國立清華大學電機資訊學院資訊工程研究所
碩士論文
Department of Computer Science
College of Electrical Engineering and Computer Science
National Tsing Hua University
Master Thesis

建立模型用於調適雲端虛擬實境遊戲系統以最佳化使用者體驗

Building Gamer QoE Models to Adapt Cloud VR Gaming for Optimal User Experience

李冠佑
Kuan-Yu Lee

學號：111062576
Student ID:111062576

指導教授：徐正炘 博士
Advisor: Cheng-Hsin Hsu, Ph.D.

中華民國 113 年 3 月
March, 2024

國立清華大學
資訊工程研究所

碩士論文

建立模型用於調適雲端虛擬實境遊戲系統以最佳化使用者體驗

李冠佑

112

中文摘要

雲端虛擬實境遊戲將需要大量運算資源的VR遊戲轉移到資源豐富的資料中心。隨著虛擬實境設備的普及，雲端虛擬實境遊戲吸引了許多學術界以及工業界的關注。然而，要在雲端虛擬實境遊戲中確保良好的使用者體驗本質上是具有挑戰性的，因為虛擬實境遊戲的玩家會需要高視覺品質、短反應時間以及可忽略的不適程度。在這篇論文中，我們研究了雲端虛擬實境遊戲的使用者體驗並建立了一個擁有最佳化使用者體驗的系統。首先，我們建立了一個能夠模擬各種網路狀況的雲端虛擬實境遊戲實驗平台。利用這個實驗平台，我們進行了全面的使用者體驗評估，利用使用者研究來評估不同因素（如編碼設定、網路狀況和遊戲類型）對遊戲玩家使用者體驗的影響。其次，我們利用使用者體驗評估的結果構建了雲端虛擬實境遊戲的首個使用者體驗模型。我們的使用者體驗模型在Pearson線性相關係數上達到0.93（$\sigma = 0.02$），在Spearman等級相關係數上達到0.92（$\sigma = 0.02$），其中σ代表標準差。最後，我們利用我們的使用者體驗模型來動態調整實驗平台的編碼設定。廣泛的實驗顯示，與當前的系統相比，我們的可調適雲端虛擬實境遊戲系統在平均5分的平均意見分數中改善了：（i）整體品質0.87（$\sigma = 0.44$），（ii）視覺品質0.61（$\sigma = 0.45$）以及（iii）互動品質1.20（$\sigma = 0.48$）。

Abstract

Cloud Virtual Reality (VR) gaming offloads computationally-intensive VR games to resourceful data centers. As VR devices become increasingly popular, cloud VR gaming has attracted attention from both academia and industry. 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 thesis, 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).

致謝

在我的碩士生涯中，我想要感謝幾位重要的人。首先，我要感謝我的指導教授徐正炘，在我碩士一年級開始就協助我找到適合的研究方向。他提供漸進的建議以逐步擴展我的研究，每當我遇到困難或挑戰時，他總能給予我適當的建議來協助我解決問題，讓我的研究得以順利進行。除此之外，他還教導了我如何製作出高品質的簡報投影片，包括言簡意賅的摘要投影片和詳細介紹研究內容的投影片，並安排我在不同的場合對不同的人來介紹自己的研究，訓練我的口條以及彙整資料的能力，其中包括了資助我出國去會議上發表論文，讓我可以跟來自世界各地不同的研究者交流，增進自己的眼界。

接下來，我想要感謝我的實驗室夥伴，他們不管在研究上和實驗室日常生活上都給予了我很大的幫助。特別感謝我的MediaTek夥伴，方嘉瑋和孫元駿。除了一起參與了許多的會議和報告外，他們也協助我發表了我的第一片論文MMVE'23，給予我研究上的建議，讓我有機會可以出國去參加會議。

最後，我想感謝來自CWI的Dr. Ashutosh Singla和Dr. Pablo Cesar。他們寶貴的建議讓我順利地將MMVE'23延伸成TOMM的期刊論文，同時也是我的碩士論文。他們的建議讓我學習到了如何才能將論文以及實驗的做得更嚴謹，為我提供了寶貴的經驗和見解。

Acknowledgments

In my master's journey, I would like to express my gratitude to several individuals. Firstly, I would like to thank my advisor, Cheng-Hsin Hsu, who has been helping me since my first year in guiding me towards a suitable research direction. He provided progressive advice to gradually expand my research, and whenever I faced difficulties or challenges, he offered insightful suggestions to help me overcome obstacles, ensuring the smooth progress of my research. Additionally, he taught me the art of creating presentation slides of good quality, including concise summary slides and comprehensive slides detailing my research. He also arranged opportunities for me to present my research at different events, improving my presentation skills and data organization abilities. This included sponsoring my participation in conferences, facilitating valuable exchanges with researchers worldwide.

Next, I extend my appreciation to my labmates, who provided significant support both in research and daily life in the lab. Special thanks to my MediaTek project partners, Jia-Wei Fang and Yuan-Chun Sun. Beyond facing numerous project meetings and presentations together, they assisted in the development of my first paper for MMVE'23, offering valuable research insights and enabling me to attend conferences.

Lastly, I want to express my gratitude to Dr. Ashutosh Singla and Dr. Pablo Cesar from CWI. Their precious advice allowed me to seamlessly extend MMVE'23 into a journal paper for TOMM, which also served as my master's thesis. Their guidance taught me how to rigorously structure both the paper and experiments, providing me with valuable lessons and insights.

Contents

List of Figures

List of Tables

Chapter 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 [24] indicated that the VR gaming market is projected to demonstrate a Compound Annual Growth Rate (CAGR) of 32.75% until 2028. The same report also stated that the number of both VR and Augmented Reality (AR) gamers are 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 [25, 69]. 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 GPU power and battery capacity of standalone HMDs can detrimentally affect the gaming experience. 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*. Fig. 1.1 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 gaming experiences to VR gamers.

In particular, VR gamers require *short response time* and *high visual quality* when

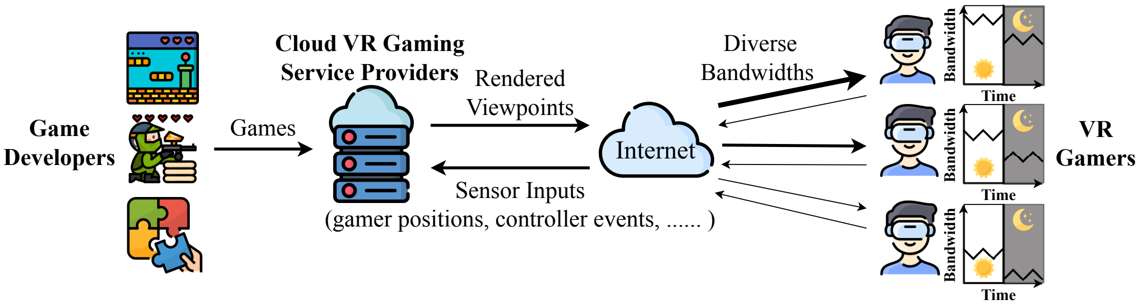


Figure 1.1: A typical cloud VR gaming system.

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 address these challenges, we build and optimize a cloud VR gaming system in a few steps. First, we construct a cloud VR gaming testbed to study the impact of different parameters on gamer Quality-of-Experience (QoE) scores. Second, we develop regression models to predict QoE scores given measureable Quality-of-Service (QoS) metrics, such as throughput, delay, and packet loss rate. Third, we leverage these QoE models to optimize gamer QoE with an adaptation algorithm in cloud VR gaming services.

1.1 Contributions

To fulfill the needs of VR gamers, this thesis extends our preliminary QoE evaluations [41], and makes the following contributions:

- We build an open-source cloud VR gaming testbed that enables us to emulate diverse and dynamic Wide Area Networks (WANs). We design and carry out QoE evaluations using a user study on this open-source testbed [39] 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 for the research community [40].
- We construct cloud VR gaming QoE models utilizing findings from our QoE evaluations to predict gamer QoE scores under various factors. Given measurable QoS metrics, such as throughput, delay, and packet loss rate, our QoE models achieve high correlation, reaching 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) [65]*, 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 ($\sigma = 0.38$) under congested networks. *Our proposed algorithm is a first of its kind, as QoE-driven adaptation of cloud VR gaming has never been done in the literature.*

1.2 Limitations

In a typical cloud gaming system, a single cloud server may need to support multiple gamers simultaneously engaging in gaming activities. This simultaneous usage can lead to negative interactive impacts, such as insufficient bandwidth. Additionally, the required bandwidth varies depending on the game genres being played. To simplify the complexity of the research problem, this thesis focuses solely on the gamer QoE when an individual is experiencing cloud VR gaming. Furthermore, due to the high time cost associated with QoE user studies, this thesis is constrained to consider a limited number of parameters and system conditions. We concentrate specifically on exploring the influence of individual parameters without accounting for their mutual interactions. This implies that the primary objective of this thesis is to gain a preliminary understanding of the impact of individual parameters on QoE, with the potential for future extensions to broader aspects.

1.3 Organization

The rest of this thesis is organized as follows. Ch. 2 gives background knowledge for remote rendering and the machine learning regression models we use in this thesis, encompassing Random Forest, Gradient Boosting, and Ada Boosting. In Ch. 3, we offer an extensive review of related work, including QoE evaluations, QoE modeling, and QoE-driven adaptation. Ch. 4 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 Ch. 5. Ch. 6 focuses on constructing QoE models using results from the user study using different machine learning regression models. The QoE-driven adaptation algorithm of encoding settings is developed in Ch. 7. Ch. 8 evaluates the performance of our QoE-optimized cloud VR gaming system compared to the baselines. Finally, we summarize the conclusions and list potential future works in Ch. 9.

Chapter 2

Background

In this chapter, we discuss knowledge related to this thesis, including remote rendering, Quality-of-Service versus Quality-of-Experience, Quality-of-Experience evaluations, and adaptation to system and network dynamics.

2.1 Remote Rendering

Remote rendering is a graphic processing technique that offloads rendering tasks originally intended for local devices to remote servers. With this method, rendering workloads operate on powerful servers, and the resulting images or frames are transmitted over the network to the user's device. This enables users to experience applications with high-performance requirements on devices with limited computational resources, overcoming constraints imposed by the hardware resources of local devices.

Remote rendering can be categorized based on several aspects: (i) streaming direction, (ii) network setting, and (iii) rendering content. When classified by streaming direction, remote rendering falls into two categories: one-way and bidirectional systems, often referred to as non-interactive and interactive systems. Non-interactive systems, prevalent in the movie and animation industry, do not need to handle user interaction and are commonly known as render farms [75]. Render farms serve as efficient solutions for distributing rendering tasks across multiple servers, concurrently accelerating the rendering process. In contrast, interactive systems [56] require real-time handling of user inputs to update the streamed rendering content. Applications like cloud gaming exemplify interactive systems, known for their stringent delay and quality requirements.

When classified by network setting, remote rendering can be performed under Local Area Network (LAN) and WAN. For instance, in cloud gaming, if the streaming occurs within a local network between the server and client, it is referred to as LAN cloud gaming. Otherwise, if the streaming involves the Internet, it is WAN cloud gaming. While

WAN cloud gaming faces more challenging network conditions due to the dynamic nature of the Internet, it also has the potential to leverage more powerful servers and accommodate a larger number of gamers.

Finally, when classified by rendering content, remote rendering can be categorized into 2D and 3D remote rendering systems. A 3D remote rendering system is typically associated with VR/AR, where both of them introduce additional challenges. For example, compared to a conventional cloud gaming system [20], a cloud VR gaming system [5] imposes higher requirements on both game delay and visual quality. This, in turn, signifies the need for increased computational resources. Therefore, remote rendering is crucial for VR/AR applications, where rendering workloads are offloaded to servers, ensuring smooth and immersive experiences on VR/AR headsets with diverse computational capabilities.

2.2 Quality-of-Service versus Quality-of-Experience

QoS refers to a set of measurements and standards that ensure the performance, reliability, and efficiency of a network or service [78]. It plays a crucial role in maintaining a consistent and satisfactory user experience by managing the delivery of data traffic, incorporating various parameters such as bandwidth, delay, and packet loss rate. These metrics collectively define the overall quality of a communication system. In networking, QoS mechanisms prioritize specific types of traffic to meet distinct service requirements. This is particularly significant in scenarios where different applications or services share the same network infrastructure, such as voice over IP (VoIP), video streaming, and online gaming. By implementing QoS policies, resources can be allocated appropriately, ensuring that different users and applications receive suitable treatment.

