The Dissertation Committee for Christos George Bampis
certifies that this is the approved version of the following dissertation:

Perceptual Video Quality and Quality of Experience for Adaptive Video Streaming

Committee:

Alan C. Bovik, Supervisor

Joydeep Ghosh

Haris Vikalo

Wilson Geisler

Zhi Li

Perceptual Video Quality and Quality of Experience for Adaptive Video Streaming

by

Christos George Bampis

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2018

Dedicated to my family and friends.

Acknowledgments

I would like to thank those that have helped me in completing this dissertation. I am indebted to my advisor Alan C. Bovik for inspiring, supporting and guiding me throughout those years. For the past four years, I have always felt that I can count on him for advice and research ideas. He has truly become a role model for me as a researcher and an academic. I am very grateful to Zhi Li, who has been actively involved in this dissertation and has played a pivotal role in my academic progress for the past three years. Further, I would like to acknowledge the contributions of Ioannis Katsavounidis, Anush K. Moorthy, Anne Aaron, Te-Yuan Huang, Chaitu Ekanadham and the entire Video Algorithms and Streaming Client teams at Netflix, whose help and financial support has greatly contributed to this work. In addition, a thank you to my PhD committee: Joydeep Ghosh, Haris Vikalo, Wilson Geisler and Zhi Li for providing valuable comments and feedback to this work.

I am deeply grateful to my undergraduate advisor and mentor at NTUA Petros Maragos. He was the first one to introduce me to the areas of Image Processing and Computer Vision and excited my research interests since then. Appreciation is due to Mia Markey for advising me during my first PhD year and to Marios Pattichis for his career advice during this last year. I would like to express my gratitude to Gowri Somanath and Oscar Nestarez for supporting

my first summer internship at Intel, which gave me valuable work experience. Also, I want to thank all the people from UT Austin who have volunteered to participate in the experimental studies described herein; their help has been very important for completing this work.

I would also like to thank my friends and labmates at LIVE: Todd, Zeina, Janice, Deepti, Leo, Lark and Praful whose advice and encouragement was plentiful. I was also very happy during my PhD journey to meet the newer members of LIVE, Xiangxu, Meixu, Yize, Somdyuti and Sungsoo. I am thankful to all of my friends with whom I have shared all those valuable moments in life: John, Dimitris, Spiros, Michael, Nick, Oddyseus and Alexander and those that I met since I arrived to the US: Petri, Todd, Orestis, Ioakeim, Antonis, Jason, Stefanos and Vassilina.

I would like to thank my whole family and especially my parents George and Penelope and my sister Valia for being there for me always, both in good and bad times. I feel indebted to my grandfather Christos and my grandmother Stavroula for their love during my childhood years. I will always remember them. I want to thank everyone with whom I have crossed paths until now, since they have greatly determined me as person. This last word of acknowledgment I have saved for my greatest passion: poetry. It has always been my escape route.

Perceptual Video Quality and Quality of Experience for Adaptive Video Streaming

Publication No. _____

Christos George Bampis, Ph.D.
The University of Texas at Austin, 2018

Supervisor: Alan C. Bovik

We live in a world where images and videos dominate our everyday lives. Every day, an enormous amount of video data is being shared in social media and consumer applications, while video streaming is becoming a new form of digital entertainment. Large-scale video streaming on demand has become possible thanks to numerous engineering achievements in fields such as video compression, high-speed computation and display technologies. Nevertheless, the skyrocketing needs for bandwidth and network resources consumed by video applications challenges modern video content delivery.

Since the available bandwidth resources are limited, streaming service providers have to mediate between operation costs, bandwidth efficiency and maximizing user quality of experience. However, these goals are inherently conflicting and require knowledge of how user quality of experience is affected by the network-induced changes in video quality. Being able to understand

and predict user quality of experience and perceptually optimize rate allocation, can have significant effects in better network utilization, reduced costs for service providers and improved user satisfaction. The goal of this dissertation is to study and predict user quality of experience in video streaming applications, by exploiting perceptual video quality and human behavioral responses to streaming-related video impairments.

To this end, I present the details of three large-scale video subjective studies which target video streaming under multiple viewing conditions, such as display device, session duration, content characteristics and network/buffer conditions. By analyzing how humans react to changes in visual quality and streaming video impairments, I also design numerous video quality and quality of experience prediction models that can be used to evaluate the overall and the continuous-time perceived video quality. Throughout this dissertation, my goal is to perceptually optimize various stages of the video streaming pipeline, such as video encoding and video quality control as well as client-based rate adaptation. Ultimately, I envision that the outcome of this dissertation can be useful for video streaming applications at global scale.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Perceptual Video Quality Assessment and Quality of Experience in Adaptive Video Streaming

Global mobile data traffic grew 74% and mobile video traffic accounted for 55 percent of total mobile data traffic in 2015 [9]. According to the Cisco Visual Networking Index and global mobile data traffic forecast, mobile data traffic will grow 8-fold from 2015 to 2020, which constitutes a compound annual growth rate of 53%. Adding to the delivery over fixed networks, this large and growing volume of mobile video data, video streaming providers such as Netflix, Youtube and Hulu are processing, storing and delivering vast amounts of video data on a daily basis.

Given the exploding use of mobile video devices and the tremendous network bandwidth demands of streaming users, the biggest challenge in video content delivery is to create better network-aware strategies to improve end-users' quality of experience (QoE). QoE is a measure of the delight or annoyance of a customer's experiences with a service, and being able to accurately predict it could enable providers to offer better video streaming services. In this direction, HTTP Adaptive Streaming (HAS) is perhaps the most common method being used by content providers as a way of dealing with network

fluctuations.

In mobile video streaming applications, available network resources are not always sufficient for high-quality video streaming. For example, Netflix recently expanded its video streaming services to many countries around the world [1], such as India, where the available bandwidth is sometimes low. Due to network fluctuations and bandwidth limitations, client rate adaptation methods may lead to frequent quality switching [125] which corresponds to changes in the encoding resolution and/or bitrate, leading to compression and scaling artifacts. Meanwhile, when the available bandwidth is low and the client buffer is emptied, start-up delays and/or stalling (or rebuffering) events occur. Given that HAS uses TCP as the transfer protocol, HAS applications are resilient to video quality degradations related to packet loss, such as glitches and other transient artifacts [33, 40, 107, 131].

These network-related video impairments in HAS applications adversely affect end-user quality of experience (QoE) ubiquitously; hence studying QoE has become a major priority of streaming video companies, network providers and video QoE researchers. For example, to better account for fluctuating bandwidth conditions, industry standard HTTP-based adaptive streaming protocols have been developed [19, 30, 61, 82, 103, 104, 135] that divide streaming video content into chunks, represented at various quality levels; whereby the quality level (or representation) to be played at any given time is selected based on the estimated network condition and/or buffer capacity. These adaptation algorithms seek to reduce the frequency and number of rebuffer-

ing events, while minimizing occurrences of low video quality and/or frequent quality switches, all of which can significantly and adversely affect viewer QoE [50, 149, 150]. Being able to predict end users' QoE resulting from these adjustments could lead to perceptually-driven network resource allocation strategies that would deliver streaming content of higher quality to clients, while being cost effective for providers. To this end, a number of QoE predictors have been developed, but they do not always capture the interplay between video quality and stalling.

While the motivation for perceptually-driven models is obvious, QoE prediction is still far from being an easy task. The early (front-end) human visual system (HVS) is complex and driven by non-linear processes that are not yet completely understood. Moreover, there are also cognitive factors that influence perceived QoE, adding further layers of complexity, complicating the analysis of human subjective data and the design of QoE prediction models. For example, QoE is affected by recency: more recent QoE experiences often have a higher impact on currently perceived QoE [56]. QoE studies can be divided into two categories: retrospective QoE and continuous-time QoE studies. In studies of retrospective QoE, subjects provide a single score describing their overall QoE on an entire video, after it finishes playing. Studies of continuous-time QoE involve the real-time measurement of each subject's current QoE, which may be triggered by changes in video quality or streaming and by short or long term memory effects.

In this thesis, my goal is to cover various aspects of studying and pre-

dicting user QoE in adaptive video streaming, such as collecting and analyzing human opinion scores on videos afflicted by streaming-related impairments and designing QoE prediction models. My ultimate goal is to inject principles of visual neuroscience and human behavior modeling into the video data resource allocation strategies.

1.2 Contributions

In this dissertation, I propose a number of research efforts towards better understanding subjective and predicting QoE for modern adaptive streaming applications. These contributions can be broadly classified into two main categories:

1.2.1 Analysis of Subjective Quality of Experience

I designed three large video quality and quality of experience databases: LIVE-NFLX Video Quality of Experience database (Chapter 2), VMAF+ video quality database (Chapter 5) and LIVE-NFLX-II Video Quality of Experience database (Chapter 6). These three databases are used to study the effects of streaming-related impairments to subjective video quality and QoE, design better video quality assessment and QoE prediction models and take a first step towards perceptually optimizing various components of adaptive video streaming architectures, such as video encoding and quality control on the server and rate adaptation and video quality measurements on the client device. These databases capture multiple effects of actual streaming condi-

tions, such as multiple viewing devices (mobile, laptop, television), long- or short-term viewing sessions, realistic and/or actual network conditions and buffer state representations, rate adaptation policies. These databases investigate how the content characteristics afflict the video streaming user experience hence they include multiple content types (action, drama, anime and cartoon), lightning conditions (dark and light scenes) and in-source distortions (film grain noise, low resolution video capture).

1.2.2 Quality of Experience Prediction Modeling

By collecting subjective video data, it is possible to devise QoE prediction models. Towards this end, this thesis describes the Video ATLAS (Chapter 3), the G-NARX and G-RNN QoE (Chapter 4) prediction models and the SpEED-QA, ST-VMAF and E-VMAF video quality assessment models (Chapter 5). The former approaches exploit statistical models of videos and images to predict visual quality, while the latter target a more general scenario where video quality measurements (such as ST-VMAF) are combined with rebuffering and memory-related measurements to predict retrospective and continuous-time quality of experience.

Specifically, the proposed VQA models (SpEED-QA, ST-VMAF and E-VMAF), rely on the statistics of video frames and frame differences to extract rich spatial and temporal information sensitive to the presence of spatial and temporal quality degradations. By extracting these perceptually-motivated video quality features, I train multiple regression models to predict perceptual

video quality.

To predict retrospective (overall) quality of experience, I propose Video ATLAS: a feature-based approach to making retrospective HAS-QoE predictions when the videos are afflicted by both bitrate changes and stalling. A unique feature of this approach is that it combines perceptually-driven VQA modeling together with stalling and memory information into a single QoE response. The proposed model is scalable, i.e. it can also incorporate additional inputs, if they can improve the performance.

I also propose a variety of recurrent dynamic neural networks that conduct continuous-time subjective QoE prediction (G-NARX and G-RNN). By formulating the problem as one of time-series forecasting, I train a variety of recurrent neural networks and non-linear autoregressive models to predict QoE using several recently developed subjective QoE databases (inclduing the ones introduced in this thesis). These models combine multiple, diverse neural network inputs such as predicted video quality scores, rebuffering measurements, and data related to memory and its effects on human behavioral responses. Instead of finding a single time-series prediction model, I propose and evaluate ways of aggregating different models into a forecasting ensemble that delivers improved results with reduced forecasting variance.

Chapter 2

Study of Temporal Effects on Subjective Video Quality of Experience

2.1 Introduction

HTTP adaptive streaming is being increasingly deployed by network content providers such as Netflix and YouTube. By dividing video content into data chunks encoded at different bitrates, a client is able to request the appropriate bitrate for the segment to be played next based on the estimated network conditions. However, this can introduce a number of impairments, including compression artifacts and rebuffering events which can severely impact an end-user's quality of experience (QoE). Given that the end goal of every content provider is to maximize the end-user's QoE while mediating parameters to accommodate network changes and changing bandwidth, subjective modelling of streaming video QoE becomes an important objective.[1]

Subjective testing is an established way of measuring QoE under different scenarios and settings. Many successful studies have been developed using

[1] This chapter appears in the paper: C. G. Bampis, Z. Li, A. K. Moorthy, I. Katsavounidis, A. Aaron and A. C. Bovik, "Study of Temporal Effects on Subjective Video Quality of Experience", IEEE Transactions on Image Processing, vol. 26, no. 11, pp. 5217-5231, 2017. Christos George Bampis has designed the subjective study, carried out the subjective experiment and studied the subjective data collected from the experiment.

short video sequences of 10-15 seconds (or even less) as in [45, 83, 100, 131]. However, these studies do not reflect typical video streaming situations, where subjects view videos that could be minutes long. Hence, it is not possible to analyze long-term memory effects as they relate to critical factors affecting subjective QoE such as the recency effect [56].

Longer video sequences were considered in [145], where video delivery over HAS was simulated on tablet devices. The authors studied combinations of bitrate changes and rebuffering events, but their analysis was limited to 6 sequences, 3 playout scenarios and 26 subjects. Longer video sequences were also used in [51], using video contents ranging from 30 to 60 sec. The authors studied the effect of rebuffering events as a function of location and density in a video sequence. However, temporal ratings were not collected, hence their analysis was based only on a final summary rating (retrospective score). As we will show later, retrospective ratings tend to be affected by recency biases. The study of temporal pooling techniques in [136] also included longer video sequences, and concluded that current temporal pooling strategies are mostly effective on short videos. On the long videos they used, simple mean pooling was found to be superior to all other methods. However, they only used two video contents in their analysis.

In all previous studies, playout patterns were chosen without considering the role of bandwidth usage. Generally, subject rejection strategies have been based only on retrospective scores or conducted on a per frame basis. We argue that such methodologies are inappropriate when gathering temporal

scores, particularly when studying the complex temporal effects that affect subjective QoE.

To sum up, previous efforts suffer from at least one of the following:

1. a small number of contents, playout patterns or number of subjects
2. a lack of practical network or buffer constraints on the subjective test design
3. use of short video sequences that do not capture long term temporal effects
4. not including both temporal and retrospective QoE scores
5. not deploying temporal subject rejection methods

Here, we describe a set of experiments that we conducted to gather data that will help us develop tools to create perceptually optimized network allocation protocols. We conducted experiments to measure subjective QoE in a typical mobile video streaming setting, where the human subjects were exposed to diverse real-world content, realistic network conditions and client-based strategies, while viewing video sequences of durations of at least one minute, displayed on a small mobile screen at low bitrates.

The outcome of these experiments is the new LIVE-Netflix mobile VQA database, which consists of 112 distorted videos evaluated by over 55 human subjects on a mobile device. The publicly available video content as well as

metadata for all of the videos in the new database can be found at `http://live.ece.utexas.edu/research/LIVE_NFLXStudy/nflx_index.html`. The distorted videos were generated from 14 video contents of spatial resolution 1080p at 24, 25 and 30 fps by imposing a set of 8 different playout patterns including: dynamically changing H.264 compression rates, rebuffering events and mixtures of both. While more recent compression standards such as H.265/HEVC and VP9 are currently being developed, H.264 is currently the most widely used format. Further, while H.265 achieves higher efficiency than H.264 does, it is not conceptually different from H.264: it uses the same motion-compensated/transform/lossless entropy coding hybrid model and essentially the same coding tools. Therefore, coding artifacts are perceptually similar among these two codecs; we thus expect the results of this study to apply to H.265-based streaming, with appropriately lowered encoding bitrates.

The database contains 11 different types of content provided by Netflix (drama, action, comedy, anime etc.) and 3 publicly available video contents from the Consumer Digital Video Library (CDVL) [2]. To provide a more realistic viewing experience, the audio track was included and played without distortion when the subjects viewed each sequence. Figure 2.1 shows an example of the type of impairments introduced on the videos in the LIVE-Netflix Dataset.

Given the lack of available subjective datasets driven by practical network constraints or streaming client strategies, our goal was to design a dataset of significant practical value. Hence, we designed the LIVE-Netflix dataset

Figure 2.1: Network impairment simulation using H.264 compression (left) and rebuffering events (right). The red box indicates a compression artifact.

based on playout scenarios that are common when streaming under practical bandwidth constraints and buffer size limitations. We also gathered both continuous and retrospective QoE scores towards achieving a more complete understanding of how humans combine different aspects of temporal perception into a single, overall impression of QoE. We believe that this work offers the possibility to bring human behavior modeling in this context closer to traditional video quality assessment (VQA) research. To both demonstrate the value of the database, as well to provide an engineering comparison of practical worth, we evaluated various state-of-the-art VQA algorithms and temporal pooling strategies on the new database. We also extensively studied temporal effects on subject QoE by analyzing the collective and per video impairment behavior of the subjects.

Our analysis led us to draw various observations. First, we observed that rebuffering severely affected subject QoE regardless of the content. Therefore, subjects tended to prefer transient bitrate drops over rebuffering on low complexity contents, even when the selected bitrate was low. However, a con-

stant low bitrate - to avoid rebuffering - was not tolerated by subjects. Finally, the gathered subjective data strongly manifested known QoE phenomena such as the recency effect (more recent video segments have a disproportionate effect on perceived visual quality) and the non-linearity of human responses, but it also challenges the use of retrospective scores or global subject rejection methodologies for QoE assessment on long videos.

The rest of this Chapter is organized as follows. Section 2.2 describes the dataset design, the encoding pipeline and the source contents used. Section 2.3 presents the subjective testing methodology, and Section 2.4 discusses the processing of subjective scores and the proposed subject rejection method. Sections 2.5 and 2.6 analyze the collected retrospective and continuous QoE scores, while Section 2.7 explores the cognitive aspects of subjective QoE in light of the collected human data. Section 2.8 analyzes the performance of various VQA algorithms and Section 2.9 gives conclusions.

2.2 Subjective assessment of mobile video quality

2.2.1 Network Assumptions and Buffer Limitations

When designing resource allocation strategies, content providers seek to answer the question: given a fixed amount of network resources, which strategy delivers the highest possible QoE? We consider here the tradeoffs that occur on end-users' QoE when mediating between rebuffering events and bitrate reduction under a mobile low bitrate regime. To do so, we designed a set of realistic playout patterns, assuming the same network resources and same

buffer limitations. To simulate realistic network conditions, we used a channel with time-varying capacity, shown in Fig. 2.2. The available bandwidth starts at 250 kbps, followed by a temporary bandwidth drop to 100 kbps of duration $d = 22.2167$ seconds until the bandwidth recovers to its previous 250 kbps value. This simple example of a bandwidth drop can be used as a building block to simulate models of more complex network conditions. Using this available bandwidth model, we derived eight test patterns based on the premise that the average playout rate of the client side cannot exceed that of the average bandwidth. The only exception to this rule is when the client uses some of the available buffer. Next, we discuss the buffer usage aspects of the designed patterns.

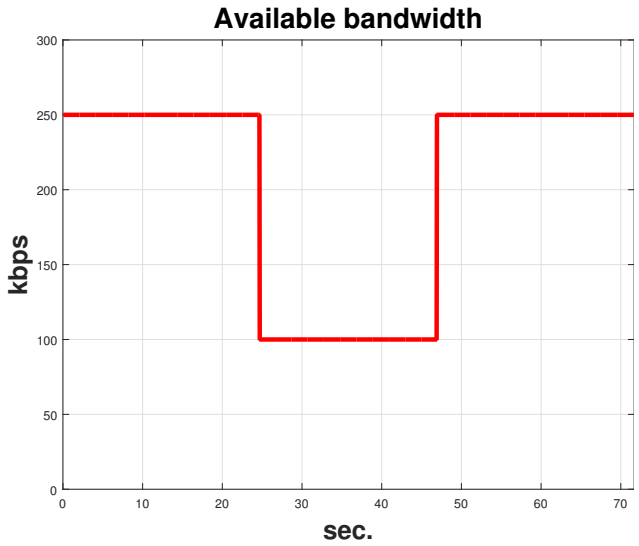

Figure 2.2: Available bandwidth model used in the LIVE-Netflix dataset. All of the test sequences were designed to consume the same amount of network resources (bandwidth).

To ensure the practical worth of the derived sequences, it is necessary to take into account the available buffer size. As shown in [61], a buffer-based strategy can be a simple and useful way to reduce the number of rebuffering events and bitrate switches that occur. Clearly, there are three possibilities:
1. The (instantaneous) playout rate is smaller than the (instantaneous) available bandwidth; the buffer is being filled with more data.
2. The playout rate is larger than the available bandwidth; the buffer is being emptied.
3. The playout rate is equal to the available bandwidth; the buffer state does not change over time.

Given our network assumption, we also considered a specific initial buffer state for streaming, where the buffer of size B_0 was filled with video chunks encoded at 250 kbps. We further assumed two possible initial buffer states: $B_0 = 1333$ kbits or $B_0 = 0$ kbit. The former scenario corresponds to "steady state" streaming where the initial buffer is filled, while the latter assumes that there is no initial buffer available. All patterns were designed so that the buffer is emptied at the end of the bandwidth drop shown in Fig. 2.2.

2.2.2 Playout Patterns

Based on the aforementioned network scenario and possible values for B_0, we simulated the following client approaches (see also Fig. 2.3 for an overview):

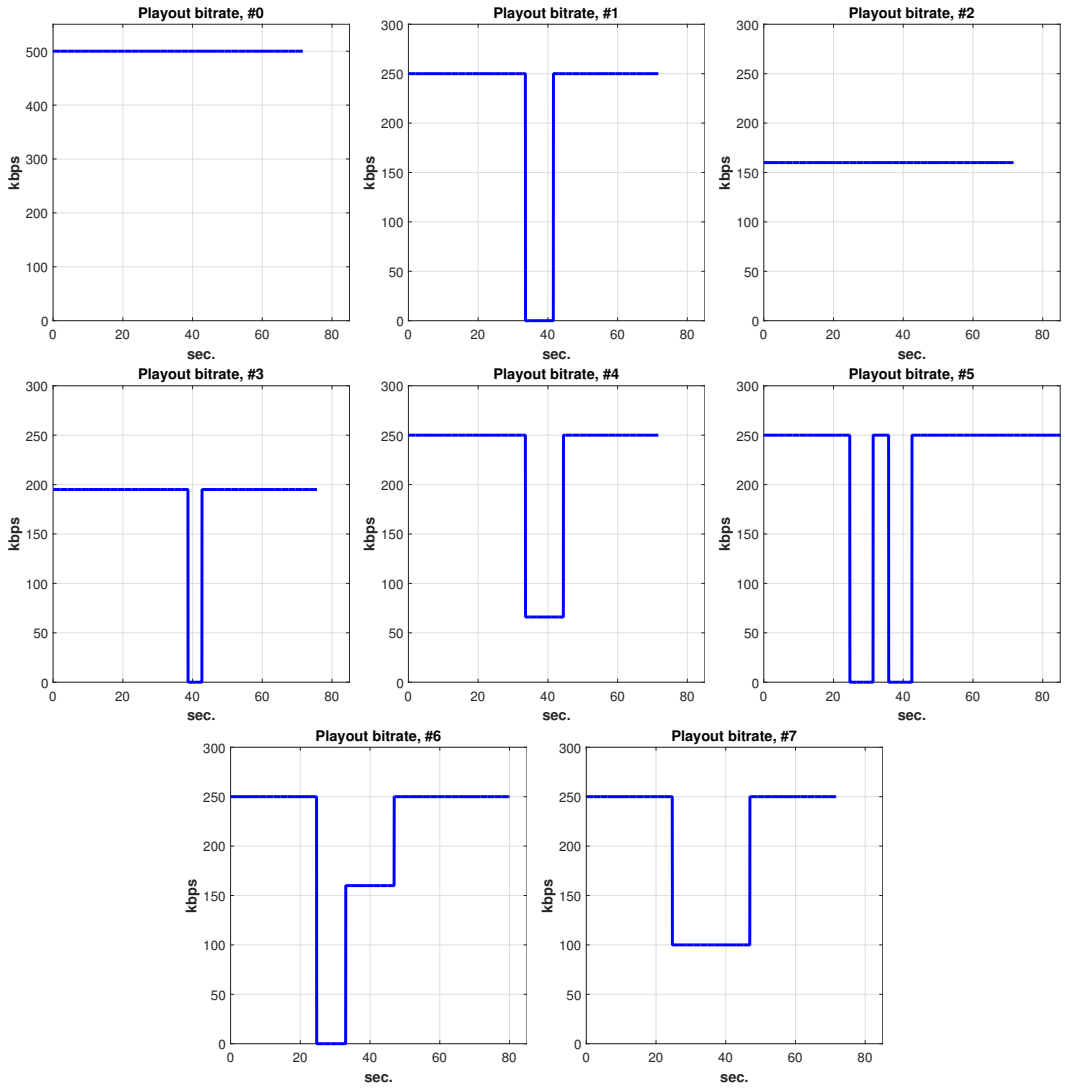

Figure 2.3: Playout patterns used in the subjective study. First row: patterns #0 until #3, second row: patterns #4 until #7. The horizontal axis corresponds to frame indices while the vertical corresponds to the instantaneous playout bitrate in kbps.

0. A constant encoding bitrate of 500 kbps. This playout pattern assumes an impairment-free network condition where the bandwidth is sufficient to allow such a playout rate by the client. In this case, the buffer is not used at all. This pattern is the only one that does not satisfy the bandwidth and buffer constraints. Although we included this pattern among the viewed playout patterns, it did not serve as a "hidden reference" [131].

1. One video chunk encoded at 250 kbps followed by an 8 sec. stall, followed by another 250 kbps chunk (see Fig. 2.4). The client drains the buffer completely before the rebuffering event occurs. Before the available bandwidth recovers, the client decides to resume playback after the 8 second rebuffer. By the end of the pattern, the buffer is emptied.

2. A single video chunk of $R_2 = 160$ kbps. The client side is very conservative throughout the video playback by always picking a playout rate of R_2, so that there is no rebuffering and the available buffer is depleted.

3. One video chunk encoded at 195 kbps, followed by a 4 sec. stall, followed by another 195 kbps chunk. Here, the client strategy is to reduce the rebuffering duration by half (4 sec.), by using a lower encoding bitrate. As before, during the rebuffering event, the client has a zero playout rate but an encoding bitrate of 100 kbps (equal to the available bandwidth) which allows the buffer level to partially recover and then be used to stream at 195 kbps before bandwidth recovers (see also Fig. 2.4).

4. One video chunk encoded at 250 kbps followed by a 66 kbps chunk, followed by another 250 kbps chunk. This playout pattern is an alternative to pattern #1, where the client tries to avoid any rebuffering events by switching to a lower playout rate (66 kbps) than the available bandwidth (100 kbps) during the bandwidth drop.

By removing the assumption on the availability of the buffer on the client side ($B_0 = 0$), a second set of playout patterns can also be simulated. This set of patterns is likely to deliver lower QoE scores to subjects since more severe impairments have to be introduced to deal with the bandwidth drop.

5. One video chunk at 250 kbps, followed by a 6.66 sec. rebuffering event, followed by a chunk at 250 kbps, followed by another 6.66 sec. rebuffering event, followed by the last 250 kbps chunk. In pattern #5, the unavailability of the buffer leads to rebuffering. By filling some of the buffer, the client is able to play out for a small interval of time at 250 kbps until the buffer is depleted. This leads to the rebuffering event, which is followed by a recovery at 250 kbps playout over a small time interval until the bandwidth also recovers.

6. One video chunk at 250 kbps, followed by a 8.33 sec. rebuffering event, followed by a chunk at 160 kbps, then a final video chunk at 250 kbps. Here, the client seeks to avoid a second rebuffering event by a gradual bitrate recovery.

7. One video chunk at 250 kbps is followed by a chunk at 100 kbps and then another chunk encoded at 250 kbps. Here it is assumed that the client is immediately able to adjust to the network conditions by using a playout rate that is always equal to the available bandwidth/encoding bitrate. This pattern may be the least practical among all the considered playout patterns. However, it is of interest to be able to study the subjective data resulting from such an "ideal" client reaction.

In the Appendix, we give an example of how some of the previous parameters were specified. Note that the original video sequences were of different durations, and that the playout patterns (of a given content) may also be of different durations because of delays introduced by rebuffering events.

2.2.3 Encoding Pipeline

We developed an encoding pipeline that generates the different parts of the final video and appropriately concatenates them based on an encoding map that indicates the time intervals of every quality level, the location and the duration of each rebuffering event. First, the source video stream (in H.264 format) was decoded, yielding an uncompressed raw .yuv file. The encoding map was then used to split the .yuv file in a frame-accurate manner, yielding .yuv chunks.

Meanwhile, the final frame of a video chunk immediately before a rebuffering event was used to generate a rebuffering video chunk. A customized

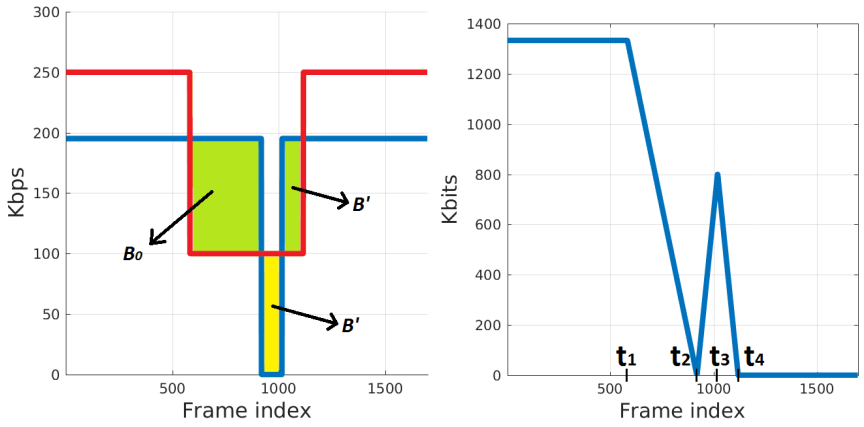

Figure 2.4: Left: Blue denotes the playout pattern #3 while red denotes the available bandwidth. The green areas correspond to buffer consumption while the yellow area indicates the buffer build-up. B_0 corresponds to the available buffer at the beginning of the bandwidth drop, while B' corresponds to the amount of buffered data being filled, then consumed by the client. Right: Available buffer level over time for playout pattern #3, $[t_1\ t_2]$: buffer drainage, $[t_2\ t_3]$: buffer build-up, $[t_3\ t_4]$: buffer drainage.

"loading", or spinning wheel, icon was overlaid on that frame and appropriately animated to simulate the desired video rebuffering effect. For playback purposes, and in order to match the rendering device resolution, all YUV frames were first upsampled to 1920×1080. An MP4 file was then created by lightly compressing these frames at CRF [121] (constant-rate-factor) value of 10. A more detailed description of the encoding pipeline that we used can be found in the Appendix.

2.2.4 Source Contents

A set of 14 video test contents were used containing a wide variety of spatiotemporal characteristics. Of the 14 contents, 11 belong to the Netflix

Figure 2.5: Some frames from the LIVE-Netflix dataset. From left to right: ElFuente and Chimera sequences from the dataset.

catalog of titles including action scenes, drama, adventure, anime and cartoons. The remaining 3 contents were obtained from the publicly available Consumer Digital Video Library (CDVL) [2]. A few frames from the video sequences are shown in Fig. 2.5. The test contents have a variety of frame rates and resolutions. For example, the ElFuente sequence has 4K resolution (4096x2160) and a frame rate of 60 fps, whereas most of the videos from the Netflix catalog have 1080p (1920x1080) resolution and frame rates of either 24, 25 or 30 fps. To deal with this difference, the ElFuente sequence was downscaled to 1080p and the frame rate was converted from 60 fps to 30 fps.

Measurements of spatial and temporal complexity give a rough idea of the content variety in a subjective database [161]. Let F_n denote the luminance channel of a video frame at time n and (i, j) the spatial coordinates of this frame. Next, consider the following simple Spatial Information (SI) and Temporal Information (TI) metrics [64]:

$$\text{SI} = \max_n \left\{ \text{std}_{i,j}[\text{Sobel}(F_n)] \right\}, \;\; \text{TI} = \max_n \left\{ \text{std}_{i,j}[M_n(i,j)] \right\}$$

where $M_n(i,j) = F_n(i,j) - F_{n-1}(i,j)$, $\text{std}_{i,j}(.)$ denotes the standard deviation over all pixels (i,j) and $\max_n$ denotes the maximum over all frames. As shown

in Fig. 2.6, the video content we use widely spans the SI-TI space [64].

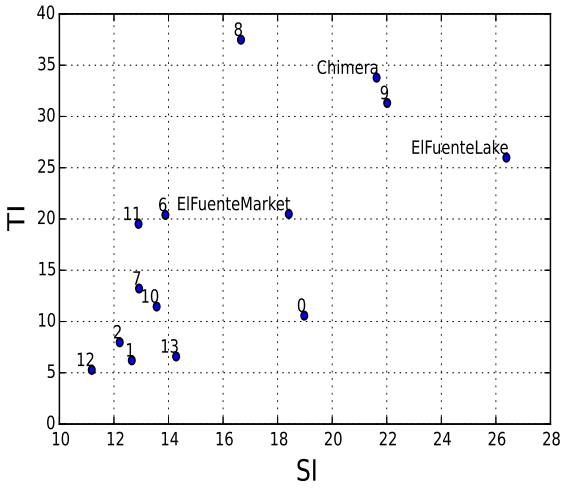

Figure 2.6: Spatial Information (SI) plotted against Temporal Information (TI) for the 14 video test contents in the LIVE-Netflix dataset.

2.3 Subjective Testing

2.3.1 Subjective Study Design

A single-stimulus continuous quality evaluation study [63] was conducted over a period of three weeks at The University of Texas at Austin's LIVE subjective testing lab. We collected subjective data from 56 subjects and a total of 4928 continuous scores together with the corresponding retrospective scores. Visual fatigue is an important consideration when designing subjective studies, so we split the study into three sessions, spaced by at least 24 hours to minimize subject fatigue [63]. Each session contained video content at most 35 minutes long, and the overall duration of each session was

about 45 minutes.

Due to necessary limitations on the duration of a subjective study, video QoE studies invariably must limit the number of different contents that are shown. When using longer video sequences, this is even more challenging. Driven by a desire to deploy as diverse and large set of contents as possible, we employed the following strategy. Each subject was assigned 11 contents (of the 14) in a circular fashion e.g. if subject i as assigned contents 1 through 11, then subject $i+1$ watched contents 2 through 12. This could result in a slightly different number of temporal and retrospective scores per content, but given the large number of subjects, we deemed this to be a statistically insignificant difference. All 8 playout patterns for these 11 contents were displayed to the subject only once. In order to remove any memory effects, we randomly shuffled the contents and the corresponding playout patterns while ensuring that the same content was not consecutively displayed to a subject in any session.

Android Studio was used to modify an earlier version of the human subject interface used in [100], which was made available to us by the authors. Using the previously described encoding pipeline, the generated .mp4 files were displayed on a Samsung S5 mobile device with a 1080p resolution and 5.1" screen size. This device had no problems playing the videos which were stored locally on an external SD card. The use of an external SD card did not introduce any latency when displaying the videos. The mobile device was not calibrated, but the brightness level was held constant at approximately

75% of maximum throughout the study. The sampling rate on the continuous scores was such that one score was measured per frame. Given the different frame rates of the input sequences, we parameterized the number of samples per video content depending on each video's frame rate.

2.3.2 Subjective Testing Walkthrough

Here we describe subjective testing procedure as it occurred during the first (training) session of each subject. Once seated, each subject was briefly instructed regarding the subjective testing process. They were asked to rate both their continuous and their overall QoE based on everything that they viewed on the screen. They were also asked not to make QoE judgments based on the level of interestingness of the video content or the audio quality. To remove any rating biases, the subjects were informed that there were no right or wrong answers in the experiment. No formal visual acuity test was performed, but the subjects verbally verified that they had normal or corrected-to-normal acuity. If a subject normally used corrective lenses when watching videos, they were asked to use them during the study.

Then, the subjects were introduced to the interface and the different video impairments they would be exposed to. Three different video contents, each with a different playout pattern were displayed as each subject became familiar with the testing interface. These contents were the same for all subjects but were not among the test contents used to gather the subjective data. After the first session, no training videos were shown, since subjects were assumed

to be adequately familiar with the testing procedure and interface.

The video sequences were displayed one after the other and a continuous scale rating bar was displayed at the bottom of the mobile device screen. The ratings on the continuous (Likert) scale ranged from 0 (Bad) to 5 (Excellent). After each video finished, the subjects were asked to give an overall rating of their QoE using the same rating bar. Then, a screen prompt allowed the subjects to take a short break before they could initiate the playout of the next video. Examples of these steps can be seen in Fig. 2.7.

Figure 2.7: Subjective testing interfaces. Left: continuous QoE scoring; Right: retrospective scoring.

2.4 Post Processing of Subjective Scores

2.4.1 Normalization of Subjective Scores

Following the subjective data collection, z-score normalization [131] was applied on a per session and per subject basis to account for differences in the use of the rating scale by each subject, for each of the 3 viewing sessions. Let $s_{ijk}(t)$ and f_{ijk} denote the continuous scores and the retrospective score assigned by subject i to video j during session k and let t denote the frame

number. Note that the set of all j videos viewed by subject i may not have been exactly the same for another subject i'. Consider the following operations:

$$\hat{s}_{ijk}(t) = \frac{s_{ijk}(t) - \mu_{s,ik}}{\sigma_{s,ik}}, \ \hat{f}_{ijk}(t) = \frac{f_{ijk} - \mu_{f,ik}}{\sigma_{f,ik}} \tag{2.1}$$

where $\mu_{s,ik}$, $\mu_{f,ik}$ are the mean continuous and retrospective scores assigned to all videos at session k of subject i and $\sigma_{s,ik}$, $\sigma_{f,ik}$ are the corresponding standard deviations. Since the generated video patterns are of different duration because of the introduction of rebuffering events, computing temporal Differential Mean Opinion Scores (DMOS) was not possible.

2.4.2 Subject Rejection using Continuous Scores

Using the subjective data in the form of z-scores, the next step was to apply subject rejection strategies to identify potential outliers in the rating process. In video quality studies with longer videos, it is possible that subjects demonstrate less motivation and/or attention on some videos than on others. While subject rejection is not a sophisticated model of human attention, we think that it is sufficient to filter out inattentive subjective responses. In a recent work, a model of subjective consistency and bias was proposed for recovering improved subjective scores in the retrospective QoE setting [80].

We believe that subject rejection methodologies based only on retrospective scores are questionable for the following two reasons. First, if some subject is rejected based on only a single score per video but then is also discarded from all other video sequences he or she viewed (as is typically done),

such a strict rejection criterion may needlessly reduce the amount of data. In our case, applying subject rejection only on the retrospective scores as suggested in [63, 131] led to 7 subjects being marked as outliers. Since we focused on the temporal effects of subjective QoE, we considered it sensible to enrich the subject rejection strategy by taking into account the temporal dimension of subjective QoE.

In our preliminary design of temporal subject rejection schemes, we experimented with simple heuristics. First, we applied the frame-to-frame equivalent of retrospective score rejection [38, 63, 131] which yielded inconsistent results. We believe this was due to the fact that introducing both dynamic bitrate changes and rebuffering events led to more complex subject reactions with different response and lag times. An alternative approach is to apply a simple thresholding method: discard subjects that are un-responsive during any rebuffering event. However, we encountered instances where subjects did not react to a rebuffering event but were very unforgiving of a second rebuffering. This observation led us to avoid using such simple *ad hoc* methods.

We instead deployed a more sophisticated dynamic time warping (DTW) [32] strategy on the subjective ratings to identify similarities in aligned *temporal* subject responses. Subjects that were completely un-responsive during a time period where most of the other subjects reacted were noted. To demonstrate the usefulness of the DTW approach to study and identify inconsistent temporal behavior among subjects, consider the examples shown in Fig. 2.8. Both examples depict the most and the least consistent human raters of a

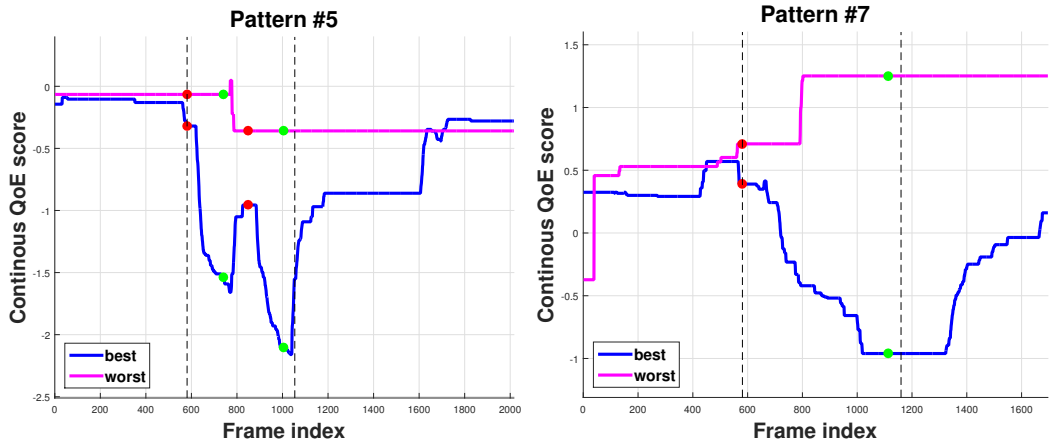

Figure 2.8: Temporal ratings with the highest (blue) and the lowest (purple) degree of consistency for two playout patterns in a given video content. Left: pattern #5; Right: pattern #7. The red dots denote the start of a video impairment (rebuffer or compression) while the green dots the end of the impairment. The dashed lines mark the time interval (in frames) used in the DTW.

given video sequence, one with two rebuffers and one with a bitrate drop to 100 kbps. In the first case, it is clear that the least consistent subject did not react to any of the rebuffering events, whereas the most consistent subject had a more predictable QoE reaction. Similar behavior occurs in the second case: the subject marked with blue lowered the QoE during the bitrate drop, while the least consistent subject had a highly unreliable QoE reaction: during the bitrate drop, the recorded QoE increased.

We now define the input to the DTW. Consider subject i and the temporal rating waveform s_{ij}, where j denotes a video content using one of the 8 playout patterns. Our main focus was occurrences of rebuffering or compression events since those are the key aspects that determine subjective

QoE. Therefore, we trimmed the s_{ij} waveforms by selecting the time interval between the first video impairment (rebuffer or bitrate change) that took place until the last one occured. To capture the temporal behavior when normal playback (playout rate of 250 kbps) resumed, we lengthened this time interval by 4 seconds. An example of a considered time interval can be seen in Fig. 2.8. We set the DTW window size to be 10% of s_{ij}. Similar values of the window size ranging between 5% and 10% yielded similar results.

We collected all warped distances between subjects i and k, i.e., $d_{ik} = \text{DTW}(s_{ij}, s_{kj})$, where d_{ik} denotes the temporal misalignment between subjects i and k. This is a measure of dis-similarity: a large d_{ik} could mean that subject i reacted very rapidly to some stimuli whereas subject k reacted more slowly. Subject ratings having large distances from most of the others can be thought of as unreliable. As we have already explained, however, only per video rejection decisions were made, i.e., if subject i had unreliable ratings on some video j it did not imply rejection of all the other subject's ratings. To eliminate biases introduced by the individuality of subject scoring strategies, each subject's continuous rating waveform was linearly scaled independently to cover the range $[0,\ 1]$.

Computing the DTW warped distances, d_{ik} yielded a matrix $\mathbf{D} = [d_{ik}]$ describing the temporal misalignments between all subjects that viewed video j. Since the DTW distance is symmetric, we computed only the upper triangular part of the matrix and set the diagonal entries to 0. Then, the sum of the DTW distances across the rows (or columns) of $\mathbf{D}$ may be considered

to be a measure of how unreliable a subject is: a large accumulated distance implies a subject whose responses were consistently mis-aligned with respect to other subjects when rating the same video.

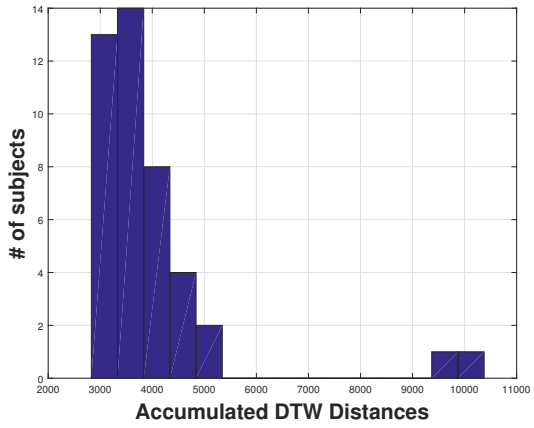

Figure 2.9: Distribution of accumulated DTW distances computed on one test video. The rightmost subjects have a higher chance of being outliers.

