SalientVR: Saliency-Driven Mobile 360-Degree Video Streaming with Gaze Information

Shibo Wang[1], Shusen Yang[1*], Hailiang Li[1], Xiaodan Zhang[1], Chen Zhou[1], Chenren Xu[2], Feng Qian[3], Nanbin Wang[4], Zongben Xu[1]
[1]Xi'an Jiaotong University, [2]Peking University, [3]University of Minnesota - Twin Cities, [4]Huawei

ABSTRACT

Mobile 360° video streaming has grown significantly in popularity but the quality of experience (QoE) suffers from insufficient wireless network bandwidth. The state-of-the-art solutions are limited by the temporal correlation assumption. Recent studies are aware of the potential of saliency to further QoE improvement, but several fundamental challenges about saliency judgment, saliency acquirement, and quality adaptation are still not fully addressed. To solve these challenges, we present SalientVR, a saliency-driven mobile 360° video streaming system integrated with gaze information. We design (i) a precise gaze-driven saliency judging criterion for mobile VR viewers, (ii) two pragmatic gaze-driven, tile-level saliency acquiring methods based on cross-user similarity and a specific content-aware deep neural network respectively, and (iii) a lightweight saliency-aware quality adaptation algorithm with a motion-assisted online correction, which is robust to wireless bandwidth vagaries and saliency bias. Moreover, we contribute a gaze-annotated dataset and a gaze-driven quality assessment metric for 360° videos. By extensive prototype evaluations (based on dataset tests and user studies), compared to alternatives, SalientVR significantly enhances the video quality and reduces the rebuffering ratio over 4G/LTE network emulations and in the wild, which achieves a 43.68% QoE improvement.

ACM Reference Format:
S. Wang, S. Yang, H. Li, X. Zhang, C. Zhou, C. Xu, F. Qian, N. Wang, and Z. Xu. 2022. SalientVR: Saliency-Driven Mobile 360-Degree Video Streaming with Gaze Information. In *The 28th Annual International Conference On Mobile Computing And Networking (ACM MobiCom '22), October 17–21, 2022, Sydney, NSW, Australia.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3495243.3517018

1 INTRODUCTION

The virtual reality (VR) market was valued at USD 15.81 billion in 2020 and is expected to grow at an annual growth rate of 18.0% from 2021 to 2028 [1]. As one of the most potential VR applications, mobile 360° video streaming, i.e., viewing 360° video streams on untethered mobile VR headsets, has experienced a considerable increase in popularity [2–5]. However, insufficient wireless network bandwidth limits the quality of experience (QoE). First, the delivery of high-definition 360° videos would not be fully supported by the current wireless network capacity. A 4K-resolution 360° video demands at least 25 Mbps bandwidth [6], while the LTE speeds are only 5-12 Mbps in many regions [7]. Second, mobile 360° video streaming would suffer from unexpected rebuffering due to highly variable wireless network bandwidth [8]. Either low-definition VR display or rebuffering would severely degrade the QoE for mobile VR viewers, such as disorientation and nausea [9]. Thus, enhancing the QoE of mobile 360° video streaming under limited, dynamic wireless network bandwidth has become critical and urgent.

The majority of previous research adopts *head movement trajectory (HMT)-driven* optimization methods [10–15]. A basic assumption underlying their approaches is that the HMT of a viewer has a strong temporal correlation, which means that the future HMT can be accurately predicted by the historical HMT alone during the playback. However, this temporal correlation would not always be strong enough to achieve accurate HMT predictions, which would severely degrade the QoE in the HMT-driven solutions [10]. Moreover, since the prediction horizon of HMT has to exceed buffer occupancy, the HMT-driven approaches are obliged to set a small buffer size (i.e., maximum buffer length) to reduce prediction horizons due to less correlation with the longer interval. The scarce buffer would lead to a high risk of rebuffering, especially under highly dynamic wireless network environments.

