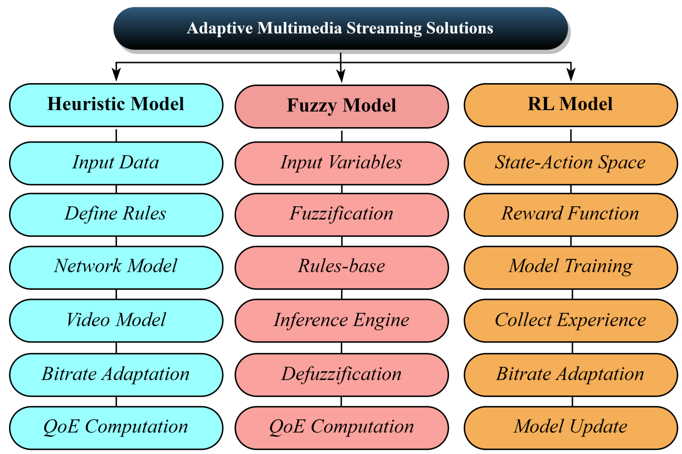


Fig. 1: Generalized workflow of heuristic-based, fuzzy-based, and RL-based adaptive streaming models.

A. Yaqoob is with the Performance Engineering Lab and Insight SFI Centre for Data Analytics, School of Electronic Engineering, Dublin City University, Ireland (e-mail: abid.yaqoob2@mail.dcu.ie).

G.-M. Muntean is with the Performance Engineering Lab, School of Electronic Engineering, Dublin City University (e-mail: gabriel.muntean@dcu.ie).

This work is funded by the Science Foundation Ireland (SFI) grants 12/RC/2289_P2 (INSIGHT) and 21/FFP-P/10244 (FRADIS).

factors such as video quality, segment sizes, buffer levels, network bandwidth, and delay, among others, complicates the learning of an optimal policy, potentially leading to a slow and inefficient learning process [31], [32]. As a result, these methods exhibit poor training efficiency and heightened sensitivity to hyperparameters, requiring a larger number of samples for each gradient step update. Additionally, striking a balance between exploring the state space and exploiting existing knowledge presents a complex trade-off in uncertain and dynamic environments [33], [34].

Achieving high QoE is crucial for ensuring user engagement and satisfaction in the competitive video streaming market. Users may have varying reactions to streaming issues, with some being more sensitive to rebuffering and others more impacted by low video quality [35]. As a result, enhancing QoE has become a paramount priority in the adaptive video streaming domain. Existing solutions perform relatively well for standard QoE models, such as *Linear QoE* [9], *Log QoE* [12], *HD QoE* [19], or *Video Multimethod Assessment Fusion (VMAF) QoE* [18], under predetermined weights. However, these solutions can be highly sensitive to changes in QoE weight coefficients, which determine the relative importance of factors such as video quality, buffering events, and playback smoothness [36]. As a result, their effectiveness may be limited to specific scenarios or QoE objectives. Therefore, by addressing challenges in representation selection, balancing trade-offs between video quality and network conditions, and developing low-complex and sample-efficient algorithms, it is possible to meet the ever-evolving challenging user expectations.

In this paper, we present a novel approach called ***F**uzzy **Re**inforcement Learning **D**riven Adaptive Streaming Solution for Improved **Vi**deo **Q**oE (**FReD-ViQ**)*, which combines advantages of both fuzzy logic [37] and advanced DRL mechanisms to address the key challenges associated with adaptive streaming. First, fuzzy logic effectively models uncertain and imprecise network dynamics while handling the high dimensionality of the state space, which often hinders learning-based algorithms. Secondly, our proposed off-policy DRL algorithm extends the conventional Double Deep Q-Network (DDQN) [38] by incorporating a Dueling architecture [39], adaptive noise injection [40], and a prioritized sampling strategy using prioritized experience replay (PER) [41]. These enhancements enable a compact representation of acquired experience, efficient knowledge utilization, improved exploration, and faster convergence. Finally, FReD-ViQ expertly balances exploration and exploitation by ensuring rapid adaptation to a non-stationary environment. Its application results in improved QoE performance measured in terms of the instantaneous visual quality of each segment, quality variations across video segments, and frequency of freezing or rebuffering events. The major contributions of this work are as follows:

1) **Enhanced Fuzzy Logic Decision-Making:** We introduce an advanced FReD-ViQ fuzzy logic controller which enables efficient modelling of membership functions for a more granular representation of the input and output spaces. The proposed controller includes 49 fuzzy rules that govern the decision-making process. By utilizing fuzzy logic to process past bitrate and bandwidth information, the proposed FReD-ViQ solution can determine the next optimal bitrate with higher accuracy. This improved decision-making process, augmented by DRL advanced techniques, helps to overcome the rigidity of existing solutions by seamlessly integrating the adaptability of fuzzy logic with the advanced learning capabilities of DRL.
2) **Innovative DRL Algorithm for Adaptive Streaming:** We propose an innovative DRL algorithm that utilizes a Dueling Double Deep Q-Network (Dueling DDQN), noisy networks, and prioritized experience replay techniques to improve the efficiency and effectiveness of reinforcement learning models. By separately estimating state values and state-dependent action advantages, the Dueling DDQN architecture facilitates faster learning of the optimal action-value function. The noisy networks technique injects adaptive noise into the network weights to encourage exploration and achieve better training results. Finally, the PER method further optimizes our algorithm by prioritizing experiences with higher learning potential, thereby improving learning efficiency.
3) **Improved Exploration-Exploitation Trade-off:** We balance the exploration-exploitation trade-off in an uncertain and dynamic streaming environment by leveraging the adaptive nature of fuzzy logic and the advanced exploration techniques offered by the advanced DRL framework. This combination allows for better handling of abrupt changes in network conditions and helps in dynamic adaptive streaming environments. The systematic fusion of these techniques allows the model to discover optimal policies while efficiently utilizing existing knowledge, improving the decision-making process.

This article is organized as follows: Section II provides an overview of the most recent related works. Section III presents system modelling and problem formulation, outlining its key objectives and constraints. Section IV describes the proposed FReD-ViQ architecture and the proposed solution's major algorithms. Section V describes the experimental setup and performance analysis of the experimental testing results. Finally, Section VI provides the conclusions and future research directions.

II. Related Works

Multimedia streaming has seen significant advancements, with the introduction of various approaches aimed at improving the end-user QoE. This section covers a comprehensive overview of the state-of-the-art three major categories of ABR approaches: (i) Heuristic-based; (ii) Fuzzy-based; and (iii) Reinforcement Learning-based. The strengths and key contributions of the closest approaches are highlighted, and a comparison with our proposed approach is also made to demonstrate its superiority and unique features.

A. Heuristic-based ABR Approaches

Heuristic-based ABR approaches have gained prominence as a means to deliver a seamless and uninterrupted video streaming experience. These solutions utilize mathematical models to provide a seamless viewing experience with minimal disruptions. Huang et al. [8] employed a set of buffer rules to determine the bitrate for the upcoming segment, with the goal of maintaining a stable buffer space. The approach, referred to as BB (buffer-based) adaptation, operates independently of throughput measurements and switches to the highest available bitrate when the buffer exceeds 15s.

Spiteri et al. proposed BOLA [9], a cutting-edge buffer-based adaptation algorithm that leverages Lyapunov optimization to enhance video quality and minimize rebuffering occurrences. Unlike traditional algorithms that emphasize bandwidth measurements, BOLA prioritizes video quality and rebuffering reduction. This algorithm is widely utilized in Bilibili[1] as a streaming solution. The BOLA client adopts a greedy approach to occupy network bandwidth, resulting in near-optimal performance and, in many instances, significantly superior results compared to conventional algorithms. Jiang et al. [10] aimed to achieve a systematic balance between fairness, efficiency, and stability in HTTP streaming through their proposed solution, FESTIVE. Unlike other solutions that prioritize QoE, FESTIVE utilizes fairness, efficiency, and stability metrics to guarantee reliable video adaptation streaming to multiple clients. This approach provides a robust solution that ensures equitable distribution of resources and stable streaming performance. Different from BB, BOLA, and FESTIVE, De Cicco et al. [11] introduced ELASTIC, a solution that generates sustained TCP flows in DASH through the implementation of feedback control theory. ELASTIC integrates both throughput and buffer levels in its control mechanism, resulting in the convergence of buffer occupancy to a specified level. Similarly, Yin et al. [12] introduced a model predictive controller (MPC) that optimally blends throughput and buffer occupancy feedback signals to maximize QoE. The QoE metric takes into account several factors including video quality, fluctuations in quality, rebuffering events, and startup delay.

Heuristic-based ABR solutions [8]–[12] have proven to be successful in delivering an improved multimedia experience in a controlled testing environment. However, these solutions have limitations, especially related to sensitivity of controller- or rule-based approaches to long-term network bandwidth dynamics. Moreover, historical data and resource-intensive computations can lead to a lack of adaptability across different devices and conditions. This can result in subpar QoE levels in real-world internet conditions, with issues such as low video quality, frequent rebuffering, and inconsistent quality levels.

B. *Fuzzy-based ABR Streaming*

Fuzzy-based ABR solutions offer a distinct perspective in video streaming optimization when compared to traditional heuristic-based approaches. The integration of fuzzy logic algorithms enables a more flexible and intelligent adjustment of video bitrates in response to network variability. Hou et al. [13] proposed a fuzzy logic solution to overcome the video streaming challenges in mobile networks. The proposed controller takes into account normalized throughput, buffer level, and buffer variations to mitigate the impact of limited bitrate levels and maintain a stable system. The performance of the controller was rigorously evaluated under a range of scenarios, including slow, rapid, and sudden changes in throughput, and under real LTE conditions using recorded traces. Rahman et al. [14] introduced a buffer- and segment-aware fuzzy logic approach to dynamically adjust the video quality in real-time for multiple streaming clients in the MPEG-DASH system. The algorithm takes into account several key factors, including segment duration, playback buffer length and buffer difference, as well as estimated throughput, to make informed decisions on the selection of video bitrates for the next segments. To further optimize the selection process, the authors incorporated a bitrate switching minimization algorithm, to refine the bitrate selection determined by fuzzy logic. The proposed solution based on 26 fuzzy rules exhibits significant improvement in both video bitrate and bandwidth efficiency, although there is space for further refinement to minimize bitrate switches.

