Adaptive Cloud VR Gaming Optimized by Gamer QoE Models

KUAN-YU LEE, National Tsing Hua University, Hsin-Chu, Taiwan
ASHUTOSH SINGLA and PABLO CESAR, Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
CHENG-HSIN HSU, National Tsing Hua University, Hsin-Chu, Taiwan

Cloud Virtual Reality (VR) gaming offloads computationally intensive VR games to resourceful data centers. However, ensuring good Quality of Experience (QoE) in cloud VR gaming is inherently challenging as VR gamers demand high visual quality, short response time, and negligible cybersickness. In this article, we study the QoE of cloud VR gaming and build a QoE-optimized system in a few steps. First, we establish a cloud VR gaming testbed capable of emulating various network conditions. Using the testbed, we conduct comprehensive QoE evaluations using a user study to evaluate the influence of diverse factors, such as encoding settings, network conditions, and game genres, on gamer QoE scores. Second, we construct the very first QoE models for cloud VR gaming using our QoE evaluation results. Our QoE models achieve up to 0.93 ($\sigma = 0.02$) in Pearson Linear Correlation Coefficient (PLCC) and 0.92 ($\sigma = 0.02$) in Spearman Rank-Order Correlation Coefficient (SROCC), where σ stands for the standard deviation. Last, we leverage our QoE models for dynamically adapting encoding settings in our testbed. Extensive experiments revealed that, compared to the current practice, our adaptive cloud VR gaming system improves: (i) overall quality by 0.87 ($\sigma = 0.44$), (ii) visual quality by 0.61 ($\sigma = 0.45$), and (iii) interaction quality by 1.20 ($\sigma = 0.48$) on average in 5-point Mean Opinion Score (MOS).

CCS Concepts: • **Human-centered computing** → **Virtual reality**; • **Information systems** → **Multimedia streaming**;

Additional Key Words and Phrases: VR gaming, cloud gaming, QoE modeling, dynamic adaptation

ACM Reference format:
Kuan-Yu Lee, Ashutosh Singla, Pablo Cesar, and Cheng-Hsin Hsu. 2025. Adaptive Cloud VR Gaming Optimized by Gamer QoE Models. *ACM Trans. Multimedia Comput. Commun. Appl.* 21, 9, Article 256 (September 2025), 24 pages.
https://doi.org/10.1145/3680551

1 Introduction

The **Virtual Reality (VR)** gaming market has witnessed substantial growth and is anticipated to continue its expansion in the forthcoming years. For example, a recent market report [21] indicated that the VR gaming market is projected to demonstrate a Compound Annual Growth Rate of 32.75%

K.-Y. Lee was partially supported by MediaTek and the NSTC of Taiwan (#112-2221-E-007-090-MY3).
Authors' Contact Information: Kuan-Yu Lee, National Tsing Hua University, Hsin-Chu, Taiwan; e-mail: kyl@gapp.nthu.edu.tw; Ashutosh Singla, Centrum Wiskunde & Informatica, Amsterdam, The Netherlands; e-mail: Ashutosh.Singla@cwi.nl; Pablo Cesar, Centrum Wiskunde & Informatica, Amsterdam, The Netherlands; e-mail: P.S.Cesar@cwi.nl; Cheng-Hsin Hsu (corresponding author), National Tsing Hua University, Hsin-Chu, Taiwan; e-mail: chsu@cs.nthu.edu.tw.

ACM 1551-6865/2025/9-ART256
https://doi.org/10.1145/3680551

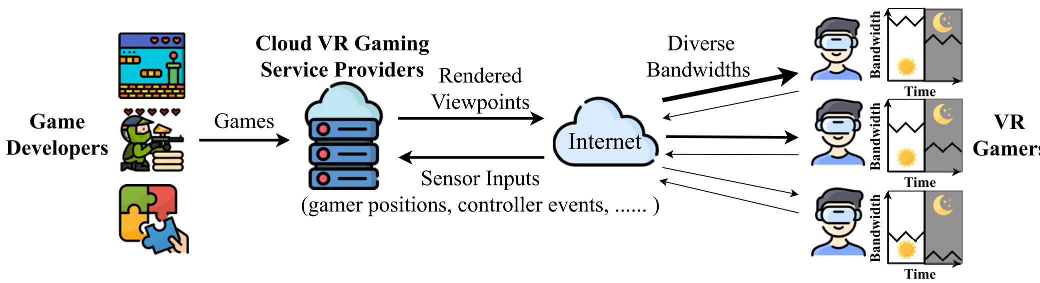


Fig. 1. A typical cloud VR gaming system.

until 2028. The same report also stated that the number of both VR and **Augmented Reality (AR)** gamers is anticipated to reach 216 million by 2025. Key consumer electronic manufacturers, such as Meta, HTC, and Apple, continue to compete for the VR gaming market with substantial investment [22, 62]. Most modern VR games dictate **Head-Mounted Displays (HMDs)** and game controllers for gamer interaction. HMDs can be classified into two types: *tethered* and *standalone*. Standalone HMDs offer gamers freedom, making them preferable for VR gaming without the constraint of cables.

However, the limited **Graphics Processing Unit (GPU)** power and battery capacity of standalone HMDs can detrimentally affect the gamer **Quality of Experience (QoE)**. One possible solution involves wirelessly transferring the rendering workloads to resourceful cloud servers. In fact, as high-speed wireless networks, such as WiFi and 4G/5G cellular networks, are ubiquitously available, they can "glue" VR games and cloud services into *cloud VR gaming systems*. Figure 1 depicts a typical cloud VR gaming system, which consists of three parties: *game developers*, *cloud VR gaming service providers*, and *VR gamers*. Cloud VR gaming service providers obtain VR games from game developers, while these games are executed in virtual machines or containers for individual gamers. The rendered game scenes are captured, compressed, and streamed through the Internet in real-time to VR gamers' HMDs. Simultaneously, the HMDs and controllers intercept, compress, and stream back sensor inputs, enabling gamers to interact with VR games. Different VR gamers have diverse access networks with dynamic bandwidths, which impose additional complications for cloud VR gaming service providers to offer immersive experience to VR gamers.

In particular, VR gamers require for a *short response time* and *high visual quality* when playing cloud VR games. Unlike presentational streaming services [1], such as YouTube, Netflix, and Hulu, cloud VR gaming employs interactive bidirectional communications, in which any deterioration in response time and visual quality could turn gamers away from services. To fulfill the needs of VR gamers, this article extends our preliminary QoE evaluations [35] and makes the following contributions:

— We build a cloud VR gaming testbed that enables us to emulate diverse and dynamic WANs. We design and carry out QoE evaluations using a user study on this open-source testbed [33] to quantify the impacts of different factors, such as encoding settings (bitrate, frame rate, and resolution), network conditions (delay), and game genres on gamer QoE scores. *Our user study is the first investigation conducted on a WAN-based cloud VR gaming system.* We make our user study data available to the research community [34].
— We construct cloud VR gamer QoE models utilizing findings from our QoE evaluations to predict gamer QoE scores under various factors. Given measurable **Quality-of-Service (QoS)** metrics, such as throughput, delay, and packet loss rate, our QoE models achieve high

correlation, reaching up to 0.93 (σ = 0.02) in **Pearson Linear Correlation Coefficient (PLCC)** and 0.92 (σ = 0.02) in **Spearman Rank-Order Correlation Coefficient (SROCC)** [58], where σ stands for the standard deviation. *Our models are the very first ones built for cloud VR gaming systems*. Our models are also available upon request for research purposes.
— We develop a QoE-driven adaptation algorithm at the cloud servers in our system. This algorithm dynamically selects the encoding settings to maximize gamer QoE by considering the current network and system dynamics. Furthermore, we carried out real experiments to assess the effectiveness of our adaptation algorithm in comparison to two baseline approaches. In our cloud VR gaming system, the overall QoE scores in 5-point **Mean Opinion Score (MOS)** are improved by up to 1.86 (σ = 0.38) under congested networks. *Our proposed algorithm is a first for its kind, as QoE-driven adaptation of cloud VR gaming has never been done in the literature.*

The rest of this article is organized as follows. In Section 2, we offer an extensive review of related work. Section 3 delves into the design of our testbed and outlines the associated research challenges. We elaborate on the setup, procedures, and analysis of our QoE evaluations in Section 4. Section 5 focuses on constructing QoE models using results from the user study. The QoE-driven adaptation algorithm for encoding settings is developed in Section 6. Section 7 evaluates the performance of our QoE-optimized cloud VR gaming system. Section 8 draws our conclusions.

2 Related Work

In this section, we survey cloud gaming systems from three aspects: QoE evaluations, QoE modeling, and QoE-driven adaptation.

2.1 QoE Evaluations

Several QoE evaluations have been conducted through user studies to assess gamer QoE of cloud gaming. For example, Jarschel et al. [26] evaluated gamer QoE under diverse delays and packet loss rates and identified the key factors using their home-brew cloud gaming testbed. Sackl et al. [48] manipulated the delay between the server and client to investigate its impacts on gamer QoE across different game genres on the Steam In-home streaming platform. Slivar et al. [55] adopted the same platform for another user study of different encoding settings with two game genres. GamingAnywhere [18] was the first open-source cloud gaming platform which can be extended for user studies. For example, we conducted a user study using GamingAnywhere to analyze how different parameters, such as resolution, bitrate, frame rate, and network delay, affect the mobile gamer QoE [19]. *Different from our current work, these papers* [19, 26, 48, 55] *considered traditional cloud gaming rather than cloud VR gaming.*

The challenges become more complicated when VR is introduced, given the heightened requirements for low delay and increased sensitivity to quality impairments. This is particularly evident in VR gaming, where the interactive nature of games places significant demands on both delay and quality compared to other VR applications. More recently, QoE evaluations of VR gaming have also been investigated. For example, Vlahovic et al. [63] designed two user studies to find out the relationship between network delay and gamer QoE in a first-person shooter VR game. Their observations highlighted that contextual factors, such as social context and difficulty levels, can mask the negative effects due to long network delays. Slivar et al. [56] evaluated gamer QoE in a user study across various networks (4G, 5G, and Ethernet), considering two multiplayer VR game genres. Their study also delved into the influence of social context on gamer QoE. *These works* [56, 63] *only focused on local VR gaming rather than cloud VR gaming.*

For cloud VR gaming, we designed a remote VR gaming testbed [37] on the basis of **Air Light VR (ALVR)** [46] and conducted a user study under different network conditions using three game genres. We reported that insufficient bandwidth and a high packet loss rate may cause higher negative impacts on the QoE than additional delay. *That work employed a remote VR gaming system on a LAN.* In contrast, we recently applied dynamic foveation to WAN-based cloud VR gaming built upon **Air Light XR (ALXR)** [47]. Specifically, we conducted a small user study [13] by varying foveation parameters, including the foveal region size and the compression ratio of the peripheral area. *The current article presents more comprehensive QoE evaluations focusing on gamer QoE, which enables the construction of QoE models and QoE-driven adaptation algorithm.* The preliminary results of our QoE evaluations were given by Lee et al. [35].