To address the above limitations, we present SalientVR, a saliency-driven mobile 360° video streaming system integrated with gaze information. Saliency [16, 17] is a term in computer vision literature, representing the property of grabbing attention for an object or a region. Saliency provides content-aware prior information for accurate long-term attention estimations without depending on the HMT [16], and therefore, it offers potential for further QoE improvement. Recently, saliency has been investigated for saving bandwidth in 360° video compression and transmission [18, 19], but several fundamental challenges are still not fully addressed in saliency-driven mobile 360° video streaming.

First, the precise judgment of saliency groundtruth for mobile VR viewers is a fundamental issue but has not been studied by previous researches. The saliency groundtruth determines the saliency degrees of different spatial regions in a 360° video frame from a human-centric perspective. Most of the previous works assume that the head-determined viewport region is equally high-salient. However, it is imprecise because only a small percentage region of viewport is viewed with high visual acuity [20, 21]. Adopting the imprecise saliency judgment would miss the opportunity of more aggressive quality allocation strategies to further improve the perceived quality. Therefore, we ran a user study to build a more precise saliency judgment criterion for mobile VR viewers leveraging gaze information that is regarded as the more precise human attention proxy [22, 23].

Second, 360° video streaming over saliency requires the tile-level saliency map but still suffers from the inaccurate, impractical

* Corresponding author. Email: shusenyang@mail.xjtu.edu.cn.

ACM MobiCom '22, October 17–21, 2022, Sydney, NSW, Australia

ACM ISBN 978-1-4503-9181-8/22/10…$15.00
https://doi.org/10.1145/3495243.3517018

saliency map acquirement. A tile-level saliency map is an $M \times N$ (same as tile segmentation) numerical matrix whose entries are saliency scores of different tiles in a video chunk, representing the saliency degrees (i.e., the degrees of viewers' interest) of different tile regions. The saliency maps of a video are prepared offline for potential users that have not watched this video before as their attention distribution predictions. Unfortunately, existing efforts [24–26] fail to achieve satisfactory accuracies of saliency analysis. Moreover, viewing data (e.g., gaze points) of users are crucial for accurate saliency acquirement in practice. However, enough gaze/head data could hardly be collected in some cases. Therefore, to promote both accuracy and practicality, we pragmatically design two new saliency acquiring methods, based on cross-user similarity and deep neural network (DNN)-based content analysis (i.e., a new specific DNN, TSMNet), respectively.

Third, existing quality adaptation strategies fail to tap the full potential of saliency for QoE improvement of mobile 360° video streaming. It is promising but challenging to utilize saliency to achieve an excellent tradeoff between quality and rebuffering under variable wireless network conditions. Huge search space of tile-level quality allocations is also troublesome, which would cause the unacceptable calculation delay for crude solving schemes (e.g., exhaustive search). Moreover, the saliency maps are determined by collective preferences without considering individual viewers. Inevitable saliency bias would influence the QoE of outlier viewers. Therefore, we design a lightweight saliency-aware quality adaptation algorithm with a motion-assisted online correction, which is robust to bandwidth vagaries and saliency bias by combining offline saliency information and online viewing data.

Gaze data is indispensable annotations for precise 360° video quality assessment (VQA) [23]. However, to our knowledge, the videos in public gaze datasets [23, 27, 28] are too short (< 1 minute) to evaluate the effects of long-term network variations and behavioral changes, and the gaze-driven VQA for mobile VR viewers is still unexplored. Therefore, for more solid algorithm evaluations, we built the first gaze-annotated long 360° video dataset[1] and the first gaze-driven 360° VQA metric for mobile VR viewers. Note that SalientVR does not require gaze tracking on-the-fly and users without gaze tracking devices can also benefit from SalientVR, as long as enough gaze data is collected offline.

To our knowledge, SalientVR is the first mobile 360° video streaming system with the integration from gaze to saliency throughout the system. We implemented a prototype of SalientVR and conducted extensive evaluations (both dataset-based evaluation and survey-based user study) over LTE network emulations and in the wild. The dataset-based evaluation, consisting of more than 150 hours of playback, shows that SalientVR achieves an up to 7.39 dB improvement in gaze-driven Peak-Signal-to-Noise-Ratio (PSNR) and reduces the rebuffering ratio by up to 3.07×, compared to alternatives. The survey-based user study with 30 participants across 10 videos also demonstrates that SalientVR receives a 43.68% higher mean opinion score (MOS), which is a user rating measure used in the domain of QoE [11, 29].