Mowafi et al. [15] proposed an energy-efficient variant of the fuzzy-based DASH [42] adaptation algorithm which is based on the same metrics as FDASH and includes power as an additional metric. The proposed solution aims to extend the playback time of a video while maintaining a decent quality level and avoiding abrupt changes in video bitrate. The fuzzy logic controller considers the buffering time, the differential buffering time, and the available device power as inputs and produces an increase/decrease/keep the same bitrate as output. However, the authors did not consider the throughput measurements in their decision-making process, which could result in fetching wrong bitrates. Kim et al. [16] proposed a modified FDASH (mFDASH) algorithm by incorporating history-based TCP throughput estimation (HBTTE) [43], a segment bitrate filtering module (SBFM), and a start and sleep mechanism. The ranges of the input membership functions in mFDASH were adjusted to more reasonable values to enhance the selection of the next segment's bitrate. The evaluation results reveal that the proposed mFDASH algorithm effectively manages the buffer, addressing the challenge of overflows and delivering a superior QoE in a variety of network environments. Li et al. [17] presented a fuzzy controller chunked transfer-encoding (FCTE) solution for low-latency live ABR streaming. The proposed solution begins with the prediction of the mean and standard deviation of the throughput, followed by the filtering of the correct chunk transmission duration from the arrival time of the video chunk. The fuzzy logic controller then incorporates the buffer occupancy and normalized throughput metrics to determine an aggressive factor that guides bitrate selection. The aggressive factor is derived through the processes of fuzzification, fuzzy engine, and defuzzification in the FCTE. This aggressive factor and the mean throughput measurements are used to further refine the selection of the next segment bitrate.

Fuzzy logic-based ABR approaches continue to be a popular choice for adaptive bitrate selection due to their capacity to handle the complexity and uncertainty of real-world networks. Most of the existing solutions [13], [15], [42] are effective in controlled environments. To optimize the performance of these solutions, it is crucial to properly model the membership func-

[1]https://www.bilibili.tv/en

tions and fuzzy rules, and select the most relevant streaming features. This can help to mitigate some of the limitations such as sensitivity to input data, challenges in tuning, risk of overfitting, and limited prediction accuracy.

C. Learning-based ABR Streaming

Reinforcement Learning-based ABR approaches support real-time, more personalized, and advanced dynamic decision-making based on both network conditions and user experiences. For instance, Mao et al. [19] introduced an RL-based adaptive video streaming solution called Pensieve, which continuously learns through interaction with the streaming environment. The observation state in Pensieve takes into account various factors such as past chunk throughput, buffer size, and download time, among others. The action space of the solution consists of different bitrate options for the next video chunk. The action space includes different bitrates for the next video chunk. Huang et al. [21] presented a joint solution for video quality prediction and learning-based quality adaptation called QARC. This solution consists of two parallel components, Video Quality Prediction Network (VQPN) and Video Quality Reinforcement Learning (VQRL). The reward signal used by the algorithm is based on the QoE metric, which evaluates the performance of the video streaming service in terms of video quality, bitrate, and delay. In another work [18], the authors presented the Comyco solution, which leverages imitation learning to enhance the performance of learning-based methods. The system consists of a neural network, which is trained using expert policies provided by an instant solver. Comyco employs a 1D-CNN and GRU layer architecture and includes an experience replay buffer to store expert policies and train the neural network using a customized loss function. However, the implementation of Comyco restricts the ability of the agent to explore its environment.

Shi et al. [22] proposed a solution for adaptive video streaming at the edge, called Learning-based Fuzzy Bitrate Matching (LFBM), which leverages the capacity of both network and edge servers to intelligently interact with the ABR environment. LFBM architecture involves interaction between several components, including the cache manager, client information collector, network information collector, and RL agent. The client and network information collectors provide information on user states and network conditions to the RL agent, which makes decisions on bitrate selection and chunk retrieval either from the cache or the original server. The experimental results showed that the LFBM architecture achieves higher QoE and cache hit ratios. However, the use of an on-policy A3C algorithm for training the RL agent in the LFBM architecture can result in longer training times and increased computational resources. Gadaleta et al. [23] utilized a combination of deep learning and reinforcement learning to optimize the linear reward function in DASH streaming. By taking the raw system state as input, the proposed approach uses a learning architecture that includes twin neural networks and a replay memory to simulate the network environment with greater realism. The model's design eliminates arbitrary choices that could impact performance and effectively handle very large state spaces.

Ma et al. [24] presented a QoE-aware Adaptive Video Bitrate Aggregation (QAVA) scheme for multi-user live streaming, which utilizes edge computing technology. QAVA is deployed at the central smart edge and is responsible for aggregating all the traffic requests from clients for the same live-streaming service. The bitrates of these requests are adapted using a controlled DRL policy, which takes into account network conditions, client states, and video characteristics. However, implementing QAVA requires overcoming the challenges posed by variations in network conditions, diverse client behaviors and characteristics, and the difficulty in controlling client QoE. Yuan et al. [25] introduced an ensemble learning-based ABR framework for DASH clients. The framework is designed to take advantage of multiple ABR methods, including a rate-based method [44], a proportion differentiation (PD) controller-based method [45], and an online learning-based method [46]. The proposed framework adaptively selects the method that provides the highest QoE through the decision support of a method controller. The method controller decides between instant method switching and intermittent method switching, providing a simple yet effective solution for improving QoE in DASH streaming.

On-policy and off-policy learning based ABR solutions often face sample inefficiency and hyperparameter sensitivity in highly dynamic adaptive streaming environments. Solutions like Comyco [18], QAVA [24], and QARC [21] employ computationally expensive function approximators and require extensive interaction with the environment to learn effective policies. These solutions can be successful with a limited set of fine-tuned weights for specific QoE models. In contrast, this work presents a novel fuzzy-assisted DRL framework that achieves a stable and sample-efficient exploration-exploitation process to discover optimal bitrate strategies in noisy, real-world streaming scenarios.

III. FReD-ViQ System Modelling and Problem Formulation

Fig. 2 depicts the end-to-end streaming architecture of MPEG-DASH-based FReD-ViQ clients, where video content is temporally segmented into small units of data, known as segments, denoted by $V = \{V_1,\ V_2, .., \ V_k,\ ..,\ V_K\}$. Each segment is encoded into multiple representations of different bitrates, i.e., $J = \{j_k^1,\ j_k^2, ..,\ j_k^q,\ ..,\ j_k^Q\}$. The FReD-ViQ clients continuously monitor and capture the environment states to predict the most appropriate bitrate, i.e., j_k^q, for the next segments. The clients request the next segment over an HTTP persistent cellular or Wi-Fi connection after completely downloading the previous segment. Upon receiving the new segment, FReD-ViQ clients decode the compressed data and pass it to the media player. The integrated media player within each device, i.e., smartphone, laptop, and monitor, is designed to render segments proficiently, catering to high-resolution multimedia streaming requirements. The advanced display configurations across the entire range of devices guarantee exceptional display quality.

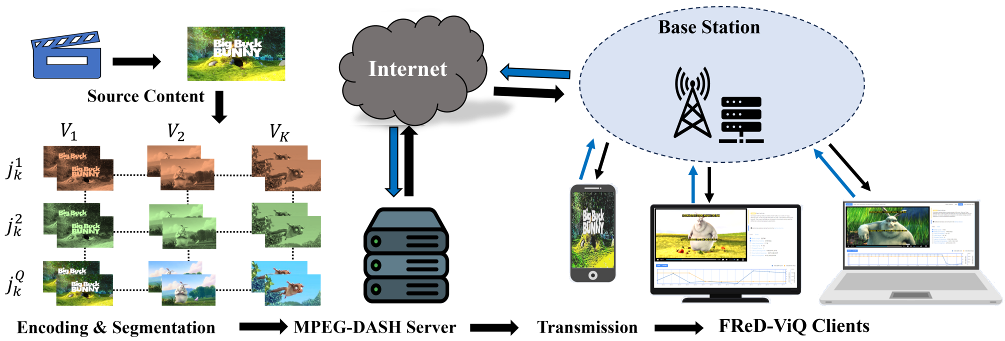


Fig. 2: End-to-end streaming architecture featuring a smart MPEG-DASH server and high-resolution FReD-ViQ clients, with DASH segments transmitted over cellular or Wi-Fi networks.

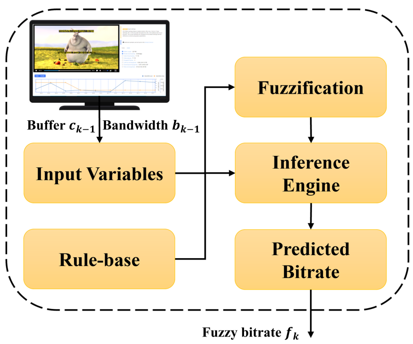


Fig. 3: FReD-ViQ: Fuzzy logic controller-based next bitrate prediction model.

A. Problem Formulation

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 Markov Decision Process (MDP) that represents an agent-environment interaction can be described using a tuple, i.e., $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ and $\mathcal{A}$ represent the continuous state and action space, $\mathcal{P}$ is the state transition probability, and $\mathcal{R}$ is the returned reward. To ensure a comprehensive understanding of the environment, the agent needs to effectively explore the state space $\mathcal{S}$. In the FReD-ViQ solution, a state $s_k \in \mathcal{S}$ to select the bitrate for kth segment is represented as a tuple, and is defined as follows:

$$s_k = (F_k, J_k, C_k, B_k, D_k, N_k, M_k) \tag{1}$$

where $F_k = \{f_{k-p}, ..., f_{k-1}, f_k\}$ is a vector which represents the fuzzy bitrates (the outputs from FReD-ViQ fuzzy model, described in Section IV-A) and p represents the number of past samples. $J_k = \{j^q_{k-p-1}, ..., j^q_{k-2}, j^q_{k-1}\}$ is a vector representing video bitrates of the previously selected segments. $C_k = \{c_{k-p-1}, ..., c_{k-2}, c_{k-1}\}$ and $B_k = \{b_{k-p-1}, ..., b_{k-2}, b_{k-1}\}$ represent the buffer occupancy and bandwidth vectors after downloading $(k-1)$th segment. The download time vector is represented as $D_k = \{d_{k-p-1}, ..., d_{k-2}, d_{k-1}\}$, wherease $N_k = \{n_{k-p-1}, ..., n_{k-2}, n_{k-1}\}$, signifies the vector of number of remaining segments. Lastly, $M_k = \{m^1_k, ..., m^{Q-1}_k, m^Q_k\}$ corresponds to the sizes of the video segments. After processing the state, the FReD-ViQ client selects an action $a_k \in A$ which corresponds to the selected video bitrate level for the kth video segment. The agent acts to transform the environment, resulting in a state transition probability $\mathcal{P}(s_{k+1}|s_k, a_k)$ from s_k to s_{k+1} under action a_k. When segment k is completely downloaded, the agent determines the bitrate for $(k+1)$th segment, based on the observed state s_{k+1}.

In a client-centric end-to-end HTTP adaptive streaming architecture, the observed reward function $r_k \in \mathcal{R}$ for a given state-action pair for segment k can be expressed in terms of QoE metric, which considers factors such as visual quality, playback smoothness, and the video quality stability.

$$QoE_k = w_1 \cdot Q(j^q_k) - w_2 \cdot \mathcal{T}_k - w_3 \cdot |(Q(j^q_k) - Q(j^q_{k-1})| \tag{2}$$

where w_1, w_2, and w_3 parameters are QoE weight coefficients and reflect the relative importance of video quality, rebuffering, and quality variations. $Q(j^q_k)$ maps the qth bitrate of kth segment to the quality perceived by the user, $\mathcal{T}_k$ is the amount of time the client waits for the playback to resume, and the last term reflects the quality variations between two consecutive segments. We employed classic representations of perceived video quality $Q(j^q_k)$ by using: (i) *Linear QoE model:* which uses the bitrate as the perceived quality j^q_k; (ii) *Log QoE model:* which is represented as the log of the ratio of the selected bitrate to the minimum bitrate $log(j^q_k/j^1_k)$; and (iii) *HD QoE model:* which uses the HD quality weights for requested bitrates.