QoE is a comprehensive metric that shows the overall satisfaction and perception of users when interacting with a particular service or application [30]. Unlike QoS, which primarily focuses on technical aspects like network performance, QoE takes the user's subjective experience into consideration. QoE encompasses various factors such as system, human, and context factors during the interaction with a service or application [47]. It reflects how well the user's expectations align with their actual experience and satisfaction levels. System factors, such as bandwidth, resolution, and jitter, pertain to technical aspects influencing the service or application's quality. Human factors, like gender, age, and skills, involve user characteristics affecting perceived quality. Context factors, including task type, duration, and location, encompass situational properties describing the user's environment. Understanding and optimizing QoE are crucial for service providers to deliver services and applications that not only meet technical specifications but also resonate positively with users, fostering loyalty and user engagement.

2.3 Quality-of-Experience Evaluations

QoE evaluations, also known as QoE assessments, involve the evaluation of user experience when interacting with specific services and applications [74]. Generally, these evaluations can be categorized into two main types: subjective and objective evaluations. Subjective evaluations directly collect users' feelings and perceptions during their experience, relying on their firsthand feedback. On the other hand, objective evaluations utilize specific objective metrics to match and quantify the observed quality, aiming to assess QoE using measurable criteria.

Subjective evaluations are commonly conducted through user studies, which can be categorized into various types based on different experimental procedures and settings. For instance, concerning the stimulus times in experiments, there are two primary types: single-stimulus tests and double-stimulus tests. In a single stimulus test, a representative method is Absolute Category Rating (ACR) [27]. In this approach, participants observe a single test sequence and provide ratings on a discrete scale ranging from 1 to 5, representing bad to excellent. On the other hand, in a double stimulus test, notable methods include Degradation Category Rating (DCR) [27] and Double Stimulus Impairment Scale (DSIS) [26]. In this method, users view a pair of test sequences each time, with one being a reference and the other being the object for evaluation. Participants rate the impairment significance between these test sequences on a five-level scale, assessing the degree of impairment. Another classification is based on test modality, according to ITU-T recommendation P.809 [28], distinguishing between passive and interactive tests. Passive tests involve subjects passively receiving experimental content, such as watching 360° videos. In contrast, interactive tests require users to interact with the experiment, such as playing games. Further classifications include lab tests and crowdsourcing tests based on the experimental environment. Lab tests are conducted in a controlled laboratory setting, while crowdsourcing tests leverage a diverse pool of subjects and offer a more realistic testing environment. The choice of experimental methods and environments should be carefully considered based on the specific goals and content of the study to ensure the reliability of the experiment.

While subjective evaluations provide direct access to user ratings, they come with high time costs and also the data are not reproducible. These limit its application and make it unsuitable for real-time usage due to its low efficiency. As an alternative QoE evaluation method, objective evaluations offer a more time-efficient approach. Objective evaluations often employ the computing of objective metrics as indicators of the QoE. For example, in the context of video streaming, using metrics like PSNR, SSIM, or VMAF to evaluate QoE can show the quality perceived by the user and expose the user's QoE to some degree. Though this method is straightforward, it is often less accurate since user's QoE cannot be adequately captured by these objective metrics.

To leverage the strengths of both evaluation methods, a hybrid approach can be employed [45]. This method acts as an objective quality predictor but relies on subjective results from previous user studies. Hybrid methods often utilize polynomial regression, machine learning, or even deep learning to establish mapping between input metrics and the predicted QoE output, which is known as QoE modeling. These inputs cover various aspects, including encoding settings such as bitrate, frame rate, resolution, and network metrics like throughput, delay, and packet loss rate. Even human factors and application content can be utilized as inputs for predicting QoE. These models empower service providers to efficiently assess the user's QoE, incorporating insights from user ratings to ensure that the predicted results closely align with the actual experiences of users.

2.4 Adaptation to System and Network Dynamics

Adaptation to System and Network Dynamics refers to the capability of a system or network to adjust and optimize its operation in response to changes in the surrounding environment. The goal of employing adaptive processes in the system is to enhance the overall system performance and also the user experience by adapting system and network parameters dynamically. System dynamics adaptation involves adjusting computational resources and also the settings of the applications. On the other hand, network dynamics adaptation encompasses adjusting the bitrate and the routing path of the data.

For service providers, optimizing costs and maximizing profits within the constraints of limited resources is a crucial challenge. System dynamics adaptation, such as cloud resource adaptation [22], becomes a significant aspect in addressing this challenge. This involves adapting components like CPU, VM migration, and storage to balance the trade-off between maintaining users' QoE and reducing costs, including power consumption. Service providers can employ methods such as control theory or machine learning to dynamically adapt their system dynamics.

As for network dynamics adaptation, it is crucial for service providers, especially for video streaming service providers [35]. These services often need to support a large number of users, making it essential to maintain user's QoE under limited bandwidth. Rate adaptation is a mechanism addressing this situation, dynamically adjusting the bitrate used for transmission based on the current network conditions, content being transmitted, and resources available to the service provider. The goal is to maximize resource efficiency while meeting user requirements. There are various methods for rate adaptation, primarily categorized into push-based and pull-based adaptation. Push-based adaptation operates on the server side, and so it is suitable for applications like cloud gaming. On the other hand, pull-based adaptation runs on the client side and is commonly used in video streaming services, such as Dynamic Adaptive Streaming over HTTP (DASH).

By continuously monitoring and adapting to evolving conditions, systems can maintain a satisfactory user's QoE even in the face of fluctuating network conditions. This adaptability is particularly significant in real-time applications such as video streaming, cloud gaming, and VR applications, where delay and quality are critical components of user satisfaction.

Chapter 3

Related work

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

3.1 QoE Evaluations

Several QoE evaluations have been conducted through user studies to assess gamer QoE of cloud gaming. For example, Jarschel et al. [31] 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. [54] 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. [62] adopted the same platform for another user study of different encoding settings with two game genres. GamingAnywhere [20] 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 gaming experience [21]. *Different from our current work, these papers [21, 31, 54, 62] 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. [70] 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 delay. Slivar et al. [63] 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 [63, 70] only focused on local VR gaming rather than cloud VR gaming.*

For cloud VR gaming, we designed a remote VR gaming testbed [43] on the basis of *Air Light VR (ALVR)* [5], and conducted a user study under different network conditions using three game genres. We reported that insufficient bandwidth and 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)* [6]. Specifically, we conducted a small user study [15] by varying foveation parameters, including the foveal region size and the compression ratio of the peripheral area. *The current thesis presents more comprehensive QoE evaluations focusing on gamer QoE, which enables the construction of QoE models and QoE-driven adaptation algorithms.* The preliminary results of our QoE evaluations were given in Lee et al. [41].

3.2 QoE Modeling

Several research groups have built QoE models for cloud gaming. For example, Wang and Dey [72] proposed a QoE model that considers game genres, encoding settings, video quality, response time, and packet loss rates as inputs to predict mobile gaming experience. They derived impairment functions from the QoE evaluations to predict the Game Mean Opinion Score (GMOS) of each gamer. Slivar et al. [61] modeled game-dependent QoE using a quadratic function, which takes the frame rate and bitrate as inputs. Furthermore, they considered game genres and gaming experience in their models. Different from directly using the bitrate and frame rate as inputs, Zadtootaghaj et al. [77] introduced structural QoE models based on several intermediate factors derived from other raw inputs. ITU-T recommendation G.1072 [29] presented an opinion model for predicting cloud gaming QoE scores. The model provides 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 [29, 61, 72, 77] considered traditional cloud gaming rather than cloud VR gaming.* Several works [7, 42, 68, 76] derived QoE models for consuming 360° VR videos; however, little has been done to VR gaming. Although Krogfoss et al. [34] presented a video and a gaming QoE model based on parameters like the 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.

3.3 QoE-Driven Adaptation

Several works have been done to adapt the bitrate on the fly in video streaming sessions. For example, Cofano et al. [10] and Sobhani et al. [64] proposed bitrate adaptation algorithms for HTTP Adaptive Streaming (HAS) systems. *Different from our work, these adaptation algorithms are not QoE-driven.* For QoE-driven adaptation, several studies [52, 55, 73] 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. [33] 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 [10, 33, 52, 55, 64, 73] are for video streaming rather than more challenging cloud gaming systems, and most of them only take bitrate into consideration, excluding frame rates and resolutions.*

QoE-driven adaptation in cloud gaming has only been recently considered, e.g., Slivar [60] introduced three adaptation algorithms for the bitrate and frame rate. These algorithms were built upon the findings in their QoE evaluations. Our prior work [19] facilitated adaptive cloud gaming in GamingAnywhere [20] 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 thesis introduces a QoE-driven adaptation algorithm in cloud VR gaming instead of traditional cloud gaming [19, 60].*

Chapter 4

Building a Cloud VR Gaming System

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

- **360° Video-on-Demand (VoD) [14].** 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 [8].** Traditional cloud gaming systems like GamingAnywhere [20] 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 novel immersive experiences. Therefore, QoE models are essential in cloud VR gaming to efficiently estimate the gamer QoE.
- **Locally-rendered VR applications [71].** 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 thesis, 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 [49] and ALVR [5]. NVIDIA CloudXR supports streaming XR content using the OpenVR Application Programming Interface (API) for Android and Windows devices. Unfortunately, NVIDIA only makes CloudXR's client side open-source. This prevents researchers from integrating their innovations into the server side for experiments. In contrast, ALVR is an open-source project on both the server and client sides. Vanilla ALVR streams game scenes from PCs to HMDs over LANs. ALVR uses OpenVR API to obtain game scenes from SteamVR games. However, OpenVR runtime only supports a limited number of HMD models. ALXR [6] is an extension to ALVR, which adopts OpenXR on the client side to support more HMD models. Hence, we built our open-source cloud gaming system on top of ALXR.

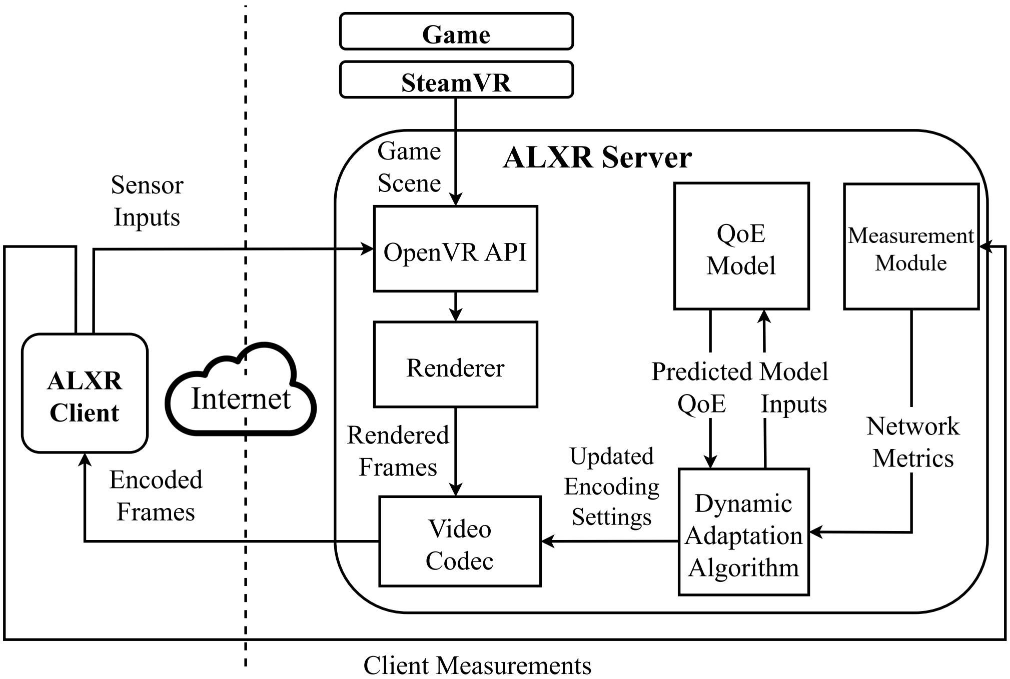


Figure 4.1: Cloud VR gaming architecture.