In Fig. 2.9, the distribution of accumulated DTW distances is shown for one of the test videos. The horizontal axis corresponds to the sum of the rows in $\mathbf{D}$, while the vertical axis indicates the number of subjects having the corresponding DTW distance. The distribution of accumulated distances is skewed to the right, making outlier identification more challenging. A standard technique is to apply Tukey's boxplot [152] rule, i.e., mark all observations that are smaller than or that exceed 1.5IQR as outliers, where IQR is the interquartile range Q_3-Q_1 where Q_1 is the 25th percentile and Q_3 the 75th percentile. However, this rule assumes an underlying normal distribution. To address the skewness of the data distribution, we can either transform the data using an appropriate transformation (e.g. a Box-Cox [129] transformation) or

use an adjusted boxplot technique like the one in [62]. We used the adjusted boxplot method. Then, an observation is considered to be an outlier if it lies outside the interval:

$$[Q_1 - h_l(\text{MC})\text{IQR}\ Q_3 - h_u(\text{MC})\text{IQR}] \tag{2.2}$$

where h_l and h_u are functions of the medcouple (MC), which is a skewness measure [62]. We used the exponential model proposed in [62] i.e. $h_l = 1.5\exp\{^{\alpha\text{MC}}\}$ and $h_u = 1.5\exp\{^{\beta\text{MC}}\}$, where α and β are weighting factors. We picked $\alpha = -4$ (default value) and $\beta = -1$ since the DTW distributions are right skewed, and a small value of β produced a more robust estimator. Using this skewness-driven boxplot, we identified potential outliers on each test video and removed them from the collected data.

2.5 Analysis of Retrospective Scores

We next discuss how we analyzed the subject scores using retrospective scores. First, we considered the overall distribution of the retrospective MOS before z-scoring. Figure 2.10a shows the distribution of raw retrospective MOS. It can be observed that the scores varied over the interval [1.5, 4.5], hence the entire scale [0, 5] was not used. However, the subjects were not prompted to use the entire scale, since this could introduce bias. Instead they were allowed to give their natural responses. Also, note that patterns #1 to #7 were given similar MOS scores, while pattern #0 was consistently rated higher by subjects (over all contents). This not surprising since this pattern

assumes a rebuffering-free scenario where the encoding bitrate is a constant 500 kbps.

In typical streaming applications, subjects are exposed to long video sequences, and events that occur early on may have less effect on the overall rating given by a subject. This is known as the "recency effect" [56] where recent events more heavily influence the current perception of one's viewing experiences.

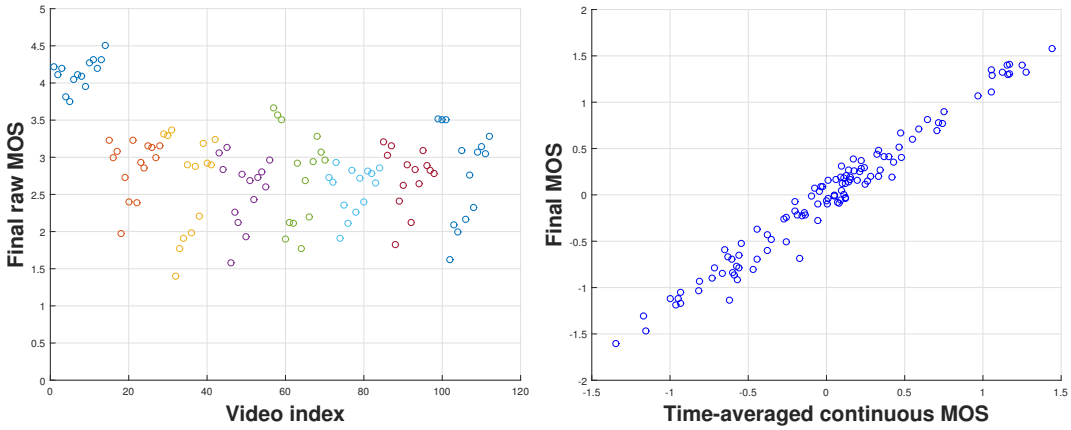

Figure 2.10: a) Raw MOS for all 8 patterns. Only pattern #0 is significantly different from the other 7. b) Scatter plot of the frame-averaged continuous scores (horizontal axis) against the retrospective MOS (vertical axis) for all test videos.

To examine these biases further, we conducted a preliminary statistical analysis to determine whether the playout patterns were actually (retrospective) scored differently by the subjects. We verified that the score distributions were not very skewed, then applied the Wilcoxon ranksum test [139] (using significance level $\alpha = 0.05$). We observed that, in many cases, the

statistical comparisons between the retrospective scores assigned to playout patterns yielded statistically insignificant differences. This could be explained by recency (latest experiences matter for retrospective evaluations) and the duration neglect effect [56]: subjects may lower their temporal scores if a long lasting video impairment occurs. However, even if they did recall the duration of an impairment, they tended to be insensitive to its duration when making retrospective QoE evaluations. Also, note that, by the time the subjects were asked to give an overall evaluation of each test video, more than 15 or 20 seconds of the 250 kbps playout had occurred. Given the tendency of subjects to evaluate videos based on more recent experiences, the test videos were possibly rated in response to the most recent video behavior.

If one is seeking a simple and direct QoE analysis, then it would seem desirable to obtain a single QoE value for each test video. Since the retrospective scores are affected by recency and duration neglect, we used simple frame averaging on the temporal scores to obtain a summary rating of each test video. Unfortunately, averaging continuous subjective scores without first applying temporal alignment does not account for the temporal QoE behavior of each subject (such as subject response delays). However, the DTW is appropriate only for pairwise time-series alignment, and may not produce an output having the same duration as the original waveforms. In our search for a recency-insensitive summary rating, we found that simple averaging correlated well with the retrospective scores, as seen in Fig. 2.10b. This observation aligns with two previous subjective studies: one where the test videos lasted

only 10 seconds [131] and one with longer videos [136].

Apart from frame averaging, we were also interested in explicitly capturing the subjective responses due to the impairments caused by the available bandwidth drop. In order to study those time intervals where the only visual impairments were due to the available bandwidth drop, we applied the following protocols: on patterns #1, #4, #5, #6 and #7, we applied averaging on all frames after the available bandwidth drop occurs. By contrast, on patterns #2 and #3, where there was heavier compression even prior to the bandwidth drop, all the frames were used. Since pattern #0 was impairment-free, we did not include it in the comparisons.

Using the averaged scores as the summary ratings, we compared the playout patterns of each content as shown in Fig. 2.11. Clearly, the ratings given to patterns #5 and #6, which belong to the second category, where no buffer was utilized, were statistically inferior to those of the patterns from the first category (#1 to #4), since the available buffer was zero and fewer bits were spent; hence there were more frozen frames due to rebuffering events and/or lower bitrate values. By comparing pattern #4 with #2 we observed that a consistently low bitrate value (to avoid rebuffering), as in the "conservative" client strategy #2, was not tolerated by subjects. Further, subjects preferred a long rebuffering (#1) if it meant better quality elsewhere rather than the combination of a short rebuffering event combined with an intermediate recovery bitrate (#3).

An important aspect of the interactions between rebuffering and com-

pression is whether there exists a "compression threshold", i.e., a bitrate level below which rebuffering will be preferred over a highly compressed stream. Clearly, this "threshold" may be different across contents depending on the content's spatio-temporal complexity. Here, we can perform such a comparison directly, since both playback states (normal playback at a much lower bitrate as in #4 and playback interruption as in #1) are equalized in terms of bandwidth usage.

By comparing rebuffering with transient bitrate drops (see first row and fourth column of Fig. 2.11) we found that the outcome of the statistical comparison depended on the level of content complexity. Out of the 14 contents, subjects preferred a very low bitrate in 4 of them, rebuffering in 3 and for the remaining 7, the statistical test yielded a statistical equivalence betwen #1 and #4. Notably, all 4 contents where subjects preferred #4 were slow motion scenes (e.g. a dialogue between actors) and/or low spatial complexity scenes, while the 3 contents where rebuffering was preferred were contents of either high spatial complexity (as in the ElFuente fountain scene) or high temporal complexity (e.g. a fight scene rich in motion); hence they required more bits to be encoded. This observation strongly highlights the trade-off between rebuffering and compression artifacts in perceived QoE.

Notably, pattern #7 had the best performance among patterns in the second category ($B_0 = 0$) and was comparable to #2 and #3. Again, this shows that subjects preferred transient bitrate drops. Surprisingly, #7 used fewer bits than #2 and #3 but yielded similar QoE. While #7 assumed an

ideal client that could immediately adapt to the network conditions, this comparison demonstrates the merits of QoE-aware network policies: using fewer bits does not always mean that perceived quality is lower. However, we also observed that patterns #5 and #6 were statistically indistinguishable over all contents. This brings up another aspect of the subjective test's design: apart from recency, allocating the same number of bits under these circumstances could signify a similar retrospective QoE or summary rating. This underlines the need to exploit the temporal aspects of QoE, since retrospective ratings reveal only some aspects of subject QoE.

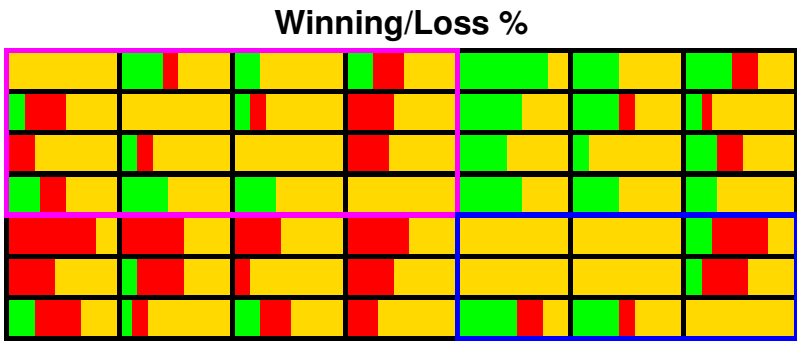

Figure 2.11: Wilcoxon ranksum test using $\alpha = 0.05$ on the averaged temporal scores for all patterns, represented as a 7×7 matrix. Each entry shows the winning percentage of the row compared to the column for all 14 video contents. Green shows the number of contents that the pattern in the row is QoE superior to the column, red shows the contents where the row is inferior to the column and orange shows that the row and column are indistinguishable. The purple box shows the comparisons only between patterns #1 to #4 ($B_0 = 1333$ kbits) and the blue box shows the comparisons only between patterns #5 to #7 ($B_0 = 0$ kbits).

2.6 Analysis of Temporal Scores

Temporal scores are a rich source of subjective QoE. Similar to the frame averaging used earlier, we performed frame averaging on the continuous subjective scores and show the result for several patterns in Fig. 2.12. We now focus on a comparison between patterns #1 and #7. Clearly, rebuffering (#1) severely and sharply damages subjective QoE for all contents. Further, the QoE recovers at a slower pace than it originally dropped, suggestive of the hysteresis phenomenon: there is a lag between subjective QoE scores and current video quality or playback status. We earlier observed that subjects were not forgiving of rebuffering events. By contrast, when the bitrate dropped from 250 to 100 kbps, the subjective QoE reactions varied depending on each content. On scenes having higher spatiotemporal complexities, compression artifacts may be more visible and affect the QoE heavily and sharply, while others may not be affected to the same extent. Similar observations may be made for all patterns that contain at least one rebuffering event (where the video freezes and the rebuffering icon appears), which are obvious and unpleasant to viewers, whereas bitrate drops have a different impact on subjective QoE depending on each content's complexity.

Notably, the constant encoding bitrate employed in #2 had a temporally varying effect on the perceived QoE. Given the long duration of the video contents and the different video characteristics present in each content (such as scene changes), it is clear that the subjects' QoE also changed over time even when the encoding scheme was static. This observation strongly supports

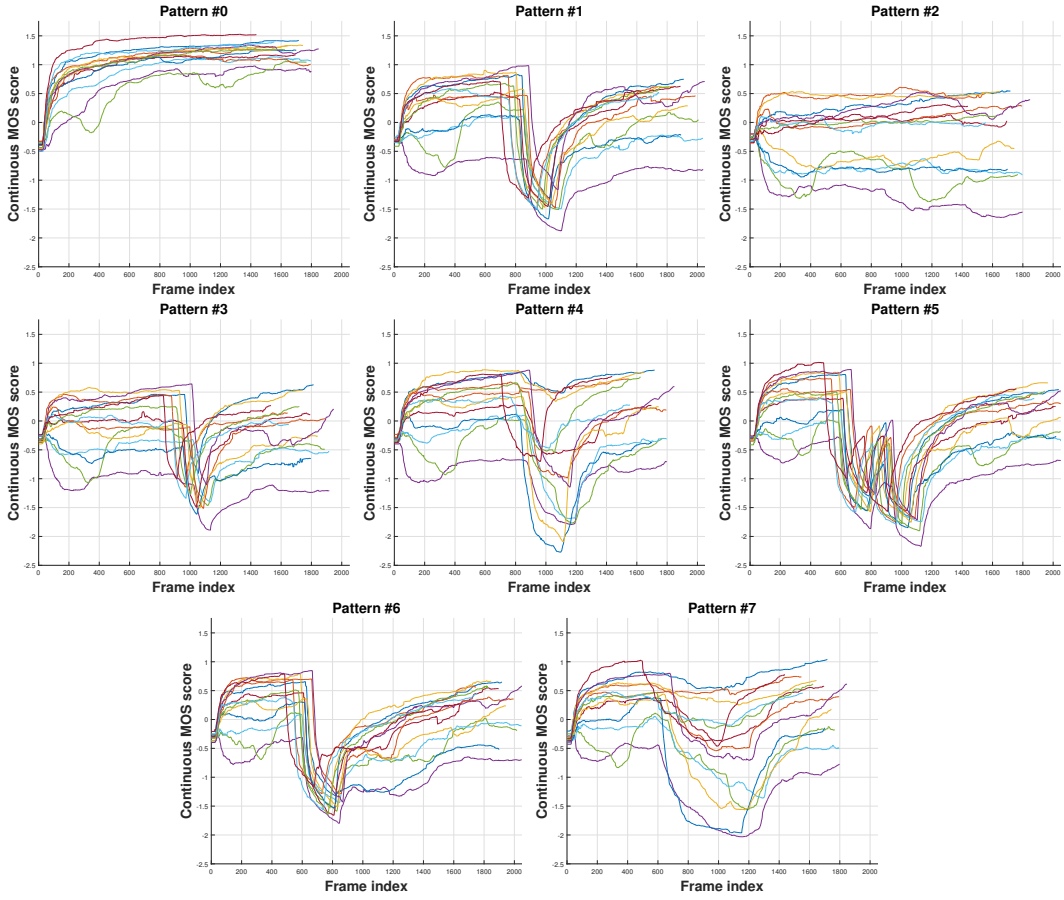

Figure 2.12: Temporal ratings across all contents for all playout patterns after subject rejection. First row: patterns 0 to 3; second row: patterns 4 to 7.

a "per chunk" encoding strategy [42], where each video content is first split into short video chunks and then, based on the video complexity during this chunk, an appropriate encoding scheme can be chosen.

To investigate the interplay between rebuffering and compression artifacts under a different light, we split the test contents into two sets based on their complexity: Set 1 includes source contents of low complexity and Set 2 those of higher complexity. To determine the two sets we considered the following: contents with high motion and/or spatial complexity require more encoding bits, hence subjective scores would likely be lower on such sequences. To determine content complexity, the authors of [48] defined a criticality measure as the logarithm of the sum of the SI and TI indices.

Given that the quality impairments of the otherwise very high quality videos being viewed are dominated by H.264 compression, an excellent measure of the content complexity to a fixed bitrate are the scores of a high performance objective quality engine such as ST-RRED [143]. ST-RRED is an information-theoretic approach to VQA that builds on the innovations in [137, 138]. It achieves quality prediction efficiency without the need to compute motion vectors unlike [133, 134].

To avoid any subjective biases due to content, we computed ST-RRED [143] between the original pristine video and #2 - constant encoding bitrate under the same total bit budget constraint. The computed ST-RRED value (on the constant bitrate encodes) was a way of describing the content complexity: the higher the ST-RRED value, the less "complex" the content was

assumed to be. As we will show later, ST-RRED performed the best among the VQA models studied across the subset of video sequences without any rebuffering, hence it was deemed suitable for this purpose. Finally, as shown in Fig. 2.13, there are 5 contents (shown in red color) that have a relatively higher encoding complexity than the rest. Therefore, we considered those 5 as Set 2 while the rest were assigned to Set 1.

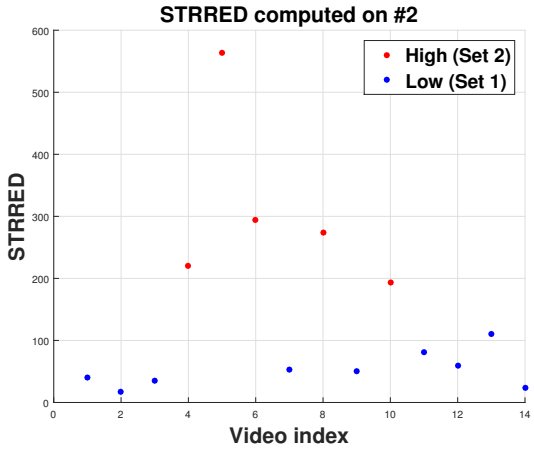

Figure 2.13: ST-RRED values between pattern #2 and the original source video for all 14 contents. Blue points correspond to low complexity contents while red points correspond to high complexity contents.

Next, we found the average (per frame) MOS score over all contents for each of the 8 different patterns, as shown in Fig. 2.14. The effects of content complexity were evident: after a rebuffering event occurred, the QoE recovered more slowly for contents in Set 2 (high complexity) as shown by the green arrow in #6. Meanwhile, the videos in Set 2 tended to have larger standard errors against the videos in Set 1, since the increased encoding complexity may have led to a larger variance in the subjective QoE reactions. Overall, during

normal playback, the contents in Set 2 have a lower QoE than the contents in Set 1.

We also observed the following interaction: a relatively long rebuffer event (as in playout patterns #1 and #6) led to larger drops in the reported subjective QoE on Set 1, as compared to Set 2 (see the black arrows in the plots for playout patterns #1 and #6). It is likely that the subjects were more annoyed by rebuffering events when they occurred during the playback of higher quality video content. A similar observation was also made in [45] using retrospective QoE ratings on short video sequences. However, for shorter rebuffering events (playout patterns #3 and #5) quality drops due to rebuffering between the two sets was similar. Notably, the second rebuffering in pattern #5 led to the opposite effect: given that one rebuffering event had already occurred, the quality drop on Set 2 was larger than the one for Set 1. This may be attributed to the effects of memory of a recent rebuffering event on currently perceived QoE.

By comparing patterns #1, #3 and #5, it is also evident that when the number or the durations of the rebuffering events increases, there is a larger drop in the temporal QoE scores. Again, these effects of rebuffering on the subjective QoE were harder to capture when we used the retrospective QoE ratings.

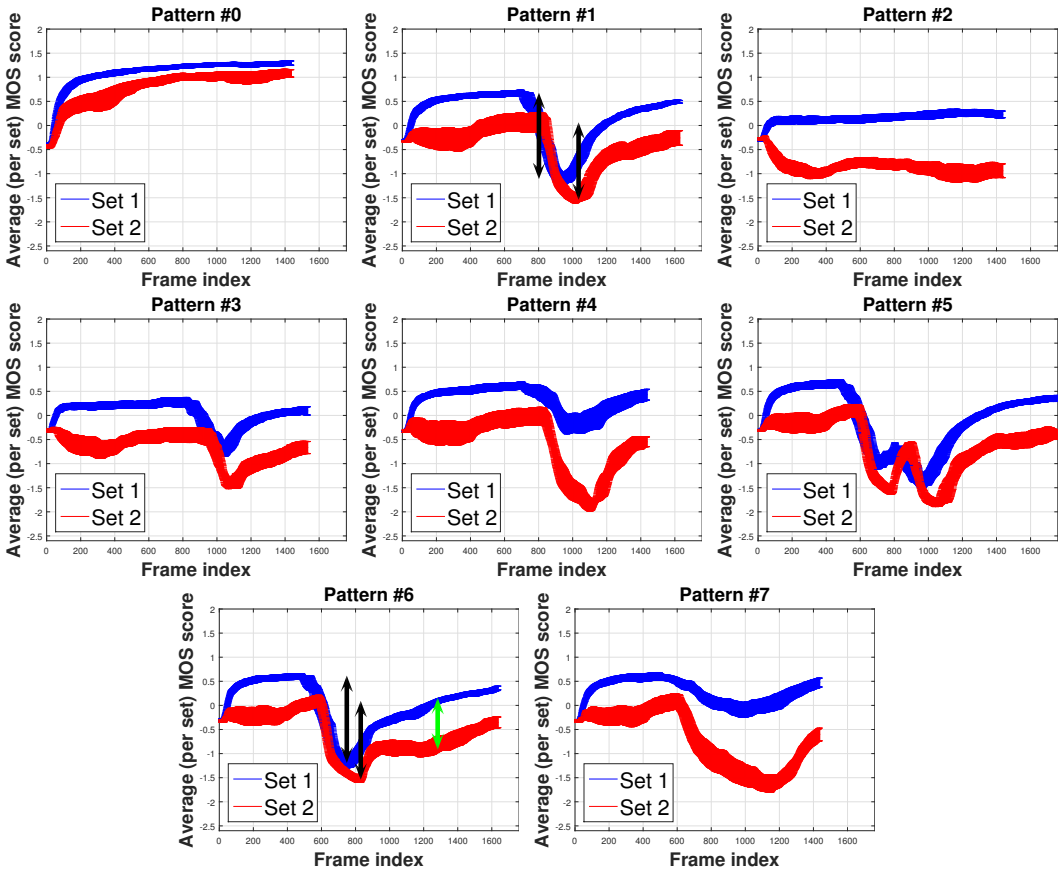

Figure 2.14: Averaged temporal ratings and standard errors for content Sets 1 and 2 for all playout patterns after subject rejection. First row: patterns 0 to 3; second row: patterns 4 to 7. Due to the different video lengths, we trimmed the axis of the plot to the duration of the shortest video sequence. The black arrows show the effect of rebuffering for the high vs. low complexity sets. The green arrow shows the different rates of QoE recovery for these sets.

2.7 Cognitive Aspects in Subjective QoE

2.7.1 Recency Effects

As already discussed, subject QoE might depend heavily on more recent experiences. To further investigate this claim, we performed local averaging on the temporal scores using a sliding window, then measured the correlations of those averages against the retrospective scores. Let κ denote the size of the sliding window in seconds, τ be the total duration of a video and $\mu(a,\ b)$ be the average of the temporal scores from frame a to frame b and f be the retrospective score assigned to that video. Figure 2.15 shows the SROCC between $\mu(a,\ b)$ and f using $\kappa = 10$ seconds. It is clear that local temporal

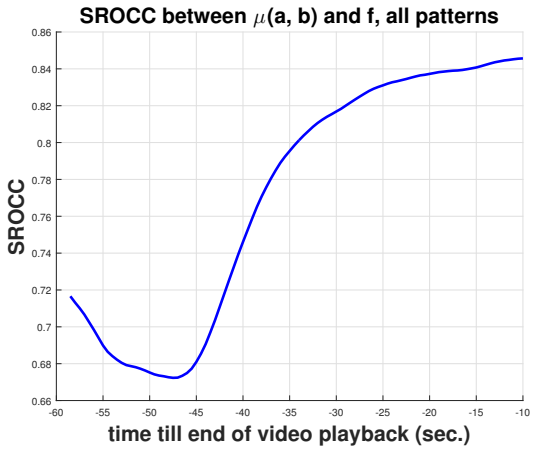

Figure 2.15: SROCC between the averaged temporal scores (over a 10 sec. window) and the retrospective MOS.

averaging produced stronger correlations over the more recent time intervals. This agrees strongly with the recency effect observed on the subjects' QoE.

2.7.2 Non-linearities in Subjective QoE

Non-linearities in human responses to video quality are usually not considered in depth. Here, we are able to examine these effects given the richness of the collected temporal data. Fig. 2.15 shows that, as the observation window is increased further into the past, the rank correlation decreases until approximately 45 seconds, at which point it increases. This could be due to the fact that after the first 15 seconds most of the video impairments begin to occur, hence a local temporal window of "high disagreement" between subjects occurs as the impairments take place. By high disagreement, we refer to different response times between subjects, different recovery times and different use of the rating scale. Note that even after z-scoring normalization, the subject ratings are still dependent on the rating behavior over time. We refer to both bitrate changes and rebuffering events during those time intervals as "events" where non-linearities in the human responses are activated and intensified. As a result, linearly combining the scores still produces non-linear measurements that do not correlate as well as when such events are not taking place.

To examine our hypothesis, we considered three different cases in Fig. 2.16: when the encoding bitrate is constant (patterns #0 and #2), when there is a single event (or two consecutive events) such as a lone rebuffering event or one followed by a bitrate drop (patterns #1, #3, #4, #6 and #7) and when there are two distinct events (pattern #5). The first case demonstrates the recency effect: more recent scores correlate more highly with the retrospective score. In the second case, a combination of recency and human non-linearities

is demonstrated: past experiences correlate less with the retrospective score, especially when there is a bitrate drop or a rebuffering event. However, recency by itself is not enough to explain subject QoE when a very negative QoE experience has occurred in the past. As shown in the third case, the correlation is much lower even over very recent time intervals due to the two rebuffering events that have happened earlier.

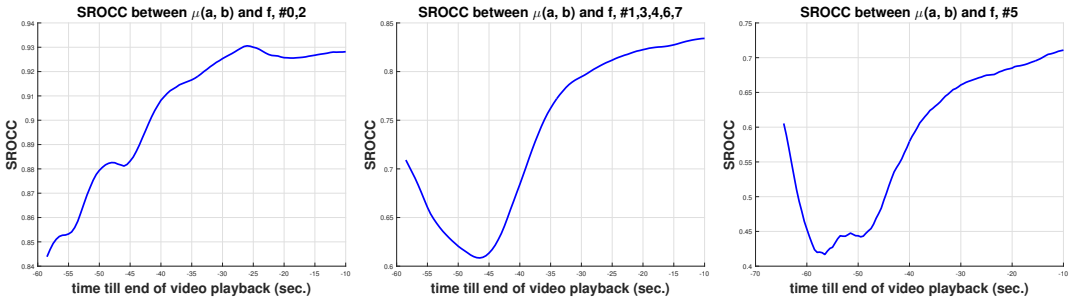

Figure 2.16: Spearman's rank correlation coefficient for different pattern sets. From left to right: patterns 0 and 2, patterns 1, 3, 4, 6, 7 and pattern 5.

2.7.3 Recency vs. Primacy

The previous analysis gives rise to the following contradiction: if subjects tend to bias their ratings based on the recency effect, why would a rebuffering event (or a bitrate drop) that happened much earlier matter? In the cognitive science literature, the *primacy* effect refers to the human tendency to recall events that occurred at the beginning of a series of events [55]. We can apply this concept to the various events to which subjects are exposed when viewing streaming videos, such as bitrate changes. It is likely that, when giving a retrospective evaluation, both primacy and recency effects affect the

subjects' responses. If the perceived video quality is relatively stable, subjects tend to internally rely more heavily on their latest experiences to make a retrospective decision, yet negative QoE events that occur early on can also activate longer term reactions.

Given our previous analysis of both retrospective and temporal scores, it is important to summarize the different aspects of each. For long video sequences in streaming applications, retrospective scores are simple and efficient QoE descriptors but do not capture all aspects of QoE. When integrating their temporal experiences into a single QoE score, subjects may be biased towards recent experiences (recency) or much earlier but memorable - typically unpleasant - ones (primacy), but they may also be insensitive to how long these unpleasant viewing experiences were (duration neglect). By contrast, temporal scores are rich and descriptive QoE indicators. However, the different response times between subjects and other temporally varying QoE aspects make temporal scores harder to analyze.

2.8 Objective Video Quality Assessment

2.8.1 Is Objective VQA Enough?

Most VQA algorithms are not applicable to frame freezes; hence video sequences with playback interruptions are usually not considered in objective quality analysis studies [100]. As a way of understanding how well these "standard" VQA models predict subjective QoE, we ask the question: "How well do VQA algorithms perform on video sequences when excluding frame freezes?"

To answer this question, we considered the set S_q of videos without any rebuffering, the set S_r of videos having at least one rebuffering event and the whole dataset ($S_{all} = S_q \bigcup S_r$). Clearly, S_r and S_q are disjoint. Then, we applied various quality metrics on S_q and S_{all}. We compared several leading full reference (FR), reduced reference (RR) and no reference (NR) image (IQA) or video (VQA) quality assessment algorithms [34, 99]: PSNR, PSNRhvs [113], SSIM [159], MS-SSIM [160], NIQE [94], VMAF [79], ST-RRED [143] and GMSD [164]. We refer the interested reader to [87, 109, 134] for other perceptual VQA models that have been developed. When applying them on the videos in S_q, we calculated the quality scores only on normal playback frames and measured the correlation with the retrospective scores after subject rejection. For PSNRhvs we used the publicly available Daala [10] implementation and for the other methods we used the available implementations. All models were applied on the luminance channel of the test videos and the black borders around the videos were removed. The results are presented in Table 2.1.

As shown in the first column, NIQE unsurprisingly performed the worst since it is a frame-based NR model, while PSNR and PSNRhvs performed the worst across all FR algorithms, followed by GMSD. The results on S_{all} were much lower than on S_q; indicating that the tested IQA/VQA systems were unable to predict QoE as well when rebuffering events were present. Note that SSIM performed better than MS-SSIM and close to the best predictor (ST-RRED) on S_{all}. This suggests that the subjects were internally responding strongly to rebuffering events rather than evaluating quality only. To investi-

Table 2.1: Spearman's rank correlation coefficient (SROCC) for various image/video quality assessment algorithms (IQA/VQA) against the retrospective scores after performing mean pooling on the no-rebuffering subset (S_q) and on the whole dataset (S_{all}). The best result per subset is in boldface.

IQA/VQA metric	S_q	S_{all}
PSNR (IQA, FR)	0.5535	0.5257
PSNRhvs [113] (IQA, FR)	0.5884	0.5465
SSIM [159] (IQA, FR)	0.7862	0.7230
MS-SSIM [160] (IQA, FR)	0.7647	0.6979
NIQE [94] (IQA, NR)	0.3811	0.1300
VMAF [79] (VQA, FR)	0.7607	0.6079
ST-RRED [143] (VQA, RR)	**0.8216**	**0.7257**
GMSD [164] (IQA, FR)	0.6665	0.5937

gate the performance of VQA models on videos afflicted by rebuffering, Fig. 2.17 shows the performance of ST-RRED on videos with rebuffering and on videos without rebuffering coded by color and symbol shape. It is important to observe that the predictive power of ST-RRED decreases when rebuffering events are introduced, which is not surprising: almost all perceptual IQA/VQA models only consider the effects of visual quality on the perceived QoE. Therefore, in the presence of rebuffering, objective video quality models become less reliable predictors of subjective QoE. This implies the need to develop more general QoE-aware methods. In the next Chapter, we will be introducing such models and evaluating their performance on the LIVE-NFLX dataset.

2.8.2 Temporal Pooling Strategies for Objective VQA

Simple averaging of frame quality scores is broadly used to pool quality scores computed on short videos, but more sophisticated perception-driven

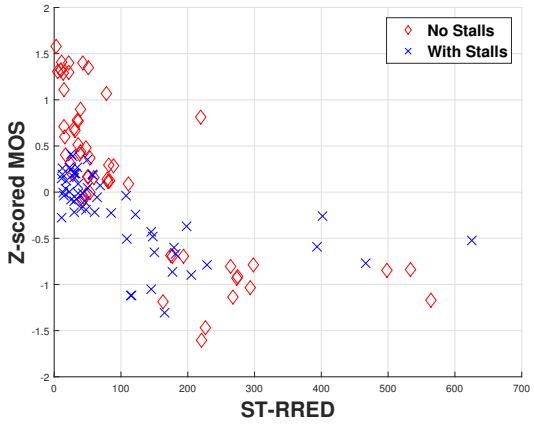

Figure 2.17: Performance of ST-RRED on videos with and w/o rebuffering.

temporal pooling strategies have been proposed, including hysteresis [132], VQ pooling [105] and temporal percentile pooling [98]. For percentile pooling, we sorted the frame-based values, then averaged the 5% of them which corresponded to lowest quality. We next investigated whether adopting these approaches could produce better correlations against human subjective scores on S_q and S_{all}. The results of this experiment are presented in Table 2.2.

On S_q, most methods benefited from a temporal pooling strategy, except ST-RRED (where performance was improved only slightly by percentile pooling). Otherwise, these improvements were not significant for S_{all}. This suggests that deploying temporal strategies designed for short sequences may not significantly improve QoE prediction on long sequences suffering from both rebuffering and bitrate changes: temporal pooling strategies operate on the numerical scores produced by objective video quality models. Again, ST-RRED performed the best on S_q in terms of SROCC. On S_{all}, SSIM (with hysteresis

pooling) was able to reach the maximum predictive performance of ST-RRED. Notably, percentile pooling was beneficial to FR methods such as SSIM and MS-SSIM on S_q, but in the case of NIQE, the prediction performance dropped considerably. This is likely because NIQE is frame-based, does not capture temporal information critical to QoE prediction, does not capture artifact fluctuations nor does it benefit from reference information. Therefore, NIQE scores may unreliably reach extreme values.

Table 2.2: SROCC against the retrospective scores achieved when using temporal pooling strategies on the LIVE-Netflix dataset, sets S_q and S_{all}. For each quality metric and subset (S_q/S_{all}), the best pooling method is in boldface. The best combination (quality model and pooling) per subset is in boldface and italic font.

Set	S_q				S_{all}			
Model/Pooling	mean	hysteresis	VQ	percentile	mean	hysteresis	VQ	percentile
PSNR	0.5535	0.5518	0.5621	**0.5869**	0.5257	0.5360	0.5398	**0.5581**
PSNRhvs [113]	0.5884	0.5960	0.6134	**0.6379**	0.5465	0.5601	0.5668	**0.5781**
SSIM [159]	0.7862	0.7971	0.7899	**0.8049**	0.7230	**0.7298**	0.7028	0.7051
MS-SSIM [160]	0.7647	0.7686	0.7593	**0.7800**	0.6979	**0.7037**	0.6680	0.6772
NIQE [94]	0.3811	0.4094	**0.4185**	0.2720	0.1300	0.1412	**0.1590**	0.0110
VMAF [79]	0.7607	**0.7760**	0.7679	0.7663	0.6079	**0.6347**	0.6116	0.5006
ST-RRED [143]	0.8216	0.8154	0.8032	**0.8232**	**0.7257**	0.7235	0.7139	0.7174
GMSD [164]	0.6665	0.6465	0.6416	**0.7502**	0.5937	0.5634	0.5684	**0.6369**

2.9 Discussion and Conclusion

We described a subjective study that focused on the temporal aspects of subjective video QoE under various network, buffer and low bitrate constraints. The study gathered both continuous time and retrospective data that we processed to extract useful information regarding those factors that

affect QoE, such as the network condition, the encoding bitrate and the spatio-temporal complexities of the video contents being viewed. Overall, we hope that QoE researchers find the new database to be a useful tool for studying the temporal aspects of subjective quality of experience. This remains a relatively unexplored area of research that poses many challenges.

Objective prediction models that incorporate spatio-temporal aspects of videos and that predict human reactions to both bitrate dynamics and rebuffering events could ultimately help streaming video companies address resource allocation problems more efficiently and in a user-adaptive way. In the following Chapters, we describe our research efforts towards this direction, by describing overall (endpoint) QoE prediction models [45], [24], as well as on continuous-time QoE prediction [23, 25, 28, 38] models that we have recently designed.

Chapter 3

Retrospective Quality of Experience Prediction

3.1 Introduction

In the previous Chapter, we demonstrated the need for QoE-aware prediction models that integrate video quality, stalling and memory measurements. In this Chapter, we present the Video Assessment of TemporaL Artifacts and Stalls (Video ATLAS) model, which integrates video quality, stalling and memory information to predict retrospective (overall) quality of experience. Video ATLAS is made publicly available at `http://live.ece.utexas.edu/research/VideoATLAS/vatlas_index.html`.

The rest of this Chapter is organized as follows. Section 3.2 discusses previous work on video streaming QoE prediction and Section 3.3 gives an overview of the subjective video QoE databases that we used to study and predict QoE. Section 3.4 describes the proposed feature-based QoE prediction model. Section 3.5 presents experimental results and Section 3.6 concludes the observations in this Chapter.

3.2 Previous work on QoE Prediction

There is a large variety of QoE prediction models that can be categorized by the type of information they use and the application environment. To facilitate a description of previous work on QoE prediction, we consider the following three categories of QoE prediction models.

3.2.1 QoE Prediction on Videos with Normal Playback

The most typical HAS scenario is to apply an adaptive bitrate allocation strategy such that bandwidth consumption is optimized. The effects of bitrate changes on the retrospective QoE may vary according to a number of QoE-related scene aspects: low-level content (slow/fast motion scenes), previous bitrates, frequency of bitrate shifts and their noticeability, the display device being used and so on [50, 100, 135, 150]. Apart from compression artifacts, HAS streams may also suffer from scaling artifacts, when the encoding resolution is less than the display resolution [79]. A commonality of these impairments is that there are no implied playback interruptions. These video quality degradations have been deeply studied within the context of video streaming [46, 50, 79, 135, 150] but also video quality [44, 100, 161].

To capture the perceptual effects of these video quality degradations, a wide variety of video quality assessment (VQA) models have been proposed. As already discussed, these models can be classified as full-reference (FR) [79, 87, 134, 155, 159, 160], reduced-reference (RR) [29, 77, 143] and no-reference (NR) [68, 127, 165, 166] models. The basic principle behind these

models is to model the statistical regularities of high-quality video frames and measure the deviations of a given (possibly) distorted video sequence. Based on these perceptually-driven VQA models, continuous-time QoE prediction models were designed in [23, 38] using Hammerstein-Wiener and non-linear autoregressive approaches; but these models do not account for the effects of stalling in HAS video QoE.

3.2.2 QoE Prediction on Videos with Playback Interruption

When the available bandwidth reaches a critical value (e.g. in a mobile streaming scenario), playback interruption is sometimes very difficult to avoid. While the effects of stalling on QoE are not yet well understood, various studies have shown that the duration, frequency and location of stalling events severely affects QoE [22, 43, 51, 52, 58, 135, 167]. In [58], the effects of initial delay were compared to those of stalling events that occur while watching. By making use of global stalling statistics, Quality of Service (QoS) models such as FTW [59] and VsQM [124] have been proposed. A parametric relationship between stalling and QoE was derived in [167] using cosine functions, but this approach makes it hard to integrate more inputs (such as video quality measurements), if needed. More recent efforts [52] have sought to both model the effects of stalling on user QoE, and to integrate them with models of recency [56, 57].

3.2.3 General QoE Prediction Models

Combining objective video quality models and stalling-related information into single QoE scores is a difficult proposition which is also partly due to the unavailability of suitable subjective data. Nevertheless, more general approaches that seek to combine video quality information together with stalling information have also been proposed. In [106] the effects of frame drops and image sharpness were separately modelled; then multiplied together yielding an overall QoE score. In [141], the authors fed Quantization Parameter (QP) values and stalling-related features into a Random Neural Network learning model to make QoE predictions. However, their method was evaluated on only 4 contents and on short video sequences of 16 seconds, hence they did not consider longer term memory effects which are prevalent on video streaming applications. Similarly, [163] weighted QP values against the impact of stalling, which was related to the motion complexity of the last decoded frame.

A shortcoming of these works is that video quality is ascribed or equated to average bitrate and/or QP values which do not carry as rich perceptual information as perceptually-motivated FR quality prediction algorithms (such as SSIM and MS-SSIM). More recently, the authors of [45] combined such FR approaches with stalling information, yielding the Streaming Quality Index (SQI). SQI was evaluated in the Waterloo database [45], which contains 10 second clips of videos afflicted by quality changes and stalling. While SQI delivered good performance and is simple to compute, it is not clear how to integrate other QoE-related inputs (such as audio quality). The continuous

QoE prediction problem was recently addressed using a neural network approach in [25], by integrating features computed using video quality models and stalling information.

In recent years, the VQEG/ITU-T P.1201-3 standard [3, 117, 123, 130] has also been proposed as a QoE prediction framework that combines audio-visual quality measurements together with stalling information. The standard determines three visual degradation types: upscaling (D_u), temporal (D_t) and quantization (D_q) degradation. Based on the amount of available video information (modes 0 to 3), P.1201-3 measures the quantization degradation D_q using a number of basic video-related attributes such as bitrate, resolution, frame size or QP. The upscaling degradation D_u is expressed as a logarithmic function of the scaling factor (ratio of display and encoding resolutions), while D_t is calculated using a parametric expression of framerate, D_u and D_q. The video quality module then sums the three terms together to describe the visual quality degradations due to scaling, compression and/or jerkiness. There has also been a Track 2 in P.NATS [11], where a pixel-based FR model was tested against the previous modes in P.1201-3.

Besides visual quality, the P.NATS approach also takes into account the audio quality per segment and the effects of stalling using multiple parametric expressions to derive the final QoE score. Our proposed approach shares similarities with these models in that we have also considered statistical descriptors (features) that capture the effects of stalling events and memory and combined them with video quality measurements. However, our line of work

is different in a number of ways.

We have deployed *perceptually-relevant* FR-VQA models which mimic properties of the human visual system and have been shown to highly correlate with human subjective scores. We do not express video quality as a parametric model of specific degradations, but instead exploit the statistical regularities of pristine images and video and measure perceptual quality as a deviation from these spatiotemporal regularities. Importantly, calculating the FR-VQA features requires access to the pristine video data which may be unavailable on the client side. Nevertheless, the VQA calculations can be carried out offline on the server side, where both the distorted and pristine segments are available, then sent to the client as part of the metadata. This allows for a QoE-driven, client-based adaptation strategy.

In P.1201-3, the final QoE score is estimated by summing a machine learning prediction and the product between audiovisual quality and stalling information, using many parameters that were determined by training on multiple databases. Instead, our data-driven approach uses a single SVR and explicitly describes the non-linear relationships between visual quality, stalling and memory, without assuming a parametric model. While we do not capture audio quality information in the proposed model, such information can be also integrated, if it is readily available. Lastly, the training data and the proposed QoE prediction model are publicly available to the research community.

The product of our work is the Video Assessment of TemporaL Artifacts and Stalls (Video ATLAS) model: a new QoE prediction model that integrates

objective, perceptually-driven FR-VQA models with stalling- and memory-related features to conduct retrospective HAS-QoE prediction in a unified way. To design our approach, we relied on two recently designed subjective video QoE databases which contain videos suffering from temporal rate/quality changes and stalling events. In the next section, we introduce these databases and highlight their use for training and evaluating our proposed model.

3.3 Subjective Video QoE Databases

A key component of the proposed Video ATLAS is that it relies on subjective ground truth data for training and testing purposes. To build accurate retrospective QoE predictors, it is important to collect a large number of subjective scores under different video impairments including, video quality changes (due to the multiple encoding bitstream representations of the high-quality source content), stalling events (due to throughput and buffer limitations) and combinations of the two. There have been plenty of subjective studies on HAS QoE [50, 135, 145, 146, 149, 150], but most existing video quality databases are not *publicly available* thereby hampering their practical benefit.

Other works consider these two impairment categories either in isolation [38, 51, 52] or in an *ad hoc* fashion, i.e., without a realistic network design in mind. In addition, due to the difficulty of designing and carrying out large video subjective studies, many of these datasets are of quite limited size in terms of video content, video duration and/or the number of participants.

Overall, the majority of these databases cannot be used to train and evaluate retrospective QoE predictors for HAS applications.

Towards filling this gap, we already discussed the design of the LIVE-NFLX Video QoE Database in the previous Chapter. Besides the LIVE-Netflix database (LIVE-Netflix DB), another *sizeable* and *publicly available* video QoE database that considers interactions between stalling and quality changes is the Waterloo Video QoE Database [45] (Waterloo DB). This recently developed database consists of 20 RAW HD 10 sec. reference videos. Each video was encoded using H.264 into three bitrate levels (500Kbps, 1500Kbps and 3000Kbps) yielding 60 compressed videos. For each of those sequences, two more categories of video sequences were created by simulating a 5 sec. stalling event either at the beginning or at the middle of the video sequence. In total, 200 video sequences were evaluated by more than 25 subjects. Based on the collected subjective data, the authors designed the Streaming QoE Index (SQI) to "account for the instantaneous quality degradation due to perceptual video presentation impairment, the playback stalling events, and the instantaneous interactions between them."

Unlike the LIVE-Netflix DB, Waterloo DB consists of short video sequences (which may not reflect the experiences of viewers watching minutes or hours of video content), used fewer subjects, and importantly, the stalling events and the bitrate/quality changes were not driven by any realistic assumptions on the available network or the buffer size. However, given its simplicity and the lack of availability of other public domain databases of this

type, applying our proposed model on this database may yield a comparison of practical worth.

3.4 Learning A QoE Predictor

In the previous Chapter, our experiments on the LIVE-NFLX database demonstrated that VQA algorithms, which do not consider playback interruptions, do not perform well. We also found that MS-SSIM, while an algorithm that has a provably better performance than its single scale counterpart (SSIM) [160], did not perform better than SSIM. Both of these observations verify the notion that quality assessment tools should be used under the application context and hence integrating QoE-aware information for HAS-QoE applications is highly relevant.

In this direction, we next describe a new feature-based model which integrates objective video quality, stalling-related and memory features to significantly improve QoE prediction. While Video ATLAS is not an explicit model of the cognitive properties that affect human QoE, it uses subjective data to capture effects related to the perceived QoE. Using machine learning methods to predict subjective quality is not a new concept [79, 127], but our goal here is to predict subjective HAS-QoE, which is a different concept. Figure 3.1 shows a block diagram of Video ATLAS, while Table 3.1 summarizes the attributes of our data-driven approach.