2.2 QoE Modeling

Several research groups have built QoE models for cloud gaming. For example, Wang and Dey [65] proposed a QoE model that considers game genres, encoding settings, video quality, response time, and packet loss rates as inputs to predict mobile gamer QoE. They derived impairment functions from the QoE evaluations to predict the Game MOS of each gamer. Slivar et al. [54] modeled game-dependent QoE using a quadratic function, which takes the frame rate and bitrate as inputs. Furthermore, they considered game genres and gamer QoE in their models. Different from directly using the bitrate and frame rate as inputs, Zadtootaghaj et al. [68] introduced structural QoE models based on several intermediate factors derived from other raw inputs. ITU-T recommendation G.1072 [25] presented an opinion model for predicting cloud gamer QoE scores. The model works in two modes: one that takes game genres into account and another that does not. The model calculates various impairment factors based on encoding and network metrics to predict the gamer QoE. *Different from our work, these studies* [25, 54, 65, 68] *considered traditional cloud gaming rather than cloud VR gaming.* Several works [5, 36, 61, 67] derived QoE models for consuming 360° VR videos; however, little has been done to VR gaming. Although Krogfoss et al. [29] presented a video and a gamer QoE model based on parameters like delays and packet loss rates, their QoE models were not built upon real user-study results. Instead, their models were essentially heuristics based on findings in the literature.

2.3 QoE-driven Adaptation

Several works have been done to adapt the bitrate on the fly in video streaming sessions. For example, Cofano et al. [8] and Sobhani et al. [57] proposed bitrate adaptation algorithms for HTTP Adaptive Streaming systems. *Different from our work, these adaptation algorithms are not QoE-driven.* For QoE-driven adaptation several studies [44, 49, 66] adapted streaming frameworks leveraging either the QoE models or QoE-related metrics. These algorithms are mostly pull-based and thus are inapplicable to push-based cloud gaming. For push-based adaptation, Khan et al. [28] proposed a QoE-driven bitrate adaptation scheme built upon fuzzy logic. It calculated the levels of congestion and degradation according to packet loss rates and QoE models, respectively. It then changed the bitrate accordingly. *Most of these studies* [8, 28, 44, 49, 57, 66] *are for video streaming rather than more challenging cloud gaming systems, and most of them only take the bitrate into consideration, excluding the frame rate and resolution.*

QoE-driven adaptation in cloud gaming has only been recently considered, e.g., Slivar [53] introduced three adaptation algorithms for the bitrate and frame rate. These algorithms were built upon the findings in their QoE evaluations. Our prior work [17] facilitated adaptive cloud gaming in GamingAnywhere [18] by dynamically reconfiguring the encoding settings considering the bitrate and frame rate. Additionally, we developed techniques for optimal bitrate allocation, selecting the most suitable bitrate and frame rate for each gamer to maximize the overall gamer QoE scores. *The* *current article introduces a QoE-driven adaptation algorithm in cloud VR gaming instead of traditional cloud gaming* [17, 53].

3 Building a Cloud VR Gaming System

Cloud VR gaming presents unique challenges compared to the following relevant systems:

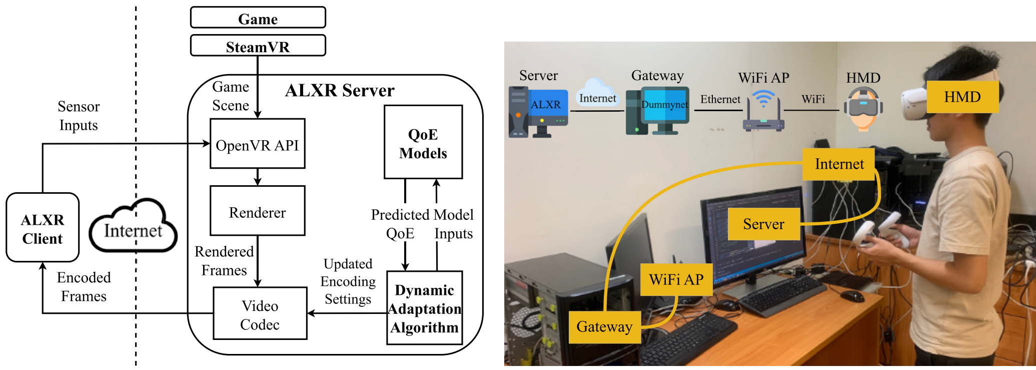
— 360° **Video-on-Demand (VoD)** [12]. 360° VoD systems like YouTube operate with unidirectional streaming. As a result, videos can be downloaded and buffered at each client for relatively long durations to mitigate the negative impacts due to network delay and jitter. In contrast, cloud VR gaming streams bidirectionally. The server renders scenes based on the gamer's position received from the client in real time and then transmits it to the client. Consequently, delays and jitters cannot be mitigated by a large buffer. Understanding the behaviors of bidirectional cloud VR gaming systems with small buffers requires us to build a real cloud VR gaming system and measure its detailed performance in various metrics.
— *Traditional cloud gaming* [6]. Traditional cloud gaming systems like GamingAnywhere [18] operate with 2D monitors. Compared to HMDs used in cloud VR gaming, QoE with 2D monitors is well-studied. While cloud VR gaming systems employ HMDs for potentially higher gamer QoE, the added dimensions of QoE factors increase the complexity level to deliver immersive experiences. Therefore, QoE models are essential in cloud VR gaming to efficiently estimate the gamer QoE.
— *Locally rendered VR applications* [64]. Local VR applications do not engage in remote rendering, thereby remaining unaffected by imperfect network conditions. In contrast, cloud VR gaming renders game scenes on potentially far-away cloud servers and thus is sensitive to bad network conditions. Consequently, adaptation to network and system dynamics becomes crucial to alleviate their negative impacts on gamer QoE.

In this article, we set out to develop a cloud VR gaming system and address its unique challenges mentioned above. Compared to commercial cloud VR gaming systems, open-source systems are easier to augment and enhance for research. Among the most prominent open-source systems are NVIDIA CloudXR [41] and ALVR [46]. NVIDIA CloudXR supports streaming XR content using the OpenVR API for Android and Windows devices. Unfortunately, NVIDIA only open-sources CloudXR's client. This prevents researchers from integrating their innovations into CloudXR's server for experiments. In contrast, ALVR is an open-source project on both the server and client, which is a better choice. ALVR uses OpenVR API to obtain game scenes from SteamVR games, which only supports a limited number of HMD models. ALXR [47] is an extension to ALVR, which adopts OpenXR in the client to support more HMD models. Hence, we built our open-source cloud gaming system on top of ALXR.

Figure 2(a) presents our proposed cloud VR gaming architecture of a client-server pair. Once the connection between them is established, the ALXR server extracts the game scenes from SteamVR into video frames through OpenVR API.[1] Then, it encodes the frames and sends them to the client through the Internet. Meanwhile, the client displays the received frames and sends the sensor inputs back to the server. According to the sensor inputs from the client, ALXR server replays the gamer's motions, extracts new game scenes, and sends the updated frames.

Figure 2(b) shows our ALXR-based cloud VR gaming testbed. We use a Windows 10 PC as our server. It comes with an Intel Core i9 CPU, 64 GB RAM, and an NVIDIA GeForce RTX 3080 Ti GPU and is connected to the Internet through a GigE cable. We use a Meta Quest 2 HMD

[1] ALXR project reuses ALVR's server implementation built on OpenVR API.



(a) (b)

Fig. 2. Cloud VR gaming: (a) architecture and (b) testbed.

as our client. It comes with a Qualcomm Snapdragon XR2 CPU, 6 GB RAM, and an Adreno 650 GPU and is connected to a WiFi 6 AP. Between the Internet and WiFi AP, we add a FreeBSD 13.1 gateway running Dummynet [14] to emulate diverse and dynamic network conditions. We install ALXR version 18.2.3. Originally, ALXR assumes LAN environments, which is less challenging than our envisioned cloud VR gaming scenario. To conduct WAN-based realistic cloud VR gaming experiments, we enhanced ALXR into a cloud VR gaming system [33]. In particular, we transformed the original server-centric ALXR architecture, where the server discovers the client, into a client-centric ALXR, where the client connects to a user-specified cloud gaming server.

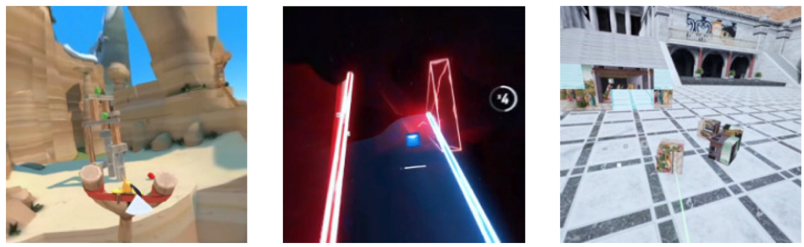
Developing cloud VR gaming systems with *short response time* and *high visual quality* is no easy task because of the best-effort Internet, non-real-time operating systems, and hard-to-predict human perception. We face three primary challenges when doing so. First, multiple factors, such as network conditions, encoding settings, and game genres, affect gamer QoE. Second, gathering gamer QoE scores takes time, as controlled QoE evaluations are time-consuming by nature. Third, even if we can estimate the QoE scores, it is not trivial to leverage them in our cloud VR gaming system for optimizing the gamer QoE. We addressed these challenges in three steps. In Section 4, we conduct comprehensive QoE evaluations using a user study on our open-source cloud VR gaming system. In Section 5, we analyze gamer QoE scores and build corresponding QoE models. In Section 6, we incorporate the QoE models to enable QoE-driven dynamic adaptation of encoding settings at runtime.

4 QoE Evaluations

In this section, we conduct QoE evaluations using a user study.

4.1 Setup

To be comprehensive, we aimed to employ VR game genres with diverse characteristics. In particular, we employed **Temporal Perceptual Information (TI)** and **Spatial Perceptual Information (SI)** [23] to characterize game genres following prior works [54, 59]. Between them, TI captures object motions, behavioral patterns, and changes occurring over time across video frames. In contrast, SI focuses on the characteristics of individual frames, including the spatial layout of pixels and static



(a) (b) (c)

Fig. 3. Sample scenes of three considered games: (a) AngryBird, (b) BeatSaber, and (c) ArtPuzzle.

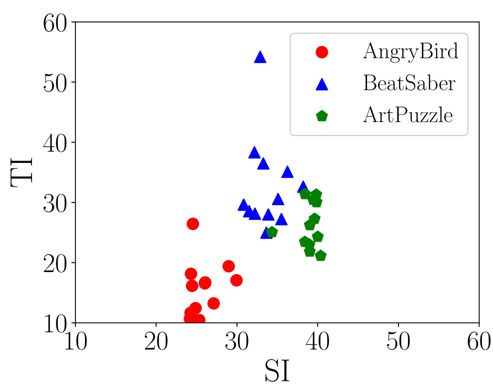


Fig. 4. TI vs. SI values from different game genres.

properties of objects, such as colors. After considering multiple candidate VR games, we chose the following three VR games, as shown in Figure 3:

— *AngryBird.* A player uses a slingshot to launch birds with the goal of knocking down all pigs. It is a leisure game.
— *BeatSaber.* A player slashes through the moving boxes on the beats with specified directions. It is a fast-paced game.
— *ArtPuzzle.* A player manipulates pieces to complete each puzzle. It is a slow-paced game with many texture details.

To understand their temporal and spatial characteristics, we plot the TI and SI values of the rendered game scenes from 12 subjects (reported in Section 4.3) in Figure 4. This scatter plot reveals that the game scenes from different games naturally scatter into three clusters. In particular, we observe that (i) *AngryBird* has low TI and SI values and is less sensitive to time and quality, (ii) *BeatSaber* has the highest TI values and is time-sensitive, and (iii) *ArtPuzzle* has the highest SI values and is quality sensitive. That is, these three representative games cover the spectrum of diverse temporal and spatial characteristics.