The ultimate goal of an adaptive client is to continuously select optimal bitrates during each adaptation interval, thus maximising the aggregated QoE of all video segments. Therefore, the optimization problem in our case can be expressed mathematically as follows:

Problem P1:

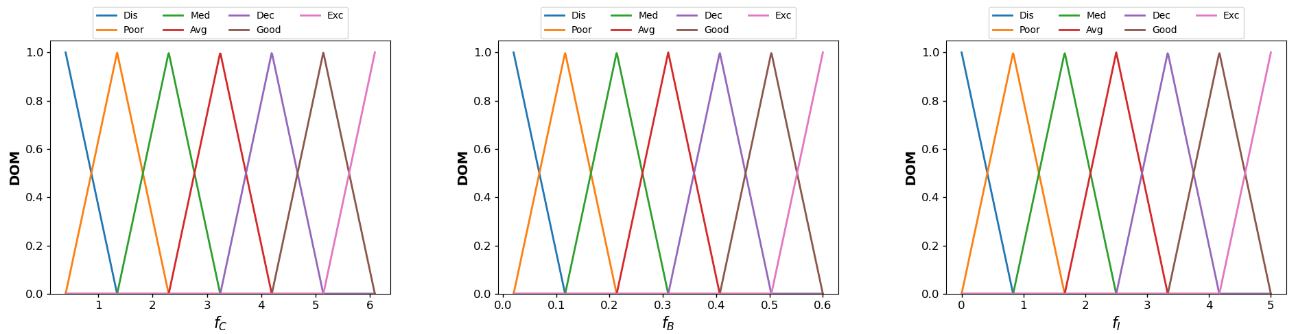
$$max \sum_{k \in K} QoE_k \tag{3}$$

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

Problem P2:

$$\max_{\pi} \mathbb{E}\left[\sum_{k \in K} \gamma^k \cdot r_k \middle| \pi\right] \tag{4}$$

where π is the policy that maps states to actions, K accounts for the number of segments in the streaming session, and



(a) Input: Buffer Ocuupancy (b) Input: Bandwidth (c) Output: Bitrate

Fig. 4: FReD-ViQ membership functions modelling buffer occupancy f_C, bandwidth f_B, and bitrate f_J.

$0 < \gamma \leq 1$ is the discount factor for future rewards. In order to optimize the long-term QoE, we consider an advanced fuzzy DRL framework in which the proposed FReD-ViQ agent learns an optimum course of action (i.e., the best-fit bitrate) by continuously interacting with the environment during each adaptation interval.

IV. FReD-ViQ Proposed Architecture and Algorithms

This section explores the core components of FReD-ViQ adaptive streaming solution. Specifically, it examines the fuzzy logic-based adaptation model, advanced neural network architecture, and the active training mechanism designed to optimize ABR performance efficiently.

A. FReD-ViQ Fuzzy Logic-based Adaptation Model

A fuzzy-based controller is a highly effective control system that is well-suited to control the adaptive streaming process. By leveraging the power of fuzzy logic techniques, this controller is capable of incorporating the most important features and expert experiences into its decision-making process, resulting in improved system performance. Fig. 3 depicts the integrated design of the FReD-ViQ fuzzy logic controller based on the Mamdani model. Comprised of four main components, i.e., fuzzification, rule base, inference engine, and bitrate prediction modules, FReD-ViQ leverages network and buffer input data to dynamically select the video bitrate for each adaptation interval. The inputs are passed through a fuzzifier, which maps them to corresponding fuzzy values using established membership functions determined by expert knowledge of relevant fuzzy sets. The inference engine then applies fuzzy rules to map these inputs to a fuzzy output. Finally, the bitrate prediction module transforms this fuzzy output into a practical bitrate decision. By employing this entire process, FReD-ViQ creates a robust control mechanism that can effectively manage uncertainties that arise in multimedia streaming systems, resulting in high-QoE levels for users.

1) Fuzzy Membership Functions Modelling: Before making a bitrate selection decision, it is highly necessary to process and analyze inputs in a way that can account for the inherent uncertainties in the streaming system. In the FReD-ViQ solution, the buffer and bandwidth variables are defined as "Antecedent" variables, which represent the normalized input values to the system. The bitrate variable is defined as a "Consequent" variable, which represents the output variable of the system. Let f_C, f_B, and f_J represent the buffer occupancy, bandwidth, and bitrate variables, respectively. The membership functions are created for these variables. Let M_C, M_B, and M_J represent the lists of linguistic labels for f_C, f_B, and f_J, respectively. We employed seven linguistic variables: Dismal (Dis), Poor, Mediocre (Med), Average (Avg), Decent (Dec), Good, and Excellent (Exc), and are represented as follows:

$$M_i = [\text{"Dis"}, \text{"Poor"}, \text{"Med"}, \text{"Avg"}, \text{"Dec"}, \text{"Good"}, \text{"Exc"}] \tag{5}$$

$\forall i \in M_C, M_B$, or M_J. By using multiple membership functions with different boundary points as shown in Fig. 4, FReD-ViQ can better capture the peculiarities and complexities of the data, leading to more accurate and reliable output results. A higher level of granularity is especially useful when the input data is highly variable or uncertain. The buffer membership functions in Fig. 4a divide the input range of the buffer variable, which goes from 0.4 to 6.0, into fuzzy subsets that represent different degrees of membership (DOM). The membership functions are triangular, with the base of the triangle representing the range of the fuzzy subset and the peak representing the maximum degree of membership within that range. As the buffer level approaches "Poor", the system may take measures to mitigate the risk of playback interruption, such as compromising video quality to ensure uninterrupted playback. Conversely, as the buffer level increases and approaches beyond the "Dec" level, the system can increase video quality to enhance the user experience.

Fig. 4a depicts the bandwidth levels normalized between 0.02 and 0.6. The adaptation algorithms attempt to optimize bandwidth utilization by selecting the highest available video bitrate, which is less than the available connection speed [47]. This approach ensures that the available bandwidth is utilized efficiently, enabling a seamless streaming experience for the end users. When the available bandwidth is "Good" or "Exc", FReD-ViQ places more emphasis on increasing the quality. On the other hand, when the bandwidth drops, a conservative bitrate adaptation is employed in order to download the segment quickly. Fig. 4c depicts the seven membership functions for the bitrate variable f_J, which is a

Algorithm 1: FReD-ViQ: Fuzzy Rule Creation Algorithm

Input : $f_C, f_B, f_J \leftarrow$ Fuzzy variables for buffer occupancy, bandwidth, and bitrate
$M_C, M_B, M_J \leftarrow$ Fuzzy membership functions for buffer occupancy, bandwidth, and bitrate
Result : $f_r \leftarrow$ Fuzzy rules

$f_r \leftarrow []$;
for i *in range(len*(M_C)*)* **do**
  for j *in range(len*(M_B)*)* **do**

$$f_{rule} \leftarrow \left\{ \begin{array}{ll} Rule(f_C[M_C[i]] \& f_B[M_B[j]], f_J[M_J[0]]) & \textbf{if } i \leq 1 \\ Rule(f_C[M_C[i]] \& f_B[M_B[j]], f_J[M_J[min(i, 2 \times j)]]) & \textbf{else} \end{array} \right\}$$

    f_r.append(f_{rule});

consequential variable in multimedia streaming systems. The bitrate is dependent on the action dimension of the system, which in this case ranges from 0 to 5. The predicted output of the FReD-ViQ system is rounded to determine the final bitrate for the next segment, i.e., f_k. This approach facilitates the modelling of uncertain relationships between variables and promotes informed decision-making based on the available information.

2) Fuzzy Rules Creation: Having mapped the system information to the linguistic variables, the FReD-ViQ controller then takes advantage of the advanced fuzzy rules defined in Algorithm 1 in order to determine the next video bitrate. Algorithm 1 uses nested for-loops to create a set of rules that relate the membership functions of the buffer and bandwidth inputs to the membership functions of the output bitrate. The *Rule* method within the SKFuzzy[2] library facilitates defining the fuzzy rules by mapping the degrees of membership for buffer and bandwidth inputs to the corresponding degree of membership for the output bitrate. The *if-else* statement within the loop checks whether the buffer membership function is "Dis" or "Poor", and assigns the output bitrate membership function to "Dis" if so. Otherwise, the output bitrate membership function is assigned based on the minimum degree of membership between the buffer and bandwidth membership functions, i.e., $min(i, 2 \times j)$[3]. It could lead to overly sensitive output bitrates if the output bitrate membership function is solely determined by the buffer or bandwidth membership function. By using $min(i, 2 \times j)$, FReD-ViQ strikes a balance between these two extremes and ensures that the output bitrate is influenced by both inputs in a proportional and balanced way. The resulting fuzzy rules are stored in the fuzzy rules list, i.e., f_r. This rule creation process is streamlined by the SKFuzzy library. It facilitates the development of effective fuzzy-based ABR approaches, offering customizable graded membership functions and flexible rule definitions that are readily adaptable to meet the conflicting objectives in ABR streaming environments. Moreover, SKFuzzy is well-suited to model complex and nonlinear relationships between different playback factors.

[2]https://pypi.org/project/scikit-fuzzy/
[3]The i and j indices navigating through the fuzzy membership functions allows to process all combinations of buffer occupancy and bandwidth values.

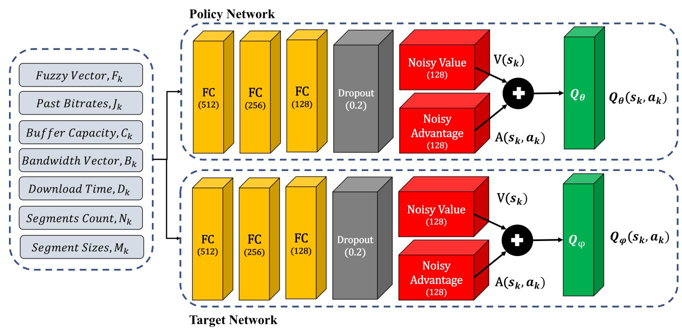


Fig. 5: FReD-ViQ client's neural network architecture.