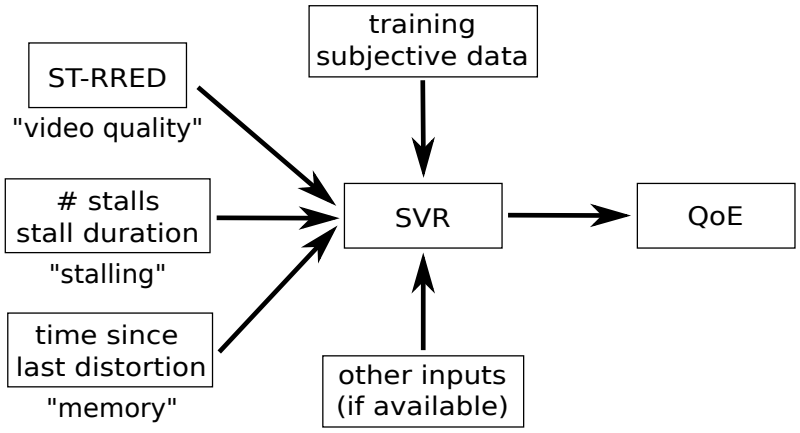

Figure 3.1: Outline of the Video ATLAS QoE predictor.

Table 3.1: Description of the various features used in Video ATLAS.

Name	Description
VQA	video quality feature, using ST-RRED.
R_1	duration of stalling event in sec.
R_2	number of stalling events
M	time since the last impairment
M_{stall}	time since the last stall
I	duration of the impairment in sec.

3.4.1 Proposed Model

Perceptual video quality during normal playback is a critical factor of overall perceived QoE; hence high-performing video quality models are desirable. However, not all video quality models perform the same. Recent experiments have demonstrated that ST-RRED [143] is a robust and high-performing video quality model when tested on a very wide spectrum of video quality datasets [13], on multiple resolution and device types.

ST-RRED is an information-theoretic approach to VQA that builds

on the innovations in [137]. It relies on decomposing video frames as well as frame differences using a steerable wavelet decomposition of both the spatial and temporal components. A Gaussian Scale Mixture model [156] of these wavelet coefficients is used to derive closed-form conditional entropy measurements on both the reference and distorted videos. ST-RRED produces a final quality score by differencing locally weighted spatial and temporal entropies between the reference and distorted videos. ST-RRED achieves quality prediction efficiency without the need to compute motion vectors, unlike [134]; hence it is highly suitable for streaming applications. Compared to using QP values or average bitrate, ST-RRED captures a number of perceptually-motivated properties, such as contrast masking, temporal masking and suprathreshold effects [143].

Since we are focused on predicting retrospective QoE scores, a pooling strategy was chosen that collapses per-frame objective quality measurements into a single value. A number of different pooling strategies have been proposed [100, 105, 136] that capture QoE aspects such as recency or the peak-end effect (the worst and best parts of an event affect the QoE more). For simplicity, we deployed simple averaging of the QoE scores as suggested in [136], reserving recency modeling as a separate input feature.

Stalling greatly impacts the perceived QoE; hence we require stalling descriptors that capture these stalling-induced QoE effects. It is clear from the literature (e.g. in [51, 59]) that the number or density of stalling events is a key factor of overall perceived QoE. When the number of stalling events increases,

the overall QoE tends to decrease. To account for the effects of stalling [51, 52, 59, 107, 115], we included the length of each stalling event measured in seconds (R_1) and the number of stalling events (R_2) as an input feature. The length of the stalling event(s) was normalized to the duration of each video.

While the previous features consider stalling and quality changes, we also computed the time (in sec.) per video over which a bitrate drop took place; following the simple notion that the relative amount of time that a video is more heavily distorted is directly related to the overall QoE. This feature (I) was normalized to the duration of each video.

Besides stalling and video quality, QoE is also driven by numerous cognitive (or memory) factors. For example, more recent experiences have a larger weight when making retrospective evaluations, also known as the recency effect [56]. Figure 3.2 demonstrates the underlying relationship between averaged continuous scores in the LIVE-NFLX database within the last 5 seconds of a video, and overall QoE scores. Memory is also activated by non-linear mechanisms such as recency, primacy and duration neglect [27, 55, 56].

To model the effects of memory/recency, we computed the time since the last stalling event or rate drop took place and was completed, *viz.*, the number of seconds with normal playback at the maximum possible bitrate until the end of the video. By using this feature, we seek to capture the relationship between the recency of QoE experiences, and the recorded retrospective subjective scores. This memory-related feature (M) was normalized to the duration of each video. Other memory-related approaches such as those in

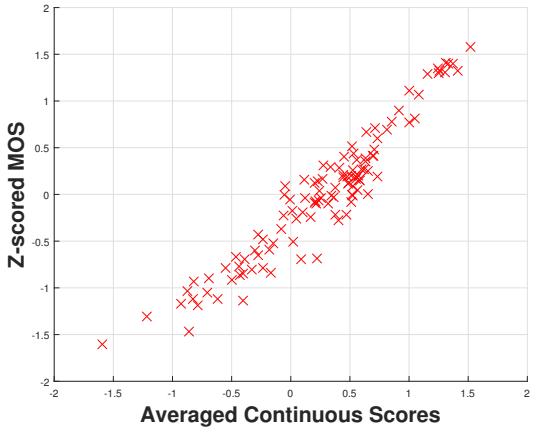

Figure 3.2: The recency bias strongly affects QoE: averaged (over the last 5 sec.) continuous scores are highly correlated with retrospective QoE scores. The retrospective scores were collected on 60-70 second video clips [27].

[57, 116, 167] could lead to performance improvements, but we chose a simpler, though highly efficient approach to model memory.

Video quality measurements, stalling events and memory have a non-linear relationship with QoE [27, 45, 50, 167]. For example, there is a fundamental difference between the perceptions of stalling events and compression artifacts: stalling is always noticeable and annoying to subjects, while compression artifacts may be less obvious on some scenes (such as those that can be easily encoded) [27]. To model the non-linearities between compression, stalling and memory, we used a Support Vector Regressor (SVR). SVRs are popular machine learning engines that can model the non-linear relationships between input features without the need for very large datasets and that have been successfully deployed in many other image/video quality applications e.g. in [127].

3.4.2 Feature Extraction

We now describe the feature extraction process. First, we remove black borders (if any) from the input videos, since those may affect the performance of VQA models. Next, consider all frame pairs $(i, i + j)$, where i indexes the ith frame of the pristine video and $i+j$ indexes the corresponding frame of the distorted video, where $j \geq 0$. If there are no stalling events in the distorted video, then $j = 0\ \forall i$; else determine j based on the number of stalled frames until this point. In other words, these two frames must be synchronized in order to be able to extract meaningful objective quality measurements. Next, apply ST-RRED to measure the per-frame objective quality, then average pool these values across frames to obtain a single quality-predictive feature that will be used later. In addition, all the other features are collected, assuming that for retrospective QoE prediction, the number of rebuffered frames as well as the locations of the bitrate changes are known. For ST-RRED, adjacent frames are needed to compute frame differences, hence we ensured that frame differencing takes place only between two consecutive frames that both have normal playback.

3.4.3 Video ATLAS as a General QoE Framework

A unique property of the proposed Video ATLAS model is that it is a simple and application-independent approach to predict streaming video QoE. We now discuss the general aspects of the proposed model.

During normal playback, any good video quality algorithm can be used

to measure objective QoE scores. Our method allows the use of any full reference (FR) or no reference (NR) image/video quality model as appropriate for the application context. In Appendix C.1, we showcase examples of using several models that are both highly compute-efficient and that deliver accurate VQA predictions, rather than using compute-intensive models [133, 134].

Meanwhile, a flexible number of input features can be considered depending on the application. As an example, the Waterloo Video QoE Database [45] (Waterloo DB) is, by design, simpler than the LIVE-NFLX DB; hence a smaller number of input features must be used when training or testing using this database. Video ATLAS can be easily trained and tested using a smaller number of features (see also Appendix). An enriched set of features can also be used, given a suitable database or application. Viewed in a different light, Video ATLAS can adapt to a diverse set of streaming applications, where both stalling and compression artifacts may or may not occur.

Lastly, by carefully designing the input features so that their values are normalized with respect to the video duration, Video ATLAS can be tested on diverse types of databases. In the experimental section, we demonstrate an example of training/testing using both the LIVE-NFLX and the Waterloo databases.

3.4.4 Practical Considerations and Limitations

The proposed model is feature-based; hence its performance may be adversely affected by the amount and quality of training data. Unlike many

computer vision applications where the number of features and the available training data are in the order of thousands (or even millions!), subjective data carefully collected in a controlled environment are of much lower dimension. Therefore, the Video ATLAS model is typically trained on approximately few hundreds of data points, using a small (but highly descriptive) number of features and uses regressors that are relevant to a small number of features and/or training data. Notably, given the problem's dimensionality, "deeper" learning approaches may not yield substantial performance improvements. In Appendix C.2, we show how varying the amount of training data affects the performance of Video ATLAS.

It is important to discuss how Video ATLAS can be deployed in a more practical setting. In a client-driven adaptive streaming scenario, the client is in better position to perform QoE calculations. Since the reference video is typically not available to the client and VQA calculations may be compute intensive, the VQA measurements can be calculated on the server side. Similarly, Video ATLAS can be trained on the server side, and the model parameters, together with the VQA information, can be shared with the client as part of the metadata. The client that is aware of the playback and buffer status, can then use the features to perform QoE prediction in real-time. In our experiments, training Video ATLAS can be carried out within few seconds and testing the model is even faster, which is due to the simplicity in the feature set and the SVR. In practice, and when the model is trained accordingly, these QoE predictions can be carried out on a per-chunk basis (e.g.

every 2 sec.), and can be used to drive QoE-aware bitrate selection decisions.

QoE is not only influenced by the three Video ATLAS input types: video quality, stalling and memory. It is also affected by other factors [110] such as resolution changes [21], varying audio quality, the display device, user expectations regarding the streaming service and/or the viewing environment, and the video content itself. Indeed, this is a limiting aspect of the training data in the LIVE-NFLX database, which does not fully address these factors. However, while Video ATLAS does not account for these factors, it is a flexible framework that can integrate such QoE-related inputs, if they are made available. Another important consideration is that Video ATLAS is trained on subjective data that does not capture every aspect of real-world visual QoE. As with any experimentally derived QoE model, it may have its own biases. Ultimately, the success of these kinds of models must be measured by their performance in real-world application.

3.5 Training and Evaluation of the Proposed Model

3.5.1 Experiments on the LIVE-Netflix Video QoE Database

To demonstrate the potential of the proposed method, we evaluated its performance in the LIVE-Netflix DB by conducting two different experiments. The first one (Experiment 1) consisted of creating two disjoint content sets: one for training and one for testing. Within each content (training or testing), all patterns were used for training or testing. While this is a common approach used to account for content dependencies in feature-based VQA methods, it

may also occur that the different "distortions" or playout patterns induce pattern dependencies, resulting in overestimation of the true predictive power of a feature-based method.

To examine pattern independence we also conducted a second experiment (Experiment 2), where we picked one of the playout patterns as a test pattern and the rest as training patterns. Thus, for each testing pattern there were 14 test points (one for each content) and 98 training points. On both tests, we applied the proposed model to predict the QoE scores of the test set, given the input training features and MOS scores. Since our model does not produce continuous scores, we only used the retrospective QoE scores from LIVE-Netflix DB.

After the proposed model was trained on a given set of training features and MOS scores, we applied regression on the test features to make QoE predictions. Then, we correlated the regressed values with the MOS scores in the test set and calculated the Spearman Rank Order Correlation Coefficient (SROCC) and the Pearson Linear Correlation Coefficients (LCC). The former measures the monotonicity of the regressed values and the latter the linearity of the output. Before computing the LCC, we first applied a non-linear (logistic) regression step on the output QoE scores of every method we compared, as suggested in [63]. This step is not needed for our feature-based approach, but we did it to ensure comparability with all other methods. Note that the SROCC is a non-parametric measure of correlation between subjective data and QoE predictions; hence non-linear regression is not needed to compute

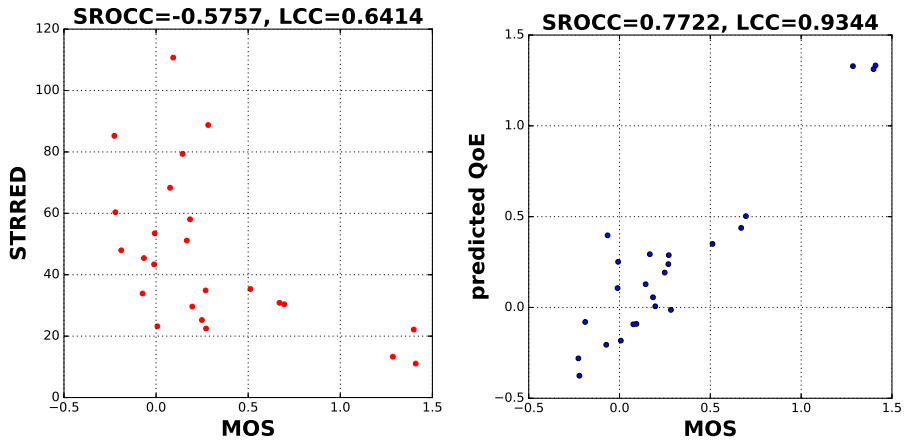

Figure 3.3: MOS against predicted QoE scores on a test subset when using the proposed model. Left: before regression (BR) when using only ST-RRED; Right: Proposed model. When using the regressed values the monotonicity may change sign (here it becomes increasing) and the scale of the vertical axis may also change. The figure shows a single train/test split where the 24 test points correspond to 8 distortion patterns and 3 contents per distortion.

SROCC.

3.5.1.1 Experiment 1: Testing for Content Independence

As a first step, we visually demonstrate the performance of the learned QoE model (see Fig. 3.3). In this example, the use of QoE-aware features can better capture QoE both in terms of monotonicity and linearity, compared to a model that does not consider stalling events or memory.

Then, we quantitatively analyzed the prediction performance of Video ATLAS, in Experiments 1 and 2. For Experiment 1, we conducted $\binom{14}{11} = 364$ unique train/test combinations where, in each trial, there are 11 training and 3 testing contents. The SROCC and LCC calculations were repeated on each

of the trials yielding a distribution of SROCC and LCC values for all possible train/test content combinations. Taking the median value of this distribution of correlation scores yields a single number describing the performance level of each method.

To demonstrate the promise of Video ATLAS, we compared it with various types of QoE models, including VQA models [79, 94, 143, 159, 160, 164], QoS models such as FTW [59] and VsQM [124], variants of the SQI index proposed in [45], modes 0 and 3 from P.1201-3 [3] and the NARX model [167]. To get a single QoE score from NARX, we first calculated the continuous-time scores, then averaged them across time. We were not able to find a publicly available implementation for modes 0 and 3 of [3], but we did our best to reproduce the steps described in the standard. Table 3.2 shows the SROCC, LCC and root-mean-squared error (RMSE) results for every model.

We found that Video ATLAS outperformed all other QoE prediction models, and we found these differences to be statistically significant (see Appendix C.2). In Appendix C, we investigate the effects of using various VQA models and feature sets for Video ATLAS. As expected, QoS and VQA models did not perform as well as QoE models, since they do not jointly capture the effects of stalling and video quality changes on the perceived QoE. By contrast, SQI combines *perceptually-relevant* FR VQA models with stalling information for retrospective QoE prediction and greatly improved the tested VQA models, with the exception of ST-RRED. The NARX model lacked in performance compared to Video ATLAS, which may be due to the fact that

it was specifically designed for continuous-time QoE prediction.

It is interesting that P.NATS modes 0 and 3 did not perform as expected in this dataset, which may be attributed in part to differences in the subjective data they were designed on. The LIVE-NFLX dataset focuses on very low bitrates and encoding resolutions, and has only a small number of bitrate shifts per distorted video, while the audio quality is fixed across all distorted videos and does not contribute to QoE variations. It is possible that if the LIVE-NFLX dataset used a broader range of resolutions and bitrates, the performance of P.NATS modes 0 and 3 could be improved. In the next section, we report our experimental analysis from Experiment 2.

Table 3.2: Results on the LIVE-Netflix DB over 364 pre-generated 80% train and 20% test splits. The best result is denoted with bold.

Model	SROCC	LCC	RMSE
FTW [59]	0.34	0.30	1.30
VsQM [124]	0.32	0.24	1.31
PSNR	0.60	0.57	0.77
SSIM [159]	0.68	0.75	0.58
MS-SSIM [160]	0.68	0.73	0.60
NIQE [94]	0.21	0.42	0.77
VMAF [79]	0.61	0.75	0.50
GMSD [164]	0.65	0.70	0.64
ST-RRED [143]	0.68	0.75	0.56
PSNR+SQI [45]	0.55	0.60	0.79
SSIM+SQI [45]	0.75	0.81	0.47
MS-SSIM+SQI [45]	0.75	0.79	0.49
ST-RRED+SQI [45]	0.57	0.67	0.66
P.1203 mode 0 [3]	0.46	0.68	1.73
P.1203 mode 3 [3]	0.44	0.30	1.28
NARX [167]	0.79	0.87	0.28
Video ATLAS	**0.88**	**0.94**	**0.23**

3.5.1.2 Experiment 2: Testing for Pattern Independence

As already mentioned, we carried out Experiment 2 to investigate pattern independence when applying Video ATLAS, i.e. Experiment 2 excludes the same distortion (such as 2 stalling events) from being present in both the train and test sets. As before, we compared Video ATLAS to various VQA and QoE models, but this time we excluded the QoS and the P.1201-3 models, since they did not perform as well. We summarize our findings in Table 3.3, which shows that all models performed worse than Experiment 1 (see Table 3.2). While these performances are not directly comparable since they correspond to a different experimental setup, Video ATLAS performed very well in both experiments. Note that Experiment 2 includes only 8 train/test combinations (one per distortion pattern), hence investigating statistical significance was not feasible.

Table 3.3: Experiment 2: Results on the LIVE-Netflix DB using various VQA models, SQI, NARX and Video ATLAS.

Model	SROCC	LCC	RMSE
PSNR	0.57	0.59	0.57
SSIM	0.79	0.83	0.30
MS-SSIM	0.74	0.83	0.31
NIQE	0.43	0.41	0.64
VMAF	0.49	0.48	0.63
GMSD	0.57	0.62	0.51
STRRED	0.81	0.81	0.31
PSNR+SQI	0.56	0.60	0.57
SSIM+SQI	0.80	0.84	0.30
MS-SSIM+SQI	0.74	0.83	0.30
ST-RRED+SQI	0.80	0.78	0.38
NARX	0.83	0.72	0.61
Video ATLAS	0.83	0.87	0.28

3.5.2 Experiments on the Waterloo Video QoE Database

It is also important to study the performance of Video ATLAS under different scenarios represented in other, independent resources, such as the Waterloo DB. As in Experiment 1, we compared the predictive power of Video ATLAS with that of SQI [45], and with several VQA models. We found that FTW [59] and VsQM [124] performed poorly on this dataset and hence those results are excluded. When conducting direct comparisons, we used the quality prediction models that were used to define SQI in [45]: PSNR, SSIM, MS-SSIM and SSIMplus [120]. To ensure that SQI yielded its best results on this dataset, we used the parameters suggested in [45] (different for each quality model). Note that the NARX model is continuous-time and hence it cannot be tested on the Waterloo DB, since per-frame subjective ground truth is not available in this dataset. We also excluded the P.1201-3 models which have not been trained on sequences that are less than 60 sec. long.

Given the simple playout patterns, only the VQA+M+R_2 feature set could be used in Video ATLAS, i.e. features I and R_1 become inactive. Since the videos in the Waterloo DB do not suffer from dynamic rate changes, the M feature was modified to instead be the amount of time since a stalling event took place (M_{stall}). We carried out the following two experiments:

Experiment 3: We conducted 1000 trials of 80% train, 20% test splits on the Waterloo DB, by using pre-generated indices of content-independent splits. The results are tabulated in Table 3.4. As before, we found that MS-SSIM and SSIMplus did not perform better than SSIM (though within

statistical uncertainty), even though both have been shown to yield better results than SSIM on the IQA and VQA problems [120, 160]. This verifies our earlier observation that the interplay between stalling events and quality changes complicates the predictive performance of traditional VQA models. Therefore, a better IQA/VQA model may not always correlate better with QoE measured in a lab setting. Overall, Video ATLAS delivered performance statistically indistinguishable to that of SQI. This is likely in part since the playout patterns in that dataset are simpler, the feature variation is smaller, and the number of input features was reduced to only three. Given that SQI was designed on the Waterloo DB, the Video ATLAS results are quite good.

Table 3.4: Experiment 3: Results on the Waterloo DB over 1000 pre-generated 80% train and 20% test splits.

Model	SROCC	LCC	RMSE
PSNR	0.66	0.65	20.96
SSIM [159]	0.82	0.85	14.81
MS-SSIM [160]	0.79	0.82	15.74
ST-RRED [143]	0.83	0.84	14.37
SSIMplus [120]	0.81	0.84	15.38
PSNR+SQI [45]	0.78	0.75	17.99
SSIM+SQI [45]	0.91	0.90	11.87
MSSIM+SQI [45]	0.89	0.88	12.85
SSIMplus+SQI [45]	0.89	0.89	11.60
Video ATLAS w/ M_{stall} but w/o I or R_1	0.90	0.90	11.45

Experiment 4: In our last experiment, we studied the performance of Video ATLAS on a database-independent scenario, by training on the Waterloo DB then testing on the LIVE-Netflix DB (see Table 3.5). In this case, we applied 10-fold cross validation on the entire Waterloo dataset to determine the best parameters of each model. For SQI, since we trained on the

Waterloo DB, we again used the suggested optimal parameters from [45]. As in Experiment 3, we excluded the NARX and P.1201-3 models, which are not applicable when the Waterloo database is used either for training or testing. Notably, we found that when testing on a different database than the testing one, normalizing the features by the video duration had a positive effect on the performance of Video ATLAS.

Video ATLAS demonstrated exceptional predictive performance and outperformed SQI in terms of both SROCC and LCC. While Video ATLAS performed better, it should be noted that it used only 3 of the 5 input features (VQA+M_{stall}+R_2), given the simple design of the Waterloo DB. A more general dataset for training could potentially increase the predictive performance of Video ATLAS even further. The simplicity of Video ATLAS is highly desirable: it uses features that capture the three main properties of QoE (video quality, stalling and memory); hence the QoE predictions are more explainable and less likely to overfit on unseen test data.

Table 3.5: Experiment 4: Training on the Waterloo DB and testing on the LIVE-Netflix DB. The best result is denoted with bold.

Model	SROCC	LCC
FTW [59]	0.34	0.29
VsQM [124]	0.32	0.24
PSNR	0.53	0.51
SSIM [159]	0.72	0.75
MS-SSIM [160]	0.70	0.73
STRRED [143]	0.73	0.74
PSNR+SQI [45]	0.60	0.60
SSIM+SQI [45]	0.78	0.74
MS-SSIM+SQI [45]	0.75	0.71
Video ATLAS w/ M_{stall} but w/o I or R_1	**0.84**	**0.80**

3.6 Conclusions

We described a feature-based approach for QoE prediction that integrates video quality models, stalling-aware, and memory features into a single QoE prediction model. This approach embodies our first attempt to develop an integrated retrospective QoE model, where stalling events and quality changes are considered in a unified way. Nevertheless, there is still room for improvement in two main directions: continuous-time models, such as the recently proposed ones in [25] and better subjective data that will fuel the design of better QoE predictors.

While the LIVE-NFLX database makes an effort to reflect realistic network and playout patterns, it does not include sufficient diversity in terms of quality and spatial resolution switching, stalling events, bitrate ranges and uses a single, simplistic network condition. Larger and more diverse subjective databases that include wider ranges of bitrate levels, that use actual network traces, and that include large numbers of bitrate and resolution switches are worthy goals. Such databases could be used to design and train even better QoE predictors by exploiting improved feature sets, with increased relevance to realistic HAS-QoE prediction scenarios. In the next Chapters, we will discuss our ongoing research efforts towards both developing continuous-time models and collecting richer subjective data.

Chapter 4

Continuous-Time Quality of Experience Prediction

4.1 Introduction

In this Chapter[1], we present a family of *continuous-time* streaming video QoE prediction models that process inputs derived from perceptual video quality algorithms, rebuffering-aware video measurements and memory-related temporal data. Our major contribution is to re-cast the continuous-time QoE prediction problem as a time-series forecasting problem. An implementation of this work can be found at `https://github.com/christosbampis/NARX_QoE_release`.

In the time-series literature, a wide variety of tools have been devised ranging from linear autoregressive-moving-average (ARMA) models [35, 92] to non-linear approaches, including artificial neural networks (ANNs). ARMA models are easier to analyze; however they are based on stationarity assumptions. However, subjective QoE is decidedly non-stationary and is affected by

[1]This chapter appears in the paper: C. G. Bampis, Z. Li, I. Katsavounidis and A. C. Bovik, "Recurrent and Dynamic Models for Predicting Streaming Video Quality of Experience", IEEE Transactions on Image Processing, vol. 27, no. 7, pp. 3316-3331, 2018. Christos George Bampis has designed and implemented the objective prediction models and carried out the experimental analysis of this paper.

dynamic QoE-related inputs, such as sudden quality changes or playback interruptions. This suggests that non-stationary models implemented as ANNs are more suitable for performing QoE predictions.

We specifically focus on the most practical and pressing problem: predicting *continuous-time* QoE by developing QoE system models driven by a mixture of quality, rebuffering and memory inputs to ANN-based dynamic models. Building on preliminary work in [23, 25], we advance progress towards this goal by devising efficient QoE prediction engines employing dynamic neural networks including recurrent neural networks, NARX [23, 25] and Hammerstein Wiener models [38, 52]. We thoroughly test these models on a set of challenging new subjective QoE datasets, and we conduct an in-depth experimental analysis of model and variable selection. We also study a variety of new ways of aggregating the time-series responses produced in parallel by different QoE models and initializations into a single robust continuous-time QoE estimate, and we provide demonstrations and guidance on the advantages and shortcomings of evaluation metrics that might be used to assess continuous time QoE prediction performance. We also compare the abilities of our proposed models against upper bounds on performance, i.e, human predictions.

The rest of this Chapter is organized as follows. Section 6.2 studies previous work on video quality assessment and QoE, while Section 4.3 discusses the design of our general QoE predictor. Next, Section 4.4 describes the proposed predictor that we have deployed and experimented with, and the complementary continuous-time inputs that feed it. In Section 4.5 we introduce

the forecasting ensemble approaches that are used to augment performance, and in Section 4.6 a general class of QoE predictors that we designed are described. Section 4.7 explains the experimental setup and Section 4.8 describes and analyzes our experimental results. Section 4.9 concludes this Chapter.

4.2 Related Work

Ultimately, video QoE research aims to create QoE prediction models that can efficiently address the resource allocation problem while ensuring the visual satisfaction of users. As already discussed, QoE prediction models can be conveniently divided into *retrospective* and *continuous-time* QoE predictors. In previous Chapters, we discussed retrospective QoE prediction models such as [45, 141] and proposed the Video ATLAS predictor. Here we focus on the continuous-time QoE prediction problem for HTTP-based adaptive streaming.

Similar efforts have been recently initiated as part of the P.NATS Phase 2 project [4], a joint collaboration between VQEG and ITU Study Group 12 Question 14, which includes numerous industry and university proponents. These research efforts have the same broad goal as our work, which is to design objective QoE prediction models for HTTP-based adaptive streaming [5, 117]. The P.NATS models combine information descriptive of rebuffering and video quality as determined by bitstream or pixel-based measurements. These approaches operate on a temporal block basis (e.g. on GOPs). Our work has two fundamental differences. First, we deploy continuous-time predictors that measure QoE with finer, per-frame granularity and these QoE responses can

be further aggregated over any desired time interval when designing adaptive rate allocation strategies. Furthermore, we train neural network models that exploit long-term memory properties of subjective QoE, which is a distinctive feature of our work.

Continuous-time QoE prediction using perceptual VQA models has received much less attention and is a more challenging problem. In [38], a Hammerstein-Wiener dynamic model was used to make continuous-time QoE predictions on videos afflicted only by dynamic rate changes. In [23], it was shown that combining video quality scores from several VQA models as inputs to a non-linear autoregressive model, or simply averaging the individual forecasts derived from each can deliver improved results. In [167], a simple model called DQS was developed using cosine functions of rebuffering-aware inputs, which was later improved using a learned Hammerstein-Wiener system in [52]. The system only processed rebuffering-related inputs, using a simple model selection strategy. Furthermore, only the final values of the predicted time-series were used to assess performance. As we will explain later, time-series evaluation metrics need to take into account the temporal structure of the data. To the best of our knowledge, the only approach to date that combines perceptual VQA model responses with rebuffering measurements is described in [25], where a simple non-linear autoregressive with exogenous variables (NARX) model was deployed to predict continuous QoE.

A limitation of previous QoE prediction studies has been that experimental analysis was carried out only on a single dynamic model and on a

single subjective database. Since predictive models designed or learned and tested on a specific dataset run the risk of inadvertent "tailoring" or over-training, deploying more general frameworks and evaluating them on a variety of different datasets is a difficult, but much more valuable proposition. We also believe that insufficient attention has been directed towards how to properly apply evaluation metrics to time-series QoE prediction models. Optimal model parameters can significantly vary across different test videos; hence carefully designed cross-validation strategies for model selection are advisable. In addition, it is possible to better generalize and improve QoE prediction performance by using forecast ensembles that filter out spurious forecasts. Finally, previous studies of continuous QoE have not investigated the limits of QoE prediction performance against human performance; calculating the upper bounds of QoE model execution is an exciting and deep question for QoE researchers.

To sum up, previous research studies on the QoE problem have suffered from at least one, and usually several, of the following limitations:

1. including either quality or rebuffering aware inputs
2. relying on a single type of dynamic model
3. limited justification of model selection
4. using evaluation metrics poorly suited for time-series comparisons
5. limited evaluation on a single video QoE database

6. do not exploit time-series ensemble forecasts

7. do not consider *continuous-time* human performance

Our goal here is to surmount 1-7 and to further advance efforts to create efficient, accurate and real-time QoE prediction models that can be readily deployed to perceptually optimize streaming video network parameters.

4.3 Designing General Continuous-Time QoE Predictors

In our search for a general and accurate continuous-time QoE predictor, we realized that subjective QoE is affected by the following:

1. Visual quality: low video quality (e.g. at low bitrates) or bandwidth-induced fluctuations in quality [27, 150] may cause annoying visual artifacts [100, 131] thereby affecting QoE.

2. Playback interruption: frequent or long rebuffering events adversely affect subjective QoE [59, 153]. Compared to degradations on visual quality, rebuffering events have remarkably different effects on subjective QoE [27, 50].

3. Memory (or hysteresis) effects: Recency [27, 56, 150] is a phenomenon whereby current QoE is more affected by recent events. Primacy occurs when QoE events that happen early in a viewing session are retained in memory, thereby also affecting the current sense of QoE [55].

Broadly, subjective QoE "is a non-linear aggregate of video quality, rebuffering information and memory" [25, 50, 149, 150]. The learning-driven Video ATLAS model [24], that we proposed, combines these different sources of information to predict QoE in general streaming environments where rebuffering events and video quality changes are commingled. Nevertheless, that model is only able to deliver overall (end) QoE scores. Towards solving the more difficult continuous-time QoE prediction problem, the following points should be considered:

1. At least three types of "QoE-aware inputs" must be fused: VQA model responses, rebuffering measurements and memory effects.

2. These inputs should have high descriptive power. For example, high-performance, perceptually-motivated VQA models should be preferred over less accurate indicators such as QP values [141] or PSNR. QoE-rich information can reduce the number of necessary inputs and boost the general capabilities of the QoE predictor.

3. Dynamic models with memory are able to capture recency (or memory) which is an inherent property of QoE.

4. These dynamic models should have an adaptive structure allowing for variable numbers of inputs. For example, applications where videos are afflicted by rebuffering events are not always relevant.

5. Multiple forecasts may be combined to obtain robust forecasts when monitoring QoE in difficult, dynamically changing real-world video streaming environments.

An outcome of our work is a promising tool we call the General NARX (GN) QoE predictor. Table 4.1 summarizes the notation that we will be using throughout the Chapter. In the following sections, we motivate and explain the unique features of this new method.

Table 4.1: Description of the acronyms and variables used throughout the Chapter.

Acronym	Description	Acronym	Description
VQA	video quality assessment	r	# training data splits for cross-validation
QoE	quality of experience	N_T	# training QoE time-series
QP	quantization parameter	S	# shuffles for performance bounds
NARX	non-linear autoregressive neural network	N_f	# frames for a given video
RNN	recurrent neural network	OL	open-loop configuration
HW	Hammerstein-Wiener	CL	closed-loop configuration
V-N/R/H	VQA-driven QoE with NARX/RNN/HW	ANN	artificial neural network
R-N/R/H	rebuffering-driven QoE with NARX/RNN/HW	FR	full-reference
G-N/R/H	general QoE-aware with NARX/RNN/HW	RR	reduced-reference
R_1	playback status indicator at time t	NR	no-reference
R_2	# rebuffering events until time t	RMSE	root-mean-squared error
M	time elapsed since last distortion (memory)	OR	outage ratio
D_1	LIVE HTTP Streaming Video Database [38]	DTW	dynamic time warping
D_2	LIVE Mobile Stall Video Database-II [52]	$\mathbf{D}$	pairwise DTW distance matrix
D_3	LIVE-Netflix Video QoE Database [27]	CI	confidence interval
d_u	# external variable lags	SROCC	Spearman's rank order correlation coefficient
d_y	# input lags	PLCC	Pearson's linear correlation coefficient
H	# hidden nodes	LD	number of layer delays in RNN
N	# videos in a subjective QoE database	α	significance level for hypothesis testing
T	# training initializations	m	# comparisons in Bonferroni correction

4.4 The GN-QoE Predictor

Our proposed GN-QoE prediction model is characterized by two main properties: the number and type of continuous-time features used as input and the prediction engine that it relies on. In this section, we discuss in greater

detail the QoE-aware inputs of our system and the neural network engine that we have deployed for continuous-time QoE prediction.

4.4.1 QoE-Aware Inputs

The proposed GN-QoE Predictor relies on a non-linear dynamic approach which integrates the following *continuous-time QoE-aware* inputs:

1. The high-performing ST-RRED metric as the VQA model. Previous studies [13, 24, 25, 27], have shown that ST-RRED is an exceptionally robust predictor of subjective video quality. As was done in [23], it is straightforward to augment the GN-QoE Predictor by introducing additional QoE-aware inputs, if they verifiably contribute QoE prediction power. For example, the MOAVI key indicators [74] of bluriness or blur loss distortion could be applied in order to complement the current VQA input. At the same time, we recognize that simple and efficient models are desireable in practical settings, especially ones that can be adapted to different types of available video side-information.

Quality switching [119, 150] also has a distinct effect on subjective video QoE. While we do not explicitly model quality switching, the memory component of the NARX engine allows it to exploit ST-RRED values over longer periods of time as a proxy for video segments having different qualities.

2. We define a boolean continuous-time variable R_1 which describes the playback status at time t which takes value $R_1 = 1$ during a rebuffering event and $R_1 = 0$ at all other times. This input captures playback-related information.

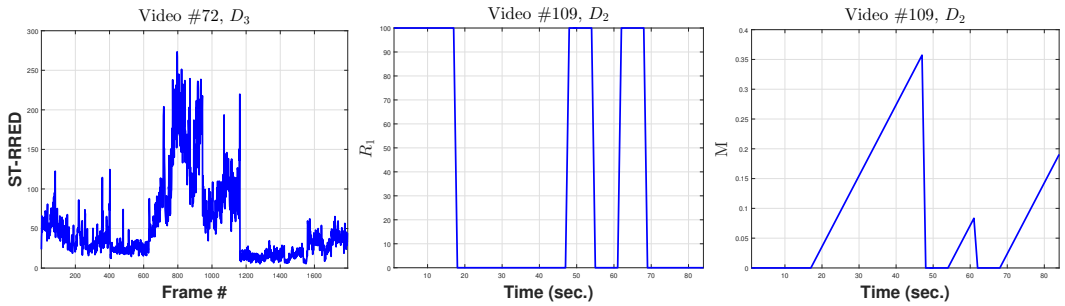

Figure 4.1: Examples of the proposed continuous time QoE variables. Left to right: ST-RRED computed on video #72 of the LIVE-NFLX Video QoE Database (D_3), and R_1 and M on the LIVE Mobile Stall Video Database-II (D_2).

We also define the integer measure R_2 to be the number of rebuffering events that have occurred until time t.

3. M: the time elapsed since the latest network-induced impairment such as a rebuffering event or a bitrate change occurred. M is normalized to (divided by) the overall video duration. This input targets recency/memory effects on QoE.

Figure 4.1 shows a few examples of these continuous-time inputs measured on videos from various subjective databases.

4.4.2 NARX Component

The GN-QoE Predictor relies on the non-linear autoregressive with exogenous variables (NARX) model [25, 84, 140]. The NARX model explicitly produces an output y_t that is the result of a non-linear operation on multiple

past inputs $(y_{t-1}, y_{t-2}, \dots)$ and external variables $(\mathbf{u}_t)$:

$$y_t = f(y_{t-1}, y_{t-2}, ..., y_{t-d_y}, \mathbf{u}_t, \mathbf{u}_{t-1}, \mathbf{u}_{t-2}, ..., \mathbf{u}_{t-d_u}) \tag{4.1}$$

where $f(\cdot)$ is a non-linear function of previous inputs $\{y_{t-1}, y_{t-2}, ..., y_{t-d_y}\}$, and previous (and current) external variables $\{\mathbf{u}_t, \mathbf{u}_{t-1}, \mathbf{u}_{t-2}, ..., \mathbf{u}_{t-d_u}\}$, where d_y is the number of lags in the input and d_u is the number of lags in the external variables. To capture the recency effects of subjective QoE, the memory lags d_y and d_u need to be large enough. In practice, we determine these parameters using cross-validation (see Section 4.7.2). In Section E.4 we show that GN-QoE is able to capture recency effects when predicting QoE.

In a NARX model, there are two types of inputs: past outputs that are fed back as future inputs to the dynamic model, and external (or "exogenous") variables (see Fig. 4.1). The former are scalar past outputs of the NARX model, while the latter are past and current values of QoE-related information, e.g. the video quality model responses, and can be vector valued. To illustrate this, Fig. 4.3 shows an example of the NARX architecture: there are three exogenous inputs $\mathbf{u}(t)$, each containing a zero lag component and five past values. By contrast, past outputs cannot contain the zero lag component.

The function $f(\cdot)$ is often approximated by a feed-forward multi-layer neural network [101] possibly having variable number of nodes per hidden layer. Here we focus on single-hidden layer architectures having H hidden nodes. There are two approaches to training a NARX model. The first approach is to train the NARX without the feedback loop, also known as an open-loop

(OL) configuration, by using the ground truth values of $\mathbf{y}_t$ when computing the right hand side of (4.1). An example of the ground truth scores is shown in Fig. 4.2. The second approach uses previous estimates of y_t, also known as a closed-loop (CL) configuration [23]. Both approaches can be used while training; however, application of the NARX must be carried out in CL mode, since ground truth subjective data is not available to define a new time-series. The advantages of the OL approach are two-fold: the actual subjective scores are used when training, and the neural network to be trained is feed-forward; hence static backpropagation can be used [16].

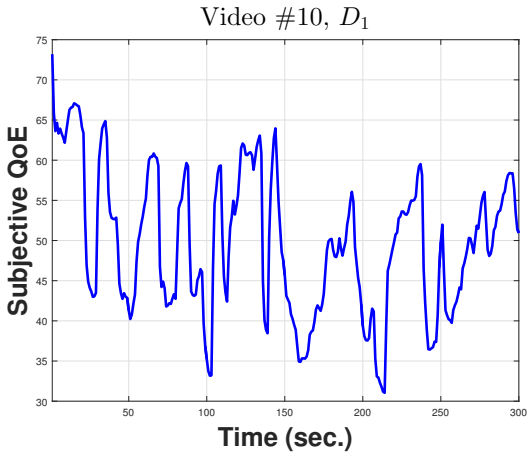

Figure 4.2: Exemplar subjective QoE scores on video #10 from the LIVE HTTP Streaming Video Database (denoted by D_1).

It has been shown [23] that, in practice, the CL configuration requires longer training times and yields worse predictive performance; hence we use the OL configuration when training and the CL configuration only when testing. An example of the CL configuration of the NARX model is shown in Fig.

4.3. For simplicity, we used a tangent sigmoid activation function and a linear function in the output layer. The role of the linear function is to scale the outputs in the range of the subjective scores, while the sigmoid activation function combines past inputs and external variables in a non-linear fashion. Given that the problem is of medium size, we chose the Levenberg-Marquardt [76, 90] algorithm to train the model [15]. To reduce the chances of overfitting in the OL training step, we used an early stopping approach [17]: the first 80% of the samples were used to train the OL NARX, while the remaining 20% were used to validate it. In Appendix D, we discuss these implementation details of the NARX predictor, including the choice of the training algorithm, the activation function and data imputation strategies.

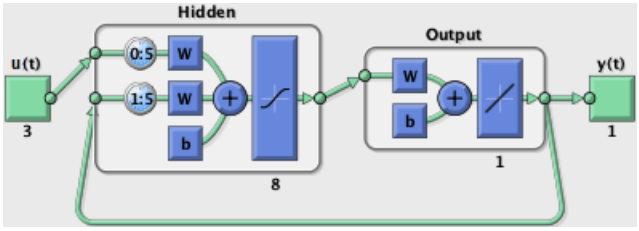

Figure 4.3: The dynamic CL NARX system with 3 inputs, 8 neurons in the hidden layer and 5 feedback delays. The recurrency of the NARX occurs in the output layer [16].

The GN-QoE Predictor follows a learning-driven approach which requires careful cross-validation and design. Still, preliminary experiments led us to the conclusion that a single time-series prediction may be insufficient for the challenging problem of continuous-time QoE prediction. Next, we describe another unique feature of the GN-QoE Predictor: the use of forecasting ensembles.

4.5 Forecasting Ensembles

4.5.1 Motivation

Ensemble learning is a long-standing concept that has been widely applied in such diverse research fields as forecasting [75, 171] and neural network ensembles [70, 173]. We are specifically interested in time-series forecasting ensembles, where two or more continuous QoE predictions are aggregated. In our application, we utilize a variety of dynamic approaches that have various parameters, such as the number of input delays. The results of these models may also depend on the neural network initialization. Generally, relying on a single model may lead to drawbacks such as:

1. Uncertain model selection. For example, in the stationary time-series and ARMA literature [35, 92], model order selection typically relies on measurements of sample autocorrelations or on the Akaike Information Criterion. However, in neural network approaches, this problem is not as well-defined.

2. Using cross-validation for model selection may not always be the best choice. Different choices of the evaluation metric against which the QoE predictor is optimized may yield different results. Furthermore, an optimal model for a particular data split may not be suitable for a different test set. While much larger QoE databases could contribute towards ameliorating this issue, the barriers to creating these are quite formidable, suggesting multi-modal approaches as an alternative way to devise effective and practical solutions.

3. The QoE dynamics within a given test video may vary widely, reducing the effectiveness of a single model order.

Since a single time-series predictor might yield subpar prediction results, we have developed ensemble prediction models that deliver more robust prediction performance by deemphasizing unreliable forecasts. These ensemble techniques were applied to each of the forecasts generated. For example, testing GN-QoE using κ different combinations of model orders d_u and d_y, λ different neural network initializations and μ possible values for the neurons in the hidden layer, produces $\kappa\lambda\mu$ forecasts which are then combined together yielding a single forecast. In the next section, we discuss these ensemble methods in greater detail.

4.5.2 Proposed Ensemble Methods

We have developed two methods of combining different QoE predictors. The first determines the best performer from a set of candidate solutions. We relied on the dynamic time warping (DTW) distance [32] which measures the similarity between two time-series that have been time-warped to optimally match structure over time: a larger DTW distance between two time-series signifies they are not very similar. The benefit of DTW is that it accounts for the temporal structure of each time-series and that it makes it possible to compare signals that are similar but for rebuffering-induced delays. We computed pairwise DTW distances between all predictors, thereby producing a symmetric matrix of distances $\mathbf{D} = [d_{ij}]$, where $d_{ij} = d_{ji}$ is the DTW distance

between the ith and jth time-series predictions. Similar to the subject rejection method proposed in [27], we hypothesize that $\nu_i = \sum_j \mathbf{D}_{ij}$, i.e., the sum across rows (or columns) of $\mathbf{D}$ is an effective measure of the reliability of the ith predictor. A natural choice is

$$i_o = \arg\min_i \nu_i, \tag{4.2}$$

where i_o denotes the single best predictor. Note that i_o may not necessarily coincide with the time-series prediction resulting from the best model parameters (as derived in the cross-validation step). The second approach is to assign a probabilistic weight to each of the C candidate predictors:

$$\tilde{y}_t = \sum_{c=1}^{C} w_c \hat{y}_{ct}, \;\; w_c = \frac{1/\nu_c}{\sum_c 1/\nu_c}, \tag{4.3}$$

where $w_c \in [0, 1]$ determines (weights) the contribution of the cth predictor to the ensemble estimate $\tilde{y}_t$. Along with these two ensemble methods, we also evaluated several other commonly used ensemble methods, including mean, median and mode ensembles. Mean ensembles have proven useful in many forecasting applications [147], while median and mode ensembles are more robust against outliers [69].