Fig. 4.1 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 and client measurements back to the server. According to the sensor inputs from the client, the ALXR server replays the gamer's motions and extracts new game scenes. Meanwhile, the client measurements are collected in a measurement module, and several network metrics are computed. The dynamic adaptation algorithm utilizes these metrics to assess the predicted QoE from the QoE model and updates the encoding settings of the video codec. Subsequently, the

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

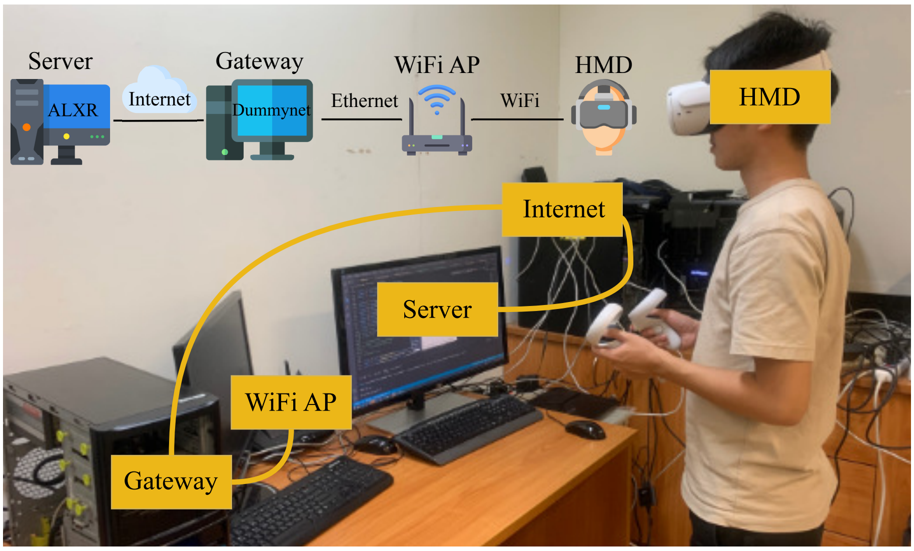


Figure 4.2: Cloud VR gaming testbed.

ALXR server encodes the frame with the new encoding settings and sends the updated frames to the client side.

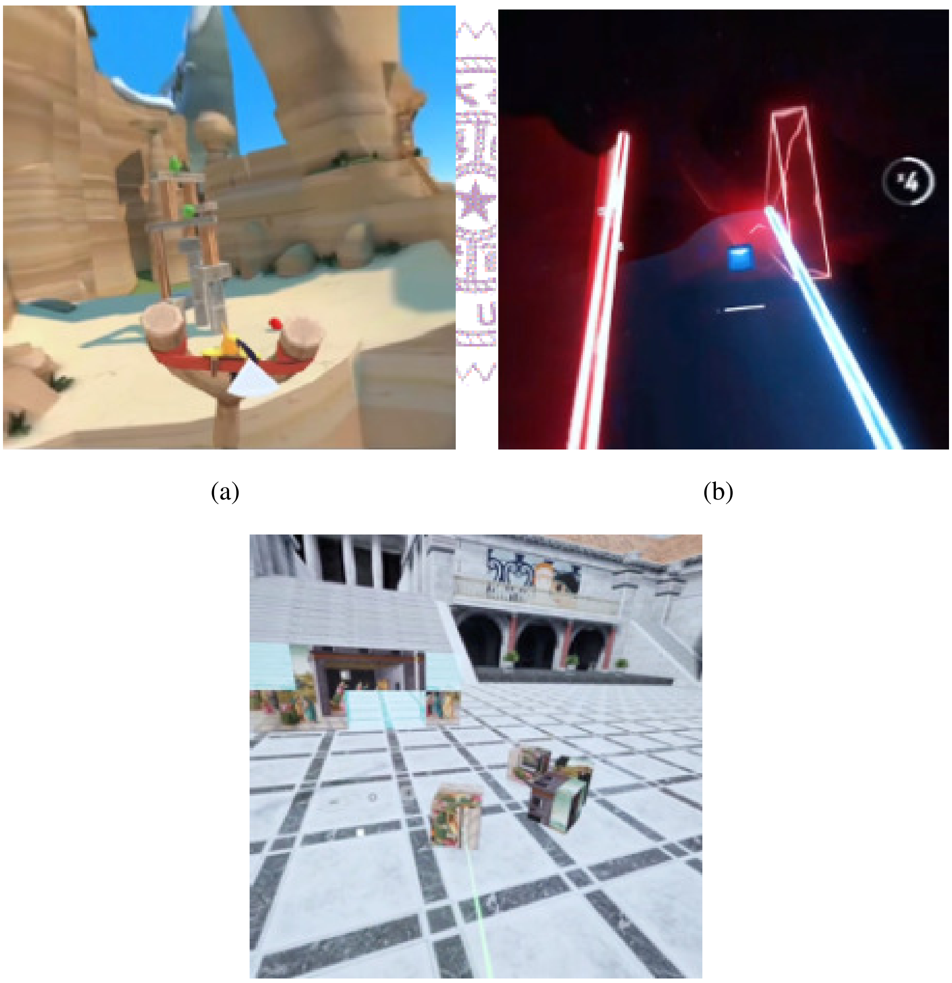
Fig. 4.2 shows our ALXR-based cloud VR gaming testbed. We use a Windows 10 PC as our server. It comes with an Intel Core i9 CPU, 64 GB RAM, an NVIDIA GeForce RTX 3080 Ti GPU, and is connected to the Internet through a GigE cable. We use a Meta Quest 2 HMD as our client. It comes with a Qualcomm Snapdragon XR2 CPU, 6 GB RAM, 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 [16] 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 [39]. 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 gaming experience. We addressed these challenges in three steps. In Ch. 5, we conduct comprehensive QoE evaluations using a user study on our open-source cloud VR gaming system. In Ch. 6, we analyze gamer QoE scores and build corresponding QoE models. In Ch. 7, we incorporate the QoE models to enable QoE-driven dynamic adaptation of encoding settings at runtime.

Chapter 5

QoE Evaluations

In this chapter, we conduct QoE evaluations using a user study to learn about the effects of various factors on gamer QoE scores.



(a) (b)

(c)

Figure 5.1: Sample scenes of three considered games: (a) AngryBird, (b) BeatSaber, and (c) ArtPuzzle.

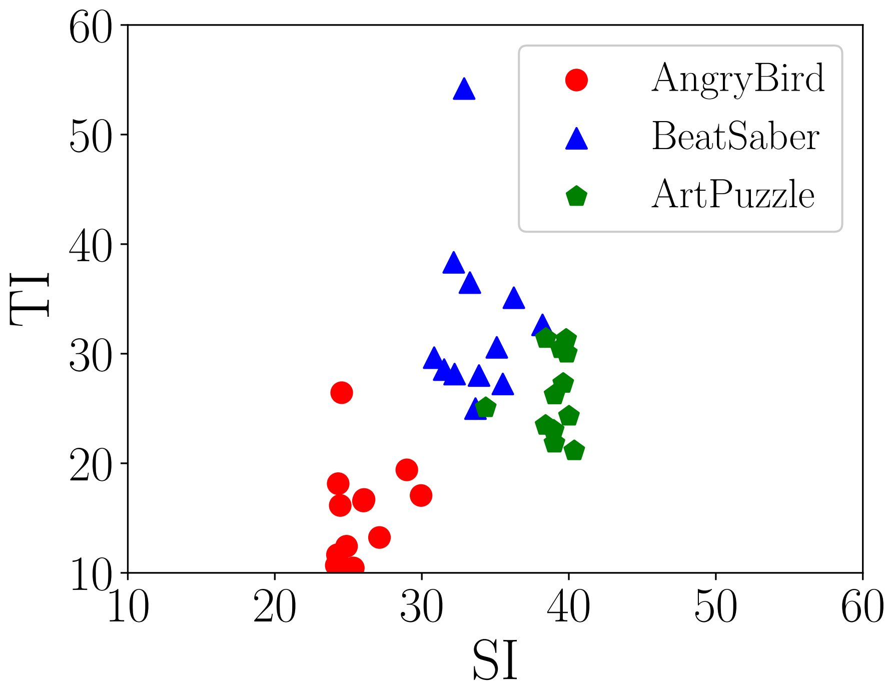


Figure 5.2: TI versus SI values from different game genres.

5.1 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)* [27] to characterize game genres following prior works [61,66]. 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 properties of objects, such as colors. After considering multiple candidate VR games, we chose the following three VR games, as shown in Fig. 5.1:

- **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 Ch. 5.3) in Fig. 5.2. 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 5 subjects to adjust the parameter values. In this test, 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 a 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 of 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 thesis.
- **Delay.** The local Round-Trip Time (RTT) 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.

5.2 Measurement Methodology

We measure the following metrics:

- **Throughput.** The receiving speed at the client, which is denoted as p.
- **Frame loss rate.** The fraction of lost frames.
- **Delay.** The round-trip delay between the server and client, which is denoted as d.

- **Packet loss rate.** The fraction of lost packets, which is denoted as l.
- **Peak Signal-to-Noise Ratio (PSNR).** A widely used video quality metric in the decibel scale [44, Ch. 8].
- **Structural Similarity Index (SSIM).** Another video quality metric that takes human perception into consideration [44, Ch. 12].
- **Video Multimethod Assessment Fusion (VMAF).** A learning-based video quality metric based on human perception [23][1].

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

Table 5.1: 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

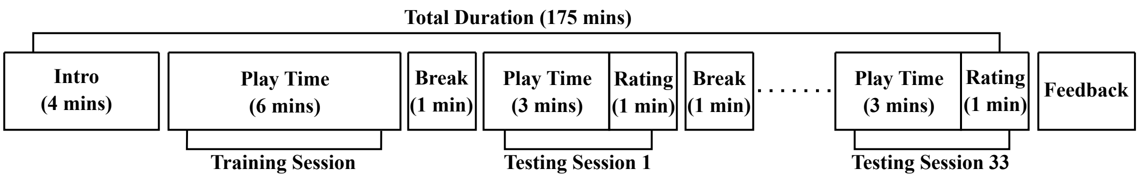


Figure 5.3: Procedure of the user study.

[1] We followed the recommendation for 360° video scenarios [48] and employed the default 1080p model.

Table 5.2: 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 5.3: QoE Questionnaire for QoE User Study

QoE	Question	Rating	QoE Experiments (Ch. 5)	Performance Evaluations (Ch. 8)
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)	How is your assessment about 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)	×	✓

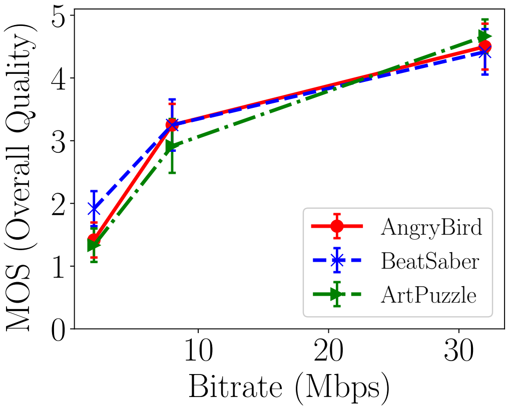
5.3 User Study

We recruited 12 subjects to conduct our user study, of whom 10 were males. All subjects were college students between 20–25 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 5.1, we categorized all subjects into three GE levels: (i) novice ($<$ 1 hour game time per week), (ii) intermediate ($\geq$ 1 and $<$ 5 hours), and advanced ($\geq$ 5 hours). We enumerated the GE levels into 1, 2, and 3 for the sake of presentation. There were 3, 3, and 6 gamers in the GE levels, respectively. Table 5.1 also shows that we classified all subjects into two VE levels using a Boolean value, where 0 means no prior VR experience.

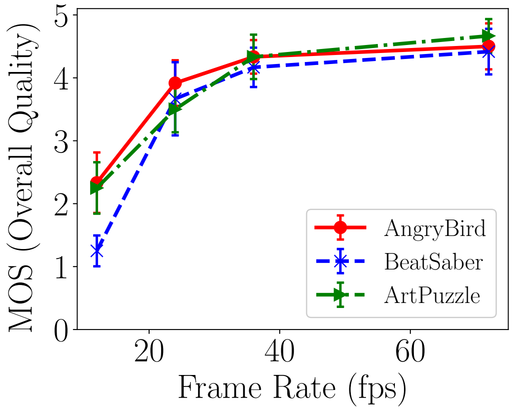
Fig. 5.3 shows the procedure of our user study. At the beginning, we provided an introduction to each subject. In the training session, the players played all three games to get familiar with the HMD and controllers. The game scenes/levels we used in the training sessions were different from those in the testing sessions. To avoid fatigue, we only varied one factor at a time, leading to 11 scenarios (sessions) for each game. Table 5.2 lists all the scenarios. Since we had three considered games, each subject underwent 33 sessions. There was a 1-minute break after each session. According to ITU-T recommendation P.809 [28], we conducted short interactive tests. Due to some game loading time, we set the playtime for each session to 3-minutes. The order of sessions was random to avoid the learning effect. We recorded each subject's inputs for our QoE questions given in Table 5.3. This table consists of all questions used in this chapter (upper half) and in the performance evaluations chapter (Ch. 8). In this chapter, we asked all questions except for the last row of the table. Particularly, there are five questions [51, 57, 61, 67]: *Overall Quality (O)*, *Visual Quality (V)*, *Immersive Level (I)*, *Cybersickness (S)*, and *Continue (C)*. The ratings are on a 1–5 scale using 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 [32] to prevent the prolonged duration of each session, which would limit the number of tested conditions [59]. Furthermore, our focus was on the mean cybersickness score, and thus the longer cybersickness questionnaires may not be necessary [18]. 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 analyze the results below.