We varied multiple parameters in the user study. A pilot test was conducted with five subjects to adjust the parameter ranges. In particular, we explored a broader range of values and then selected a narrower range. This narrower range is sufficient for subjects to perceive differences, thereby achieving a balance in experiment duration to avoid subject fatigue. The values of each parameter are presented below, with bold font indicating default settings:

— *Bitrate*. The number of bits per second used for encoding. Higher bitrate offers better quality at the cost of larger compressed scene size, while lower bitrate reduces the size at the expense of lower quality. We denote the bitrate as b, where $b \in \mathbb{Q}^+$. We vary it in {2, 8, **32**} Mbps.
— *Frame rate*. The number of frames every second. Higher frame rates offer smoother videos but incur higher computational and storage costs, while lower frame rates reduce these costs but may result in choppier videos. We denote the frame rate as f, where $f \in \mathbb{Z}^+$. We vary it in {12, 24, 36, **72**} frame-per-second (fps).
— *Resolution*. The number of pixels contained in each game scene. Higher resolution offers finer details but introduces more information to compress, while lower resolution leads to less information but lacks detail. We denote the resolution as r, where both width and height are $\in \mathbb{Z}^+$. We vary r in {1408×768, 2112×1184, **2880×1568**}. For ease of expression, we refer to these resolutions as 768p, 1184p, and 1568p in the rest of this article.
— *Delay*. The local Round-Trip Time is about 10 ms in our system. We inject an extra round trip delay of {**0**, 100, 300, 500} ms on the gateway, representing the domestic delay as well as delays between the USA and Europe, East Asia and South America, and Oceania and Africa, respectively.

We group *bitrate*, *frame rate*, and *resolution* into encoding settings. We consider *delay* as the key parameter of network conditions due to the strict real-time requirement of cloud VR gaming services.

4.2 Measurement Methodology

We measure the following metrics:

— *Throughput*. The receiving speed at the client is denoted as p.
— *Frame loss rate*. The fraction of lost frames.
— Delay. The round-trip delay between the server and client is denoted as d.
— *Packet loss rate*. The fraction of lost packets is denoted as l.
— **Peak Signal-to-Noise Ratio (PSNR)**. A widely used video quality metric in the decibel scale [38, Ch. 8].
— **Structural Similarity Index (SSIM)**. Another video quality metric that takes human perception into consideration [38, Ch. 12].
— **Video Multimethod Assessment Fusion (VMAF)**. A learning-based video quality metric based on human perception [20].[2]

In terms of measurements, we measure the throughput, delay, and packet loss rate by instrumenting the source code. To calculate PSNR, SSIM, and VMAF, we capture the rendered frames at the server to be *reference frames*. For *decoded frames*, due to hardware limitations, we cannot directly save the frames on the client side in real time. Moreover, decoded frames must go through some matrix transformation to compensate for lens distortion, which further complicates the task at hand. Thus, we develop a two-step approach. First, we augment the encoder at the server to compute the *encoding* distortion. To account for frame loss due to packet loss, we add QR codes to the reference frames on the server. We then match the QR codes between them and the decoded frames captured on the client. Once a frame is lost, we duplicate the previously decoded frame for error concealment. Last, we compute the objective video quality of the concealed frames for the

[2]We followed the recommendation for 360° video scenarios [40] and employed the default 1080p model.

Table 1. Human Factors in GE and VE

Factor	**GE Levels (game time per week in hr)**			**VE Levels (prior VR experience)**	
Desc.	Novice (< 1)	Intermediate (≥ 1 and < 5)	Advanced (≥ 5)	No	Yes
Percent.	25%	25%	50%	50%	50%
Enum.	1	2	3	0	1

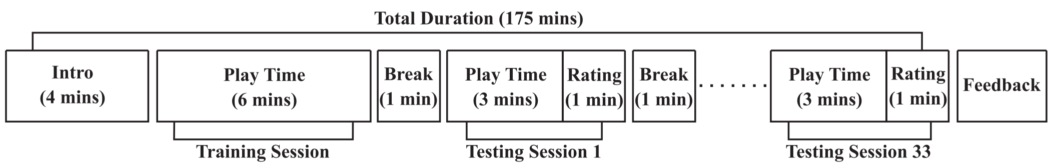


Fig. 5. Procedure of the user study.

transmission distortion. We sum up the encoding and transmission distortion for the final video quality.

4.3 User Study

We recruited 12 subjects to conduct our user study, of whom 10 were males. All subjects were college students between 20 and 25 years old with 20/20 corrected vision in the Snellen test. They also passed the Ishihara test for color vision. We considered two human factors: **Gaming Experience (GE)** and **VR Experience (VE)** levels. As summarized in Table 1, we categorized all subjects into three GE levels: (i) novice (< 1 hour game time per week), (ii) intermediate (≥ 1 and < 5 hours), and advanced (≥ 5 hours). We enumerated the GE levels into 1, 2, and 3 for the sake of presentation. There were 3, 3, and 6 subjects in the GE levels, respectively. Table 1 also shows that we classified all subjects into two VE levels using a Boolean value, where 0 means no prior VE.

Figure 5 shows the procedure of our user study. At the beginning, we provided an introduction to each subject. In the training session, the players played all three games to get familiar with the HMD and controllers. The game scenes/levels we used in the training sessions were different from those in the testing sessions. To avoid fatigue, we only varied one factor at a time, leading to 11 scenarios (sessions) for each game. Table 2 lists all the scenarios. Since we had three considered games, each subject underwent 33 sessions. There was a 1-minute break after each session. The order of sessions was random to avoid the learning effect. We recorded each subject's inputs for our QoE questions, which are given in Table 3. This table consists of all questions used in this section (upper half) and in the performance evaluations section (Sec. 7). In this section, we asked all questions except for the last row of the table. Particularly, there are five questions [43, 50, 54, 60]: *Overall Quality (O)*, *Visual Quality (V)*, *Immersive Level (I)*, *Cybersickness (S)*, and *Continue (C)*. The ratings are on a 1–5 scale using **Absolute Category Rating (ACR)**, where higher is better, except for: (i) Continue, which is a Boolean value and (ii) Cybersickness, where lower is better. It is worth noting that we avoided long cybersickness questionnaires [27] to prevent the prolonged duration of each session, which would limit the number of tested conditions [52]. Furthermore, our focus was on the mean cybersickness score, and thus, the longer cybersickness questionnaires may not be necessary [16]. Even after doing so, the user study duration of each subject was still too long, so we had to separate each subject's sessions into two days, for varying: (i) encoding settings on day 1, which lasted for about 120 minutes, and (ii) network conditions on day 2, which lasted for about 45 minutes. It took us about 45 hours to complete the user study. Given that we had 12 subjects and 33 sessions each, we gathered a total of 396 responses throughout the user study. We analyzed the results below.

Table 2. Scenarios for Each Game Genre

Bitrate (Mbps)	Frame Rate (fps)	Resolution	Delay (ms)
2	72	2880×1568	0
8	72	2880×1568	0
32	72	2880×1568	0
32	12	2880×1568	0
32	24	2880×1568	0
32	36	2880×1568	0
32	72	2112×1184	0
32	72	1408×768	0
32	72	2880×1568	100
32	72	2880×1568	300
32	72	2880×1568	500

Table 3. QoE Questionnaire for QoE User Study

QoE	Question	Rating	QoE Experiments (Section 4)	Performance Evaluations (Section 7)
Overall Quality (O)	How would you rate the overall quality of this gaming session?	1 (Bad) – 5 (Excellent)	✓	✓
Visual Quality (V)	How would you rate the visual quality of this gaming session?	1 (Bad) – 5 (Excellent)	✓	✓
Immersive Level (I)	What is your assessment of the sense of immersion during this gaming session?	1 (Low) – 5 (High)	✓	×
Cybersickness (S)	Are you feeling any sickness or discomfort now?	1 (No problem) – 5 (Unbearable)	✓	✓
Continue (C)	Would you like to continue to play under this condition?	0 (No) – 1 (Yes)	✓	×
Interaction Quality (A)	How responsive was the environment to actions that you performed?	1 (Not responsive) - 5 (Completely responsive)	×	✓

4.4 Results

Bitrate affects the gamer QoE the most among other encoding settings. Figure 6 gives the MOS scores of overall quality under different encoding settings. Slopes in Figure 6(a) are generally steeper compared to those in Figures 6(b) and (c), showing that the bitrate imposes the most significant impact on the gamer QoE. We performed Wilcoxon signed-rank tests between the MOS of the lowest and highest values for each encoding setting. We found that the p-values of bitrate are almost consistently lower than those of the frame rate and resolution across all three games (except the p-value of the frame rate in BeatSaber). This confirms that the bitrate is the most important encoding setting. Note that all these p-values are below 0.001, demonstrating a clear statistical difference.

MOS growth rate decelerates as bitrate increases. Figure 7 presents sample quality and immersion results under different bitrates. We observe that both MOS of visual quality and objective quality metrics, i.e., VMAF, improve rapidly from 2 to 8 Mbps, with an average slope of 0.25 and 2.87, respectively. However, the improvement decelerates from 8 to 32 Mbps, with an average slope of 0.08 and 1.31, respectively. The same behavior of the immersive level can be seen in Figure 7(c). While we cannot show all figures due to the space limitation, a similar trend was also observed with other QoE questions, e.g., MOS of overall quality in Figure 6(a). These observations indicate that as the bitrate increases, the growth rate of MOS decelerates gradually.

Different game genres have different requirements. Figure 6(a) reveals that MOS of overall quality is more sensitive in ArtPuzzle under different bitrates. The same can be said with visual quality and immersive level in Figures 7(a) and (c), compared to other game genres. In these cases, the p-values for ArtPuzzle on MOS are lower than those in AngryBird and BeatSaber, and all of the values are

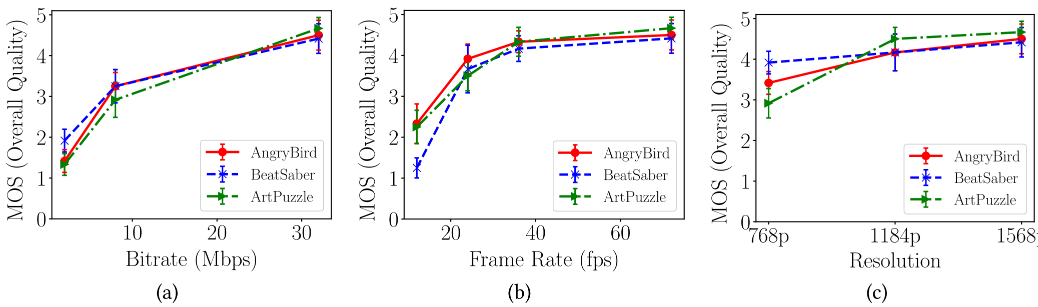


Fig. 6. MOS of overall quality under different settings, sample results under default encoding and network factors with varying: (a) bitrate (72 fps, 2880×1568), (b) frame rate (32 Mbps, 2880×1568), and (c) resolution (32 Mbps, 72 fps).