4.6 The G- Family of QoE Predictors

The GN-QoE Predictor is versatile and can exploit other VQA inputs than the high performance ST-RRED model [13]. Indeed, it allows the use of any VQA model (FR, RR or NR), depending on the available reference

information. As in [24, 25], this enables the deployment of these models in a wide range of QoE predictions applications.

Taking this a step forward, we have developed a wider family of predictors based on the ST-RRED, R_1 and M inputs, that also deploy other dynamic model approaches. For example, Layer-Recurrent Neural Networks (denoted here as RNNs)[47] or the Hammerstein-Wiener (HW) dynamic model [38, 52] can be used instead of NARX, yielding models called GR-QoE and GH-QoE, respectively. This general formulation also allows us to consider model subsets that relate and generalize previous work. For example, the GH-QoE model, when using only ST-RRED as input (denoted by VH in Table 4.2) may be considered as a special case of [38]. We summarize the proposed family of G-predictors and other predictors that use subset of these inputs, and their characteristics in Table 4.2. Since the same QoE features are shared across GN-, GR- and GH-QoE, we next discuss the learning models underlying GR-QoE and GH-QoE.

4.6.1 GR-QoE Models

Recurrent Neural Networks (RNNs) [47] have recently gained popularity due to their successful applications to various tasks such as handwriting recognition [54] and speech recognition [128]. The main difference between the NARX and RNN architectures, is that while the former uses a feedback connection from the output to the input, RNNs are feedforward neural networks that have recurrent connections in the hidden layer. Therefore, the structure

Table 4.2: Summary of the various compared QoE predictors. X denotes that the predictor in the row possesses the property described in the column. We have found that including R_2 in the G-predictors produces no additional benefit (see E).

QoE Predictor	Learner	VQA	R_1	R_2	M	ensemble
VN	NARX	X				X
RN	NARX		X	X		X
RMN	NARX		X	X	X	X
GN	NARX	X	X		X	X
VR	RNN	X				X
RR	RNN		X	X		X
RMR	RNN		X	X	X	X
GR	RNN	X	X		X	X
VH	HW	X				X
RH	HW		X	X		X
RMH	HW		X	X	X	X
GH	HW	X	X		X	X

of an RNN allows it to dynamically respond to time-series input data. The recurrency property of RNNs allows them to model the recency properties of subjective QoE. An example of such a neural network is shown in Fig. 4.4.

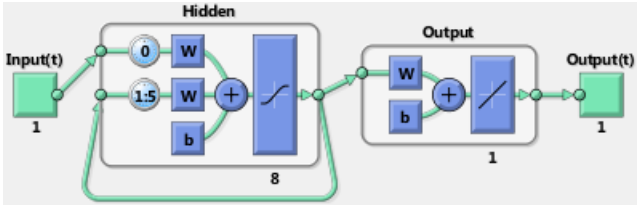

Figure 4.4: The dynamic RNN approach with 1 input, 8 neurons in the hidden layer and 5 layer delays: the recurrency occurs in the hidden layer rather than in the output layer [16].

Given that the amount of available subjective data is insufficient to train a deep neural network, we decided to train relatively simple RNN models, i.e., neural networks having only one hidden layer and up to 5 layer delays. As in NARX, we used a tangent sigmoid activation function and a linear function

at the output layer.

4.6.2 GH-QoE Models

Unlike the NARX and RNN models, the HW model, which is block-based (see Fig. 4.5), has only been deployed for QoE prediction on videos afflicted by rate drops [38] or rebuffering events [52]. The HW structure is relatively simple: a dynamic linear block having a transfer function with n_f poles and n_b zeros, preceded and followed by two non-linearities. The poles and zeros in the transfer function allow the HW model to capture the recency effects in subjective QoE, while the non-linear blocks account for the non-linear relationship between the input features and QoE.

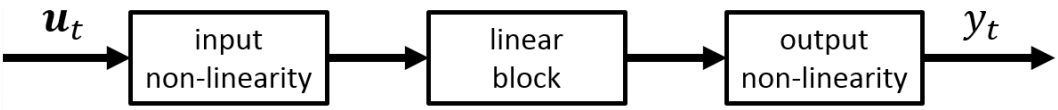

Figure 4.5: The HW dynamic approach.

The family of G-QoE predictors (see Table 4.2) can be applied to any subjective database containing videos afflicted by quality changes, rebuffering events or both, by simply choosing the model (QoE feature) subset that is applicable to each case. Following our G-notation, we also define predictors V- (which use only VQA model responses), R- (only rebuffering features) and RM- (rebuffering and memory). We next describe the various subjective datasets we used to evaluate the various approaches.

4.7 Subjective Data and Experimental Setup

We now discuss the experimental aspects behind our QoE prediction systems. We first describe the three different subjective QoE databases that we used and our parameter selection strategy. Next, we discuss the advantages and caveats of various continuous-time performance metrics and their differences. We conclude this section with a discussion on performance bounds of continuous-time QoE predictors.

4.7.1 Subjective Video QoE Databases

In [38], a subjective video QoE database (denoted by D_1 for brevity) was created containing 15 long video sequences afflicted by quality fluctuations relevant to HTTP rate-adaptive video streaming. This database consists of 8 different video contents of 720p spatial resolution encoded at various H.264 bitrate levels, with associated time-varying subjective scores. Rebuffering events were studied in [51] using a different database (denoted by D_2), where diverse rebuffering patterns were inserted into 24 different video contents of various spatial resolutions. Unlike [38], this subjective QoE database allows the study of rebuffering-related characteristics (such as the number, locations and durations of the rebuffering events) and their effects on time-varying and overall QoE. A total of 174 distorted videos are part of this database.

A deficiency of these early studies is that they were not driven by any bandwidth usage models and did not contain videos containing both rebuffering events and quality variations. In realistic streaming applications, dynamic

rate adaptations and rebuffering events occur, often in temporal proximity depending on the client device's resource allocation strategy [30, 61, 82]. As already described, we built the new LIVE-NFLX Video QoE Database [27] (D_3) to bridge this gap.

We used these three subjective databases to extensively study the performance of the continuous-time GN-, GR- and GH-QoE predictors. Next, we describe the cross-validation strategy that we used to determine the best parameter setting for each of these prediction engines.

4.7.2 Cross-validation Framework for Parameter Selection

We now introduce our cross-validation scheme for continuous-time QoE prediction. Notably, the proposed recurrent models are highly non-linear; hence the traditional time-series model estimation techniques used in ARMA models [35] are not possible. Further, subjective QoE prediction is highly non-stationary; therefore the most suitable model order may vary within a given QoE time-series or across different test time-series. As a result, determining the best model parameters, e.g., the input and feedback delays in the GN-QoE model (d_u and d_y), the number of poles (n_f) and zeros (n_b) in the transfer function of a GH model, or the number of layer delays (LD) in a GR model, must be carefully validated (see Table 4.3).

Here we propose a novel cross-validation framework that is suitable for *streaming video* QoE predictors. This idea builds on a simpler approach that was introduced in [23]. In data-driven quality assessment applications, the

available data is first split into content-independent training and testing subsets, then the training data is further split into smaller "validation" subsets for determining the best parameters. Content independence ensures that subjective biases towards different contents is alleviated when training and testing. In the case of data-driven continuous-time QoE predictors, it is more realistic to split the data in terms of their distortion patterns, since the testing network conditions (which have a direct effect on the playout patterns) are not known *a priori*.

The non-deterministic nature of these time-series predictions adds another layer of complexity. As an example, given a set of QoE time-series used for training, we have found that different initial weights produce different results for GN- and GR-QoE Predictors; hence their performance should be estimated across initializations. By comparison, previous continuous-time QoE prediction models [25, 38, 52] have used a single model order. To sum up, training a successful continuous-time QoE predictor requires:

1. Determining the best set of parameters using cross-validation on the available continuous-time subjective data.

2. Ensuring content-independent train and test splits.

3. Distorted videos corresponding to the same network or playout pattern should belong only in the train or the test set.

4. To account for different neural network initializations, multiple iterations need to be performed on per training set.

Based on these properties, we now discuss our cross-validation strategy in detail. Let $i = 1 \ldots N$ index the video in a database containing N videos. First, randomly select the ith video as the test time-series. To avoid content and other learning biases, remove from the training set all videos having similar properties as the test video, such as segments that belong to the same video content. Depending on which subjective database is used, we applied the following steps. For D_3, we removed all videos having either the same content or the same distortion pattern [25]. For D_1 and D_2, we removed all videos having the same content. This process yielded a set of N_T training QoE time-series for each test video, where $N_T = 10$, 129 and 91 for D_1, D_2 and D_3 respectively.

Next, we divided the training set further into a training subset and a validation subset. This step was repeated r times to ensure sufficient coverage of the data splitting. We also found that the HW component of the GH-QoE model was sensitive to the order of the training data in a given training set. To account for this variation, we also randomized the order of the time-series in this second training set. Then, we evaluated each model configuration on every validation set in terms of root-mean-square error (RMSE), and averaged the RMSE scores, yielding a single number per model configuration. The model parameters that yielded the minimum RMSE were selected to be the ones used during the testing stage. When testing, we used all of the training data and the optimized model parameters that were selected in the cross-validation step. To account for different weight initializations, we repeated the training

process T times; then averaged the performances across initializations.

Table 4.3: Parameters used in our experiments. On all three databases we fixed $r = 3$ and $T = 5$. K can be any of the following three: G, V or RM depending on the subjective database that the predictors were applied.

Model	KN			KR		KH		
parameter	d_u	d_y	H	LD	H	n_b	n_f	H
D_1	[10,12,14]	[10,12,14]	[5,8]	[3,4,5]	[5,8]	[10,12,14]	[10,12,14]	10
D_2	[4,5,6]	[4,5,6]	[5,8]	[3,4,5]	[5,8]	4	4	10
D_3	[8,10,15]	[8,10,15]	[5,8]	[3,4,5]	[5,8]	[8,10,15]	[8,10,15]	10

During cross-validation, we used the RMSE evaluation metric to select the best performing model configuration. Nevertheless, other evaluation metrics may also contribute important information when comparing continuous-time QoE prediction engines. In the following section, we investigate these metrics in greater detail.

4.7.3 Evaluation Metrics

After performing the time-series predictions, it is necessary to select suitable evaluation metrics to compare the output p with the ground truth time g. In traditional VQA, e.g, in [134] and in hybrid models of retrospective QoE [24, 45], the Spearman rank order correlation coefficient (SROCC) is used to measure monotonicity, while Pearson's Linear Correlation Coefficient (PLCC) is used to evaluate the linear accuracy between the ground truth subjective scores and the VQA/QoE predicted scores. These evaluation metrics have also been used in studies of continuous-time QoE prediction [23, 38, 52].

Yet, it is worth asking the question: "Is there a single evaluation met-

ric suitable for comparing subjective continuous-time QoE scores?" We have found that each evaluation metric has its own merits; hence they have to be considered collectively.

We now discuss the advantages and shortcomings of the various evaluation metrics that can be used to compare a ground truth QoE time-series $g = [g_i]$ and a predicted QoE waveform $w = [w_i]$ where i denotes the frame index. Continuous-time subjective QoE is inherently a dynamic system with memory; hence we have developed continuous-time autoregressive QoE models. However, SROCC and PLCC are only valid under the assumption that the samples from each set of measurements were independently drawn from within each set; whereas subjective QoE contains strong time dependencies and inherent non-stationarities.

There are other evaluation metrics that are more suitable for time-series comparisons, i.e.,

1. The root-mean-squared error (RMSE), which captures the overall signal fidelity: $\sqrt{(\sum_{i=1}^{N_f}(w_i - g_i)^2)/N_f}$, where N_f is the number of frames.

2. The outage rate (OR) [38], which measures the frequency of times when the prediction w_i falls outside twice the confidence interval of g_i:

$$\frac{1}{N_f}\sum_{i=1}^{N_f}\mathbb{1}(|w_i - g_i| > 2\mathrm{CI}_{g_i}), \tag{4.4}$$

where CI_{g_i} is the 95% confidence interval of the ground truth g at frame i across all subjects.

3. The dynamic time warping (DTW) distance can also be employed [25, 27, 32] to capture the temporal misalignment between w and g.

Each of these metrics has shortcomings:

1. The RMSE is able to capture the scale of the predicted output, but cannot account for the temporal structure.

2. The OR is intuitive and suitable for continuous-time QoE monitoring, but does not give information on how the predicted time-series behaves within the confidence bounds.

3. DTW captures temporal trends, but the DTW distance is hard to interpret, e.g., a smaller distance is always better but a specific value is hard to interpret.

We demonstrate these deficiencies in Figs. 4.6, 4.7 and 4.8. Figure 4.6 shows that the outage rate on the left is lower; however the predicted QoE is noisy. By contrast, while the predicted QoE on the right has a larger OR, it is more stable and it appears to track the subjective QoE more accurately. Figure 4.7 shows that, while the DTW distance between the two time-series predictions is very different, both predictions nicely capture the QoE trend. Lastly, while RMSE captures the correct QoE range, an artificially generated time-series containing a zero value performs better than the temporal prediction but misses all of the trends (see Fig. 4.8). Clearly, any single evaluation metric is likely to be insufficiently descriptive of performance; hence we report all three

of these metrics, along with the SROCC, to draw a clearer picture of relative performance.

4.7.4 Continuous-time Performance Bounds

While the previously discussed evaluation metrics can be used to compare QoE predictors, they do not yield an absolute ranking against the putative upper bound of human performance. As stated in [131]: "The performance of an objective model can be, and is expected to be, only as good as the performance of humans in evaluating the quality of a given video." We measured the "null" (human) level of performance as follows. We divided the subjective scores of each test video into two groups of the same size, one considered as the training set and the other as the test set. Let A_i and B_i be the two sets, i.e., A_i is the train set for the ith test video and B_i the corresponding test set. For a given evaluation metric, we averaged the subjective scores in A_i and B_i and compared them. To account for variations across different splits, this process was repeated S times per test video, yielding subsets A_{is} and B_{is} at each iteration s. We fixed $S = 10$. Then, we computed the median value over s, yielding the median prediction performance of the ith test video. Finally, to obtain a single performance measure on a given database, we calculated the median value over all test videos.

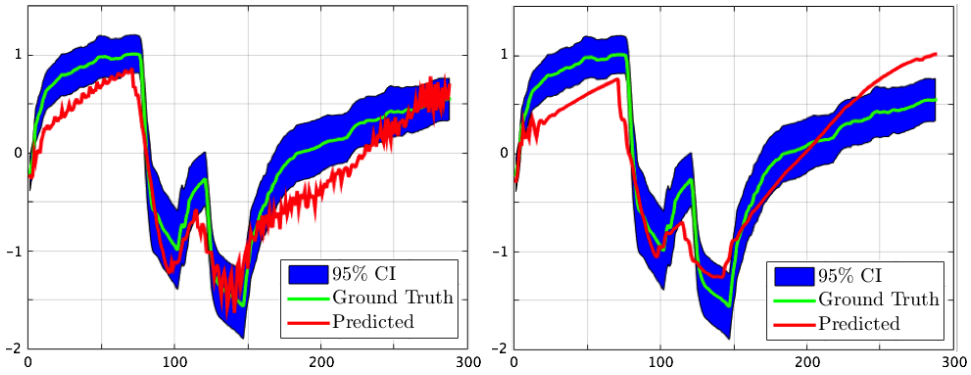

Figure 4.6: Vertical axis: QoE; horizontal axis: time (in samples). OR does not describe the prediction's behavior within the CI. Left: OR = 5.90; Right: OR = 13.19.

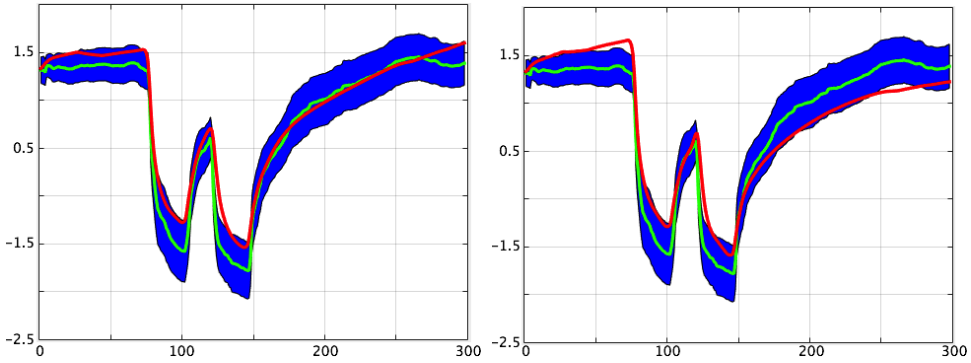

Figure 4.7: Vertical axis: QoE; horizontal axis: time (in samples). DTW better reflects the temporal trends of the prediction error although it is harder to interpret. Left: DTW = 2.96; Right: DTW = 19.56.

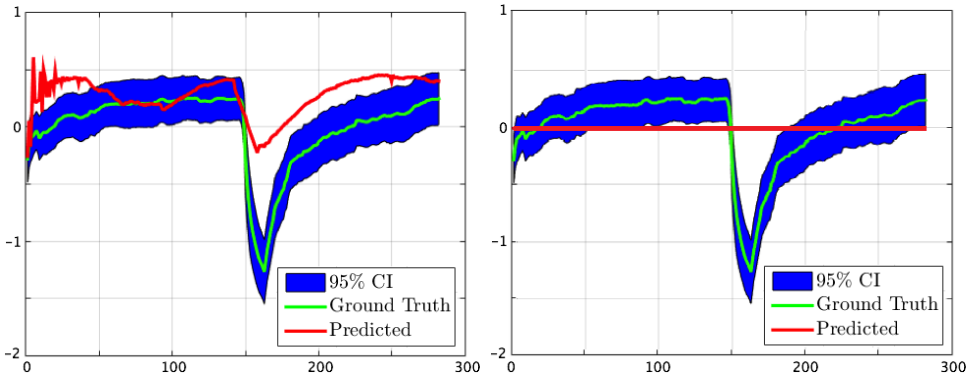

Figure 4.8: Vertical axis: QoE; horizontal axis: time (in samples). RMSE does not effectively account for the local temporal structure of the prediction error. Left: RMSE = 0.36; Right: RMSE = 0.33.

4.8 Experimental Results

In this section, we thoroughly evaluate and compare the different approaches in terms of their qualitative and quantitative performance. Recall that only database D_3 contains both quality changes and playback interruptions; hence we applied the V-predictors on D_1, the RM-predictors on D_2 and the G-predictors on D_3.

To examine statistical significance, we used the non-parametric Wilcoxon significance test [139] using a significance level of $\alpha = 0.05$. To account for multiple comparisons, we applied Bonferroni correction which adjusts α to $\frac{\alpha}{m}$, where m is the number of comparisons. In all of the reported statistical test results, a value of '1' indicates that the row is statistically better than the column, while a value of '0' indicates that the row is statistically worse than the column; a value of '-' indicates that the row and column are statistically equivalent.

4.8.1 Qualitative Experiments

We begin by visually evaluating the different models on a few videos from all three QoE databases. Figure 4.9 shows the performance of the VN-QoE Predictor on video #8 of database D_1; the continuous time predictions of the best cross-validated model closely follow the subjective QoE, and all individual models yielded similar outputs. In such cases, it may be that forecasting ensembles yield little benefit.

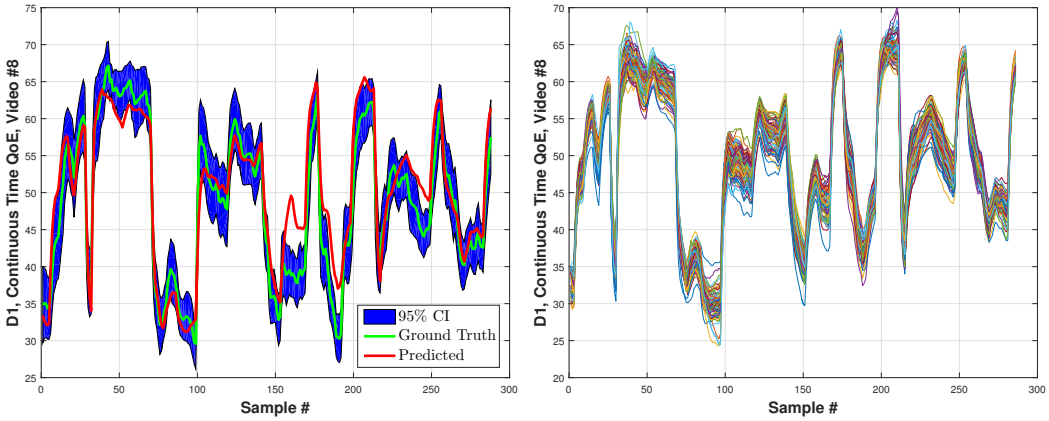

Figure 4.9: The VN-QoE Predictor on video #8 of database D_1. Top: prediction using the best cross-validated model; bottom: predictions from all the models.

By contrast, Fig. 4.10 shows QoE prediction on video #16 of database D_2. All three dynamic approaches suffered either from under- or over-shoot. The RMR-QoE Predictor produced some spurious forecasts. In this instance, an ensemble method could increase the prediction reliability, but, in this example, the RMH-QoE Predictor performed well.

The example in Fig. 4.11 proved challenging for both the GN- and GR-

QoE Predictors: the best cross-validated GN model was unable to capture the subjective QoE trend, while the GR model produced an output that did not capture the first part of the QoE drop. These examples highlight some of the challenges of the problem at hand: finding the best neural network model can be difficult. By contrast, the GH model was able to produce a much better result. Notably, all three dynamic approaches suffered from spurious forecasts, again suggesting that forecasting ensembles could be of great use.

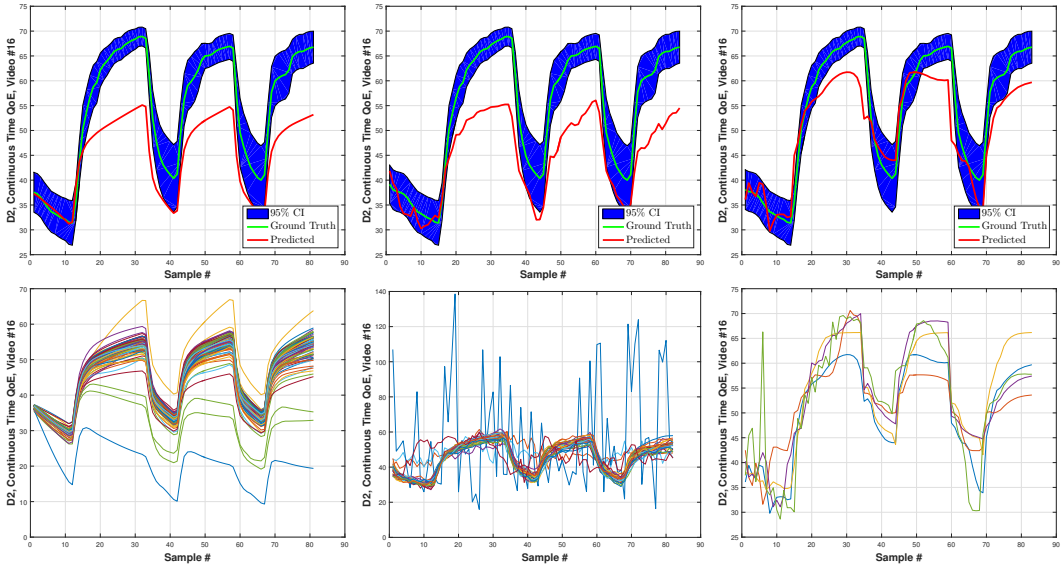

Figure 4.10: Columns 1 to 3: The RMN-, RMR- and RMH-QoE Predictors applied to video #16 of database D_2. First row: prediction using the best cross-validated model; second row: predictions from all models.

4.8.2 Quantitative Experiments - D_1

We begin our quantitative analysis by discussing the prediction performances of the compared QoE prediction models (class V-) on the LIVE

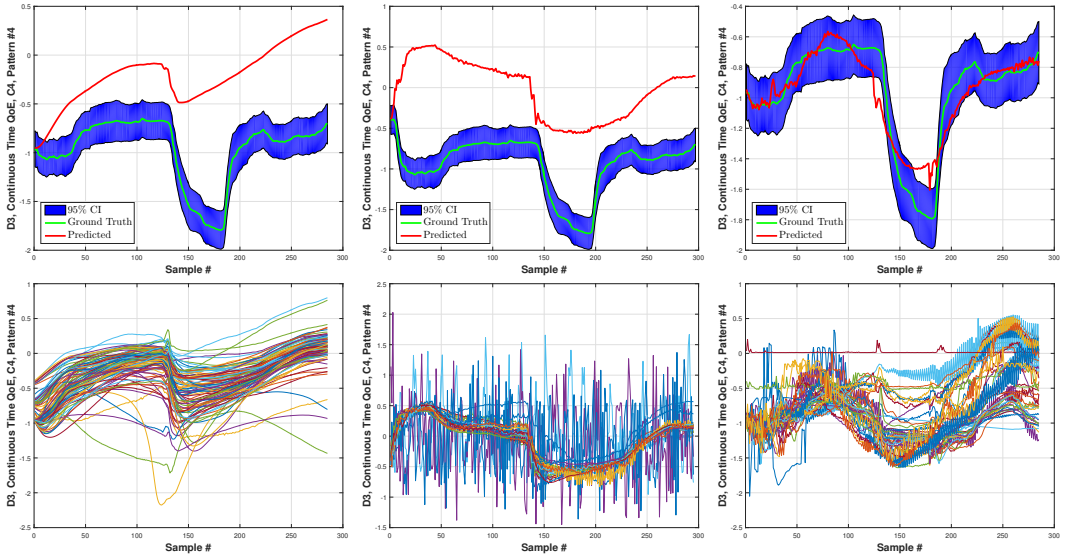

Figure 4.11: Columns 1 to 3: The GN-, GR- and GH-QoE Predictors applied on pattern #4 of database D_3. First row: prediction using the best cross-validated model; second row: predictions from all models.

HTTP Streaming Video Database (D_1). We first statistically compared the VN, VR and VH predictors in terms of OR when using ST-RRED (see Table 4.4). Among the three compared dynamic approaches, the VN-QoE Predictor consistently outperformed the VR and VH models. It has been previously demonstrated [84] that the NARX architecture is less sensitive than RNN models when learning long-term dependencies.

Table 4.4: OR significance testing ($m = 3$) on the class of V-predictors (without ensembles) on D_1 using ST-RRED.

Model Type	VN	VR	VH
VN	-	1	1
VR	0	-	-
VH	0	-	-

In D_1, there is no rebuffering in the distorted videos and hence it is straightforward to study the performance between various leading VQA models: PSNR, NIQE [94], VMAF (version 0.3.1) [79], MS-SSIM [160], SSIM [159] and ST-RRED [143] (see Table 4.5).

Table 4.5: Median OR performance for the class of V- QoE predictors on database D_1 (see also Table E.4).

Model Type	VN	VR	VH
NIQE [94]	34.79	42.84	42.78
PSNR	25.07	36.16	29.51
VMAF [79]	12.38	24.05	23.04
MS-SSIM [160]	5.73	17.64	31.82
SSIM [159]	5.46	17.43	30.69
ST-RRED [143]	5.90	20.81	15.31

Unsurprisingly, NIQE performed the worst across all dynamic approaches; after all, it is a no-reference frame-based video quality metric. PSNR delivered the second worst performance, but it does not capture any perceptual quality information. MS-SSIM, SSIM and ST-RRED all performed well when deployed in the VN-QoE Predictor; but when it was inserted into the HW model, ST-RRED delivered the best performance. As shown in Table 4.6, the OR performance differences between VMAF 0.3.1, MS-SSIM, SSIM and ST-RRED were not statistically significant for the VN model; but all three of them performed significantly better than PSNR and NIQE. It should be noted that these statistical comparisons were performed at a very strict confidence level of $\frac{\alpha}{m} = \frac{0.05}{15}$ (due to Bonferroni correction with $m = 15$), hence these comparisons are conservative.

Table 4.6: OR significance testing ($m = 15$) when the VN-QoE Predictor was applied on D_1 across various VQA models. Similar results were produced by the other evaluation metrics.

Model	NIQE	PSNR	VMAF	MS-SSIM	SSIM	ST-RRED
NIQE	-	0	0	0	0	0
PSNR	1	-	0	0	0	0
VMAF	1	1	-	-	-	-
MS-SSIM	1	1	-	-	-	-
SSIM	1	1	-	-	-	-
ST-RRED	1	1	-	-	-	-

Our results show that perceptual VQA models, when combined with dynamic models that learn to conduct continuous-time QoE prediction, do not perform equally well; hence deploying high performance VQA models can contribute to improved QoE prediction. Deciding upon the choice of the VQA feature is application-dependent; yet we believe injecting perceptual VQA models into these models is much more beneficial than using QP or bitrate information.

We now study the efficacy of ensemble forecasting approaches. The naming convention of the ensemble methods is as follows: "best": pick best (from cross-validation) model parameters when testing, "avg": averaging of all forecasts, "med": taking the median of all forecasts, "mod": estimating the mode, "DTW-single": determining i_o in (4.2), "DTW-prob": probabilistic weighting of forecasts in (4.3).

Table 4.7 shows that NARX again performed better than the other models across all ensemble methods. Using an ensemble method different than

Table 4.7: Median OR performance for various time-series ensemble methods applied on the class of V-predictors on database D_1 using ST-RRED (see also Table E.5).

Model Type	VN	VR	VH
best	5.90	20.81	15.31
avg	5.25	15.59	14.69
med	5.25	9.15	13.99
mod	4.55	8.48	13.99
DTW-single	5.59	8.81	15.04
DTW-prob	5.25	10.51	14.69

the mean yielded results similar to the mean. This suggests that the VN-QoE predictions were stable across different initializations and configurations (see also Fig. 4.9), given that more robust estimators such as the non-parametric mode produced results similar to the mean ensemble which can be sensitive to outliers. Unlike VN and VH, using better ensemble estimators improved the OR performance of VR predictions by 5-10%. This may be explained by the larger uncertainty involved in the VR predictions, which is alleviated by our forecasting ensembles. Notably, determining the single best predictor using DTW in (4.2) performed better than the predictions based on the "best" model parameters during cross-validation. This verifies our earlier observation: the optimal model may vary over different data splits. The probabilistic weighting scheme in (4.3) delivered performance that was competitive with other ensemble methods, such as the median. Given that this scheme is also non-parametric and data-driven, these results are encouraging.

4.8.3 Quantitative Experiments - D_2

Next, we discuss our results on LIVE Mobile Stall Video Database-II (D_2) (see Table 4.8). Overall, the RMN-QoE Predictor outperformed both the RMR and RMH-QoE Predictors, by achieving an excellent outage rate. We found these improvements to be statistically significant. Notably, using ensemble methods greatly improved OR (by more than 10% for both the RMR and RMH models) across all dynamic models. Using an ensemble method other than the mean led to a drop of OR by almost 15% in the case of the RMR-QoE Predictor. This again demonstrates the merits of using a forecasting ensemble for QoE prediction. Note that an outage rate of 0 does not mean that the prediction is perfect; it only indicates that the ensemble predictions were within two times the confidence interval.

Table 4.8: Median OR performance for various time-series ensemble methods applied on the class of RM-predictors on database D_2 (see also Table E.6).

Model Type	RMN	RMR	RMH
best	6.84	21.08	16.22
avg	0.00	11.48	3.71
med	0.00	6.62	4.29
mod	0.00	7.60	4.03
DTW-single	0.00	7.25	3.88
DTW-prob	0.00	7.25	3.38

We also compared the performance of the proposed continuous-time QoE predictors with a subset of the subjective predictions as an upper bound, as described in Section 4.7.4. We found that ensemble forecasts can improve on the prediction performance, but that there is still room for performance

improvements (see Appendix E.3).

When tested on databases D_1 and D_2, the prediction performance of the proposed dynamic approaches was promising; especially when the predictions were combined in an ensemble. However, neither of these databases models both rebuffering events and video quality changes. In the next subsection, we explore the prediction performance of the studied QoE prediction models on the more challenging database D_3.

4.8.4 Quantitative Experiments - D_3

We investigated the performance of the class of G-predictors applied to the more complex problem of QoE prediction when both rate drops and rebuffering occur by using database D_3. Due to rebuffering, computing VQA models is not possible without first removing the stalled frames from each distorted video. Using the publicly available metadata [102], we identified stalled frames and removed them from the distorted YUV video, then calculated the VQA feature, e.g. ST-RRED, on the luminance channels of the distorted and reference videos. As shown in see Table 4.9, the GH-QoE Predictor performed statistically better than the GN-QoE Predictor, while the GR-QoE Predictor lagged in performance. It is likely that more hidden neurons would enable the GN and GR models to perform better.

We also investigated the performance improvements of forecasting ensembles (see Table 4.10). Overall, all forecasting ensembles greatly improved the performance of all dynamic models.

Table 4.9: RMSE significance testing ($m = 3$) on the class of G-predictors (without ensembles) on D_3 using ST-RRED.

Model Type	GN	GR	GH
GN	-	1	0
GR	0	-	0
GH	1	1	-

Table 4.10: Median RMSE performance for various time-series ensemble methods applied on the class of G-predictors on database D_3 using ST-RRED (see also Table E.7).

Model Type	GN	GR	GH
best	0.28	0.37	0.22
avg	0.24	0.29	0.16
med	0.29	0.29	0.11
mod	0.24	0.28	0.10
DTW-single	0.25	0.30	0.13
DTW-prob	0.24	0.29	0.12

As with D_2, we also compared the performance of these QoE predictors with their upper bound (see Appendix E). Interestingly, we found that the ensemble predictions sometimes delivered better performance than the subjective upper bound; an observation that we revisit in Appendix E.

4.9 Conclusions

In this Chapter, we designed simple, yet efficient continuous-time streaming video QoE predictors by feeding QoE-aware inputs such as VQA measurements, rebuffering and memory information into dynamic neural networks. We explored three different dynamic model approaches: non-linear autoregressive models, recurrent neural networks and a block-based Hammerstein-Wiener

model. To reduce forecasting errors, we also proposed ensemble forecasting approaches and evaluated our algorithms on three subjective video QoE databases. We hope that this work will be useful to video QoE researchers as they address the challenging aspects of continuous-time video QoE monitoring.

We now ask a more fundamental question: moving forward, which design aspect of these predictors is most important? Is it the choice of the dynamic model e.g. HW vs. NARX or selecting more sophisticated continuous-time features? The results in Tables 4.6 and E.3, E.4 (see Appendices E.2 and E.3) demonstrate that a better VQA model (e.g. ST-RRED vs. MS-SSIM) or adding more rebuffering-related continuous-time inputs may not always yield statistically significant performance improvements. Tables 4.5, 4.7, 4.8, 4.9 (and Tables E.5, E.6 and E.7 in Appendix E.3) demonstrate that, among the three dynamic models, the RNN were consistently poorly performing while the performance differences between the NARX and HW components were not conclusive: on D_1 and D_2 the NARX-based predictors were better than HW, while for D_3 the HW component improved upon NARX. Meanwhile, using ensemble prediction methods yielded performance improvements in most cases by producing reliable and more robust forecasts. However, these improvements may not be significant if the individual forecasts are similar to each other.

In our preliminary experiments, we also discovered that when our proposed QoE prediction engines were trained on one publicly available database, then tested on another, they delivered poor performance likely due to their dif-

ferent design, e.g., only D_3 studies both rebuffering events and quality changes. This highlights an issue that is at the core of data-driven, continuous-time QoE prediction: lack of publicly-available and diverse subjective data. Existing databases, including D_3, are limited in that they do not sufficiently cover the large space of adaptation strategies, where time-varying quality, network conditions and buffer capacity are all tied together. Therefore, without large and more diverse subjective databases, introducing more sophisticated continuous-time inputs or deploying more complex neural networks will yield relatively small performance gains. In the next Chapter, we describe a large subjective experiment that we designed in order to collect an adequate amount of such data, which will allow us to leverage even more sophisticated learning techniques as in [89] and potentially incorporate other inputs, such as quality switching. Such systems may as well exploit realistic network information extracted from the client side and be used to perceptually optimize bitrate allocation and/or network and bandwidth usage.

Chapter 5

Perceptual Video Quality Assessment for Adaptive Video Streaming

As already discussed, perceptual video quality measurements are an integral component of an adaptive streaming pipeline. To measure video quality, objective video quality assessment (VQA) models are typically deployed. In this Chapter, we focus on FR VQA models for video quality prediction.

There have been numerous approaches to the design of FR VQA algorithms. Image-based approaches [138, 159] exploit only spatial information by capturing statistical and structural irregularities between distorted video frames and corresponding reference frames. A common principle underlying many of these models is that bandpass-filtered responses of high-quality video frames can be modeled as Gaussian Scale Mixture (GSM) vectors [114, 156] and that distorted frames can be quantified in terms of statistical deviations from the GSM model. The GSM approach has been applied in the spatial [29, 159], wavelet [137, 143] and DCT [127] domains. Importantly, frame differences of high quality videos can also be modeled as GSM vectors in order to measure temporal video distortions [29, 137, 143].

Video-based FR VQA models have also been studied in the literature

[87, 109, 134, 155]. In [155], the notion of spatio-temporal slices was derived and the "most apparent distortion principle" [72] was applied to predict video quality. Optical flow measurements were also used in [87], where video distortions were modeled by deviations between optical flow vectors. A space-time Gabor filterbank was used in [134] to extract localized spatio-spectral information at multiple scales. VQM-VFD [109, 111] used a neural network trained with a large number of features such as edge features. These algorithms often deliver good performance on small size videos, but are computationally inefficient on long HD video sequences, since they apply time-consuming spatio-temporal filtering operations.

Data-driven models hold great promise for the VQA problem [73, 79, 95, 109, 162]. Netflix recently announced the Video Multimethod Fusion Approach (VMAF), which is an open-source, learning-based FR VQA model. VMAF combines multiple elementary video quality features using an SVR trained on subjective data, and focuses on compression and upscaling artifacts. Nevertheless, it does not fully exploit temporal quality information sensitive to temporal video distortions.

The open-sourced VMAF framework can be used as a starting point to develop better VQA models by integrating stronger quality features and training data. Here we leverage these capabilities by proposing two ways to improve upon the current VMAF framework. The first approach, called SpatioTemporal VMAF, integrates strong temporal features into a single regression model. The second enhancement (Ensemble VMAF) trains two separate models and

then performs prediction averaging to predict video quality. Both approaches rely on statistical models of frame differences and hence avoid computationally expensive spatio-temporal filtering. To train our models, we designed a large subjective experiment (VMAF+ database) and evaluated these models in three experimental applications: video quality prediction, QoE prediction, and Just-Noticeable Difference (JND) prediction.

The rest of the Chapter is organized as follows. Section 5.1 describes the current VMAF system and highlights its capabilities and limitations. Sections 5.2 and 5.3 discuss the SpatioTemporal and Ensemble VMAF improvements. Section 5.4 gives an overview of the VMAF+ subjective dataset that we built. Section 5.5 details experimental results, while Section 5.6 concludes this Chapter.

5.1 Background on VMAF

5.1.1 How VMAF works

VMAF extracts a number elementary video quality metrics as features and feeds them into an SVR [79]. This allows VMAF to preserve and weight the strengths of each individual feature and align the objective predictions with ground truth subjective data. The VMAF system includes the following steps (see Fig. 5.1): feature extraction and aggregation, training/testing and temporal pooling.

The first step is to extract a number of quality metrics as perceptually-relevant features: DLM [78], VIF [138] and the luminance differences between

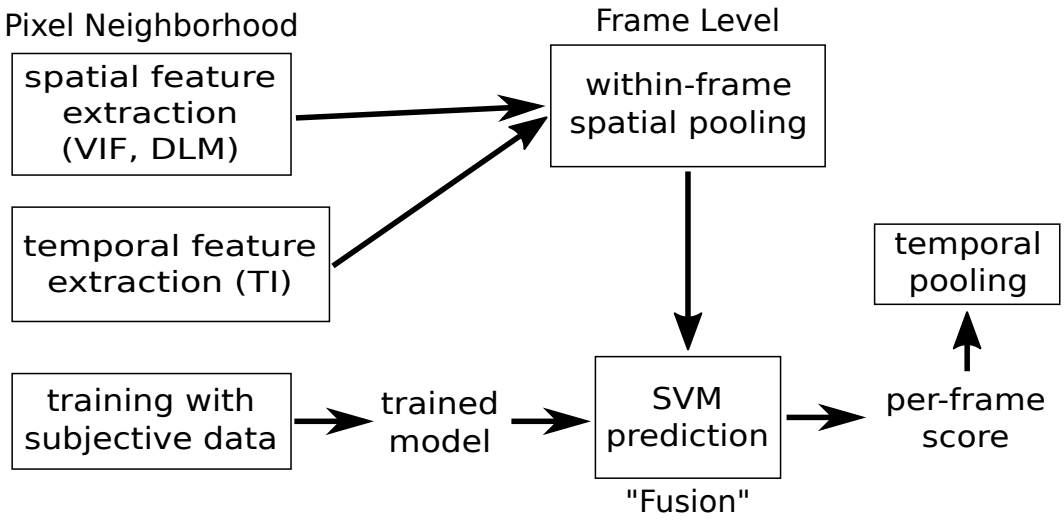

Figure 5.1: Outline of the current VMAF system.

pairs of frames (Temporal Information - TI). The DLM feature captures detail losses and is calculated by a weighted sum of DLM values over four different scales. The VIF feature captures losses of visual information fidelity and is computed at four scales, yielding four VIF features. The TI feature aims to capture temporal effects due to motion changes which are quantified by luminance differences, resulting in six features overall. The TI feature is currently the only source of temporal quality measurement in VMAF.

Each of these six features is extracted as a feature map of size equal to the corresponding scale. Next, the average value of each feature map is calculated, to produce one feature value per video frame and feature type. For training purposes, VMAF aggregates the per frame features over the entire video sequence, yielding one feature value per training video. These six feature values are fed, together with the corresponding subjective ground truth, to an

SVR model. For testing purposes, VMAF predicts one value per video frame and calculates the arithmetic mean over all per frame predictions to predict the overall video quality.

5.1.2 VMAF Limitations and Advantages

VMAF has been developed with a particular application context in mind. For the Netflix use case, there are two main video impairments that are of interest: compression and scaling artifacts. Compression artifacts are typically observed as blocky regions within a frame, while scaling artifacts arise when the encoding resolution is lower than the display resolution and are usually observed as jerky regions around edges. Both of these artifact types are introduced while encoding the video content. Packet loss transmission distortions are not a problem for HTTP adaptive streaming applications which rely on the TCP transfer protocol.

Under this specific application context, VMAF achieves good predictive performance by weighting the elementary video quality features. Figure 5.2 illustrates an example of the performance gains afforded by VMAF fusion. Importantly, VMAF has been trained on video sources which contain film grain noise. The effects of film grain on perceived video quality are not always clear, since film grain may be reduced due to compression and sometimes possesses an aesthetic subjective appeal. By training on the presence of film grain, VMAF "learns" to account for these phenomena when performing video quality predictions.

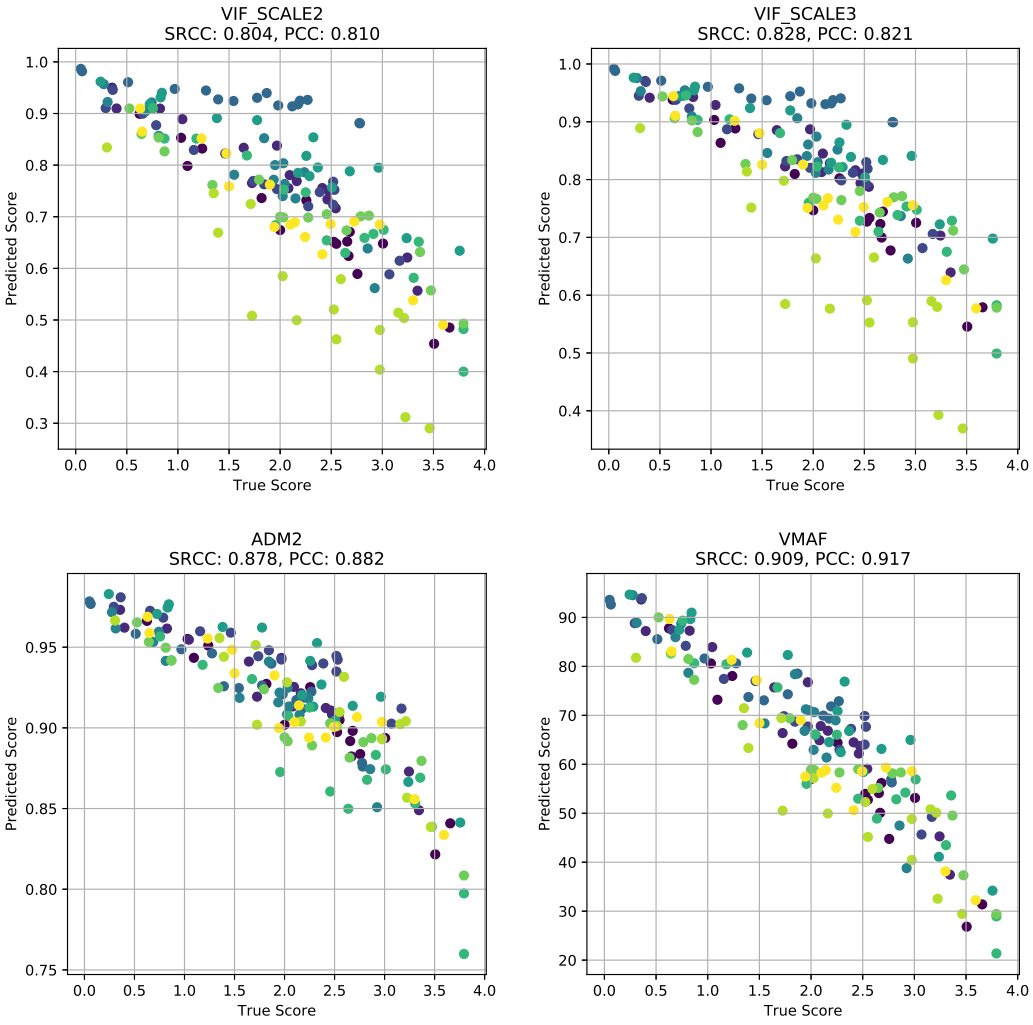

Figure 5.2: Performances of the individual VMAF features and the fusion result on the LIVE Mobile VQA Database [100]. Left to right: VIF calculated at scales 2 and 3; DLM; VMAF fusion. When training VMAF, we relied on the NFLX dataset [79]. The performance metrics and our model evaluation are described in greater detail in Section 5.5.