5.4 Results

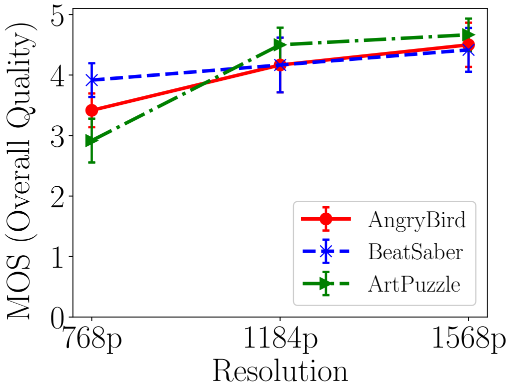
Bitrate affects the gamer QoE the most among other encoding settings. Fig. 5.4 gives the MOS scores of overall quality under different encoding settings. Slopes in Fig. 5.4(a) are generally steeper compared to those in Figs. 5.4(b) and 5.4(c), showing that the bitrate imposes the most significant impact 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



(a)



(b)



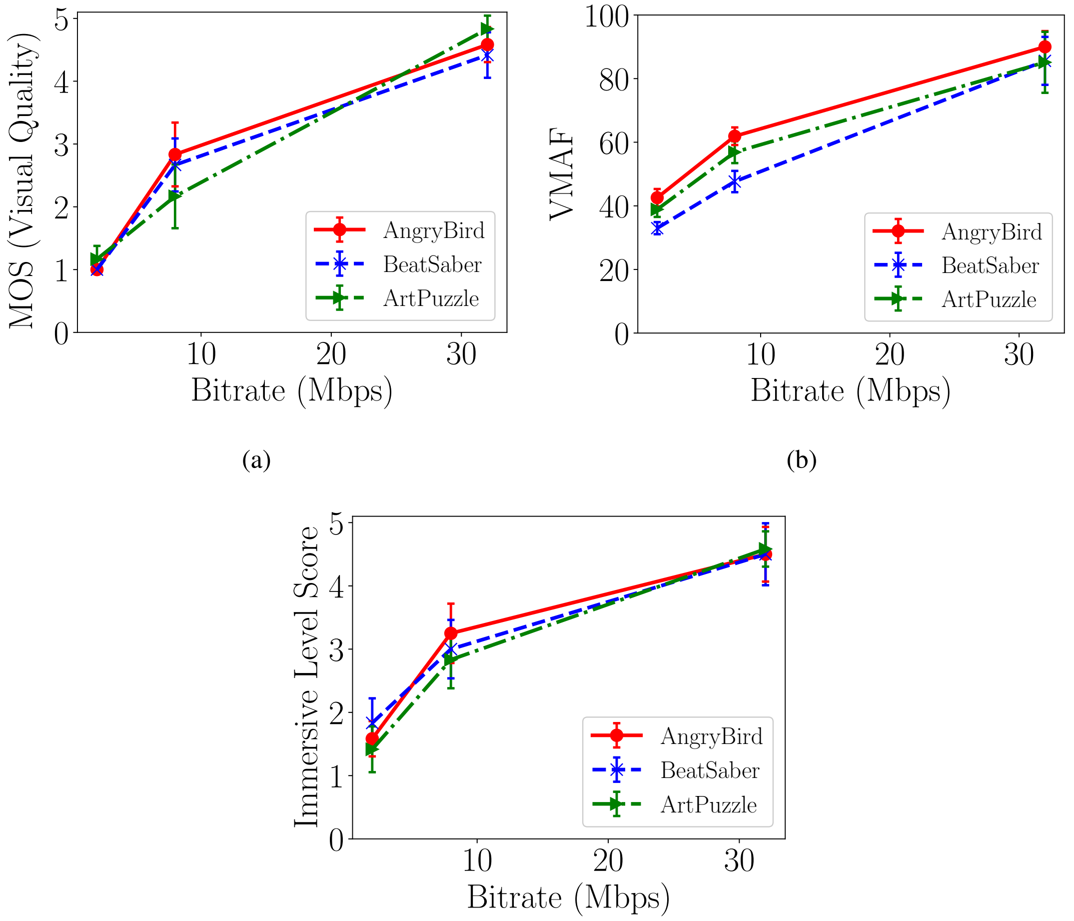
(c)

Figure 5.4: 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).

that all these p-values are below 0.001, demonstrating clear statistical difference.

MOS growth rate decelerates as bitrate increases. Fig. 5.5 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 Fig. 5.5(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 Fig. 5.4(a). These observations indicate that as the bitrate increases, the growth rate of MOS decelerates gradually.

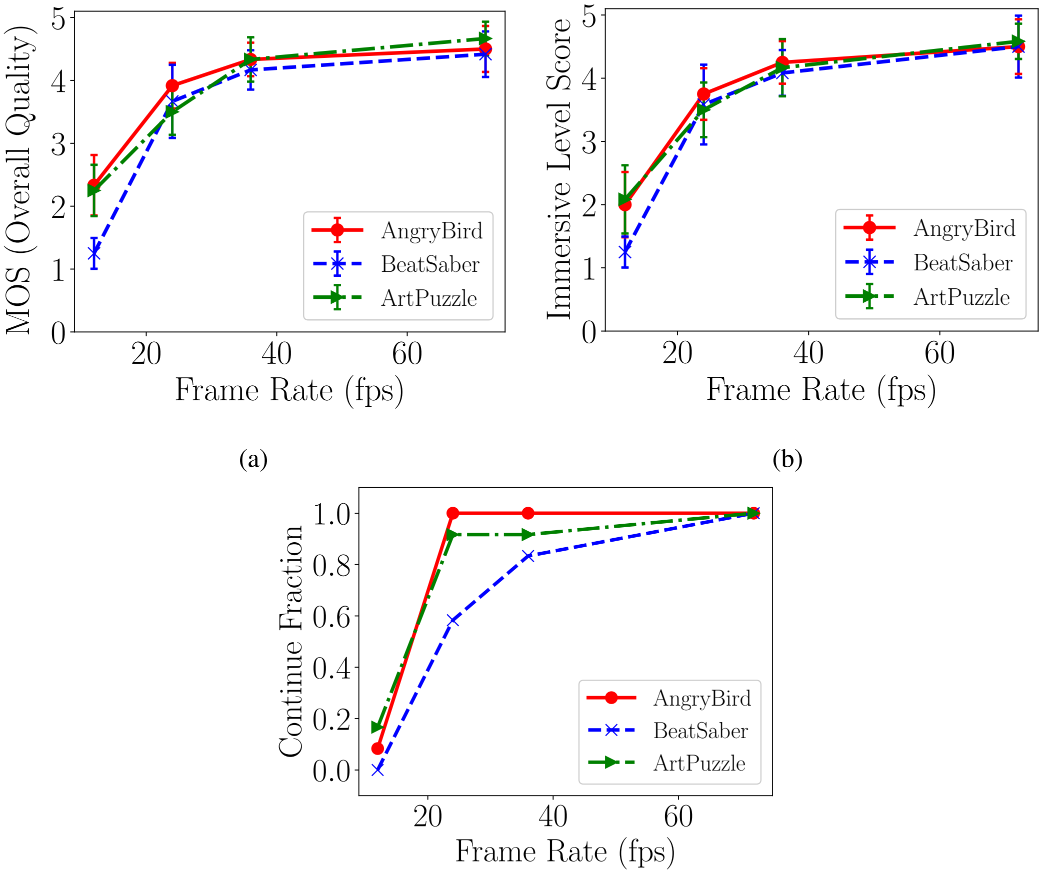
Different game genres have different requirements. Fig. 5.4(a) reveals that MOS of overall quality is more sensitive in ArtPuzzle under different bitrates. The same can be said with visual quality and immersive level in Figs. 5.5(a) and 5.5(c), compared to other



(a) (b) (c)

Figure 5.5: Implications 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.

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 below 0.001 after conducting the Wilcoxon signed-rank tests, showing statistical difference. This is intuitive, as ArtPuzzle needs higher visual quality due to its texture details. Fig. 5.6 reports the influence of varying frame rates. Figs. 5.6(a) and 5.6(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. Fig. 5.6(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. Fig. 5.7 presents the implication of extra delay on overall quality and immersive level. Similar to Fig. 5.6, BeatSaber is more sensitive to injected delays, as gamers may not react in time. The p-values between 0 and 500 ms of



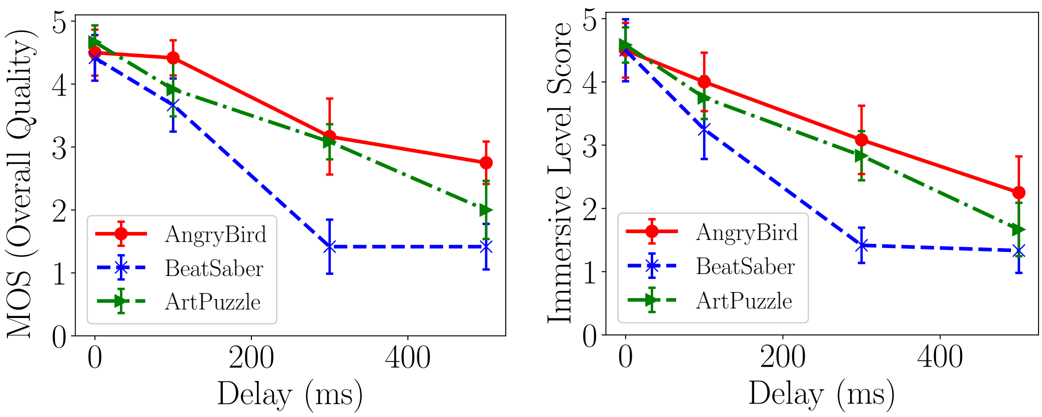
(a) (b)

(c)

Figure 5.6: Implications 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.

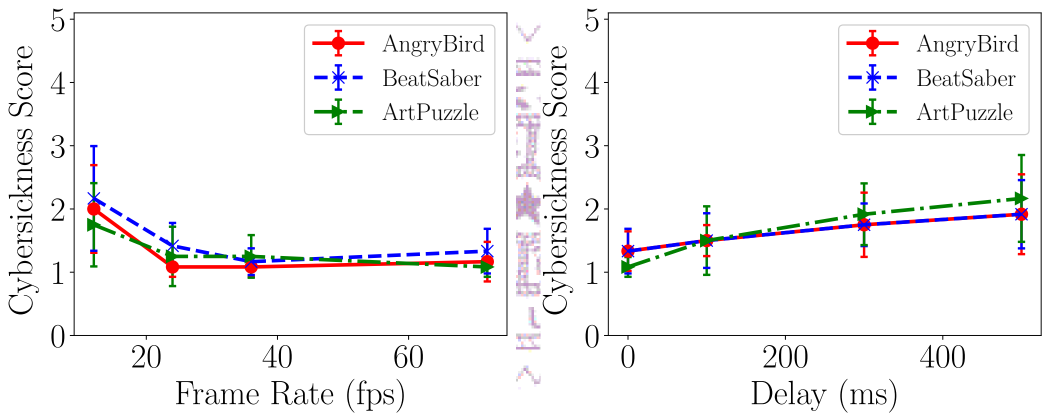
injected delay are both below 0.001, with MOS differences of 3.02 and 3.17, respectively. From the observations above, it is statistically significant that diverse game genres incur different requirements on the QoS.

Cybersickness highly depends on subjects. Fig. 5.8 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 (Fig. 5.8(a)), or the delay approaches 500 ms (Fig. 5.8(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 frame rate of 12 fps, 50% of the subjects gave a rating of 1 (No problem), as illustrated in Fig. 5.8(c). We conclude that cybersickness scores largely depend on subjects. Thus, we leave modeling cybersickness as one of our future works.

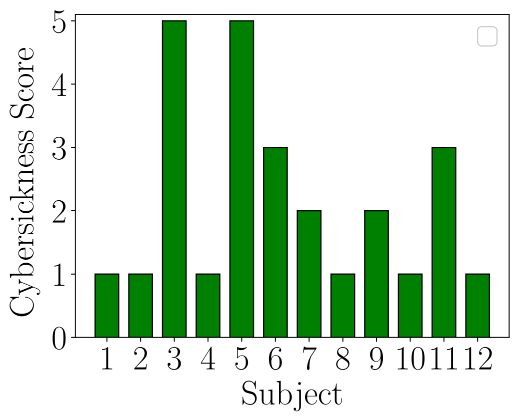


(a) (b)

Figure 5.7: Implications 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.