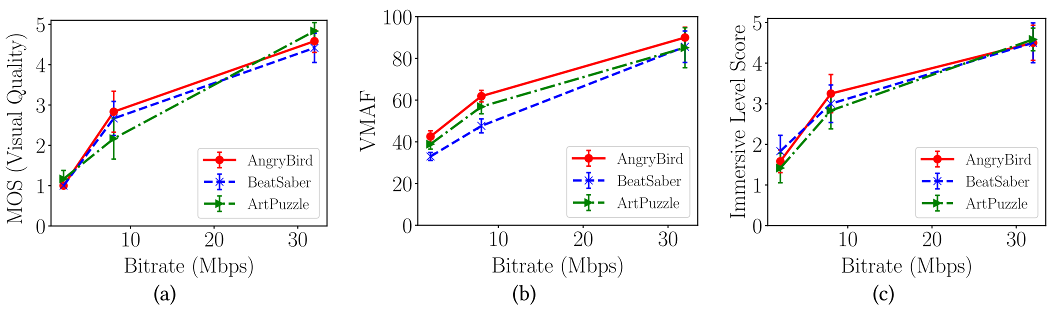


Fig. 7. Implication of bitrate with default frame rate (72 fps), resolution (2880×1568), and delay on: (a) MOS of visual quality, (b) objective quality in VMAF, and (c) immersive level score.

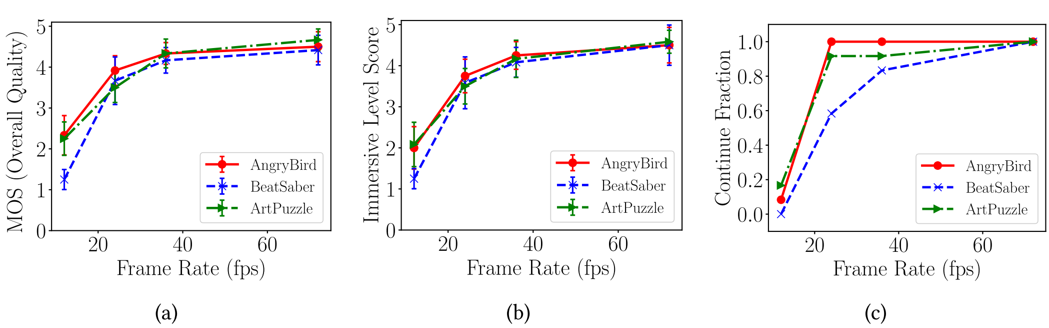


Fig. 8. Implication of frame rate with default bitrate (32 Mbps), resolution (2880×1568), and delay on: (a) MOS of overall quality, (b) immersive level score, and (c) fraction of continue.

below 0.001 after conducting the Wilcoxon signed-rank tests, showing statistical differences. This is intuitive, as ArtPuzzle needs higher visual quality due to its texture details. Figure 8 reports the influence of varying frame rates. Figures 8(a) and (b) depict that when the frame rate drops below 24 fps, the MOS of overall quality and immersive level score drop drastically, especially for BeatSaber. The p-values between 24 and 12 fps are both below 0.001, with MOS differences of 2.08 and 2.33, respectively. Figure 8(c) shows that no one wants to continue playing BeatSaber at 12 fps, while AngryBird and ArtPuzzle are still acceptable to 10% and 20% of gamers, respectively. Figure 9 presents the implication of extra delay on overall quality and immersive level. Similar

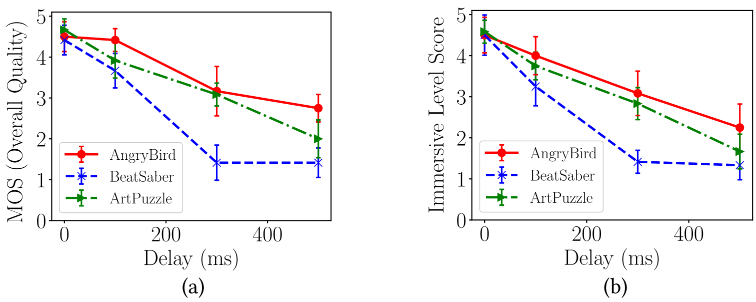


Fig. 9. Implication of delay with default bitrate (32 Mbps), frame rate (72 fps), and resolution (2880×1568) on (a) MOS of overall quality and (b) immersive level score.

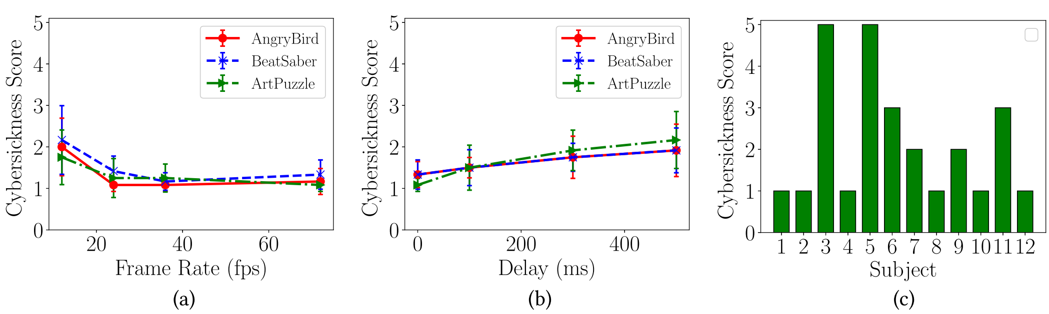


Fig. 10. Cybersickness score with default parameters and different: (a) frame rate (32 Mbps, 2880×1568), (b) delay (32 Mbps, 72 fps, 2880×1568), and (c) subjects at 12 fps (32 Mbps, 2880×1568).

to Figure 8, BeatSaber is more sensitive to injected delays, as gamers may not react in time. The p-values between 0 and 500 ms of injected delay are both below 0.001, with MOS differences of 3.02 and 3.17, respectively. From the observations above, it is statistically significant that diverse game genres incur different requirements on the QoS.

Cybersickness highly depends on the subjects. Figure 10 summarizes the cybersickness scores under diverse factors. We observe that the cybersickness score remains relatively consistent across most frame rate and delay settings unless the frame rate drops below 24 fps (Figure 10(a)) or the delay approaches 500 ms (Figure 10(b)). In these extreme cases, the average cybersickness score is increased by 0.78 and 0.81, respectively. However, the p-values are all above 0.01 in these cases after conducting Wilcoxon signed-rank tests, which indicates minor significance. A deeper investigation indicates that even under these unfavorable settings, such as a frame rate of 12 fps, 50% of the subjects gave a rating of 1 (No problem), as illustrated in Figure 10(c). We conclude that cybersickness scores largely depend on subjects. Thus, we leave modeling cybersickness as one of our future works.

5 QoE Modeling

In this section, we model the gamer QoE scores using the data collected from our QoE evaluations. We leave modeling *cybersickness* as our future work. We also exclude modeling *continue* since we fail to see immediate applications.

Table 4. QoE Model Inputs

Category	Input
Encoding Setting	Bitrate, Frame Rate, Resolution
Network Condition	Throughput, Frame Loss Rate[a], Packet Loss Rate, Delay
Video Quality Metric	PSNR[a], SSIM[a], VMAF[a]
Human Factor	GE, VE
Game Genre	TI, SI

[a]Measured by external measurement tools rather than instrumented code.

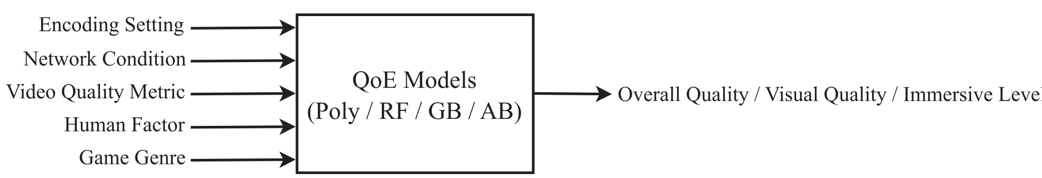


Fig. 11. Block diagram of the QoE models.

5.1 Modeling Approach

We model the *overall quality*, *visual quality*, and *immersive level* as: $Q_O(b, f, r, \ldots)$, $Q_V(b, f, r, \ldots)$, and $Q_I(b, f, r, \ldots)$, where $1 \leq Q_O(\cdot), Q_V(\cdot), Q_I(\cdot) \leq 5$. These QoE models take five categories of inputs: *encoding settings*, *network conditions*, *video quality metrics*, *human factors*, and *game genre*. In total, our QoE models take 14 inputs. Table 4 summarizes the inputs, where: (i) encoding settings include *bitrate* b, *frame rate* f, and *resolution* r; (ii) network conditions encompass *throughput*, *frame loss rate*, *packet loss rate*, and *delay*; (iii) video quality metrics include *PSNR*, *SSIM*, and *VMAF*; (iv) human factors cover *GE* and *VE* levels; and (v) game genre is captured by *TI* and *SI*.

To understand their pros and cons, we build two classes of models: *per-game* and *general*, where the latter models are meant for all game genres. Since the former models are for each game, we remove the game genre (TI/SI) from their inputs. We consider four regression models as functions for predicting QoE, including **Polynomial regressor (Poly)** and decision tree-based regressors. Poly is chosen because it is a popular baseline model. Among decision tree-based regressors, **Random Forest (RF)**, **Gradient Boosting (GB)**, and **Ada Boosting (AB)** are widely used [7]. We use Scikit-Learn [42] to implement these regression models in Python. We adjust the key hyper-parameters of these regressors: (i) the *degree of polynomial* and *intersection-only* in Poly and (ii) the *number of estimators* and *minimum samples per-leaf* in decision tree-based solutions (RF/GB/AB). Figure 11 highlights the inputs and outputs of these regressor models.

In particular, for each regressor, we performed a grid search on the key hyper-parameters, resulting in 6 combinations for Poly and 35 combinations for RF/GB/AB, using the results from the QoE evaluations in the following steps. First, we need to split the dataset to evaluate the QoE models. This can be done in two ways: (i) some earlier work [11] split the dataset into training,

Table 5. Hyper-Parameters: AngryBird/BeatSaber/ArtPuzzle/General

Model	**Hyper-parameter**	
Poly	**Degree**	**Intersection**
	1 / 1 / 1 / 1	With / With / With / With
RF	No. Estimators	Minimum Samples
	200 / 200 / 200 / 350	2 / 4 / 2 / 2
GB	No. Estimators	Minimum Samples
	250 / 200 / 50 / 250	2 / 16 / 8 / 4
AB	No. Estimators	Minimum Samples
	350 / 350 / 100 / 100	4 / 2 / 8 / 16

validation, and testing sets, while (ii) others [3, 10, 69] split the dataset into training and testing sets only. We opt for the latter approach as we have fewer subjects than Fan et al. [11]. Second, we perform 3-fold cross-validation on overall quality by subjects. In particular, we take two-thirds of the subjects as training data and the rest as testing data. We consider all 495 possible train-test splits and evaluate the average performance in PLCC and SROCC. Third, we select the best hyper-parameters leading to the highest performance for the corresponding regressor models, as given in Table 5. Note that since the degree of Poly is one, it is equal to linear regression. Last, after determining the hyper-parameters, we include an additional metric, **R squared ($\mathbf{R^2}$)**, in addition to PLCC and SROCC, to compare the performance between per-game and general models, as well as across different regressor models. It is important to note that general models can be trained with more samples than per-game models. To ensure a fair comparison, we retain only one-third of random samples for general models, which is referred to as *adjusted general models.*

5.2 Resulting Models

We make the following observations on various QoE models considered by us:

