Towards Perceptually Optimized Adaptive Video Streaming-A Realistic Quality of Experience Database

Christos G. Bampis, Zhi Li, Ioannis Katsavounidis, Te-Yuan Huang, Chaitanya Ekanadham, and Alan C. Bovik

***Abstract*— Measuring Quality of Experience (QoE) and integrating these measurements into video streaming algorithms is a multi-faceted problem that fundamentally requires the design of comprehensive subjective QoE databases and objective QoE prediction models. To achieve this goal, we have recently designed the LIVE-NFLX-II database, a highly-realistic database which contains subjective QoE responses to various design dimensions, such as bitrate adaptation algorithms, network conditions and video content. Our database builds on recent advancements in content-adaptive encoding and incorporates actual network traces to capture realistic network variations on the client device. The new database focuses on low bandwidth conditions which are more challenging for bitrate adaptation algorithms, which often must navigate tradeoffs between rebuffering and video quality. Using our database, we study the effects of multiple streaming dimensions on user experience and evaluate video quality and quality of experience models and analyze their strengths and weaknesses. We believe that the tools introduced here will help inspire further progress on the development of *perceptually-optimized* client adaptation and video streaming strategies. The database is publicly available at http://live.ece.utexas.edu/research/LIVE_NFLX_II/live_nflx_plus.html.**

***Index Terms*— Adaptive video streaming, subjective testing, perceptual video quality, QoE prediction.**

I. Introduction

HTTP-BASED adaptive video streaming (HAS) is becoming the *de facto* standard for modern video streaming services, such as Netflix and YouTube. The main idea behind HAS is to encode video content into multiple streams of various bitrate and quality levels, and to allow for client-driven stream selection to meet the time-varying network bandwidth. Under this setting, the client device is responsible for deciding on the bitrate/quality level of the video chunk to be played next. These client decisions are usually based on past network throughput values, future throughput estimates and other client-related information, e.g., the buffer level [2].

In HAS, TCP is used as the transfer protocol; hence packet loss is not an issue [3]. Nevertheless, depending on the available bandwidth, client devices may adapt to different quality levels and hence users may suffer from compression/scaling artifacts and rebuffering. When the available bandwidth drops, a client may use a higher compression ratio and/or a lower encoding resolution to reduce the video bitrate, leading to compression and/or scaling artifacts [3], which do not interrupt video playback. Compression artifacts are usually visible as blocky artifacts leading to loss of details and texture information, whereas scaling artifacts are typically visible as blurry edges and reduced sharpness. If the throughput reaches a very low value and the buffer is emptied, a client must pause its video playback (video rebuffering), wait for the network to recover, and fill the buffer with video data before resuming play. There are alternatives to reduce video bitrate, when bandwidth drops, e.g., using a lower frame rate. We focus on changes in video quality, such as compression, scaling artifacts [3] and rebuffering. Nevertheless, there are other important aspects relevant to video streaming, such as start-up delays.

Here we will define video quality to be the instantaneous or overall quality of a video sequence in reference to an encoder's source. Changes in video quality, along with rebuffering, can adversely affect user Quality of Experience (QoE), i.e., the overall level of user satisfaction [4] while viewing streaming content. Being able to predict QoE, and act upon those predictions, is important for improving the overall quality of experience. Towards this goal, we can design algorithms to optimize the QoE while effectively utilizing the available bandwidth and, subsequently, reducing operational costs. Modeling QoE is a difficult task, since it is affected by many complex and sometimes inaccessible factors, while obtaining ground truth QoE data that reflects these many factors is difficult. In the rest of this work, we use the term QoE to refer to the opinion collected from subjects when asked about the quality of experience they had during viewing.

Understanding and predicting QoE is an emerging research area [5]–[13]. Recently, there is raised interest in building more sophisticated QoE prediction models. For example, an LSTM approach was used to predict QoE in [14] and a knowledge-driven QoE model was proposed in [15]. A survey on the topic of QoE modeling and its challenges can be found in [16]. On a similar note, the recent work in [17] studied multiple facets of video streaming QoE, such as start-up delay, video quality changes, rebuffering and resolution switching.

Regarding the design of subjective studies, a number of existing QoE studies do not fully capture important aspects of

Manuscript received April 2, 2020; revised October 22, 2020 and February 21, 2021; accepted April 5, 2021. Date of publication April 20, 2021; date of current version May 26, 2021. This article was presented at the IS&T International Symposium on Electronic Imaging 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Charith Abhayaratne. *(Corresponding author: Christos G. Bampis.)*

Christos G. Bampis was with the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712 USA. He is now with Netflix Inc., Los Gatos, CA 95032 USA (e-mail: bampis@utexas.edu).

Zhi Li, Te-Yuan Huang, and Chaitanya Ekanadham are with Netflix Inc., Los Gatos, CA 95032 USA.

Ioannis Katsavounidis is with Facebook Inc., Menlo Park, CA 94025 USA.

Alan C. Bovik is with the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712 USA.

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2021.3073294, provided by the authors.

Digital Object Identifier 10.1109/TIP.2021.3073294

practical systems, e.g., they do not incorporate actual network measurements or client adaptive bitrate adaptation (ABR) algorithms. To this end, we built the LIVE-NFLX-II database, a large subjective QoE database that integrates perceptual video coding and quality assessment, using real measurements of network and buffer conditions, and client-based adaptation. To better understand QoE, we collected scores during viewing of video playback (continuous-time scores) as well as overall (endpoint) scores at the end of each playback. Compared to overall or evenly spaced QoE scores (e.g. every 10s.), continuous scores contain valuable granular temporal information that better reflects time-varying QoE. Further, continuous scores can also be used to train continuous QoE predictors that can guide bitrate adaptation algorithms.

We now describe the roadmap of our work (see also supplementary material for a detailed figure). We first generate a large set of end-user experiences on top of a comprehensive collection of video contents, network conditions and ABR algorithms. Then, we conduct a subjective study to build enhanced QoE metrics. Ultimately, these metrics can be used to inform better ABR algorithms or encoding strategies.

A unique characteristic of the new subjective database is that we exploit recent developments in large-scale video encoding and ABR algorithms. To generate compressed videos, we make use of an encoding optimization framework [18] that selects encoding parameters on a per-shot basis, guided by a state-of-the-art video quality assessment algorithm (VMAF) [3].

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 ABR algorithms, the database captures multiple streaming adaptation aspects, such as video quality fluctuations, rebuffering events of varying durations, and content types. The subjective data consists of both overall and continuous-time scores, making it ideal for training various QoE models. The video database is considerably larger than other public-domain video QoE databases [10], [19], [20].

The main observations from the collected data can be summarized as follows. A better bandwidth prediction model can improve most objective streaming metrics, such as the playout bitrate and the number and duration of rebuffers. Start-up is the most challenging part of a session for all ABR algorithms, since ABR algorithms have not built up the video buffer and hence network variations can easily reduce QoE. While this is in line with previous studies [2], we go a step further. The collected data shows that humans perceive these differences during start-up, even if they are forgiving and/or forgetful when an overall QoE score is recorded. These observations highlight the importance of temporal studies of QoE, especially during start-up, for practical applications.

From a QoE model development perspective, we trained an overall QoE predictor on the dataset and determined that QoE predictions were mostly influenced by average video quality, followed by rebuffering duration. We also trained G-NARX and G-RNN [21], two state-of-the-art continuous-time QoE prediction models, and evaluated them on the continuous-time subjective data collected in this database. We found that the prediction performance of these algorithms is promising, but that they still fall short in their ability to capture trends in human responses. This finding suggests the need for better models of human responses to these temporal phenomena.

The rest of this paper is organized into two main parts. The first part (Sections II, III and IV) describes the design and construction of the QoE database and the subjective test that we carried out. The second part (Sections V, VI and VII) focuses on our analysis of the database and the collected human subject data, along with an evaluation of existing QoE prediction models against the ground truth scores. Specifically, Section II gives an overview of previous QoE studies and Section III discusses the streaming database and streaming pipeline model. In Section IV, the subjective testing procedure is discussed and an objective analysis of the database is presented in Section V. Following that, Section VI studies the collected human opinion scores. Section VII evaluates video quality assessment (VQA) and QoE prediction models on the new dataset while Section VIII concludes with future work.

II. Related Work

Many databases have been designed towards advancing progress on the general problem of video quality [5], [6], [22]–[27] and streaming QoE [7]–[11], [13], [19], [28]–[32]. We give a brief overview of these previous studies and point out limitations of past work which we seek to address.

In [19], the time-varying quality of long HTTP streams was investigated. A set of three contents were used to create 15 distorted videos having durations of 5 minutes. Using the collected data, the authors studied the effects of time-varying video quality on QoE, such as the hysteresis/recency phenomenon. They also designed a QoE prediction model using a Hammerstein-Wiener model [33]. However, this study did not consider the interplay between time-varying video quality and rebuffering events and/or client ABR algorithms. In [29], an experimental comparison among three HTTP-based clients was carried out. This was the first crowdsourced QoE study on Dynamic Adaptive Streaming over HTTP (DASH), which showed that video bitrate and the number of stalls are the main influence factors on subject QoE. However, only one video content was used, and continuous QoE was not studied.

More recently, in [10] and [34], the effects of rebuffering and quality changes were systematically studied on different content types, under simulated network conditions and using ABR algorithms. Given the shorter durations of the videos in these databases, these works focused more on overall QoE rather than on continuous QoE effects. In [11], the effects of compression and rebuffering on continuous QoE were studied on $\approx$1 minute videos. Interestingly, the authors found that on video contents that required more bits to be encoded, compression artifacts were not preferred over rebuffering events. A simple design was used to model buffer and network conditions, using a set of eight pre-defined bandwidth patterns. Therefore, only eight distortions were generated per content. The database is not available in its entirety.

A common approach that most previous efforts have taken is to systematically control and simulate network conditions, e.g., as suddenly decreasing or gradually increasing bandwidth patterns. By varying the position and the length of these events, it is indeed possible to recreate intuitive network patterns. However, real network conditions are far more complex, and hence challenge ABR algorithms to a greater extent. In this work, we have undertaken a more realistic approach, where real network traces have been used to drive the database generation. This is arguably a risky choice; choosing a specific set of network traces in our design may not generalize to unseen network types. However, we are confident that by carefully selecting the traces, this approach will generalize better than a hand-crafted way.

TABLE I
HIGH-LEVEL COMPARISON WITH OTHER RELEVANT VIDEO STREAMING SUBJECTIVE STUDIES. FOR LIVE-NFLX-II, EVERY TEST VIDEO IS WATCHED 23.2 TIMES ON AVERAGE

Description	[29]	[9]	[7]	[32]	[31]	[19]	[10]	[20]	[11]	[34]	LIVE-NFLX-II
client adaptation	X									X	X
continuous QoE		X				X		X	X		X
actual traces	X										X
buffer model	X								X	X	X
public					X	X	X	X		X	X
> 400 videos										X	X
rebuff. + quality	X	X		X			X		X	X	X
content-aware ladder			X		X						X

Further, in previous studies, fixed bitrate ladders were commonly used, without considering content-aware encoding strategies which are gaining popularity within the streaming video industry and the research community [18], [35]. The main idea of a content-aware ladder is that, due to differences in the characteristics of each content, not all video contents need the same amount of bits to be encoded at the same quality level. Some contents require a larger number of bits to achieve the same quality level compared to others. For example, video contents containing rich spatial textures or significant motion require more bits to encode, as compared to relatively simpler scenes with little motion or few textures. Therefore, a content-aware ladder, which takes content characteristics into account, can achieve bitrate savings for streaming providers and better video quality for consumers.