In summary, our key contributions are as follows:

- We design and implement SalientVR, a holistic saliency-driven mobile 360° video streaming system tightly integrated with gaze information.
- We present a precise gaze-driven saliency judging criterion for mobile VR viewers, and two pragmatic gaze-driven, tile-level saliency acquiring methods based on cross-user similarity and deep content analysis.
- We design a lightweight saliency-aware quality adaptation algorithm with a motion-assisted online correction, which is robust to bandwidth vagaries and saliency bias.
- Through extensive prototype experiments and real-world user studies, we demonstrate that SalientVR achieves significant QoE improvements over current approaches.
- We constructed a public gaze-annotated 360° video dataset and a gaze-driven 360° VQA metric, which would benefit the research community in precise quality assessment and attention behavior analysis.

[1] Available at https://github.com/salientVR/gazedata

2 BACKGROUND AND MOTIVATION

2.1 Mobile 360° Video Streaming

Untethered mobile VR headsets enable viewers to enjoy mobile 360° video streaming without wired hindrance, but suffer from insufficient wireless network bandwidth. Inspired by the fact that human view is restricted, past works mainly adopt the HMT-driven approaches [10, 12–15, 30]. They split each video chunk spatially into multiple tiles (e.g., 4×6 grids), and encode each tile with different quality levels, such as quantization parameters (QP). During the playback, they constantly predict the future HMT in the next 1-3 seconds according to the recently historical HMT using linear regression (LR)-based [10] or DNN-based [30] methods. Based on the predicted HMT, they assess the *importance* of each tile of the next chunk and assign the corresponding quality.

However, these HMT-driven solutions overly rely on the temporal correlation of HMT, leading to two shortcomings. First, the future HMT is scarcely correlated with the historical HMT in some cases even with a short prediction horizon (a.k.a, prediction window or *pw*). Figure 1 shows a case of LR-based HMT predictions with different *pw* based on historical head directions. The sudden brandishing swiftly draws the viewer's attention and therefore causes the invalid HMT predictions, which would severely degrade the QoE in HMT-driven approaches. For mobile VR viewers, the untethered freedom further reduces this temporal correlation.

Second, the HMT-driven approaches are highly sensitive to the size of *pw* due to rapid correlation decrease as the time interval increases. Even three seconds of *pw* degrades the HMT prediction accuracy into 35.2% [10]. Due to the prefetching feature of on-demand video streaming, *pw* has to exceed the buffer occupancy during the playback. Therefore, these approaches set a very small buffer size to shorten *pw* in practice, but suffer from frequent rebuffering especially over variable wireless network conditions. From Figure 2, when the bandwidth encounters an unexpected decline, the rapid consumption of buffer occupancy causes rebuffering for Flare (a typical HMT-driven approach) [10] that reluctantly adopts a small buffer size for more accurate HMT predictions.

2.2 Saliency and Gaze Information

The design of SalientVR is inspired by the fact that attention behaviors of humans are remarkably correlated with the saliency of pixels [16, 17, 31]. The gaze-driven saliency introduces a reasonable method of accurate long-term attention estimations with the potential of further QoE improvement for mobile 360° video

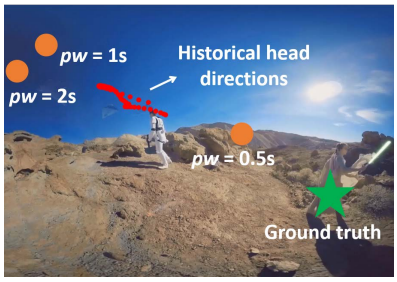


Figure 1: LR-based HMT predictions (depicted by orange circles) with different pw.

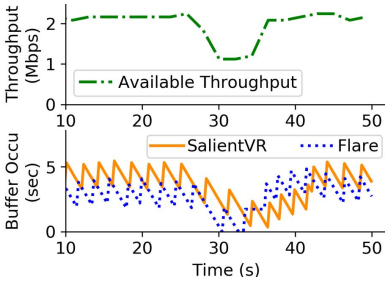


Figure 2: Network throughput and the corresponding buffer occupancy of SalientVR and Flare.

