Online Profiling and Adaptation of Quality Sensitivity in Internet Video

YIHUA CHENG, JUNCHEN JIANG

Video streaming systems separate two processes: (1) online video streaming (which optimizes quality metrics, such as higher bitrates, and fewer stalls), and (2) *offline* modeling of quality sensitivity (*i.e.,* how the quality metrics affect average user experience). As bandwidth scarcity and resource contention worsen, it is pressingly needed to better allocate resources by finer-grained modeling of how quality sensitivity varies *during* each video. However, per-video quality-sensitivity modeling has been impractical, especially for live videos, as traditional offline user studies can be too slow for a new video before viewers watch it.

We explore an alternative architecture, where quality sensitivity is modeled *online* by analyzing user actions in real video sessions as they stream the same video. The challenge is how to model quality sensitivity reliably and apply it in near-realtime to improve concurrent and future video sessions. We address the challenge in the context of SensitiFlow, a controller that orchestrates adaptive-bitrate (ABR) logic of video sessions to optimize the common user-satisfaction metric of user engagement (view time per session). SensitiFlow creates an *online control loop* that *(i)* gradually profiles quality sensitivity *per video segment* as more user experience-related feedback (*e.g.,* exit or skip) is received from video sessions, and *(ii)* optimizes the ABR decisions of the video sessions to jointly improve their user engagement and generate more feedback. SensitiFlow's control loop is fast enough to profile quality sensitivity online and optimize bitrate decisions under common viewer arrival patterns of live events (*e.g.,* live sports and TV shows). Using the real traces collected from 7.6M video sessions, we show that compared to a state-of-the-art (baseline) ABR logic agnostic to the variation of quality sensitivity within a video, SensitiFlow (without using more bandwidth) can realize 80% of the improvement in engagement that would have been obtained by a hypothetical "oracle" system having the knowledge of quality sensitivity in advance. Our user study also confirms that SensitiFlow can improve the mean opinion score (MOS) by 40% over the baseline ABR logic, suggesting that SensitiFlow's online profiling of quality sensitivity is effective.

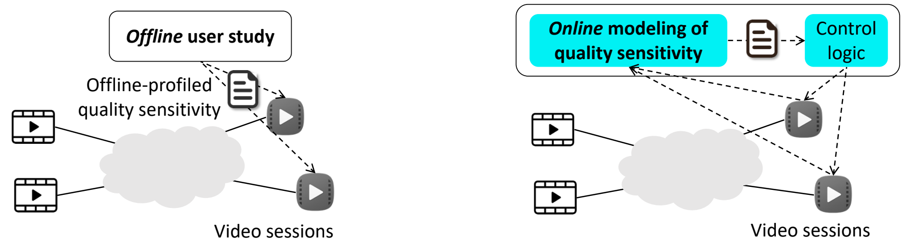
1 INTRODUCTION

Service providers across wired, wireless, and cellular networks struggle to catch up with the rapid growth of video traffic fueled by the proliferation of mobile videos, live content, ultra-high resolution videos, etc [1]. The challenge can be particularly acute in live videos, in which bandwidth demands may spike anytime, causing tremendous bandwidth contention [4, 7, 14, 20].

A key driving force for the high bandwidth demand is the traditional assumption that users' experience is equally sensitive to the quality *throughout* a video. Fortunately, recent works suggest that this assumption is unnecessary—user's true quality sensitivity *varies* greatly with the content in a video, *e.g.,* normal playtime in a sports video vs. key moments such as scoring, or heated conversations in a drama video vs. slow scenic transitions [36, 37, 73]. Thus, by prioritizing quality on more quality-sensitive video segments video streaming systems *could* improve user experience (or serve more users) *without* more bandwidth.

This observation has so far been studied only on a few example videos using lab-based or crowdsourced surveys with limited scales. Before applying it to a broader context, we perform the first-of-its-kind study

Author's address: Yihua Cheng, Junchen Jiang.



(a) Today's architecture: profiling quality sensitivity offline. (b) This work: profiling quality sensitivity *online*.

Fig. 1. *The traditional approach (offline modeling of user experience) vs. our approach (online profiling of quality sensitivity).*

based on a large real-world measurement dataset (§2.1). Unlike previous works that use session-level metrics (*e.g.,* average bitrate or rebuffering ratio over an entire session of 10s of minutes) [9, 19, 20, 45], we aggregate individual quality incidents and user actions (*e.g.,* skip, quit) per video chunk. We show that substantial variability of quality sensitivity exists among video segments in both on-demand (VoD) and live videos (§2.2). For instance, a one-second rebuffering stall can make users 3× more likely to exit if the stall occurs during one video segment than if a one-second stall occurs in another video segment within 15 seconds.

To leverage such variability of quality sensitivity, prior works [73, 74] *offline profile* quality sensitivity of individual video segments (Figure 1(a)), which requires crowdsourced tests or a large pre-determined number of history sessions. However, they suffer from two problems. *(i)* Offline modeling can take too long, making it *impractical* to optimize for live videos (*e.g.,* it takes the state-of-the-art scheme [73] tens of minutes to model the quality sensitivity of even a short video). *(ii)* Even for on-demand videos, it is difficult to pre-determine how much data (crowdsourcing ratings or user feedback from history sessions) to collect for offline profiling. Too much data will waste time, and fewer users will be optimized based on quality sensitivity, but insufficient data will have too much noise to guide correct decisions.

This paper explores an alternative architecture (Figure 1(b)), where quality sensitivity is profiled *online* (rather than offline) using user-satisfaction-related actions (*e.g.,* quit, replay, skip) of real video sessions watching the same videos to improve the user-satisfaction metric of user engagement (view time per session)[1]. We present a concrete system of the new architecture (§3), called SensitiFlow, which orchestrates the adaptive-bitrate (ABR) logic of video sessions. SensitiFlow gradually profiles quality sensitivity per video segment as it collects more user actions and quality information from different video sessions, and in the meantime, it uses the profiled quality sensitivity to make ABR decisions in a way that trades the quality of less quality-sensitive video segments for higher quality during more quality-sensitive ones.

SensitiFlow addresses the key challenge facing online profiling of quality sensitivity: how to make online profiling fast enough for live videos, despite measurement noises of user actions. It leverages two empirical

[1]While there are other measures of user satisfaction (*e.g.,* mean-opinion-score as in [73]), the online video industry often use engagement as a key user experience metric for three reasons. First, user engagement can be calculated by directly observing user actions (skip, replay, exit, etc) logged in passively collected data from all clients, while subjective user ratings must be actively elicited by surveys that have limited or biased user samples. Second, user engagement could differentiate quality's impact on different user actions (skip vs. exit), while survey-based rating only summarizes user experience in only one number. Third, substantial literature on video streaming has shown that user experience measured by user engagement is strongly correlated with video quality metrics [21, 29].

insights driven by the analysis of large-scale measurement data. First, most views of a video segment in live videos (both live events and live TV shows) span 30-90 seconds (or longer), which corroborates with our conversation with domain experts and industry reports on live streaming delays [13, 15, 17, 18]. This provides a short time window, which if used carefully could allow early viewers' user-action feedback to be used to profile quality sensitivity and improve overall user experience. Second, despite the inherent noise in user actions, such measurement noise can be compensated by a large number of video sessions of popular live videos (this does not apply to videos with very few viewers). Fortunately, for over 70% video segments in our dataset, quality sensitivity of early sessions is highly correlated (with Pearson's coefficient of 0.7) with that of later sessions, suggesting that feedback from different users can inform quality sensitivity of other users watching the same video.

SensitiFlow embraces these opportunities with several optimizations (§4). First, unlike offline profiling that uses a pre-determined number of user study samples, SensitiFlow tries to optimize more sessions by dynamically determining whether current estimates of quality sensitivity are sufficient to inform ABR decisions of a given video session (*e.g.,* it will not explore the suboptimal quality of a video segment if the uncertainty of its quality sensitivity is low enough to make the best ABR decision that maximizes user experience). Moreover, SensitiFlow restricts its ABR decisions to those that are not worse than the classic ABR baseline that is agnostic to quality sensitivity (*e.g.,* the same number of bitrate switches or rebuffering but occurring in different positions).

To quantify SensitiFlow's benefits, we create a testbed driven by the large measurement dataset of 7.6M sessions and realistic network bandwidth traces. We show that for both VoD and live videos, compared to a state-of-the-art ABR logic agnostic to quality sensitivity within a video, SensitiFlow (without using more bandwidth) can achieve 80-85% of the improvement in engagement that would have been obtained by a hypothetical "oracle" system having the knowledge of quality sensitivity in advance (*i.e.,* offline profiling but excluding the profiling overheads). Alternatively, SensitiFlow can maintain the same average user engagement as the baseline ABR logic but serve 50-100% more concurrent video sessions. Our real user study on 840 participants also confirms that SensitiFlow's online profiling technique can improve the mean opinion score (MOS) by 40% over the baseline ABR algorithm, suggesting that SensitiFlow's online profiling of quality sensitivity is highly effective.

2 MOTIVATION

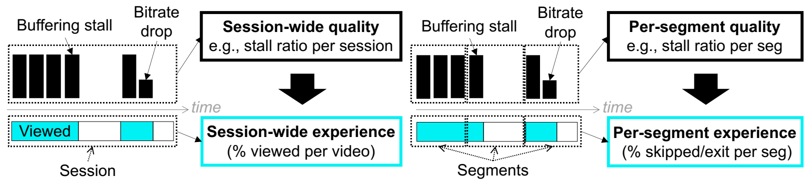
Previous studies have shown, on 10s of participants and short videos, that users' sensitivity to low-quality incidents (*e.g.,* a one-second rebuffering or a drop of bitrate) varies greatly with video content. Yet, a thorough study on the variation of quality sensitivity in videos *in the wild* remains missing.

To fill this gap, we first analyze a large measurement dataset (summarized in Table 1), which includes 18 days of user-side measurements collected from 7.6M sessions of 4 popular content providers.[2] Due to business and anonymity considerations, we anonymize the names of the videos and providers. Similar video

[2]Each session is a single view of a video by a user. We use "user" and "viewer" interchangeably, and "session" and "view" interchangeably.