B. FReD-ViQ Neural Network Architecture

The primary goal of FReD-ViQ is to effectively utilize network resources and playback information to optimize bitrate selection during each adaptation interval. FReD-ViQ features a unique neural network architecture, as depicted in Fig. 5. This architecture employs a series of states based on the feedback received during previous adaptive intervals, enabling the generation of adaptive bitrate selection rules. The FReD-ViQ neural network architecture comprises a Dueling DDQN model, which includes policy and target networks. Each of these networks receives various inputs such as fuzzy outputs F_k, past bitrates J_k, buffer occupancy C_k, bandwidth vector B_k, download time D_k, segments count N_k, and segment sizes M_k. A tailored strategy extracts relevant features for each type of input, ensuring efficient and robust decision-making. The feature extraction component within the FReD-ViQ architecture includes linear layers with 512, 256, and 128 hidden units, followed by ReLU activation functions. After feature extraction, the processed data is passed through a dropout layer with a dropout rate of 0.2, which forms the basis for the value and advantage streams. The value stream estimates the value of being in a given state, while the advantage stream estimates the advantage of taking each action in that state. These streams utilize noisy linear layers, injecting noise into the weights and biases during training to encourage exploration. This approach is particularly valuable in time-sensitive streaming applications, where accurate exploration is essential for optimizing bitrate selections under various network conditions. The resulting Q-values are computed by combining the value $V(s_k)$ and advantage $A(s_k, a_k)$ estimates, with the mean advantage subtracted to stabilize learning. This architecture combines the strengths of the Dueling DDQN and noisy networks learning frameworks, where the Dueling DDQN framework allows the network to learn which actions are valuable in each state, while the noisy networks encourage the network to explore more effectively. This combination leads to a more robust and effective bitrate selection policy, ultimately improving the overall video streaming experience.

C. FReD-ViQ Training Mechanism

Algorithm 2 presents the training mechanism of the FReD-ViQ agent which includes a Dueling DDQN combined with noisy networks and prioritized experience replay. In the training procedure, we first initialize the policy network $Q_\theta(s,a)$ with parameter θ. The policy network is responsible for estimating the Q-values for each state-action pair and is updated during the training process. We initialize the target network $Q_\phi(s,a)$ with parameter ϕ. Initially, the target network parameters ϕ are set equal to the policy network parameters θ. The target network is used to compute target Q-values for updating the policy network and is updated periodically during training. Next, we initialize the agent's buffer $\mathcal{B}$ with capacity N. The buffer is used to store the agent's experiences in the form of transitions $(s_k, a_k, r_k, s_{k+1}, d_k)$, which are later sampled for training. The optimizer O is initialized with a learning rate η along with initializing noise reset countdown for policy and target networks. The optimizer is responsible for updating the policy network parameters θ during the training process, based on the calculated gradients. For each episode ep the environment state is reset. At each step during an episode, the agent selects an action based on the current state s_k. The action is chosen by sampling from a probability distribution obtained using a softmax function over the Q-values, with a temperature parameter μ:

$$\mathcal{G}(a_k|s_k) = \frac{\exp(Q_\theta(s_k,a_k)/\mu)}{\sum_{a'}\exp(Q_\theta(s_k,a')/\mu)} \quad (6)$$

The agent then samples an action from this probability distribution:

$$a_k \sim \mathcal{G}(a_k|s_k) \quad (7)$$

The agent interacts with the environment and stores the transition $(s_k, a_k, r_k, s_{k+1}, d_k)$ into the replay buffer $\mathcal{B}$ with maximum priority, where d_k is a binary flag indicating if the next state is terminal or not. The state is updated with the next state. For each update step, the agent samples a mini-batch of transitions $(s_k, a_k, r_k, s_{k+1}, d_k)$ from the buffer $\mathcal{B}$ with probabilities proportional to the priorities [41]:

$$p_k = \frac{P_k^\alpha}{\sum_i P_i^\alpha} \quad (8)$$

where α is a hyperparameter that determines the degree of prioritization, and $\sum_i P_i^\alpha$ is the sum of priorities raised to the power of alpha over all transitions in the buffer. The sampling weights w_k for a transition k are computed as in [41]:

$$w_k = \frac{(N \cdot p_k)^{-\beta(t)}}{\underset{i}{max}\ w_i} \quad (9)$$

where N represents the size of the agent's replay buffer, β is another hyperparameter that controls the degree of importance sampling. The importance sampling weights are normalized to ensure the stability of the learning process. For each episode, the value of β is computed as follows:

$$\beta_k = \beta_\text{s} + (\beta_\text{e} - \beta_\text{s}) \cdot \min\left(\frac{ep}{\beta_\text{d}}, 1\right) \quad (10)$$

where β_d is the decay duration, and β_s and β_e represent the starting and ending values of beta, respectively. By using this time-dependent β_k value, the prioritization effect will be reduced over time, making the sampling process less biased and more uniform as the agent becomes more experienced.

Next, the current Q-values Q_cur for a sampled state s_k and action a_k are computed using the policy network with parameter θ [39].

$$Q_\text{cur} = Q_\theta(s_k,a_k) = V_\theta(s_k) + A_\theta(s_k,a_k) - \frac{1}{|A|}\sum_{a'} A_\theta(s_k,a') \quad (11)$$

where $V_\theta(s_k)$ is the state-value function, $A_\theta(s_k,a_k)$ is the advantage function, and $|A|$ is the number of actions available.

The expected Q-values Q_exp are computed using the target network with parameter ϕ and the done flag d_k. For each transition in the mini-batch, the expected Q-values are calculated as follows:

$$Q_\text{exp} = r_k + \gamma \cdot Q_\phi(s_{k+1}, \arg\max_{a'} Q_\theta(s_{k+1},a')) \cdot (1-d_k) \quad (12)$$

where $Q_\phi(s_{k+1}, \arg\max_{a'} Q_\theta(s_{k+1},a'))$ is the Q-value from the target network for the next state s_{k+1} and the action a' that maximizes the Q-value in the policy network, γ is the discount factor. Next, the temporal difference (TD) errors represented by δ for each transition in the mini-batch are computed:

$$\delta_k = Q_{cur} - Q_\text{exp} \quad (13)$$

Each transition in the buffer $\mathcal{B}$ is assigned a priority P_k based on the absolute TD error plus a small positive constant ε to ensure that no transition has zero priority:

$$P_k = |\delta_k| + \varepsilon \quad (14)$$

The agent computes the TD errors for the mini-batch, and the loss function is defined as the smooth L1 loss, weighted by the importance sampling weights:

$$L(\theta) = w_k \cdot \rho(\delta_k) \quad (15)$$

where $\rho(x)$ is the smooth L1 loss function, which is similar to Huber loss. It is a combination of Mean Squared Error (MSE) and Mean Absolute Error (MAE) and is defined as follows:

$$\rho(x) = \begin{cases} \frac{1}{2}x^2, & \text{for } |x| \le 1 \\ |x| - \frac{1}{2}, & otherwise \end{cases} \quad (16)$$

The smooth L1 loss function transitions from a quadratic to a linear function as the absolute value of x increases, which helps to reduce the impact of large errors in the training process. The agent updates the policy network $Q_\theta(s,a)$ by performing backpropagation and gradient descent. The gradient

Algorithm 2: FReD-ViQ Training Procedure

Input: State space $\mathcal{S}$, action space $\mathcal{A}$, replay buffer capacity N, batch size b, discount factor γ, learning rate η, soft update factor τ, noise reset interval $\mathcal{I}$, β decay parameters: β_s, β_e, and β_d, batch size b

Initialize policy network Q_θ with parameter θ
Initialize target network Q_ϕ with parameter $\phi \leftarrow \theta$
Initialize replay buffer $\mathcal{B}$ with capacity N
Initialize optimizer $\mathcal{O}$ with learning rate η
Initialize noise reset countdown T using eq. (20)
for ***ep*** $\in E$ **do**
    Reset environment state
    for $k \in K$ **do**
        Compute action probabilities $\mathcal{G}(a_k|s_k)$
        Sample action $a_k \sim \mathcal{G}(a_k|s_k)$
        Execute action a_k, observe reward r_k next state s_{k+1}, and video termination flag d_k
        Store experience tuple $(s_k, a_k, r_k, s_{k+1}, d_k)$ in buffer $\mathcal{B}$ with maximal priority
        Update state $s_k \leftarrow s_{k+1}$
        Compute β_k using eq. (10)
        if *done* **then**
            if $len(\mathcal{B}) \geq b$ **then**
                Sample mini-batch of transitions from $\mathcal{B}$ with probabilities proportional to priorities
                Compute importance sampling weights w_k using eq. (9)
                Calculate current Q-values Q_{cur} and expected Q-values Q_{exp} using eq. (11-12)
                Compute TD errors δ_k using eq. (13)
                Update priorities P_k in buffer $\mathcal{B}$ using eq. (14)
                Compute loss function $L(\theta)$ using eq. (15)
                Update policy network Q_θ using optimizer $\mathcal{O}$ and gradient clipping using eq. (17-18)
                Softly update target network Q_ϕ with mixing factor τ using eq. (19)
                $T \leftarrow -1$
                if $T \leq 0$ **then**
                    Reset noise in policy and target networks
                    Compute T using eq. (20)

of the loss function $L(\theta)$ with respect to the model parameter θ is as follows:

$$\boldsymbol{g} \leftarrow \nabla_\theta L(\theta) \tag{17}$$

The gradient is clipped to prevent it from exploding during backpropagation. Next, the model parameters are updated using the computed gradients:

$$\theta \leftarrow \theta - \eta \cdot \boldsymbol{g}_{clipped} \tag{18}$$

where η is the learning rate, and θ is the updated model parameters. The target network $Q_\phi(s, a)$ is softly updated with a mixing factor τ:

$$\phi \leftarrow (1 - \tau) \cdot \phi + \tau \cdot \theta \tag{19}$$

The noise in the policy and target networks is reset if the noise reset countdown variable T is less than or equal to zero:

$$T \sim \text{Exp}(\lambda) \tag{20}$$

where $\lambda = \frac{1}{\mathcal{I}}$ is the rate parameter, and $\mathcal{I}$ is the noise reset interval. After resetting the noise, a new noise reset countdown variable T is sampled from the same exponential distribution. The training procedure iterates through these steps for a fixed number of episodes E. During this process, the agent learns an optimal policy for selecting actions in the environment. Fig. 6 shows the overall workflow of the FReD-ViQ training mechanism,

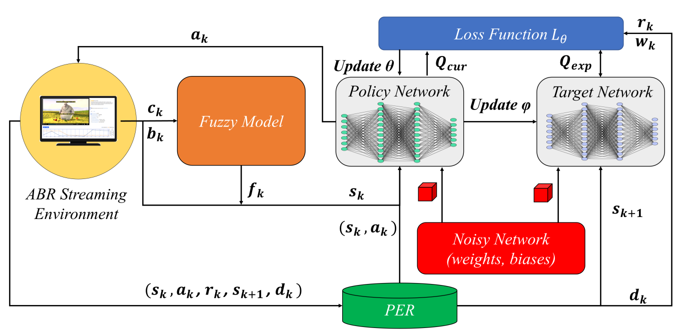


Fig. 6: FReD-ViQ client training mechanism.