We compare the differences between the LIVE-NFLX-II database and prior, similar resources in Table I. Notably, [34] has some similarities with this work, in that it is a sizable video database that studies ABR algorithms and is publicly available. Nevertheless, [34] does not study continuous QoE, which is an important aim of our work. As already mentioned, continuous QoE is an essential part of better understanding streaming QoE, and designing continuous QoE prediction models. Furthermore, actual network traces nor a content-aware ladder were used in [34], both of which are characteristics of realistic video streaming applications. To summarize, we believe that the new LIVE-NFLX-II database takes a step further towards more realistic designs of adaptive video streaming scenarios.

III. RECREATING A COMPREHENSIVE END-USER EXPERIENCE

A. Overview of the Streaming System

We designed 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. To overcome the limitations of previous QoE studies, we built our database using a highly realistic adaptive streaming pipeline model, which comprises four main modules, as shown in Fig. 1. The modules include an encoding module, a video quality module, a network transmission module and a client-based video playout module.

The encoding module constructs a content-driven bitrate ladder which is then fed into the Dynamic Optimizer (DO) [18]: a state-of-the-art encoding optimization approach, which determines the encoding parameters (encoding resolution and Quantization Parameter - QP) to produce compressed videos of optimized quality. The video quality module performs VMAF [3] quality measurements that drive the encoding and

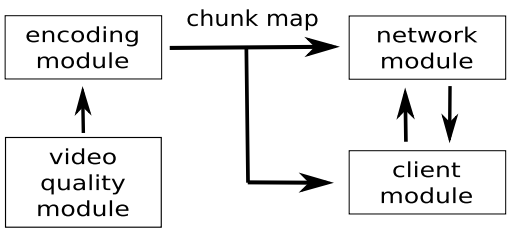

Fig. 1. Overview of the adaptive streaming pipeline.

client modules. We used the latest VMAF (version 0.6.1), which was trained as described in [36]. The VMAF quality measurements are stored in a chunk map and made available on the client side for client bitrate adaptation. A chunk map contains information about every encoded video segment (see Fig. 1 in the supplementary material). 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. In the supplementary material, we provide more details regarding the encoding and video quality modules and the streaming pipeline model that we built.

This streaming model allowed us to recreate a comprehensive end-user experience by focusing on three streaming dimensions: encoding, network throughput and the choice of ABR algorithm. To study each of these dimensions, we incorporated 15 video contents, 7 actual network traces and 4 adaptation algorithms, yielding 420 video streams. Next, we explore the diverse characteristics of each dimension.

B. Video Contents

To design a diverse encoding space, we considered multiple video contents and encoded them at multiple bitrate values (bitrate ladder). We collected 15 video contents, which span a diverse set of content genres, including action, documentary, sports, animation and video games. The video sequences also contain computer-generated content, such as Blender [37] animation and video games. The videos were shot/rendered under different lighting conditions ranging from bright scenes (Skateboarding) to darker ones (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). Contents having source resolutions larger than 1920×1080 and/or frame rates larger than 30 fps were downsampled to 1920×1080 and/or 30 fps. For reference, Table II provides the acronyms for each content in the database. More details on the content characteristics and sample frames for each content are included as supplementary material.

An alternate description of encoding/content diversity is encoding complexity. One approach to describe content is via the spatial and temporal activity (SI-TI) plot [27], but we chose to use a description that more closely relates to the encoding behavior of each content. Contents with high motion and high spatial activity (textures) tend to be harder to compress, hence subjective scores are generally lower for those contents, given a fixed number of available bits.

To measure content encoding complexity, we used the bitrate produced by a constant-quality encoding mode [38], following the process used in [39]. Specifically, we generated

TABLE II
ACRONYMS OF CONTENTS USED IN THE DATABASE

Video Source	ID	Video Source	ID	Video Source	ID
AirShow	AS	ElFuenteDance	ED	SkateBoarding	SB
AsianFusion	AF	ElFuenteMask	EM	Soccer	SO
Chimera1102353	CD	GTA	GTA	Sparks	SP
Chimera1102347	CF	MeridianConversation	MC	TearsOfSteelRobot	TR
CosmosLaundromat	CL	MeridianDriving	MD	TearsOfSteelStatic	TS

one-pass, fixed constant rate factor (CRF) compressed videos using libx264 [38]. CRF is a constant-quality encoding mode that aims to maintain a certain level of quality by varying the amount of quantization accordingly. It should be noted that, while the CRF mode attempts to achieve constant quality, it may not always be able to produce constant perceptual quality. The encoding resolution was set to 1920×1080 and the CRF parameter was set to 23. After generating these compressed videos, we then measured the resulting bitrate (see Fig. 2). 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. We note that we could have also used the results of the DO optimization to perform this content encoding complexity analysis, but preferred the aforementioned CRF approach as a simple and more intuitive alternative.

C. Video Encoding

A comprehensive encoding space design requires a wide range of encoding bitrates and video quality levels. To this end, we derived a target bitrate ladder, i.e., a set of possible bitrate values, one for each content, using VMAF [3] to generate equally spaced (in terms of VMAF) bitrate points, then fed these bitrate points to DO [18]. The per-content bitrate ladders we produced covered various encoding rates ranging from about 150 kbps up to almost 6 Mbps. It should be noted that the construction of the encoding bitrate ladder does not depend on the selection of the network traces. The encoding bitrate ladder design is orthogonal to the actual network conditions which are not known *a priori*. We provide additional details of the encoding ladder in the supplementary material.

The DO framework selects the encoding resolution and QP for each shot, such that the overall quality (as measured by VMAF) is maximized for a given target bitrate. We used 6 encoding resolutions: 384×216, 480×270, 640×360, 960×540, 1280×720, 1920×1080 and 10 QP values: starting from 43 (worst quality) to 16 (best quality), in steps of 3. However, for display purposes, all compressed videos were upsampled to 1920×1080 to match the display device resolution.

It should be noted that the video sequences are approximately 25 seconds long and typically contain multiple shots. This design choice is different from commonly used single-shot 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 shots, 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 shots also aligns well

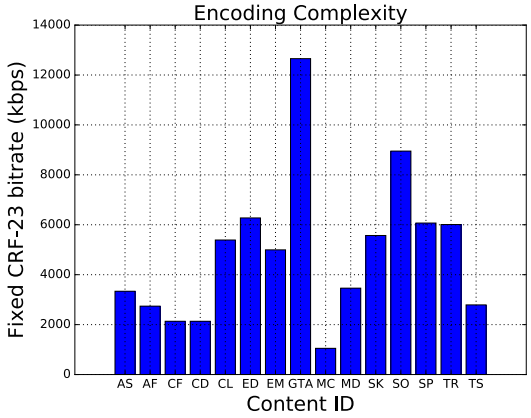

Fig. 2. Content (encoding) complexity for all 15 contents.

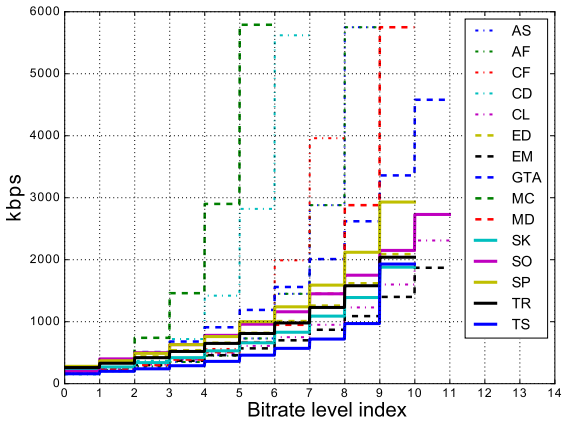

Fig. 3. Video bitrate ladders across different video contents in the LIVE-NFLX-II database.

with the DO encoding approach [18], which leverages different shot complexities to achieve better encoding efficiency.

Figure 3 shows the encoding ladders that were generated by the DO optimization framework. It can be observed that there are various encoding rates ranging from about 150 kbps up to almost 6 Mbps. The low bitrate range, i.e., 150 kbps to 1 Mbps is sampled more heavily, which aligns well with our deep interest in challenging network conditions.

D. Network Simulation

Until now, we have only considered the encoding dimension in the video streaming design space. Importantly, the number of available bits is not constant in a streaming session and network resources can vary significantly. To capture network variability effects, we manually selected 7 network traces from the HSDPA dataset [40], [41], which contains actual 3G traces collected from multiple travel routes in Norway, using various means of transportation, e.g., car, tram and train, together with different network conditions. This dataset has been widely used to compare ABR algorithms [42] and is suitable for modeling challenging, low-bandwidth network conditions.

Table III provides some details on the network traces we used. There are multiple types of transportation and multiple routes included in the selected traces, which cover the range of 9 kbps up to almost 3900 kbps.

As shown in Fig. 4, the selected traces are approximately 40 seconds long and have varying network behaviors. For example, the TLJ trace has the lowest average bandwidth but does not vary much over time, while the MKJ trace has a much

TABLE III

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

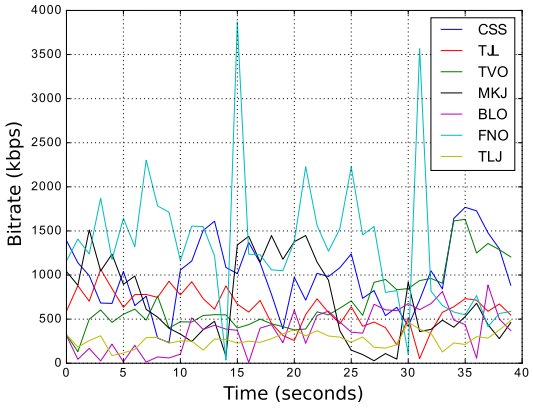

Fig. 4. Network traces used in our streaming pipeline model. We refer the reader to Table III for more details on the acronyms.

more volatile behavior than TLJ. The network traces densely cover bandwidths up to 1Mbps, and there are also samples in the 1Mbps-3Mbps range.

To model challenging network conditions and train QoE models that can reliably measure QoE under such network conditions, we chose traces that are likely to cause sudden bitrate/quality changes and rebuffers even if the average bandwidth is relatively high. Figure 5 shows all 7 pairs of $(\mu_B, \sigma_B/\mu_B)$, where μ_B and σ_B denote the average and standard deviation of the available bandwidth over time. The ratio σ_B/μ_B is the coefficient of variation, which we use to describe network volatility. This design may not necessarily cover all possible combinations (e.g. low μ_B and high σ_B or high μ_B and low σ_B), but is challenging in that both low bandwidth conditions and highly varying network conditions are captured. Even for a higher (on average) bandwidth condition, e.g., FNO, the network variations can potentially lead to quality changes and rebuffers.