- *Adjusted general models deliver good enough performance.* Table 6 gives the overall performance across per-game and adjusted general models. For all regressors, the adjusted general models achieve similar performance with per-game ones. Take RF as an example, the highest improvements of per-game models over general ones are merely 0.02 in R^2, 0.001 in PLCC, and 0.01 in SROCC on overall quality. *Hence, we employ the general model below, if not otherwise specified.*
- *Random forest achieves the best performance.* Next, we train our general models with all samples and give results in Figure 12. We find that the RF model performs the best. For example, in Figure 12(a), RF achieves up to 0.85 in R^2, 0.93 in PLCC, and 0.92 in SROCC on overall quality. A closer look depicts that among all inputs, the throughput and round-trip delay have the highest impacts with coefficients of 0.40 and 0.39, which are rather intuitive as they directly affect the response time and visual quality. Figure 13 plots the relationship between the predicted and ground-truth MOS. This figure depicts that RF results in a stronger linear correlation compared to Poly. Hence, *we adopt RF for building our QoE models in the rest of this article.*
- *Immersive level is relatively hard to model.* Compared to overall and visual quality, the performance of the immersive level is a bit lower, as illustrated in Figure 12. There may be two possible reasons. First, the immersive level is influenced more by game genres and subject preferences. This in turn makes their scores harder to be modeled by our regressors. The second reason is the impact of the QoE experiment duration. According to ITU-T recommendation

Table 6. QoE Modeling Results on Overall Quality: AngryBird/BeatSaber/ArtPuzzle/Adjusted General

Model	Metric		
	R^2	PLCC	SROCC
Poly	0.68 / 0.77 / 0.78 / 0.77	0.87 / 0.90 / 0.93 / 0.91	0.88 / 0.90 / 0.92 / 0.92
RF	0.80 / 0.84 / 0.84 / 0.82	0.93 / 0.93 / 0.93 / 0.93	0.91 / 0.88 / 0.91 / 0.90
GB	0.81 / 0.85 / 0.84 / 0.82	0.93 / 0.94 / 0.93 / 0.93	0.91 / 0.89 / 0.91 / 0.91
AB	0.83 / 0.84 / 0.80 / 0.81	0.94 / 0.94 / 0.92 / 0.91	0.92 / 0.88 / 0.90 / 0.90

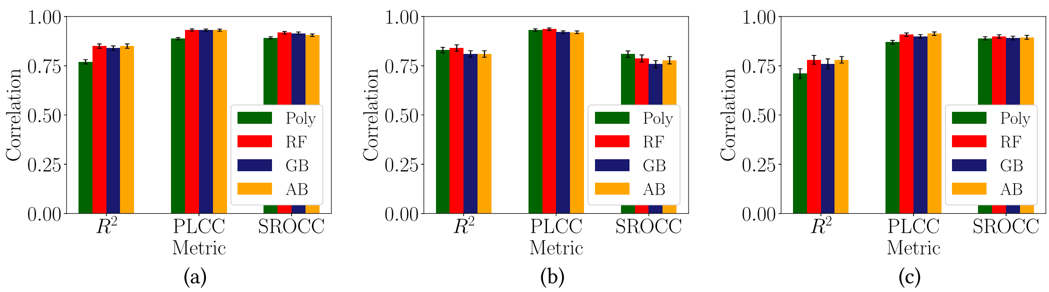


Fig. 12. Performance of general models on (a) MOS of overall quality, (b) MOS of visual quality, and (c) immersive level score.

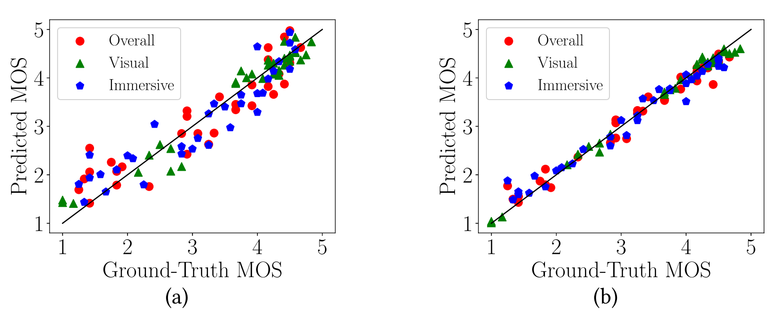


Fig. 13. Predicted vs. ground-truth MOS: (a) Poly and (b) RF.

P.809 [24], immersive levels are better investigated in experiments with longer durations. Since our QoE experiments duration of each session is not long, this might lead to more noise in ratings. With that said, we can still achieve acceptable performance of 0.78 in R^2, 0.91 in PLCC, and 0.90 in SROCC on immersive levels.

Although our models perform well when estimating the gamer's QoE, some of its inputs may be hard to measure at runtime. In particular, frame loss rate, PSNR, SSIM, and VMAF are measured externally from other tools in our testbed. To make our QoE model more suitable for real-life scenarios, we train *light-weight* models without these inputs. The light-weight models approximate the original ones and are denoted as: $\tilde{Q}_O(b, f, r, \dots)$, $\tilde{Q}_V(b, f, r, \dots)$, and $\tilde{Q}_I(b, f, r, \dots)$, where $1 \leq \tilde{Q}_O(\cdot), \tilde{Q}_V(\cdot), \tilde{Q}_I(\cdot) \leq 5$. We observe that the light-weight models produce QoE predictions fairly close to those from the original models. More specifically, the performance gaps between $Q_O(\cdot)$ and $\tilde{Q}_O(\cdot)$ are 0.02 in R^2, 0.01 in PLCC, and 0.02 in SROCC; those between $Q_V(\cdot)$ and $\tilde{Q}_V(\cdot)$ are 0.01 in R^2, 0.01 in PLCC, and 0.02 in SROCC; and those between $Q_I(\cdot)$ and $\tilde{Q}_I(\cdot)$ are 0.02 in R^2, 0.01 in PLCC, and 0.01 in SROCC. *Hence, we recommend and adopt the light-weight models in the rest of this article.*

6 QoE-driven Encoding Settings Adaptation

In this section, we develop an algorithm to select the optimal encoding settings under dynamic networks and systems.

6.1 Problem Formulation

We use encoding settings as control knobs, striving to find the optimal settings $e^* = (b^*, f^*, r^*)$, among all possible bitrate b, frame rate f, and resolution r, to maximize the expected QoE. More specifically, we periodically select and set e^* for every δ-sec adaptation time window. We choose δ empirically by investigating multiple time windows. If not otherwise specified, we let $\delta = 3$ seconds to strike a balance between the system overhead and the update frequency. While our approach is applicable to overall quality, visual quality, and immersive level using the proposed models $\tilde{Q}_O(\cdot)$, $\tilde{Q}_V(\cdot)$, and $\tilde{Q}_I(\cdot)$, we consider overall quality $\tilde{Q}_O(\cdot)$ for concrete discussion. Other QoE aspects can be readily adopted in the objective function if needed. The key constraint of our problem is end-to-end bandwidth, denoted as B. Notice that b represents *encoding* bitrate, which is smaller than *streaming* bitrate that accounts for various overheads, such as segmentation, protocol, and error correction. We use α to denote the overhead, proportional to the encoding bitrate. We use $\alpha = 15\%$ following Li et al. [37] if not otherwise specified. With the above symbols, we formulate our optimization problem as:

$$\begin{aligned} e^* &= \underset{e=(b,f,r)}{argmax}\ \tilde{Q}_O(b, f, r, \dots) \\ &\text{s.t. } (1+\alpha)b \le B. \end{aligned} \tag{1}$$

We note that the dots in $\tilde{Q}_O(\cdot)$ represent seven non-encoding-setting inputs of our QoE models (see Table 4). Among these seven inputs, four of them are *constants*: the subject's GE and VE levels and the game genre's TI and SI values. The remaining three inputs are *measured* in real-time, which are throughput p, delay d, and packet loss rate l. By solving the optimization problem once every adaptation window, our cloud VR gaming system adapts to the network and system dynamics in a QoE-aware fashion.

6.2 QoE-driven Adaptation Algorithm

Solving the optimization problem in Equation (1) is challenging for three reasons. First, QoE evaluations are time-consuming. Therefore, only a few (ten, more precisely) encoding settings were tested in our QoE evaluations, while additional encoding settings can and should be derived before solving the adaptation problem. Second, three measured inputs, which are throughput p, delay d, and packet loss rate l, vary in rather large ranges, leading to huge search space of optimal solutions. Last, numerically solving the QoE-driven optimization problem leads to excessive running time, which is not suitable for real-time cloud VR gaming.

To address the first challenge, we adopt quadratic functions to interpolate QoE of encoding settings that were not included in the QoE evaluations. More specifically, to *densify* the encoding settings, we fit a quadratic function along each dimension of bitrate, frame rate, and resolution. To ensure these quadratic functions to be monotonically non-decreasing, we add two *control bitrates* at 35 and 38 Mbps and two *control frame rates* at 84 and 90 fps. The QoE values of these control sample points are set to be the same as those of the closest encoding setting from our QoE evaluations. With these quadratic functions, we interpolate the QoE of encoding settings with $b \in \{2, 3, 4, 5, 6, 7, \dots, 31\}$, $f \in \{48, 60\}$, and $r \in \{1760 \times 960, 2496 \times 1376\}$ to increase the considered encoding settings from 10 to 42. For the second challenge, we discretize the range of each measured input into multiple bins to reduce the search space. Specifically, we employ a binning method based on data characteristics called *Freedman Diaconis* [15], which makes sure individual bins have enough data points. Following this method, we create 7, 7, and 3 bins for throughput p, delay d, and packet loss rate l, respectively. For the third challenge, to speed up the adaptation decisions, we construct a lookup table $\hat{Q}_O(b, f, r, \dots)$ for e^* using $\tilde{Q}_O(b, f, r, \dots)$. Because the lookup table is built offline, doing so incurs no runtime complexity with a small memory footprint ≤ 700 KB.

We propose a QDA algorithm based on the lookup table $\hat{Q}_O(b, f, r, \dots)$. The algorithm measures network conditions for individual frames and applies **Exponentially Weighted Moving Average (EWMA)** to filter out high-frequency noise. In particular, a 30% weight is assigned to the latest measurement. QDA algorithm is executed at the ALXR server once every δ seconds. First, the EWMA values are placed into bins. The algorithm then takes the middle points of the bins, human factors, and game genres and iterates through all feasible encoding settings that do not violate the bandwidth constraint. Among all feasible encoding settings, we choose e^* that maximizes $\hat{Q}_O(b, f, r, \dots)$, which is then used to reconfigure the video codec at the ALXR server. We note that this lookup can be done efficiently: throughout our experiments, the QDA algorithm always terminates in ∼ 20 ms on a commodity Intel i9 workstation.

7 Performance Evaluations

We evaluate our cloud VR gaming system, especially the QDA algorithm with an additional user study in this section. This user study is based on the QoE models constructed with the results obtained from the previous user study in Section 4.