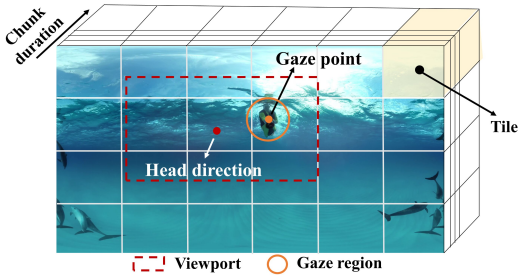


Figure 3: Viewport, gaze region, chunk, and tiles (4×6 segmentation).

streaming. Therefore, we present SalientVR that addresses three key challenges.

Challenge 1: How to precisely judge saliency's groundtruth for mobile VR viewers? Naturally, the viewing data of users could be deemed a groundtruth proxy of saliency. Along the HMT-driven line, the entire viewport (~110° in width and ~90° in height) is considered as the high-salient region. However, it is imprecise because only a small percentage region (i.e., gaze region) of the viewport is viewed with high visual acuity [20, 21], described in Figure 3. Imprecise saliency judgment would miss the opportunity of more aggressive quality allocation strategies to further improve viewer QoE. For instance, we can enhance the perceived quality by raising the video quality level in the gaze region, while simultaneously lowering the quality level in the region of viewport except the gaze region. For more precise saliency judgment, the exact scope of gaze region and saliency degrees from gaze region to viewport should be fundamentally determined for mobile VR viewers.

Challenge 2: How to pragmatically acquire accurate tile-level saliency maps? Though saliency object detection (SOD) is not new in computer vision community [24–26, 32, 33], existing efforts fail to achieve accurate acquirement of tile-level saliency maps required by 360° video streaming with specific demands. Inaccurate saliency acquirement would severely impair the advantages of saliency-based solutions. Gaze information provides a more precise and fine-grained saliency proxy than head directions. However, it is non-trivial to take advantage of gaze information to achieve accurate tile-level saliency analysis. Moreover, enough gaze/head data of users could hardly be collected in some cases especially for fresh videos that just came out or privacy-sensitive videos. More pragmatical saliency acquiring methods should be proposed to improve both accuracy and practicality.

Challenge 3: How to robustly adapt the quality of mobile 360° video streaming using saliency? Saliency-based quality adaptation has potential for QoE improvement; however, it is challenging to achieve an excellent tradeoff between quality and rebuffering, especially under the dynamic network bandwidth. Besides, compared to chunk-level adaptation, the search space of tile-level adaptation is far larger (e.g., 5^{24} combos in one chunk for 4×6 tiling and five quality levels). It would incur unbearable calculation delay for exhaustive search due to limited computing resources on mobile devices. Moreover, the saliency maps are prepared offline and are determined by collective preferences without considering individual users. Inevitable saliency bias may result in poor

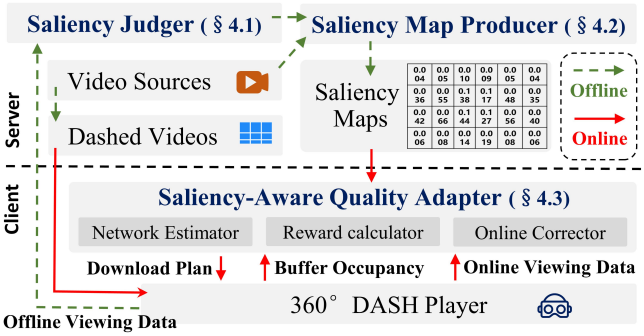


Figure 4: Overview of SalientVR.

experiences for few individual users with distinctive attention preferences, i.e., outlier viewers. For instance, a viewer that prefers to gaze at collectively-determined non-salient regions would perceive weak QoE due to the quality adaptation using saliency alone (the less salient the lower allocated quality). Therefore, a saliency-aware quality adaptation algorithm with proper calculation acceleration and bias correction is crucial for robustness to bandwidth vagaries and saliency bias.