At this point, we take a step back to recognize that 3G networks are being migrated to 4G, which could mean that the network traces in our dataset become outdated. However, our work here is focused on collecting a set of networks traces which challenge the behavior of ABR algorithms, and that emphasize actual tradeoffs between video quality and rebuffering at low bitrates and/or variable networks. Importantly, there are many growing markets for mobile streaming, e.g., in developing countries, where bandwidth resources are limited and/or unpredictable, and where the selected network traces remain highly relevant. There are no publicly-available 4G network traces, due to proprietary and competitiveness reasons. Also, we believe that 4G networks still carry some of the constraints of 3G networks due to mobile carrier networks. In practice, in dashboard monitoring, WiFi would be treated

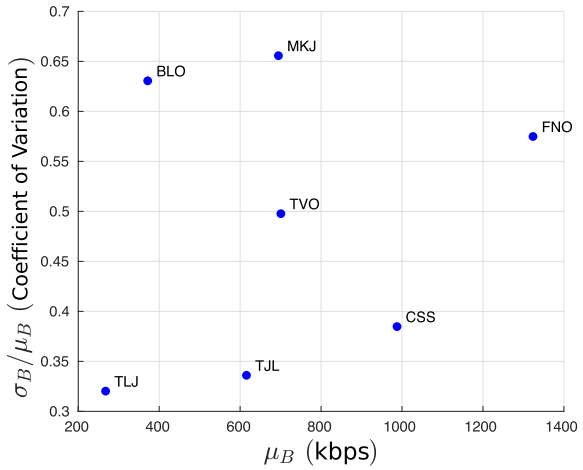

Fig. 5. Plot of $(\mu_B, \sigma_B/\mu_B)$ pairs for the 7 network traces. We refer the reader to Table III for more details on the acronyms.

separately from cellular, but cellular traffic would not generally be broken further down into 3G and 4G. This is related to the fact that latencies for 3G and 4G are significantly higher compared to WiFi traces. A similar observation can be made with video client applications, which typically do not provide a different rate limit between 3G and 4G. Lastly, regarding throughput and RTT, we believe that 3G and 4G can be considered to fall into the same bucket. Generally, while the technologies underlying 3G and 4G are very different, the network traffic characteristics are similar. Naturally, as 4G network data becomes publicly available, and years further on, 5G data, it will be interesting to refine these kinds of studies taking that data into account.

E. Client ABR Algorithm

In client-based video streaming, the client is responsible for requesting the next chunk to be played. To decide the appropriate quality representation, the client module is aware of its buffer status, and may estimate future bandwidth (based on past client measurements). The client may also have information regarding the bitrate/quality levels for each video segment. In practice, this can be implemented as part of the manifest exchange between server and client.

Client-based ABR strategies can be broadly classified as: throughput-based [43]–[45], buffer-based [2], [46]–[48] and hybrid/control-theoretical approaches [49]–[55]. 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 [51]. Recently, raw network observations fed to neural networks were used to achieve adaptive rate selection [42].

The design space of adaptation algorithms is very large, and hence we selected four representative adaptation algorithms. Each one of them focuses on different design aspects, such as preserving buffer status, maximizing download bitrate, or mediating between chunk quality and buffer level. Table IV defines some of the acronyms used hereafter.

We implemented the buffer-based (BB) approach from [2], which decides the rate of the next chunk to be played, as a function of the current buffer occupancy. We included this

TABLE IV
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	seconds	1	QB, OQB
B_h	max allowed buffer size	seconds	10	QB, OQB
T_a	actual throughput	kbps	varies	BB, RB, QB, OQB
h	horizon	seconds	10	QB, OQB
B_t	target buffer	seconds	3	QB, OQB
r	reservoir for BB	seconds	5	BB
c	cushion for BB	seconds	4.5	BB
w	throughput estimation window	# chunks	5	RB, QB, OQB

algorithm because it is simple to implement and is commonly evaluated or cited in the ABR literature. A reservoir of $r = 5$ seconds and a cushion of $c = 4.5$ seconds was used. We manually selected these parameters to achieve satisfying performance on a set of tests that we carried out offline. 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 angle, 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 future throughput, an average of $w = 5$ past chunks is computed. Selecting w can affect adaptation performance, if the network varies significantly. A low value of w could be insufficient to produce a reliable bandwidth estimate, while a large w might include redundant past samples and have diminishing impact. One downside of the RB approach is that, when channel bandwidth varies significantly, it may lead to excessive rebuffering and aggressive bitrate/quality switching.

Using video bitrate as a proxy for quality may yield sub-optimal results; a complex shot (rich in spatial textures or motion) requires more bits to be encoded at the same quality compared to a static shot having a uniform background and low motion. Therefore, it is interesting to explore how well a quality-based (QB) adaptation algorithm will correlate against subjective scores. We relied on the consistent-quality adaptation algorithm presented in [56]. We use VMAF measurements (using the video quality module - see also Section B in the supplementary material) as a utility function to be maximized within a finite horizon h (in seconds). This was formulated as a dynamic programming (DP) problem solved at each step, which determines the chunk to be played next.

In our QB implementation, the network conditions are estimated similar to our RB implementation. We assume that future throughput (within the horizon h) will be equal to the average throughput over the past $w = 5$ chunks. However, different from RB, QB maximizes video quality in terms of VMAF, instead of video bitrate. 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 seconds). 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), the QB algorithm uses a "fallback" mode: if B_l cannot be achieved, then QB selects the lowest quality stream, while if B_h is surpassed, then QB pauses downloading until the buffer frees up and then downloads the highest available stream.

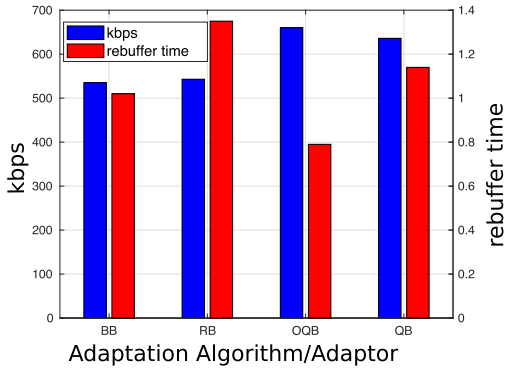

Fig. 6. Average bitrate and rebuffer time (in seconds) for all adaptors. We also recorded the percentage of time that the maximum encoding resolution (1080p) was achieved by each adaptor: BB: 34%, RB: 18%, OQB: 37% and QB: 33%.

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 exploited. For the latter, many adaptation algorithms assume that network conditions are constant over short time scales, and apply filtering using previous network measurements, as in QB. Since accurate knowledge of 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).

To demonstrate the diversity of ABR algorithms, Fig. 6 shows the average bitrate (in kbps) and rebuffering time for the 4 adaptation algorithms. We observed bitrate values in the range of 535 to 660 kbps and average rebuffering times from 0.8 to 1.35 seconds (see also Table V). We revisit the ABR algorithms by studying more QoE indicators in Section V-C.

F. Visualizing Quality for the Generated Video Streams

The combination of 15 different contents, 7 network conditions and 4 bitrate adaptation algorithms produced 420 video streams of time-varying qualities and various content and impairment characteristics. We visualize the quality changes over time for two contents in Fig. 7. It can be seen that different contents may have very different quality profiles, depending on the encoding complexity. Meanwhile, there can be multiple rebuffering events of varying duration, especially during start-up, since the video buffer may not be adequate to absorb network variations.

Given the comprehensive nature of the encoding, network conditions and ABR designs, we are able to create a rich streaming QoE database by conducting a large subjective test on human perception. Next, we describe the specifics of this test, which led to the creation of the LIVE-NFLX-II database.

IV. SUBJECTIVE TEST ON THE RECREATED EXPERIENCE

We conducted a single-stimulus continuous quality evaluation study [57] over a period of four weeks at The University of Texas at Austin's LIVE subjective testing lab. We collected overall and continuous-time QoE scores on an HP 1080p

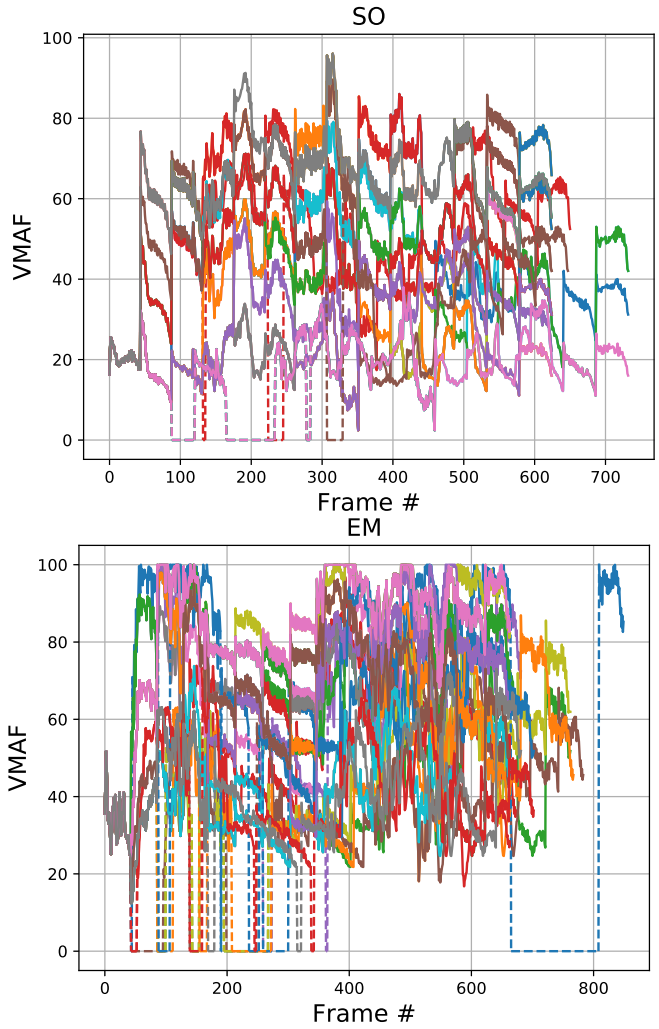

Fig. 7. VMAF changes over time for the Soccer (SO) and ElFuenteMask (EM) contents. There are 28 sequences (4 adaptors and 7 traces) for each content. We represent rebuffering intervals by a dashed line.