Aside from average frame difference measurements (TI feature), VMAF does not exploit temporal quality measurements. The TI feature attempts to capture motion masking effects, i.e., the reduction of distortion visibility due to large motion changes. Nevertheless, TI measurements are more related to the video content itself and do not effectively account for temporal masking. Temporal video distortions, such as ghosting, flickering and motion estimation errors, are quite complex in nature and deeply impact perceived video quality [134]. Since compression standards are evolving and even lower encoding rates are being used [12], it is important for FR VQA models, such as VMAF, to generalize well on unseen video distortions.

5.2 SpatioTemporal VMAF

5.2.1 S-SpEED and T-SpEED features

Extracting temporal quality information is important for VQA models, but space-time VQA models are often computationally intensive, since they employ motion estimation or spatiotemporal filtering. To extract temporal quality measurements, we exploit statistical models of frame differences in high-quality videos similar to [137, 143] and [29]. The main idea is to model the bandpass-filtered map responses of frames and frame differences as GSM vectors [114, 156] and use entropic differencing to predict visual quality. To calculate these entropy values, a conditioning step is applied which removes local correlations from band-pass filtered coefficients. Conditioning is equivalent to divisive normalization [126]; a process that is known to occur in the

early stages of vision [37, 49, 93].

We build our work on the recently developed SpEED-QA model [29], which extracts information-theoretic information in the spatial domain. A diagram of the feature extraction steps is shown in Fig. 5.3. First, let F_i be the ith video frame and $D_i = F_{i+1} - F_i$ be the ith frame difference of the reference or the distorted video. Then, downsample D_i to the kth scale, which yields the $D_{i,k}$ frame difference map. Then filter $D_{i,k}$ with a spatial Gaussian filter and perform local mean subtraction in the spatial domain. This local operation approximates the multi-scale multi-orientation steerable filter decomposition used in [143] and is very compute-efficient.

Entropy measurements and entropic differencing have been shown to correlate quite highly with human judgements of video quality [29, 143]. Therefore, our next step is to calculate the local entropies in the reference and the distorted video for the local mean-subtracted response map (MS map). These steps are visualized in Fig. 5.4. We split the response map into $b \times b$ non-overlapping blocks yielding the coefficients C_{mk} for block m and scale k. These coefficients can be modeled as a GSM vector, i.e., $C_{mk} = S_{mk}U_{mk}$, where S_{mk} represents the variance field and is a non-negative random variable independent of $U_{mk} \sim \mathcal{N}(0, \mathbf{K}_{U_k})$. We model the neural noise present along the visual pathway using an additive white noise model, i.e, $C'_{mk} = C_{mk} + W_{mk}$, where $W_{mk} \sim \mathcal{N}(0, \sigma_w^2 \mathbf{I}_N)$, $\mathbf{I}_N$ is the $b \times b$ identity matrix and $N = b^2$ is the number of coefficients per block. In our implementation, we fix $\sigma_w^2 = 0.1$, $b = 5$ and use a 7×7 isotropic gaussian filter of standard deviation $7/6$.

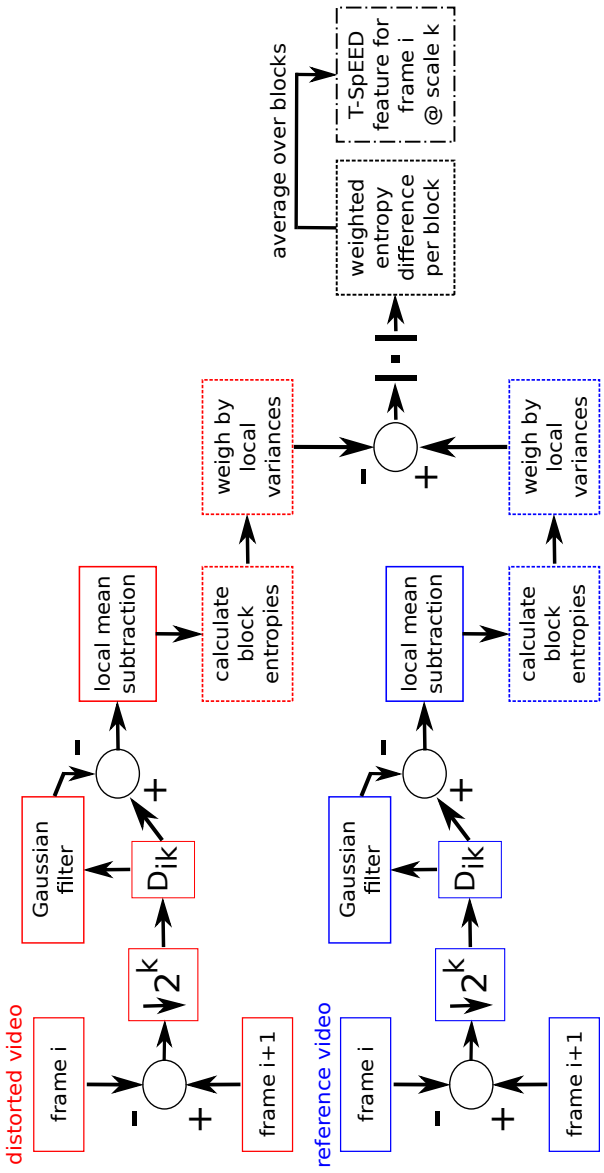

Figure 5.3: T-SpEED feature extraction. Blue and red colors denotes the reference and distorted videos respectively. A dashed box outline denotes that these operations are performed on each block of the MS map, while dashed and bulleted outline denotes a single value per frame. When extracting the S-SpEED features, the diagram remains the same, except that whole video frames are used instead of frame differences.

To predict video quality, SpEED-QA calculates the entropy differences between a reference and a distorted video at the lowest scale ($k = 4$). To this end, we also apply conditioning on the block variances s_{mk}, which are realizations of S_{mk} and compute the entropies of the noisy bandpass coefficients C'_{mk}, i.e.,

$$h(C'_{mk}|S_{mk} = s_{mk}) = \frac{1}{2}\log[(2\pi e)^N|s^2_{mk}\mathbf{K}_{U_k} + \sigma^2_w\mathbf{I}_N|] \tag{5.1}$$

To determine s_{mk}, calculate the sample variance on every non-overlapping block of the MS map. To estimate the $b \times b$ covariance matrix $\mathbf{K}_{U_k}$, we use a sliding window to collect all *overlapping* blocks from the MS map and compute the sample covariance. The use of overlapping blocks in this step ensures that a sufficient number of samples is available for covariance estimation, especially for lower scales.

Following entropy calculation, the block entropies are further weighted by a logarithmic factor, i.e., $\log\big(1 + s^2_{mk}\big)$. This logarithmic factor is applied twice; once for the temporal and once for the corresponding spatial variances. This step lends a local nature to the model and ensures numerical stability at regions of low spatial or temporal variance. To measure the statistical distance between the GSM models of the distorted and reference video frames, the weighted block entropy values are differenced and the absolute values of those differences are computed. The absolute values are averaged over all blocks, yielding the T-SpEED feature for frame i and scale k. This feature captures the information loss due to temporal video distortions. To capture spatial

quality degradations, we can also define the corresponding S-SpEED feature by performing local mean subtraction on $F_{i,k}$ instead of $D_{i,k}$, then following the exact same steps. The only difference is that the logarithmic weighting only involves the spatial variance term, unlike T-SpEED.

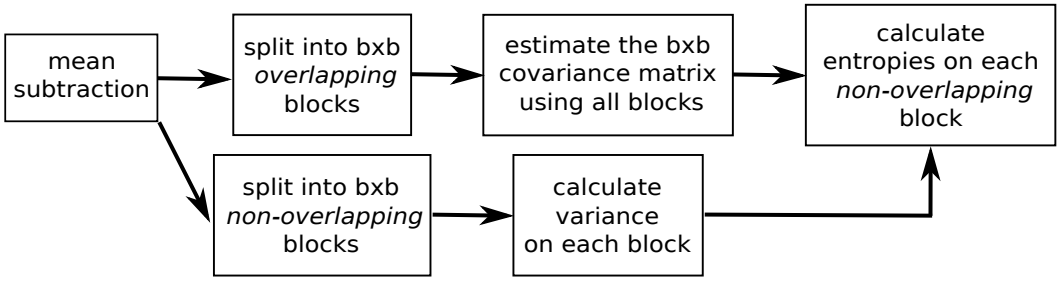

Figure 5.4: Details on entropy calculation for S-SpEED and T-SpEED.

5.2.2 SpatioTemporal Feature Integration

Despite the good performance of SpEED-QA in a number of databases [29], it does not account for the effects of film grain on the perceived visual quality and does not exploit multiscale information. Previous studies have established the merits of multiscale information for image and video quality assessment [160]. The human visual system processes visual information in a multiscale fashion, while images demonstrate significant self-similarities. Notably, multiscale algorithms incorporate the effects of different display sizes and viewing distances. Unfortunately, unlike image quality applications, incorporating multiscale information for VQA is not as easy.

In preliminary experiments, we discovered that the use of temporal entropy differences across multiple scales yields complementary perceptual in-

formation. To exploit this observation and combine information across scales, we adopt a data-driven approach to learn the contribution from each scale and predict visual quality. Due to a motion downshifting phenomenon [143], lower scales yield stronger features, hence we extract T-SpEED features from scales 2, 3 and 4. The use of scale k denotes that the frame difference MS map is downscaled by a factor of 2^k, which allows for more efficient feature extraction.

To complement the T-SpEED features, we found that applying VIF on the frame difference signal [137] across multiple scales leads to further improved performance. We call these features T-VIF (4 features calculated from scales 0, 1, 2 and 3). Both T-SpEED and T-VIF measure temporal information loss using the GSM statistical model [156] on frame differences, but T-VIF relies on mutual information between wavelet coefficients. Since the 5 spatial VMAF features (DLM and VIF from 4 scales) sufficiently capture spatial quality degradations, we include them in our model as well. Overall, the proposed SpatioTemporal VMAF (ST-VMAF) approach deploys 12 perceptually relevant features (5 from VMAF, 3 from T-SpEED and 4 from T-VIF) which capture both spatial and temporal information. Compared to other feature candidates, we found that the proposed feature set delivers the best performance.

Similar to the original VMAF approach, we average the per frame features during training but perform per frame ST-VMAF predictions when testing. This design choice did not have an effect on the predictive performance of the ST-VMAF model. This also enables ST-VMAF to be used as an input

to a larger, online, QoE prediction system (see Section 5.5).

To calculate the aggregate quality over an entire video sequence, we applied the hysteresis temporal pooling method in [132]. Human opinion scores vary smoothly over time, while objective predictions respond sharply to visual changes. Meanwhile, subjective quality perception is driven by memory/recency, i.e., more recent experiences tend to more deeply affect current visual impressions. Based on these observations, we applied a linear low-pass operator and a non-linear rank order weighting on the objective prediction scores, as suggested in in [132].

5.3 Ensemble VMAF

5.3.1 Why an Ensemble Model?

In the previous section, we described a simple way to integrate strong temporal quality measurements into VMAF, by concatenating the spatial VMAF features with the T-VIF and T-SpEED features. However, in cases where the available subjective video data is limited, increasing the feature dimensionality (or using deep neural networks) may lead to overfitting. Video databases are usually pretty diverse in their design and contents and hence a particular feature subset may work well on one dataset, but not on another. For example, we have empirically observed that the ADM feature carries a large weight for spatial degradations, but does not generalize well on unseen data. One option is to carefully tune the regression model parameters to effectively regularize the predictions. Another alternative, which we have decided

to follow here, is to consider fusion approaches.

Model fusion (or bagging) is a well-studied concept [112] which combines multiple individual learners. The main idea is to fuse multiple simple models that are easier to tune, and that complement each other towards reducing the prediction variance. Among other fusion possibilities, we experimented with training multiple SVRs on different video databases or training different regressors (e.g. a Random Forest and a SVR) on the same dataset. Nevertheless, we found that aligning predictions coming from models that were trained on subjective data collected under different experimental conditions and/or assumptions was a difficult proposition. We also found that the SVR predictions always outperformed Random Forest predictions and hence their combination was not beneficial. The performance merits of using an SVR for image and video quality assessment have also been demonstrated in [79, 93, 127]. These observations led us to the design of Ensemble VMAF (E-VMAF), which we describe next.

5.3.2 An Ensemble Approach to Video Quality Assessment

We propose E-VMAF, an ensemble enhancement to VMAF, wherein multiple feature subsets are used to train diverse VQA models that are then aggregated to deliver a single prediction value. Nevertheless, training and combining multiple models can significantly increase the complexity, which can be challenging for a VQA model if it is to be deployed at a global scale. Driven by simplicity, we trained two SVR models on the VMAF+ database

(see also Section 5.4), and then averaged the individual predictions, as shown in Fig. 5.5.

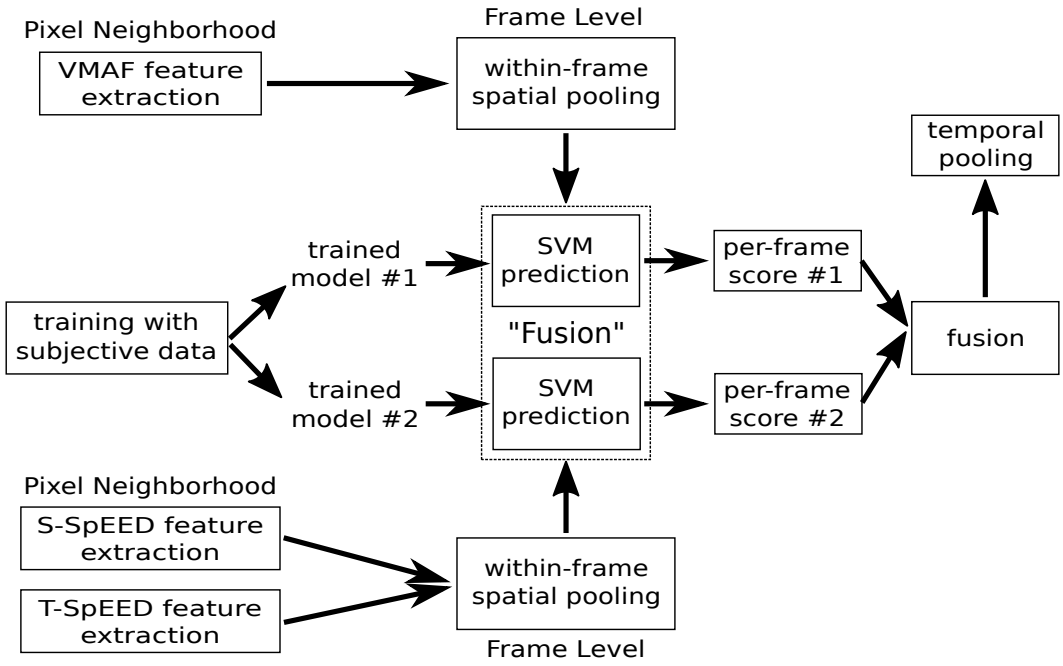

Figure 5.5: Overview of the ensemble approach.

Given that the VMAF feature set already captures the combined effects of compression and scaling (and the predominance of these distortions in practice), we use the same features (6 features) for the first individual model, denoted by M_1. To design M_2, it is desirable to capture both spatial and temporal quality measurements, such that the individual predictions are accurate enough. Motivated by the perceptual relevance of the T-SpEED features used in ST-VMAF, we combined the 3 T-SpEED features with the 3 S-SpEED features calculated at the same scales (2, 3 and 4).

The VIF features of M_1 and the S-SpEED features of M_2 both exploit

the GSM model of high-quality video frames, but they also have some differences. The S-SpEED features are based on conditional entropies which are weighted by local variances, while VIF uses mutual information between reference and distorted image coefficients. Temporal quality measurements are complementary between the two models: T-SpEED of M_2 expresses temporal information loss by conditioning and applying temporal variance weighting, while the TI feature of M_1 measures motion changes as a proxy for temporal masking effects.

Interestingly, we found that optimizing weighted averages of the individual predictions from M_1 and M_2 did not yield significant performance gains. This suggests that the prediction power of the two learners are at near-parity. The prediction averaging step produces a single prediction per frame which is then averaged over all frames of each test video. For the time averaging step, we again employed the hysteresis pooling method [132], as in ST-VMAF.

5.4 The VMAF+ Subjective Dataset

Data-driven approaches to VQA deeply depend on the training data that is used to train the regressor engines. We believe that a useful training dataset should include a diverse and realistic set of video contents and simulate diverse yet practical distortions of varying degradation levels. Collecting consistent subjective data has the potential to significantly increase the performance of data-driven VQA models on unseen data.

To this end, we conducted a large-scale subjective study targeting mul-

tiple viewing devices and video streams afflicted by the most common distortions encountered in large geographic-scale video streaming: compression and scaling artifacts. We first gathered 29 10-second video clips from Netflix TV shows and movies, from a variety of content categories, including, for example, drama, action, cartoon and anime. The source videos were of different resolutions, ranging from 720x480 up to 1920x1080, while the frame rates were 24, 25 or 30 frames per second. In our content selection, we also included darker scenes, which are particularly challenging for encoding and video quality algorithms. It should be noted that some of the source videos contain film grain noise, which is more often found in older (legacy) content. This allows us to gather valuable subjective data on videos that not only suffer from compression and scaling artifacts, but importantly, where there may be degradations of quality in the original source video.

To describe content variation and encoding complexity, we employed an approach different from the usual SI-TI plots [161]. We encoded all video contents using a fixed Constant Rate Factor (CRF) setting of 23, then measured the bitrate of each video file. Figure 5.6 shows that the video contents span a large range of encoding complexities, from less than 1Mbps up to around 19 Mbps.

In streaming applications, the source video is usually divided into smaller chunks (e.g. of 2 seconds each) and stored in multiple representations, where each representation is defined by a specific pair of an encoding resolution and bitrate level. To generate the distorted videos, we downsampled each

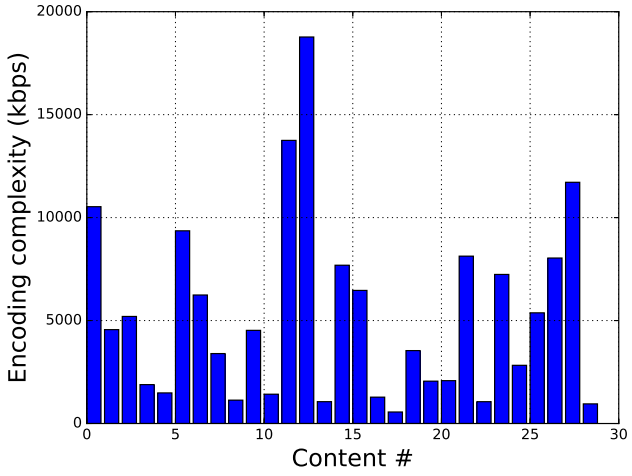

Figure 5.6: Encoding complexity across contents, expressed as the bitrate (in terms of kbps) of a fixed CRF 23 encode using libx264.

source video to six different encoding resolutions: 320x240, 384x288, 512x384, 720x480, 1280x720 and 1920x1080, then encoded them using the H.264 codec using three different CRF values: 22, 25 and 28, thereby yielding 18 distorted videos per content. For display purposes, all of the videos were upscaled to 1920x1080 display resolution. Both the downscaling and upscaling operations were performed using a lanczos filter. Due to copyright restrictions, the videos cannot be made publicly available.

To avoid subjective fatigue, we employed a content selection scheme, where each subject only viewed a subset of all video contents. To avoid any memory biases, we ensured that video contents were displayed in a random order such that no video content was consecutively displayed. Overall, we gathered more than 20000 scores from 167 subjects on three different viewing devices: mobile, laptop and television. When training our models, we applied

standard subject rejection protocols [63] and used the laptop subset of human opinion scores. Figure 5.7 shows the distribution of the raw subjective data. It can be seen that the scores widely cover the subjective range. The outcome of our subjective test is the VMAF+ video quality database, which we found to be an excellent source of training data for developing learning-based FR-VQA models (see Section 5.5).

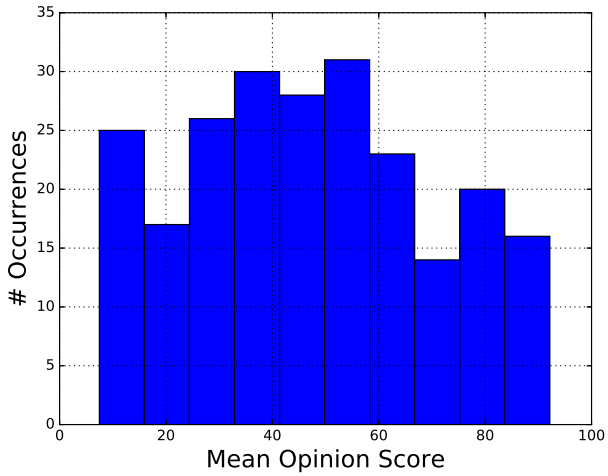

Figure 5.7: Mean opinion score distribution on the VMAF+ database.

5.5 Experimental Analysis

In this section, we discuss a series of experiments on three different and important video quality applications: subjective video quality prediction, Just-Noticeable Difference (JND) prediction and video QoE prediction. For evaluation purposes, we used the Spearman Rank Order Correlation Coefficient (SROCC), the Pearson Linear Correlation Coefficient (PLCC) and the root-mean-squared error (RMSE). The SROCC measures the monotonic rela-

tionship between the objective predictions and the ground truth data, while PLCC measures the degree of linearity between the two. The SROCC and PLCC correlation coefficients describe the overall agreement between subjective and objective scores, hence a better objective metric should produce a higher correlation number. Before computing PLCC, a non-linear logistic fitting was applied to the objective scores as outlined in Annex 3.1 of ITU-R BT.500-13 [63].

We evaluated the proposed approaches against a number of popular FR (and RR) VQA models. We tested the following VQA methods:[1] PSNR, PSNR-hvs [113], SSIM [159], MS-SSIM [160], ST-RRED [143], SpEED-QA [29], ST-MAD [155], VQM-VFD [109] and VMAF version 0.6.1 [79]. For VMAF 0.6.1 we used the suggested parameters, which were also used for E-VMAF. We found this simple parameter selection scheme to work very well for E-VMAF. In our experiments, performing cross-validation for ST-VMAF on the VMAF+ dataset led to overfitting of unseen distortions, hence we empirically fixed $C = 0.5$ and $\gamma = 0.04$. This parameter selection delivered consistent results across a large number of databases, as we will demonstrate next.

In the experiments, we relied on a wide variety of subjective video databases: LIVE VQA [131], LIVE Mobile [100][2], CSIQ-VQA [154], NFLX [79], SHVC [20][3], VQEG-HD3 [18], EPFL-Polimi [44], USC-JND [158], LIVE-

[1]We did not test MOVIE [134] since it is very time consuming when applied on HD videos.

[2]We excluded frame freezes and used only the mobile subset.

[3]We excluded videos from Session 3 due to content overlap with the NFLX set.

Table 5.1: Subjective Database Overview. TE: transmission errors, RA: rate adaptation, MJPEG: motion JPEG compression, WC: compression using wavelet, AWN: additive white noise, QoE: rate adaptation and/or rebuffering. yuv420p8b: planar YUV 420, 8-bit depth, yuv420p10b: planar YUV 420, 10-bit depth.

Database	# Videos	Resolution	Duration	Frame Rate	Format	Distortion Type
LIVE VQA [131]	150	768x432	10 sec.	25, 50	yuv420p8b	H.264, MPEG-2, TE
LIVE Mobile [100]	160	1280x720	15 sec.	30	yuv420p8b	H.264, TE, RA
CSIQ-VQA [154]	216	832x480	10 sec.	24, 25, 30 50, 60	yuv420p8b	H.264, H.265, MJPEG WC, TE, AWN
VMAF+	290	1920x1080	10 sec.	24, 25, 30	yuv420p8b	H.264 and scaling
NFLX [79]	300	1920x1080	6 sec.	24, 25, 30	yuv420p8b	H.264 and scaling
SHVC [20]	64	1920x1080	$\cong$ 10 sec.	25, 50	yuv420p8b yuv420p10b	HEVC
VQEG HD3 [18]	135	1920x1080	10 sec.	30	yuv420p8b	H.264, MPEG-2, TE
EPFL [44]	144	352x288 704x576	10 sec.	30	yuv420p8b	H.264, TE
USC-JND [158]	3520	1920x1080, 1280x720 960x540, 640x360	5 sec.	24, 30	yuv420p8b	H.264
LIVE-NFLX [27]	112	1920x1080	¿ 60 sec.	24, 25, 30	yuv420p	QoE
LIVE-HTTP [38]	15	1280x720	300 sec.	30	yuv420p8b	QoE

NFLX [27] and LIVE-HTTP [38]. These databases contain a large variety of distortion types, including H.264 and HEVC compression and dynamic rate adaptation, scaling, packet loss, transmission errors and rebuffering events. Importantly, our experimental analysis includes videos with various resolutions, ranging from 352x288 up to 1920x1080, and frame rates (24, 25, 30, 50 and 60 fps). An overview of these databases is given in Table 5.1.

5.5.1 Video Quality Prediction

We begin our experimental analysis with the problem of video quality prediction. To accurately evaluate performance, we focused on cross-database results, i.e., we relied on the VMAF+ subjective dataset for training and

Table 5.2: SROCC performance comparison on multiple Video Quality Subjective Databases. VMAF, ST-VMAF and E-VMAF were trained on the VMAF+ dataset. The best overall performance is denoted by boldface.

Database	LIVE VQA	LIVE Mobile	CSIQ-VQA	NFLX	SHVC	VQEG HD3	EPFL	overall SROCC	overall PLCC
PSNR	0.523	0.687	0.579	0.705	0.755	0.770	0.753	0.691	0.677
PSNR-hvs	0.662	0.757	0.599	0.819	0.828	0.798	0.904	0.785	0.788
SSIM	0.694	0.757	0.698	0.788	0.754	0.907	0.712	0.771	0.752
MS-SSIM	0.732	0.748	0.749	0.741	0.715	0.898	0.931	0.808	0.791
ST-RRED	0.805	0.892	0.805	0.764	0.889	0.912	0.944	0.872	0.777
SpEED-QA	0.776	0.897	0.741	0.781	0.879	0.909	0.936	0.861	0.759
ST-MAD	0.825	0.663	0.735	0.768	0.611	0.847	0.901	0.782	0.769
VQM-VFD	0.804	0.816	0.839	0.931	0.863	0.939	0.850	0.873	0.870
VMAF 0.6.1	0.756	0.906	0.614	0.928	0.887	0.850	0.836	0.847	0.853
ST-VMAF	0.809	0.905	0.784	0.927	0.888	0.932	0.945	**0.897**	**0.898**
E-VMAF	0.792	0.929	0.761	0.930	0.892	0.906	0.942	0.894	0.895

tested on the rest of the video databases. For each VQA model, we report the SROCC values per testing dataset, as well as an aggregate SROCC and PLCC value. To compute the aggregate correlation score, we applied Fisher's z-transformation [39], i.e.,

$$z = \frac{1}{2}\ln\frac{1+r}{1-r}, \text{where } r \text{ is SROCC or PLCC,} \tag{5.2}$$

to the correlation values and then averaged them over all tested databases. The average value was then transformed back using the inverse of (5.2). Table 5.2 shows the results of this experimental analysis.

Among image-based models, such as PSNR and SSIM, PSNR delivered the worst performance. This is expected, since it is a signal fidelity metric that does not exploit perceptual information. SSIM and PSNR-hvs performed considerably better and MS-SSIM achieved further performance gains, likely due to the multiscale properties captured therein. Nevertheless, none of these

spatial metrics exceeded an aggregate SROCC of 0.81, which demonstrates the importance of capturing temporal information.

Regarding video-based models, ST-MAD did not perform well and was very time-consuming (see Section A.2). VMAF 0.6.1 delivered excellent performance on the NFLX dataset, which is expected, given that it mostly captures compression and scaling artifacts. However, it demonstrated poor generalization capabilities on unseen distortions, such as the CSIQ-VQA database. ST-RRED and SpEED-QA performed well on most databases in terms of SROCC, but neither algorithm performed well on the NFLX dataset, which may be due to the presence of film grain in some of the source content. Notably, the aggregate PLCC of ST-RRED and SpEED-QA was relatively low. Unlike VQA models trained on subjective data, such as VMAF or VQM-VFD, the ST-RRED and SpEED-QA predictions were highly non-linear with ground truth. VQM-VFD delivered similar SROCC performance, but, unlike ST-RRED and SpEED-QA, it uses a number of basic features that are fed to a neural network trained on a very large number of subjective datasets.

From the above analysis, it can be seen that VMAF does not fully exploit temporal information and does not generalize well on unseen distortions. At the same time, untrained VQA models such as ST-RRED do not exhibit a linear relationship with subjective ground truth, do not capture the effects of film grain and do not combine multiscale information. The methods we have developed here aim to bridge this gap and combine the best of both worlds. Table 5.2 shows that ST-VMAF achieved standout aggregate perfor-

mance across all databases, while E-VMAF functioned nearly as well. Both models achieve this excellent level of video quality prediction power using a *single training dataset and a single parameter setting*. It should be noted that both ST-VMAF and E-VMAF considerably improve on VMAF, although they were trained on the VMAF+ dataset, which focuses only on compression and scaling artifacts. For example, on the CSIQ-VQA database, which contains multiple distortion types other than compression, ST-VMAF and E-VMAF both perform quite well. This strongly suggests that these new models possess excellent generalization capabilities beyond their demonstrated state-of-the-art VQA performance. In Appendix A, we further analyze the performance gains and the computational efficiency of the proposed VQA models.

5.5.2 JND and QoE Prediction

Another interesting application of VQA models is JND detection, i.e., identifying JND points, and comparing them with the detection capabilities of humans. The USC-JND dataset [158] was designed specifically for this purpose. It contains a large number of videos, JND points and human opinion scores. We selected several leading VQA models and reported their JND prediction performance in Table 5.3. For this detection task, we did not employ the hysteresis temporal pooling, since detection is a different task. Both ST-VMAF and E-VMAF outperformed other powerful VQA models, including ST-RRED and VMAF.

An important emerging application of perceptual video quality models

Table 5.3: USC-JND performance comparison. VMAF, ST-VMAF and E-VMAF were trained on the VMAF+ dataset. The best performing algorithms are denoted by boldface.

Database	SROCC	PLCC
PSNR	0.616	0.589
SSIM	0.718	0.602
MS-SSIM	0.815	0.739
ST-RRED	0.844	0.735
SpEED-QA	0.843	0.727
VMAF 0.6.1	0.853	0.854
ST-VMAF	**0.877**	**0.856**
E-VMAF	**0.875**	**0.869**

is streaming video QoE prediction. In streaming applications, the reference video is usually available, hence reference models are more relevant. The predominant video impairments that occur during video streaming are compression, spatial scaling artifacts, and rebuffering events. We studied the behavior of the ST-VMAF and E-VMAF models on the recently released LIVE-NFLX Video QoE Database [27], which simulates realistic buffer and network constraints, and contains rebuffering events, rate adaptations and constant bitrate encodes. Table 5.4 shows that none of the considered FR-VQA models performed particularly well, since they do not model the effects of rebuffering. This suggests that more sophisticated QoE predictors (than just VQA algorithms) are required for the more general problem of QoE assessment. However, both of the new models achieved better performance than all the other

models, especially in terms of PLCC[4].

Table 5.4: Quantitative performance comparison on the LIVE-NFLX Video QoE Database [27], including both compression and rebuffering events. The best performing algorithm is denoted by boldface.

VQA	SROCC	PLCC
PSNR	0.515	0.507
PSNR-hvs	0.535	0.545
SSIM	0.701	0.726
MS-SSIM	0.683	0.710
ST-RRED	0.702	0.715
SpEED-QA	0.712	0.727
VMAF 0.6.1	0.607	0.667
ST-VMAF	**0.735**	**0.780**
E-VMAF	0.721	0.772

We also examined the potential of incorporating the ST-VMAF and E-VMAF VQA models into an existing continuous-time QoE predictor. We tested the revised QoE predictor using the LIVE-HTTP Video QoE Database [38] which studies the effects of HTTP-based rate adaptation on 5 min. long HD video sequences. Table 5.5 reports the outcomes of the experiments when using the NARX QoE predictor [25], which has demonstrated promising results on the few available video QoE databases. First, we split the database into content independent train-test splits, then determined the best NARX configuration on the training set. Next, we tested the selected parameter setting on the test videos using a number of leading VQA models as integral components

[4]VQM-VFD cannot be applied to videos of duration more than 15 sec. and hence is excluded

of the NARX QoE predictor. For evaluation purposes, we reported the SROCC and root mean squared error (RMSE) values between the continuous QoE predictions and the continuous ground truth data. It can be seen that ST-VMAF outperformed all of the other VQA models when used in this way, suggesting that it is an excellent choice for inclusion in future perceptually-driven online QoE prediction systems. Ultimately, we envision deploying high-performance QoE predictors to design practical perception-driven rate adaptation and network allocation protocols.

Table 5.5: Quantitative performance comparison on the LIVE-HTTP [38] Video QoE Database when using the continuous-time NARX [25] QoE predictor. The best performing algorithm is denoted by boldface.

VQA	SROCC	RMSE
PSNR	0.731	6.708
SSIM	0.901	3.844
MS-SSIM	0.881	4.248
ST-RRED	0.885	4.226
VMAF 0.6.1	0.883	4.321
ST-VMAF	**0.924**	**3.515**
E-VMAF	0.922	3.666

5.5.3 Observations and Takeaways

In our experiments, we demonstrated that both ST-VMAF and E-VMAF performed very well for video quality and JND prediction and have the potential to be integrated with QoE predictors. Between the two, their performances are quite similar: E-VMAF was slightly better in terms of PLCC

for JND prediction (see Table 5.3) and ST-VMAF was a bit better in terms of SROCC in the LIVE-NFLX experiment (see Table 5.4). The benefit of using E-VMAF is that it is easier to tune, since using the same SVR parameters as VMAF yielded excellent results. By contrast, to train ST-VMAF, its larger number of features (compared to the VMAF baseline) had to be regularized using more careful SVR tuning. Nevertheless, in applications where a compact feature and model representation is required, ST-VMAF might be a preferred solution.

5.6 Conclusion

In this chapter, we developed two high-performing, data-driven full reference video quality assessment models. We have shown how strong temporal and spatial quality measurements can be integrated into a recently developed video quality prediction system. Both models can be easily deployed into the Netflix VMAF ecosystem and hence can be applied to perceptual video quality at global scale. In the future, we plan to further improve those models by combining NR source VQA measurements with the FR system towards accounting for possible degradations of the original source/reference video. To do so, we also plan to develop better data-driven NR video quality models that can be used in lieu of existing NR VQA [127] approaches. Towards achieving this goal, it will be very interesting to exploit the ensemble fusion idea proposed here on the NR VQA problem.

Chapter 6

End-to-end Perceptual Adaptive Streaming: A Subjective Study

6.1 Introduction

Understanding and predicting QoE for adaptive video streaming is an emerging research area [22, 24, 25, 27, 45, 50, 58, 96, 135, 145, 149, 150]. Existing QoE studies do not fully capture important aspects of practical video streaming systems, e.g., they do not incorporate actual network measurements and client-adaptation strategies. To this end, we built the LIVE-NFLX-II database, a large subjective QoE database that integrates perceptual video coding and quality assessment, using actual measurements of network and buffer conditions, and client-based adaptation. To construct our database, we relied on an adaptive streaming prototype that consists of four modules; an encoding module, a network module, a video quality module and a client module.

A unique characteristic of the subjective database presented herein is that we incorporate recent developments in large-scale video encoding and adaptive streaming. To generate video encodes, we make use of an encoding optimization framework [67] that selects encoding parameters on a per-shot

basis, guided by a state-of-the-art video quality assessment algorithm (VMAF) [79].

To model video streaming, we use actual network measurements and a pragmatic client buffer simulator, rather than just simplistic network and buffer occupancy models. Given the plethora of network traces and adaptation strategies, the database captures multiple streaming adaptation aspects, such as video quality fluctuations, rebuffering events of varying durations and numbers, spatial resolution changes, and video content types. The subjective data consists of both retrospective and continuous-time scores, which makes it ideal for training various QoE models. Lastly, the video database is considerably larger and publicly-available. To highlight the contribution of the new database, Table 6.1 shows its advantages over existing ones.

Table 6.1: High-level comparison with other relevant video streaming subjective studies.

Description	[118]	[145]	[150]	[168]	[83]	[38]	[45]	[53]	[27]	LIVE-NFLX-II
client adaptation	X									X
continuous QoE		X				X		X	X	X
actual network traces	X									X
buffer model	X								X	X
public					X	X	X	X		X
$>$ 400 test videos										X
$>$ 60 subjects	X		X							X
rebuffering + quality	X	X		X			X		X	X
content-based encoding			X		X					X

6.2 Previous Works

To develop QoE-aware video streaming systems, it is important to analyze human subjective data of user experience, develop QoE prediction models

and integrate these models into client-adaptation algorithms. With this setting in mind, we constructed the LIVE-NFLX-II database to collect human subjective data, train QoE models and study client adaptation algorithms. Therefore, it is important to relate our work with previous works on each of these QoE-related aspects.

6.2.1 Subjective Analysis of HTTP QoE

Subjective video quality assessment [161] is important for better understanding human video perception and validating better objective models. Many databases have been designed towards advancing progress on the more general problem of video quality [18, 44, 60, 100, 154] and streaming [27, 38, 45, 58, 83, 118, 142, 145, 149–151, 168]. There are also two valuable survey papers in the field in [50, 135]. Here we give only a brief overview to elucidate important shortcomings in previous studies, and to suggest potentially important improvements.

The time-varying quality of long HTTP streams was investigated in [38], without considering rebuffering events and/or client adaptation strategies. A crowdsourcing experimental comparison among three representative HTTP-based clients was carried out in [118], but only one video content was used and the time-varying QoE was not investigated. In [45], the effects of rebuffering and quality changes were jointly considered, but the videos used were of short duration and the generated distortions did not simulate any actual client adaptation strategy. A simplistic buffer and network approach was

derived in [27] (also introduced in Chapter 2) to study the trade-offs between compression and rebuffering; but only eight distortions were generated and the database is not available in its entirety. Further, it was quite common for the aforementioned experiments to use a fixed bitrate ladder, without considering content-aware encoding strategies which are gaining popularity.

To summarize the main shortcomings of these previous studies, we think that they do not study all of the multiple important dimensions in the client adaptation space, they do not use actual network measurements or a buffer occupancy model to depict an actual streaming scenario, they are small or moderate in size, and they are not always publicly available. Here we present our efforts on advancing the field of **perceptually-optimized** adaptive video streaming by designing a new and unique QoE database, whereby perceptual video quality principles are injected into various stages of a modern streaming system: encoding, quality monitoring and client adaptation. We hope that these efforts will help pave the way for the development of optimized perceptual streaming systems in video streaming industry.

6.2.2 Objective Models for QoE Prediction

The design of optimized user QoE protocols requires accurate QoE prediction models [25, 31, 38, 45, 86, 97, 117, 123, 124, 163]. These models either predict continuous-time QoE or retrospective (after the viewing session ends) QoE. To form these predictions, video quality information is combined with other QoE indicators, such as the location and duration of rebuffering events,

the effects of user memory, the effect of quality switching, and other factors. In most cases, video quality is equated with the video encoding bitrate, the quantization parameter (QP), or is measured using perceptually-designed objective video quality models [11, 120, 159]. Predicting QoE in actual streaming scenarios is a very challenging proposition; to be able to train better algorithms and evaluate existing ones, diverse and detailed video databases are needed.

6.2.3 Client-based Adaptation Algorithms

The design space of adaptation algorithms is very large, but client-based Adaptive Bitrate (ABR) strategies can be broadly classified as: throughput-based [65, 85, 148], buffer-based [30, 61, 91, 144] and hybrid/control-theoretical approaches [41, 82, 88, 157, 169, 172]. Throughput-based approaches rely on TCP throughput estimates to select subsequent rate chunks, while buffer-based approaches use measurements of buffer occupancy to drive these decisions. Hybrid algorithms use both throughput estimates and buffer occupancy, and deploy control-theoretical or stochastic optimal control formulations to maximize user QoE [169]. Recently, raw network observations were also fed to neural networks to achieve adaptive rate selection [89]. An excellent survey of adaptation strategies is found in [71]. Designing QoE-aware adaptation strategies is strongly connected to QoE prediction models, since their optimization goal is to maximize some QoE metric [89, 169]. In turn, subjective experiments are important sources of ground truth data to train and develop better QoE

predictors [27, 45].

6.3 Adaptive Video Streaming Pipeline

6.3.1 Overview

To overcome the limitations of previous QoE studies and train better QoE models, we build our database based on a highly realistic adaptive streaming pipeline model, which comprises four main modules, as shown in Fig. 6.1. The encoding module splits a source video into segments (chunks), determines the per-chunk encoding parameters (encoding resolution and QP) and produces encodes of optimized quality. The video quality module calculates the per-chunk video quality, which drives the encoding and client modules. The network module incorporates the selected network traces, and is responsible for communication between the encoding, video quality and client modules. The client module is responsible for requesting the next chunk to be played. To decide the bitrate/quality level, the client module is aware of its buffer status, and may estimate the future bandwidth (based on past client measurements). Lastly, the playout sequence is generated by concatenating the downloaded encodes, and by adding rebuffered frames when playout pauses. Next, we discuss the network and client modules in greater detail; more details about the encoding module, the video quality module and the overall streaming system can be found in Appendix F.

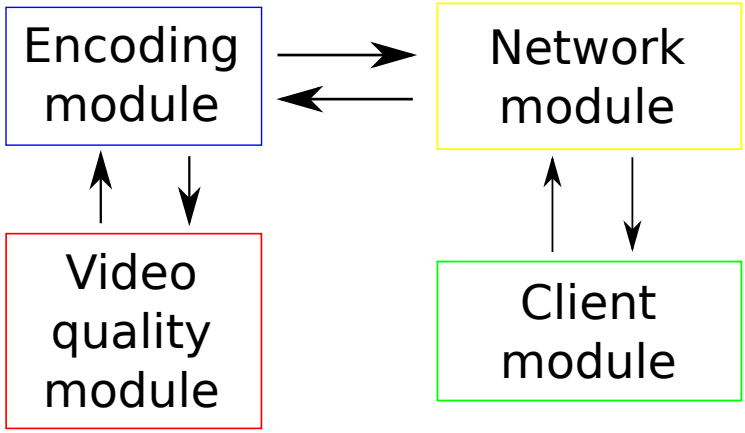

Figure 6.1: Adaptive streaming ecosystem overview.

6.3.2 Network Module

The network module utilizes actual network traces to inform the client regarding the bitrate/quality levels for each segment in the chunk map. In practice, this can be implemented as part of the manifest exchange between the server and the client. To capture the effects of network variability, we manually selected 7 network traces from the HSDPA dataset [6, 122], which contains actual 3G traces collected from multiple travel routes in Norway, using various means of transportation, including car, tram and train, together with different network conditions. This dataset has been widely used to compare adaptation algorithms and is suitable for modeling challenging low-bandwidth network conditions.