(a) (b)



(c)

Figure 5.8: 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).

Chapter 6

QoE Modeling

In this chapter, 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 6.1: QoE Model Inputs

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

6.1 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 6.1 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*.

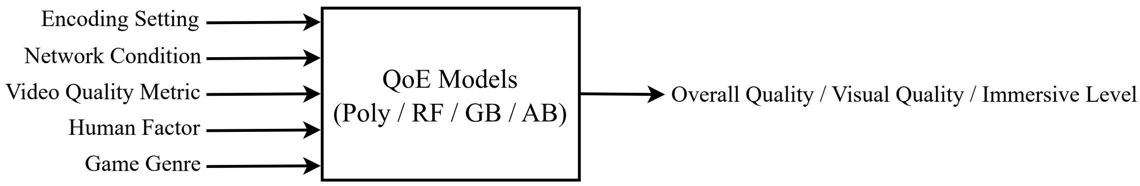


Figure 6.1: Block diagram of the QoE models.

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 and decision tree-based regressors. Polynomial regressor (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 [9]. 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). Fig. 6.1 highlights the inputs and outputs of these regressor models.

We use Scikit-Learn [50] to implement these regression models in Python.

Table 6.2: 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

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 [13] split the dataset into training, validation, and testing sets, while (ii) others [3, 12, 79] split the dataset into training and testing sets only. We opt for the latter approach as we have fewer subjects than Fan et al. [13]. 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 6.2. 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* (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*.

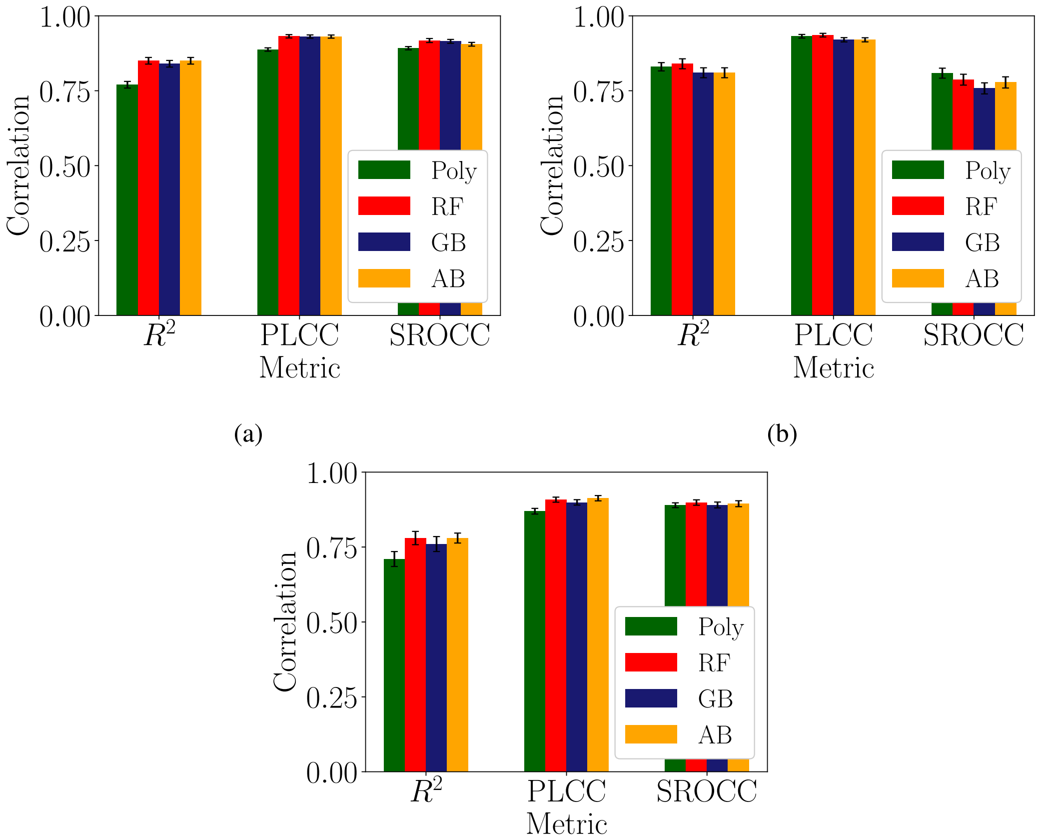
Table 6.3: QoE Modeling Results on Overall Quality: AngryBird/BeatSaber/ArtPuzzle/Adjusted General

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

6.2 Resulting Models

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

- **Adjusted general models deliver good enough performance.** Table 6.3 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 Fig. 6.2. We find that the RF model performs the best. For example, in Fig. 6.2(a), RF achieves up to 0.85 in R^2, 0.93 in PLCC,



(a) (b)

(c)

Figure 6.2: Performance of general models on: (a) MOS of overall quality, (b) MOS of visual quality, and (c) immersive level score.

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. Fig. 6.3 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 thesis.*

- **Immersive level is relatively hard to model.** Compared to overall and visual quality, the performance of immersive level is a bit lower, as illustrated in Fig. 6.2. There may be two possible reasons. First, 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 experiments duration. According to ITU-T recommendation P.809 [28], 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 to ratings.

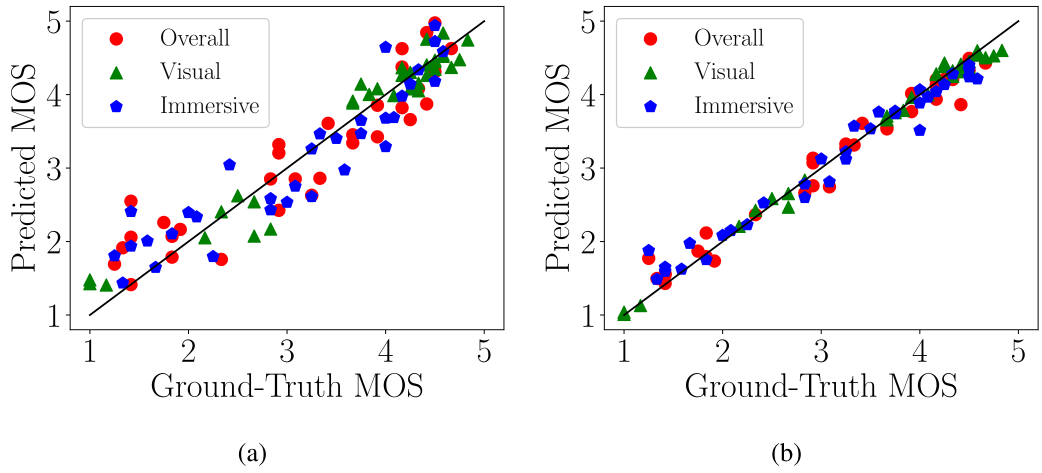


Figure 6.3: Predicted vs. ground-truth MOS: (a) Poly and (b) RF.

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 run-time. 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 thesis.*

Chapter 7

QoE-driven Encoding Settings Adaptation

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

7.1 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 δ = 3 seconds. 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. [43] if not otherwise specified. With the above symbols, we formulate our optimization problem as:

$$
\begin{aligned}
& e^* = \underset{e=(b,f,r)}{argmax}\, \tilde{Q}_O(b, f, r, \dots) \\
& \text{s.t. } (1+\alpha)b \leq B.
\end{aligned}
\tag{7.1}
$$

We note that the dots in $\tilde{Q}_O(\cdot)$ represent seven non-encoding-setting inputs of our QoE models (see Table 6.1). 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.

7.2 QoE-Driven Adaptation (QDA) Algorithm

Solving the optimization problem in Eq. (7.1) 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 $b \in \{2, 3, 4, 5, 6, 7, \ldots, 31\}$, $f \in \{48, 60\}$, 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* [17], 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 $\hat{Q}_O(b, f, r, \ldots)$ for e^* using $\tilde{Q}_O(b, f, r, \ldots)$. Because the lookup table is built offline, doing so incurs no runtime complexity with a memory footprint $\leq$ 700 KB.

We propose a QoE-driven adaptation (QDA) algorithm based on the lookup table $\hat{Q}_O(b, f, r, \ldots)$. 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.

Chapter 8

Performance Evaluations

We evaluate our cloud VR gaming system, especially the QDA algorithm with an additional user study in this chapter. This user study is based on the QoE models constructed with the results obtained from the previous user study in Ch. 5.

8.1 Technical Setup

To drive our experiments, we adopt a real 5G network dataset [53], 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[1] 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 $\sim$ 3 Mbps[2]. 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.

In particular, we conduct a user study to compare our QDA algorithm against the following two baseline algorithms:

[1] We opt for the file-downloading traces for enough traffic loads.

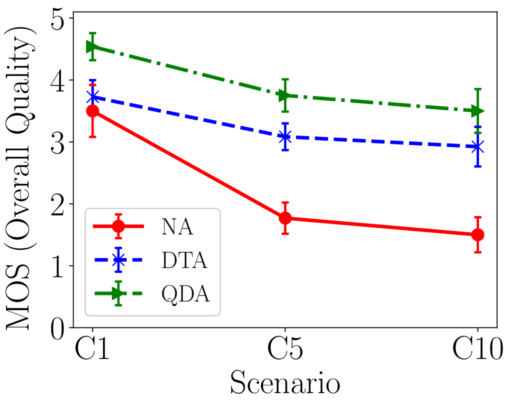
[2] This is enforced by a watchdog mechanism.

- **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, 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_Δ = 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.

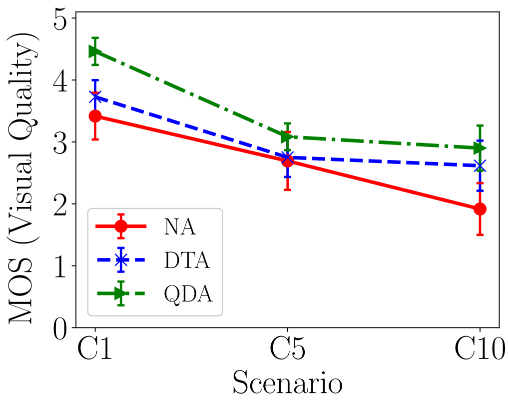
8.2 Test Method

We designed a new user study to evaluate the performance between QDA algorithms and two baseline algorithms. We utilized the same set of game genres mentioned in Ch. 5.1. The user study design is based on that in our first user study described in Ch. 5.3 (see Fig. 5.3), but with a few changes on questionnaires, as summarized in Table 5.3. 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 [57]. 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 ($\sim 60\%$) 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.

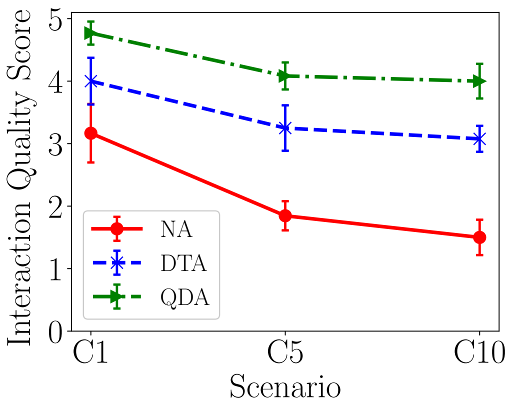
We enlisted 20 subjects (17 males) aged between 20–26 years old. All of them passed the Snellen and Ishihara tests. Among these subjects, 6, 3, and 11 were categorized as novice, intermediate, and advanced gamers. In addition, eight of them had prior VR experience. In total, with 20 subjects and 16 sessions 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.



(a)



(b)



(c)

Figure 8.1: 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.