Number of days	18
Sessions (views)	7.6M
Unique viewers	2.5M
View hours	3.09M
Video hours	233
Content providers	4

Table 1. Dataset summary



(a) Session-wide analysis (traditional) **(b) Segment-level analysis (this paper)**

Fig. 2. Measuring the variation of quality sensitivity across video segments.

datasets exist, but our analysis is unique in the following sense. We use the dataset to calculate not only session-level quality metrics (*e.g.,* aggregated rebuffering ratio over a 20-minute session), but *individual quality incidents* (*e.g.,* duration of each buffering event) and the associated *viewer actions* (*e.g.,* skip, quit), allowing us to associate the change of engagement in response to low-quality incident within a video segment, rather than an entire session. The dataset has a favorable density of measurements—*e.g.,* among the video *segments* (15 seconds for VoD and 3 seconds for live), 90% of them have at least 6.5K unique views.

2.1 Terminology

Per-segment quality metrics: Traditional video quality is measured *per session*, by aggregating all quality-related incidents (buffering, bitrate switch, etc) during a view of a video [47, 69]. To understand users' sensitivity to per-segment quality, we first chop a VoD (or live) video into shorter segments of chunk length (Figure 2(b)). We define *per-segment* quality as a linear function of *BufRatio* (the fraction of time spent in buffering stalls), *AvgBitrate* (average bitrate in Mbps), and *BitrateSwitch* (the sum of bitrate switches in Mbps) when playing a video segment i.

$$q_i = \alpha \cdot \text{BufRatio}_i + \beta \cdot \text{AvgBitrate}_i + \gamma \cdot \text{BitrateSwitch}_i \tag{1}$$

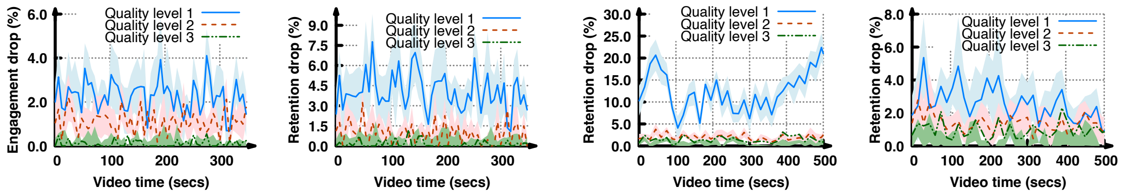
with $\alpha = -30, \beta = 1, \gamma = -1$. These weights are borrowed from prior session-wide quality models [47, 69], so if we average the segment-level quality over a video, we will conveniently get the same session-level quality as in prior work. That said, this paper does not depend on particular weights.

Per-segment quality sensitivity: For a video segment i, quality sensitivity is how per-segment quality (defined above) affects average user engagement while users watch the segment. Specifically, given a segment i and a per-segment quality level[3] q, quality sensitivity is defined in two aspects:

- *Engagement drop* is the reduction of average segment-level engagement (*i.e.,* view time) $Engage(q, i)$ when the segment has quality q from the segment-level engagement $Engage(q^*, i)$ when the segment has the perfect quality q^* (*i.e.,* no rebuffering, highest bitrate throughout).

$$EngagementDrop(q, i) = \frac{Engage(q^*, i) - Engage(q, i)}{Engage(q^*, i)} \tag{2}$$

[3] We bucketize video quality to discrete quality levels and calculate the average engagement of sessions per quality bucket. The buckets are created with a fixed bucket range of 2 (equivalently, 6.6% more buffering, 2Mbps lower bitrate, or 2Mbps more bitrate switches), starting from the highest quality q^*. For instance, the quality buckets of a video with max bitrate 6Mbps will be $(-\infty, 0.2)$ $[0.2, 2.2)$, $[2.2, 4.2)$, $[4.2, 6)$, $[6, 6]$ (*i.e.,* q^*). Appendix A describes more details.



(a) Live video (b) VoD video

Fig. 3. *Substantial variability of quality sensitivity at each quality level in an example VoD video and an example live video. The error bar represents the standard deviation of the corresponding engagement drop and retention drop.*

- *Retention drop* is the reduction of retention rate $Retention(q, i)$ (*i.e.,* fraction of viewers *not* exiting the session at the segment i) from the retention rate $Retention(q^*, i)$ under perfect quality q^*.

$$RetentionDrop(q, i) = \frac{Retention(q^*, i) - Retention(q, i)}{Retention(q^*, i)} \tag{3}$$

Caveats: These metrics of quality sensitivity are *relative*. For instance, an engagement drop of 20% does *not* mean viewers always watch 80% of the video segment; instead, it means viewers under a given low quality watch 20% less than viewers who watch the video segment in perfect quality—this normalization helps to reduce such quality-agnostic influence on engagement as low-interestingness content.

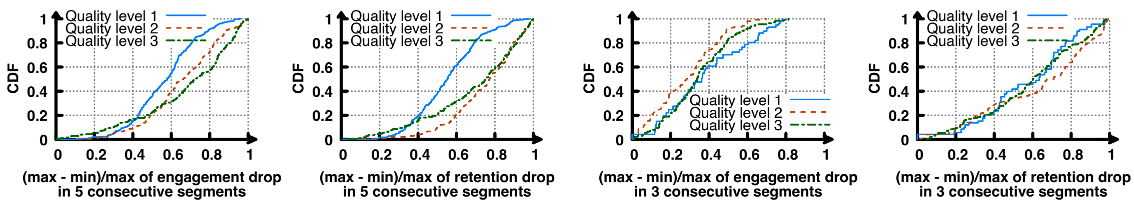
Eq. (2) and Eq. (3) assume that user engagement in a segment is associated largely with the quality of the segment. We validate this assumption by showing that exits at one segment have much more marginal correlations with other segments' quality than their correlations with the same segment's quality (details in Appendix C). Intuitively, this might be a result of the *memory effect* that the impact of historical events tend to be less than that of recent events.

For enough statistical confidence, we only compute engagement and retention drops only on video segments with >100 views at each quality level. We also restrict our analysis to videos where viewers of different segments have similar distributions of geographic locations, player platforms, and network speeds, in order to minimize confounders of the variation of quality sensitivity across segments.

2.2 Variability of quality sensitivity

We begin with the *video-wide* variability of quality sensitivity over an entire video at a given quality level and then the *session-wide* variability of quality sensitivity during a video session, in which a real user might watch a portion of a video at time-varying quality.

Variation of sensitivity with content: Figure 3 shows two concrete examples (one VoD video and one live video), which have substantial video-wide variability of quality sensitivity in both engagement drops and retention drops. The peaks of each curve mark the video segments where the average user engagement is most sensitive to low video quality during a segment. For instance, the engagement drop due to low quality (quality level 1) around 50$^{\text{th}}$ second of the VoD video is about 3× higher than around 100$^{\text{th}}$ second, and the quality sensitivity in the live video varies by 1.5-2× within tens of seconds.



(a) Live videos (b) VoD videos

Fig. 4. *Substantial variability of quality sensitivity among the consecutive segments requested by the same session.*

Dynamics of sensitivity within a session: Next, we focus on its variation within a short time window (of three consecutive video segments) during real video sessions in our dataset. Focusing on short time windows makes the findings actionable (*e.g.,* ABR algorithm in [73] can decide bitrates and buffering events for chunks within a buffer). For each time window of three segments, we calculate the sensitivity variation by $\frac{max-min}{max}$ of segment-level quality sensitivity (in engagement drops and retention drops). For instance, Figure 4 shows that for 40% of the three-segment windows in live sessions, the engagement drop (and retention drop) at the lowest quality level vary by over 51% (and 62%).

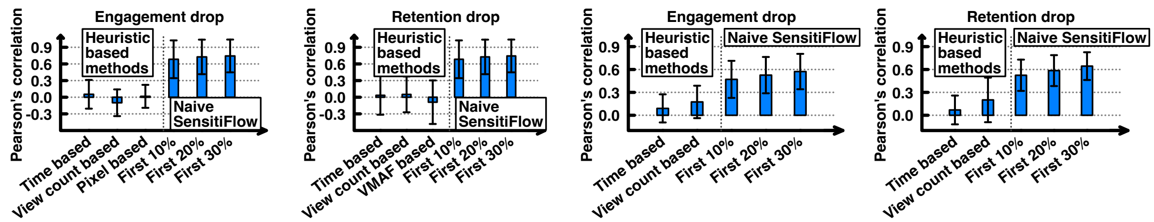
To give an intuition of how video content affects quality sensitivity, we zoom-in on two example segments. One segment is part of a fairly predictable conversation between two people. When quality is bad, viewers tend to skip the content probably in the hope to catch important content later, rather than abandoning the session, causing high engagement drop on bad quality (though not a high retention drop). The other segment has a high retention drop than an engagement drop. Upon a closer look, we found it is part of a long interlude after a scene just ends. In this case, any buffering stalls tend to cause people to abandon the session altogether rather than waiting for the interlude to end. These content-related quality sensitivity corroborate previous small-scale studies [36, 37, 73]. We stress that our goal is *not* to establish causality between content and quality sensitivity, but to show their potential correlation.

2.3 Challenge of estimating quality sensitivity

We have shown the variability of quality sensitivity, which arises as a consequence of video content's impact on user engagement. The natural question then is how to estimate the variability of quality sensitivity *in a new video*. Two high-level approaches to this problem exist but they suffer following limitations.

Survey-based methods are too slow: One approach relies on offline survey studies. It recruits a set of participants (in lab studies or on crowdsourced platforms), asks them to watch and rate the quality of multiple versions of the videos with each version being played at different video quality levels, and finally aggregates their ratings to model users' sensitivity to video quality.

However, such offline survey studies can be quite slow. A recent effort tries to automate crowdsourcing for the estimation of quality sensitivity [73], thus reducing the survey delay compared to the traditional in-lab studies (*e.g.,* [32]). Unfortunately, it still takes at least 78 minutes to accumulate enough crowsourced user ratings to profile quality sensitivity of a 30-second video, which is too slow for most live videos.