Gaze-annotated dataset and gaze-driven VQA. Most of previous works [10, 34] adopt the viewport-driven VQA that computes the quality (e.g., PSNR) of viewport uniformly. However, the visual acuity of humans is unevenly distributed inside the viewport. The imprecise viewport-driven VQA would lead to imprecise quantified validation of video streaming algorithms. Gaze information supplements extra attention annotations for precise VQA [23]. The latest efforts [23, 27, 28] release several gaze datasets, however, whose videos are too short (< 1 minute) to evaluate the effects of long-term network variations and behavioral changes. Gaze-driven VQA has been studied for conventional videos [35], whereas it is still unexplored for mobile VR viewers.

3 OVERVIEW OF SALIENTVR

Figure 4 illustrates an overview of SalientVR that includes the offline saliency map preparation phase and the online saliency-aware video streaming phase.

Saliency map preparation. When a 360° video is uploaded, the media server transcodes and dashifies the video into multiple-bitrate chunks, and then splits each chunk into $M \times N$ equal-sized tiles, on top of the DASH standard. Next, the *saliency map* (i.e., the

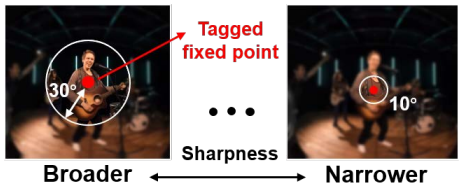


Figure 5: Different sharpness distributions to determine the precise scope of the gaze region.

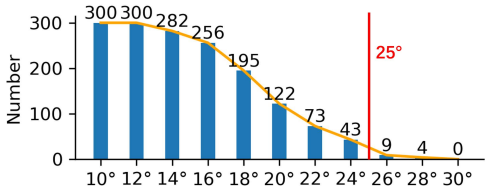


Figure 6: The number of groups for each scope where the participant perceived the quality degradation.

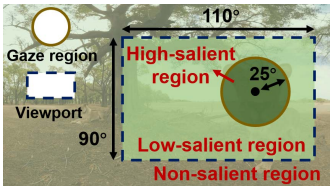
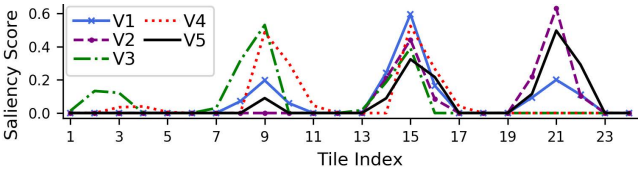


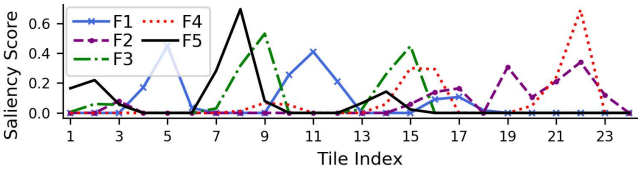
Figure 7: The gaze-driven saliency judging criterion for mobile VR viewers.

$M \times N$ numerical matrix) of each chunk is generated offline on the server by the *saliency judger* (§4.1) and the *saliency map producer* (§4.2) collaboratively, based on video content and collected viewing data. Finally, the saliency maps are packaged in *JSON* format for ease of transmission and deployment.

Saliency-aware video streaming. At the beginning of video streaming, the corresponding saliency maps are downloaded to the client player firstly. During the playback, referring to saliency maps, the saliency-aware *quality adapter* (§4.3) computes the QoE reward combining the throughput estimation and the buffer occupancy to decide the quality allocation (i.e., download plan) of streaming. Simultaneously, the *online corrector* (§4.3) utilizes the online viewing data to correct the unfrequent but improper quality allocations caused by saliency bias. According to the download plan, the videos with corresponding quality levels are downloaded, decoded, projected, and displayed by the client player.



(a) Saliency scores across different viewers (V) for the same frame.



(b) Saliency scores across different frames (F) for the same viewer.

Figure 8: Saliency variations across viewers and frames.