8.3 Results

MOS scores on overall, visual, and interaction quality. Fig. 8.1 compares the overall, visual, and interaction quality achieved by various adaptation algorithms under different scenarios. Sample results from AngryBird are shown; results from other game genres (BeatSaber and ArtPuzzle) are similar and omitted. Figs. 8.1(a) and 8.1(c) depict that

Table 8.1: QoE Scores from NA/DTA/QDA Algorithms; Scenario C10

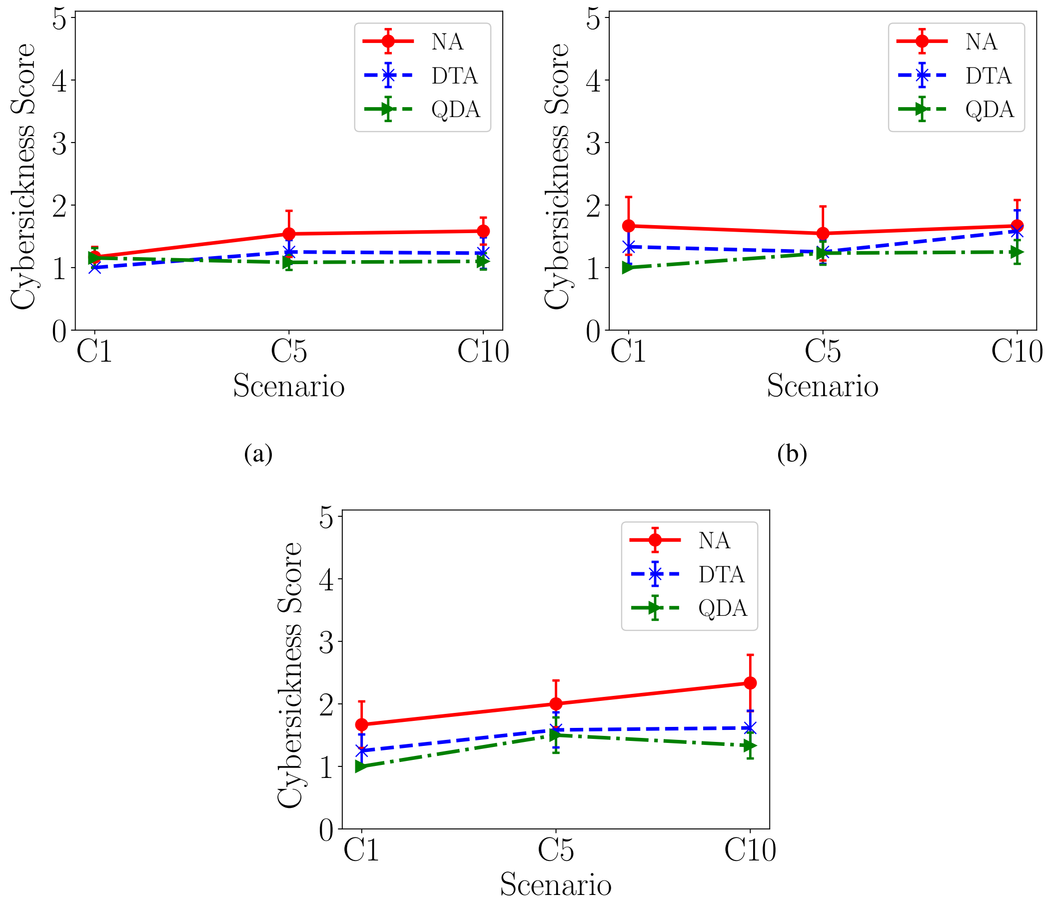
QoE	AngryBird	BeatSaber	ArtPuzzle
Overall Quality	1.50/2.92/3.50	1.17/1.92/3.25	1.42/2.23/2.92
Visual Quality	1.92/2.62/2.90	1.75/1.92/2.92	1.58/1.77/2.33
Interaction Quality	1.50/3.08/4.00	1.08/2.08/3.25	1.41/2.15/3.67

QDA 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; and (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, Fig. 8.1(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 8.1 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 across all three game genres amount to averagely 1.86 ($\sigma = 0.38$) in overall quality, 0.97 ($\sigma = 0.45$) in visual quality, and 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. *Fig. 8.1 and Table 8.1 confirm that our proposed QDA algorithm significantly improves the QoE scores on overall, visual, and interaction quality compared to the baseline algorithms.*

Cybersickness scores. Fig. 8.2 presents the cybersickness scores achieved by different adaptation algorithms in different game genres. This figure shows that among the three game genres, QDA 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. This outcome 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 on our user study are consistent with our previous study on simulator sickness [58], where lower visual quality resulted in higher simulator sickness scores. *Fig. 8.2 confirms that our proposed QDA algorithm leads to relatively lower cybersickness scores, compared to the baseline algorithms.*

Objective quality. Fig. 8.3 reports sample VMAF results achieved by different adaptation algorithms with different game genres and under diverse scenarios. This figure

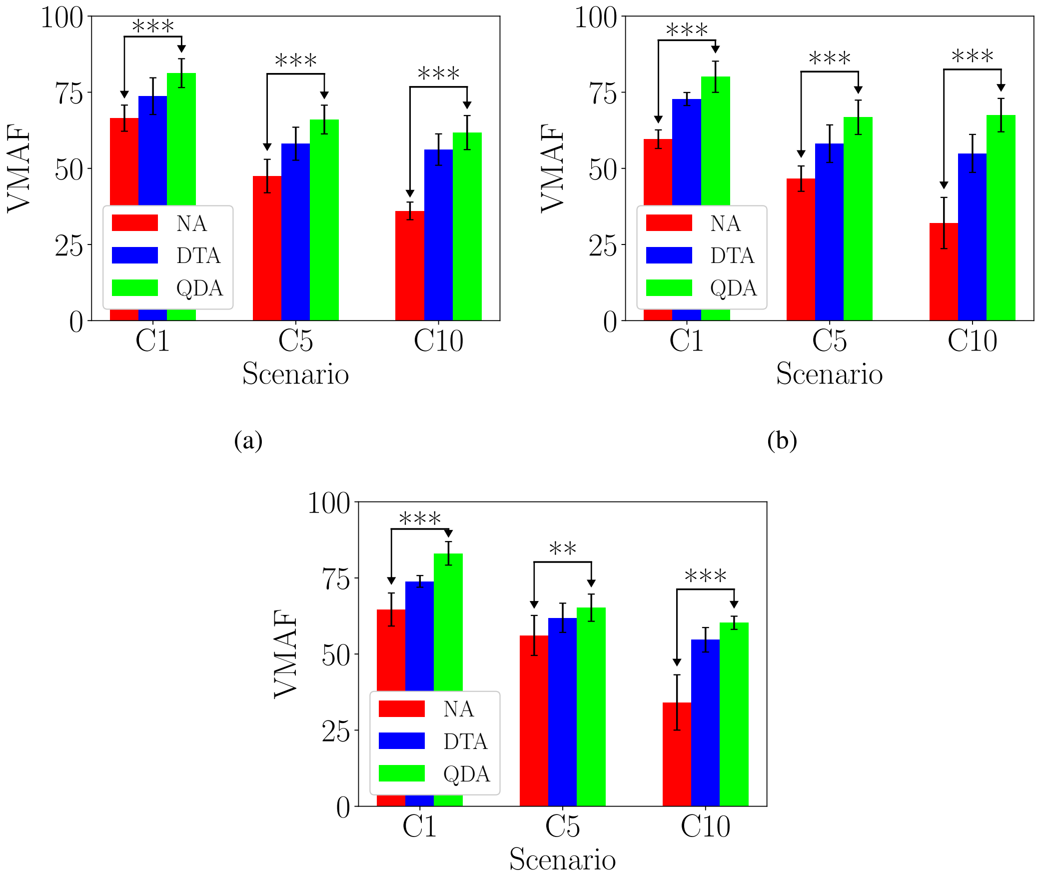


(a) (b) (c)

Figure 8.2: Comparison of cybersickness score across different game genres: (a) Angry-Bird, (b) BeatSaber, and (c) ArtPuzzle.

illustrates that our QDA 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 8.2 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 up to 9.62 dB in PSNR, 0.29 in SSIM, and 35.43 in VMAF are observed. *Fig. 8.3 and Table 8.2 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%



(a) (b) (c)

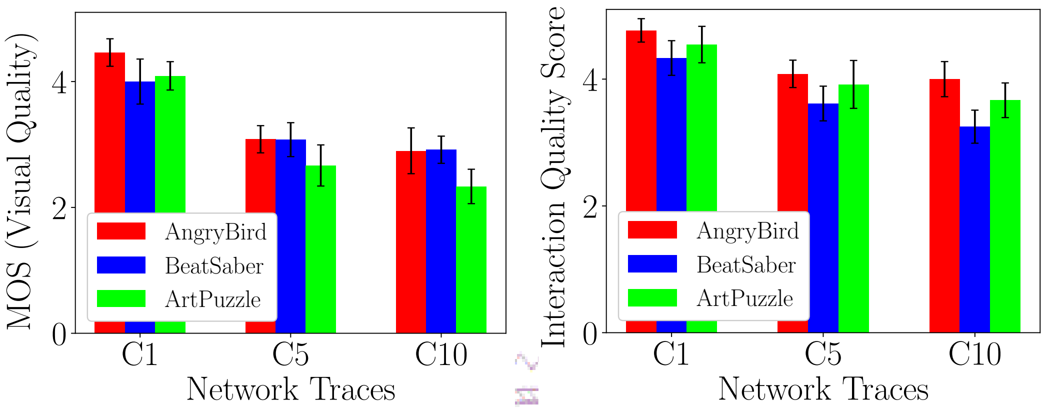
Figure 8.3: Comparison of VMAF across diverse adaptation algorithms with different game genres: (a) AngryBird, (b) BeatSaber, and (c) ArtPuzzle. Significance values are defined as: *: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$.

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.

Implications of game genres. Fig. 8.4 illustrates the impact of different game genres on visual and interaction quality with our QDA algorithms under diverse scenarios, respectively; results from other algorithms (NA and DTA) are similar and omitted. Fig. 8.4(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 algorithms. Similarly, Fig. 8.4(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.

Table 8.2: 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



(a) (b)

Figure 8.4: Implications of different game genres with QDA algorithms on: (a) MOS of visual quality and (b) interaction quality score.

Fig. 8.4 confirms that the results related to visual and interaction quality are consistent with our expectations outlined in Ch. 5.1. Specifically, ArtPuzzle exhibits sensitivity to quality, BeatSaber to time, and AngryBird demonstrates less sensitivity to both factors.

Chapter 9

Conclusion

In this chapter, we conclude the thesis and outline future directions for the current work.

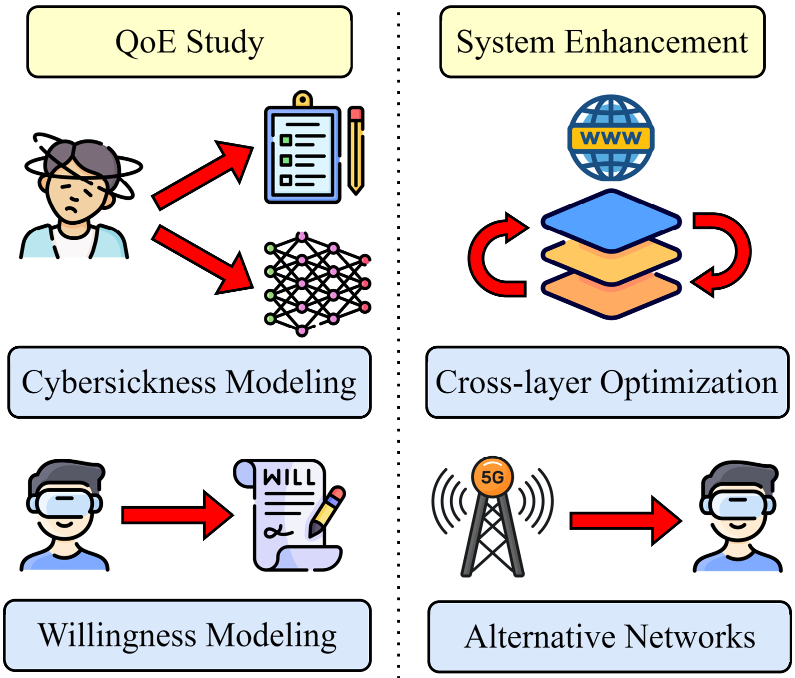


Figure 9.1: Illustration of future work.

9.1 Key Take-Away Messages

In this thesis, 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 gaming QoE, e.g., it improves the MOS of overall quality by up to 1.86 ($\sigma = 0.38$) and reduces the cybersickness scores by up to 0.63 ($\sigma = 0.42$) averagely across different game genres.