16:9 LCD 24" computer monitor from a total of 65 subjects (50 male and 15 female, ages 18-30). To simulate average viewing conditions, the ambient illumination was maintained at an average level (consistent across sessions and avoiding any reflections) and no color or backlight calibration was performed. The brightness of the monitor was set to 50%. According to the specification of the HP monitor, its typical brightness setting is 250 cd/m^2. To collect continuous scores, a rating bar was displayed on the bottom of the screen throughout the video playout, allowing subjects to report their QoE in real time. Overall QoE scores reflect the final QoE after viewing each video sequence in its entirety, while continuous scores capture the time-varying nature of QoE due to quality changes and rebuffering.

The subjects were comfortably seated at a distance of about three times the display's physical height (about 0.762 meters or 2.5 feet) from the computer monitor. Subjects were instructed to rate their viewing experience using a continuous slider and to disregard how interesting the video content was. For the continuous QoE evaluation, subjects were instructed to take into account their viewing experience "up to and including the current moment." It should be noted that we did not ask participants whether this part of the instruction affected their scores, but observed a strong recency phenomenon in the collected scores, as discussed in Section VI-B. All instructions were presented both in oral and in written form. To ensure that the oral presentation of the instructions was not biased, we relied on verbally communicating the same set of instructions as the ones included in a written form and presented them consistently and in the same order. The subjects were all students with limited knowledge in video processing who voluntarily participated in the study, without receiving compensation. Before doing the test, all participants signed a consent form and were informed that their participation was voluntary and they could discontinue the test at any point during the test. The collected data was also anonymized. We also include a picture from the lab where the actual test took place in the supplementary material.

Given practical constraints on the study duration, we used a round-robin approach to assign distorted videos to subjects. Each subject viewed all 15 contents, but only 10 distorted (2 adaptors and 5 network traces) videos per content. Given the sequence of adaptors BB, RB, QB and OQB and network traces 0 to 7, we assigned them to subjects in a circular fashion. For example, if subject i was assigned to BB and RB and network traces 0 to 4, then subject $i+1$ was assigned to RB and QB and traces 1 to 5. This led to a slightly uneven distribution of subjects per distorted video, but we considered this to have a minor effect. The benefit of a round robin approach is 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 per subject). Each session was conducted at least 24 hours apart to minimize subject fatigue [57]. To reduce 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. We did not carry out an Ishihara color test, but verbally asked participants about any color blindness. At the start of the first session, every subject viewed three training videos with representative qualities and contents that were not used in the actual study. The purpose of the training stage was to introduce the subjects to the subjective testing procedure and interface and the types of distortions present in the test.

To design the experimental interface, we relied on Psychopy [58], which generates and displays visual stimuli with high precision, which is very important when collecting continuous, per-frame subjective data. The interface is available at https://github.com/christosbampis/Psychopy_Software_Demo_LIVE_NFLX_II. The test subjects used a computer mouse to interact with the Psychopy interface. Using this interface, we collected opinion scores in the range of [1, 100]. To facilitate the evaluation process, the words "Bad" and "Excellent" were displayed on the two extremes of the rating bar.

We found Psychopy to be quite reliable to measure responses to real-time stimuli. The recorded scores were aligned with the number of frames displayed. For example, if the video playback consisted of 850 frames, then 850 measurements (one per frame) would be recorded. This does not mean that the user reaction time was diminished, but we found that the reaction times were consistently within 1-2 seconds for all users. We roughly estimated this average reaction time by measuring the time difference between a rebuffering event and a sudden change in the measured QoE scores. This observation

aligns well with earlier studies in continuous QoE [11]. Even though we did not account for each user's response lag, we were still able to successfully analyze the data and draw valuable conclusions as part of the study.

The final database consists of 420 distorted videos (15 contents, 7 network traces and 4 adaptation algorithms) with an average of 23.2 scores (overall and continuous) for 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 overall scores and 9750 continuous-time waveforms to study subjective QoE.

Following data collection, we applied z-score normalization per subject and per session [11], [59] to account for subjective differences when using the rating scale. The z-score normalization for the overall scores was derived as follows. Let $s_{ijk}(t)$ denote the overall 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:

$$\hat{s}_{ijk}(t) = \frac{s_{ijk}(t) - \mu_{s,ik}}{\sigma_{s,ik}} \tag{1}$$

where $\mu_{s,ik}$ is the mean overall score assigned to all videos at session k of subject i and $\sigma_{s,ik}$ is the corresponding standard deviation.

To reliably calculate the Mean Opinion Score (MOS) of overall QoE, subject rejection techniques are commonly applied [60]. Notably, while we applied subject rejection on the z-scored values, none of the subjects were rejected. We found that the overall QoE scores were in high agreement, exhibiting a between-group (splitting the scores per video into two groups) Spearman's Rank Order Correlation Coefficient (SROCC) of 0.96.

For the continuous scores, we performed a similar z-score normalization and then averaged the normalized continuous scores per subject to compute a continuous MOS score for each frame. While more advanced subject rejection techniques could have been used as in [11], we found that the average (across subjects) continuous-time scores did not significantly change after continuous-time rejection.

V. Objective Analysis of LIVE-NFLX-II

Before studying the collected subjective data, we analyze the generated video streams using simple QoE indicators, like video quality or buffer level across various dimensions. First, we will present how video quality is affected on each content and demonstrate the content/encoding diversity in the database (Section V-A). Then, we analyze the network traces and the adaptors that were tested (Sections V-B and V-C).

A. Content Analysis

Besides constructing a bitrate ladder, which is typically carried out on the server side, we can also measure the end-user quality received on the client device. Given that bitrate is not sufficient to capture perceptual quality, we use the VMAF perceptual index [3]. We used VMAF to measure video quality over all 420 videos and averaged the values for each content, as shown in Fig. 8. Contents having low complexities, such as MC, CF and CD, were delivered with better VMAF values. By contrast, challenging contents, like GTA and Soccer

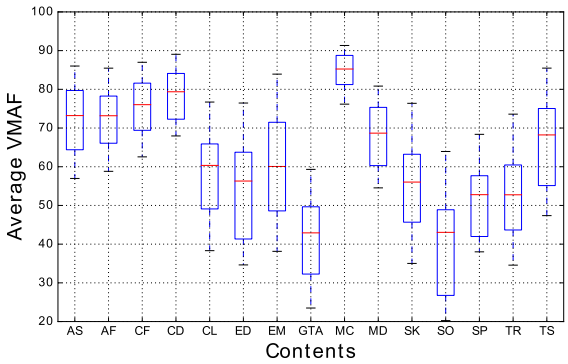

Fig. 8. Per-content quality distribution averaged over traces, adaptors and all segments per video stream (rebuffering was not taken into consideration). We use the boxplot notation defined in [61], and re-use it in Fig. 13. For clarification, Fig. 7 depicts the VMAF variations over time for the Soccer (SO) content, whereas this figure shows the distribution of the time-averages across all network traces and adaptors.

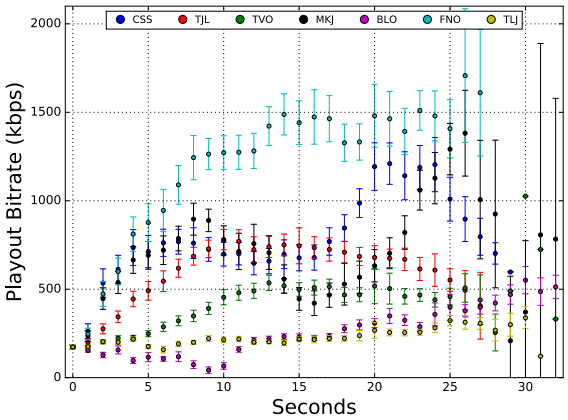

Fig. 9. Playout bitrate over time for different network traces. A value of 0 is used for the video bitrate during rebuffering. The error bars indicate the 95% confidence interval.

(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 strategies, where encoding/streaming parameters are customized to the video content streamed by each client. We provide more details in Section A of the supplementary material.

B. Network Condition Analysis

To analyze the behavior across network traces, we collected measurements of the playout bitrate, averaged over each second (and across contents and adaptors) and present its behavior per network trace in Fig. 9. Since video contents are at most 25 seconds long, only sessions that experienced rebuffering lasted longer than 25 seconds. Thus there are fewer samples after 25 seconds and the confidence intervals are larger.

As expected, better network conditions (FNO and CSS) allowed better playout bitrates when compared to low-bandwidth cases, as in TLJ and BLO. Volatile traces, such as MKJ and CSS, led to significant differences in bitrate, but this was not the case for FNO. Since FNO provides better network throughput on average than MKJ and CSS, the video buffer was sufficiently filled to account for sudden drops.

C. Adaptation Algorithm Analysis

To study adaptation behavior, we first collected key metrics (e.g. number of rebuffers) for all distorted videos generated

TABLE V

OBJECTIVE COMPARISON BETWEEN ABR ALGORITHMS FOR ALL 105 VIDEOS (15 CONTENTS AND 7 TRACES) PER ADAPTOR. THE BITRATE VALUES ARE IMPUTED WITH A VALUE OF 0 DURING REBUFFERING INTERVALS, AND VMAF IS CALCULATED ONLY ON PLAYBACK FRAMES. WE USE BOLDFACE TO DENOTE THE BEST ADAPTOR IN EACH CASE. THE AVERAGE VMAF DIFFERENCE BETWEEN CHUNKS (LAST ROW) WAS CALCULATED BY TAKING AN AVERAGE OF THE DIFFERENCES BETWEEN THE AVERAGE VMAF OF CONSECUTIVE CHUNKS WITHIN THE VIDEO PLAYBACK. FOR EXAMPLE, IF THREE SEGMENTS A, B AND C HAD AN AVERAGE VMAF OF 60, 65 AND 85, THEN THE AVERAGE VMAF DIFFERENCE BETWEEN CHUNKS IS 70

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 (seconds)	1.02	1.35	**0.79**	1.14
per chunk avg. VMAF	58.05	62.58	**64.52**	63.19
avg. VMAF difference between chunks	9.67	7.51	**6.89**	8.59

by each adaptation algorithm. Table V shows that the OQB adaptor improved most of the objective streaming metrics, e.g. the playout bitrate or the rebuffering duration. This demonstrates that a better bandwidth prediction model can improve the behavior of an ABR algorithm.

By contrast, RB led to the largest amount of rebuffering, since it picks the subsequent chunk rate in a greedy fashion, it is myopic (does not look ahead in time) and does not consider 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 bitrate. Between RB and BB, QB offers a better tradeoff between playout bitrate and rebuffering. These results are not surprising: maximizing quality/bitrate or avoiding rebuffering are conflicting goals, hence, designing adaptors should focus 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 $B_h = 10$ seconds, and $h = 10$ seconds, the dynamic programming solution may fail to return an optimal solution. We found that by increasing B_h and h, both OQB and QB could reduce rebuffering, but we decided to challenge the behavior of these ABR algorithms by selecting low/volatile bandwidth traces and a low buffer size. By doing so, we could collect valuable subjective responses on video sequences under difficult streaming conditions, including significant video quality degradations and rebuffering.