V. Performance Evaluation

A. FReD-ViQ Modelling and Implementation Details

1) ABR Streaming Environment: We modelled the FReD-ViQ solution utilizing scikit-fuzzy [48], a Python-based fuzzy logic toolbox, in conjunction with PyTorch [49], a widely-used

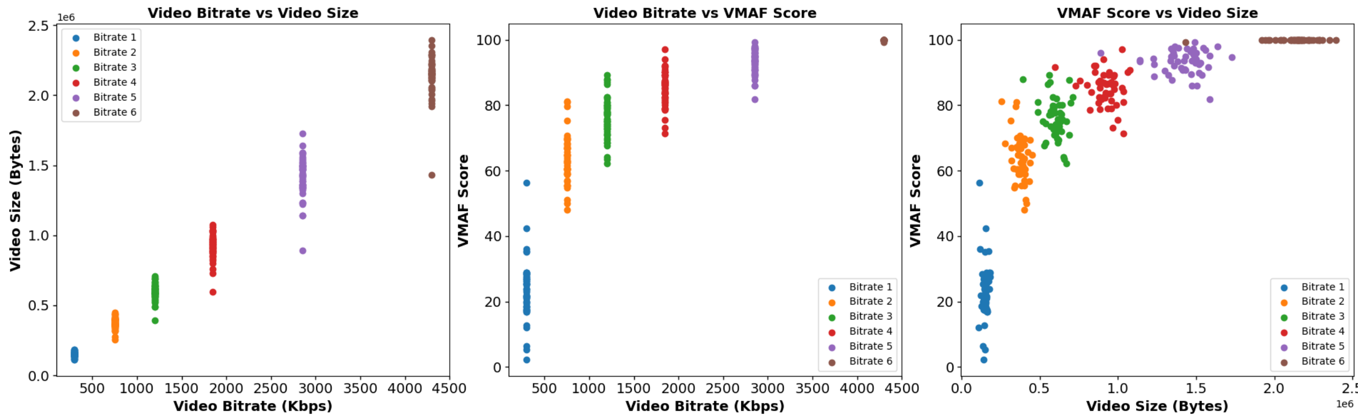


Fig. 7: Relationships between video sizes, bitrates, and VMAF for the test clip used (EnvivioDASH3).

TABLE I: Hyperparameters employed in FReD-ViQ

Parameter	Notation	Value
Batch size	b	32
Replay Buffer capacity	N	500000
Buffer sampling	$\beta_s, \beta_e, \beta_d$	0.4, 1.0, 5x10^4
Discount factor	γ	0.99
Learning rate	η	10^{-3}
Mixing factor	τ	0.001
Noise reset interval	$\mathcal{I}$	1000
Number of episodes	E	100000
Number of segments	K	48 segments
Optimizer	O	Adam
Past samples	p	8
Positive constant	ε	10^{-6}
Probability alpha	α	0.6
Temperature parameter	μ	1.0

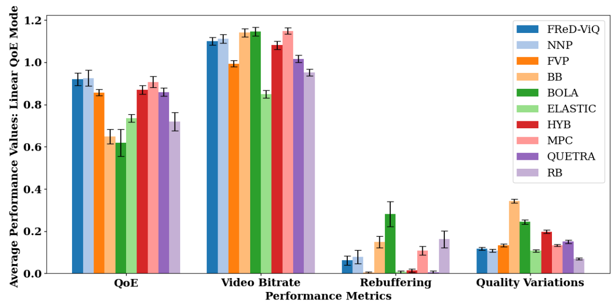
open-source machine learning library. The experiments were conducted on a 64-bit Intel Core i7-7500U CPU with a 2.7 GHz quad-core processor and 16 GB of memory. To ensure the effectiveness of our solution in a realistic streaming scenario, we employed an ABR streaming simulation environment provided by the state-of-the-art PENSIEVE [19] solution. This environment is compatible with the Mahimahi [50] network emulation tool, which enables the accurate assessment of a new algorithm's performance under a wide range of real-world network conditions. For our experiments, we utilized real-world network traces from the publicly available 3G/HSDPA-Norway dataset [51]. This dataset is widely recognized for its diverse and representative collection of network traces, providing a suitable benchmark for evaluating the performance of streaming solutions.

2) Hyperparameters Settings: Table I presents the hyperparameters, notations, and respective values utilized in the training process of the FReD-ViQ solution, specifically chosen to enhance video streaming performance. During each iteration, we use 32 training samples, with a replay buffer capacity set to 500,000 samples and buffer sampling parameters of 0.4, 1.0, and 5x10^4 to dynamically manage sample prioritization within the buffer. The discount factor and learning rates are set to 0.99 and 10^{-3}, respectively. The mixing factor is set to 0.001, regulating the soft update of the target network in the learning process. A noise reset interval of 1000 represents the frequency at which noise is reset in the model. We conduct 100,000 training episodes and employ the widely-used optimization algorithm, Adam [52], for FReD-ViQ training. We set the past samples parameter p to 8 based on previous studies [18], [19], [53], [54], representing the number of previous samples used for input to the model.

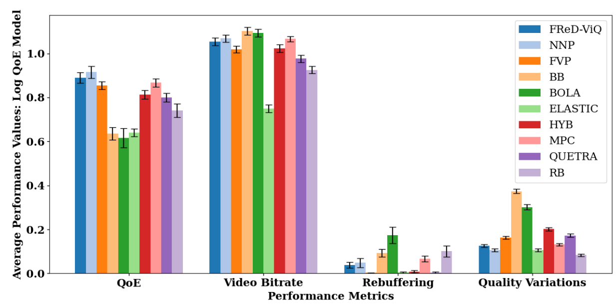
3) Video Data: In our experiments, we used the EnvivioDASH3 [55] video clip, with six different bitrates: 300 (240p), 750 (360p), 1200 (480p), 1850 (720p), 2850 (1080p), and 4300 (1440p) Kbps. This video clip was divided into 48 segments, each with a duration of approximately 4s, resulting in a total playback time of 193s. Fig. 7 provides a visual representation of the relationship between video size (bytes), bitrates (Kbps), and VMAF scores for the selected testing clip. The examination of these relationships offers critical insights into the trade-offs necessary to achieve optimal video quality while reducing buffering and preserving the lowest quality fluctuations in adaptive streaming applications. The scatter plot emphasizes the positive correlation among these three factors, indicating that higher bitrates and larger video files typically result in improved perceived quality, as indicated by higher VMAF scores. Utilizing a video clip with a variety of bitrates and VMAF levels allows one to correctly model and simulate solutions in a wide range of video streaming scenarios.

4) Comparative Solutions: We conducted comprehensive comparisons of FReD-ViQ and its two counterparts, the neural network part (NNP) and fuzzy vector part (FVP), against seven widely recognized and highly cited bitrate selection models in adaptive streaming. To ensure a fair comparison, NNP and FVP were implemented with the same settings as FReD-ViQ. The comparative models include the following:

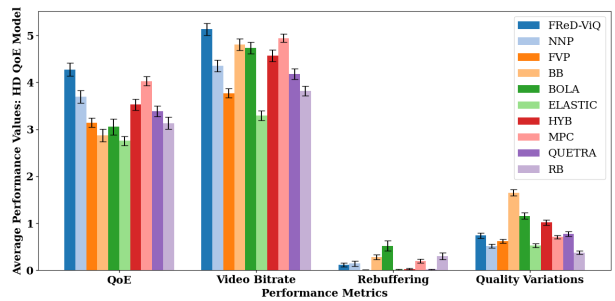
1) **BB** [8]: A simple yet effective bitrate selection algorithm, which primarily focuses on maintaining the buffer occupancy within 2s to 4s.
2) **BOLA** [9]: A buffer-centric Lyapunov optimization techniques-based model that improves the video quality irrespective of the bandwidth.
3) **ELASTIC** [11]: A throughput-based model that adapts to network fluctuations by adjusting the bitrate of the video segments according to the anticipated future bandwidth.



(a) Linear Performance Metrics



(b) Log Performance Metrics



(c) HD Performance Metrics

Fig. 8: Average performance metrics of FReD-ViQ, NNP, FVP, BB, BOLA, ELASTIC, HYB, MPC, QUETRA, and RB algorithms across Linear, Log, and HD QoE models.

4) **HYBrid (HYB)** [56]: A throughput and buffer occupancy-based adaptation approach which aims to improve video smoothness.
5) **MPC** [12]: An algorithm that uses a predictive control framework to optimize QoE for the next 5 segments by estimating future network conditions and considering the impact on the buffer level.
6) **QUETRA** [57]: A model that leverages queuing theory to optimally converge buffer occupancy towards ideal conditions without requiring user-configured weights or thresholds.
7) **Rate-Based (RB)** [58]: A rate-based algorithm, which employs the harmonic mean to forecast throughput and subsequently selects the highest accessible bitrate.

5) Performance Metrics: To facilitate a comprehensive comparison between various streaming mechanisms, we selected several evaluation metrics that effectively measure and demonstrate the performance of each solution. The chosen evaluation metrics are as follows:

1) **QoE Models:** Linear, Log, and HD QoE models were considered to accurately represent the performance improvements achieved by each streaming algorithm.
2) **Video Bitrate, Rebuffering, and Quality Variations:** For each QoE model, we measured video bitrates, rebuffering occurrences, and quality fluctuations.
3) **Buffer Metrics:** The evaluation of *average* and *maximum* buffer levels was carried out to gain insights into the buffer management and stability provided by each streaming mechanism. Additionally, the analysis of *buffer underflow* occurrences helped to estimate the number of times the buffer levels become less than a set threshold of 4s.
4) **Playback Smoothness and Stability:** Both the *frequency* and *magnitude* of quality variations [59] were examined to evaluate the consistency of video streaming quality across different solutions. The *number of rebuffering events* was considered to measure the smoothness of video playback and the ability of each solution to maintain continuous streaming.
5) **Perceived Quality Assessment:** To evaluate the perceived video quality of the streaming mechanisms, we measured the VMAF [60], which provides a reliable estimation of the viewer's experience.
6) **Video Bitrate Choices:** The percentage of video bitrate chosen by each solution was analyzed to compare their adaptability and efficiency in delivering optimal video quality under various network conditions.