7.1 Technical Setup

To drive our experiments, we adopt a real 5G network dataset [45], which contains throughput traces with two mobility patterns: *static* and *driving* and two applications: *file downloading* and *video streaming*. Because cloud VR gaming clients: (i) are static and (ii) incur a tremendous amount of network traffic, we select the static file-downloading trace[3] with the highest standard deviation to approximate the available bandwidth under the most challenging network conditions. The average bandwidth in this trace is 121 Mbps (σ = 88.44), and the maximum bandwidth reaches 254 Mbps. Built upon the trace, we consider three test scenarios: (i) C1, where the bandwidth is dedicated to one client, (ii) C5, where the bandwidth is equally divided among five clients, and (iii) C10, where the bandwidth is equally divided among 10 clients. As the number of clients increases, the bandwidth becomes more constrained. We note that our cloud VR gaming system ceases to work when the network bandwidth goes below 3 Mbps.[4] Hence, we scan through C1, C5, and C10, and skip any bandwidth samples < 3 Mbps. In total, 10.66%, 18.07%, and 31.20% bandwidth samples were skipped from C1, C5, and C10, respectively. The resulting traces are still long enough for our user study. We use Dummynet to emulate diverse network conditions in three scenarios: C1, C5, and C10.

We conduct a user study to compare our QDA algorithm against the following two baseline algorithms:

— **No Adaptation (NA)**. In vanilla ALXR, a gamer has an option to disable the bitrate adaptation algorithm altogether.
— **Delay Threshold-based Adaptation (DTA)**. ALXR provides a delay threshold-based bitrate adaptation algorithm. This algorithm dynamically adjusts the bitrate based on a target delay d_T and a tolerance interval d_Δ. It also keeps track of the streaming bitrate b_s at the ALXR server and considers a bitrate threshold b_T. The algorithm is executed once each frame. Specifically,

[3] We opt for the file-downloading traces for enough traffic loads.
[4] This is enforced by a watchdog mechanism.

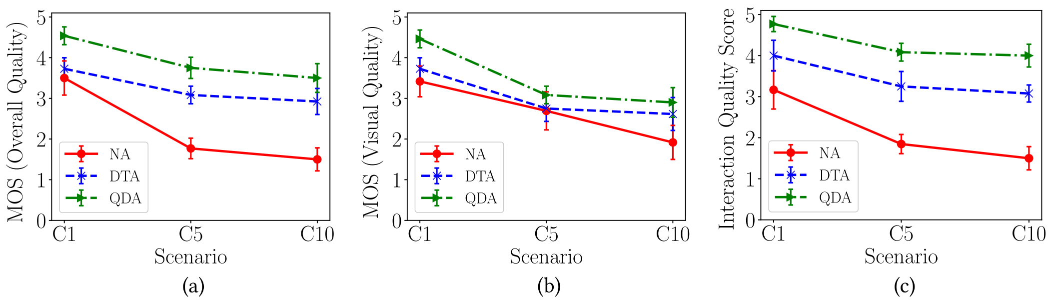


Fig. 14. Comparison of QoE quality among different adaptation algorithms for AngryBird: (a) MOS of overall quality, (b) MOS of visual quality, and (c) interaction quality score.

if the measured delay exceeds $d_T + d_\Delta$, the bitrate is decreased by 3 Mbps. Conversely, if the measured delay falls below $d_T - d_\Delta$ and the streaming bitrate b_s surpasses the threshold b_T, the bitrate is increased by 1 Mbps. We let $d_T = 12$ ms, $d_\Delta = 3$ ms, and $b_T = 0.7b_s$, following ALXR's default settings. Unlike our QoE-driven algorithm, DTA does not consider the frame rate and resolution when making decisions.

7.2 Test Method

We designed a new user study to evaluate the performance between QDA algorithm and two baseline algorithms. We utilized the same set of game genres mentioned in Section 4.1. The user study design is based on that in our first user study described in Section 4.3 (see Figure 5), but with a few changes on questionnaires, as summarized in Table 3. First, we removed the immersive level from the QoE questionnaire because we found that it was not easy for our subjects to properly rate the immersive levels given the relatively short gaming sessions. In addition, prolonging the gaming session is not an option due to potential subject fatigue. Second, we add a new question on *Interaction Quality (A)*, which has been shown to be crucial for interactive VR applications [50]. The ratings are also on a 1–5 scale using ACR. Moreover, focusing on the dynamics, the interaction quality is a better indicator to evaluate the effectiveness of adaptation algorithms in dynamic networks and systems. Third, we ask each subject to play a fraction (60%) of all sessions with different network scenarios, adaptation algorithms, and game genres to avoid subject fatigue. By doing so, each subject's user study duration is limited to 90 minutes. More specifically, among 27 total possible gaming sessions (3 network scenarios, 3 adaptation algorithms, and 3 game genres), each subject gets to play 16 random ones. Last, we dropped continue (C) from the QoE questionnaire to further reduce the user study duration.

We enlisted 20 subjects (17 males) aged between 20–26 years old. All of them passed the Snellen and Ishihara tests. Among these subjects, 6, 3, and 11 were categorized as novice, intermediate, and advanced gamers. In addition, eight of them had prior VE. In total, with 20 subjects and 16 sessions each, we completed 320 gaming sessions. On average, each combination of network condition, adaptation algorithm, and game genre accumulated 11.85 (standard deviation $\sigma = 0.80$) gaming sessions. In order to objectively assess the performance and study their relationship with subjective results, we measure two kinds of objective metrics: (i) *network metrics*, including delay and packet loss rate and (ii) *video quality metrics*, including PSNR, SSIM, and VMAF.

7.3 Results

MOS scores on overall, visual, and interaction quality. Figure 14 compares the overall, visual, and interaction quality achieved by various adaptation algorithms under different scenarios. Sample

Table 7. QoE Scores from NA/DTA/QDA Algorithms; Scenario C10

QoE	AngryBird	BeatSaber	ArtPuzzle
Overall Quality	1.50/2.92/3.50	1.17/1.92/3.25	1.42/2.23/2.92
Visual Quality	1.92/2.62/2.90	1.75/1.92/2.92	1.58/1.77/2.33
Interaction Quality	1.50/3.08/4.00	1.08/2.08/3.25	1.41/2.15/3.67

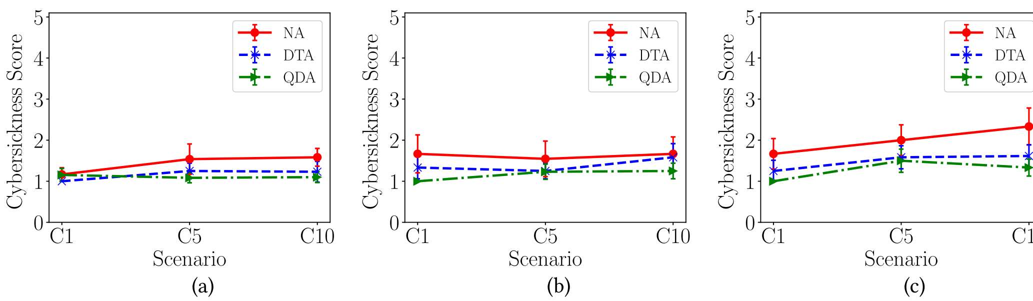


Fig. 15. Comparison of cybersickness score across different game genres: (a) AngryBird, (b) BeatSaber, and (c) ArtPuzzle.

results from AngryBird are shown; results from other game genres (BeatSaber and ArtPuzzle) are similar and omitted. Figures 14(a) and (c) depict that QDA algorithm delivers much better QoE in overall and interaction quality, compared to NA and DTA. The boost is particularly evident in the bandwidth-limited C10 scenario; the QoE gaps on (i) overall quality reach up to 2.00 ($\sigma = 0.45$) compared to NA, and up to 0.58 ($\sigma = 0.47$) compared to DTA; on (ii) interaction quality reach up to 2.50 ($\sigma = 0.40$) compared to NA, and up to 0.92 ($\sigma = 0.45$) compared to DTA. Regarding visual quality, Figure 14(b) reveals that the gaps are relatively smaller than those of overall and interaction quality. This discrepancy can be attributed to the need to reduce the encoding bitrate in challenging scenarios to prevent lagging and artifacts during gameplays.

Table 7 gives the QoE scores of overall, visual, and interaction quality under different game genres and adaptation algorithms under the bandwidth-limited C10 scenario. Compared to NA, the average improvements of our proposed QDA algorithm across all three game genres amount to an average of 1.86 ($\sigma = 0.38$) in overall quality, that of 0.97 ($\sigma = 0.45$) in visual quality, and that of 2.31 ($\sigma = 0.35$) in interaction quality. Compared to DTA, the average improvements stand at 0.87 ($\sigma = 0.44$) in overall quality, 0.61 ($\sigma = 0.45$) in visual quality, and 1.20 ($\sigma = 0.48$) in interaction quality. *Figure 14 and Table 7 confirm that our proposed QDA algorithm significantly improves the QoE scores on overall, visual, and interaction quality compared to the baseline algorithms.*

Cybersickness scores. Figure 15 presents the cybersickness scores achieved by different adaptation algorithms in different game genres. This figure shows that among the three game genres, QDA algorithm demonstrates much lower cybersickness scores under scenarios C5 and C10. Especially in C10, QDA algorithm's cybersickness scores are 0.63 ($\sigma = 0.42$) lower than those of NA and 0.25 ($\sigma = 0.34$) lower than those of DTA on average. These observations can be attributed to more effective adaptations made by our QDA algorithm: better-optimized encoding settings lead to less lagging and artifacts and thus better QoE. The observations in our user study are consistent with our previous study on simulator sickness [51], where lower visual quality resulted in higher simulator sickness scores. *Figure 15 confirms that our proposed QDA algorithm leads to relatively lower cybersickness scores, compared to the baseline algorithms.*

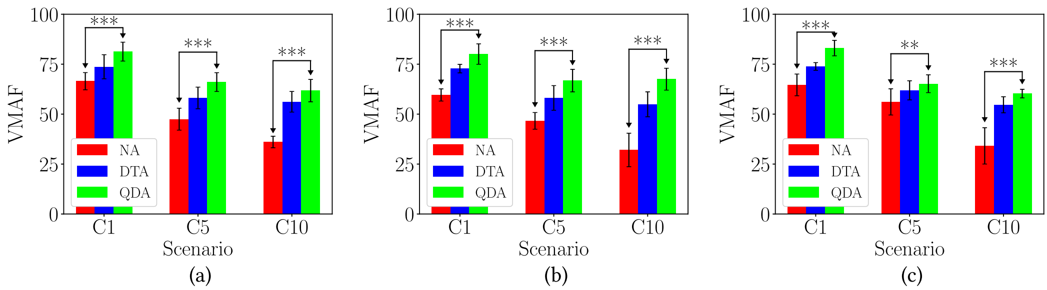


Fig. 16. Comparison of VMAF across diverse adaptation algorithms with different game genres: (a) AngryBird, (b) BeatSaber, and (c) ArtPuzzle. Significance values are defined as: $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$.

Table 8. Video Quality From NA/DTA/QDA Algorithms; Scenario C10

Metric	AngryBird	BeatSaber	ArtPuzzle
PSNR (dB)	23.99/29.96/31.25	22.15/26.89/31.77	19.57/24.81/27.31
SSIM	0.63/0.82/0.88	0.80/0.87/0.93	0.53/0.69/0.82
VMAF	36.04/56.17/61.75	32.06/54.91/67.49	34.10/54.68/60.26

Objective quality. Figure 16 reports sample VMAF results achieved by different adaptation algorithms with different game genres and under diverse scenarios. This figure illustrates that our QDA algorithm achieves the highest VMAF scores compared to other baseline algorithms. Further analysis through the Friedman test reveals significant differences among the three adaptation algorithms under the same network scenarios and game genres. The p-values are lower than 0.001, except for scenario C5 in ArtPuzzle. Even in that extreme case, we still observe a p-value lower than 0.01, indicating statistical significance.