Next, we study how each adaptation algorithm behaves over time within each session. As before, we measure the per second playout bitrate and buffer level, and show the per adaptation evolution in Fig. 10. In terms of bitrate, RB starts aggressively for the first few seconds, but then tends to have a lower bitrate compared to quality-based adaptors. By contrast, BB is the most conservative strategy in terms of bitrate, while QB and OQB deliver start-up bitrates between RB and BB. However, after about 15 seconds, QB and OQB consistently deliver higher bitrates. Note that video can only be longer than 25 seconds due to rebuffers. Therefore, for time intervals

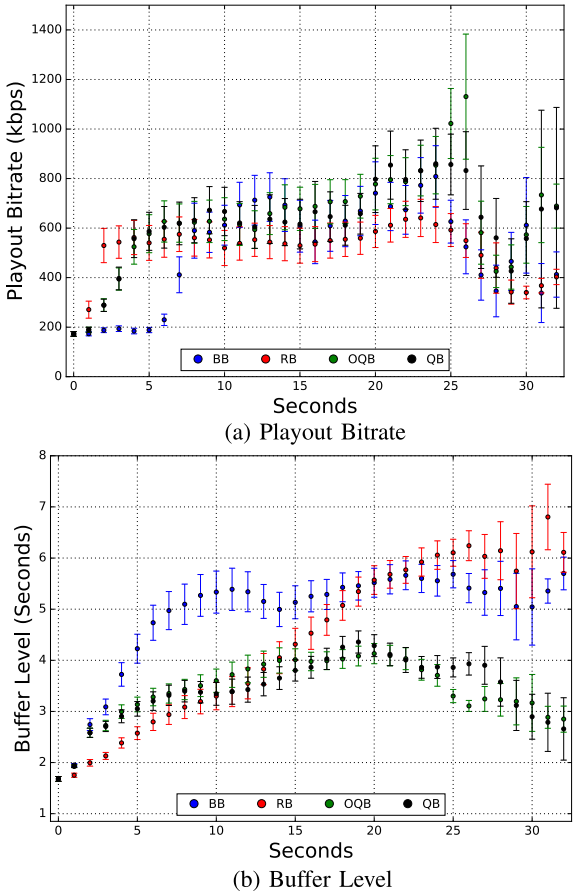

Fig. 10. Playout bitrate and buffer level (in seconds) over time across adaptors (with 95% confidence interval). A value of 0 is used for the video bitrate during rebuffering.

greater than 25 seconds, there are video sequences under more challenging conditions and hence it is expected that the bitrate decreases and the buffer level decreases or stays the same.

Fig. 10 shows that ABR algorithms mainly differ from each other during the start-up phase. For example, RB chooses playout bitrate aggressively by closely following the available bandwidth, while BB is more conservative and prioritizes on accumulating video buffer. After the start-up phase, the ABR algorithms converge and make similar decisions.

During the start-up phase, there is little to no video buffer to absorb the impact of network variations, hence different ABR decisions will lead to different buffer levels. For example, a combination of aggressive quality switching and network volatility leads RB to produce the worst rebuffering in the start-up phase (see also Fig. 11) and slows down buffer build-up (Fig. 10b). Nevertheless, after sufficient time, the RB buffer level increases and even surpasses the BB buffer level. In the case of QB and OQB, both adaptors try to reach the target buffer $B_t = 3$. While we did not specify a maximum buffer size for RB and BB, as shown in Fig. 10b, none of the adaptors achieve $B_h = 10$ seconds, given the challenging network traces.

Until now, we have been mostly contrasting ABR algorithms. Nevertheless, we have also found an 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. Figure 11 shows that all adaptors

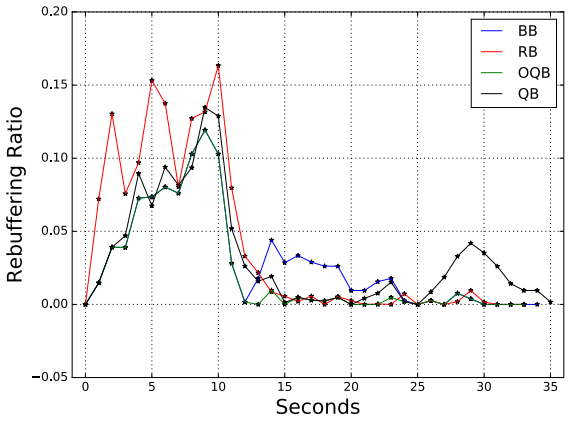

Fig. 11. Rebuffering ratio per adaptor, defined as the fraction of videos that rebuffered within a one second window.

had 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 playout starts (see supplementary material, Section C). Between adaptors, there are, of course, differences as well. RB experiences heavier early rebuffering, since the buffer level is not taken into account. QB can lead to rebuffering much later in the video than OQB, which is aware of the entire network trace, and is able to minimize rebuffering events from occurring at a later time.

VI. Subjective Analysis

So far, we have studied 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 QoE models. Here we analyze the database by means of the collected subjective scores.

A. Analysis Using Overall Scores

To identify the main QoE factors, Fig. 12 highlights the relationships between overall scores and average VMAF values (calculated on non-rebuffered frames), and the number and duration of rebuffering events respectively. Unsurprisingly, the presence of rebuffering (red points) negatively impacts the overall correlation of VMAF with subjective scores, since VMAF does not account for the effects of rebuffering on user QoE. Naturally, a larger number of rebuffering events tends to decrease QoE. In Section VII, we show how QoE prediction models based on VMAF can deliver improved performance.

As an exception, the points with 3, 4 and 5 rebuffering events are not in decreasing MOS order. We found that the corresponding average rebuffering durations were 4.33, 3.49 and 2.93 seconds respectively, meaning that larger rebuffering occurrence did not necessarily imply larger rebuffering duration. Therefore, Fig. 12b demonstrates that subjects are sensitive to a combined effect of rebuffering occurrence and duration.

In Fig. 12c, we observe that a longer rebuffering time lowers QoE, but when rebuffering is longer than 4 seconds, *duration neglect effects* [62] 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 (after a certain cutoff) when making overall QoE evaluations.

We also compared the overall QoE scores among different adaptors (Fig. 13). We observed that the opinion scores are 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. 11), and because, just before the video finishes playing (and the overall score is recorded), the adaptation algorithms have built-up enough buffer to better handle bitrate/quality variations, even if the network is varying significantly. Therefore, it is likely that recency effects [11], [62] led to biases in overall QoE evaluations, i.e., subjects are forgiving/forgetful when recording overall QoE.

To validate this recency effect, we averaged the continuous scores over one second windows and calculated the correlation with the final scores, as in [11]. For example, we found that the average continuous scores calculated over the $[4, 5]$ second window correlated weakly with the overall QoE scores (correlation of 0.58). However, by averaging the continuous scores over the $[24, 25]$ second window (20 seconds later), the correlation increased to 0.94. Meanwhile, per adaptor differences in terms of average VMAF were not considerably different, e.g., between RB and OQB, (see Table V) and hence the overall scores were also similar across adaptors.

Lastly, we investigated the per-subject variations for all the distorted videos in the database (see Figure 14). For simplicity, we looked at the subjective scores prior to z-score normalization, such that the data is still in the [1, 100] scale. We found that the confidence intervals ranged from 3 to 10, with an average value of about 6.

B. Analysis Using Continuous Scores

Following our per-second objective analysis in Section V, Fig. 15 depicts the continuous-time user experience across adaptation algorithms. 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. This also means that subjects preferred increased early rebuffering, if it meant better start-up quality, as in the case of RB. Within the first 12 seconds, BB is 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, which are statistically close. After 25 seconds, QoE measurements are decreasing and have larger confidence intervals, since they correspond to videos that rebuffered, and their count decreases over time.

Notably, as in Fig. 13, we found that OQB is not statistically better than QB, even though it has perfect knowledge of the future bandwidth and performs the best in terms of objective metrics. As already explained, for the majority of distorted videos, rebuffering and quality degradations occurred earlier during video playout and this led to smaller differences in the subjective opinions per adaptor and over time. This experimental result does not suggest that better bandwidth prediction is not an important goal, but it does show that better bandwidth prediction does not significantly influence overall QoE scores. Meanwhile, the significant differences in QoE between adaptation strategies in the start-up phase underlines that temporal studies of QoE are highly relevant for adaptive video streaming, given that ABR algorithms are especially challenged during start-up.

Viewed from the network condition perspective, we found that continuous-time subjective scores are affected by dynamic video quality changes and rebuffering. Figure 16 shows that, for all traces, a few seconds are needed to build up video

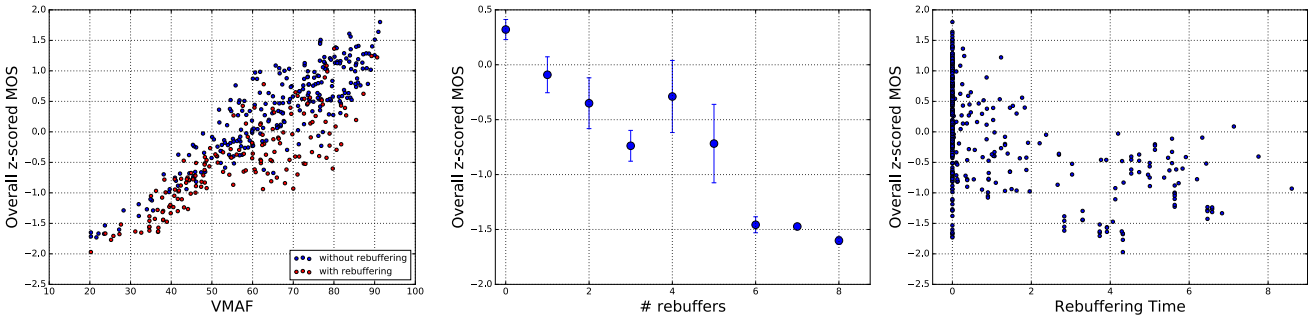

Fig. 12. Left to right: (a) VMAF and MOS. Blue points correspond to videos displayed without rebuffering. Red points correspond to videos impaired by rebuffering of non-zero duration. Videos that are affected by rebuffering that have similar VMAF scores as videos without any rebuffering, tend to have noticeably lowered MOS. (b) # rebuffers and MOS (95% conf. intervals) and (c) Rebuffer duration (in seconds) and MOS. Around 40% of the video sequences have at least one rebuffering event.

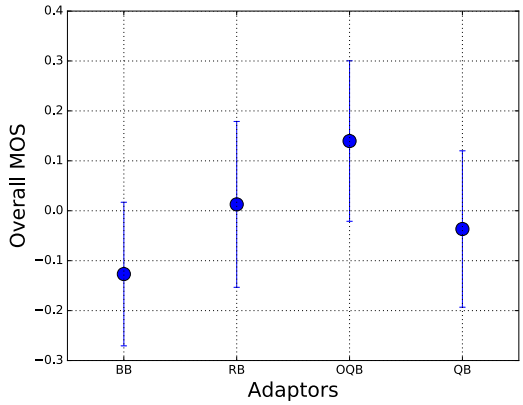

Fig. 13. Overall QoE score distribution (with 95% confidence interval) for different adaptation algorithms.