B. Experimental Results

1) Linear, Log, and HD QoE Metrics: Figure 8 displays the performance results of our proposed FReD-ViQ, NNP, and FVP solutions, along with seven other comparative solutions, across Linear, Log, and HD QoE models. Following Eq. 2 in [19], we set γ to 1 for Linear, Log, and HD QoE models. However, the value of β varies for each model. Specifically, for the Linear, Log, and HD QoE models, we set the value of β to 4.3, 2.66, and 8, respectively. The proposed FReD-ViQ and NNP solutions consistently achieve the highest average QoE scores for all three models, showcasing their superior performance in delivering an exceptional video streaming experience. For Linear QoE model (Fig. 8a), FReD-ViQ outperforms BB by 41.87%, BOLA by 48.59%, ELASTIC by 25.16%, HYB by 5.83%, MPC by 4.09%, Quetra by 7.14%, and RB by 28.01%. Likewise, for the Log QoE model (Fig. 8b), FReD-ViQ surpasses BB, BOLA, ELASTIC, HYB, MPC, QUETRA, and RB solutions by achieving higher QoE scores of 39.79%,

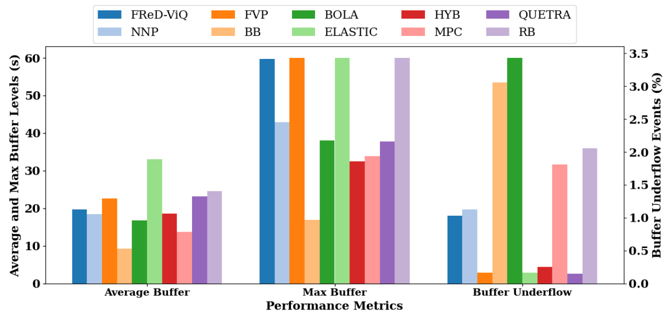


Fig. 9: Buffer Metrics: Average and maximum buffer levels, along with buffer underflow events, for FReD-ViQ, NNP, FVP, BB, BOLA, ELASTIC, HYB, MPC, QUETRA, and RB algorithms.

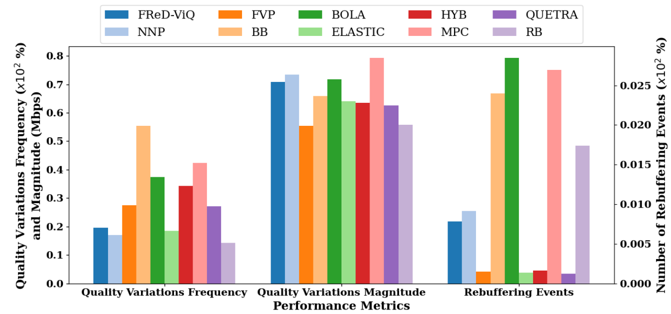


Fig. 10: Frequency and magnitude of quality variations, as well as the number of rebuffering events observed by FReD-ViQ, NNP, FVP, BB, BOLA, ELASTIC, HYB, MPC, QUETRA, and RB algorithms.

44.21%, 39.05%, 9.43%, 4.03%, 11.25%, and 20.07%, respectively. Although FReD-ViQ's average bitrate values for Linear and Log QoE models are slightly lower than those of BB, BOLA, and MPC, it still achieves better performance due to reduced rebuffering and quality variations compared to these solutions (Fig. 8a-Fig. 8b). FReD-ViQ achieves slightly lower Linear and Log QoE values compared to NNP due to marginally higher rebuffering values observed in these models. The results presented in Fig. 8c show that our proposed FReD-ViQ solution achieves the highest QoE scores for the HD QoE model, outperforming the other solutions by a significant margin, i.e., BB by 48.71%, BOLA by 39.90%, ELASTIC by 55%, HYB by 21.21%, MPC by 6.74%, Quetra by 26.01%, and RB by 36.37%. This is because FReD-ViQ observes the highest average HD bitrate values and achieves the highest improvement over the ELASTIC solution (55.66%), followed by RB (34.49%), Quetra (22.82%), and HYB (12.27%). Moreover, NNP and FReD-ViQ experience quality variations across all three models, while maintaining optimal video quality. FVP on the other hand observes the lowest rebuffering penalty compared to other solutions. Solutions such as ELASTIC, HYB, and QUETRA exhibit the lowest rebuffering values due to their tendency to compromise on video quality, whereas buffer-based solutions like BB and BOLA have the highest bitrates for Linear and Log QoE models. FVP enables FReD-ViQ to explore unique streaming patterns to ensure meaningful trade-offs between different streaming metrics. This results in an average improvement of 23.10% (Linear QoE), 23.97% (Log QoE), and 33.42% (HD QoE) over comparative solutions.

2) Buffer Metrics: Figure 9 illustrates the performance of FReD-ViQ, NNP, and FVP solutions compared to seven other solutions, focusing on three buffer metrics: average buffer, maximum buffer, and buffer underflow events. The results highlight FReD-ViQ's ability to strike a well-balanced performance across these metrics. Among proposed solutions, FVP maintains a steadier buffer level and attains the highest average buffer values (22.62s) compared to FReD-ViQ (19.65s) and NNP (18.42s) solutions. The fuzzy-assisted decision-making in FReD-ViQ enables it to achieve a higher average buffer value than BB (9.21s), BOLA (16.75s), HYB (18.64s), and MPC (13.66s). Although FReD-ViQ's average buffer is lower than ELASTIC (33.05s), QUETRA (23.09s), and RB (24.54s), this difference does not adversely affect its overall performance. FVP and FReD-ViQ's maximum buffer value (60s) is on par with ELASTIC and RB and surpasses those of BB (16.85s), BOLA (38.07s), HYB (32.44s), MPC (33.90s), and QUETRA (37.79s). This implies that FReD-ViQ can effectively manage diverse network conditions and minimize the likelihood of playback disruptions due to the precise and balanced selection of membership functions, fuzzy rules, and enhanced exploration-exploitation trade-offs compared to its NNP counterpart. The balanced strategy employed by FReD-ViQ leads to a relatively low number of buffer underflow events (1.03%), outperforming the results of BB (3.06%), BOLA (3.43%), MPC (1.81), and RB (2.05%). FVP on the other hand observes the lowest (0.16%) buffer underflow events along with ELASTIC, HYB, and QUETRA solutions.

3) Playback Smoothness and Stability: Figure 10 presents a comparison of various streaming solutions concerning playback smoothness and stability. We evaluated the frequency and magnitude of quality variations as well as the number of rebuffering events for each solution. Quality variation frequency refers to the rate at which video quality changes between two consecutive segments. FReD-ViQ experiences marginally higher quality variation frequency compared to NNP. This behavior is attributed to FVP, which enables switching using an intermediate bitrate (i.e., 1850 Kbps) for smoother transitions. FReD-ViQ exhibits a moderate quality variation frequency of 19.55%, which is lower than BB (55.32%), BOLA (37.46%), HYB (34.36%), MPC (21.41%), and QUETRA (27.03%). However, it is slightly higher than ELASTIC (18.50%) and RB (14.25%).

Quality variation magnitude represents the extent of changes in video bitrate during streaming, measured in Mbps. FReD-ViQ achieves a quality variation magnitude of 0.70 Mbps, surpassing the magnitudes attained by BB (0.65 Mbps), ELASTIC (0.64 Mbps), HYB (0.63 Mbps), QUETRA (0.62 Mbps), and RB (0.55 Mbps). This indicates that FReD-ViQ facilitates more aggressive quality switches compared to these solutions, ensuring a high-quality streaming experience. However, FReD-ViQ's quality variation magnitude is marginally lower than those of NNP (0.73 Mbps) and MPC (0.8055 Mbps). Despite these differences, FReD-ViQ still delivers a more stable and smooth playback experience, exhibiting a low number of rebuffering events (0.77%), which outperforms BB (2.3%),

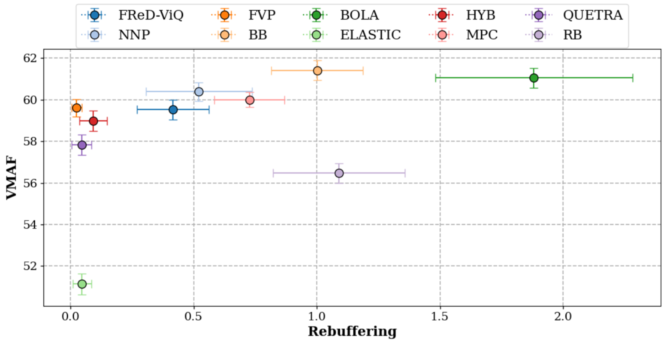


Fig. 11: VMAF vs. Rebuffering experienced by FReD-ViQ, NNP, FVP, BB, BOLA, ELASTIC, HYB, MPC, QUETRA, and RB algorithms.

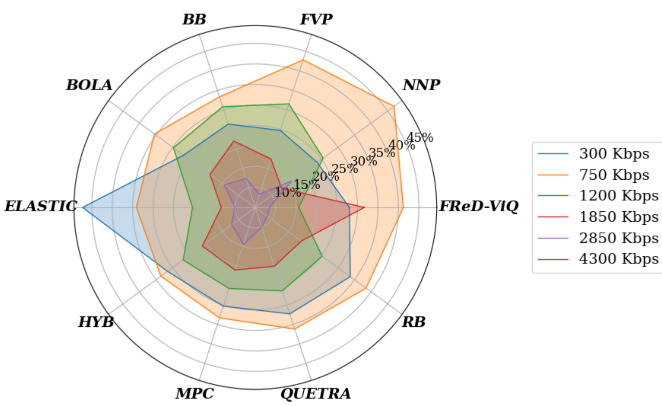


Fig. 12: Percentage of video bitrate choices made by FReD-ViQ, NNP, FVP, BB, BOLA, ELASTIC, HYB, MPC, QUETRA, and RB algorithms.

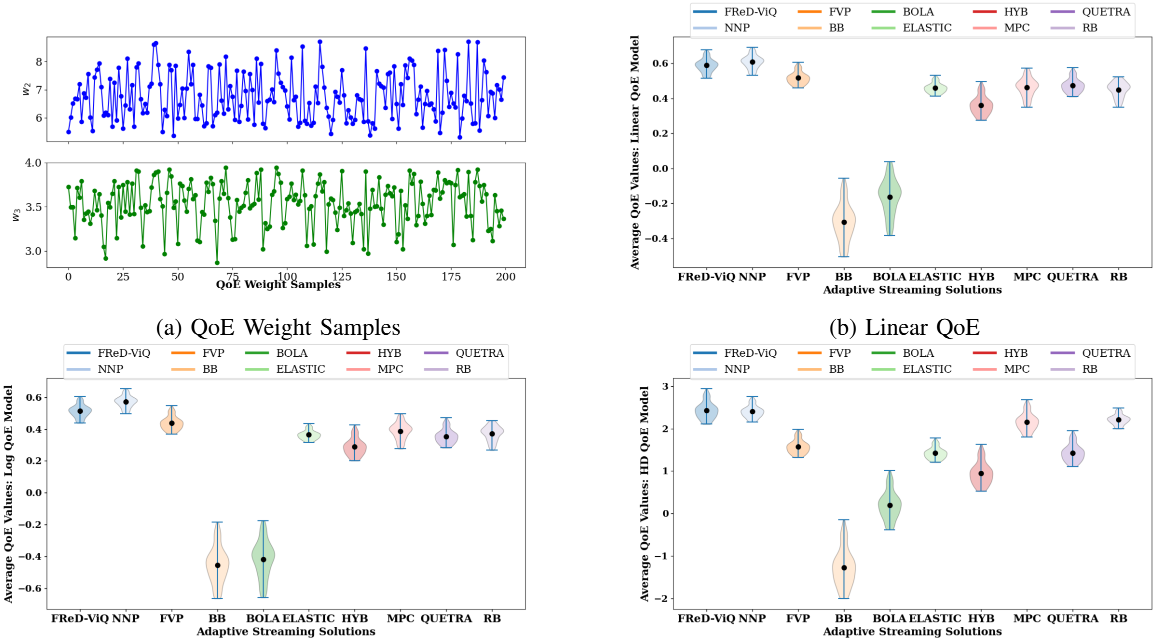
BOLA (2.85%) and MPC (1.54%).