(a) VoD videos (b) Live videos

Fig. 5. *Correlations between actual quality sensitivity and different estimators. Compared to using session time, view count, and visual quality (VMAF), naive SensitiFlow (which uses early sessions) shows the highest correlation.*

Heuristics-based methods are inaccurate: Alternatively, one may use heuristics, which use pixel values or viewing history to infer quality sensitivity. They do not require offline survey studies, but how well they correlate with quality sensitivity and its variation remains unclear. Here, we consider three particular heuristics: (1) VMAF [16], a popular *pixel-based* visual quality index (*e.g.,* [50]), (2) the time position during a session, which is believed to affect users' sensitivity to video quality (*e.g.,* [45]), and (3) the popularity of the segment (the number of views of a segment), which intuitively indicates user attention when viewing the segment (*e.g.,* if viewers tend to skip a part of the video, they will likely exit when the quality is poor).

Figure 5 (the first three bars in each subfigure) shows, for each heuristic and each video, the Pearson's correlation coefficients between these heuristics and the segment-level engagement drops and retention drops of sessions watching the same video. The figure 5 shows the distribution of the correlation coefficients over the 50 videos, and we can see that they are weakly correlated (if at all) with quality sensitivity (absolute Pearson's correlation coefficients lower than 0.3). This suggests that the impact of video content on quality sensitivity might be more complex than what can be captured by these heuristics (though it remains an open question if a more sophisticated heuristic, *e.g.,* a computer-vision model, could predict quality sensitivity).

Summary of findings: In summary, this section shows that:

- Quality sensitivity varies significantly with a temporal variation of video content in both VoD and live videos. For instance, in response to the same low-quality incident, the drop of engagement can vary by 50% depending on where the low quality occurs within a 3-segment window.
- Existing methods are unable to estimate quality sensitivity both accurately and quickly. Heuristics such as VMAF are not accurate indicators of quality sensitivity (engagement drops and retention drops), while an offline survey-based study is too slow, especially for live videos.

3 SENSITIFLOW: ONLINE PROFILING AND ADAPTATION OF QUALITY SENSITIVITY

We present SensitiFlow, a controller for ABR logic of video players. Unlike prior work that relies on offline user studies, SensitiFlow profiles quality sensitivity using *online* feedback from real video sessions.

3.1 Overview of SensitiFlow

Figure 6 depicts SensitiFlow's high-level workflow. SensitiFlow's *global coordinator* constantly collects online measurements of per-segment quality metrics and engagement-related feedback (*e.g.,* exit or skip) from

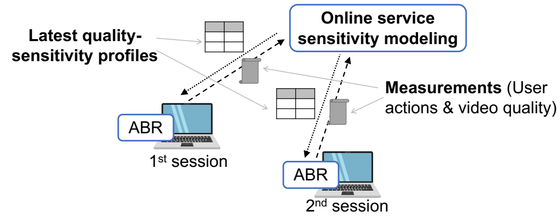


Fig. 6. Each session in SensitiFlow uses the latest quality-sensitivity profile to make ABR decisions and updates the global coordinator with the latest user actions and video quality.

Key <Segment ID, Quality level>	Value <EngagementDrop, RetentionDrop>
<Seg 1, QualityLevel 2>	<20%, 18%>
<Seg 2, QualityLevel 2>	<13%, 9%>
<Seg 5, QualityLevel 1>	<25%, 29%>
...	...

Fig. 7. An example quality-sensitivity profile: a key-value mapping from each quality level of a segment to the user sensitivity to the quality (average engagement drop and retention drop) at the segment.

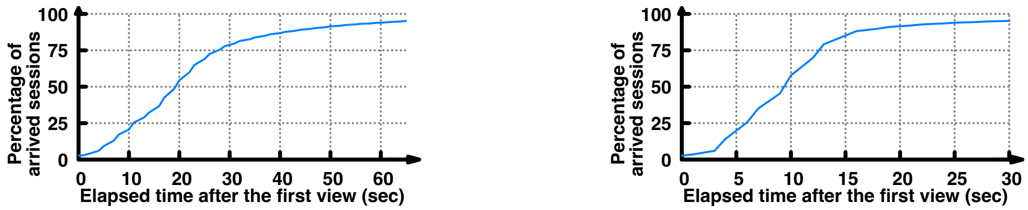
video sessions to maintain an up-to-date view of the *quality-sensitivity profile* of each video. It then shares the profile with each client, which makes ABR decisions based on both dynamic network conditions and the quality-sensitivity profile. This high-level framework is compatible with existing logically centralized control platforms (*e.g.*, [5, 6, 11, 35, 43, 49]), operated by content providers and third parties that have client-side instrumentation to measure viewers' reactions to video quality and share information with clients.

From an abstract view, a quality-sensitivity profile maps each segment and quality level to the estimation and variance of quality sensitivity, in engagement drops (Eq. (2)) and retention drops (Eq. (3)). Thus, it can answer the "what-if" question needed by the clients: *what would the expected drop in engagement/retention for a given quality at each segment?* When a session finishes playing a segment, it calculates the segment-level quality and engagement and sends it to the global controller to update the quality sensitivity profile. When a session's ABR logic decides the bitrate of the next video segment, it will query the quality-sensitivity profiles and execute the quality-sensitivity-aware algorithm described in the next section.

Why online profiling of quality sensitivity? Unlike offline modeling of quality sensitivity, the key advantage of SensitiFlow, as illustrated in Figure 1, is that it can profile quality sensitivity *on the fly* as more viewers of a new video joint. It thus could enable new quality-sensitivity-driven optimizations on any new videos (especially live videos), which would be *impractical* with offline profiling of quality sensitivity.

Though VoD video traffic is still important, live videos pose particular challenges for a multitude of reasons [1]. Live videos' unpredictable workloads and flashcrowd nature make demand prediction and resource provisioning particularly challenging (*e.g.*, many live events do not have enough history to predict their real popularity). At the same time, live viewers often pay more attention to content and tend to be fairly sensitive to low quality. In this context, SensitiFlow's potential to profile quality sensitivity online and improve user experience without more bandwidth is well-suited.

That said, there are two potential concerns with SensitiFlow's online profiling of quality sensitivity. First, compared to offline user studies, measuring user engagement in real video sessions could be as noisy (if not more). Fortunately, since we only use the average quality sensitivity *across* users, such measurement noise could be compensated by a large number of video sessions. Popular live videos such as those in our dataset



(a) A typical live-linear video (TV show)

(b) A typical live-event video (sports event)

Fig. 8. Example arrival patterns of views of a live video segment. Even in live-event videos, 20% of sessions watch the same content at least 3 seconds earlier than 60% of sessions.

do contain enough sessions to reduce the statistical variances. For instance, Figure 5 (the last three bars in each subfigure) shows that average quality sensitivity calculated based on the first 10-30% views of each segment is strongly correlated with the quality sensitivity calculated based on the last 70% sessions (which do not overlap with the first 10-30% views). Therefore, the online measurements of user engagement from early sessions can be used to accurately predict the quality sensitivity for other sessions.

Second, if most sessions of live videos watch the same content almost simultaneously, it will cripple the possibility of using the quality sensitivity of early sessions to optimize other sessions. Fortunately, our analysis shows that most views of a live video segment occur during a non-trivial time span of 30-40 seconds after it first being viewed. Figure 8a and 8b show the relative wall-clock time of sessions watching a chunk of a live-linear video and a live-event video As shown in Figure 8, live linear streaming[8] refers to those 24/7 live programs. The content of live linear videos usually includes talk shows, TV plays, and movies. Live event streaming represents those standalone live programs of a few hours, such as sports events.

Our conversation with domain experts has confirmed that such time discrepancies among live video viewers is commonly accepted. While during the early days of the internet video industry, users expected to watch live broadcasts synchronously (much like today's video conferencing), over the last decades, the modest time difference (of 10-30 seconds) among viewers has become an accepted feature (rather than a bug) of live internet videos (*e.g.*, people usually share live scores over out-of-band channels, such as chat or social media), and the industry has been lukewarm to increase system complexity for reduced asynchronicity[4]. As a result, the view pattern shown in Figure 8 is unlikely to change in the conceivable future.

3.2 Profiling and adaptation of quality sensitivity

So far, we have seen that online profiling of quality sensitivity *could* be feasible. To this end, SensitiFlow's ABR control logic needs to use the latest quality-sensitivity profile to maximize the overall engagement for sessions watching the same video while avoiding any sessions having worse video quality than a default ABR logic agnostic to quality sensitivity. To motivate our design, let us first consider a natural strawman which works in two phases.

[4]Some notable exceptions are online conferencing and gaming. In both cases, synchronicity is crucial, but they target a much smaller group of audience compared to large-scale live events, and they are built on a different software stack (based on WebRTC and UDP).

Profiling phase: For each video, a *base ABR logic* (*e.g.,* FastMPC[69]) is used by the first N sessions, whose per-segment user engagement and quality metrics are collected and used to estimate the quality sensitivity of the segment. N should be set such that the quality sensitivity of each segment and each quality level can be reliably estimated based on the feedback of these sessions (Figure 5). The value of N is dependent on the popularity of the video and the network conditions of its viewers, and we found $N = 4000$ works empirically better in our dataset than other values. This phase produces reliable estimates of quality sensitivity without negatively impacting the early sessions' quality compared to the default ABR logic.

Optimization phase: After N sessions, each session runs a variant of the quality-sensitivity-aware ABR algorithm proposed in [73]. The algorithm takes as input the player's current state (history throughput, buffer length, etc) and the quality sensitivity of the next H segments in the *look-ahead* horizon, and returns as output ABR decision for the next chunk. More specifically, new ABR logic picks the action $a = (B, t)$ (bitrate and inserted rebuffering time [73] for the next chunk[5]) that maximizes the *reward* defined as

$$R(a) = -\sum_{i \in H} (l_i^{\text{seg}} \cdot EngagementDrop_i(Q(a,i)) + l_i^{\text{rest}} \cdot RetentionDrop_i(Q(a,i))) \tag{4}$$

where $Q(a, i)$ is the quality caused by the action a at segment i in the lookahead horizon H, and l_i^{seg} and l_i^{rest} are the lengths of segment i and the remaining content after i, respectively. Effectively, Eq. 4 is the negate of the expected decrement on overall engagement caused by using quality q_i at segment i, where the first term is the engagement drop per segment ($l_i^{\text{seg}} \cdot EngagementDrop_i(q_i)$) and the second term is how more likely the quality would cause viewers to exit without watching the remaining content ($l_i^{\text{rest}} \cdot RetentionDrop_i(q_i)$).