9.2 Future Work

Fig. 9.1 gives an illustration of future work. This thesis can be extended in multiple directions, including but not limited to:

9.2.1 Building QoE Models for Cybersickness Scores

During the experience of cloud VR gaming, gamers often experience discomfort and dizziness, which is known as cybersickness. The origins of cybersickness are diverse, stemming from factors like inappropriate QoS, characterized by excessive delay and low image quality. Additionally, game content, particularly that involving high head movements, and the duration of gameplay itself also contribute to the discomfort experienced by gamers. Cybersickness significantly influences the overall gaming experience, making it a pivotal aspect to address in the realm of cloud VR gaming. Fig. 9.2 shows the block diagram of the cybersickness model. Modeling cybersickness requires considering additional factors which were not addressed in this thesis, including subject differences, duration of each gameplay, and accumulated fatigue levels. The problem statement here is to "identify the important factors for modeling cybersickness and, based on the characteristics of these factors, find appropriate methods for modeling." Addressing this issue is challenging as the complexity of the model increases with the inclusion of more factors, requiring further exploration. Additionally, given that the model may encompass high dimensions with diverse input domains, relying solely on regression models may prove insufficient for handling this problem. Consequently, more complicated modeling methods, such as deep learning, could be considered. Moreover, to conduct user studies involving gamers with diverse characteristics, crowdsourcing user studies may be a more fitting approach in this scenario compared to the controlled lab environment. The resulting models can assist cloud VR gaming service providers in adjusting QoS and system settings to meet the diverse needs of gamers to play comfortably.

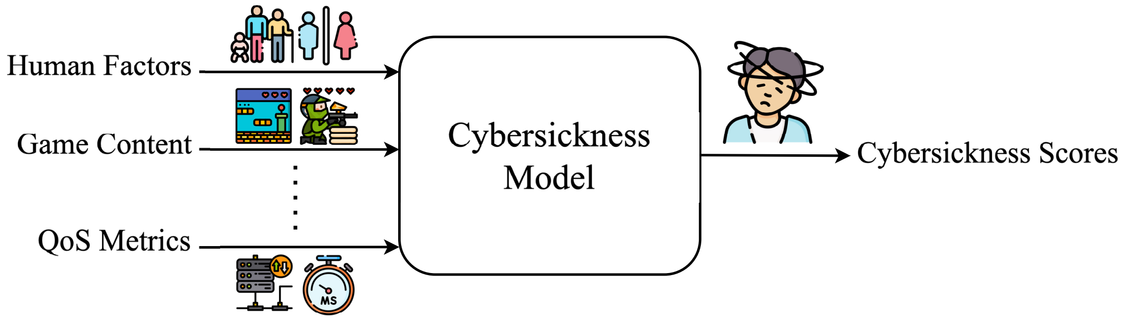


Figure 9.2: Block diagram of the cybersickness model.

9.2.2 Predicting Gamer Willingness to Continue Playing

When users are experiencing multimedia content, they often choose to discontinue the experience prematurely. Aside from personal preferences, a significant reason for this is the instability of the current network conditions, leading to QoS not meeting the user QoE requirements. Therefore, if cloud service providers can continuously monitor users' willingness to continue and adjust service quality accordingly, it not only encourages users to continue their experience but also maximizes the efficient utilization of limited resources. This approach benefits both users and cloud service providers. In the literature, Lebreton and Yamagishi [37, 38] 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. Fig. 9.3 gives the block diagram of the willingness model. The problem statement here is to "examine the impact of various system factors on gamer willingness to continue playing and build models to predict the willingness for identifying whether adaptations are needed." In addressing this problem, it is crucial to investigate how various factors influence gamers' willingness to continue and determine the approach, timing, and type of adaptation to implement. For example, cloud service providers should not wait until users reach the critical point of terminating their experience to initiate adaptation. Instead, they should adjust the degree of adaptation based on the proximity to this critical point. This can be achieved by establishing a function to control the extent of adaptation, ensuring the preservation of the user's positive experience. The resulting models are highly valuable for cloud VR gaming service providers to maintain profitability.

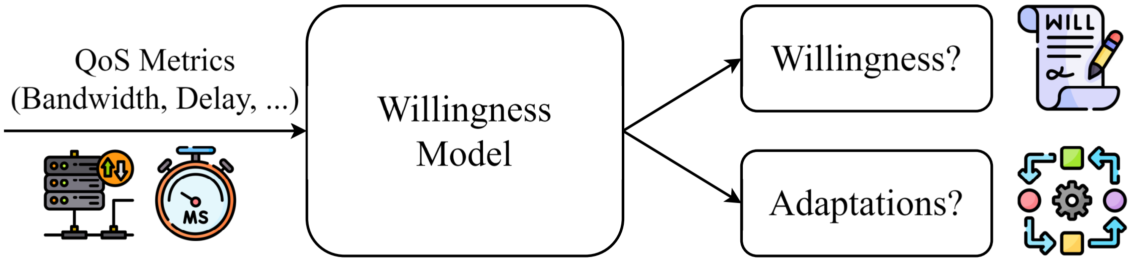


Figure 9.3: Block diagram of the willingness model.

9.2.3 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. RNIS is a specialized service designed to provide comprehensive information about radio networks, facilitating efficient communication and resource utilization. With the low-layer information provided by this service, combined with the high-layer details within the system, the entire system can engage in more effective optimization to enhance overall performance. While RNIS has been employed for cross-layer optimization in flow control [11] and 360° VR video streaming [46], it has not been used in cloud VR gaming. The problem statement here is to "integrate RNIS into the cloud VR gaming system, and identify and implement appropriate optimization tasks within the system using the information provided by RNIS." To achieve this goal, it is necessary to establish an RNIS module within the system. The first approach involves using existing radio network traces to simulate radio statistics under real network conditions. Additionally, when specific network requirements arise, a simulator can be employed to generate virtual network traces for the RNIS module. The second approach is to directly obtain radio statistics from Internet Service Providers (ISPs), providing the most realistic information. Once the RNIS module is established, cross-layer optimization can be performed based on system requirements. Cross-layer optimization can be applied to various applications. For instance, when there are significant differences in the importance or timeliness of transmitted data within the system, packet prioritization can be employed. Additionally, if the system frequently encounters congestion issues, congestion control can be implemented. Lastly, in cases where the system has limited resources and efficient resource utilization is crucial, resource allocation strategies can be employed. Fig. 9.4 shows the overview of cross-layer optimization in cloud VR gaming. These applications can assist the cloud VR gaming system in achieving better performance and enhancing the gamer's overall QoE.

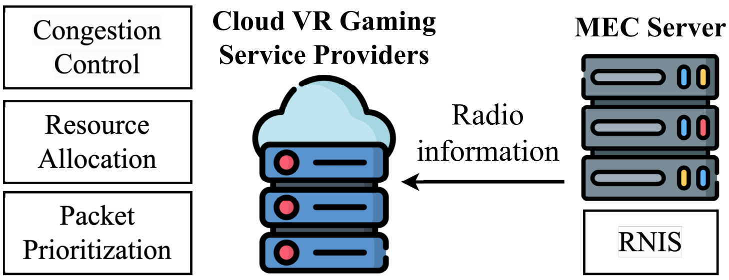


Figure 9.4: Overview of cross-layer optimization in cloud VR gaming.

9.2.4 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. The User needs to install Customer Premises Equipment (CPE) in a location with a strong signal, such as near a window or outdoors. This CPE serves as a receiver for wireless signals and can establish a wireless connection with the nearby 5G base station to enable FWA. Fig. 9.5 gives the overview of FWA. FWA supports high bandwidth and low delay and thus enables new VR/AR applications in rural areas [4, 36]. For example, VR applications place a strong emphasis on immersion, requiring delicate visual scenes and minimal delay for an optimal user QoE. A high level of immersion contributes to an enhanced user experience. However, achieving this often demands higher bandwidth, and FWA can meet this demand. The high bandwidth and low delay provided by FWA support high-performance VR applications, delivering a superior sense of immersion to users. In the case of AR, FWA also plays a pivotal role. AR applications are often operated in large-scale environments, and this aligns with the strengths of FWA. The extensive network coverage and cable-free nature of FWA allow users to have a more immersive experience without constraints. This, in turn, leads to a higher QoE for users. Therefore, cloud VR gaming can be a potential application for FWA. The problem statement here is to "replace WiFi networks with FWA networks and take an initial step to study the QoE for gamers under this novel access network." Experimenting with cloud VR gaming over FWA could reveal new challenges and opportunities under its unique network workload in emerging 5G-based access networks.

9.2.5 Comparing Local and Cloud VR Gaming

Although cloud VR gaming offers an alternative way for gamers to access VR games, not all types of games are suitable for experiencing in the cloud VR gaming environment. For instance, lightweight games with simple game scenes may still be better experienced using local VR gaming settings. This is because offloading the rendering task to the server

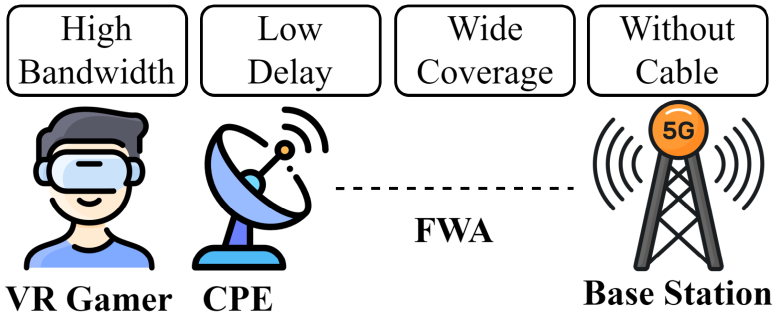


Figure 9.5: Overview of FWA.

may not significantly enhance the gamer QoE, but could introduce additional delays and costs, which are not cost-effective from a business perspective. Therefore, it is valuable to study and determine the best settings for different kinds of games. The problem statement here is to "identify the characteristics of different game genres and optimize the gaming experience with the ideal settings." One potential solution is to conduct user studies across a wide range of game genres to take the first step in examining the gamer QoE for different game genres under both local and cloud VR gaming settings. By studying the differences between local and cloud VR gaming, cloud VR gaming service providers can improve the cost-effectiveness of their services while also enhancing the gamer QoE.

9.2.6 Utilizing Various QoE Modeling Methods

In this thesis, we employed well-known machine learning methods to establish our QoE models. However, there are numerous methods available for constructing QoE models, such as using curve fitting techniques or neural networks. Each method has its own pros and cons; some may entail higher overhead but offer greater precision, while others might have lower overhead but potentially lower accuracy. In the context of cloud VR gaming, overhead is a primary consideration, especially given its real-time system requirements. The problem statement here is "how to select the modeling method that provide users with the best gaming experience." To address this issue, we need to experiment with various methods for constructing models and conduct thorough subjective and objective analyses for each approach. By doing so, QoE-driven adaptation based on the QoE model can let cloud VR gaming service providers offer gamers the optimal gamer QoE.

Bibliography

[1] M. Abdallah, C. Griwodz, K.-T. Chen, G. Simon, P.-C. Wang, and C.-H. Hsu. Delay-sensitive video computing in the cloud: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 14(3s):1–29, 2018.

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

[3] A. Ahmad, A. B. Mansoor, A. A. Barakabitze, A. Hines, L. Atzori, and R. Walshe. Supervised-learning-based qoe prediction of video streaming in future networks: a tutorial with comparative study. *IEEE Communications Magazine*, 59(11):88–94, 2021.

[4] K. Aldubaikhy, W. Wu, N. Zhang, N. Cheng, and X. Shen. mmwave ieee 802.11 ay for 5g fixed wireless access. *IEEE Wireless Communications*, 27(2):88–95, 2020.

[5] ALVR. The github of alvr, 2019. https://reurl.cc/rrQ9y4.

[6] ALXR. The github of alxr, 2021. https://reurl.cc/xLOa5L.

[7] M. S. Anwar, J. Wang, W. Khan, A. Ullah, S. Ahmad, and Z. Fei. Subjective qoe of 360-degree virtual reality videos and machine learning predictions. *IEEE Access*, 8:148084–148099, 2020.

[8] W. Cai, R. Shea, C.-Y. Huang, K.-T. Chen, J. Liu, V. C. M. Leung, and C.-H. Hsu. A survey on cloud gaming: Future of computer games. *IEEE Access*, 4:7605–7620, 2016.

[9] M. Á. Carreira-Perpiñán and A. Zharmagambetov. Ensembles of bagged tao trees consistently improve over random forests, adaboost and gradient boosting. *FODS*, 20:19–20, 2020.

[10] G. Cofano, L. D. Cicco, T. Zinner, A. Nguyen-Ngoc, P. Tran-Gia, and S. Mascolo. Design and performance evaluation of network-assisted control strategies for http

adaptive streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 13(3s):1–24, 2017.

[11] M. Diarra, W. Dabbous, A. Ismail, B. Tetu, and T. Turletti. Rapid: A ran-aware performance enhancing proxy for high throughput low delay flows in mec-enabled cellular networks. *Computer Networks*, 218:109357, 2022.

[12] N. Eswara, S. Ashique, A. Panchbhai, S. Chakraborty, H. P. Sethuram, K. Kuchi, A. Kumar, and S. S. Channappayya. Streaming video qoe modeling and prediction: a long short-term memory approach. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(3):661–673, 2019.