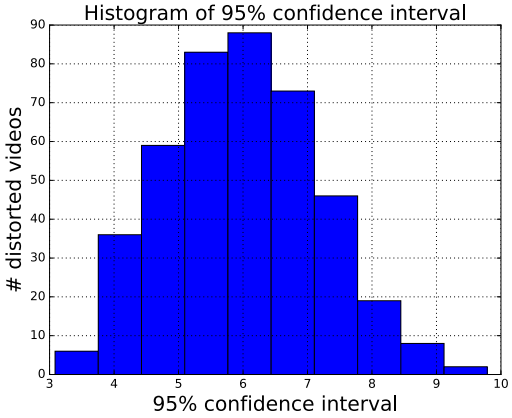

Fig. 14. Distribution of 95% confidence intervals (using the original scale [1, 100]) for all 420 distorted videos.

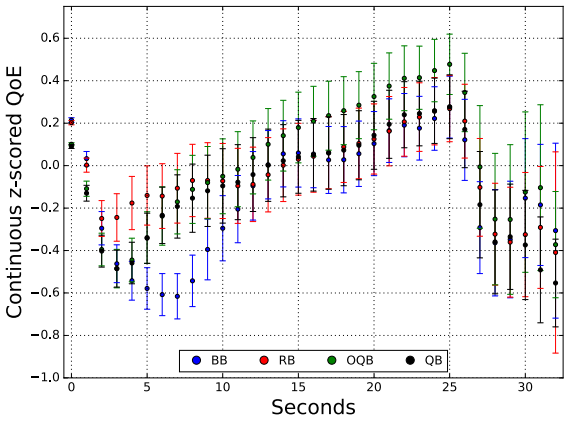

Fig. 15. Continuous-time scores across adaptation algorithms.

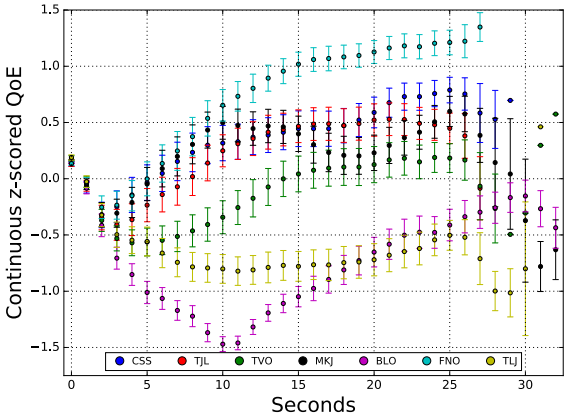

Fig. 16. Continuous-time scores across network traces.

buffer and hence continuous scores are relatively low. Under better network conditions (e.g. FNO), user experience steadily improves after some time, due to the adaptors switching to higher resolution and lower compression ratio. By contrast, challenging cases such as BLO and TLJ recover slowly or do not recover at all, while very volatile conditions, as in MKJ, can lead to large drops in QoE much later during playback. We refer the reader to Table III for a reminder on the acronyms used for the network traces.

C. Adaptation Algorithm Performance Discussion

Following our earlier between-adaptor analysis, it is natural to ask which adaptation algorithm performs the best. For overall scores, we could not make statistically significant comparisons, in part due to recency effects. Similarly, using continuous-time scores, we found that OQB performed marginally better for a period of time, but the differences were not statistically significant even though OQB has perfect knowledge of future bandwidth. By contrast, BB was conservative during startup and did not select high quality streams.

Comparing RB and QB, we found that they delivered similar QoE over 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 taking into account the buffer level in its optimization scheme), but leads to many quality switches (see also Table V). 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 large quality switches.

At this point, an important question can be raised: if overall scores were not very different across adaptation algorithms and since continuous scores tend to get closer and closer for the second half of the playback, what is the point of designing perceptually-optimized adaptive streaming algorithms? To answer this, there are two main points to be raised. The first one relates to the use of overall scores. In practical adaptive video streaming applications, playback sessions last for tens of minutes and hence overall scores, which are significantly biased by recency effects, are not reflective of the continuous quality of experience. More importantly, if there is a sudden network change that leads to significant rebuffering or video quality drop, the viewer may never finish watching the entire video sequence. This demonstrates that the use of overall scores is quite limiting and does not capture the time-varying quality of experience.

With regards to continuous scores, it is important to understand that many adaptive video streaming algorithms will perform similarly when the network is stable and a steady state is reached with respect to network variations. This applies to the second half of the video sessions, where we see that all four adaptation algorithms have similar behavior. However, the most challenging aspect of designing these algorithms is during start-up or when there is a sudden and unpredictable network change. This is where perceptually-optimized adaptive streaming algorithms can shine: they can potentially make better decisions that improve the quality of experience.

D. 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 some remaining limitations. We recognise that QoE is not only affected by the factors investigated herein, such as video quality, recency, rebuffering or quality switching, but also by other factors such as audio quality, 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. Furthermore, start-up delay was not taken into account in the subjective database design, but it should be emphasized that it is indeed an important aspect of QoE. We plan to investigate this aspect as part of future work. Further, future work could rely on more future-looking network traces as they become available.

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. Furthermore, the streaming sessions we generated are only between 25 and 36 seconds long (overall duration) and the network simulator does not consider the underlying TCP behavior, such as its slow restart property. Nevertheless, given the very large design space, it is virtually impossible to vary and explore all of the above streaming conditions simultaneously. Finally, we think that the design of lab-based subjective studies for adaptive video streaming applications, such as LIVE-NFLX-II, is naturally limited by the number of subjects. It is very challenging to have a large number of subjects for every possible video streaming scenario. In the future, crowdsourcing studies could be a promising alternative to gather larger amounts of data to better cover the different aspects of QoE.

VII. Perceptual Video Quality and QoE

The perceptual optimization of adaptive video streaming requires accurate QoE prediction models [10], [19], [21], [63]–[70]. An important goal of our database is to use it as a development testbed for such QoE prediction models. In this section, we evaluate a number of representative video quality assessment (VQA) and QoE prediction models. Given that the database contains both overall and continuous-time scores, we studied the performance of these algorithms both for overall 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 videos with non-16:9 aspect ratio and, prior to VQA calculations, we removed black bars to measure quality only on active pixels. For videos containing rebuffered frames, we removed all of those frames and calculated video quality on the aligned YUV files [11]. In the next sections, we investigate the predictive performance of leading VQA models and study their predictive performance when combined with QoE-driven models for overall and continuous-time QoE prediction.

A. Objective Models for Overall QoE Prediction

Our first experiment was to evaluate several well-known video quality and QoE metrics, including PSNR, PSNRhvs [71], SSIM [72], MS-SSIM [73], ST-MAD [74], ViS3 [24], VQM-VFD [75], V-BLIINDS [76], ST-RRED [77], VMAF [3] (version 0.6.1), SQI [10] and Video ATLAS [70]. PSNRhvs is an extension of the traditional PSNR metric which incorporates properties of the human visual system. SSIM uses local image statistics to capture structural image degradations while MS-SSIM performs similar calculations across multiple scales. ST-MAD relies on a most apparent-distortion model of videos, while V-BLIINDS uses DCT-based features to model distortions of natural video statistics. ViS3 uses spatial and temporal slices to predict video quality, while VQM-VFD feeds a number of perceptual features into a neural network.

ST-RRED is a VQA metric that relies on entropic differencing of wavelet coefficients of frames and frame differences. VMAF combines multiple elementary perceptual quality measurements as features and feeds them to a support vector regressor. SQI and Video ATLAS combine video quality measurements and rebuffering statistics to measure QoE. The original Video ATLAS model [70], was designed and tested on the LIVE-NFLX and Waterloo databases, where quality switching events were much less diverse. Given the flexibility of Video ATLAS and the diversity of our newly designed database, we re-trained the model to include changes on resolution and quality. Note that we did not include complex VQA metrics like MOVIE [78] in our evaluation, which can be computationally inefficient for 1080p input videos and sometimes lead to out-of-memory issues [39].

We used VMAF as the VQA feature, average absolute difference of encoding resolution and rebuffer duration as features. The average absolute difference of encoding resolution can provide valuable insights on the amount/frequency of resolution changes; more frequent and larger encoding resolution switches can drastically affect quality of experience.

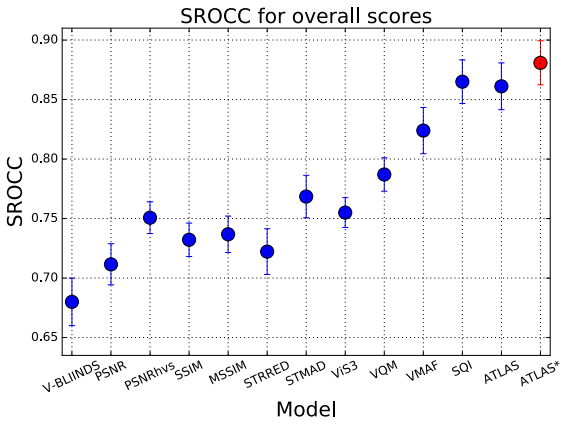

Fig. 17. SROCC results using overall scores, with 95% confidence intervals across multiple train/test splits. For Video ATLAS, we report SROCC twice: 1. when trained on the database in [34] using the reduced feature set (in blue) and 2. when trained on each training split of LIVE-NFLX-II (in red, ATLAS*). The same test splits were used across all methods.

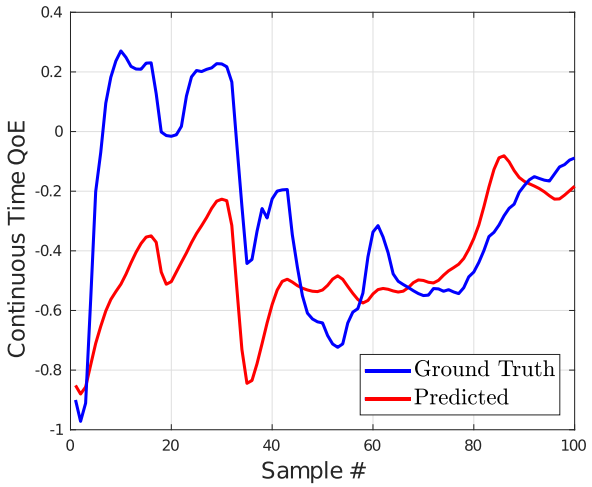

Fig. 18. An example where the G-NARX QoE prediction does not capture trends in subjective QoE.

As an alternative to this feature, we could have also considered features such as the average VMAF variation. Future work can investigate how more sophisticated features can better capture the effects of quality variations.

We also kept track of the time in seconds (from the end of the video) since the most recent minimum quality was observed (TLL feature). For SQI, VMAF was also used as the VQA model. We excluded the P.1201-3 models [79], since they are trained for longer video sequences (>1min.). To evaluate performance, we use SROCC and report the results in Fig. 17.