Unfortunately, this strawman is not efficient in practice on live videos (Figure 8), because the measurements collected during the profiling phase can take several seconds, which we refer to as the *propagation gap* (Figure 9), to collect (*e.g.,* viewers must watch the segments first) and update the quality-sensitivity profile of the segments in the lookahead horizon of sessions that arrive during the optimization phase. As live viewers watch the same segment within a relatively short time frame (tens of seconds), many sessions that arrive after the profiling phase may fall within the propagation gap and thus will not be optimized.

4 OPTIMIZATIONS OF SENSITIFLOW

SensitiFlow entails several optimizations to make this strawman more effective, especially for live videos.

Optimized objective: One insight is that SensitiFlow's online quality-sensitivity profiling and sensitivity-aware ABR logic can be cast as a multi-armed bandit problem: at a high level, for each session and each segment, the ABR logic selects an action from a list of available ones to maximize the overall reward (user experience) defined in Eq. (4) across sessions. Thus, we build SensitiFlow's ABR logic on the common framework of Upper Confidence Bound (UCB) algorithm [38].

[5]The action of inserting a rebuffering stall was first proposed in Sensei [73]. The idea is that if a stall is expected to occur in the next chunk, which is more quality sensitive to the current chunk, then it would make sense to deliberately add a short stall early in the hope to avoid stalls in the more quality-sensitive chunk. This is shown to give a better user experience than traditional ABR schemes which only initiate rebuffering events when the buffer is empty.

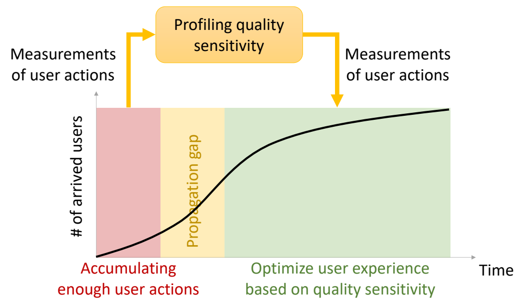


Fig. 9. An illustration of the propagation gap.

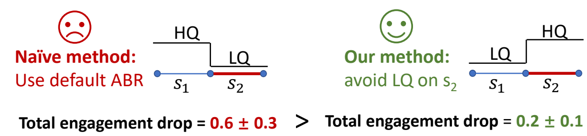


Fig. 10. An example showing the opportunity to improve a session during the profiling phase: Despite high uncertainty in sensitivity measurements, limited history data is enough to show that actively avoiding low quality (LQ) on S_2 is better than using the default ABR.

More specifically, SensitiFlow's ABR algorithm will pick the action a that maximizes the sum of the reward $R(a)$ and the upper confidence bound $U(a)$, where $R(a)$ is defined in Eq. (4). For a given lookahead horizon H, we define $U(a)$ as: $U(a) = \sum_{i \in H} \sqrt{\frac{C_{i,Q(a,i)}}{N_{i,Q(a,i)}}}$, where $Q(a,i)$ is the quality caused by the action a at segment i in the lookahead horizon H. $N_{i,Q(a,i)}$ represents the number of sessions that have already watched segment i at quality $Q(a,i)$, and $C_{i,Q(a,i)}$ is the number of sessions that have the predicted quality $Q(a,i)$ for the unwatched segment i[6]. $U(a)$ is effectively defining how UCB algorithm should explore the action space: *(i)* an action will be prioritized if it has been tried less often (*i.e.,* $N_{i,Q(a,i)}$ is smaller), and *(ii)* An action will be prioritized if later sessions are more likely to take it (*i.e.,* $C_{i,Q(a,i)}$ is larger) .

Unlike the naive method (§3.2), the optimized objective allows dynamically determining whether to apply the sensitivity-aware optimization, which enables optimizing early sessions that arrive during the profiling phase. Figure 10 illustrates an example : An early session comes during the profiling phase, and it is about to experience low quality in one of the coming segments s_1 and s_2. The naive algorithm will not optimize the session as it is in the profiling phase. However, if measurements received so far are enough to show s_2 is more sensitive to low quality than s_1, a better action can selected: *e.g.,* adding a rebuffering event in s_1 (to improve quality during s_2) instead of waiting for the buffer to draw out and buffering in s_2.

Avoid being worse than default ABR: In practice, using the UCB algorithm has a problem: when the uncertainty of the quality-sensitivity profiles is high, it may choose to explore an action that leads to the quality worse than using the default ABR. To avoid such a problem, we manually limit the action space before applying the UCB algorithm.

First, we add the actions that are highly likely better than the default ABR's action based on the current quality-sensitivity profiles. More specifically, for each action and its subsequent quality q_i at each segment i in the look-ahead horizon, the quality-sensitivity profiles return the distribution (mean-variance pair) of $EngagementDrop_i(q_i)$ and that of $RetentionDrop_i(q_i)$, based on which we can then calculate the distribution (a mean-variance pair) of the reward (Eq. (4)) of the action. Now, based on these estimations, we compare if

[6]To help calculate C_{i,q_i}, we let each video session update the global coordinator with the predicted quality levels of future video segments in the look-ahead horizon (this is already done by traditional ABR logics, such as RobustMPC [69] and Fugu [67]).

one of the actions is already very likely better than another: the action with a mean reward μ and variance σ^2 such that the other action whose mean reward μ' and variance σ'^2 is worse than the first one by a large margin, *i.e.,* $\mu - \mu' \geq (\sigma'^2 + \sigma^2)^{\frac{1}{2}}$.

Another key observation is that within a horizon, different actions may lead to similar quality, but they may help quality-sensitivity profiling to a different extent. For example, two different actions may have the same rebuffering length in the horizon while the rebuffering event happens at different segments. In such a case, they will share the same quality calculated by the QoE function defined in Eq. (1), but they will update different quality levels at different segment Therefore, we extend the action list with the actions that lead to the same QoE value (calculated by Eq. (1)), which allows us to prioritize profiling a specific quality while not worsening the overall quality of a given session.

Minimizing the propagation gap: Finally, we reduce the propagation gap by minimizing the communication and computing overhead of SensitiFlow's global coordinator. To maintain the quality-sensitivity profiles, logically the global coordinator must collect the playback quality, user actions (engagement per segment), and the predicted quality from all ongoing video sessions at the boundary of each video segment. A naive implementation therefore would have prohibitive overheads in communication and computation. For instance, in Figure 8, the peak number of concurrent sessions from video sessions is 1M per second. According to our microbenchmarking on the SparkStreaming-based global coordinator (Figure 21), the update delay would be 5 seconds (with 8 cores), which lower bounds the propagation delay. This puts a tremendous strain on live video systems whose resource provisioning is already challenging.

To reduce the overhead, we leverage the following key characteristics of our framework: By aggressively caching the history states of quality-sensitivity profiles, we only need to update them when the video player changes its state (*e.g.,* start rebuffering, bitrate switch, or user exit), which is relatively rare: by letting sessions update only when one of these events happen, we can reduce the number of updates by about 70%. We also reduce the number of requests sent by the sessions as follows. If all segments in the look-ahead horizon can already have the perfect quality, the session will not request the quality-sensitivity profiles from the global coordinator, which further reduces the number of requests by about 50% in our dataset.

5 EVALUATION

Finally, we evaluate the potential of SensitiFlow in the context of improving user engagement and/or serving more sessions without using more bandwidth. We leverage the real-world video viewing patterns and quality sensitivity derived from the dataset collected by the production streaming systems (§2). Our trace-driven experiments and real user studies show the following.

- Hypothetically, If the per-segment quality sensitivity is known in advance (*i.e.,* offline profiling but without the profiling cost or delay), under realistic video-viewing workloads, average user engagement can be improved by 8-13.5%, compared to a default ABR logic (an impressive gain based on our interaction with content providers). Using optimized online profiling of quality sensitivity, SensitiFlow can realize

60-80% of this gain (*i.e.,* normalized engagement gain), whereas the gains of a state-of-the-art ABR logic and an unoptimized version of SensitiFlow over the default ABR logic are much less.

- Using real user study, we show that SensitiFlow can improve the mean opinion score (MOS) by 40% over the baseline ABR algorithm, suggesting that SensitiFlow 's online profiling of quality sensitivity and ABR algorithm is highly effective.
- Our implementation of SensitiFlow's global coordinator can achieve a peak update and query rate of 2 million concurrent sessions using a single machine with moderate resources.

5.1 Setup

Trace-driven simulator: We built a custom simulator that takes the following configurations as input: (1) session-related (*e.g.,* each session's arrival time, hashed identifier of the video, and starting position in the video) and video-related (*e.g.,* each video's chunk length and the chunk size at each bitrate) the information obtained from the same dataset described in §2, and (2) the empirically observed probability distributions of engagement drops and retention drops at each video segment and each video quality bucket.

The simulator replays the sessions that watch the same video in their logged chronological order. In each session, the client runs a state-of-the-art ABR algorithm (FastMPC [69]) to decide when to download each chunk at which bitrate. The per-session simulation is logically equivalent to prior work (*e.g.,* [47]): each time a client requests a chunk, the simulator uses the assigned bandwidth timeseries (explained shortly) and chunk size (obtained from static information of the video) to estimate the delay of downloading each chunk, and the player updates its buffer length when a chunk is received. A client plays the content as soon as the buffer has at least one video chunk and keeps playing until the buffer drains or the simulator decides to exit based on the empirical probability distributions in the dataset.

Our trace-driven simulator embeds a necessary assumption: different users share the same distribution of quality sensitivity at the same video segment and video quality. Given the inherent randomness of user actions, it is hard to simulate the exact same actions of different users at each segment and each quality level. But we make sure that when the simulator is fed with the same segment-level quality logged in the trace of each session, the *distribution* of user engagement matches the distribution of the logged user engagement with negligible differences. (see Appendix D for details.)

Emulator testbed: We also developed an emulator for real-world experiments. As shown in Figure 11, our emulator has three components: a video content server, an emulated video player, and the global coordinator. The global coordinator maintains the measured quality-sensitivity information and sends the corresponding part to the player based on its request. The emulated player can run both baseline ABR and SensitiFlow to determine which bitrate to download for a video chunk. It then downloads the video chunks from the video content server (an HTTP server hosting the video chunks at different bitrates). We emulate the network between the video content server and player by Mahimahi [51].

Strategies: We implement five strategies.