4) VMAF Assessment: We compared FReD-ViQ with other solutions in terms of VMAF and rebuffering values, as illustrated in Fig. 11. The rebuffering values were computed based on the QoE's rebuffering metric presented in [18]. FReD-ViQ achieves a mean VMAF of 59.52, surpassing ELASTIC (51.15), HYB (58.97), QUETRA (57.84), and RB (56.47). This result indicates that FReD-ViQ offers superior average perceived video quality compared to these solutions. However, FReD-ViQ's mean VMAF is marginally lower than NNP (60), BB (61.41), BOLA (61.05), and MPC (60.14). Simultaneously, FReD-ViQ exhibits a mean rebuffering value of 0.42, which is lower than NNP (0.52), BB (1.00), BOLA (1.88), MPC (0.91), and RB (1.09). Among the solutions, the RB solution registers the second-lowest VMAF and the second-highest rebuffering values. FVP achieves the lowest rebuffering while maintaining an acceptable VMAF score. This demonstrates how FVP aids FReD-ViQ in effectively balancing VMAF and rebuffering, ultimately delivering a high-quality and stable video streaming experience.

5) Video Bitrate Choices: Fig. 12 illustrates the streaming behavior of the proposed and comparative solutions in terms of selecting bitrate percentages for 300, 750, 1200, 1850, 2850, and 4300 Kbps. It can be observed how each solution distinctly selects bitrate during the entire playback. FReD-ViQ primarily allocates its bitrate selection to 750 Kbps (36.22%) and 1850 Kbps (26.66%), while opting for 1200 Kbps by only 10.61% of the streaming duration.In contrast, NNP and FVP solutions select 750 Kbps for 41% and 37% of the time respectively. Interestingly, both NNP and FVP demonstrate the lowest percentage of selections for the lowest bitrate, 300 Kbps. Conversely, BB and BOLA solutions are more conservative in selecting higher bitrates, with a higher preference for 300 Kbps and 750 Kbps, while maintaining a moderate distribution across the other bitrate levels. ELASTIC, on the other hand, is heavily biased towards lower bitrates, selecting 300 Kbps (42.25%) and 750 Kbps (29.08%) for a substantial proportion of its streaming decisions. HYB and MPC solutions exhibit similar bitrate selection patterns, with a relatively even distribution across the bitrate levels, except for 2850 Kbps. QUETRA and RB solutions follow a similar trend, although they are more inclined to choose lower bitrate levels, with RB having a higher preference for the 300 Kbps bitrate.

6) Ablation Study: **— Impact of QoE Weight Coefficients:** In this section, we investigate and evaluate the influence of QoE weight coefficients on the playback performance of several adaptive streaming solutions. For Linear, Log, and HD QoE models (Eq. 2), we employed different values of w_2 in the range of (5-10) and w_3 in the range of (2-4). The QoE weight coefficients for 200 samples are shown in Fig. 13a. The results presented in Fig. 13 reveal that the proposed FReD-ViQ and NNP solutions consistently outperforms the other methods, achieving the highest QoE scores across all combinations of QoE weight samples. In terms of Linear QoE (Fig. 13b), NNP achieves the highest score of 0.60, with FReD-ViQ closely following at 0.59. FVP secures the third-best QoE level, while QUETRA and MPC solutions trail behind. Similarly, for Log QoE values (Fig. 13c), NNP maintains the highest average of 0.57, with FReD-ViQ following at 0.51. Notably, FReD-ViQ outperforms the fourth-best solution, MPC, by a significant 30% margin. At the same time, BB and BOLA obtain negative scores of -0.437 and -0.399, respectively. This is because higher values of w_2 and w_3 coefficients for the Log QoE model result in higher rebuffering and quality variation penalties. When considering HD QoE (Fig. 13d), FReD-ViQ excels with a score of 2.43, significantly outperforming other solutions. Interestingly, the RB method achieves the third-highest scores. This highlights the effectiveness of FReD-ViQ in delivering a high-quality video streaming experience across different QoE models and weight coefficient combinations.

— Precision Control in Fuzzy-Driven ABR: Our fuzzy-driven ABR solution, FVP, is carefully designed with optimized membership functions and fuzzy rules to improve bitrate selection. We compared FVP with its counterpart, FVPM (Fuzzy Vector Part with updated Membership functions), which reduces membership functions from seven to five with linguistic variables ["Poor", "Med", "Avg", "Dec", "Good"]. We also updated the fuzzy rules creation expression from $min(i, 2\times j)$ to $min(i, j)$, resulting in the FVPR variant (Fuzzy Vector Part with updated Rules). Fig. 14 depicts the streaming performance of FVP, FVPM, and FVPR solutions for the underlying video, QoE, and network settings. FVP demonstrates



(a) QoE Weight Samples

(b) Linear QoE

(c) Log QoE

(d) HD QoE

Fig. 13: QoE achieved by FReD-ViQ, NNP, FVP, BB, BOLA, ELASTIC, HYB, MPC, QUETRA, and RB algorithms under various QoE weight coefficients.

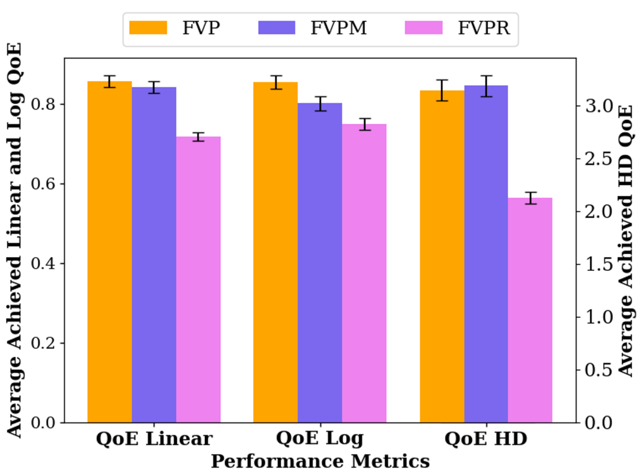


Fig. 14: Linear, Log, and HD QoE values achieved by FVP, FVPM, and FVPR solutions.

the highest average performance in terms of Linear and Log QoE metrics compared to FVPM and FVPR. This is because FVP achieves the highest bitrate and lowest rebuffering and quality variation values compared to its variants. Additionally, FVP exhibits higher HD QoE values compared to FVPR and nearly matches FVPM performance. It is also interesting to note that FVP is highly scalable and adaptable to different buffer, bandwidth, and bitrate values.

7) Findings and Discussions: The extensive experimental results demonstrate that the proposed FReD-ViQ solution consistently outperforms other state-of-the-art methods in terms of different QoE models. The superior QoE performance of the FReD-ViQ underscores its versatility and effectiveness in adapting to different network conditions and QoE preferences and providing a high-quality video streaming experience. Key findings from our experiments include:

1) FVP always leads to the lowest rebuffering values and this behavior positively influences FReD-ViQ's decision-making to minimize rebuffering events.
2) The performance of learning-based models like NNP can be degraded severely when testing under different QoE settings (e.g., HD QoE). However, FReD-ViQ offers more consistent and reliable performance when employed across diverse QoE preferences.
3) While quality variations in FVP are slightly higher than in NNP, this pattern is also seen in FReD-ViQ.
4) FVP is highly scalable, computationally efficient, and easily deployable in any streaming scenario, regardless of the number of bitrate representations (e.g., 6, 8, etc.), QoE models, or network traces. Its counterpart, NNP may require action-space or dimension adjustments when bitrate representations change. Therefore, FVP decreases the training time for FReD-ViQ compared to using NNP alone.
5) Existing solutions (BB, BOLA, ELASTIC, HYB, MPC, QUETRA, RB) rely on fixed heuristics or potentially unreliable assumptions that may not always be accurate or adaptable to fluctuating network conditions.
6) This can lead to suboptimal decision-making and compromised streaming quality. These solutions often demonstrate lower QoE scores, unbalanced rebuffering penalties, and reduced perceived quality.
7) Buffer-based solutions (BB, BOLA) lack throughput learning mechanisms, limiting their capacity to adjust and refine decision-making over time.
8) Solutions dependent on network throughput (RB, ELASTIC) face reliability issues, especially in wireless net-

works where estimated throughput is not always a true indicator of network conditions.

9) Solutions like QUETRA and MPC struggle to balance bitrate exploration, leading to overly conservative or aggressive streaming behavior that negatively impacts viewer experience.

In summary, FReD-ViQ demonstrates exceptional adaptability and effectiveness across a wide range of network conditions and QoE preferences. In addition, the lightweight and integrated design of the FReD-ViQ solution further improves its performance and reliability, making it an ideal choice for real-world applications.

VI. Conclusions and Future Works

This paper introduced FReD-ViQ, an innovative adaptive video delivery solution capable of handling diverse QoE preferences and network conditions. FReD-ViQ employs a combination of fuzzy logic and advanced DRL mechanisms, which enables a more effective value estimation and encourages exploration by introducing stochasticity into the learning process. Furthermore, the lightweight and integrated design of the FReD-ViQ allows for quick learning and faster adaptation over time, improving its decision-making process. Our comprehensive experimental results confirm FReD-ViQ's exceptional performance, achieving important improvements in Linear QoE (23.10%), Log QoE (23.97%), and HD QoE (33.42%), when compared against state-of-the-art solutions.

As part of our future work, we plan to extend the FReD-ViQ framework to incorporate additional content characteristics for advanced interactive multimedia streaming services, such as multi-duration VR and 360° videos. FReD-ViQ will be evaluated in more complex and dynamic network environments, including multi-path and network slicing scenarios, to assess its suitability for 5G cellular networks.

References

[1] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62–67, April 2011.

[2] P. Falkowski-Gilski and T. Uhl, "Current Trends in Consumption of Multimedia Content Using Online Streaming Platforms: A User-Centric Survey," *Computer Science Review*, vol. 37, p. 100268, 2020.

[3] A. Yaqoob, M. A. Togou, and G.-M. Muntean, "Dynamic Viewport Selection-Based Prioritized Bitrate Adaptation for Tile-Based 360° Video Streaming," *IEEE Access*, vol. 10, pp. 29 377–29 392, 2022.

[4] A. Yaqoob, C. Muntean, and G.-M. Muntean, "Flexible Tiles in Adaptive Viewing Window: Enabling Bandwidth-Efficient and Quality-Oriented 360° VR Video Streaming," in *IEEE Int. Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2022, pp. 1–8.

[5] S. Park, A. Bhattacharya, Z. Yang, S. R. Das, and D. Samaras, "Mosaic: Advancing User Quality of Experience in 360-Degree Video Streaming With Machine Learning," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 1000–1015, 2021.