We have summarized the characteristics of the selected network traces in Table 6.2. As shown in Fig. 6.2, the selected traces are approximately 40 seconds in duration and have varying network behaviors. For example, the

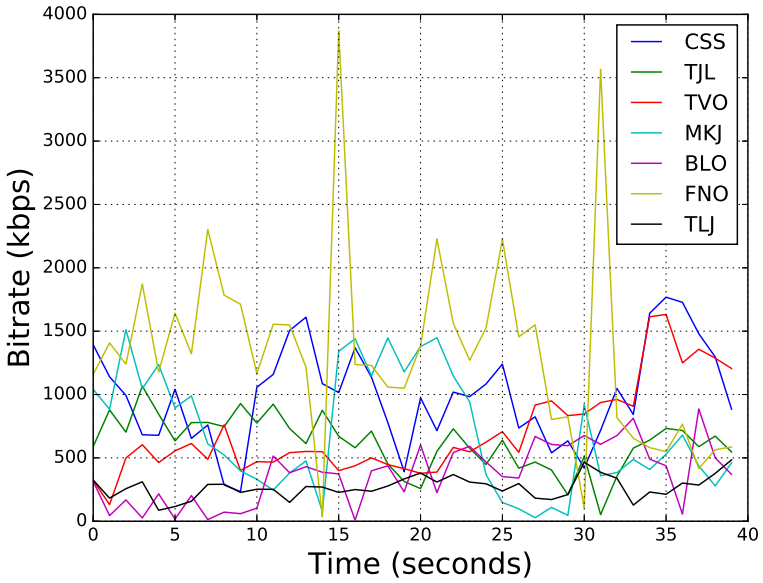

Figure 6.2: Network traces used in our streaming pipeline.

TLJ trace has the lowest average bandwidth but does not vary much over time, while the MKJ trace has a much more volatile behavior compared to TLJ. It may be observed that the network traces densely cover download speeds up to 1Mbps, and there are also samples falling within the 1Mbps-3Mbps range.

6.3.3 Client Module

The client module is responsible for monitoring and updating the buffer status and deciding the next chunk to be played. To marginalize the effects of pre-buffering on our analysis, the client module pre-fetched $B_0 = 1$ chunk for each generated playout sequence. This pre-fetched chunk was always encoded at the lowest bitrate/quality. To perform client adaptation, we decided to

Table 6.2: Summary of the network traces used in LIVE-NFLX-II. The available bandwidth B is reported in kbps. We denote by min B, max B, μ_B and σ_B the minimum, maximum, average and standard deviation of the available bandwidth.

ID	Type	min B	max B	μ_B	σ_B	From	To
CSS	Car	234	1768	989	380	Snaroya	Smestad
TJL	Tram	52	1067	617	207	Jernbanetorget	Ljabru
TVO	Train	131	1632	702	349	Vestby	Oslo
MKJ	Metro	28	1511	696	456	Kalbakken	Jernbanetorget
BLO	Bus	9	886	373	235	Ljansbakken	Oslo
FNO	Ferry	35	3869	1325	761	Nesoddtangen	Oslo
TLJ	Tram	86	485	269	86	Ljabru	Jernbanetorget

implement four adaptation algorithms that are representative of the very large design space of adaptation algorithms. Each of these four algorithms focuses on different design aspects, such as preserving the buffer status, maximizing the download bitrate, or mediating between chunk quality and buffer level. Table 6.3 defines some of the acronyms used hereafter.

We implemented the buffer-based (BB) approach from [61], which decides the rate of the next chunk to be played, as a function of the current buffer occupancy. For BB, a reservoir of $r = 5$ sec. and a cushion of $c = 4.5$ sec. was used. The advantage of the BB approach is that it can reduce the amount of rebuffering by only accessing buffer occupancy. Viewing adaptation from a different perspective, we also implemented a rate-based (RB) approach which selects the maximum possible bitrate such that, based on the estimated throughput, the downloaded chunk will not deplete the buffer. To estimate the future throughput, an average of $w = 5$ past chunks is computed. Intuitively, selecting w can affect the adaptation performance, if the network varies significantly. A low w could be insufficient to make a reliable band-

width estimation, while a very large w might include redundant past samples and have a diminishing return effect. Another downside of the RB approach is that, when channel bandwidth varies significantly, it may lead to excessive rebuffering and aggressive bitrate/quality switching.

Using the video bitrate as a proxy for quality may yield sub-optimal results; a complex scene (rich in spatial textures or motion) requires more bits to be encoded at the same quality as compared to a static scene having a uniform background and low motion. Therefore, it is interesting to explore how a quality-based (QB) adaptation algorithm will correlate against subjective scores. We relied on the dynamic programming consistent-quality adaptation algorithm presented in [81]. The main idea is to use a video quality model (such as VMAF) as a utility function to be maximized within a finite horizon h (in sec.). This utility maximization is formulated as a dynamic programming (DP) problem solved at each step, which determines the stream to be played next.

In our QB implementation, the network conditions are not explicitly modeled. Instead we assume that the future throughput (within the horizon h) will be equal to the average throughput over the past $w = 5$ chunks. For the QB client, two practical limitations on the buffer size are imposed. To reduce the risk of rebuffering, the QB solution requires that the buffer is never drained below a lower bound B_l (in sec.). Also, due to physical memory limitations, QB never fills the buffer above a threshold B_h. To ensure that the B_l and B_h constraints are satisfied, the QB solution is set to converge to a target buffer

$B_t \in (B_l, B_h)$ by imposing in its DP formulation that the buffer at the end of the time horizon has to be equal to B_t. Notably, if the dynamic programming solution fails (when B_l cannot be achieved or B_h is surpassed), then QB selects the lowest (or respectively the highest) quality stream. This is also known as "fallback" mode.

It is impossible for any adaptation strategy to have perfect knowledge of future network conditions. In practice, probabilistic network modeling, or other, much simpler estimation techniques can be derived. For the latter, many adaptation algorithms assume that the network conditions are constant over short time scales, and apply filtering based on previous network measurements, as in QB. Since accurate knowledge of the future bandwidth places an upper bound on the performance of an algorithm, we also included a version of QB which uses the actual network traces, instead of throughput estimates, thereby acting as an "oracle" (OQB).

Table 6.3: Acronym definition table.

Acronym	Definition	Measured in	Value	Used in
BB	buffer-based adaptor	-	-	-
RB	rate-based adaptor	-	-	-
QB	quality-based adaptor	-	-	-
OQB	oracle quality-based adaptor	-	-	-
B_0	pre-fetched video data	# chunks	1	BB, RB, QB, OQB
B_l	min allowed buffer size	sec.	1	QB, OQB
B_h	max allowed buffer size	sec.	10	QB, OQB
T_a	actual throughput	kbps	varies	BB, RB, QB, OQB
h	horizon	sec.	10	QB, OQB
B_t	target buffer	sec.	2	QB, OQB
r	reservoir for BB	sec.	5	BB
c	cushion for BB	sec.	4.5	BB
w	window for throughput estimation	# chunks	5	RB, QB, OQB

We believe that this end-to-end streaming pipeline greatly resembles modern (and future) video streaming architectures. Therefore, we can use it to achieve our main goal: design a large subjective experiment to study streaming user experiences and train QoE models. In the next sections, we discuss the database we developed and our findings in greater detail.

6.4 Subjective Experiment

In this Section, we explain the subjective experiment we carried out to design the LIVE-NFLX-II Streaming Video database.

6.4.1 Video Contents in LIVE-NFLX-II

We collected subjective scores on the 15 video contents shown in Fig. 6.3. The selected contents span a diverse set of content genres, including action, documentary, sports, animation and video games. Notably, the videos also contain computer-generated content, such as Blender [7] animation and video games. The video sources were shot/rendered under different lighting conditions ranging from bright scenes (Skateboarding) to darker scenes (Chimera1102353). There were different types of camera motion, including static (e.g. Asian Fusion and Meridian Conversation) and complex scenes taken with a moving camera, with panning and zooming (e.g. Soccer and Skateboarding). We summarize some of the content characteristics in Table 6.4.

It should be noted that the video sequences are approximately 25 sec-

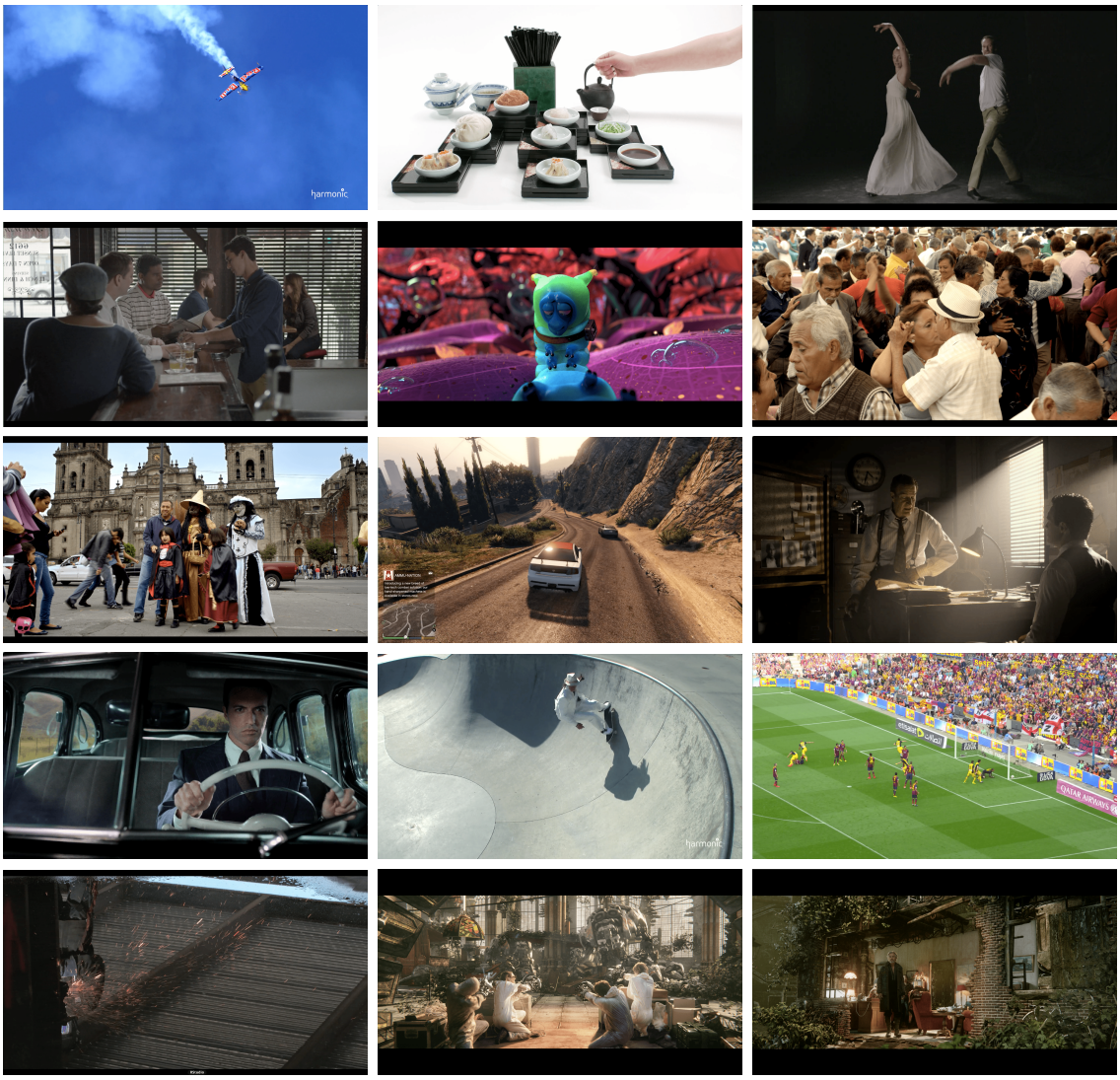

Figure 6.3: Example video frames of each of the 15 video contents in LIVE-NFLX-II (left to right and top to bottom): AirShow, AsianFusion, Chimera1102353, Chimera1102347, CosmosLaundromat, ElFuenteDance, ElFuenteMask, GTA, MeridianConversation, MeridianDriving, Skateboarding, Soccer, Sparks, TearsOfSteelRobot, TearsOfSteelStatic.

Table 6.4: Content characteristics of the video contents in LIVE-NFLX-II.

Video Source	ID	Description
AirShow	AS	Camera tracks object of interest, blue sky background.
AsianFusion	AF	Static camera, zoom-in, uniform background.
Chimera1102353	CD	Static camera on a dark background, medium motion.
Chimera1102347	CF	Multiple human faces, zoom-in, low motion.
CosmosLaundromat	CL	Blender animation, saliency, low motion, camera panning or static
ElFuenteDance	ED	Rich spatial activity, multiple human faces
ElFuenteMask	EM	Medium spatial activity, saliency
GTA	GTA	Gaming content, fast motion
MeridianConversation	MC	Low-light, human face, low motion, static camera
MeridianDriving	MD	Camera zoom-in, face close-up, low motion, human face
SkateBoarding	SB	Fast motion, complex camera motion, saliency
Soccer	SO	Fast moving camera, rich spatial and temporal activity.
Sparks	SP	Slow camera motion, human face, fire sparks, water
TearsOfSteelRobot	TR	Fast motion, multiple moving objects, complex camera motion
TearsOfSteelStatic	TS	Static camera, human close up, low motion

onds long and typically contain multiple scene cuts. This design choice is different from commonly used single-scene 10 second test videos, which are widely used in video quality testing. For video streaming applications, we found it more appropriate to use longer video contents with multiple scene cuts, for a number of reasons. Video streaming viewers tend to watch video content that is many minutes long, while the network conditions may vary considerably throughout a streaming session. Having multiple scene changes also allows us to better exploit the DO encoding approach, which leverages the different scene complexities.

6.4.2 Subjective Testing Procedure

A single-stimulus continuous quality evaluation study [63] was carried out over a period of four weeks at The University of Texas at Austin's LIVE subjective testing lab. We collected retrospective and continuous-time QoE scores on a 1080p 16:9 computer monitor from a total of 65 subjects. Retrospective scores reflect the overall QoE after viewing each video sequence in entirety, while continuous scores capture the time-varying nature of QoE due to quality changes and stalling.

Given the large number of videos to be evaluated and necessary constraints on the duration of a subjective study, we showed only a portion of the distorted videos to each subject via a round-robin approach as follows. Each subject viewed all 15 contents, but only 10 distorted (2 adaptors and 5 network traces) videos per content. We assumed the sequence of adaptors BB,

RB, QB and OQB and network traces 0 to 7, then assigned them to subjects in a circular fashion. For example, if subject i was assigned to adaptors BB and RB and network traces 0 to 4, then subject $i+1$ was assigned to adaptors RB and QB and traces 1 to 5. This led to a slightly uneven distribution of subjects viewing each distorted video, but we considered this to have a minor effect. The benefit of a round robin approach compared to a random assignment is that we can have guaranteed coverage for all traces and adaptors.

To avoid user fatigue, the study was divided into three separate 30-minute viewing sessions of 50 videos each (150 videos in total per subject). Each session was conducted at least 24 hours apart to minimize subject fatigue [63]. To minimize memory effects, we ensured that within each group of 7 displayed videos, each content was not displayed more than once. We used the Snellen visual acuity test and ensured that all participants had normal or corrected-to-normal vision. In total, the final database consists of 420 distorted videos (15 contents, 7 network traces and 4 adaptation strategies) and an average of 23.2 subjects viewed every distorted video. No video was viewed by less than 22 subjects, ensuring a sufficient number of scores per video. Overall, we gathered 65*150 = 9750 retrospective scores and 9750 continuous-time waveforms to study subjective QoE. It should be noted that the number of subjects and distorted videos in the database is significantly larger than many other subjective databases.

To design the experimental interface, we relied on Psychopy, a Python-based software [108]. Psychopy makes it possible to generate and display visual

stimuli with high precision, which is very important when collecting continuous, per-frame subjective data. To facilitate video quality research, we make our subjective experiment interface publicly available at https://github.com/christosbampis/Psychopy_Software_Demo_LIVE_NFLX_II.

Following data collection, we applied z-score normalization per subject and per session [63] to account for subjective differences when using the rating scale. To reliably calculate the retrospective Mean Opinion Score (MOS), we applied subject rejection on the z-scored values according to [63]. After subject rejection, we found that the retrospective scores were in high agreement, exhibiting a between-group (splitting the scores per video into two groups and correlating) Spearman's Rank Order Correlation Coefficient of 0.96. For the continuous scores, we simply applied mean temporal pooling. While more advanced subject rejection techniques could have been used as in [27], we found that the average (across subjects) continuous-time scores did not significantly change after rejection.

6.5 Objective Analysis of LIVE-NFLX-II

We now discuss some properties we have observed of the LIVE-NFLX-II database by analyzing the multiple dimensions of the design space, such as the adaptation algorithms, the network traces and the video contents used.

6.5.1 Video Content Analysis

Besides user content preferences, an important consideration when streaming a particular video content is its encoding complexity. One approach to describe content is the spatial and temporal activity (SI-TI) plot [161], but we were inclined to use a description that more closely relates to the encoding behavior of each content. Therefore, we decided to use content complexity, i.e., the number of encoding bits per source content, as an alternate description. We expect that contents with larger motions and high spatial activity (textures) to be harder to compress, hence subjective scores will generally be lower for those contents for a fixed number of available bits. To measure content encoding complexity, we generated one-pass, fixed constant rate factor = 23 (CRF) 1920x1080 encodes using libx264, then measured the encoding bitrate (see Fig. 6.4). It is clear that there is a large variety of content complexities ranging from low motion contents, such as MeridianConversation or Chimera1102353, medium motion and/or richer textures such as in Skateboarding or ElFuenteMask and high motion and spatial activity as in the Soccer and GTA scenes.

These encoding complexities had a direct effect on the video segments that were played out on the client side. Figure 6.5 shows that contents having low complexities, such as MC, CF and CD, were delivered with better VMAF values. By contrast, challenging content, like GTA and Soccer (SO), were streamed at significantly lower quality. This reveals the importance of content-driven encoding on the server and the potential of content-aware streaming

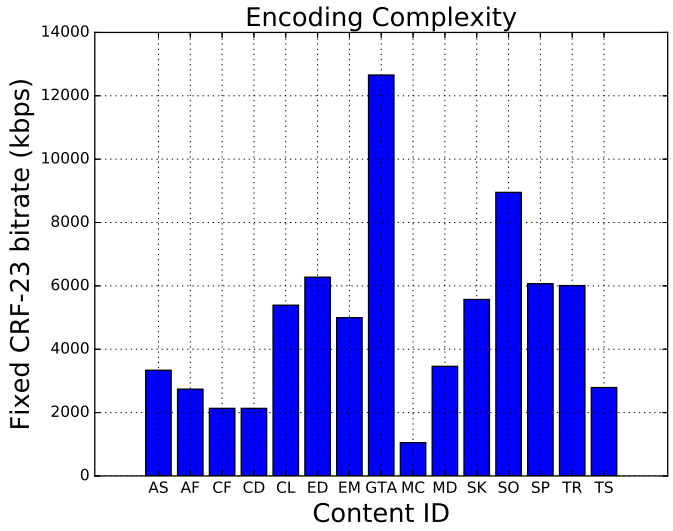

Figure 6.4: Content (encoding) complexity for the 15 contents in LIVE-NFLX-II.

strategies, where streaming parameters are customized to the video content streamed by each client.

6.5.2 Network Condition Analysis

Besides video content, we also introduced various network traces, which can influence the streamed video segments. We collected measurements of the playout bitrate, averaged it over each second (and across contents and adaptors) and show its dynamic per network condition evolution (average and 95% confidence intervals) in Fig. 6.6. Note that after approximately 25 seconds, the confidence intervals became larger, because fewer samples were available (only videos that experienced rebuffering had longer than 25 sec. duration).

As expected, better network conditions overall (FNO and CSS) achieved

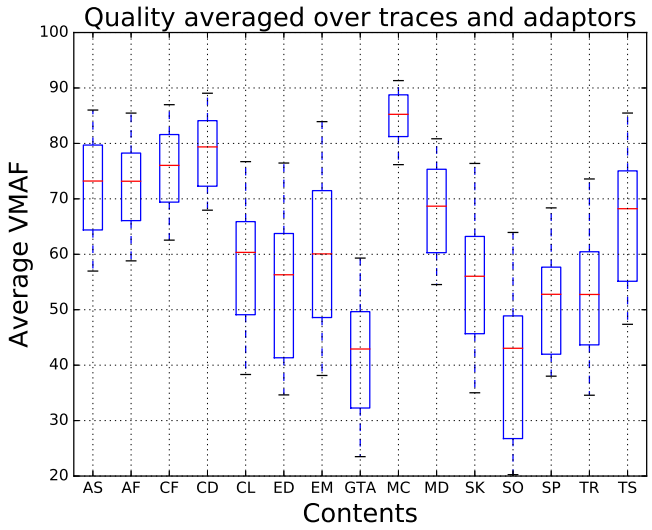

Figure 6.5: Averaged (over segments, traces and adaptors) VMAF values of the 15 contents in LIVE-NFLX-II. The rebuffering intervals were not taken into consideration when making this plot.

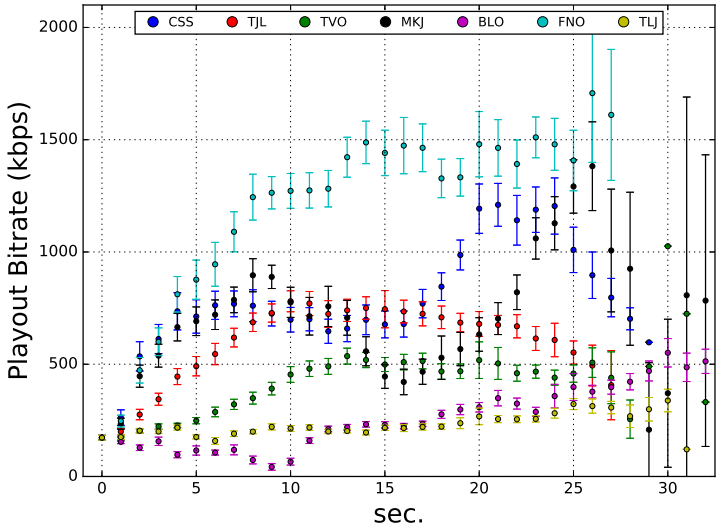

Figure 6.6: Playout bitrate over time across different network traces. To capture the effects of the rebuffering intervals, a value of 0 is used for the video bitrate during those time instants.

better playout bitrates when compared to low-bandwidth cases, as in TLJ and BLO. It is interesting to observe that volatile traces, such as MKJ and CSS, led to significant differences in bitrate, but this was not the case for FNO. Since FNO provides a better network on average than MKJ and CSS, the video buffer was sufficiently filled to account for sudden drops.

6.5.3 Adaptation Algorithm Analysis

Given an encoding chunk map and some time-varying throughput, the client's algorithm makes the ultimate decision on the playback characteristics. To study adaptation behavior, we first collected key characteristics (e.g. number of rebuffers) for all distorted videos generated by each adaptation algorithm. Table 6.5 shows that the OQB adaptor delivered the lowest amount of rebuffering and the lowest average between-chunk VMAF difference. This is to be expected since perfect knowledge of future network conditions would significantly improve the behavior of any adaptation strategy.

By contrast, the RB adaptor led to the largest amount of rebuffering, since it aggressively chooses the chunk rate, it is myopic (does not look ahead in time) and does not take into account the buffer status. The more conservative BB reduces the amount of rebuffering as compared to RB and QB, and has the least number of quality switches. Nevertheless, given that it does not explicitly seek to maximize bitrate, it delivers the lowest playout bitrate. Between RB and BB, QB offers a better tradeoff between playout bitrate and rebuffering. These results are not very surprising: maximizing quality/bitrate or avoiding

rebuffering are conflicting goals, and hence, designing adaptation algorithms should focus more on jointly capturing these factors, as in the case of QB.

At this point, let us take a step back and consider why OQB, despite knowing the entire trace, also suffers from rebuffering. In fact, by setting the maximum buffer $B_h = 10$ sec., and $h = 10$ sec. the dynamic programming solution may fail to return an optimal solution. We found that by increasing B_h and h, both OQB and QB led to significantly reduced rebuffering, which would make the number of rebuffers in the database significantly smaller for subjective analysis. Therefore, our selected parameters mediate between data diversity and meaningful adaptation behavior.

Table 6.5: Objective comparison between adaptation algorithms. Each attribute is averaged over all 105 videos (15 contents and 7 traces) per adaptor. The bitrate values are imputed with a value of 0 during rebuffering intervals, while the VMAF values are calculated only on playback frames

Description	BB	RB	OQB	QB
# switches	5.91	7.08	8.13	8.45
bitrate (kbps)	535	543	660	636
# rebuffers	0.75	1.57	0.70	0.99
rebuffer time (sec.)	1.02	1.35	0.79	1.14
per chunk avg. VMAF	58.05	62.58	64.52	63.19
per chunk VMAF diff.	9.67	7.51	6.89	8.59

.

Let us now ask a different question: what is the streaming behavior of each adaptation algorithm over time? As before, we measure the per second playout bitrate and buffer level, and show the per adaptation evolution in Fig. 6.7. In terms of bitrate, RB aggressively starts off for the first few seconds, but then tends to have have a lower bitrate than both quality-based adaptors.

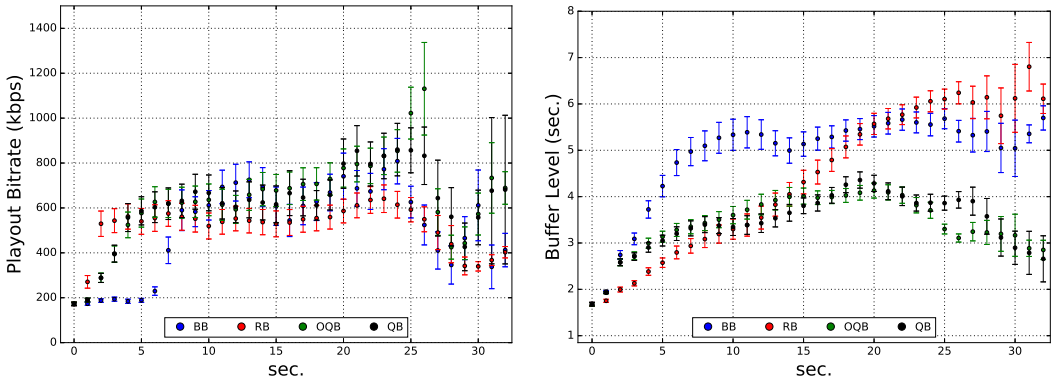

Figure 6.7: Playout bitrate and buffer level over time for different adaptation algorithms (averaged across traces and contents). To capture the effects of the rebuffering intervals, a value of 0 is used for the video bitrate during those time instants.

By contrast, BB is the most conservative strategy in terms of bitrate, while QB and OQB deliver start-up bitrates in between RB and BB. However, after about 15 seconds, QB and OQB consistently deliver higher bitrates. Notably, when a video is significantly longer than 25 sec., this is due to harsher network conditions which lead to rebuffering. Therefore, bitrate decreases and the buffer level over these time intervals ($t >= 25$ sec.) decreases or stays the same.

In terms of buffer level, BB quickly stabilizes its buffer level, while the other adaptors are significantly slower. For RB, aggressive quality switching together with network volatility leads to initial rebuffering and slows down buffer build-up. Nevertheless, after enough time elapses, the RB buffer level increases and even surpasses the BB one, due to the fact that there is enough buffer to avoid rebuffering. In the case of QB and OQB, both adaptors try to

reach the target buffer $B_t = 2$, while OQB (being an oracle) better succeeds at doing so.

It is important to observe that the RB and BB adaptors do not specify a maximum or a target buffer size and hence this might complicate a direct comparison between them and QB or OQB. With regards to the maximum buffer size, RB or BB do not generally reach a value close to $B_h = 10$ sec. when applied without any such constraint. As a result, it is the target buffer of QB/OQB that is actually making our comparisons harder. Nevertheless, it is not very clear as to how these adaptors can be modified to produce a specific target buffer.

Until now, we have identified the main differences between adaptation algorithms. Nevertheless, we have also found a very important similarity: rebuffering events tend to occur earlier in the video playout. To demonstrate this, we calculated the rebuffering ratio of each adaptor over time, i.e., the average rebuffering rate incurred by an adaptor throughout the playout. Figure 6.8 shows that all adaptors have significantly higher rebuffering ratios early on, since the buffer is not yet filled. This is also related to the fact that we only fetch one chunk before starting the playout (see Appendix F, Section F.3). Between adaptors, there are, of course differences too: RB rebuffers even earlier, since it does not take into account the buffer level. Between QB and OQB, the difference is that QB can lead to rebuffering much later in the video, while OQB, which is aware of the entire network trace, is able to minimize rebuffering events from occurring at a later time during the playout.

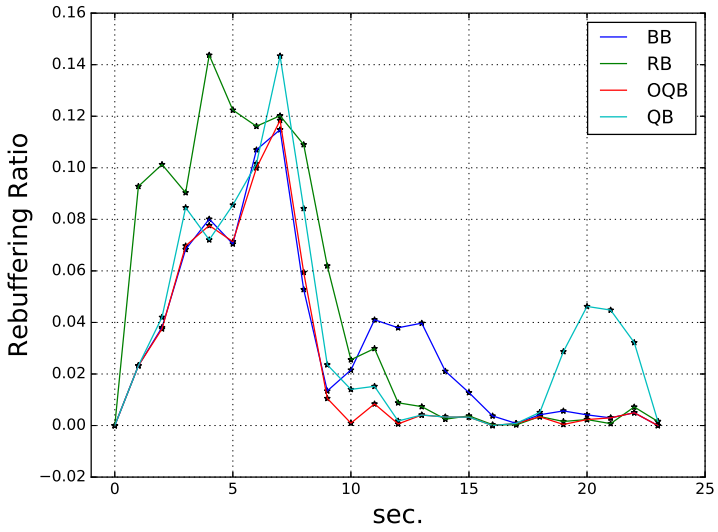

Figure 6.8: Rebuffering location for different adaptation algorithms (averaged across traces and contents). The location is normalized with respect to the original video duration.

6.6 Human Opinion Score Analysis

Up to this point, we have studied the behavior of different network traces and adaptors with respect to some QoE-related factors. Nevertheless, in streaming applications, human opinion scores serve as the ground truth when analyzing streaming video impairments and when evaluating objective models of video quality and QoE prediction. Here we analyze the video database by means of the collected retrospective and continuous-time subjective scores.

6.6.1 Analysis Using Retrospective Scores

To identify the main QoE factors, Fig. 6.9 highlights the relationships between retrospective scores and average VMAF values (calculated on

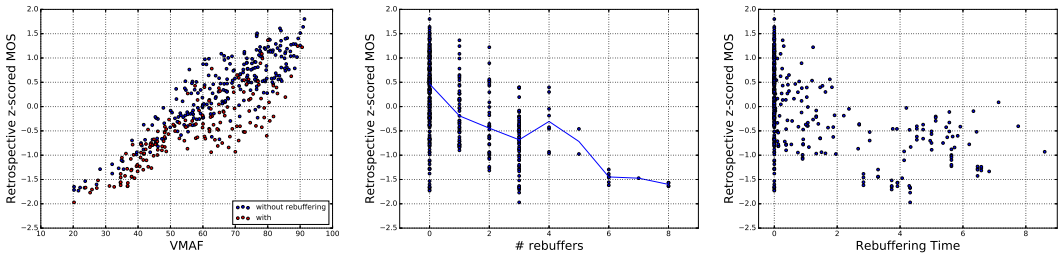

Figure 6.9: VMAF measurements, number and duration of rebuffer events against retrospective opinion scores in LIVE-NFLX-II. Around 40% of the videos have at least one rebuffering event.

non-rebuffered frames), and the number and duration of rebuffering events respectively. Unsurprisingly, the VMAF performance was lower in the presence of rebuffering (the red points negatively impact the overall correlation), since it does not account for its effects on user experience. In Section 6.7, we show how QoE prediction models based on VMAF can deliver improved performance. Meanwhile, it can be seen that a larger number of rebuffering events tends to decrease user experience, but as the number of events becomes larger than 4, there are fewer points to reach the same conclusion with statistical significance. On the rightmost part of Fig. 6.9, we observe that a longer rebuffering time also lowers QoE, but when the rebuffering time is more than 4 seconds, duration neglect effects [56] may reduce this effect. According to the duration neglect phenomenon, subjects may recall the duration of an impairment, but they tend to be insensitive to its duration when making retrospective QoE evaluations.

As in the previous section, we compared the retrospective opinion scores among different adaptors (Fig. 6.10). We observed that the opinion scores are

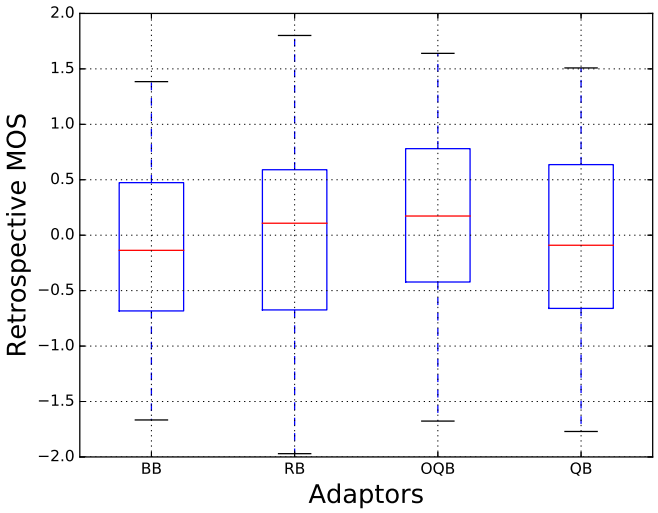

Figure 6.10: Retrospective opinion score distribution for different adaptation algorithms (averaged across traces and contents).

not very different across adaptors. This may be due to the fact that most of the rebuffering events occurred early in the video playout (as shown in Fig. 6.8), and because, just before the video finishes playing (and the retrospective score is recorded), the adaptation algorithms have built-up sufficient buffer to better handle bitrate/quality variations, even if the network is varying significantly. Therefore, it is likely that recency effects [27, 56] led to biases in the retrospective evaluations. Meanwhile, the per adaptor differences in terms of the average VMAF measurements are not considerably different (see Table 6.5) and hence the retrospective scores are also similar across adaptors. In the next section, we investigate the effects on the time-varying QoE.

6.6.2 Analysis Using Continuous Scores

Following our per-second objective analysis in Section 6.5, Fig. 6.11 depicts the continuous-time user experience across adaptation strategies. We found that, within the first few seconds, the RB aggressive rate strategy initially leads to better QoE, unlike BB, QB and OQB, which opt for buffer build-up. Within the first 12 seconds, BB is overly conservative and delivers the lowest QoE among all adaptors, while QB and OQB perform between RB and BB. Nevertheless, after 12 seconds, QB and OQB improve considerably, with OQB tending to produce higher scores for the rest of the session. BB is relatively lower than RB and QB, both of which are statistically close. As before, we note that, after 25 seconds, QoE measurements are decreasing and have larger confidence intervals, since they correspond to videos that rebuffered, and their number decreases over time.

Viewed from the network condition perspective, we found that continuous-time subjective scores are affected by dynamic quality/resolution changes and rebuffering. Figure 6.12 shows that, for all traces, a few seconds are needed to build up the video buffer and hence the continuous scores are relatively low. Under better network conditions (e.g. FNO), the user experience steadily improves after some time, due to the adaptors switching to higher resolution and lower QP values. By contrast, challenging cases such as BLO and TLJ recover slowly or do not recover at all, while very volatile conditions, as in MKJ, may also lead to noticeable drops in QoE much later in the video playout. These results, together with the improved performance of OQB, support a

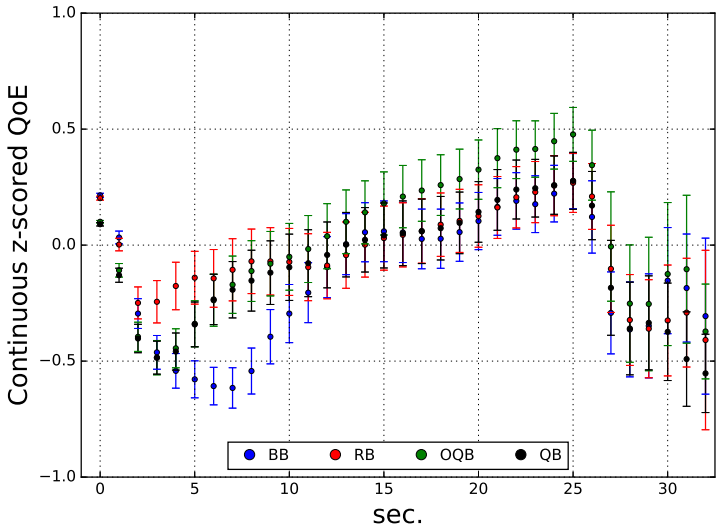

Figure 6.11: Continuous-time scores for different adaptation algorithms (averaged across traces and contents).

well-established but very challenging proposition: better bandwidth prediction is important to achieve higher QoE.

6.6.3 Adaptation Algorithm Performance Discussion

Following our earlier between-adaptor analysis, it is natural to ask which adaptation algorithm performs the best. In terms of retrospective scores, we were not able to make statistically significant comparisons, in part due to the effects of recency. However, using continuous-time scores, we found that OQB performed the best, since it acts as an oracle and has perfect knowledge of the future bandwidth, while BB was overly conservative during startup and did not select high quality streams.

Comparing RB and QB, we found that they delivered similar QoE over

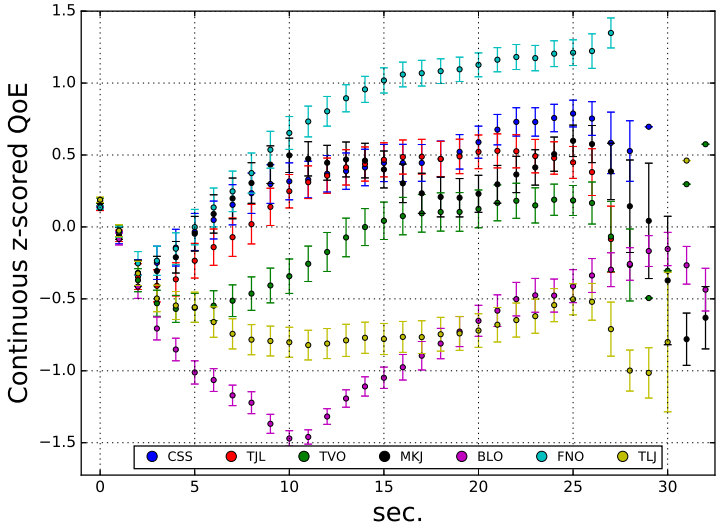

Figure 6.12: Continuous-time scores for different network conditions.

time, except during the start-up phase, where RB picked higher quality levels. The similar behavior between QB and RB can be attributed to their inherent properties: RB leads to excessive rebuffering, while QB reduces rebuffering (by adding the buffer in its optimization scheme), but leads to many quality switches (see also Table 6.5). In fact, an important consideration when designing QB is selection of the minimum buffer B_l and target buffer B_t values. When the network changes rapidly, the adaptor may not satisfy these and use its fallback mode, which leads to such large quality switches.

6.6.4 Limitations of the LIVE-NFLX-II database

Despite our efforts in designing a diverse and realistic database that relies on state-of-the-art ideas in video encoding and streaming, one cannot

overlook a number of limitations. We recognise that QoE is not only affected by the factors investigated herein, such as visual quality, recency, rebuffering or quality switching, but also by other factors such as the audio quality or contextual factors, like the display device and user expectations regarding the streaming service and/or the viewing environment. In our experiment, the audio quality was fixed and the display device was a computer monitor. Nevertheless, given the very large design space of the subjective experiment, it is virtually impossible to vary all of these streaming conditions at the same time. Meanwhile, the adaptation algorithm design space and the number of possible network conditions are immense, hence our experiment can only capture the main characteristics of these dimensions as they pertain to user experience.

6.7 Perceptual Video Quality and Quality of Experience

An important goal of our database design is to use it as a development testbed for video streaming quality and QoE prediction models. In this section, we evaluate a number of representative VQA and QoE prediction models. Given that the database contains both retrospective and continuous-time scores, we studied the performance of these algorithms both for retrospective and continuous-time QoE prediction applications.

To calculate video quality, we decoded each distorted video into YUV420 format and applied each video quality model on the luminance channel of a

distorted video and its reference counterpart. For video content with non-16:9 aspect ratio and before the VQA calculations, we also removed black bars to measure the quality only for active pixels. For videos containing rebuffered frames, we removed all of those frames and calculated video quality on the aligned YUV files [27]. In the next sections, we investigate the predictive performance of leading VQA models and study their predictive performance when they are combined with QoE-driven models for retrospective and continuous-time QoE prediction.

6.7.1 Objective Models for Retrospective QoE Prediction

In our first experiment, we applied several well-known video quality and QoE metrics, including PSNR, PSNRhvs [113], SSIM [159], MS-SSIM [160], ST-RRED [143], VMAF [79] (version 0.6.1), SQI [45] and Video ATLAS [24]. The original Video ATLAS model [24], was designed and tested on the LIVE-NFLX and Waterloo databases (see also Chapter 3), where spatial resolution changes and quality switching events were much less diverse. Given the flexibility of Video ATLAS and the diversity of our newly designed database, we can re-train the model to integrate this kind of information. We used the following features: VMAF as the VQA feature, average absolute difference of encoding resolution (to capture the effects of resolution switching), rebuffer duration, and the time since the lowest stream in the sequence (worst quality) occurred, in seconds. For SQI, VMAF was also used as the VQA model. We excluded the P.1201-3 models [3], since they are trained for video sequences

longer than one minute. To evaluate performance, we use Spearman's Rank Order Correlation Coefficient (SROCC), which measures the monotonicity between groundtruth QoE and predictions. The results of our experiment are shown in Figure 6.13.

Since Video ATLAS is a learning-based model, we split the database into multiple train/test splits. When using image and video quality databases, it is common to split the database into content-independent splits; but for the streaming scenario we propose a different approach. Given that the video contents are pre-encoded and the behavior of an adaptation algorithm is deterministic (given a network trace and a video content), it is more realistic to assume that, during training, we have collected subjective scores on a subset of the network traces. Therefore, we perform our splitting based on the network traces by choosing 5 traces for training and 2 for testing each time, which yields $\binom{7}{2} = 21$ unique combinations of 300 (15 contents, 4 adaptors and 5 traces) training and 120 (15 contents, 4 adaptors and 2 traces) testing videos. The total number of combinations may not be as large; but each train/test subset contains hundred of videos. Figure 6.13 shows boxplots of performance across all 21 iterations for all compared models.

It can be observed that all of the VQA-only models lacked in performance, which is to be expected since VQA models only capture visual quality and disregard other critical QoE aspects such as rebuffering. Nevertheless, VMAF 0.6.1 performed significantly better than all other models. As a reminder, VMAF was used on multiple occasions when generating the

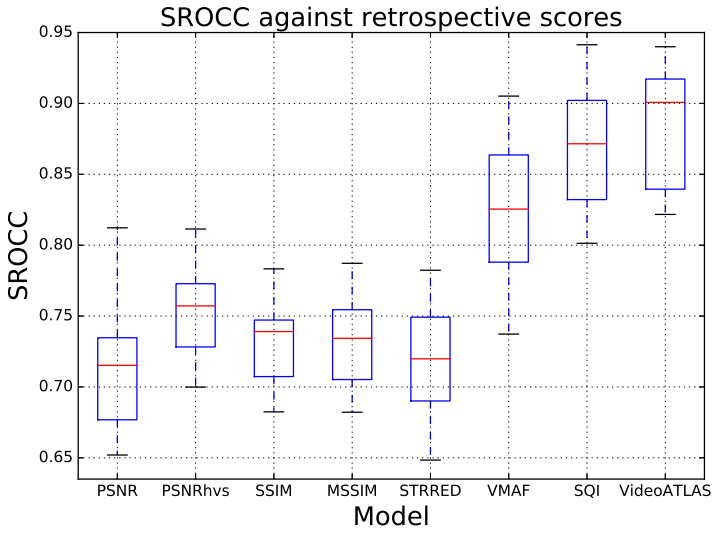

Figure 6.13: Boxplots of SROCC performance of leading VQA and QoE models using retrospective scores.

final videos, e.g., when generating the bitrate ladder, deciding on the encoding parameters and performing client-based adaptation for QB and OQB. This suggests that our system is better tuned towards the VMAF model and that the choice of the quality model has a direct impact on user experience. Using VMAF as part of the SQI and Video ATLAS QoE predictors led to significant performance gains in both cases.

6.7.2 Objective Models for Continuous-time QoE Prediction

Predicting continuous-time QoE is a harder task, given the challenges in collecting reliable ground truth data and designing models that can integrate perceptually-motivated properties into a time series prediction. Earlier approaches [38] have addressed the problem of predicting time-varying quality

and only recently similar works have addressed time-varying QoE prediction [26].

We evaluated two prediction algorithms presented in [26], one based on autoregressive neural networks (G-NARX) and the other based on recurrent neural networks (G-RNN). We note that using the SQI model in [45] to predict continuous-time QoE does not deliver scores that fall within the z-scored continuous MOS scale, since it is not trained on subjective data. Therefore, the RMSE and OR values are considerably worse. Further, using the SROCC as an evaluation metric did not yield satisfying results (around 0.41 of SROCC) and using the SROCC may not be an appropriate choice for comparing between time series [26].