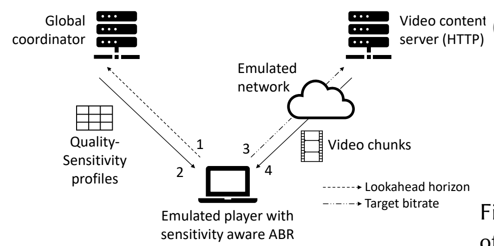


Fig. 11. Overview of the emulator

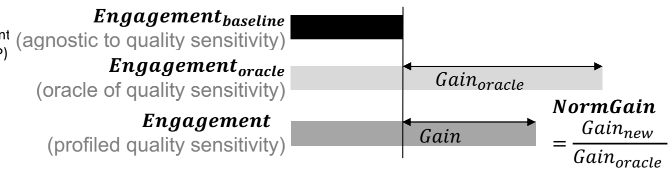
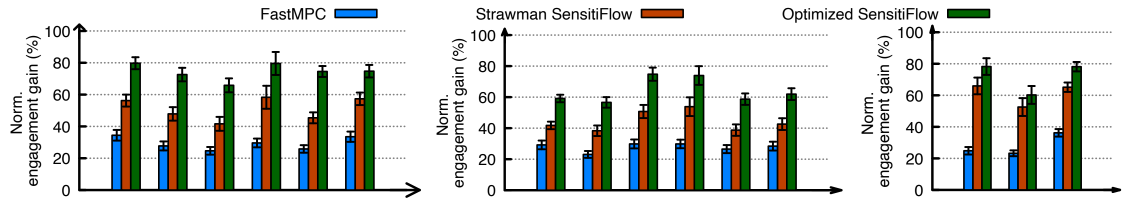


Fig. 12. Normalized engagement gain of a strategy measures how much of the gain obtained by a hypothetical system that knows quality sensitivity in advance can be obtained by the strategy under consideration.

1. *Base ABR:* We implement a base ABR logic that selects the highest possible bitrate that will not cause rebuffering in the current segment. This will serve as the basis to calculate the performance gain.
2. *FastMPC:* Decisions (bitrate adaptation) are made by maximizing quality objective in Eq. 1 using the state-of-art ABR algorithm of FastMPC [69]. This can be viewed as an intelligent ABR logic that is agnostic to the variation of quality-sensitivity *within* a video.
3. *Strawman SensitiFlow:* We apply the optimization strategy proposed in §3.2: first 4000 sessions of each video will be controlled by the baseline algorithm while their information is fed to the online quality-sensitivity profiling algorithm, and after 4000 sessions,future sessions will use the measured quality-sensitivity for optimizations. (§3.2).
4. *Optimized SensitiFlow:* Decisions are made based on the algorithm proposed in §4.
5. *Oracle:* This is a hypothetical design that optimizes SensitiFlow's objective but *knows quality sensitivity in advance*. This can be seen as an oracle point of comparison with hindsight information.

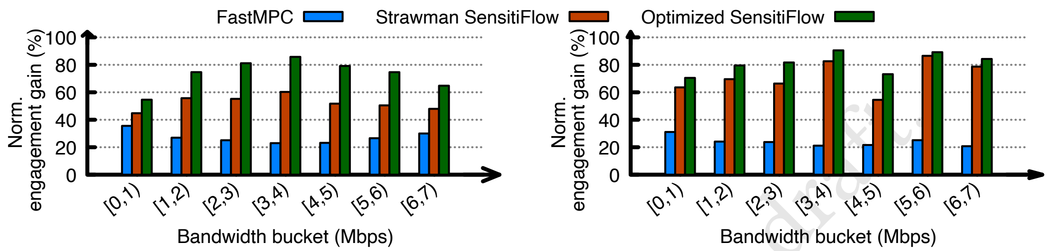
Metrics: We use the *normalized engagement gain* (Figure 12) to measure the improvement of "Baseline", "strawman", and SensitiFlow over the "Dumb ABR". Normalized engagement gain measures the increase of average engagement by the new algorithm over the increase of average engagement by the "Oracle" algorithm (*i.e.*, SensitiFlow's ABR with knowing quality-sensitivity profile for all segments in advance). For instance, a normalized engagement gain of 20% means achieving 20% of the maximum potential gain obtained by "Oracle". While we do not use absolute engagement values (which depend on the popularity of video content itself), we confirm that the engagement values are higher than 50% for most videos.

Bandwidth settings: To evaluate our quality-sensitivity-aware optimizations on realistic network conditions, we created a corpus of network traces using the public broadband dataset provided by FCC[12]. The FCC dataset contains over 1 million throughput traces, each of which logs the average throughput over 30 minutes, at a 5-second granularity. We generate 2000 traces for our test set, each with a duration of 5000 seconds, by concatenating randomly selected traces from the "Web browsing" category in the June 2015 collection. We only selected original traces whose average throughput ranges from 0.5 to 7.5 Mbps, as it covers the average logged bitrate of each session in our measurement.



(a) Different live linear videos (b) Different live event videos (c) Different VoD videos

Fig. 13. *On various live and VoD videos, SensitiFlow achieves much more engagement improvement than the baselines.*



(a) Live videos (b) VoD videos

Fig. 14. *Engagement gains grouped by different available bandwidth buckets. Though the relative differences vary, SensitiFlow is consistently better.*

5.2 Improvement analysis

Engagement improvement across videos: Figure 13 shows that, across different videos, the normalized engagement gains of the strawman method are 61-69% and 52-64% when using the session arrival pattern of live linear videos and live events (an example is shown in Figure 8), respectively. With the optimizations proposed in §4, the normalized engagement gain can be further improved to 76-84% and 66-81% for both types of live videos. SensitiFlow can also provide a gain of 73-85% for VoD videos (though the improvements over the strawman strategy is relatively marginal). These findings suggest that without more bandwidth resources, existing optimizations (by leveraging quality sensitivity) can achieve 60-80% of the maximum improvement achievable by the oracle system that knows the exact quality sensitivity in advance. We also make sure that the improvement of "Oracle" itself is non-trivial: among all the videos, it can improve the average engagement by 8-13.5%.

Impact of bandwidth on engagement improvement: Figure 14 shows the normalized engagement gain of both the strawman and optimized SensitiFlow grouped by sessions under various average bandwidth buckets. In all bandwidth buckets, the optimized SensitiFlow has normalized engagement gains about 64-89% and 78-93% for live and VoD videos, respectively. Generally, SensitiFlow is most effective when the available bandwidth is when ABR logic is most needed (when bandwidth is in the moderate range.)

Fairness of improvements: One concern is that SensitiFlow might lead to unfair outcomes as later sessions get more improvements. We compare Jain's fairness index of engagement across video sessions grouped by the available bandwidth. The result shows Jain's fairness index of all the methods is larger than 0.95 and SensitiFlow rarely decreases the fairness index compared to the baseline.

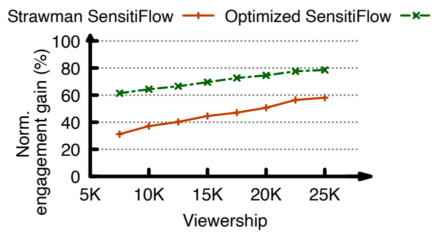
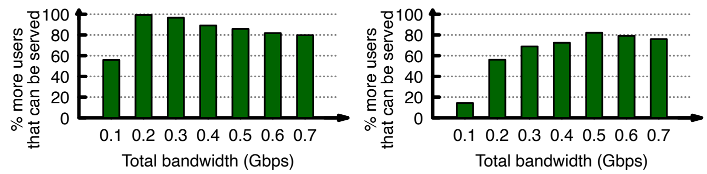


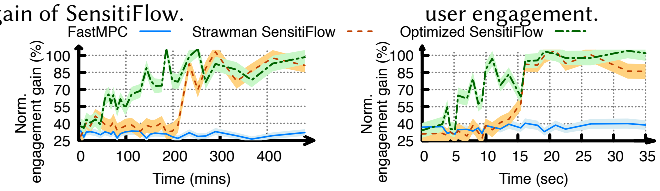
Fig. 15. The impact of viewership on the gain of SensitiFlow.



(a) Live video

(b) VoD video

Fig. 16. When sessions share a bandwidth bottleneck, SensitiFlow can serve 15-100% more sessions than the base ABR while keeping similar user engagement.



(a) VoD video

(b) Live video

Fig. 18. Trace-driven emulation: Effectiveness of different strategies on sessions with different delays when watching the same segment.

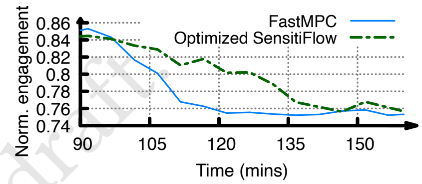
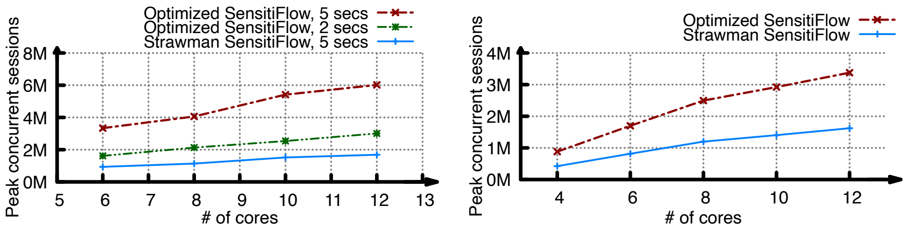


Fig. 19. Emulated test under stress. After a 100% surge in arrived sessions, SensitiFlow has much higher engagement than the baseline.

Impact of video popularity: Figure 15 shows the normalized engagement gain on a live video with the same arrival pattern distribution but different total numbers of views, *i.e.,* different number of sessions will come within the same period of time. Optimized SensitiFlow achieves normalized engagement gains around 65-81% when the viewership is ranging from 7.5K to 25K. SensitiFlow's gains approach the oracle strategy at a higher number of views, though with a non-negligible gap: this is because, despite the optimization of SensitiFlow, the propagation delay cannot be completely eliminated.

Serving more sessions without more bandwidth: We also evaluate that when a bottleneck link is being shared, how many users SensitiFlow can serve while keeping the same engagement same as using the baseline. We start the experiment by simulating multiple sessions sharing the same bottleneck link using the baseline and the optimized SensitiFlow's ABR, and keeping their engagement the same. Figure 16 shows that SensitiFlow can serve 50-100% more concurrent video sessions for live videos (15-80% more for VoD videos) while maintaining the user engagement same as the baseline.