Since Video ATLAS is a learning-based model, we split the database into multiple train/test splits. It is common to split VQA databases into content-independent splits; but for the streaming QoE scenario we propose a different approach. Given that video contents are pre-encoded and the behavior of an adaptation algorithm is deterministic (given a network trace and a video content), it is realistic to assume that, during training, we have collected subjective scores on a subset of the network traces. Using content-independent splits might have reduced content biases, but would then introduce more severe distortion biases: the same network conditions would have been used both for training and testing, leading to similar distortion patterns and increasing the chances of overfitting.

Therefore, we perform our splitting based on 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 hundreds of videos. Figure 17 shows boxplots of performance across all 21 iterations for all compared models.

All of the IQA or VQA-only models, including the no-reference V-BLIINDS model, lagged in performance, which is expected since they only capture video quality and disregard other critical aspects of QoE such as rebuffering. Nevertheless, VMAF performed significantly better than all other models. Of course, VMAF was used to generate the bitrate ladder, to decide on encoding parameters and to perform client-based adaptation for QB and OQB. Hence our system may be better tuned towards VMAF and this choice has a direct impact on user experience. Using VMAF as part of the SQI and Video ATLAS QoE predictors led to performance gains in both cases.

To validate the contribution of VMAF in predicting QoE, and measure the contribution of the other three factors in Video ATLAS (rebuffering duration, resolution switching and TLL), we also trained a simple Random Forest (RF) regressor and measured feature importance over all 21 unique train/test combinations. The number of estimators in RF was set to 100 and the feature importances were found to be: 79% for VMAF, 11% for rebuffering duration, 5% for resolution switching and 5% for TLL. This validates the observation that VMAF scores contribute strongly to QoE prediction, while rebuffering duration is less important, in part because rebuffering events occurred earlier in the streaming session and hence were less important for overall QoE evaluations. We tested different values of the number of estimators used in the random forest implementation (10 to 1000) and got similar results.

Unlike SQI, Video ATLAS is a trained model, and hence we also trained it on a different database and then tested it using the exact same test splits as above. We used the database in [34] and extracted two basic features: VMAF 0.6.1 and rebuffering duration (in seconds). We found the SROCC of Video ATLAS in that case dropped to an average of 0.86 (compare blue and red point for Video ATLAS in Fig. 17) across all 21 test sets (from 0.88). When testing Video ATLAS on a different database than the one it was trained on, it was expected that performance would drop. However, the performance drop was not large and the SROCC performance was statistically equivalent to that of SQI, as shown by the overlapping confidence intervals in Fig. 17. This result reinforces our belief that video quality and rebuffering duration are indeed good predictors of overall streaming QoE.

B. 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. To evaluate performance, we used root mean squared error (RMSE) and outlier ratio (OR). RMSE measures the prediction's fidelity to the ground truth, while OR measures the frequency of outlier points. To calculate the OR, we relied on the definition

described in ITU-T Rec. P.1401 [80]. Notably, SROCC may not be an appropriate choice for comparing time-series [21].

We evaluated two prediction algorithms presented in [21], one based on autoregressive neural networks (G-NARX) and the other based on recurrent neural networks (G-RNN). To train the G-NARX and G-RNN models, we used per-frame VMAF measurements as the continuous-time VQA feature. We also included two additional continuous-time features: a per-frame boolean variable denoting the presence of rebuffering and another denoting the time since the latest rebuffer. We used 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 seconds (as in [21]), by averaging the continuous-time variables accordingly. This subsampling step speeds up the training process by reducing the amount of training data.

G-NARX and G-RNN delivered comparable performances: G-NARX had an RMSE of 0.267 and an OR of 7.136% and G-RNN had an RMSE of 0.276 and an OR of 5.962%. Their performance is promising: only 5%-7% of the QoE predictions were significantly different from the average ground truth score. Nevertheless, we observed cases where predictions could be further improved, as in Fig. 18, where G-NARX did not accurately capture subjective QoE trends and its dynamic.

We also experimented with two additional feature sets for G-NARX to determine whether its performance would improve further. Together with the existing three continuous features, we tried two more features: a continuous feature calculated by the per-frame VMAF difference (F1), and another continuous feature calculated as the time duration (in seconds) since the lowest VMAF score (for a given session) has been observed (F2). Feature F1 aims to capture the per-frame quality variations over time, while F2 focuses on recency effects triggered by very low qualities within a given session. We observed that introducing an additional feature (either F1 or F2) to G-NARX could lead to an improvement in the OR of about 2%, but the RMSE performance did not improve significantly. This could mean that these features did not capture additional information descriptive of the complex continuous QoE phenomena. More detailed results are available in the supplementary material.

In similar test results, we identified that perhaps the main problem of these trained networks is that they do not always capture the magnitude of subjective opinion changes over time, i.e. they tend to over- or under-estimate a drop or an increase in QoE. In practical applications, more accurate predictions are needed if these QoE models are to be used for rate adaptation in actual streaming scenarios. This demonstrates the need to integrate better human perception models, to accurately capture continuous QoE responses.

VIII. Discussion and Conclusion

We presented the design of a large, comprehensive subjective video database, which relied on a realistic streaming system. The basic components of that system were: an encoding module, a network module, a quality module and a client module. The encoding module determined the encoding resolution and QP to be used for every shot and was driven by measurements generated by the quality module. The network module integrated network traces and orchestrated communications between the encoding and client modules. Lastly, the client module integrated encoding, network and quality information to determine the next chunk to be played out.

After generating a large variety of video streams, we presented them to a large number of subjects and collected ground truth continuous and overall QoE scores. The collected data allowed us to analyze overall and continuous-time user experiences under different network conditions, using different adaptation algorithms, and on diverse video contents. We found that start-up is a challenging phase for ABR algorithms, since the video buffer is not sufficient to withstand large network variations. However, human responses were forgetful of negative QoE events during start-up, which underlines the need to better understand continuous streaming QoE. Using the collected human opinion scores, we also trained and evaluated predictors of video quality and quality of experience. We found that average video quality and rebuffering duration were the most important factors contributing to accurate overall QoE prediction, but that there is significant room for improvement of continuous-time QoE models.

In the future, we plan to investigate a larger variety of streaming factors and their effect on viewing experience. For example, studying start-up delay, which plays an important role in user engagement, is a promising direction. Another potential future direction is to study more sophisticated adaptive bitrate algorithms, such as Pensieve [42]. Furthermore, given that this database relied on 3G network traces, future work could rely on more future-looking network traces, such as 5G. We also intend to use the data to build better continuous-time QoE models that integrate additional features, such as network estimates and buffer status. Our ultimate goal is to "close the loop," i.e., inject such QoE models into the client-adaptation strategy to perceptually optimize streaming.

Acknowledgment

The authors would like to thank Anush K. Moorthy for discussions regarding content encoding complexity and the content-adaptive bitrate ladder. They also thank to Anne Aaron and the entire Video Algorithms team at Netflix for supporting this work. The research was IRB-exempt 2007-11-0066. An arxiv version is available here: https://arxiv.org/abs/1808.03898

References

[1] C. G. Bampis, Z. Li, I. Katsavounidis, C. Ekanadham, and A. C. Bovik, "Subjective analysis of an end-to-end streaming system," *Electron. Imag.*, vol. 2019, no. 10, p. 321, Jan. 2019.

[2] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 187–198, 2015.

[3] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, "Toward a practical perceptual video quality metric," Netflix, Los Gatos, CA, USA, Tech. Rep.

[4] K. Brunnström *et al.*, "Qualinet: European network on quality of experience in multimedia systems and services," Tech. Rep., 2013.

[5] M.-N. Garcia *et al.*, "Quality of experience and HTTP adaptive streaming: A review of subjective studies," in *Proc. Int. Workshop Qual. Multimedia Exper.*, 2014, pp. 141–146.

[6] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hobfeld, and P. Tran-Gia, "A survey on quality of experience of HTTP adaptive streaming," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 469–492, 1st Quart., 2015.

[7] S. Tavakoli, S. Egger, M. Seufert, R. Schatz, K. Brunnström, and N. García, "Perceptual quality of HTTP adaptive streaming strategies: Cross-experimental analysis of multi-laboratory and crowdsourced subjective studies," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 8, pp. 2141–2153, Aug. 2016.

[8] S. Tavakoli, K. Brunnström, J. Gutiérrez, and N. García, "Quality of experience of adaptive video streaming: Investigation in service parameters and subjective quality assessment methodology," *Signal Process., Image Commun.*, vol. 39, pp. 432–443, Nov. 2015.

[9] N. Staelens *et al.*, "Subjective quality assessment of longer duration video sequences delivered over HTTP adaptive streaming to tablet devices," *IEEE Trans. Broadcast.*, vol. 60, no. 4, pp. 707–714, Dec. 2014.

[10] Z. Duanmu, K. Zeng, K. Ma, A. Rehman, and Z. Wang, "A quality-of-experience index for streaming video," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 1, pp. 154–166, Feb. 2017.

[11] 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 Trans. Image Process.*, vol. 26, no. 11, pp. 5217–5231, Nov. 2017.

[12] C. G. Bampis, Z. Li, and A. C. Bovik, "Continuous prediction of streaming video QoE using dynamic networks," *IEEE Signal Process. Lett.*, vol. 24, no. 7, pp. 1083–1087, Jul. 2017.

[13] T. Hossfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen, "Initial delay vs. interruptions: Between the devil and the deep blue sea," in *Proc. 4th Int. Workshop Qual. Multimedia Exper.*, Jul. 2012, pp. 1–6.

[14] N. Eswara *et al.*, "Streaming video QoE modeling and prediction: A long short-term memory approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 3, pp. 661–673, Mar. 2020.

[15] Z. Duanmu, K. Zeng, K. Ma, A. Rehman, and Z. Wang, "A quality-of-experience index for streaming video," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 1, pp. 154–166, Feb. 2017.

[16] N. Barman and M. G. Martini, "QoE modeling for HTTP adaptive video streaming—A survey and open challenges," *IEEE Access*, vol. 7, pp. 30831–30859, 2019.

[17] M. J. Khokhar, "Modeling quality of experience of Internet video streaming by controlled experimentation and machine learning," M.S. thesis, HAL, India Sophia Antipolis, Valbonne, France, 2020.

[18] I. Katsavounidis, "Dynamic optimizer—A perceptual video encoding optimization framework," Netflix, Los Gatos, CA, USA, Tech. Rep.

[19] C. Chen, L. K. Choi, G. de Veciana, C. Caramanis, R. W. Heath, Jr., and A. C. Bovik, "Modeling the time—Varying subjective quality of HTTP video streams with rate adaptations," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2206–2221, May 2014.

[20] D. Ghadiyaram, J. Pan, and A. C. Bovik, "A subjective and objective study of stalling events in mobile streaming videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 183–197, Jan. 2019.

[21] C. G. Bampis, Z. Li, I. Katsavounidis, and A. C. Bovik, "Recurrent and dynamic models for predicting streaming video quality of experience," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3316–3331, Jul. 2018.