To train the G-NARX and G-RNN models, we used per-frame VMAF measurements as the continuous-time VQA feature. The original network design was used, with 8 input delays and 8 feedback delays for G-NARX and 5 layer delays for G-RNN. Both approaches used 8 hidden nodes and the training process was repeated three times yielding an ensemble of three test predictions per distorted video that were averaged for more reliable time series forecasting. We configured the prediction models to output one value per 0.25 sec., by averaging the continuous-time variables accordingly. To evaluate their performance, we used root mean squared error (RMSE) and outage rate (OR). RMSE measures the prediction's fidelity to the ground truth, while OR measures the frequency of predictions falling outside twice the confidence interval of the subjective scores.

Table 6.6: Prediction performance of the G-NARX and G-RNN QoE models using continuous scores.

Model	RMSE	OR
G-NARX	0.267	7.136%
G-RNN	0.276	5.962%

Table 6.6 shows that both approaches delivered promising performance and similar to each other in terms of RMSE and OR. Nevertheless, we observed cases where the predictions could be further improved, as in Fig. 6.14. In this case, the G-NARX QoE prediction did not accurately capture the subjective trends and their dynamics.

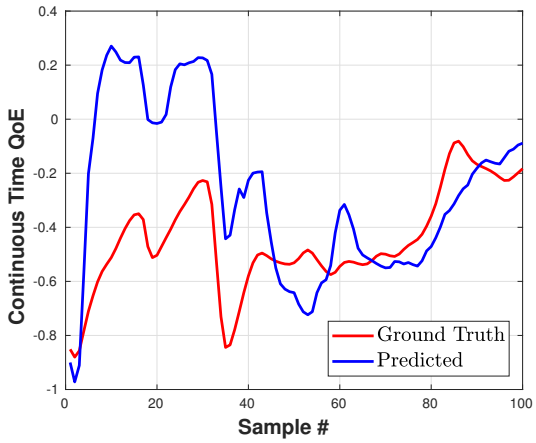

Figure 6.14: An example where the G-NARX QoE prediction does not capture the subjective trends.

Developing accurate QoE prediction models is important for improving client adaptation algorithms, which are fundamentally designed to maximise some QoE metric. For example, the adaptation strategies in [89, 169] optimize a hand-crafted linear combination of average video quality, quality switching

and rebuffering time, rather than deploying a more holistic QoE model.

6.8 Discussion and Conclusion

We presented the design of a large subjective video database, which relied on a highly realistic streaming system. The collected data allowed us to analyze overall and continuous-time user experiences under different network conditions, adaptation algorithms and video contents. Using the collected human opinion scores, we also trained and evaluated predictors of video quality and quality of experience.

In the future, we intend to use the ground truth data to build better continuous-time QoE predictors by integrating additional features, such as resolution changes, network estimates and buffer status. Inspired by similar QoE works as in [66], our ultimate goal is to "close the loop", i.e., inject such QoE-aware predictions into the client-adaptation strategy in order to perceptually optimize video streaming.

Chapter 7

Thesis Conclusion

7.1 Thesis Overview

The main topics of this dissertation were the study of perceptual video quality assessment and quality of experience for adaptive video streaming. In Chapter 2, we discussed the design of the LIVE-NFLX subjective video database, which focuses on the tradeoffs between compression and rebuffering for low bitrate video streaming. To process the continuous time QoE scores, we designed a subject rejection scheme based on dynamic time warping. We also studied other subjective effects, like recency and primacy and demonstrated the need for developing QoE models that jointly consider rebuffering effects, memory and video quality measurements to more accurately predict streaming video QoE.

Based on the collected subjective scores, Chapter 3 detailed the design of Video ATLAS, a retrospective QoE predictor, which combines QoE-aware features, like rebuffering and video quality into a support vector regressor to predict QoE. We evaluated this algorithm under various experimental settings and demonstrated its potential for QoE prediction. Chapter 4 proposed the use of autoregressive and recurrent neural networks and dynamic models

for continuous-time QoE prediction. We showed that the problem of QoE prediction can be formulated as a time series forecasting problem and that ensemble predictions can deliver improved performance on multiple subjective QoE databases.

Since all of these QoE prediction models are based on video quality measurements, it is clear that improving perceptual video quality metrics is a fundamental problem. Chapter 5 addressed the shortcomings of VMAF, a recently developed full reference metric, and discussed two successful improvements of it: ST-VMAF and E-VMAF. Both algorithms rely on extracting entropic differences as features in space or time and they demonstrated state-of-the-art performance across multiple applications, such as perceptual video quality assessment and JND prediction, as well as inputs to more general QoE predictors.

In Chapter 6, we designed LIVE-NFLX-II: a video streaming subjective database based on a perceptually-optimized end-to-end adaptive streaming system. The goal of this database was to model realistic streaming scenarios where the multiple dimensions of client adaptation, such as the video content, the client adaptation algorithm and the network condition, are taken into account. We used the collected data to compare between client adaptation strategies and actual network traces and evaluate video quality predictors, retrospective and continuous-time QoE predictors.

7.2 Conclusion and Future Work

This dissertation covered a number of aspects regarding Quality of Experience for adaptive streaming applications. Nevertheless, this dissertation paves the way for addressing multiple questions which remain to be answered. For example, in video streaming, we usually assume that the source video sequences are of high quality. However, this assumption may be violated when the source content has been upscaled, compressed or afflicted by film grain noise. Some of these artifacts may be more relevant to older (legacy) content, while others may also be added for artistic purposes. In either case, full reference video quality algorithms are not the best option, since they only capture the degradations due to the distortion and do not take into account the original source quality. To address this, it would be interesting to combine full reference and no reference algorithms, as in [170].

Following the development of continuous-time QoE predictors in Chapter 4 and the streaming database presented in Chapter 6, it is also natural to seek better continuous-time features as inputs to these QoE models. As an example, inputs like the video quality or resolution switching could be used to train more accurate predictors. Ultimately, such models can be used for client adaptation that maximizes QoE given a set of buffer constraints, similar to the quality-based adaptation strategies studied in Chapter 6. Such approaches can lead to improved network utilization and improved quality of experience for the end user. I hope that the developments presented herein will pave the way for QoE-aware client adaptation and better video streaming worldwide.

Appendices

Appendix A

Further experiments on the ST-VMAF and E-VMAF VQA models

A.1 Cross-database performance for ST-VMAF and E-VMAF

The proposed VQA models rely on three components: the VMAF+ training subjective data, the spatiotemporal feature integration and the temporal pooling step. In this section, we investigate the effects on the predictive performance of ST-VMAF and E-VMAF when each of these components varies.

First, we investigated the effects on the predictive performance of ST-VMAF when trained on other databases in Table A.1. Importantly, the VMAF+ dataset proved to be highly consistent, and served as an excellent training dataset for ST-VMAF. It is also encouraging that the aggregate SROCC values (for a fixed training dataset) achieved by ST-VMAF were very close to, or significantly exceeded 0.8 (see second to last column in Table A.1). Similar observations apply to the E-VMAF predictions.

Having established that training on VMAF+ is the best option, we studied how the performance of ST-VMAF and E-VMAF compares to that

Table A.1: Cross-database SROCC for ST-VMAF. Each element in this matrix shows the SROCC performance when training on the dataset in the row and testing on the dataset in the column. The last two columns show the aggregate SROCC and PLCC performance per training dataset. Using the VMAF+ dataset for training yielded the best overall performance and is denoted by boldface.

Database	LIVE VQA	LIVE Mobile	CSIQ-VQA	VMAF+	NFLX	SHVC	VQEG-HD3	EPFL	overall SROCC	overall PLCC
LIVE VQA	-	0.869	0.742	0.775	0.859	0.873	0.628	0.842	0.811	0.809
LIVE Mobile	0.584	-	0.736	0.791	0.900	0.891	0.653	0.836	0.794	0.786
CSIQ-VQA	0.599	0.856	-	0.757	0.855	0.879	0.669	0.830	0.795	0.794
VMAF+	0.809	0.905	0.784	-	0.927	0.888	0.932	0.945	**0.897**	**0.898**
NFLX	0.733	0.925	0.754	0.888	-	0.874	0.922	0.947	0.882	0.884
SHVC	0.700	0.893	0.759	0.808	0.866	-	0.887	0.930	0.850	0.846
VQEG-HD3	0.706	0.890	0.732	0.813	0.879	0.822	-	0.933	0.842	0.839
EPFL	0.715	0.931	0.717	0.866	0.918	0.878	0.879	-	0.862	0.859

of the individual models M_1 and M_2, and the performance gains of hysteresis pooling. To this end, we report the results (when training on VMAF+) in Table A.2. It can be observed that M_1 and M_2 deliver similar performances, but afford significant performance gains when combined using E-VMAF. Similarly, ST-VMAF combines some features that may also belong to either M_1 or M_2, but their combination performs significantly better. Hysteresis pooling further improves the predictive performance of both ST-VMAF and E-VMAF.

A.2 Computational Analysis for ST-VMAF and E-VMAF

To deploy VQA models for video quality prediction at global scale, ensuring low time complexity is a critical requirement. Therefore, we studied the per frame compute time consumed by several leading FR-VQA models[1] in

[1]Frame-based models are usually much faster and hence are excluded.

Table A.2: Cross-database Aggregate Performance (training on VMAF+ dataset). The best performance is denoted by boldface.

Database	pooling	SROCC	PLCC
M_1	mean	0.847	0.853
M_2	mean	0.845	0.847
ST-VMAF	mean	0.885	0.887
E-VMAF	mean	0.873	0.875
ST-VMAF	hysteresis	**0.897**	**0.898**
E-VMAF	hysteresis	0.894	0.895

Figure A.1. For our analysis, we selected videos from 6 different resolutions ranging from CIF (352x288) up to Full HD (1920x1080). These videos have 334 frames on average and we averaged our time calculations over 5 trials. All of the compute time analysis was carried out on a 16.04 Ubuntu LTS Intel i7-4790@3.60GHz system.

ST-MAD required the most compute time, followed by ST-RRED and VQM-VFD. When implementing ST-MAD and VQM-VFD, we encountered out-of-memory issues on long Full HD videos. This could be due to the fact that the ST-MAD implementation stores and loads the entire video into memory. Another limitation of these approaches is that they only process entire videos with no capability to produce continuous video quality scores. ST-RRED processes the video frame by frame to produce continuous scores, but requires calculating a complete multi-scale, multi-orientation steerable decomposition.

By contrast, ST-VMAF and E-VMAF are memory efficient, produce

continuous quality scores and consume less compute time, since they extract the very efficient S-SpEED and T-SpEED features. Our ST-VMAF and E-VMAF implementation uses un-optimized Matlab code to extract SpEED-QA features, while VMAF uses AVX optimization and is implemented in C. Since ST-VMAF and E-VMAF are natural extensions within the VMAF ecosystem, it is possible to adopt similar optimization approaches.

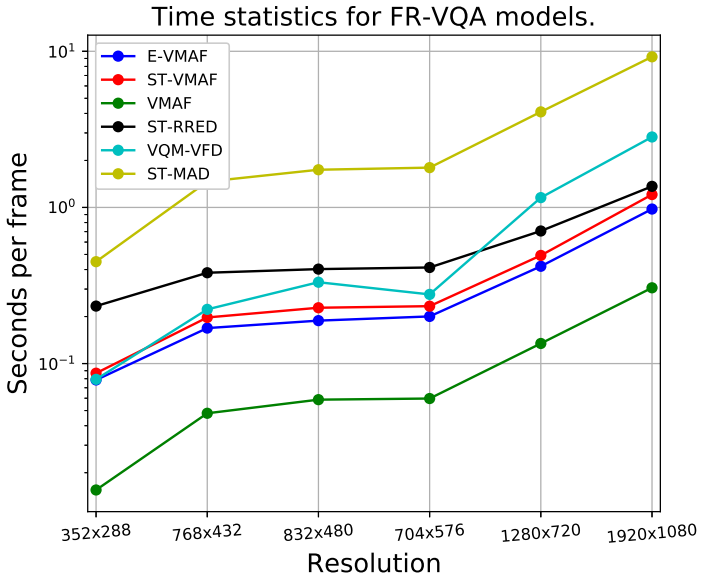

Figure A.1: Per frame compute time required for each FR-VQA model (log vertical scale).

Appendix B

Playout Patterns and Encoding Pipeline in the LIVE-NFLX Video QoE Database

B.1 Explaining the Playout Pattern Parameters

We provide an example of how some of the playout pattern parameters were determined. We fixed the rebuffer duration for pattern #1 to 8 sec. and the average bitrate for the client in pattern #2 to be $R_2 = 160$ kbps. Since there is no rebuffering event in pattern #2 but the available bandwidth is 100 kbps for d seconds, the client in #2 expends all of the available buffer B_0 in d seconds hence $(R_2 - 100)d = B_0$ yielding $B_0 = 1333$ kbits. Let t_b be the time interval after the available bandwidth drops until a rebuffering event occurs in #1. Clearly, $t_b(250 - 100) = B_0$ since the client depletes all of the buffer before the playback interruption. During the rebuffering event, the buffer fills to $B_1 = 800$ kbits in 8 seconds, given the available bandwidth of 100 kbps. The client chooses to start the playback t_a seconds before the available bandwidth recovers hence $t_a(250-100) = B_1$, since we assume that all playout patterns eventually deplete the entire buffer. Therefore, $t_a = 5.3333$ sec. and $d = t_e + 8 + t_a \approx 22.2167$ seconds.

B.2 Implementation Details of the Encoding Pipeline

Each high quality video source sequence is first encoded into H.264 format, combined with a corresponding, synchronized audio stream and placed in an mp4 container without further re-encoding. Then, following the application of a specific network-simulated pattern, the .mp4 file is divided into a number of different chunks, each at a different encoding bitrate. For example, pattern #6, which contains both bitrate changes and a rebuffering event would have three chunks: one for the rebuffering event and two corresponding to the encoded video before and after the rebuffering event.

The encoding pipeline then assembles the segments of the final video, by concatenating them using an encoding profile demarking the interval of time spent at each quality level. The location and duration of each rebuffering event is specified as: enc $<$ start $>$ $<$ stop $>$ $<$ bitrate $>$ stall $<$ start $>$ $<$ duration $>$, with time measured in seconds and bitrate in kbps. The encoding resolution was based on the used bitrate and the encoding profile was set to high.

Using this encoding profile, the encoding process was carried out as follows (see Fig. B.1). First, the source video and audio streams were transferred from Google Drive and stored locally for further encoding. Next, the source video stream (in H.264 format) was decoded, yielding an uncompressed raw .yuv file. The encoding map was then used to split the .yuv file in a frame-accurate manner, yielding .yuv chunks, e.g. three chunks for pattern #6. A two pass encoding step using FFMPEG was then applied to encode the

.yuv files into .mp4 format. For pattern #6, this corresponds to two chunks encoded at 250 kbps, and one encoded at 160 kbps. The final frame of every video chunk that occurs immediately before a rebuffering event was used to generate a "rebuffering video chunk". A familiar "loading icon", (a spinning wheel) was overlaid on that frame during the rebuffering event and animated to simulate the desired video rebuffering effect. After encoding each of the yuv chunks into .mp4 format, all of the .mp4 segments were upscaled to the device resolution (1080p), then concatenated into a single .mp4 file. For playback purposes, each concatenated .mp4 file was lightly compressed using CRF 10, since raw playback on mobile devices is not supported.

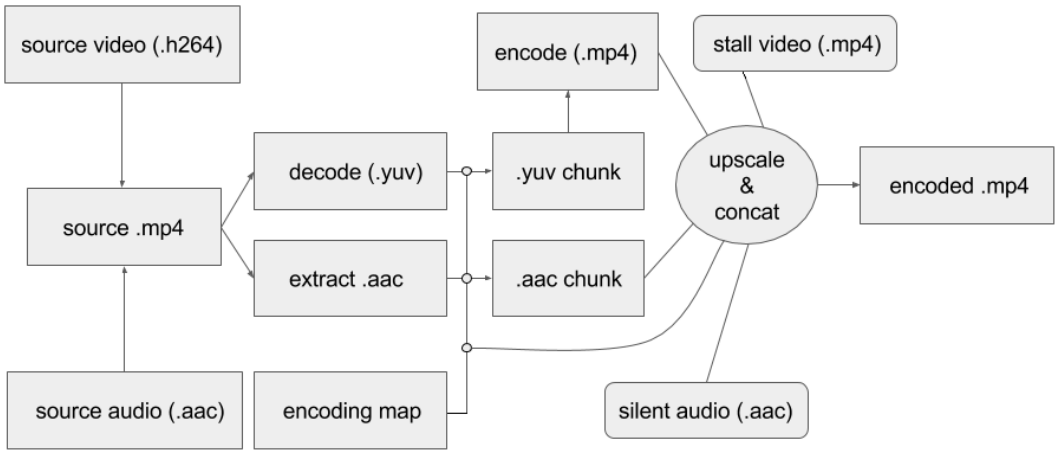

Figure B.1: Encoding pipeline used to create the playout patterns.

Appendix C

Additional Experimental Analysis for Video ATLAS

As already discussed, Video ATLAS is a flexible framework that allows for introducing additional inputs or using other regression models, depending on the application. In this Appendix, we study five different design aspects and their effects on the performance of Video ATLAS: VQA inputs, regression models, feature combination, amount of training data available and pooling strategy for the VQA feature.

C.1 Using Video ATLAS with Different VQA Models and Regressors

First, we investigated the performance of Video ATLAS when we varied both the VQA model and the regressor type using [14]: linear models (Ridge and Lasso regression), Support Vector Regression (SVR) with rbf kernel and ensemble methods such as Random Forest (RF), Gradient Boosting (GB) and Extra Trees (ET) regression. For the ensemble methods, feature normalization was not required, but we preprocessed the features for all regression models by mean subtraction and scaling to unit variance. Note that we computed the data mean and variance in the feature transformation step

using only the training data. For each of the regression models, we determined the best parameters using 10-fold cross validation on the training set. This process was repeated on all possible train/test splits, which were generated as in Experiment 1. We reported our results on Table C.1.

Table C.1: Results on different VQA and regression models. Top: SROCC; Bottom: LCC. We report the median SROCC/LCC before (BR) and after regression. The last column contains the average of the SROCC/LCC values across all quality metrics for each regression model.

VQA	PSNR	SSIM [159]	MS-SSIM [160]	NIQE [94]	VMAF [79]	ST-RRED [143]	GMSD [164]	mean
BR	0.60	0.68	0.68	0.21	0.61	0.68	0.65	0.59
Ridge	0.67	0.78	0.77	0.44	0.62	0.80	0.70	0.68
Lasso	0.65	0.77	0.76	0.44	0.62	0.80	0.70	0.68
SVR	0.62	0.86	0.84	0.52	0.58	0.88	0.70	0.71
ET	0.57	0.87	0.85	0.38	0.57	0.86	0.51	0.66
RF	0.61	0.83	0.82	0.37	0.51	0.82	0.56	0.65
GB	0.55	0.82	0.80	0.38	0.56	0.82	0.53	0.64
VQA	PSNR	SSIM [159]	MS-SSIM [160]	NIQE [94]	VMAF [79]	ST-RRED [143]	GMSD [164]	mean
BR	0.57	0.75	0.73	0.42	0.75	0.75	0.70	0.67
Ridge	0.81	0.88	0.87	0.60	0.80	0.88	0.83	0.81
Lasso	0.83	0.87	0.87	0.60	0.81	0.88	0.84	0.81
SVR	0.79	0.92	0.92	0.69	0.76	0.94	0.81	0.83
ET	0.72	0.93	0.91	0.60	0.74	0.92	0.72	0.79
RF	0.73	0.91	0.90	0.54	0.67	0.91	0.72	0.77
GB	0.72	0.91	0.88	0.61	0.73	0.91	0.72	0.78

We found that the performance was improved when using Video ATLAS for all VQA models and for at least one regressor. For VMAF, PSNR and GMSD the regression result did not improve using every regressor. While it is true that an effective regression scheme has a large positive impact on QoE prediction, not all video quality models are a good choice. For example, we have found that VMAF performs poorly before regression, which could be due to the fact that VMAF is trained on a large screen and on short video sequences. On a similar note, models like PSNR which do not capture

perceptually-relevant information, or NIQE, which does not exploit reference information, will underperform when used in the Video ATLAS framework. We found ST-RRED to be a high-performing and reliable VQA model which yielded the best performance overall (see the last columns of Table C.1).

Regarding the choice of the regressor, we found that the SVR regressor performed best followed by Ridge. The performance of the Ridge and Lasso models was somewhat higher than that of RF and ET, while GB yielded the worst performance across all regression models. It should be noted that these more complex ensemble methods (RF, GB and ET) are more suitable when the number of input features is much larger. This again highlights the merits of the proposed model: it uses a simple and efficient SVR learning engine that matches well with the dimensionality of the QoE prediction problem.

C.2 Investigating the Feature Combinations in Video ATLAS

While our proposed system deploys features that collectively deliver excellent results, it is interesting to analyze the relative feature contributions. One way to study the feature importances is by a tree-based method [36], as follows. First, we picked a sophisticated video quality model (ST-RRED) and the signal fidelity PSNR metric; then applied Random Forest regression. Figure C.1 shows the feature importances after 1000 pre-generated train/test splits.

We observed that the video quality model used plays an important role

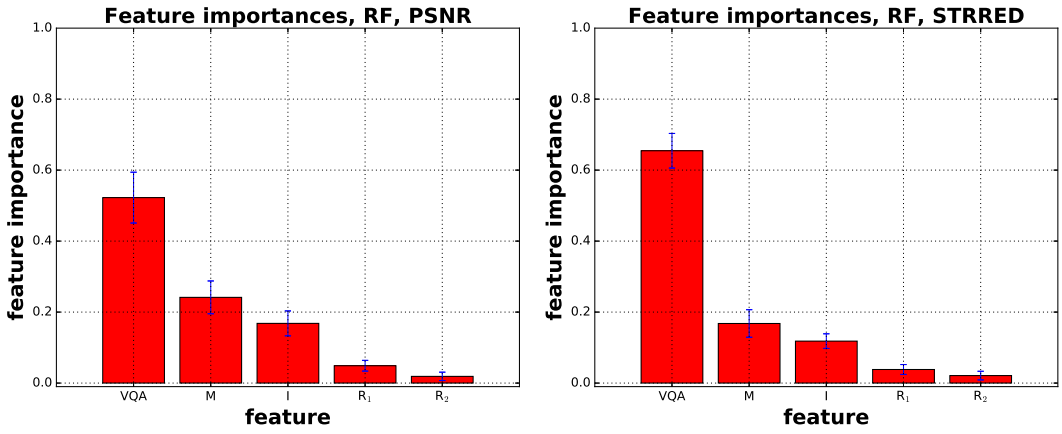

Figure C.1: Feature importances using PSNR (left) and ST-RRED (right) after 364 random train/test splits (Experiment 1) using the Random Forest regressor (RF). Horizontal axis: feature labels; vertical axis: feature importance normalized to 1. A more sophisticated VQA model has a larger VQA feature importance.

in QoE prediction. Nevertheless, when using Video ATLAS with PSNR (a weakly performing VQA model compared to ST-RRED), the feature importance of the VQA input was lower as compared to the VQA feature importance when Video ATLAS uses ST-RRED. The memory feature also makes a strong contribution, since for retrospective QoE evaluation, recent experiences are a strong QoE indicator. The stalling features deliver an important but somewhat smaller contribution. Lastly, the R_1 feature (stalling duration) had a much lower contribution, which may be explained by the duration neglect effect [56]: subjects may remember that a stalling event occurred, but may not be sensitive to its duration.

To further investigate the effects of those feature types on the retrospective QoE prediction task, we experimented further by using different feature

Table C.2: Experiment 1: Results on different feature subsets when ST-RRED was used. Top: SROCC; Bottom: LCC. The feature subsets are indexed as described in the text.

Features	1	2	3	4	5	6	7	8	9	10	11	12
Ridge	0.68	0.23	0.27	0.31	0.66	0.68	0.79	0.41	0.42	0.79	0.78	0.80
Lasso	0.68	0.23	0.27	0.32	0.70	0.70	0.80	0.41	0.40	0.80	0.80	0.80
SVR	0.60	0.38	0.28	0.37	0.72	0.64	0.84	0.42	0.48	0.85	0.85	0.88
ET	0.47	0.30	0.24	0.30	0.73	0.66	0.73	0.31	0.39	0.85	0.74	0.86
RF	0.52	0.39	0.26	0.33	0.77	0.61	0.78	0.41	0.47	0.82	0.77	0.83
GB	0.55	0.39	0.27	0.35	0.76	0.60	0.76	0.44	0.50	0.83	0.75	0.82

Features	1	2	3	4	5	6	7	8	9	10	11	12
Ridge	0.75	0.45	0.31	0.29	0.75	0.73	0.78	0.46	0.63	0.80	0.78	0.88
Lasso	0.75	0.45	0.31	0.30	0.78	0.75	0.81	0.46	0.62	0.82	0.81	0.88
SVR	0.71	0.43	0.32	0.32	0.83	0.66	0.86	0.45	0.69	0.90	0.86	0.94
ET	0.54	0.38	0.31	0.32	0.74	0.69	0.74	0.39	0.62	0.92	0.74	0.92
RF	0.60	0.45	0.35	0.32	0.79	0.68	0.79	0.47	0.71	0.90	0.79	0.92
GB	0.67	0.45	0.35	0.31	0.78	0.69	0.79	0.48	0.75	0.90	0.79	0.91

subsets, and recording the QoE prediction performance of each. First, consider the following feature subsets:

1. 1 feature types: VQA (1), M (2), I (3) and R_1+R_2 (4)

2. 2 feature types subsets: VQA+M (5) and VQA+I (6)

3. ≥ 3 subsets: VQA+M+R_2(7), M+R_1+R_2 (8),
M+I+R_1+R_2 (9), VQA+I+R_1+R_2 (10),
VQA+M+R_1+R_2 (11) and VQA+M+I+R_1+R_2 (12)

where the number indicates the index of the column in Table C.2 where the corresponding feature subset is used. The SROCC and LCC results are shown in Table C.2, where we selected ST-RRED as the VQA feature and used the train/test splits from Experiment 1.

Clearly, when using the individual components as features, the QoE prediction result was maximized when using VQA but was still very low, especially for other components such as M. Notably, the regression performance for the VQA subset was maximized in the case of the Ridge and Lasso linear regressions, but for the M (memory) and R_1+R_2 (stalling) feature types, the SROCC performance was greatly reduced using those regression models compared to SVR, RF and GB. This may be explained by the fact that the design of IQA/VQA algorithms such as ST-RRED ultimately aims for linear/explainable models. By contrast, the memory or stalling-aware features are highly non-linear, hence non-linear regression models may be expected to perform better.

We now move on to the different feature combinations and their effects on QoE prediction. First, note that when VQA is removed from the feature set (e.g. in columns 8 and 9) the prediction performance dropped considerably. When the feature set VQA+M+R_2 (see column 7 in Table C.2) was used, good prediction results were obtained. This strongly supports the importance of combining stalling with memory/recency effects on QoE when viewing longer video sequences. Overall, the combination of all feature types gave the best performance over most regression models. This suggests that a successful QoE prediction model should consider diverse QoE-aware features in order to better predict QoE.

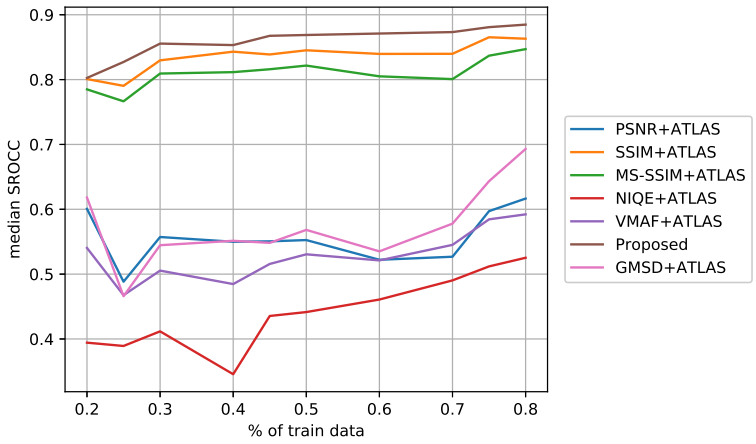

Figure C.2: Median SROCC as the amount of training data was varied for different objective video quality models.

C.2.1 Amount of Training Data and Pooling Strategy

Any data-driven learned model requires needs to have a sufficient amount of training data to perform reliably. To this end, we analyzed the effects of the amount of data used to train Video ATLAS on QoE prediction. By varying the percent of training data in the train/test split, we repeated the same process as before, over many trials. Figure C.2 shows how the SROCC changed when the amount of training data was varied between 0.2 (2 training contents) and 0.8 (11 training contents). Clearly, the prediction performance increased when the available training data was increased.

Lastly, we experimented with the type of pooling that is applied on the quality metric before it is used in the regression engine. We combined all features and used the train/test splits of Experiment 1. To collapse the frame-based objective quality scores to a single summary VQA score, we applied the

hysteresis pooling method in [132] and the VQ pooling method in [105]. The former combines past and future quality scores within a window, while the latter clusters the video frames into low and high quality regions and weights their contributions to the overall VQA score. The results are tabulated in Table C.3. For the mean pooling case, we used the results reported in Table C.1.

Given the results in Table C.3, we observed that the use of more sophisticated temporal pooling strategies did not always improve QoE prediction over mean pooling and any such improvements were not significant. This observation agrees with previous works [136] that have shown the advantages of mean pooling when processing longer video sequences to predict endpoint (retrospective) subjective quality.

Table C.3: SROCC results when using mean, hysteresis and VQ pooling for various VQA models using Video ATLAS.

VQA	mean	hysteresis	VQ
PSNR	0.62	0.62	0.69
SSIM [159]	0.86	0.85	0.81
MS-SSIM [160]	0.84	0.84	0.80
NIQE [94]	0.52	0.55	0.50
VMAF [79]	0.58	0.60	0.60
ST-RRED [143]	0.88	0.89	0.87
GMSD [164]	0.70	0.65	0.68

C.2.2 Statistical Analysis of Performance

We also carried out a statistical analysis of the results in Experiment 1. First, it is interesting to investigate how the performance varies across

train/test splits in Experiment 1. To this end, Table C.4 shows the standard deviation values of the computed SROCC for all of the compared models. It is clear that QoE-aware methods such as SQI and NARX have a more stable behavior as compared to QoS or VQA models, such as ST-RRED. Video ATLAS produced the lowest prediction uncertainty, demonstrating its robustness.

Table C.4: Statistical analysis for Experiment 1. The first column contains the median SROCC and the second the standard deviation across all train/test splits.

Model	median	σ
FTW	0.34	0.23
VsQM	0.32	0.21
PSNR	0.60	0.29
SSIM	0.68	0.19
MS-SSIM	0.68	0.21
NIQE	0.21	0.20
VMAF	0.61	0.22
GMSD	0.65	0.28
ST-RRED	0.68	0.20
PSNR+SQI	0.55	0.24
SSIM+SQI	0.75	0.15
MS-SSIM+SQI	0.75	0.15
ST-RRED+SQI	0.57	0.23
P.1203 mode 0	0.46	0.18
P.1203 mode 3	0.44	0.19
NARX	0.79	0.14
Video ATLAS	0.88	0.11

We conclude this Appendix by studying the statistical significance of the SROCC performance results presented in Experiment 1. We used the Wilcoxon ranksum test [139] with significance level 0.01 by comparing the distributions of SROCC (for the top-5 performers) across all 364 trials and present the results in Table C.5. It can be observed that ST-RRED performed significantly worse, since it does not capture QoE-aware information. The SQI

variants that used SSIM or MS-SSIM performed statistically equivalently, but were statistically worse by the NARX model. As we already mentioned, Video ATLAS performed significantly better in this experiment than NARX, since it was designed specifically for retrospective, rather than continuous-time, QoE prediction.

Table C.5: Statistical significance for top-5 performers in Experiment 1. A value of "1" indicates that the row is statistically better than the column, while a value of "0" indicates that the row is statistically worse than the column; a value of "-" indicates that the row and column are indistinguishable.

Model	STRRED	SQI SSIM	SQI MS-SSIM	NARX	Video ATLAS
STRRED	-	0	0	0	0
SQI SSIM	1	-	-	0	0
SQI MS-SSIM	1	-	-	0	0
NARX	1	1	1	-	0
Video ATLAS	1	1	1	1	-

Appendix D

Additional Analysis of the G-NARX model

The design of continuous-time QoE predictors often involves deciding upon a number of architecture-specific settings, including an imputation strategy, the activation function and the training algorithm. Next, we discuss these aspects and conclude with a note on computational complexity.

D.1 Inputs of Different Length

An important consideration when implementing the proposed model is accounting for different input durations. For example, while video quality predictions are computed on all frames of normal playback [25], the R_1 input (in the presence of rebuffering events) will have longer durations. While it is possible to train and evaluate the GN and GR QoE Prediction models without imputing missing VQA response values during rebuffering events, we found it useful to develop an imputation scheme that defines same-sized inputs for each test video. In previous studies, playback interruption has been found to be at least as annoying as very low bitrate distortions [27]; hence we selected imputed VQA values corresponding to very low video quality. Imputing with zeros is not a good idea; some video quality models never approach such low

values while others (such as ST-RRED) correspond lower values to better video quality. For simplicity, we picked the min (or the max) value of the video quality prediction corresponding to the worst quality level encountered over the entire video as the nominal VQA input value during playback interruptions. To recognize causality, we could also pick the min (or max) VQA values up until the rebuffering event occurs; we found that this did not greatly affect the final results. This imputing step is required only on the LIVE-NFLX dataset.

D.2 Activation Function

We experimented with various activation functions: logistic sigmoid (logsig), hyperbolic tangent sigmoid (tansig) and linear (purelin) and we also tried various combinations of them in the hidden and output layers. We carried out ten experiments and computed the median OR on D_1 and D_2. For D_1, we used $d_u = 10$, $d_y = 10$, a single hidden layer with 8 neurons and ST-RRED as the VQA model. For D_2, we used $d_u = 6$, $d_y = 6$, a single hidden layer with 8 neurons and the features R_1, R_2 and M. As shown in Table D.1, using tansig for the hidden layer and purelin for the output layer proved to be good choices (in terms of OR) for this task on both databases. Other evaluation metrics produced similar results.

D.2.1 Training Algorithm

We compared the default Levenberg-Marquardt algorithm against other training algorithms [15]. Table D.2 shows that using the Levenberg-Marquardt

Table D.1: OR comparison between different activation functions when training the NARX component on D_1 (VN) and on D_2 (RMN). Rows and columns correspond to the activation function used in the hidden and the output layer respectively.

Database	D_1 (VN)			D_2 (RMN)		
Activation	tansig	logsig	purelin	tansig	logsig	purelin
tansig	10.38	20.28	5.90	10.59	31.38	7.68
logsig	8.97	22.55	5.10	10.34	33.26	7.92
purelin	9.28	31.28	11.10	26.04	50.55	5.48

(trainlm) performed very close to the best performing method on D_1 (trainbfg) and was significantly better on D_2. This suggests that the use of a general training algorithm such as Levenberg-Marquardt is sufficient for QoE prediction.

Table D.2: Comparison between different training algorithms using NARX on databases D_1 (VN) and D_2 (RMN). The number of iterations was set to 1000.

Database	D_1				D_2			
Metric	RMSE	OR	DTW	SROCC	RMSE	OR	DTW	SROCC
trainlm	4.00	5.72	15.39	0.91	4.43	7.47	4.15	0.93
trainbfg	3.90	4.86	14.73	0.90	6.33	17.53	6.65	0.81
trainrp	4.26	7.79	17.59	0.89	9.21	29.25	9.96	0.71
trainscg	4.20	6.04	16.53	0.89	6.59	21.09	7.21	0.79
traincgb	4.01	5.35	15.93	0.89	6.14	18.36	6.34	0.82
traincgf	4.27	6.86	16.54	0.88	6.11	18.57	6.59	0.82
traincgp	4.07	5.83	15.59	0.89	6.46	21.09	6.57	0.80
trainoss	4.49	6.97	18.14	0.87	7.19	24.27	7.30	0.80
traingdx	6.33	17.72	22.26	0.80	11.87	38.49	10.04	0.66

D.3 Computational Complexity of G-NARX

The proposed continuous-time QoE predictors require calculating perceptual VQA models, training and testing the neural network. Therefore,

besides calculating the VQA feature, these neural networks can be trained offline and take up only a small computational overhead. To demonstrate this, we fixed the NARX architecture to $d_u = 10$ and $d_y = 10$ lags, $H = 8$ hidden nodes and a single hidden layer, then calculated the compute time for SSIM, for training and for testing the GN-QoE predictor on all 112 videos in D_3 (see Table D.3). All of the timing experiments were carried out on a 16.04 Ubuntu LTS Intel i7-4790@3.60 GHz system. Both the NARX and SSIM implementations used unoptimized Matlab code.

As shown in Table D.3, calculating SSIM and training the neural network take up considerably more time than testing it. Notably, calculating SSIM takes much more time than training, since we deployed relatively simple neural networks. For adaptive streaming applications, where the reference video and its compressed versions are readily available, the VQA measurements and the neural network training can be carried out in an offline fashion. Trained model values and associated VQA values can be sent to the client as part of the metadata and then the client side can perform such QoE predictions in real-time. Compared to simply calculating the VQA values, the only (online) computational overhead of the proposed predictors is the testing step, which is relatively fast. If the client side has low computational power, these operations could also be carried out by proxy "QoE-aware" servers.

The GN-QoE predictor uses ST-RRED as its VQA feature which, compared to SSIM, is a significantly better-performing VQA model [13], but its computational overhead may limit its potential in some practical applications.

Table D.3: Average computation times on D_3 (112 videos) for the GN-QoE predictor using SSIM.

Computation	Sec.
SSIM	290.66
Training	4.87
Testing	0.04

However, efficient approximations to ST-RRED that are implemented in the spatial domain are available [29].

Appendix E

Additional Experimental Analysis of the G-NARX models

In this section, we study in greater detail continuous-time performance bounds, the effects of using different rebuffering-related inputs for D_2 and provide more detailed results in Tables E.4, E.5 and E.7.

E.1 Details on Continuous-time Performance Bounds

Table E.1: Median performance for various time-series ensemble methods applied on the class of RM-predictors on database D_2 - direct comparison with human performance ("ref" row).

Model Type	RMN				RMR				RMH			
Model/Metric	RMSE	OR	DTW	SROCC	RMSE	OR	DTW	SROCC	RMSE	OR	DTW	SROCC
best	4.90	2.14	4.75	0.91	6.46	7.10	6.38	0.87	6.03	7.76	9.76	0.75
avg	4.34	0.00	3.85	0.95	5.74	2.13	4.55	0.93	4.61	1.33	5.85	0.87
med	4.46	0.00	3.71	0.94	5.56	1.08	3.86	0.95	4.39	1.35	6.23	0.86
mod	4.33	0.00	3.79	0.94	5.48	1.05	3.94	0.94	4.41	1.33	6.37	0.85
DTW-single	4.55	0.00	4.02	0.94	5.62	1.33	4.00	0.94	4.52	1.13	7.61	0.84
DTW-prob	4.40	0.00	3.78	0.95	5.62	1.18	3.96	0.95	4.57	1.16	5.72	0.87
ref	3.91	0.00	4.60	0.93	3.91	0.00	4.60	0.93	3.91	0.00	4.60	0.93

Following the steps described in Section 4.7.4, we compared the best performing combination (RMN-QoE Predictor) against an upper bound, i.e., human performance, using $S = 10$ shuffles. Table E.1 shows that the RMN-QoE Predictor outperformed both the RMR- and RMH-QoE Predictors, and its performance in terms of RMSE came close to the reference human perfor-

Table E.2: Median performance for various time-series ensemble methods applied on the class of G-predictors on D_3 using ST-RRED - direct comparison with human scores ("ref" row).

Model Type	GN				GR				GH			
Model/Metric	RMSE	OR	DTW	SROCC	RMSE	OR	DTW	SROCC	RMSE	OR	DTW	SROCC
best	0.29	5.48	28.39	0.78	0.38	9.65	29.52	0.69	0.24	2.37	25.56	0.76
avg	0.26	0.00	23.08	0.86	0.39	3.30	21.63	0.79	0.19	0.00	10.19	0.87
med	0.25	0.00	22.52	0.86	0.30	2.21	19.35	0.80	0.15	0.00	9.16	0.88
mod	0.25	0.00	22.03	0.85	0.30	2.17	19.94	0.80	0.14	0.00	9.28	0.89
DTW-single	0.26	0.00	19.99	0.86	0.31	3.10	21.09	0.80	0.16	0.00	13.82	0.85
DTW-prob	0.25	0.00	21.48	0.86	0.30	2.32	19.17	0.81	0.16	0.00	9.46	0.89
ref	0.20	0.00	10.71	0.90	0.20	0.00	10.71	0.90	0.20	0.00	10.71	0.90

mance. We found this difference to be statistically significant; hence there is some room for improvement. However, the performance in terms of OR was very good when any of the ensemble methods was considered. Surprisingly, the DTW and SROCC performances were not always inferior to human scores, and sometimes these differences were statistically significant.

Comparing the objective prediction scores between Tables E.6 and E.1, we discovered that, when using only a subset of the subjective scores as ground truth, the performance of the objective prediction models was reduced. This may be explained by the fact that subjects do not always agree with each other; hence using all of the subjective scores reduces both the objective and subjective uncertainty.

As in D_2, we also report the results compared against human performance in Table E.2 for D_3. We drew similar observations as in Table E.1: the objective predictions tend to get worse while human performance usually upper bounds model performance. It is intriguing that combining the different GH-QoE forecasts delivered RMSE scores better than human performance

- a difference which we found to be statistically significant. When objective prediction models are trained on subjective data, human performance should generally be superior to or at least statistically equivalent to objective predictions. However, this upper bound may be violated when we consider post-processed forecasting ensembles: human performance is the upper bound only on time-series predictions generated by an *individual* model. Our observation may be explained by the design of these two QoE databases. Database D_2 includes only rebuffering events, while D_3 involves a mixture of rebuffering and compression; a task that is even more challenging for human subjects. Therefore, the difficulty of the tasks may increase subjective uncertainty per test video; an uncertainty for which simple averaging of the continuous scores across subjects may not always be the best method of aggregating them. This reinforces our growing belief that simply averaging continuous QoE responses disregards the inherent non-linearities in these responses [27].

E.2 Rebuffering-related inputs

It has been shown [23] that combinations of VQA inputs (e.g. ST-RRED combined with SSIM) can deliver improved results. Here we investigate the effects of using different combinations of rebuffering-related inputs. We selected NARX as the dynamic model, and performed QoE predictions using a number of inputs ranging from one to three, as shown in Table E.3. We also used the parameters from Table 4.3. Notably, we found that only using the R_1 input contributed significantly greater prediction power than R_2 and M; it

is capable of effectively capturing rebuffering effects and is suitable for being used alone in the GN-, GR- and GH-prediction models. Combining all three inputs improved the OR by only 2%. This suggests that R_1 is an efficient descriptor of the effects of rebuffering events on QoE.

Table E.3: Median performance for various continuous-time feature sets on D_2 when using the NARX learner. Note that using features R_1+R_2 defines the RN-QoE Predictor while R_1+R_2+M gives the RMN-QoE Predictor.

Model	NARX			
Features/Metric	RMSE	OR	DTW	SROCC
R_1	4.65	9.03	4.00	0.94
R_2	8.38	31.15	7.29	0.82
M	6.74	23.12	6.39	0.82
R_1+R_2	4.41	8.14	4.02	0.95
R_1+M	4.86	12.12	4.26	0.92
R_2+M	6.41	21.53	6.17	0.84
R_1+R_2+M	4.49	6.84	4.08	0.93

E.3 Additional Tables

In this section we include the earlier described Tables E.4, E.5, E.6 and E.7.

Table E.4: Median performance for the class of V- QoE predictors on D_1.

Model Type	VN				VR				VH			
Model/Metric	RMSE	OR	DTW	SROCC	RMSE	OR	DTW	SROCC	RMSE	OR	DTW	SROCC
NIQE [94]	8.61	34.79	29.72	0.54	9.71	42.84	49.42	0.33	8.95	42.78	55.86	0.27
PSNR	6.76	25.07	24.37	0.72	8.10	36.16	35.55	0.56	7.19	29.51	37.49	0.67
VMAF [79]	4.95	12.38	17.80	0.89	6.42	24.05	27.86	0.73	6.44	23.03	27.42	0.81
MS-SSIM [160]	4.07	5.73	15.89	0.91	5.79	17.64	23.67	0.73	7.50	31.82	44.86	0.59
SSIM [159]	4.02	5.45	14.22	0.90	6.07	17.43	24.13	0.74	7.32	30.69	41.78	0.67
ST-RRED [143]	4.25	5.90	15.21	0.90	6.98	20.81	27.22	0.71	5.40	15.31	27.09	0.87

Table E.5: Median performance for various ensemble methods applied on the class of V-predictors on D_1 using ST-RRED.

Model Type	VN				VR				VH			
Model/Metric	RMSE	OR	DTW	SROCC	RMSE	OR	DTW	SROCC	RMSE	OR	DTW	SROCC
best	4.25	5.90	15.21	0.90	6.98	20.81	27.22	0.71	5.40	15.31	27.09	0.87
avg	3.64	5.24	14.11	0.91	4.99	15.59	17.64	0.85	4.86	14.69	16.72	0.90
med	3.69	5.24	14.01	0.91	4.23	9.15	16.31	0.90	4.85	13.99	16.46	0.90
mod	3.76	4.55	14.26	0.91	4.17	8.47	16.22	0.90	4.82	13.99	20.92	0.90
DTW-single	3.92	5.59	14.01	0.90	4.24	8.81	17.06	0.90	5.02	15.04	18.52	0.89
DTW-prob	3.67	5.25	14.11	0.91	4.20	10.51	16.35	0.89	4.84	14.69	16.72	0.90

Table E.6: Median performance for various ensemble methods applied on the class of RM-predictors on D_2.