Table 8 summarizes all video quality metrics, including PSNR, SSIM, and VMAF, achieved by different adaptation algorithms under the most challenging scenario C10. This table clearly shows that the proposed QDA algorithm leads to higher video quality than the two baseline algorithms; boosts by up to 9.62 dB in PSNR, 0.29 in SSIM, and 35.43 in VMAF are observed. *Figure 16 and Table 8 confirm that our proposed QDA algorithm significantly improves the objective video quality in PSNR, SSIM, and VMAF compared to the baseline algorithms.* Beyond the quality metrics, our QDA algorithm also demonstrates superior performance in network metrics. Specifically, the round-trip delay is 3 ms lower than DTA and 3.5 ms lower than NA, while the packet loss rate is 7.59% lower than DTA and 24.56% lower than NA on average under the most congested C10 scenario. These objective outcomes are consistent with the subjective QoE improvements.

Implication of game genres. Figure 17 illustrates the impact of different game genres on visual and interaction quality with our QDA algorithm under diverse scenarios, respectively; results from other algorithms (NA and DTA) are similar and omitted. Figure 17(a) illustrates that ArtPuzzle exhibits lower MOS than AngryBird and BeatSaber under bandwidth-limited scenarios, emphasizing its high demand for visual quality due to its quality-sensitive nature. Nevertheless, the MOS of visual quality in ArtPuzzle can maintain 2.33 ($\sigma = 0.27$) under bandwidth-limited C10 scenario with our QDA algorithm. Similarly, Figure 17(b) shows that BeatSaber has lower scores compared to AngryBird and ArtPuzzle under all scenarios, indicating its sensitivity to time and thus stringent requirements for interaction quality. However, with the help of our QDA algorithm, the interaction quality scores can still achieve 3.25 ($\sigma = 0.26$) even under the C10 scenario. *Figure 17 confirms that the results related to visual and interaction quality are consistent with our expectations outlined in*

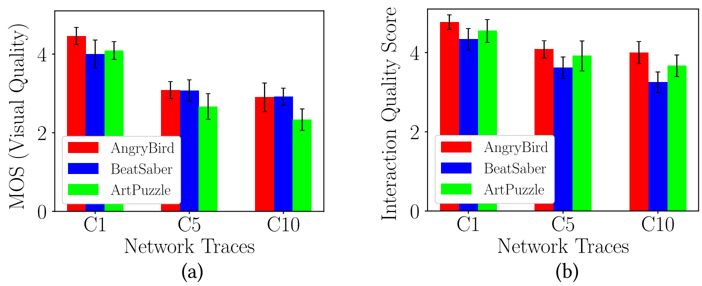


Fig. 17. Implication of different game genres with our QDA algorithm on (a) MOS of visual quality and (b) interaction quality score.

Section 4.1. Specifically, ArtPuzzle exhibits sensitivity to quality, BeatSaber shows sensitivity to time, and AngryBird demonstrates less sensitivity to both factors.

8 Conclusion

In this article, we developed and optimized a cloud VR gaming system, which has not been thoroughly studied in the literature. After conducting comprehensive QoE evaluations using a user study, we analyzed the impacts of different encoding settings and network conditions on gamer QoE scores across diverse game genres. The feedback from participants via questionnaires revealed novel insights into the correlation between gamer-perceived QoE and measurable QoS metrics. Based on the QoE evaluation results, we built general QoE models for all game genres using the RF regressor. The resulting QoE models are accurate achieving up to 0.93 ($\sigma = 0.02$) in PLCC and 0.92 ($\sigma = 0.02$) in SROCC. We also developed a QoE-driven adaptation algorithm called QDA to optimize the encoding settings under dynamic networks and systems. We conducted a second user study to evaluate the performance of our QDA algorithm in particular and the overall cloud VR gaming system in general. Compared to the existing NA and DTA algorithms, our proposed QDA algorithm leads to better cloud VR gamer QoE, e.g., it improves the MOS of overall quality by up to 1.86 ($\sigma = 0.38$) and reduces that of cybersickness by up to 0.63 ($\sigma = 0.42$) averagely across different game genres.

This article can be extended in multiple directions, including but not limited to:

— *Building QoE models for cybersickness scores*. Modeling cybersickness requires additional factors that were not considered in this article, such as subject differences, duration of each gameplay, and accumulated fatigue levels. More factors result in higher modeling complexity and thus need further investigation.
— *Predicting gamer willingness to continue playing*. Lebreton and Yamagishi [31, 32] examined the user's willingness to continue watching videos under the current QoS levels. We can generalize their approach to cloud VR gaming to create a model that determines: (i) whether the current measurable QoS metrics can retain each gamer and (ii) if network and system adaptations are necessary. The resulting models are extremely useful for cloud VR gaming service providers to maintain profitability.
— *Exploring cross-layer optimization on cloud VR gaming*. In **Mobile Edge Computing (MEC)**, cross-layer optimization exchanges insights across different layers, e.g., leveraging **Radio Network Information Service (RNIS)** [2] for live radio statistics. While RNIS has been employed for cross-layer optimization in flow control [9] and 360° VR video streaming [39], it has not been used in cloud VR gaming. Cross-layer optimization can be applied for better packet prioritization, congestion control, and resource allocation in cloud VR gaming for even better QoE.

—*Experimenting with alternative access networks.* Compared to costly wired networks and limited-range WiFi networks, 5G **Fixed Wireless Access (FWA)** offers high-speed Internet access to the home without deploying expensive cables. FWA supports high bandwidth and low delay and thus enables new AR/VR applications in rural areas [4, 30]. Experimenting with cloud VR gaming over FWA could reveal new challenges and opportunities in emerging 5G-based access networks.

References

[1] Maha Abdallah, Carsten Griwodz, Kuan-Ta Chen, Gwendal Simon, Pin-Chun Wang, and Cheng-Hsin Hsu. 2018. Delay-sensitive video computing in the cloud: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications* 14, 3s (2018), 1–29.

[2] Multi access Edge Computing (MEC) ETSI. 2019. Radio network information api. *ETSI GS MEC* 12 (2019), V2.

[3] Arslan Ahmad, Atif Bin Mansoor, Alcardo Alex Barakabitze, Andrew Hines, Luigi Atzori, and Ray Walshe. 2021. Supervised-learning-based QoE prediction of video streaming in future networks: A tutorial with comparative study. *IEEE Communications Magazine* 59, 11 (2021), 88–94.

[4] Khalid Aldubaikhy, Wen Wu, Ning Zhang, Nan Cheng, and Xuemin Shen. 2020. mmWave IEEE 802.11 ay for 5G fixed wireless access. *IEEE Wireless Communications* 27, 2 (2020), 88–95.

[5] Muhammad Shahid Anwar, Jing Wang, Wahab Khan, Asad Ullah, Sadique Ahmad, and Zesong Fei. 2020. Subjective QoE of 360-degree virtual reality videos and machine learning predictions. *IEEE Access* 8 (2020), 148084–148099.

[6] Wei Cai, Ryan Shea, Chun-Ying Huang, Kuan-Ta Chen, Jiangchuan Liu, Victor C. M. Leung, and Cheng-Hsin Hsu. 2016. A survey on cloud gaming: Future of computer games. *IEEE Access* 4 (2016), 7605–7620.

[7] Miguel Á. Carreira-Perpiñán and Arman Zharmagambetov. 2020. Ensembles of bagged TAO trees consistently improve over random forests, adaBoost and gradient boosting. In *Proceedings of the 2020 ACM-IMS Foundations of Data Science Conference* (*FODS '20*), 19–20.

[8] Giuseppe Cofano, Luca De Cicco, Thomas Zinner, Anh Nguyen-Ngoc, Phuoc Tran-Gia, and Saverio Mascolo. 2017. Design and performance evaluation of network-assisted control strategies for HTTP adaptive streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications* 13, 3s (2017), 1–24.

[9] Mamoutou Diarra, Walid Dabbous, Amine Ismail, Brice Tetu, and Thierry Turletti. 2022. RAPID: A RAN-aware performance enhancing proxy for high throughput low delay flows in MEC-enabled cellular networks. *Computer Networks* 218 (2022), Article 109357.

[10] Nagabhushan Eswara, S. Ashique, Anand Panchbhai, Soumen Chakraborty, Hemanth P. Sethuram, Kiran Kuchi, Abhinav Kumar, and Sumohana S. Channappayya. 2019. Streaming video QoE modeling and prediction: A long short-term memory approach. *IEEE Transactions on Circuits and Systems for Video Technology* 30, 3 (2019), 661–673.

[11] Ching-Ling Fan, Tse-Hou Hung, and Cheng-Hsin Hsu. 2022. Modeling the user experience of watching 360 videos with head-mounted displays. *ACM Transactions on Multimedia Computing, Communications, and Applications* 18, 1 (2022), 1–23.

[12] Ching-Ling Fan, Wen-Chih Lo, Yu-Tung Pai, and Cheng-Hsin Hsu. 2019. A survey on 360 video streaming: Acquisition, transmission, and display. *ACM Computing Surveys* 52, 4 (2019), 1–36.

[13] Jia-Wei Fang, Kuan-Yu Lee, Teemu Kämäräinen, Matti Siekkinen, and Cheng-Hsin Hsu. 2023. Will dynamic foveation boost cloud VR gaming experience. In *Proceedings of Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, 29–35.

[14] FreeBSD. 2002. A Live Network Emulation Tool. Retrieved August 5, 2024 from https://reurl.cc/LX8j34

[15] David Freedman and Persi Diaconis. 1981. On the histogram as a density estimator: L2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* 57, 4 (1981), 453–476.

[16] Jesus Gutierrez, Pablo Perez, Marta Orduna, Ashutosh Singla, Carlos Cortes, Pramit Mazumdar, Irene Viola, Kjell Brunnström, Federica Battisti, Natalia Cieplińska, Dawid Juszka, Lucjan Janowski, Mikołaj Leszczuk, Anthony Adeyemi-Ejeye, Yaosi Hu, Zhenzhong Chen, Glenn Van Wallendael, Peter Lambert, César Díaz, John Hedlund, Omar Hamsis, Stephan Fremerey, Frank Hofmeyer, Alexander Raake, Pablo César, Marco Carli, and Narciso García. 2021. Subjective evaluation of visual quality and simulator sickness of dhort 360° videos: ITU-T Rec. P. 919. *IEEE Transactions on Multimedia* 24 (2021), 3087–3100.

[17] Hua-Jun Hong, Chih-Fan Hsu, Tsung-Han Tsai, Chun-Ying Huang, Kuan-Ta Chen, and Cheng-Hsin Hsu. 2015. Enabling adaptive cloud gaming in an open-source cloud gaming platform. *IEEE Transactions on Circuits and Systems for Video Technology* 25, 12 (2015), 2078–2091.

[18] Chun-Ying Huang, Kuan-Ta Chen, De-Yu Chen, Hwai-Jung Hsu, and Cheng-Hsin Hsu. 2014. GamingAnywhere: The first open source cloud gaming system. *ACM Transactions on Multimedia Computing, Communications, and Applications* 10, 1s (2014), 1–25.

[19] Chun-Ying Huang, Cheng-Hsin Hsu, De-Yu Chen, and Kuan-Ta Chen. 2014. Quantifying user satisfaction in mobile cloud games. In *Proceedings of the Workshop on Mobile Video Delivery (MoViD)*, 1–6.