Engagement gain over time: To show how the performance of different strategies evolves, we run an emulation on 25 sessions coming at different time points (as shown on the x-axis of Figure 18), while the global coordinator simulates a whole population of 20K sessions following the arrival timeseries shown in Figure 8. Figure 17a shows that on a VoD video, the strawman strategy ends its profiling phase around the 200th minute, after which the normalized engagement gain climbs up from around 35% to 75-100%. In contrast, optimized SensitiFlow can further improve the sessions arrived between the 50th and the 250th minute with a gain ranging from 55-80%. Figure 17b presents a similar trend in a live video: optimized SensitiFlow can further improve the engagement of sessions arriving between the 5th and the 14th second, which accounts for 32% of total sessions.



(a) Measurement updates (b) Decision query

Fig. 21. The performance of the global coordinator. With 8 CPU cores, SensitiFlow can handle the requests from 2M concurrent sessions, while updating the quality sensitivity profiles every 2 seconds.

Performance under stress: Finally, we evaluate the performance of SensitiFlow under a scenario that the system is "under stress" by the emulation. We start the experiment by letting all sessions share the same bottleneck link. At the 95$^{\text{th}}$ minute, the session arrival rates doubled, but the bottleneck link bandwidth does not change. Figure 19 shows the engagement normalized against the maximum potential engagement (*i.e.,* when the session has the perfect video quality among all the segments) for different strategies facing such overloading: while the baseline algorithm shows a fast drop in the engagement, the proposed strategy can maintain the user's engagement for a longer time.

5.3 Scalability of Global Coordinator

In SensitiFlow, the component that might become a performance bottleneck is the global coordinator (Figure 6) which maintains the latest view of the quality-sensitivity profiles. Other components (*e.g.,* client-side instrumentation) are already used by current content providers and third parties (*e.g.,* [5, 6, 11, 35]).

Prototype implementation: We build a prototype of the global coordinator on a server that runs Ubuntu 18.04 with 64G RAM and 32-core Platinum 8269CY 3.10GHz CPU. The quality sensitivity measurements (each of 10-20Bytes) are fed to a cluster with Kafka [3] and SparkStreaming [71]. The SparkStreaming instance will read the measurements from Kafka and then update per-segment quality-sensitivity profiles maintained in an RDD [70] in a streaming fashion. SparkStreaming uses micro-batches to update RDDs, instead of merging each update immediately. This implementation fits our needs, since the freshness requirement of quality-sensitivity profiles is in seconds, not milliseconds.

Scalability tests: We test the performance of the prototype with the optimization proposed in §4 (optimized SensitiFlow) vs. without the optimization (the strawman strategy). Figure 21 shows the throughput of both methods, in terms of measurement updates per second and queries per second. We see that the global coordinator scales out horizontally with more compute resources. With 8 CPU cores, the prototype of optimized SensitiFlow can handle the updates and the requests from 2M and 2.5M concurrent sessions, respectively, while updating the quality-sensitivity profiles every 2 seconds.

To put these numbers in context, the most demanding workloads are large-scale live events, where millions of sessions tune in to watch a video in a short period of time and the profiling has to be done with a short delay. As Figure 8b shows, a propagation gap of 5 seconds can exclude about 40% of sessions from

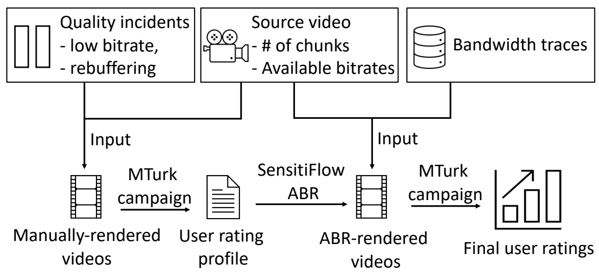


Fig. 22. Methodology of real-user study

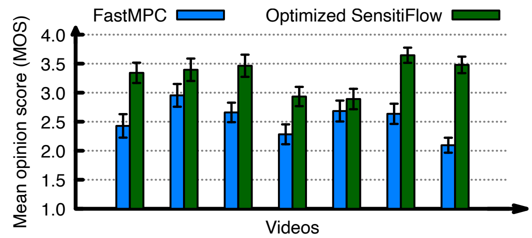


Fig. 23. User study result. Comparing mean opinion score (MOS) of the baseline ABR (FastMPC) and SensitiFlow on seven sampled videos.

the quality-sensitivity-based optimization. However, by reducing the update delay of the global coordinator to 2 seconds, the ratio of sessions arriving during the propagation gap will decrease to only 15%.

5.4 Real User Study

To complement the trace-driven evaluation, we also set up a real user study on Amazon MTurk [2] to test SensitiFlow's effectiveness in the real world. We compare two particular strategies: optimized SensitiFlow and the baseline ABR logic (FastMPC) which is agnostic to quality sensitivity profiles. We run k batches of tests (*i.e.,* MTurk campaigns), each collecting user ratings from a fixed number of participants. When an MTurk user signs up for our study, the user will be *randomly* assigned to rate their user experience (with a Likert scale of 1-5) on the videos rendered by one of the two strategies (we will explain how the videos are rendered shortly). This avoids any user-specific confounders that bias their ratings of different algorithms. Figure 23 shows the average rating (*i.e.,* mean opinion score or MOS) on the videos from each strategy.

A key difference between the user study and previous evaluations is that here we measure user experience by their average rating (MOS), rather than user engagement or quit rate, of each strategy. While it has a pragmatic reason not to rely on MTurkers' user engagement[7], MOS has been a standard user experience metric [73], and it shows the versatility of SensitiFlow to take different user experience metrics as input.

When the k-th MTurk campaign finishes, SensitiFlow will update the quality-sensitivity profile by the user ratings of users assigned to SensitiFlow in the previous k campaigns, which will be used to decide the quality levels (which bitrate for each chunk and where rebuffering stalls occur) during the $k+1$-th campaign. For simplicity, we choose not to update the quality sensitivity of SensitiFlow, within each MTurk campaign, so the tested SensitiFlow uses the same quality-sensitivity-aware ABR logic, but the online quality-sensitivity profile is updated once every campaign.

Due to proprietary reasons, we do not use the videos from the measurement dataset. Instead, we randomly sample 7 test videos from a public user-generated dataset, YouTube-UGC [63] covering three video genres: sports, gaming and animation. Video chunk (segment) length is 3 seconds and the available bitrates are {1000, 4000, 9000}Kbps, and the bandwidth trace is between 2.5Mbps to 7.5Mbps, forcing ABR algorithm to

[7]MTurkers will tend to always quit a session early if they are paid a fixed amount, and they will watch the video fully if their compensation is proportional to how long they watch.

adapt their bitrates. Each campaign will wait for 30 participants selected and calibrated by the same criteria specified in [73]. Each strategy has accumulated ratings from 840 Master MTurkers [8] of age between 25 and 50. SensitiFlow outperforms the baseline among all the videos we selected. On average, SensitiFlow' ABR improves the MOS by 40% compared to the baseline ABR, from 2.53 to 3.30.

6 RELATED WORK

Modeling perceptual quality: Traditional approaches focus on how user's perception is affected by pixel-level distortion (*e.g.,* [16, 39, 44, 55, 58, 65]) and streaming-related metrics, such as rebuffering and quality switches (*e.g.,* [19–23, 26, 28, 29, 31–33, 45, 48]), and inferring video-specific metrics from encrypted network traffic (*e.g.,* [25, 41]). Modeling the impact of video content on user experience has only recently gained attention [36, 37, 73]. Our work differs on two key fronts. The first is scale. Prior works use small-scale experiments, while our study is based on event-level logs from millions of sessions, which reveals to what extent quality sensitivity varies within sessions and across sessions. Second, they fall short of providing a viable strategy to accurately predict quality sensitivity for any new (live) video in an online fashion. Using real measurements, we make a case for using feedback from real users to profile quality sensitivity online.

Video content popularity and user behavior: The rapid growth of the Internet video industry has spurred research towards better modeling of content popularity, including video genre or video virality (*e.g.,* [24, 34, 52, 53, 61]), per video (*e.g.,* [27, 42]) or per segment (*e.g.,* [60, 64]). Parallel to the modeling of content popularity, there have also been efforts to understand user behavior when watching online/live videos, in particular, the abandonment behavior (*e.g.,* [20, 29, 45, 46, 57]) and more recently user migration across platforms (*e.g.,* [68]), but these studies analyze user behaviors at the session level, with little attention on the impact of time-varying video content.

Control decisions in video streaming: Most commercial video players today implement client-side bitrate adaptation based on industry standards [10] and a range of ABR algorithms (*e.g.,* [30, 47, 59, 67, 69, 72]). Besides client-side adaptation, resource allocation in the network and content placement has also led to many research efforts (*e.g.,* [40, 42, 50, 66]), some of which also leverages the heterogeneity in popularity across video segments. In this context, the goal of SensitiFlow is not to propose new mechanisms for quality optimization; rather, it presents a solution to enable existing solutions to utilize the variability of quality sensitivity in video sessions.

7 DISCUSSION

While we study only video systems in this paper, we think that the general approach of SensitiFlow may be applicable to other network applications such as gaming and mobile-web. In particular, the *online* control loop should include user engagement and actions as real-time input. In the case of SensitiFlow, to improve user experience under limited resources, the system continuously monitors online user actions (*e.g.,* exit,

[8] Master MTurkers are a class of reliable MTurkers who have participated in over 1000 surveys and whose feedback was accepted for over 99% of their prior surveys

skip) from video sessions to estimate *quality sensitivity* and uses it to drive online adaptation. Similar observations are made in other contexts too. For instance, users' tolerance to web page loading time is more difficult to capture by static analysis on page content than by observing users' natural actions (*e.g.,* [62, 74]).

SensitiFlow shows the early promise of a more *user-centric* approach, where *measurements on user experience and actions* are first-class citizens of system monitoring and optimization. Just like systems metrics indicate current system states, user actions and engagement reveal individual user's experience, as they watch a video, browse a web page, or use a mobile app. In SensitiFlow, we use video view time as user metric, but there is a broad range of choices for user metrics, some are readily measurable (*e.g.,* user rating of apps, how long a web user stays active on a web page/site, how often Zoom users ask for others to repeat themselves) and some will be in the near future (*e.g.,* gaze tracking, brain-signal acquisition). Current instrumentation for these signals may be subject to data noise and sparsity, but once they can be measured with sufficient precision and intensity, more research will be needed towards the efficient and automatic measurement of user experience so that systems can be driven directly by user experience measurements.