[6] A. Yaqoob, T. Bi, and G.-M. Muntean, "A DASH-based Efficient Throughput and Buffer Occupancy-based Adaptation Algorithm for Smooth Multimedia Streaming," in *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*, June 2019, pp. 643–649.

[7] A. Yaqoob, T. Bi, and G.-M. Muntean, "A Priority-aware DASH-based Multi-View Video Streaming Scheme over Multiple Channels," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, 2020, pp. 297–303.

[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," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 187–198, 2015.

[9] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal Bitrate Adaptation for Online Videos," in *IEEE International Conference on Computer Communications (INFOCOM)*. IEEE, 2016, pp. 1–9.

[10] J. Jiang, V. Sekar, and H. Zhang, "Improving Fairness, Efficiency, and Stability in HTTP-based Adaptive Video Streaming with FESTIVE," *IEEE/ACM Transactions on Networking (TON)*, vol. 22, no. 1, pp. 326–340, 2014.

[11] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, "Elastic: A Client-side Controller for Dynamic Adaptive Streaming Over HTTP (DASH)," in *2013 20th International Packet Video Workshop*. IEEE, 2013, pp. 1–8.

[12] 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

[13] Y. Hou, L. Xue, S. Li, and J. Xing, "User-Experience-Oriented Fuzzy Logic Controller for Adaptive Streaming," *The Computer Journal*, vol. 61, no. 7, pp. 1064–1074, 02 2018. [Online]. Available: https://doi.org/10.1093/comjnl/bxy010

[14] W. U. Rahman, M. D. Hossain, and E.-N. Huh, "Fuzzy-Based Quality Adaptation Algorithm for Improving QoE from MPEG-DASH Video," *Applied Sciences*, vol. 11, no. 11, p. 5270, 2021.

[15] M. Mowafi, E. Taqieddin, and H. Al-Dahoud, "Energy Efficient Fuzzy-based DASH Adaptation Algorithm," *Digital Communications and Networks*, vol. 7, no. 1, pp. 113–119, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352864819301609

[16] H. J. Kim, Y. S. Son, and J. T. Kim, "A Modification of the Fuzzy Logic based DASH Adaptation Scheme for Performance Improvement," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–16, 2018.

[17] Y. Li, X. Zhang, S. Wang, and S. Ma, "A Fuzzy-Based Adaptation Controller For Low Latency Live Video Streaming," in *IEEE International Conf. on Image Processing (ICIP)*. IEEE, 2021, pp. 2169–2173.

[18] 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

[19] 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*. ACM, 2017, pp. 197–210.

[20] 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 Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2324–2342, 2020.

[21] T. Huang, R.-X. Zhang, C. Zhou, and L. Sun, "QARC: Video Quality Aware Rate Control for Real-Time Video Streaming via Deep Reinforcement Learning," in *Proceedings of the 26th ACM international conference on Multimedia*. ACM, oct 2018. [Online]. Available: https://doi.org/10.1145%2F3240508.3240545

[22] W. Shi, Q. Li, C. Wang, L. Zou, G. Shen, P. Zhang, and Y. Jiang, "Learning-Based Fuzzy Bitrate Matching at the Edge for Adaptive Video Streaming," in *Proceedings of the ACM Web Conference 2022*, ser. WWW '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 3289–3297. [Online]. Available: https://doi.org/10.1145/3485447.3511944

[23] M. Gadaleta, F. Chiariotti, M. Rossi, and A. Zanella, "D-DASH: A Deep Q-Learning Framework for DASH Video Streaming," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 703–718, 2017.

[24] 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*, vol. 68, no. 3, pp. 661–676, 2022.

[25] H. Yuan, X. Hu, J. Hou, X. Wei, and S. Kwong, "An Ensemble Rate Adaptation Framework for Dynamic Adaptive Streaming Over HTTP," *IEEE Transactions on Broadcasting*, vol. 66, no. 2, pp. 251–263, 2020.

[26] J. Yao, C. Yan, J. Wang, and C. Jiang, "Stable QoE-Aware Multi-SFCs Cooperative Routing Mechanism Based on Deep Reinforcement Learning," *IEEE Transactions on Network and Service Management*, vol. 21, no. 1, pp. 120–131, 2024.

[27] G. Miranda, D. F. Macedo, and J. M. Marquez-Barja, "Estimating Video on Demand QoE From Network QoS Through ICMP Probes," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 1890–1902, 2022.

[28] T. Huang, R.-X. Zhang, and L. Sun, "Zwei: A Self-Play Reinforcement Learning Framework for Video Transmission Services," *IEEE Transactions on Multimedia*, vol. 24, pp. 1350–1365, 2021.

[29] A. Yaqoob, T. Bi, and G. M. Muntean, "A Survey on Adaptive 360° Video Streaming: Solutions, Challenges and Opportunities," *IEEE Communications Surveys Tutorials*, vol. 22, no. 4, pp. 2801–2838, 2020.

[30] A. A. Barakabitze, N. Barman, A. Ahmad, S. Zadtootaghaj, L. Sun, M. G. Martini, and L. Atzori, "QoE Management of Multimedia Streaming Services in Future Networks: A Tutorial and Survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 526–565, 2019.

[31] M. E. Kanakis, R. Khalili, and L. Wang, "Machine Learning for Computer Systems and Networking: A Survey," *ACM Computing Surveys*, vol. 55, no. 4, pp. 1–36, 2022.

[32] N.-N. Dao, A.-T. Tran, N. H. Tu, T. T. Thanh, V. N. Q. Bao, and S. Cho, "A Contemporary Survey on Live Video Streaming from a Computation-Driven Perspective," *ACM Computing Surveys*, vol. 54, no. 10s, pp. 1–38, 2022.

[33] F. Mason, F. Chiariotti, and A. Zanella, "No Free Lunch: Balancing Learning and Exploitation at the Network Edge," in *IEEE International Conference on Communications*. IEEE, 2022, pp. 631–636.

[34] S. Wang, S. Bi, and Y.-J. A. Zhang, "Deep Reinforcement Learning with Communication Transformer for Adaptive Live Streaming in Wireless Edge Networks," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 308–322, 2021.

[35] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the Impact of Video Quality on User Engagement," *ACM SIGCOMM computer communication review*, vol. 41, no. 4, pp. 362–373, 2011.

[36] J.-M. Martinez-Caro and M.-D. Cano, "On the Identification and Prediction of Stalling Events to Improve QoE in Video Streaming," *Electronics*, vol. 10, no. 6, p. 753, 2021.

[37] E. H. Mamdani and S. Assilian, "An Experiment in Linguistic Synthesis With a Fuzzy Logic Controller," *International journal of man-machine studies*, vol. 7, no. 1, pp. 1–13, 1975.

[38] H. van Hasselt, A. Guez, and D. Silver, "Deep Reinforcement Learning with Double Q-Learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Mar. 2016. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/10295

[39] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling Network Architectures for Deep Reinforcement Learning," in *International conference on machine learning*. PMLR, 2016, pp. 1995–2003.

[40] M. Fortunato, M. G. Azar, B. Piot, J. Menick, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin *et al.*, "Noisy Networks for Exploration," *arXiv preprint arXiv:1706.10295*, 2017.

[41] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized Experience Replay," *arXiv preprint arXiv:1511.05952*, 2015.

[42] D. J. Vergados, A. Michalas, A. Sgora, D. D. Vergados, and P. Chatzimisios, "FDASH: A Fuzzy-Based MPEG/DASH Adaptation Algorithm," *IEEE Systems Journal*, vol. 10, no. 2, pp. 859–868, 2016.

[43] Q. He, C. Dovrolis, and M. Ammar, "On the Predictability of Large Transfer TCP Throughput," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 145–156, 2005.

[44] M. Dräxler and H. Karl, "Cross-layer Scheduling for Multi-Quality Video Streaming in Cellular Wireless Networks," in *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2013, pp. 1181–1186.

[45] C. Zhou, C.-W. Lin, X. Zhang, and Z. Guo, "A Control-Theoretic Approach to Rate Adaption for DASH over Multiple Content Distribution Servers," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 4, pp. 681–694, 2013.

[46] 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.

[47] Y. Sun, X. Yin, J. Jiang, V. Sekar, F. Lin, N. Wang, T. Liu, and B. Sinopoli, "CS2P: Improving Video Bitrate Selection and Adaptation with Data-Driven Throughput Prediction," in *Proceedings of the 2016 ACM SIGCOMM Conference*. ACM, 2016, pp. 272–285.

[48] N. Warner, R. Etemadpour, O. Mastropietro, J. Harrington, and S. Tavakoli, "scikit-fuzzy," 2021. [Online]. Available: https://pypi.org/project/scikit-fuzzy/

[49] 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.

[50] R. Netravali, A. Sivaraman, S. Das, A. Goyal, K. Winstein, J. Mickens, and H. Balakrishnan, "Mahimahi: Accurate Record-and-Replay for HTTP." in *Usenix annual technical conference*, 2015, pp. 417–429.

[51] H. Riiser, P. E. Heegaard, G. Jourjon, A. Duda, and M. Welzl, "Commute Path Bandwidth Traces from 3G Networks: Analysis and Applications," in *Proceedings of the 4th ACM Multimedia Systems Conference (MMSys)*, ACM. ACM, 2013, pp. 68–79.

[52] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2017.

[53] S. Sengupta, N. Ganguly, S. Chakraborty, and P. De, "HotDASH: Hotspot Aware Adaptive Video Streaming Using Deep Reinforcement Learning," in *2018 IEEE 26th International Conference on Network Protocols (ICNP)*. IEEE, 2018, pp. 165–175.

[54] Y. Zhao, Q.-W. Shen, W. Li, T. Xu, W.-H. Niu, and S.-R. Xu, "Latency Aware Adaptive Video Streaming Using Ensemble Deep Reinforcement learning," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 2647–2651.

[55] "Envivio-DASH3," http://dash.edgesuite.net/envivio/Envivio-dash3/index.mpd, accessed: March 4, 2023.

[56] G. Tian and Y. Liu, "Towards agile and smooth video adaptation in dynamic HTTP streaming," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*. ACM, 2012, pp. 109–120.

[57] P. K. Yadav, A. Shafiei, and W. T. Ooi, "QUETRA: A Queuing Theory Approach to Dash Rate Adaptation," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1130–1138.

[58] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate adaptation for adaptive HTTP streaming," in *Proceedings of the second annual ACM conference on Multimedia systems*. ACM, 2011, pp. 169–174.

[59] C. Wang, A. Rizk, and M. Zink, "SQUAD: A Spectrum-based Quality Adaptation for Dynamic Adaptive Streaming over HTTP," in *Proceedings of the 7th International Conference on Multimedia Systems*. ACM, 2016, p. 1.

[60] "VMAF-Video Multi-Method Assessment Fusion," https://github.com/Netflix/vmaf, accessed on: May 1, 2023.

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.

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.