[22] A. K. Moorthy, L. K. Choi, A. C. Bovik, and G. de Veciana, "Video quality assessment on mobile devices: Subjective, behavioral and objective studies," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 6, pp. 652–671, Oct. 2012.

[23] F. De Simone, M. Tagliasacchi, M. Naccari, S. Tubaro, and T. Ebrahimi, "A H.264/AVC video database for the evaluation of quality metrics," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2010, pp. 2430–2433.

[24] P. V. Vu and D. M. Chandler, "ViS3: An algorithm for video quality assessment via analysis of spatial and spatiotemporal slices," *J. Electron. Imag.*, vol. 23, no. 1, Feb. 2014, Art. no. 013016.

[25] *VQEG HDTV Phase I*, Video Qual. Experts Group, 2010.

[26] V. Hosu *et al.*, "The Konstanz natural video database (KoNViD-1k)," in *Proc. 9th Int. Conf. Qual. Multimedia Exper. (QoMEX)*, May 2017, pp. 1–6.

[27] S. Winkler, "Analysis of public image and video databases for quality assessment," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 6, pp. 616–625, Oct. 2012.

[28] T. C. Thang, H. T. Le, A. T. Pham, and Y. M. Ro, "An evaluation of bitrate adaptation methods for HTTP live streaming," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 693–705, Apr. 2014.

[29] B. Rainer and C. Timmerer, "Quality of experience of Web-based adaptive HTTP streaming clients in real-world environments using crowdsourcing," in *Proc. Workshop Design, Qual. Deployment Adapt. Video Streaming (VideoNext)*, 2014, pp. 19–24.

[30] J. Søgaard, M. Shahid, J. Pokhrel, and K. Brunnström, "On subjective quality assessment of adaptive video streaming via crowdsourcing and laboratory based experiments," *Multimedia Tools Appl.*, vol. 76, no. 15, pp. 16727–16748, Aug. 2017.

[31] J. Y. Lin, R. Song, C.-H. Wu, T. Liu, H. Wang, and C.-C.-J. Kuo, "MCL-V: A streaming video quality assessment database," *J. Vis. Commun. Image Represent.*, vol. 30, pp. 1–9, Jul. 2015.

[32] H. Yeganeh, F. Qassemi, and H. R. Rabiee, "Joint effect of stalling and presentation quality on the quality-of-experience of streaming videos," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 310–314.

[33] L. Ljung, *System Identification: Theory for the User*. Upper Saddle River, NJ, USA: Prentice-Hall, 1986.

[34] Z. Duanmu, A. Rehman, and Z. Wang, "A quality-of-experience database for adaptive video streaming," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 474–487, Jun. 2018.

[35] A. Aaron, Z. Li, M. Manohara, J. D. Cock, and D. Ronca. *Per-Title Encode Optimization*. [Online]. Available: https://medium.com/netflix-techblog/per-title-encode-optimization-7e99442b62a2

[36] Z. Li *et al.*, "VMAF: The Journey continues," Netflix, Los Gatos, CA, USA, Tech. Rep.

[37] *Blender Video Content*. Accessed: Sep. 13, 2020. [Online]. Available: https://www.blender.org

[38] *FFmpeg H.264 Video Encoding Guide*. Accessed: Sep. 13, 2020. [Online]. Available: https://trac.ffmpeg.org/wiki/Encode/H.264

[39] C. G. Bampis, Z. Li, and A. C. Bovik, "Spatiotemporal feature integration and model fusion for full reference video quality assessment," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 8, pp. 2256–2270, Aug. 2019.

[40] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Commute path bandwidth traces from 3G networks: Analysis and applications," in *Proc. ACM Multimedia Syst. Conf.*, 2013, pp. 114–118.

[41] *HSDPA Dataset*. Accessed: Sep. 13, 2020. [Online]. Available: http://skuld.cs.umass.edu/traces/mmsys/2013/pathbandwidth/

[42] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proc. SIGCOMM*, 2017, pp. 197–210.

[43] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate adaptation for adaptive HTTP streaming," in *Proc. ACM Conf. Multimedia Syst.*, 2011, pp. 169–174.

[44] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with festive," *IEEE/ACM Trans. Netw.*, vol. 22, no. 1, pp. 326–340, Jan. 2014.

[45] Y. Sun *et al.*, "CS2P: Improving video bitrate selection and adaptation with data-driven throughput prediction," in *Proc. ACM SIGCOMM Conf.*, 2016, pp. 272–285.

[46] K. Miller, E. Quacchio, G. Gennari, and A. Wolisz, "Adaptation algorithm for adaptive streaming over HTTP," in *Proc. Int. Packet Video Workshop*, 2012, pp. 173–178.

[47] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," in *Proc. IEEE Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.

[48] A. Beben, P. Wiśniewski, J. M. Batalla, and P. Krawiec, "ABMA+: Lightweight and efficient algorithm for HTTP adaptive streaming," in *Proc. Int. Conf. Multimedia Syst.*, 2016, pp. 1–11.

[49] C. Zhou, C.-W. Lin, X. Zhang, and Z. Guo, "Buffer-based smooth rate adaptation for dynamic HTTP streaming," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, Oct. 2013, pp. 1–9.

[50] Z. Li *et al.*, "Probe and adapt: Rate adaptation for HTTP video streaming at scale," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 719–733, Apr. 2014.

[51] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over HTTP," *Comput. Commun. Rev.*, vol. 45, no. 4, pp. 325–338, 2015.

[52] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, "ELASTIC: A client-side controller for dynamic adaptive streaming over HTTP (DASH)," in *Proc. 20th Int. Packet Video Workshop*, Dec. 2013, pp. 1–8.

[53] C. Wang, A. Rizk, and M. Zink, "SQUAD: A spectrum-based quality adaptation for dynamic adaptive streaming over HTTP," in *Proc. 7th Int. Conf. Multimedia Syst.*, May 2016, pp. 1–12.

[54] A. Mansy, B. Ver Steeg, and M. Ammar, "Sabre: A client based technique for mitigating the buffer bloat effect of adaptive video flows," in *Proc. ACM Multimedia Syst. Conf.*, 2013, pp. 214–225.

[55] J. Kua, G. Armitage, and P. Branch, "A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1842–1866, 3rd Quart., 2017.

[56] Z. Li, A. C. Begen, J. Gahm, Y. Shan, B. Osler, and D. Oran, "Streaming video over HTTP with consistent quality," in *Proc. 5th ACM Multimedia Syst. Conf. (MMSys)*, 2014, pp. 248–258.
[57] *ITU Recommendation P.913*. Accessed: Sep. 13, 2020. [Online]. Available: https://www.itu.int/rec/T-REC-P.913-201603-I/en
[58] J. W. Peirce, "PsychoPy—Psychophysics software in Python," *J. Neurosci. Methods*, vol. 162, nos. 1–2, pp. 8–13, May 2007.
[59] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, "Study of subjective and objective quality assessment of video," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1427–1441, Jun. 2010.
[60] *Methodology for the Subjective Assessment of the Quality of Television Pictures*, document BT-500-13, International Telecommuncation Union Standard.
[61] *Matplotlib Documentation*. Accessed: Sep. 13, 2020. [Online]. Available: https://matplotlib.org/api/_as_gen/matplotlib.pyplot.boxplot.html
[62] D. S. Hands and S. E. Avons, "Recency and duration neglect in subjective assessment of television picture quality," *Appl. Cognit. Psychol.*, vol. 15, no. 6, pp. 639–657, 2001.
[63] R. K. Mok, X. Luo, E. W. Chan, and R. K. Chang, "QDASH: A QoE-aware DASH system," in *Proc. Multimedia Syst. Conf.*, 2012, pp. 11–22.
[64] D. Rodriguez, J. Abrahao, D. Begazo, R. Rosa, and G. Bressan, "Quality metric to assess video streaming service over TCP considering temporal location of pauses," *IEEE Trans. Consum. Electron.*, vol. 58, no. 3, pp. 985–992, Aug. 2012.
[65] J. Xue, D.-Q. Zhang, H. Yu, and C. W. Chen, "Assessing quality of experience for adaptive HTTP video streaming," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, Jul. 2014, pp. 1–6.
[66] Y. Liu, S. Dey, F. Ulupinar, M. Luby, and Y. Mao, "Deriving and validating user experience model for DASH video streaming," *IEEE Trans. Broadcast.*, vol. 61, no. 4, pp. 651–665, Dec. 2015.
[67] A. Bentaleb, A. C. Begen, and R. Zimmermann, "SDNDASH: Improving QoE of HTTP adaptive streaming using software defined networking," in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 1296–1305.
[68] W. Robitza, M.-N. Garcia, and A. Raake, "A modular HTTP adaptive streaming QoE model—Candidate for ITU-T P. 1203 ('P. NATS')," in *Proc. Int. Conf. Qual. Multimedia Exper.*, 2017, pp. 1–6.
[69] A. Raake, M.-N. Garcia, W. Robitza, P. List, S. Göring, and B. Feiten, "A bitstream-based, scalable video-quality model for HTTP adaptive streaming: ITU-T P. 1203.1," in *Proc. Int. Conf. Qual. Multimedia Exper.*, 2017, pp. 1–6.
[70] C. G. Bampis and A. C. Bovik, "Feature-based prediction of streaming video QoE: Distortions, stalling and memory," *Signal Process., Image Commun.*, vol. 68, pp. 218–228, Oct. 2018.
[71] N. Ponomarenko, F. Silvestri, K. Egiazarian, M. Carli, J. Astola, and V. Lukin, "On between-coefficient contrast masking of DCT basis functions," in *Proc. Int. Workshop Video Process. Qual. Metrics*, vol. 4, 2007.
[72] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[73] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, 2003, pp. 1398–1402.
[74] P. V. Vu, C. T. Vu, and D. M. Chandler, "A spatiotemporal most-apparent-distortion model for video quality assessment," in *Proc. 18th IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 2505–2508.
[75] M. H. Pinson, L. K. Choi, and A. C. Bovik, "Temporal video quality model accounting for variable frame delay distortions," *IEEE Trans. Broadcast.*, vol. 60, no. 4, pp. 637–649, Dec. 2014.
[76] M. A. Saad, A. C. Bovik, and C. Charrier, "Blind prediction of natural video quality," *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1352–1365, Mar. 2014.
[77] R. Soundararajan and A. C. Bovik, "Video quality assessment by reduced reference spatio-temporal entropic differencing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 4, pp. 684–694, Apr. 2013.
[78] K. Seshadrinathan and A. C. Bovik, "Motion tuned spatio-temporal quality assessment of natural videos," *IEEE Trans. Image Process.*, vol. 19, no. 2, pp. 335–350, Feb. 2010.
[79] *ITU Recommendation P.1203*. Accessed: Sep. 13, 2020. [Online]. Available: https://www.itu.int/rec/T-REC-P.1203
[80] *ITU Recommendation P.1401*. Accessed: Sep. 13, 2020. [Online]. Available: https://www.itu.int/rec/T-REC-P.1401-202001-I/en