The user-centric approach also calls for novel system designs which adapt both spatially (across users) and temporally (as a user interacts with the application). This may resemble prior systems that are cognizant of differences among video genres [50, 54, 56] or viewing devices [21]). However, a key distinction is that these differences are known prior to the start of a session, whereas the nature of user perception means that it can only be measured online, and often with non-trivial delays (users do not react as fast as system metrics). Thus, a user-centric approach must feature a *tight* control loop between user experience measurement and system adaptation. SensitiFlow takes a step in this direction by utilizing the limited user feedback about quality sensitivity. Working toward such a perception-driven system is an active direction of future research.

8 CONCLUSION

We present the first large-scale measurements that reveal the dynamics of quality sensitivity during real video sessions, and present the first online system, SensitiFlow, that automatically estimates quality sensitivity as new videos (including live videos) are streamed to users. SensitiFlow shows that online user feedback (*e.g.,* exit, replay and skip) from real video sessions can be used to model the fluctuation of quality sensitivity, and presents a scalable controller that collects online feedback from massive concurrent sessions and jointly adapt their bitrates to cope with dynamic network conditions and dynamic quality sensitivity. Our trace-driven simulation and emulation show that user engagement of both VoD and live videos can be substantially improved without using more bandwidth resource. Our real user study also confirms SensitiFlow can significantly improve the mean opinion score on real users

Ethics Considerations: The measurement study on user quality traces presented as part of this work is IRB-approved. In the dataset shared by industry, all user-sensitive personal identifiable information (PII) is appropriately anonymized before the data was shared with the research team. Thus, this work does not raise any ethical issues.

REFERENCES

[1] A closer look at low latency delivery. https://nscreenmedia.com/live-uhd-not-factor-ip-video-growth/.

[2] Amazon Mechanical Turk. https://www.mturk.com/.

[3] Apache kafka. https://kafka.apache.org/.

[4] CBS All Access app had technical difficulties at the start of the Super Bowl. https://www.theverge.com/2021/2/7/22271648/cbs-all-access-app-problems-super-bowl.

[5] Cedexis. https://www.cedexis.com/.

[6] Conviva. https://www.conviva.com/.

[7] DAZN hits 65m global live streams over single sporting weekend. https://www.sportspromedia.com/news/dazn-streaming-ott-broadcast-figures-serie-a-bundesliga-motogp.

[8] Linear Live Streaming 101. https://antmedia.io/linear-live-streaming-101/.

[9] Measuring Video Quality and Performance: Best Practices. https://www.akamai.com/content/dam/site/it/documents/white-paper/measuring-video-quality-and-performance-best-practices.pdf.

[10] MPEG-DASH. https://mpeg.chiariglione.org/standards/mpeg-dash.

[11] Nicepeopleatwork. https://www.nicepeopleatwork.com/.

[12] Raw data - measuring broadband america - seventh report. https://www.fcc.gov/reports-research/reports/measuring-broadband-america/raw-data-measuring-broadband-america-seventh.

[13] Streaming's high latency—No one cares, but you still should? https://www.fiercevideo.com/video/streaming-s-high-latency-no-one-cares-but-you-still-should-ring.

[14] Super Bowl 2021 was the most-streamed NFL game ever. https://www.theverge.com/2021/2/9/22274255/super-bowl-55-2021-most-streamed-nfl-game-record.

[15] Video Streaming Latency Report 2020. https://www.wowza.com/wp-content/uploads/Streaming-Video-Latency-Report-Interactive-2020.pdf.

[16] VMAF - Video Multi-Method Assessment Fusion. https://github.com/Netflix/vmaf.

[17] Why does my YouTube keep skipping? https://techshift.net/why-does-my-youtube-keep-skipping/#Is_streaming_live_TV_delayed.

[18] Why Your Live Stream Lags: Intro to Live Streaming Latency. https://www.boxcast.com/blog/live-stream-video-latency.

[19] Adnan Ahmed, Zubair Shafiq, Harkeerat Bedi, and Amir Khakpour. Suffering from buffering? detecting qoe impairments in live video streams. In *2017 IEEE 25th International Conference on Network Protocols (ICNP)*, pages 1–10, 2017.

[20] Adnan Ahmed, Zubair Shafiq, and Amir Khakpour. Qoe analysis of a large-scale live video streaming event. In *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*, pages 395–396, 2016.

[21] Athula Balachandran, Vyas Sekar, Aditya Akella, Srinivasan Seshan, Ion Stoica, and Hui Zhang. Developing a predictive model of quality of experience for internet video. *ACM SIGCOMM Computer Communication Review*, 43(4):339–350, 2013.

[22] Christos G Bampis and Alan C Bovik. An augmented autoregressive approach to http video stream quality prediction. *arXiv preprint arXiv:1707.02709*, 2017.

[23] Christos G Bampis, Zhi Li, and Alan C Bovik. Continuous prediction of streaming video qoe using dynamic networks. *IEEE Signal Processing Letters*, 24(7):1083–1087, 2017.

[24] Frank Bentley, Max Silverman, and Melissa Bica. Exploring online video watching behaviors. In *Proceedings of the 2019 ACM International Conference on Interactive Experiences for TV and Online Video*, pages 108–117, 2019.

[25] Francesco Bronzino, Paul Schmitt, Sara Ayoubi, Guilherme Martins, Renata Teixeira, and Nick Feamster. Inferring streaming video quality from encrypted traffic: Practical models and deployment experience. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(3):1–25, 2019.

[26] Chao Chen, Lark Kwon Choi, Gustavo De Veciana, Constantine Caramanis, Robert W Heath, and Alan C Bovik. Modeling the time—varying subjective quality of http video streams with rate adaptations. *IEEE Transactions on Image Processing*, 23(5):2206–2221, 2014.

[27] Yishuai Chen, Yong Liu, Baoxian Zhang, and Wei Zhu. On distribution of user movie watching time in a large-scale video streaming system. In *2014 IEEE International Conference on Communications (ICC)*, pages 1825–1830. IEEE, 2014.

[28] Johan De Vriendt, Danny De Vleeschauwer, and David Robinson. Model for estimating qoe of video delivered using http adaptive streaming. In *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, pages 1288–1293. IEEE, 2013.

[29] Florin Dobrian, Vyas Sekar, Asad Awan, Ion Stoica, Dilip Joseph, Aditya Ganjam, Jibin Zhan, and Hui Zhang. Understanding the impact of video quality on user engagement. *ACM SIGCOMM Computer Communication Review*, 41(4):362–373, 2011.

[30] Yingfei Dong, Zhi-Li Zhang, and David Hung-Chang Du. Full-sharing: efficient bandwidth scheduling for video streaming over broadband cable networks (bcns). *Multimedia Tools and Applications*, 33(2):131–156, 2007.

[31] Zhengfang Duanmu, Wentao Liu, Diqi Chen, Zhuoran Li, Zhou Wang, Yizhou Wang, and Wen Gao. A knowledge-driven quality-of-experience model for adaptive streaming videos. *arXiv preprint arXiv:1911.07944*, 2019.

[32] Zhengfang Duanmu, Kai Zeng, Kede Ma, Abdul Rehman, and Zhou Wang. A quality-of-experience index for streaming video. *IEEE Journal of Selected Topics in Signal Processing*, 11(1):154–166, 2016.

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

[34] David Foster. Factors influencing the popularity of youtube videos and users' decisions to watch them. 2020.

[35] Aditya Ganjam, Faisal Siddiqui, Jibin Zhan, Xi Liu, Ion Stoica, Junchen Jiang, Vyas Sekar, and Hui Zhang. C3: Internet-scale control plane for video quality optimization. In *12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15)*, pages 131–144, 2015.

[36] Guanyu Gao, Linsen Dong, Huaizheng Zhang, Yonggang Wen, and Wenjun Zeng. Content-aware personalised rate adaptation for adaptive streaming via deep video analysis. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–8. IEEE, 2019.

[37] Guanyu Gao, Huaizheng Zhang, Han Hu, Yonggang Wen, Jianfei Cai, Chong Luo, and Wenjun Zeng. Optimizing quality of experience for adaptive bitrate streaming via viewer interest inference. *IEEE Transactions on Multimedia*, 20(12):3399–3413, 2018.

[38] Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In *International Conference on Algorithmic Learning Theory*, pages 174–188. Springer, 2011.

[39] Rafael C Gonzalez, Richard Eugene Woods, and Steven L Eddins. *Digital image processing using MATLAB*. Pearson Education India, 2004.

[40] Yashuang Guo, Qinghai Yang, F Richard Yu, and Victor CM Leung. Dynamic quality adaptation and bandwidth allocation for adaptive streaming over time-varying wireless networks. *IEEE Transactions on Wireless Communications*, 16(12):8077–8091, 2017.

[41] Craig Gutterman, Katherine Guo, Sarthak Arora, Trey Gilliland, Xiaoyang Wang, Les Wu, Ethan Katz-Bassett, and Gil Zussman. Requet: Real-time qoe metric detection for encrypted youtube traffic. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 16(2s):1–28, 2020.

[42] K-W Hwang, David Applegate, Aaron Archer, Vijay Gopalakrishnan, Seungjoon Lee, Vishal Misra, Kadangode K Ramakrishnan, and Deborah F Swayne. Leveraging video viewing patterns for optimal content placement. In *International Conference on Research in Networking*, pages 44–58. Springer, 2012.

[43] Junchen Jiang, Shijie Sun, Vyas Sekar, and Hui Zhang. Pytheas: Enabling data-driven quality of experience optimization using group-based exploration-exploitation. In *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*, pages 393–406, 2017.

[44] Woojae Kim, Jongyoo Kim, Sewoong Ahn, Jinwoo Kim, and Sanghoon Lee. Deep video quality assessor: From spatio-temporal visual sensitivity to a convolutional neural aggregation network. In *Proceedings of the European Conference on Computer Vision*

(ECCV), pages 219–234, 2018.