[13] C.-L. Fan, T.-H. Hung, and C.-H. Hsu. Modeling the user experience of watching 360 videos with head-mounted displays. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 18(1):1–23, 2022.

[14] C.-L. Fan, W.-C. Lo, Y.-T. Pai, and C.-H. Hsu. A survey on 360 video streaming: Acquisition, transmission, and display. *ACM Computing Surveys*, 52(4):1–36, 2019.

[15] J.-W. Fang, K.-Y. Lee, T. Kämäräinen, M. Siekkinen, and C.-H. Hsu. Will dynamic foveation boost cloud vr gaming experience. In *Proc. of Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, pages 29–35, Vancouver, Canada, 2023.

[16] FreeBSD. A live network emulation tool, 2002. https://reurl.cc/LX8j34.

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

[18] J. Gutierrez, P. Perez, M. Orduna, A. Singla, C. Cortes, P. Mazumdar, I. Viola, K. Brunnström, F. Battisti, N. Cieplińska, et al. Subjective evaluation of visual quality and simulator sickness of dhort 360° videos: Itu-t rec. p. 919. *IEEE transactions on multimedia*, 24:3087–3100, 2021.

[19] H.-J. Hong, C.-F. Hsu, T.-H. Tsai, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu. Enabling adaptive cloud gaming in an open-source cloud gaming platform. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(12):2078–2091, 2015.

[20] C.-Y. Huang, K.-T. Chen, D.-Y. Chen, H.-J. Hsu, and C.-H. Hsu. Gaminganywhere: The first open source cloud gaming system. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 10(1s):1–25, 2014.

[21] C.-Y. Huang, C.-H. Hsu, D.-Y. Chen, and K.-T. Chen. Quantifying user satisfaction in mobile cloud games. In *Proc. of Workshop on Mobile Video Delivery (MoViD)*, pages 1–6, Singapore, Singapore, 2014.

[22] A. R. Hummaida, N. W. Paton, and R. Sakellariou. Adaptation in cloud resource configuration: a survey. *Journal of Cloud Computing*, 5:1–16, 2016.

[23] N. Inc. Vmaf - video multi-method assessment fusion, 2019. https://reurl.cc/eL6LeR.

[24] M. Intelligence. Virtual reality gaming market size & share analysis - growth trends & forecasts (2023 - 2028), 2023. https://reurl.cc/edbrxx.

[25] S. G. M. Intelligence. Apple pushes into ar/vr hardware, and meta plants its feet, 2023. https://reurl.cc/kaz5Rx.

[26] ITU-R. Recommendation itu-r bt.500: Methodology for the subjective assessment of quality for television pictures. 2012.

[27] ITU-T. Recommendation itu-t p.910: Subjective video quality assessment methods for multimedia applications. 2008.

[28] ITU-T. Recommendation itu-t p.809: Subjective evaluation methods for gaming quality. 2018.

[29] ITU-T. Recommendation itu-t g.1072: Opinion model predicting gaming qoe for cloud gaming services. 2020.

[30] R. Jain. Quality of experience. *IEEE multimedia*, 11:96–95, 2004.

[31] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld. An evaluation of qoe in cloud gaming based on subjective tests. In *Proc. of IEEE International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, pages 330–335, Seoul, Korea, 2011.

[32] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *International Journal of Aviation Psychology*, 3(3):203–220, 1993.

[33] A. Khan, I.-H. Mkwawa, L. Sun, and E. Ifeachor. Qoe-driven sender bitrate adaptation scheme for video applications over ip multimedia subsystem. In *Proc. of IEEE International Conference on Communications (ICC)*, pages 1–6, Kyoto, Japan, 2011.

[34] B. Krogfoss, J. Duran, P. Perez, and J. Bouwen. Quantifying the value of 5g and edge cloud on qoe for ar/vr. In *Proc. of IEEE International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–4, Athlone, Ireland, 2020.

[35] J. Kua, G. Armitage, and P. Branch. A survey of rate adaptation techniques for dynamic adaptive streaming over http. *IEEE Communications Surveys & Tutorials*, 19(3):1842–1866, 2017.

[36] A. Lappalainen and C. Rosenberg. Can 5g fixed broadband bridge the rural digital divide? *IEEE Communications Standards Magazine*, 6(2):79–84, 2022.

[37] P. Lebreton and K. Yamagishi. Predicting user quitting ratio in adaptive bitrate video streaming. *IEEE Transactions on Multimedia*, 23:4526–4540, 2020.

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

[39] K.-Y. Lee. The github of user study testbed, 2022. https://reurl.cc/dLXnYg.

[40] K.-Y. Lee. The github of qoe data and model, 2023. https://reurl.cc/v0e0Ya.

[41] K.-Y. Lee, J.-W. Fang, Y.-C. Sun, and C.-H. Hsu. Modeling gamer quality-of-experience using a real cloud vr gaming testbed. In *Proc. of International Workshop on Immersive Mixed and Virtual Environment Systems (MMVE)*, pages 12–17, Vancouver, Canada, 2023.

[42] J. Li, R. Feng, Z. Liu, W. Sun, and Q. Li. Modeling qoe of virtual reality video transmission over wireless networks. In *Proc. of IEEE Global Communications Conference (GLOBECOM)*, pages 1–7, Abu Dhabi, United Arab Emirates, 2018.

[43] Y.-C. Li, C.-H. Hsu, Y.-C. Lin, and C.-H. Hsu. Performance measurements on a cloud vr gaming platform. In *Proc. of Quality of Experience (QoE) in Visual Multimedia Applications (QoEVMA)*, pages 37–45, New York, USA, 2020.

[44] Z.-N. Li, M. S. Drew, and J. Liu. Fundamentals of multimedia: second edition. chapter 8,12. 2014.

[45] E. Liotou, D. Tsolkas, and N. Passas. A roadmap on qoe metrics and models. In *Proc. of International Conference on Telecommunications (ICT)*, pages 1–5, Thessaloniki, Greece, 2016.

[46] Y. Liu, J. Liu, A. Argyriou, and S. Ci. Mec-assisted panoramic vr video streaming over millimeter wave mobile networks. *IEEE Transactions on Multimedia*, 21(5):1302–1316, 2018.

[47] S. Möller and A. Raake. Quality of experience: advanced concepts, applications and methods. chapter 8,12. 2014.

[48] Netflix. The github of pre-trained vmaf models, 2021. https://reurl.cc/zrvgya.

[49] Nvidia. The official website of nvidia cloudxr, 2020. https://reurl.cc/10djxm.

[50] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[51] P. Pérez, N. Oyaga, J. J. Ruiz, and A. Villegas. Towards systematic analysis of cybersickness in high motion omnidirectional video. In *Proc. of International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–3, Sardinia, Italy, 2018.

[52] S. Petrangeli, J. Famaey, M. Claeys, S. Latré, and F. D. Turck. Qoe-driven rate adaptation heuristic for fair adaptive video streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 12(2):1–24, 2015.

[53] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan. Beyond throughput, the next generation: a 5g dataset with channel and context metrics. In *Proc. of ACM Multimedia Systems Conference (MMSys)*, pages 303–308, Istanbul, Turkey, 2020.

[54] A. Sackl, R. Schatz, T. Hossfeld, F. Metzger, D. Lister, and R. Irmer. Qoe management made uneasy: the case of cloud gaming. In *Proc. of IEEE International Conference on Communications Workshops (ICC)*, pages 492–497, Kuala Lumpur, Malaysia, 2016.

[55] Y. Shen, P. Ding, Y. Xue, and Y. Song. A qoe-driven adaptive transmission scheme for streaming media service. In *Proc. of IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–5, Bilbao, Spain, 2022.

[56] S. Shi and C.-H. Hsu. A survey of interactive remote rendering systems. *ACM Computing Surveys (CSUR)*, 47(4):1–29, 2015.

[57] A. Singla. Assessment of visual quality and simulator sickness for omnidirectional videos. *Assessment*, 2024.

[58] A. Singla, S. Fremerey, W. Robitza, and A. Raake. Measuring and comparing qoe and simulator sickness of omnidirectional videos in different head mounted displays. In *Proc. of International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6, Erfurt, Germany, 2017.

[59] A. Singla, S. Göring, D. Keller, R. R. R. Rao, S. Fremerey, and A. Raake. Assessment of the simulator sickness questionnaire for omnidirectional videos. In *Proc. of Virtual Reality and 3D User Interfaces (VR)*, pages 198–206, Lisbon, Portugal, 2021.

[60] I. Slivar. *Quality of experience driven video encoding adaptation strategies for cloud gaming under network constraints*. PhD thesis, University of Zagreb. Faculty of Electrical Engineering and Computing, 2021.

[61] I. Slivar, L. Skorin-Kapov, and M. Suznjevic. Cloud gaming qoe models for deriving video encoding adaptation strategies. In *Proc. of ACM International Conference on Multimedia Systems (MMSys)*, pages 1–12, Klagenfurt, Austria, 2016.

[62] I. Slivar, M. Suznjevic, and L. Skorin-Kapov. The impact of video encoding parameters and game type on qoe for cloud gaming: a case study using the steam platform. In *Proc. of IEEE International Workshop on Quality of Multimedia Experience (QoMEX)*, pages 1–6, Pilos, Greece, 2015.

[63] I. Slivar, S. Vlahovic, M. Silic, L. Skorin-Kapov, and M. Suznjevic. The impact of network and social context on quality of experience for competitive multiplayer virtual reality games. In *Proc. of ACM Workshop on Games Systems (GameSys)*, pages 16–21, Athlone, Ireland, 2022.

[64] A. Sobhani, A. Yassine, and S. Shirmohammadi. A video bitrate adaptation and prediction mechanism for http adaptive streaming. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 13(2):1–25, 2017.

[65] C. Spearman. The proof and measurement of association between two things. *American Journal of Psychology*, 15:72–101, 1961.

[66] M. Suznjevic, J. Beyer, L. Skorin-Kapov, S. Moller, and N. Sorsa. Towards understanding the relationship between game type and network traffic for cloud gaming. In *Proc. of International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 1–6, Chengdu, China, 2014.

[67] H. T. Tran, N. P. Ngoc, C. T. Pham, Y. J. Jung, and T. C. Thang. A subjective study on qoe of 360 video for vr communication. In *Proc. of International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6, Luton, UK, 2017.

[68] H. T. Tran, N. P. Ngoc, and T. C. Thang. Towards an overall qoe model for 360-degree video. In *Proc. of IEEE International Conference on Communications and Electronics (ICCE)*, pages 327–331, Phu Quoc Island, Vietnam, 2021.

[69] J. Vanian. Meta's heated rivalry with apple enters new phase as the tech giants go after headsets, 2023. https://reurl.cc/dmMr6V.

[70] S. Vlahovic, M. Suznjevic, and L. Skorin-Kapov. The impact of network latency on gaming qoe for an fps vr game. In *Proc. of International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–3, Berlin, Germany, 2019.

[71] S. Vlahovic, M. Suznjevic, and L. Skorin-Kapov. A survey of challenges and methods for quality of experience assessment of interactive vr applications. *Journal on Multimodal User Interfaces*, 16(3):257–291, 2022.

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

[73] Z. Wang and X. Jiang. A qoe-driven rate adaptation approach for dynamic adaptive streaming over http. In *Proc. of IEEE International Conference on Computing, Networking and Communications (ICNC)*, pages 224–229, Hawaii, USA, 2019.

[74] M. Yang, S. Wang, R. N. Calheiros, and F. Yang. Survey on qoe assessment approach for network service. *Ieee Access*, 6:48374–48390, 2018.

[75] J. Yao, Z. Pan, and H. Zhang. A distributed render farm system for animation production. In *Proc. of International Conference on Entertainment Computing (ICEC)*, pages 264–269, Paris, France, 2009.

[76] S.-H. Yao, C.-L. Fan, and C.-H. Hsu. Towards quality-of-experience models for watching 360 videos in head-mounted virtual reality. In *Proc. of IEEE International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–3, Berlin, Germany, 2019.

[77] S. Zadtootaghaj, S. Schmidt, and S. Möller. Modeling gaming qoe: towards the impact of frame rate and bit rate on cloud gaming. In *Proc. of IEEE International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6, Cagliari, Italy, 2018.

[78] W. Zhao, D. Olshefski, and H. G. Schulzrinne. Internet quality of service: An overview. 2000.

[79] H. Zhu, T. Li, C. Wang, W. Jin, S. Murali, M. Xiao, D. Ye, and M. Li. Eye-qoe: a novel qoe assessment model for 360-degree videos using ocular behaviors. *ACM Transactions on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(1):1–26, 2022.