[20] Netflix Inc. 2019. VMAF – Video Multi-Method Assessment Fusion. Retrieved August 5, 2024 from https://reurl.cc/eL6LeR

[21] Mordor Intelligence. 2023a. Virtual Reality Gaming Market Size & Share Analysis – Growth Trends & Forecasts (2023–2028). Retrieved August 5, 2024 from https://reurl.cc/edbrxx

[22] S & P Global Market Intelligence. 2023b. Apple Pushes into AR/VR Hardware, and Meta Plants its Feet. Retrieved August 5, 2024 from https://reurl.cc/kaz5Rx

[23] ITU-T. 2008. *Recommendation ITU-T P.910: Subjective Video Quality Assessment Methods for Multimedia Applications.*

[24] ITU-T. 2018. *Recommendation ITU-T P.809: Subjective Evaluation Methods for Gaming Quality.*

[25] ITU-T. 2020. *Recommendation ITU-T G.1072: Opinion Model Predicting Gaming QoE for Cloud Gaming Services.*

[26] Michael Jarschel, Daniel Schlosser, Sven Scheuring, and Tobias Hoßfeld. 2011. An evaluation of QoE in cloud gaming based on subjective tests. In *Proceedings of the IEEE International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, 330–335.

[27] Robert S Kennedy, Norman E Lane, Kevin S Berbaum, and Michael G Lilienthal. 1993. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *International Journal of Aviation Psychology* 3, 3 (1993), 203–220.

[28] Asiya Khan, Is-Haka Mkwawa, Lingfen Sun, and Emmanuel Ifeachor. 2011. QoE-driven sender bitrate adaptation scheme for video applications over IP multimedia subsystem. In *Proceedings of the of IEEE International Conference on Communications (ICC)*, 1–6.

[29] Bill Krogfoss, Jose Duran, Pablo Perez, and Jan Bouwen. 2020. Quantifying the value of 5G and edge cloud on QoE for AR/VR. In *Proceedings of the of IEEE International Conference on Quality of Multimedia Experience (QoMEX)*, 1–4.

[30] Andrew Lappalainen and Catherine Rosenberg. 2022. Can 5G fixed broadband bridge the rural digital divide? *IEEE Communications Standards Magazine* 6, 2 (2022), 79–84.

[31] Pierre Lebreton and Kazuhisa Yamagishi. 2020. Predicting user quitting ratio in adaptive bitrate video streaming. *IEEE Transactions on Multimedia* 23 (2020), 4526–4540.

[32] Pierre Lebreton and Kazuhisa Yamagishi. 2023. Quitting ratio-based bitrate ladder selection mechanism for adaptive bitrate video streaming. *IEEE Transactions on Multimedia* 99 (2023), 1–14.

[33] Kuan-Yu Lee. 2022. The GitHub of User Study Testbed. Retrieved August 5, 2024 from https://reurl.cc/dLXnYg

[34] Kuan-Yu Lee. 2023. The GitHub of QoE Data and Model. Retrieved August 5, 2024 from https://reurl.cc/v0e0Ya

[35] Kuan-Yu Lee, Jia-Wei Fang, Yuan-Chun Sun, and Cheng-Hsin Hsu. 2023. Modeling gamer Quality-of-Experience using a real cloud VR gaming testbed. In *Proceedings of the of International Workshop on Immersive Mixed and Virtual Environment Systems (MMVE)*, 12–17.

[36] Jie Li, Ransheng Feng, Zhi Liu, Wei Sun, and Qiyue Li. 2018. Modeling QoE of virtual reality video transmission over wireless networks. In *Proceedings of the of IEEE Global Communications Conference (GLOBECOM)*, 1–7.

[37] Yen-Chun Li, Chia-Hsin Hsu, Yu-Chun Lin, and Cheng-Hsin Hsu. 2020. Performance measurements on a cloud VR gaming platform. In *Proceedings of the of Quality of Experience (QoE) in Visual Multimedia Applications (QoEVMA)*, 37–45.

[38] Ze-Nian Li, Mark S. Drew, and Jiangchuan Liu. 2014. *Fundamentals of Multimedia* (2nd ed.). Chapter 8,12.

[39] Yanwei Liu, Jinxia Liu, Antonios Argyriou, and Song Ci. 2018. MEC-assisted panoramic VR video streaming over millimeter wave mobile networks. *IEEE Transactions on Multimedia* 21, 5 (2018), 1302–1316.

[40] Netflix. 2021. The Github of Pre-Trained VMAF Models. Retrieved August 5, 2024 from https://reurl.cc/zrvgya

[41] Nvidia. 2020. The Official Website of NVIDIA CloudXR. Retrieved August 5, 2024 from https://reurl.cc/10djxm

[42] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[43] Pablo Pérez, Nuria Oyaga, Jaime J. Ruiz, and Alvaro Villegas. 2018. Towards systematic analysis of cybersickness in high motion omnidirectional video. In *Proceedings of the of International Conference on Quality of Multimedia Experience (QoMEX)*, 1–3.

[44] Stefano Petrangeli, Jeroen Famaey, Maxim Claeys, Steven Latré, and Filip De Turck. 2015. QoE-driven rate adaptation heuristic for fair adaptive video streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications* 12, 2 (2015), 1–24.

[45] Darijo Raca, Dylan Leahy, Cormac J. Sreenan, and Jason J. Quinlan. 2020. Beyond throughput, the next generation: A 5G dataset with channel and context metrics. In *Proceedings of the of ACM Multimedia Systems Conference (MMSys)*, 303–308.

[46] ALVR Repository. 2019. The GitHub of ALVR. Retrieved August 5, 2024 from https://reurl.cc/rrQ9y4

[47] ALXR Repository. 2021. The GitHub of ALXR. Retrieved August 5, 2024 from https://reurl.cc/xLOa5L

[48] Andreas Sackl, Raimund Schatz, Tobias Hossfeld, Florian Metzger, David Lister, and Ralf Irmer. 2016. QoE management made uneasy: the case of cloud gaming. In *Proceedings of the of IEEE International Conference on Communications Workshops (ICC)*, 492–497.

[49] Yun Shen, Peng Ding, Yuyin Xue, and Yaqi Song. 2022. A QoE-driven adaptive transmission scheme for streaming media service. In *Proceedings of the of IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 1–5.

[50] Ashutosh Singla. 2024. *Assessment of Visual Quality and Simulator Sickness for Omnidirectional Videos*. Springer, Cham.

[51] Ashutosh Singla, Stephan Fremerey, Werner Robitza, and Alexander Raake. 2017. Measuring and comparing QoE and simulator sickness of omnidirectional videos in different head mounted displays. In *Proceedings of the of International Conference on Quality of Multimedia Experience (QoMEX)*, 1–6.

[52] Ashutosh Singla, Steve Göring, Dominik Keller, Rakesh Rao Ramachandra Rao, Stephan Fremerey, and Alexander Raake. 2021. Assessment of the simulator sickness questionnaire for omnidirectional videos. In *Proceedings of the of Virtual Reality and 3D User Interfaces (VR)*, 198–206.

[53] Ivan Slivar. 2021. *Quality of Experience Driven Video Encoding Adaptation Strategies for Cloud Gaming Under Network Constraints*. Ph.D. Dissertation. Faculty of Electrical Engineering and Computing, University of Zagreb.

[54] Ivan Slivar, Lea Skorin-Kapov, and Mirko Suznjevic. 2016. Cloud gaming QoE models for deriving video encoding adaptation strategies. In *Proceedings of the of ACM International Conference on Multimedia Systems (MMSys)*, 1–12.

[55] Ivan Slivar, Mirko Suznjevic, and Lea Skorin-Kapov. 2015. The impact of video encoding parameters and game type on QoE for cloud gaming: a case study using the steam platform. In *Proceedings of the of IEEE International Workshop on Quality of Multimedia Experience (QoMEX)*, 1–6.

[56] Ivan Slivar, Sara Vlahovic, Matko Silic, Lea Skorin-Kapov, and Mirko Suznjevic. 2022. The impact of network and social context on quality of experience for competitive multiplayer virtual reality games. In *Proceedings of the of ACM Workshop on Games Systems (GameSys)*, 16–21.

[57] Ashkan Sobhani, Abdulsalam Yassine, and Shervin Shirmohammadi. 2017. A video bitrate adaptation and prediction mechanism for HTTP adaptive streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications* 13, 2 (2017), 1–25.

[58] Charles Spearman. 1961. The proof and measurement of association between two things. *American Journal of Psychology* 15 (1961), 72–101.

[59] Mirko Suznjevic, Justus Beyer, Lea Skorin-Kapov, Sebastian Moller, and Nikola Sorsa. 2014. Towards understanding the relationship between game type and network traffic for cloud gaming. In *Proceedings of the of International Conference on Multimedia and Expo Workshops (ICMEW)*, 1–6.

[60] Huyen T. T. Tran, Nam Pham Ngoc, Cuong T. Pham, Yong Ju Jung, and Truong Cong Thang. 2017. A subjective study on QoE of 360 video for VR communication. In *Proceedings of the of International Workshop on Multimedia Signal Processing (MMSP)*, 1–6.

[61] Huyen T. T. Tran, Nam Pham Ngoc, and Truong Cong Thang. 2021. Towards an overall QoE model for 360-degree video. In *Proceedings of the of IEEE International Conference on Communications and Electronics (ICCE)*, 327–331.

[62] Jonathan Vanian. 2023. Meta's Heated Rivalry with Apple Enters New Phase as the Tech Giants Go After Headsets. Retrieved August 5, 2024 from https://reurl.cc/dmMr6V

[63] Sara Vlahovic, Mirko Suznjevic, and Lea Skorin-Kapov. 2019. The impact of network latency on gaming QoE for an FPS VR game. In *Proceedings of the of International Conference on Quality of Multimedia Experience (QoMEX)*, 1–3.

[64] Sara Vlahovic, Mirko Suznjevic, and Lea Skorin-Kapov. 2022. A survey of challenges and methods for Quality of Experience assessment of interactive VR applications. *Journal on Multimodal User Interfaces* 16, 3 (2022), 257–291.

[65] Shaoxuan Wang and Sujit Dey. 2012. Cloud mobile gaming: Modeling and measuring user experience in mobile wireless networks. *ACM SIGMOBILE Mobile Computing and Communications Review* 16, 1 (2012), 10–21.

[66] Ziwei Wang and Xiuhua Jiang. 2019. A QoE-driven rate adaptation approach for dynamic adaptive streaming over HTTP. In *Proceedings of the of IEEE International Conference on Computing, Networking and Communications (ICNC)*, 224–229.

[67] Shun-Huai Yao, Ching-Ling Fan, and Cheng-Hsin Hsu. 2019. Towards quality-of-experience models for watching 360 videos in head-mounted virtual reality. In *Proceedings of the of IEEE International Conference on Quality of Multimedia Experience (QoMEX)*, 1–3.

[68] Saman Zadtootaghaj, Steven Schmidt, and Sebastian Möller. 2018. Modeling gaming QoE: Towards the impact of frame rate and bit rate on cloud gaming. In *Proceedings of the of IEEE International Conference on Quality of Multimedia Experience (QoMEX)*, 1–6.

[69] Huadi Zhu, Tianhao Li, Chaowei Wang, Wenqiang Jin, Srinivasan Murali, Mingyan Xiao, Dongqing Ye, and Ming Li. 2022. EyeQoE: A novel QoE assessment model for 360-degree videos using ocular behaviors. *ACM Transactions on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 1 (2022), 1–26.

Received 29 January 2024; revised 29 January 2024; accepted 8 July 2024