[45] S Shunmuga Krishnan and Ramesh K Sitaraman. Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs. *IEEE/ACM Transactions on Networking*, 21(6):2001–2014, 2013.

[46] Pierre Lebreton and Kazuhisa Yamagishi. Study on user quitting rate for adaptive bitrate video streaming. In *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2019.

[47] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. Neural adaptive video streaming with pensieve. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 197–210, 2017.

[48] M. Hammad Mazhar and Zubair Shafiq. Real-time video quality of experience monitoring for https and quic. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pages 1331–1339, 2018.

[49] Usama Naseer and Theophilus Benson. Configtron: Tackling network diversity with heterogeneous configurations. In *9th {USENIX} Workshop on Hot Topics in Cloud Computing (HotCloud 17)*, 2017.

[50] Vikram Nathan, Vibhaalakshmi Sivaraman, Ravichandra Addanki, Mehrdad Khani, Prateesh Goyal, and Mohammad Alizadeh. End-to-end transport for video qoe fairness. In *Proceedings of the ACM Special Interest Group on Data Communication*, pages 408–423. 2019.

[51] Ravi Netravali, Anirudh Sivaraman, Keith Winstein, Somak Das, Ameesh Goyal, and Hari Balakrishnan. Mahimahi: A lightweight toolkit for reproducible web measurement. *ACM SIGCOMM Computer Communication Review*, 44(4):129–130, 2014.

[52] Jounsup Park, Mingyuan Wu, Eric Lee, Bo Chen, Klara Nahrstedt, Michael Zink, and Ramesh Sitaraman. Seaware: Semantic aware view prediction system for 360-degree video streaming. In *IEEE International Symposium on Multimedia (ISM)*, 2020.

[53] Lisa Prince. Conceptualizing television viewing in the digital age: Patterns of exposure and the cultivation process. 2018.

[54] Yanyuan Qin, Shuai Hao, Krishna R Pattipati, Feng Qian, Subhabrata Sen, Bing Wang, and Chaoqun Yue. Quality-aware strategies for optimizing abr video streaming qoe and reducing data usage. In *Proceedings of the 10th ACM Multimedia Systems Conference*, pages 189–200, 2019.

[55] Abdul Rehman, Kai Zeng, and Zhou Wang. Display device-adapted video quality-of-experience assessment. In *Human Vision and Electronic Imaging XX*, volume 9394, page 939406. International Society for Optics and Photonics, 2015.

[56] Susanna Schwarzmann, Nick Hainke, Thomas Zinner, Christian Sieber, Werner Robitza, and Alexander Raake. Comparing fixed and variable segment durations for adaptive video streaming: a holistic analysis. In *Proceedings of the 11th ACM Multimedia Systems Conference*, pages 38–53, 2020.

[57] Muhammad Zubair Shafiq, Jeffrey Erman, Lusheng Ji, Alex X Liu, Jeffrey Pang, and Jia Wang. Understanding the impact of network dynamics on mobile video user engagement. *ACM SIGMETRICS Performance Evaluation Review*, 42(1):367–379, 2014.

[58] Rajiv Soundararajan and Alan C Bovik. Video quality assessment by reduced reference spatio-temporal entropic differencing. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(4):684–694, 2012.

[59] Kevin Spiteri, Rahul Urgaonkar, and Ramesh K Sitaraman. Bola: Near-optimal bitrate adaptation for online videos. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016.

[60] Thomas Steiner, Ruben Verborgh, Rik Van de Walle, Michael Hausenblas, and Joaquim Gabarró Vallés. Crowdsourcing event detection in youtube video. In *10th International Semantic Web Conference (ISWC 2011); 1st Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web*, pages 58–67, 2011.

[61] Jonas Tana, Emil Eirola, and Mats Nylund. When is prime-time in streaming media platforms and video-on-demands services? new media consumption patterns and real-time economy. *European Journal of Communication*, 35(2):108–125, 2020.

[62] Matteo Varvello, Jeremy Blackburn, David Naylor, and Konstantina Papagiannaki. Eyeorg: A platform for crowdsourcing web quality of experience measurements. In *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*, pages 399–412, 2016.

[63] Yilin Wang, Sasi Inguva, and Balu Adsumilli. Youtube ugc dataset for video compression research. In *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–5. IEEE, 2019.

[64] Zheng Wang, Jie Zhou, Jing Ma, Jingjing Li, Jiangbo Ai, and Yang Yang. Discovering attractive segments in the user-generated video streams. *Information Processing & Management*, 57(1):102130, 2020.

[65] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[66] Kun-Lung Wu, Philip S Yu, and Joel L Wolf. Segment-based proxy caching of multimedia streams. In *Proceedings of the 10th international conference on World Wide Web*, pages 36–44, 2001.

[67] Francis Y Yan, Hudson Ayers, Chenzhi Zhu, Sadjad Fouladi, James Hong, Keyi Zhang, Philip Levis, and Keith Winstein. Learning in situ: a randomized experiment in video streaming. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, pages 495–511, 2020.

[68] Huan Yan, Haohao Fu, Yong Li, Tzu-Heng Lin, Gang Wang, Haitao Zheng, Depeng Jin, and Ben Y Zhao. On migratory behavior in video consumption. *IEEE Transactions on Network and Service Management*, 2020.

[69] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. A control-theoretic approach for dynamic adaptive video streaming over http. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pages 325–338, 2015.

[70] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauly, Michael J Franklin, Scott Shenker, and Ion Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12)*, pages 15–28, 2012.

[71] Matei Zaharia, Tathagata Das, Haoyuan Li, Timothy Hunter, Scott Shenker, and Ion Stoica. Discretized streams: Fault-tolerant streaming computation at scale. In *Proceedings of the twenty-fourth ACM symposium on operating systems principles*, pages 423–438, 2013.

[72] Ahmed H. Zahran, Jason J. Quinlan, K. K. Ramakrishnan, and Cormac J. Sreenan. Asap: Adaptive stall-aware pacing for improved dash video experience in cellular networks. *ACM Trans. Multimedia Comput. Commun. Appl.*, 14(3s), jun 2018.

[73] Xu Zhang, Yiyang Ou, Siddhartha Sen, and Junchen Jiang. Sensei: Aligning video streaming quality with dynamic user sensitivity. In *NSDI*, pages 303–320, 2021.

[74] Xu Zhang, Siddhartha Sen, Daniar Kurniawan, Haryadi Gunawi, and Junchen Jiang. E2e: embracing user heterogeneity to improve quality of experience on the web. In *Proceedings of the ACM Special Interest Group on Data Communication*, pages 289–302. 2019.

A BUCKETIZING VIDEO QUALITY

For each video, we will generate the thresholds to bucketize video quality values into quality levels. The generated thresholds should satisfy that: (1) There are at least 40K sessions in each quality level. (2) The stride between thresholds is not smaller than 2 (equivalently, 6.6% more buffering, 2Mbps lower bitrate, or 2Mbps more bitrate switches). In practice, we use the following steps to generate the thresholds:

- Step 1: Sort all the logged quality values from small to large.
- Step 2: Find the smallest value which satisfies the conditions above. Add it to the threshold list.
- Repeat step 2, until all quality values can be bucketed into quality levels. If the number of sessions in the highest quality level is less than 40K, it will be merged to the second-highest quality level.

B SEGMENT-LEVEL QUALITY SENSITIVITY

We measure *segment-level quality* by redefining the session-wide quality metrics within each segment i. There are two corner cases: (1) if a segment is not viewed in a session, then the segment-level quality/engagement

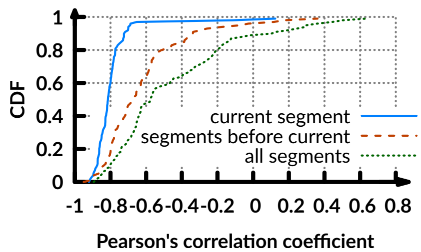


Fig. 24. *Pearson's correlation coefficient between user actions and segment quality*

is undefined; and (2) if a segment is viewed multiple times, then we take all of them into account to calculate an average segment-level quality and engagement.

We measure the user experience of a session by the viewer's *engagement*—the fraction of a video watched by the viewer before the session ends (viewer exits). While there are other aspects of user experience beyond engagement, such as user studies and opinions. They are no doubt useful, but engagement can be objectively evaluated.

We take three specific steps. First, to ensure engagement drops (Eq. (2)) and retention drops (Eq. (3)) are statistically reliable, we only consider video segments that have sufficient (>100) sessions with segment-level quality falling in each of the quality buckets. Second, we also make sure that viewers of different segments in a video have a similar distribution of geographic locations, player platforms and network speeds, so the dynamic segment-level quality sensitivity is not caused by the segments being watched by very different populations. Third, since short buffering events may naturally occur after user actions like long forward seeks that reset the buffer, we remove the buffering events subsequent to long seeks from the quality calculation.

C CORRELATION VALIDATION

To justify other segments' quality only have a negligible correlation with the current segment's user action, we consider the following three metrics: (1) The quality of the current segment. (2) The average quality of segments before the current segment. (3) The average quality of all segments in the video. For each segment, we calculate Pearson's correlation coefficient with its engagement drop. Figure 24 shows that the quality of other segments has a much weaker correlation with the user actions in the current segment: For over 60% of segments, the correlation coefficient between engagement drops and current segment quality is less than -0.8. However, there are only 17% and 10% of segments have a correlation coefficient less than -0.8 for metrics (2) and (3) respectively.

D SIMULATOR VALIDATION

To validate our simulator, we feed the logged segment-level quality from different sessions to it and compute the distribution of normalized user engagement. We assume that every session exits at the same time as in

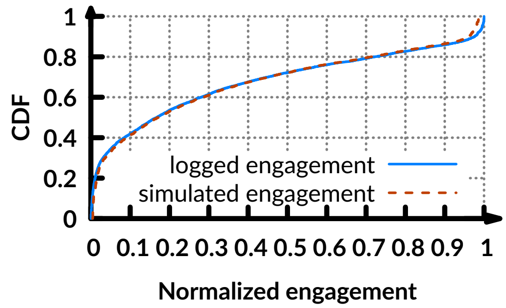


Fig. 25. *The distribution of logged user engagement and simulated user engagement*

the logged trace to remove the influence of random logouts. Figure 25 shows that the difference between distributions of logged engagement and simulated engagement is negligible.