Model Type	RMN				RMR				RMH			
Model/Metric	RMSE	OR	DTW	SROCC	RMSE	OR	DTW	SROCC	RMSE	OR	DTW	SROCC
best	4.49	6.84	4.08	0.93	6.33	21.08	5.74	0.89	5.66	16.22	9.04	0.75
avg	4.01	0.00	2.99	0.97	5.59	11.48	3.83	0.95	4.20	3.71	5.43	0.88
med	3.88	0.00	2.93	0.97	5.38	6.62	3.19	0.96	3.79	4.29	5.73	0.87
mod	3.93	0.00	3.03	0.96	5.34	7.60	3.23	0.96	3.88	4.03	5.65	0.86
DTW-single	4.15	0.00	3.03	0.97	5.39	7.25	3.36	0.95	3.99	3.88	6.84	0.86
DTW-prob	3.91	0.00	2.96	0.97	5.33	7.25	3.31	0.96	4.05	3.38	5.10	0.88

Table E.7: Median performance for various ensemble methods applied on the class of G-predictors on D_3 using ST-RRED.

Model Type	GN				GR				GH			
Model/Metric	RMSE	OR	DTW	SROCC	RMSE	OR	DTW	SROCC	RMSE	OR	DTW	SROCC
best	0.28	16.31	26.53	0.81	0.37	22.55	28.58	0.72	0.22	6.19	25.45	0.77
avg	0.24	8.31	19.82	0.88	0.29	14.87	20.11	0.81	0.15	0.33	8.08	0.90
med	0.24	6.66	21.65	0.89	0.29	13.90	18.47	0.82	0.11	0.00	7.43	0.91
mod	0.24	3.92	20.60	0.88	0.28	13.90	19.23	0.81	0.10	0.00	6.98	0.91
DTW-single	0.25	6.02	19.75	0.89	0.30	14.77	21.28	0.82	0.13	0.00	12.25	0.87
DTW-prob	0.24	6.54	20.00	0.89	0.29	14.31	18.90	0.82	0.12	0.00	7.45	0.91

E.4 Modeling Recency

To conclude this Appendix, we now show that the NARX-driven GN-QoE predictor is indeed able to capture recency effects in subjective QoE. To do so, we collected the GN-QoE predictions from D_2 and D_3, then performed a moving average operation, i.e., we averaged the predictions (and the subjective ground truths) at evenly-spaced moments separated by 10 and 5 seconds on D_2 and D_3 respectively, using corresponding sliding window sizes of 5 and 2.5 seconds respectively. Figure E.1 shows that both the subjective and objective scores are very strongly correlated with preceding time averages, indicating that the objective GN-QoE predictions are indeed able to capture the effects of recency in subjective QoE.

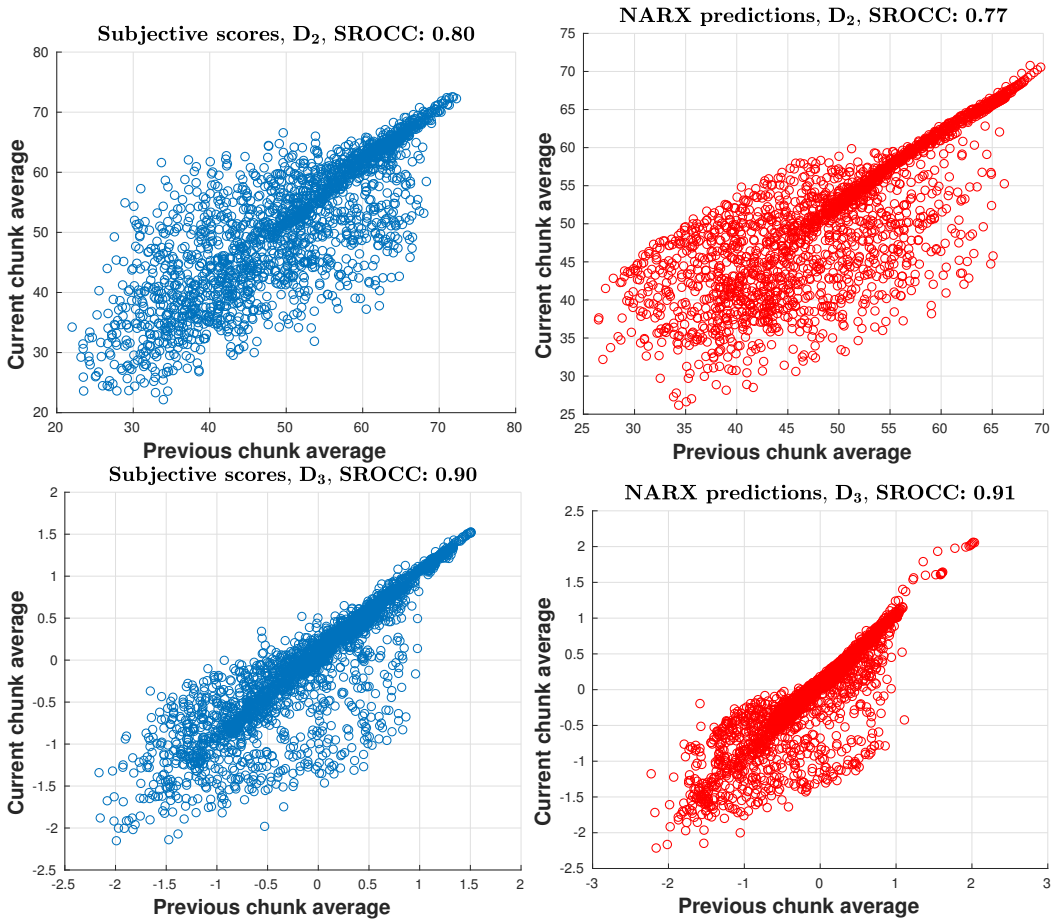

Figure E.1: Relationship between current and previous subjective and objective scores on D_2 and D_3. The objective predictions are able to capture the effects of recency.

Appendix F

Encoding Module, Video Quality Module and the Streaming Pipeline

F.1 Encoding Module

To generate the video encodes, we adopted the Dynamic Optimizer (DO) [67] approach that was recently developed and implemented by Netflix. DO determines the optimal encoding parameters per shot, such that a predefined metric is optimized at a given bitrate. The underlying assumption is that video frames within a video shot have similar spatio-temporal characteristics (e.g. camera motion and/or spatial activity), and hence should be encoded at a particular resolution and QP value. For a specific target bitrate, the DO implementation determines the "optimal" resolution and QP values per shot that achieves (but does not exceed) this bitrate while maximizing the overall quality predicted by the VMAF model [79]. Repeating this process over each target bitrate value and video segment yields a 2D encoding chunk map, where each row is a single video stream (corresponding to the same target bitrate) and each column is a different video segment over time (see Fig. F.1) encoded at the QP and resolution values selected by the DO.

Both the encoding and client modules are driven by VMAF [79] mea-

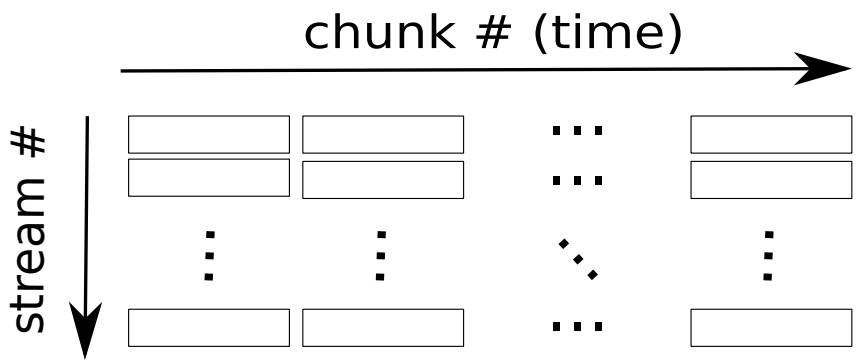

Figure F.1: An encoding chunk map representation.

surements which are carried out by the video quality module. In the following section, we discuss the video quality module in detail.

F.2 Video Quality Module

The end receiver of any video streaming service is the human eye, hence integrating video quality models in the streaming network is of paramount importance to achieve perceptually optimized video streaming. Numerous studies have shown that simple video quality indicators such as the encoding bitrate or peak signal-to-noise ratio (PSNR) do not correlate well with human perception [159, **?**]. Nevertheless, PSNR is still often used for codec optimization and codec comparisons [79], while the encoding bitrate is widely used by client adaptation algorithms.

In our model system, the video quality module performs perceptual video quality calculations that are fed to the encoding module, to determine the bitrate ladder (the set of target bitrates per content), construct the convex

hull and determine the encoding parameters, and also to the client module, to drive quality-based streaming decisions. We also use video quality measurements to perform offline analysis of the final video sequences and to compare them with human subjective scores (see Section 6.7). To effectively measure quality, we relied on the VMAF model [79]. The choice of VMAF is not restrictive, and other high-performing video quality indicators can also be used.

VMAF is used to encode and monitor the quality of millions of encodes on a daily basis [79] and exhibits a number of key properties. It has been trained on streaming-related video impairments, such as compression and the elementary features, such as the VIF model [137, 138], are highly descriptive of perceptual quality. It is also more computationally efficient than time-consuming VQA models such as MOVIE [134] and VQM-VFD [109]. Further, the VMAF framework is publicly available [8] and can be improved even further by adding other quality-aware features or regression models. Lastly, given that it is a trained algorithm, it produces scores that are linear with the subjective scale, i.e., a VMAF of 80 (VMAF $\in [0, ..., 100]$) means that, on average, viewers will rate a video with a score of 8 out of 10.

F.3 Putting the Pieces Together

After having described all four modules, we can give an overview of the end-to-end streaming pipeline, as depicted in Fig. F.2. First, the encoding module performs shot detection and splits the video content into different shots [67]. Each of these shots is then encoded at multiple encoding levels,

determined by the corresponding resolution and QP values. The video quality module calculates the average per segment VMAF values for each of the pre-encoded segments. After determining the target bitrates for each content, these bitrates are then passed (along with the VMAF values and the encoded chunks) into the DO, which decides on the per shot encoding resolution and QP values. This results in the 2D encoding chunk map (also depicted in Fig. F.1). Notably, these steps are carried out in an offline fashion and are orthogonal to the client's behavior and/or the network condition.

On the client side, the client device first pre-fetches a $B_0 = 1$ chunk. Based on the client algorithm (BB, RB, QB or OQB), the client then decides which stream should be selected for the next chunk. If the buffer is depleted, then rebuffering occurs. To simulate rebuffering, we retrieve the latest frame that was played out, and overlay a spinning wheel icon on the viewing screen. Rebuffering occurs until the buffer is sufficiently filled to display the next chunk and the client adaptation algorithm allows for the playback to resume. Before display, each encode is upscaled using bicubic interpolation, to match the 1080p display resolution. In the case of QB and OQB, the client uses the VMAF values of future segments (within the horizon h) to drive its decision. Note that for RB and QB the available throughput is estimated by averaging the download speeds of the $w = 5$ latest chunks.

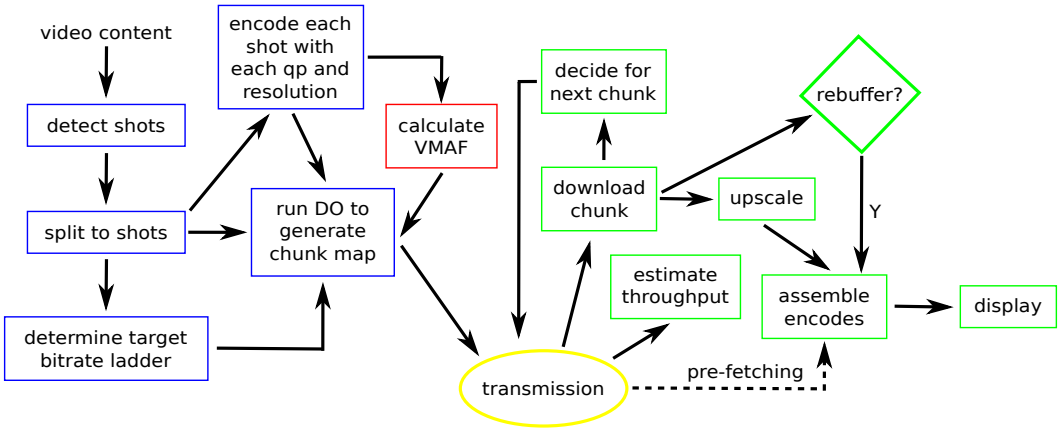

Figure F.2: Full pipeline of LIVE-NFLX-II, where each module is color-coded: blue: encoding module; red: video quality module, yellow: network module and green: client module. The client's behavior is orthogonal to the offline video encoding and quality calculations on the server side.

Bibliography

[1] https://en.wikipedia.org/wiki/International_expansion_of_Netflix.

[2] http://www.cdvl.org.

[3] https://www.itu.int/rec/T-REC-P.1203.

[4] https://www.its.bldrdoc.gov/vqeg/projects/audiovisual-hd.aspx.

[5] https://www.itu.int/rec/T-REC-P.1203-201611-I.

[6] http://skuld.cs.umass.edu/traces/mmsys/2013/pathbandwidth/.

[7] https://www.blender.org.

[8] https://github.com/Netflix/vmaf.

[9] Cisco visual networking index: Global mobile data traffic forecast update, 2016-2021.

[10] Daala codec. https://git.xiph.org/daala.git/.

[11] Hybrid-FR objective perceptual video quality measurement for HDTV and multimedia IP-based video services in the presence of a full reference signal and non-encrypted bitstream data. ITU-T Rec. J.343.6.

[12] Netflix encoding tool aims to retain video quality on slow 100kbps iphone mobile data connections.

[13] On the robust performance of the ST-RRED video quality predictor. http://live.ece.utexas.edu/research/Quality/ST-RRED/.

[14] scikit-learn: Machine learning in python. http://scikit-learn.org/stable/.

[15] https://www.mathworks.com/help/nnet/ug/choose-a-multilayer-neural-network-training-function.htmlChoose a Multilayer Neural Network Training Function.

[16] https://www.mathworks.com/help/nnet/ug/design-time-series-narx-feedback-neural-networks.htmlDesign Time Series NARX Feedback Neural Networks.

[17] https://www.mathworks.com/help/nnet/ug/improve-neural-network-generalization-and-avoid-overfitting.html.

[18] Video Quality Experts Group (VQEG), VQEG HDTV Phase I, https://www.its.bldrdoc.gov/vqeg/projects/hdtv/hdtv.aspx.

[19] ISO/IEC FCD 23001-6 part 6: Dynamics adaptive streaming over HTTP (DASH). MPEG Requirements Group, 2011.

[20] SHVC verification test results. Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, February 2016.

[21] Avşar Asan, Werner Robitza, Is-haka Mkwawa, Lingfen Sun, Emmanuel Ifeachor, and Alexander Raake. Impact of video resolution changes on QoE for adaptive video streaming. *International Conference on Multimedia and Expo*, pages 499–504, 2017.

[22] Athula Balachandran, Vyas Sekar, Aditya Akella, Srinivasan Seshan, Ion Stoica, and Hui Zhang. Developing a predictive model of quality of experience for internet video. *ACM SIGCOMM Computer Communication Review*, 4:339–350, 2013.

[23] C. G. Bampis and A. C. Bovik. An Augmented Autoregressive Approach to HTTP Video Stream Quality Prediction. *arXiv:* ***https://arxiv.org/abs/1707.02709***.

[24] C. G. Bampis and A. C. Bovik. Learning to Predict Streaming Video QoE: Distortions Rebuffering and Memory. *Signal Processing: Image Communication,* under review*, arXiv:* ***https://arxiv.org/abs/1703.00633***.

[25] C. G. Bampis, Z. Li, and A. C. Bovik. Continuous prediction of streaming video QoE using dynamic networks. *IEEE Signal Processing Letters*, 24(7):1083–1087, July 2017.

[26] C. G. Bampis, Z. Li, I. Katsavounidis, and A. C. Bovik. Recurrent and dynamic models for predicting streaming video quality of experience. *IEEE Trans. on Image Process., to appear*, 2018.

[27] C. G. Bampis, Z. Li, A. K. Moorthy, I. Katsavounidis, A. Aaron, and A. C. Bovik. Study of Temporal Effects on Subjective Video Quality of Experience. *IEEE Transactions on Image Processing*, 26(11):5217–5231, 2017.

[28] C. G. Bampis, Zhi Li, Ioannis Katsavounidis, and A. C. Bovik. Recurrent and dynamic networks that predict streaming video quality of experience. *Transactions on Image Processing,* under review.

[29] Christos G Bampis, Praful Gupta, Rajiv Soundararajan, and Alan C Bovik. Speed-QA: Spatial efficient entropic differencing for image and video quality. *IEEE Signal Processing Letters*, 2017.

[30] Andrzej Beben, P Wiśniewski, J Mongay Batalla, and Piotr Krawiec. ABMA+: lightweight and efficient algorithm for HTTP adaptive streaming. *International Conference on Multimedia Systems*, 2016.

[31] Abdelhak Bentaleb, Ali C Begen, and Roger Zimmermann. SDNDASH: Improving QoE of HTTP adaptive streaming using software defined networking. *ACM Multimedia Conference*, pages 1296–1305, 2016.

[32] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[33] Fadi Boulos, Benoît Parrein, Patrick Le Callet, and David Hands. Perceptual effects of packet loss on H. 264/AVC encoded videos. In *Work-*

shop on Video Processing and Quality Metrics for Consumer Electronics, 2009.

[34] Alan Conrad Bovik. Automatic prediction of perceptual image and video quality. *Proceedings of the IEEE*, 101(9):2008–2024, 2013.

[35] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[36] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[37] Matteo Carandini, David J Heeger, and J Anthony Movshon. Linearity and normalization in simple cells of the macaque primary visual cortex. *Journal of Neuroscience*, 17(21):8621–8644, 1997.

[38] Chao Chen, Lark Kwon Choi, Gustavo de Veciana, Constantine Caramanis, Robert W Heath, and Alan C Bovik. Modeling the time-varying subjective quality of HTTP video streams with rate adaptations. *IEEE Trans. on Image Process.*, 23(5):2206–2221, 2014.

[39] David M Corey, William P Dunlap, and Michael J Burke. Averaging Correlations: Expected values and Bias in Combined Pearson rs and Fisher's z Transformations. *The Journal of General Psychology*, 125(3):245–261, 1998.

[40] Qin Dai and Ralf Lehnert. Impact of packet loss on the perceived video quality. *International Conference on Evolving Internet*, pages 206–209, 2010.

[41] Luca De Cicco, Vito Caldaralo, Vittorio Palmisano, and Saverio Mascolo. Elastic: a client-side controller for dynamic adaptive streaming over http (DASH). *International Packet Video Workshop*, pages 1–8, 2013.

[42] Jan De Cock, Zhi Li, Megha Manohara, and Anne Aaron. Complexity-based consistent-quality encoding in the cloud. *IEEE International Conference on Image Processing (ICIP)*, 2016.

[43] Toon De Pessemier, Katrien De Moor, Wout Joseph, Lieven De Marez, and Luc Martens. Quantifying the influence of rebuffering interruptions on the user's quality of experience during mobile video watching. *IEEE Transactions on Broadcasting*, 59(1):47–61, 2013.

[44] Francesca De Simone, Marco Tagliasacchi, Matteo Naccari, Stefano Tubaro, and Touradj Ebrahimi. A H. 264/AVC video database for the evaluation of quality metrics. In *IEEE Int'l Conf. Acoust., Speech and Signal Process.*, 2010.

[45] Zhengfang Duanmu, Kai Zeng, Kede Ma, Abdul Rehman, and Zhou Wang. A quality-of-experience index for streaming video. *IEEE Journal of Selected Topics in Signal Processing*, 11(1):154–166, 2017.

[46] Sebastian Egger, Bruno Gardlo, Michael Seufert, and Raimund Schatz. The impact of adaptation strategies on perceived quality of HTTP adaptive streaming. *Workshop on Design, Quality and Deployment of Adaptive Video Streaming*, pages 31–36, 2014.

[47] Jeffrey L Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

[48] Charles Fenimore, John Libert, and Stephen Wolf. Perceptual effects of noise in digital video compression. *SMPTE journal*, 109(3):178–187, 2000.

[49] David J Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4(12):2379–2394, 1987.

[50] M-N Garcia, Francesca De Simone, Samira Tavakoli, Nicolas Staelens, Sebastian Egger, Kjell Brunnström, and Alexander Raake. Quality of experience and HTTP adaptive streaming: A review of subjective studies. *International Workshop on Quality of Multimedia Experience*, pages 141–146, 2014.

[51] Deepti Ghadiyaram, Alan C Bovik, Hojatollah Yeganeh, Roman Kordasiewicz, and Michael Gallant. Study of the effects of stalling events on the quality of experience of mobile streaming videos. *IEEE Global Conference on Signal and Information Processing*, 2014.

[52] Deepti Ghadiyaram, Janice Pan, and Alan C Bovik. A time-varying subjective quality model for mobile streaming videos with stalling events. *SPIE Conf. on Optical Engineering+ Applications*, 2015.

[53] Deepti Ghadiyaram, Janice Pan, and Alan C Bovik. A subjective and objective study of stalling events in mobile streaming videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[54] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *Trans. on Patt. Anal. and Mach. Intell.*, 31(5):855–868, 2009.

[55] Anthony J Greene, Colin Prepscius, and William B Levy. Primacy versus recency in a quantitative model: activity is the critical distinction. *Learning & Memory*, 7(1):48–57, 2000.

[56] David S Hands and SE Avons. Recency and duration neglect in subjective assessment of television picture quality. *Applied Cognitive Psychology*, 15(6):639–657, 2001.

[57] Tobias Hoßfeld, Sebastian Biedermann, Raimund Schatz, Alexander Platzer, Sebastian Egger, and Markus Fiedler. The memory effect and its implications on Web QoE modeling. *International Teletraffic Congress*, pages 103–110, 2011.

[58] Tobias Hoßfeld, Sebastian Egger, Raimund Schatz, Markus Fiedler, Kathrin Masuch, and Charlott Lorentzen. Initial delay vs. interruptions: Between the devil and the deep blue sea. *International Workshop on Quality of Multimedia Experience*, pages 1–6, 2012.

[59] Tobias Hoßfeld, Michael Seufert, Matthias Hirth, Thomas Zinner, Phuoc Tran-Gia, and Raimund Schatz. Quantification of YouTube QoE via crowdsourcing. *IEEE International Symposium on Multimedia*, 2011.

[60] Vlad Hosu, Franz Hahn, Mohsen Jenadeleh, Hanhe Lin, Hui Men, Tamás Szirányi, Shujun Li, and Dietmar Saupe. The Konstanz natural video database (KoNViD-1k). In *International Conference on Quality of Multimedia Experience*, pages 1–6. IEEE, 2017.

[61] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. A buffer-based approach to rate adaptation: Evidence from a large video streaming service. *ACM SIGCOMM Computer Communication Review*, 44(4):187–198, 2015.

[62] Mia Hubert and Ellen Vandervieren. An adjusted boxplot for skewed distributions. *Computational statistics & data analysis*, 52(12):5186–5201, 2008.

[63] Int. Telecommuncation Union Std. *BT-500-13: Methodology for the Subjective Assessment of the Quality of Television Pictures.*

[64] Int. Telecommuncation Union Std. *ITU-T Recommendation P.910: Subjective video quality assessment methods for multimedia applications.*

[65] Junchen Jiang, Vyas Sekar, and Hui Zhang. Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive. *IEEE/ACM Transactions on Networking (TON)*, 22(1):326–340, 2014.

[66] Vinay Joseph and Gustavo de Veciana. Nova: QoE-driven optimization of DASH-based video delivery in networks. *INFOCOM*, pages 82–90, 2014.

[67] I. Katsavounidis. Dynamic optimizer-a perceptual video encoding optimization framework. `https://medium.com/netflix-techblog/dynamic-optimizer-a-perceptual-video-encoding-optimization-framework-e19f1e3a277f`.

[68] Yoshikazu Kawayoke and Yuukou Horita. NR objective continuous video quality assessment model based on frame quality measure. *IEEE International Conference on Image Processing*, 2008.

[69] Nikolaos Kourentzes, Devon K Barrow, and Sven F Crone. Neural network ensemble operators for time series forecasting. *Expert Systems with Applications*, 41(9):4235–4244, 2014.

[70] Anders Krogh, Jesper Vedelsby, et al. Neural network ensembles, cross validation, and active learning. *Advances in neural information processing systems*, 7:231–238, 1995.

[71] Jonathan Kua, Grenville Armitage, and Philip Branch. A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP. *IEEE Communications Surveys & Tutorials*, 19(3):1842–1866, 2017.

[72] Eric C Larson and Damon M Chandler. Most apparent distortion: full-reference image quality assessment and the role of strategy. *Electronic Imaging*, 19(1):011006–011006, 2010.

[73] Patrick Le Callet, Christian Viard-Gaudin, and Dominique Barba. A convolutional neural network approach for objective video quality assessment. *IEEE Transactions on Neural Networks*, 17(5):1316–1327, 2006.

[74] Mikołaj Leszczuk, Mateusz Hanusiak, Mylène CQ Farias, Emmanuel Wyckens, and George Heston. Recent developments in visual quality monitoring by key performance indicators. *Multimedia Tools and Applications*, 75(17):10745–10767, 2016.

[75] Martin Leutbecher and Tim N Palmer. Ensemble forecasting. *Journal of Computational Physics*, 227(7):3515–3539, 2008.

[76] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 1944.

[77] Qiang Li and Zhou Wang. Reduced-reference image quality assessment using divisive normalization-based image representation. *IEEE journal of selected topics in signal processing*, 3(2):202–211, 2009.

[78] Songnan Li, Fan Zhang, Lin Ma, and King Ngi Ngan. Image quality assessment by separately evaluating detail losses and additive impairments. *IEEE Trans. Multimedia*, 13(5):935–949, 2011.

[79] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara. Toward a practical perceptual video quality metric.

[80] Zhi Li and Christos G Bampis. Recover subjective quality scores from noisy measurements. *Data Compression Conference*, 2017.

[81] Zhi Li, Ali C Begen, Joshua Gahm, Yufeng Shan, Bruce Osler, and David Oran. Streaming video over HTTP with consistent quality. *ACM multimedia systems conference*, pages 248–258, 2014.

[82] Zhi Li, Xiaoqing Zhu, Joshua Gahm, Rong Pan, Hao Hu, Ali C Begen, and David Oran. Probe and adapt: Rate adaptation for HTTP video streaming at scale. *Selected Areas in Communications*, 32(4):719–733, 2014.

[83] Joe Yuchieh Lin, Rui Song, Chi-Hao Wu, TsungJung Liu, Haiqiang Wang, and C-C Jay Kuo. Mcl-v: A streaming video quality assessment database. *Journal of Visual Communication and Image Representation*, 30:1–9, 2015.

[84] Tsungnan Lin, Bill G Horne, Peter Tino, and C Lee Giles. Learning long-term dependencies in NARX recurrent neural networks. *IEEE Trans. on Neur. Netw.*, 7(6):1329–1338, 1996.

[85] Chenghao Liu, Imed Bouazizi, and Moncef Gabbouj. Rate adaptation for adaptive http streaming. *ACM Conference on Multimedia Systems*, pages 169–174, 2011.

[86] Yao Liu, Sujit Dey, Fatih Ulupinar, Michael Luby, and Yinian Mao. Deriving and validating user experience model for dash video streaming. *IEEE Transactions on Broadcasting*, 61(4):651–665, 2015.

[87] K Manasa and Sumohana S Channappayya. An optical flow-based full reference video quality assessment algorithm. *IEEE Trans. on Image Process.*, 25(6):2480–2492, 2016.

[88] Ahmed Mansy, Bill Ver Steeg, and Mostafa Ammar. Sabre: A client based technique for mitigating the buffer bloat effect of adaptive video flows. *ACM Multimedia Systems Conference*, pages 214–225, 2013.

[89] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. Neural Adaptive Video Streaming with Pensieve. *Proc. of SIGCOMM*, pages 197–210, 2017.

[90] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Industrial and Applied Mathematics*, 1963.

[91] Konstantin Miller, Emanuele Quacchio, Gianluca Gennari, and Adam Wolisz. Adaptation algorithm for adaptive streaming over HTTP. *International Packet Video Workshop*, pages 173–178, 2012.

[92] Terence C Mills. *Time Series Techniques for Economists*. Cambridge University Press, 1991.

[93] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Trans. on Image Process.*, 21(12):4695–4708, 2012.

[94] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2013.

[95] Decebal C Mocanu, Jeevan Pokhrel, Juan Pablo Garella, Janne Seppänen, Eirini Liotou, and Manish Narwaria. No-reference video quality measurement: added value of machine learning. *Journal of Electronic Imaging*, 24(6):061208, 2015.

[96] Ricky KP Mok, Edmond WW Chan, and Rocky KC Chang. Measuring the quality of experience of HTTP video streaming. *IEEE International Symposium on Integrated Network Management and Workshops*, pages 485–492, 2011.

[97] Ricky KP Mok, Xiapu Luo, Edmond WW Chan, and Rocky KC Chang. QDASH: a QoE-aware DASH system. *Multimedia Systems Conference*, pages 11–22, 2012.

[98] Anush K Moorthy and Alan C Bovik. Perceptually significant spatial pooling techniques for image quality assessment. *Electronic Imaging*,

2009.

[99] Anush Krishna Moorthy and Alan Conrad Bovik. Visual quality assessment algorithms: What does the future hold? *Multimedia Tools and Applications*, 51(2):675–696, 2011.

[100] Anush Krishna Moorthy, Lark Kwon Choi, Alan Conrad Bovik, and Gustavo De Veciana. Video quality assessment on mobile devices: Subjective, behavioral and objective studies. *IEEE Journal on Selected Topics in Signal Processing*, 6(6):652–671, 2012.

[101] Kumpati S Narendra and Kannan Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on neural networks*, 1(1):4–27, 1990.

[102] LIVE Netflix Video Quality of Experience Database, 2017.

[103] Ozgur Oyman and Sarabjot Singh. Quality of experience for HTTP adaptive streaming services. *Communications Magazine*, 50(4), 2012.

[104] R. Pantos and W. May. HTTP live streaming, 2016.

[105] Jincheol Park, Kalpana Seshadrinathan, Sanghoon Lee, and Alan Conrad Bovik. Video quality pooling adaptive to perceptual distortion severity. *IEEE Trans. on Image Process.*, 22(2):610–620, 2013.

[106] Ricardo R Pastrana-Vidal and Jean-Charles Gicquel. A no-reference video quality metric based on a human assessment model. *International*

Workshop on Video Processing and Quality Metrics for Consumer Electronics, 2007.

[107] Ricardo R Pastrana-Vidal, Jean Charles Gicquel, Catherine Colomes, and Hocine Cherifi. Sporadic frame dropping impact on quality perception. *SPIE IS&T Conf. on Electronic Imaging*, 2004.

[108] Jonathan W Peirce. Psychopy-psychophysics software in python. *Journal of neuroscience methods*, 162(1):8–13, 2007.

[109] Margaret H Pinson, Lark Kwon Choi, and Alan Conrad Bovik. Temporal video quality model accounting for variable frame delay distortions. *IEEE Transactions on Broadcasting*, 60(4):637–649, 2014.

[110] Margaret H Pinson, Lucjan Janowski, Romuald Pépion, Quan Huynh-Thu, Christian Schmidmer, Phillip Corriveau, Audrey Younkin, Patrick Le Callet, Marcus Barkowsky, and William Ingram. The Influence of Subjects and Environment on Audiovisual Subjective Tests: An International Study. *IEEE Journal of Selected Topics in Signal Processing*, 6(6):640–651, 2012.

[111] Margaret H Pinson and Stephen Wolf. A new standardized method for objectively measuring video quality. *IEEE Transactions on Broadcasting*, 50(3):312–322, 2004.

[112] R. Polikar. Ensemble based systems in decision making. *IEEE Circ. Syst. Mag.*, 6(3):21–45, 2006.

[113] Nikolay Ponomarenko, Flavia Silvestri, Karen Egiazarian, Marco Carli, Jaakko Astola, and Vladimir Lukin. On between-coefficient contrast masking of DCT basis functions. *International Workshop on Video Processing and Quality Metrics*, 2007.

[114] Javier Portilla, Vasily Strela, Martin J Wainwright, and Eero P Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Trans. Image Process.*, 12(11):1338–1351, 2003.

[115] Yining Qi and Mingyuan Dai. The effect of frame freezing and frame skipping on video quality. *International Conference on Intelligent Information Hiding and Multimedia*, 2006.

[116] Alexander Raake. Short-and long-term packet loss behavior: towards speech quality prediction for arbitrary loss distributions. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(6):1957–1968, 2006.

[117] Alexander Raake, Marie-Neige Garcia, Werner Robitza, Peter List, Steve Göring, and Bernhard Feiten. A bitstream-based, scalable video-quality model for HTTP adaptive streaming: ITU-T P. 1203.1. *International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6, 2017.

[118] Benjamin Rainer and Christian Timmerer. Quality of experience of web-based adaptive http streaming clients in real-world environments

using crowdsourcing. *Works. on Des., Qual. and Deploym. of Adapt. Vid. Stream.*, pages 19–24, 2014.

[119] Abdul Rehman and Zhou Wang. Perceptual experience of time-varying video quality. In *International Workshop on Quality of Multimedia Experience*, pages 218–223, 2013.

[120] Abdul Rehman, Kai Zeng, and Zhou Wang. Display device-adapted video quality-of-experience assessment. *SPIE IS&T Conf. on Electronic Imaging*, 2015.

[121] Iain E Richardson. *The H. 264 advanced video compression standard*. John Wiley & Sons, 2011.

[122] Haakon Riiser, Paul Vigmostad, Carsten Griwodz, and Pål Halvorsen. Commute path bandwidth traces from 3G networks: analysis and applications. *ACM Multimedia Systems Conference*, pages 114–118, 2013.

[123] Werner Robitza, Marie-Neige Garcia, and Alexander Raake. A modular HTTP adaptive streaming QoE model—Candidate for ITU-T P. 1203 ("P. NATS"). *International Conference on Quality of Multimedia Experience*, pages 1–6, 2017.

[124] Demostenes Z Rodriguez, Julia Abrahao, Dante C Begazo, Renata L Rosa, and Graca Bressan. Quality metric to assess video streaming service over TCP considering temporal location of pauses. *IEEE Transactions on Consumer Electronics*, 58(3):985–992, 2012.

[125] Demóstenes Z Rodríguez, Zhou Wang, Renata L Rosa, and Graça Bressan. The impact of video-quality-level switching on user quality of experience in dynamic adaptive streaming over HTTP. *EURASIP Journal on Wireless Communications and Networking*, 2014(1):216, 2014.

[126] Daniel L Ruderman. The statistics of natural images. *Network: computation in neural systems*, 5(4):517–548, 1994.

[127] Michele A Saad, Alan C Bovik, and Christophe Charrier. Blind prediction of natural video quality. *IEEE Trans. on Image Process.*, 23(3):1352–1365, 2014.

[128] Hasim Sak, Andrew W Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *INTERSPEECH*, pages 338–342, 2014.

[129] RM Sakia. The box-cox transformation technique: a review. *The statistician*, pages 169–178, 1992.

[130] Shahid Satti, Christian Schmidmer, Matthias Obermann, Roland Bitto, Lavita Agarwal, and Michael Keyhl. P. 1203 evaluation of real OTT video services. *International Conference on Quality of Multimedia Experience*, pages 1–3, 2017.

[131] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack. Study of subjective and objective quality assessment of video. *IEEE Trans. on Image Process.*, 19(6):1427–1441, June 2010.

[132] Kalpana Seshadrinathan and Alan C Bovik. Temporal hysteresis model of time varying subjective video quality. *IEEE Int'l Conf. Acoust., Speech and Signal Process.*, 2011.

[133] Kalpana Seshadrinathan and Alan C Bovik. A structural similarity metric for video based on motion models. In *IEEE Int'l Conf. Acoust., Speech and Signal Process., Honolulu, HI*, June 2007.

[134] Kalpana Seshadrinathan and Alan Conrad Bovik. Motion tuned spatio-temporal quality assessment of natural videos. *IEEE Trans. on Image Process.*, 19(2):335–350, 2010.

[135] Michael Seufert, Sebastian Egger, Martin Slanina, Thomas Zinner, Tobias Hobfeld, and Phuoc Tran-Gia. A Survey on Quality of Experience of HTTP Adaptive Streaming. *IEEE Communications Surveys & Tutorials*, 17(1):469–492, 2015.

[136] Michael Seufert, Martin Slanina, Sebastian Egger, and Meik Kottkamp. To pool or not to pool: A comparison of temporal pooling methods for HTTP adaptive video streaming. *International Workshop on Quality of Multimedia Experience*, 2013.

[137] Hamid R Sheikh and Alan C Bovik. A visual information fidelity approach to video quality assessment. *Workshop on Video Processing and Quality Metrics for Consumer Electronics*, 2005.

[138] Hamid R Sheikh and Alan C Bovik. Image information and visual quality. *IEEE Trans. on Image Process.*, 15(2):430–444, 2006.

[139] Sidney Siegel. *Non-parametric Statistics for the Behavioral Sciences.* McGraw-Hill, 1956.

[140] Hava T Siegelmann, Bill G Horne, and C Lee Giles. Computational capabilities of recurrent NARX neural networks. *IEEE Trans. on Sys., Man, and Cyber.*, 27(2):208–215, 1997.

[141] Kamal Deep Singh, Yassine Hadjadj-Aoul, and Gerardo Rubino. Quality of experience estimation for adaptive HTTP/TCP video streaming using H. 264/AVC. *IEEE Consumer Communications and Networking Conference*, 2012.

[142] Jacob Søgaard, Muhammad Shahid, Jeevan Pokhrel, and Kjell Brunnström. On subjective quality assessment of adaptive video streaming via crowdsourcing and laboratory based experiments. *Multimedia tools and applications*, 76(15):16727–16748, 2017.

[143] Rajiv Soundararajan and Alan C Bovik. Video quality assessment by reduced reference spatio-temporal entropic differencing. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(4):684–694, 2013.

[144] Kevin Spiteri, Rahul Urgaonkar, and Ramesh K Sitaraman. BOLA: Near-optimal bitrate adaptation for online videos. *IEEE International*

Conference on Computer Communications, pages 1–9, 2016.

[145] N. Staelens, J. De Meulenaere, M. Claeys, G. Van Wallendael, W. Van den Broeck, J. De Cock, R. Van de Walle, P. Demeester, and F. De Turck. Subjective quality assessment of longer duration video sequences delivered over http adaptive streaming to tablet devices. *IEEE Transactions on Broadcasting*, 60(4):707–714, Dec 2014.

[146] Nicolas Staelens, Jonas De Meulenaere, Maxim Claeys, Glenn Van Wallendael, Wendy Van den Broeck, Jan De Cock, Rik Van de Walle, Piet Demeester, and Filip De Turck. Subjective quality assessment of longer duration video sequences delivered over http adaptive streaming to tablet devices. *IEEE Transactions on Broadcasting*, 60(4):707–714, 2014.

[147] James H Stock and Mark W Watson. Combination forecasts of output growth in a seven-country data set. *Journal of Forecasting*, 23(6):405–430, 2004.

[148] Yi Sun, Xiaoqi Yin, Junchen Jiang, Vyas Sekar, Fuyuan Lin, Nanshu Wang, Tao Liu, and Bruno Sinopoli. Cs2p: Improving video bitrate selection and adaptation with data-driven throughput prediction. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 272–285. ACM, 2016.

[149] Samira Tavakoli, Kjell Brunnström, Jesús Gutiérrez, and Narciso García. Quality of experience of adaptive video streaming: Investigation in ser-

vice parameters and subjective quality assessment methodology. *Signal Processing: Image Communication*, 39:432–443, 2015.

[150] Samira Tavakoli, Sebastian Egger, Michael Seufert, Raimund Schatz, Kjell Brunnström, and Narciso García. Perceptual quality of HTTP adaptive streaming strategies: Cross-experimental analysis of multi-laboratory and crowdsourced subjective studies. *IEEE Journal on Sel. Areas in Comm.*, 34(8):2141–2153, 2016.

[151] Truong Cong Thang, Hung T Le, Anh T Pham, and Yong Man Ro. An evaluation of bitrate adaptation methods for HTTP live streaming. *IEEE Journal on Selected Areas in Communications*, 32(4):693–705, 2014.

[152] John W Tukey. *Exploratory Data Analysis*. 1977.

[153] Sven Van Kester, Tonjiao Xiao, Robert E Kooij, OK Ahmed, and K Brunnstrom. Estimating the impact of single and multiple freezes on video quality. *Proc. of SPIE*, 7865, 2011.

[154] Phong V Vu and Damon M Chandler. Vis3: an algorithm for video quality assessment via analysis of spatial and spatiotemporal slices. *J. of Electron. Imaging*, 23(1):013016–013016, 2014.

[155] Phong V Vu, Cuong T Vu, and Damon M Chandler. A spatiotemporal most-apparent-distortion model for video quality assessment. *IEEE International Conference on Image Processing*, 2011.

[156] Martin J Wainwright and Eero P Simoncelli. Scale mixtures of gaussians and the statistics of natural images. In *Neural Information Processing Systems*, 1999.

[157] Cong Wang, Amr Rizk, and Michael Zink. Squad: A spectrum-based quality adaptation for dynamic adaptive streaming over HTTP. *International Conference on Multimedia Systems*, page 1, 2016.

[158] Haiqiang Wang, Ioannis Katsavounidis, Jiantong Zhou, Jeonghoon Park, Shawmin Lei, Xin Zhou, Man-On Pun, Xin Jin, Ronggang Wang, Xu Wang, et al. Videoset: A large-scale compressed video quality dataset based on JND measurement. *J. Visual Commun. Image Repres.*, 46:292–302, 2017.

[159] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. on Image Process.*, 13(4):600–612, 2004.

[160] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. *Asilomar Conference on Signals, Systems and Computers*, 2003.

[161] Stefan Winkler. Analysis of public image and video databases for quality assessment. *IEEE Journal of Selected Topics in Signal Processing*, 6(6):616–625, 2012.

[162] Jingtao Xu, Peng Ye, Yong Liu, and David Doermann. No-reference video quality assessment via feature learning. pages 491–495, 2014.

[163] Jingteng Xue, Dong-Qing Zhang, Heather Yu, and Chang Wen Chen. Assessing quality of experience for adaptive http video streaming. *IEEE International Conference on Multimedia and Expo Workshops*, 2014.

[164] Wufeng Xue, Lei Zhang, Xuanqin Mou, and Alan C Bovik. Gradient magnitude similarity deviation: a highly efficient perceptual image quality index. *IEEE Trans. on Image Process.*, 23(2):684–695, 2014.

[165] Fuzheng Yang, Shuai Wan, Yilin Chang, and Hong Ren Wu. A novel objective no-reference metric for digital video quality assessment. *IEEE Signal Processing Letters*, 12(10):685–688, 2005.

[166] Kai-Chieh Yang, Clark C Guest, Khaled El-Maleh, and Pankaj K Das. Perceptual temporal quality metric for compressed video. *IEEE Transactions on Multimedia*, 9(7):1528–1535, 2007.

[167] H. Yeganeh, R. Kordasiewicz, M. Gallant, D. Ghadiyaram, and A.C. Bovik. Delivery quality score model for internet video. *IEEE International Conference on Image Processing*, 2014.

[168] Hojatollah Yeganeh, Farzad Qassemi, and Hamid R Rabiee. Joint effect of stalling and presentation quality on the quality-of-experience of streaming videos. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 310–314. IEEE, 2017.

[169] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. A control-theoretic approach for dynamic adaptive video streaming over HTTP. *Computer Communication Review*, 45(4):325–338, 2015.

[170] X. Yu, **C. G. Bampis**, P. Gupta, and A. C. Bovik. Predicting Encoded Picture Quality in Two Steps is a Better Way. `https://arxiv.org/abs/1801.02016`[PDF] `https://github.com/xiangxuyu/2stepQA`[Code].

[171] G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.

[172] Chao Zhou, Chia-Wen Lin, Xinggong Zhang, and Zongming Guo. Buffer-based smooth rate adaptation for dynamic HTTP streaming. *Asia-Pacific Sig. Inform. Process. Assoc. Ann. Summ. Conf.*, pages 1–9, 2013.

[173] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks: many could be better than all. *Artificial intelligence*, 2002.

Vita

Christos Bampis received the M.Eng. Diploma in Electrical and Computer Engineering from the National Technical University of Athens (NTUA) in 2014 and the M.Sc.Eng. in Electrical and Computer Engineering from The University of Texas at Austin in 2016. He is currently pursuing the Ph.D. degree with the Laboratory for Image and Video Engineering (LIVE) at The University of Texas at Austin. His research interests include image and video quality assessment and quality of experience with an emphasis on adaptive video streaming.

Contact: cbampis@gmail.com

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.