4 SALIENTVR DESIGN

4.1 Gaze-Driven Saliency Judgment

To precisely judge the groundtruth of tile-level saliency maps, we firstly ran a user study to determine the scope of gaze region with 30 participants and 10 360° videos. Based on the gaze region, we utilize the viewing data of mobile VR viewers to build a precise gaze-driven saliency judging criterion, which will be applied for saliency map acquirement (§4.2) and video quality assessment (§5).

Videos and participants. The 360° videos involve various genres (dance, diving, scenery, etc.) and diverse scene complexity from the gaze-annotated 360° video dataset, Gaze18 [23]. The participants, aging from 20 to 51, are students, staff, and faculty members from two universities and a company. 47% of them are female, 57% of them wear glasses, and 63% of them are first-time viewers for 360° videos. Note that there are three different user studies (§4.1, §5, and §7) and a user data collection (§5) in this work. The users' demographics are similar as illustrated here although the participants are diverse for research validity. All user studies in our work were IRB approved without raising any ethical issues.

A user study to determine the scope of gaze region. We masked videos in different *sharpness distributions* as depicted in Figure 5. The participants wore the untethered mobile VR headsets [2] and watched the same video from *broad* sharpness to *narrow* sharpness in order. The broader sharpness means the wider scope (measured in radius angles) of the circular region with high definition. The participants were asked to gaze at the *tagged fixed point* during the viewing to keep the same gaze trajectory for the same video. For each scope, we counted the number of groups (30 participants × 10 videos = 300 groups in total) where the participant perceived the quality degradation at this scope, shown in Figure 6. A person that perceived the quality degradation at a certain scope would perceive the degradation at smaller scopes for the same video. *From this study, we treat the 25°-radius gaze-centric circular region as the gaze region.* Under this setting, 95.7% of groups did not perceive the quality degradation.

Gaze-driven judgment for saliency map groundtruth. Based on the gaze region that is the precise spatial scale with high visual acuity, the saliency of a frame given the viewing data of a viewer is judged as follows (shown in Figure 7):

- **High-salient region:** the gaze region.
- **Low-salient region:** the viewport except the gaze region.
- **Non-salient region:** the frame except the viewport.

Specifically, given a pair of viewing data including a gaze point and a head direction for a viewer and a frame, we set the saliency score for each pixel (spatial dimension), this viewer (user dimension), and this frame (temporal dimension), as one in the high-salient region, ϵ in the low-salient region, and zero in the non-salient region, where $\epsilon \in (0, 1)$. Then, we average the pixel-level saliency scores in each tile for a viewer and a frame to get the tile-level saliency map, which is regarded as the groundtruth of the tile-level saliency map for this viewer and this frame.

Saliency variations across viewers and frames. Figure 8 shows the saliency score of each tile in the tile-level saliency map (4 × 6 segmentation) across different mobile VR viewers and different 360° video frames respectively. We observe that the saliency distribution of tiles widely varies across different frames for the same viewer but is similar across different viewers for the same frame. It suggests

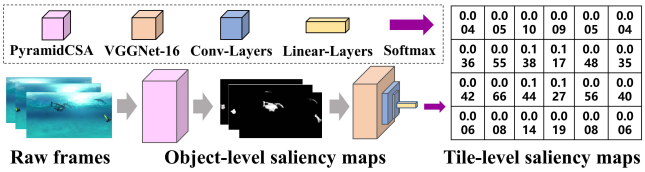


Figure 9: TSMNet structure (Dashed box: legends).

the cross-user attention similarity and the effects of video content on viewer attention.

4.2 Saliency Map Acquirement

Based on the gaze-driven saliency judgment, we pragmatically design two saliency acquiring methods (i.e., the similarity-driven and content-driven methods) to accurately generate the 360° video-specific, tile-level saliency map for each chunk.

Similarity-driven method. Based on cross-user similarity, we utilize the collected viewing data of users to yield saliency maps. First, we get the tile-level saliency map for each viewer and each frame using the gaze-driven saliency judgment (§4.1). Then, we average the tile-level saliency scores *across different viewers* to acquire the tile-level saliency map for each frame. Finally, the tile-level saliency maps of frames *in the same chunk* are merged to generate the normalized tile-level saliency map for each chunk.

Two issues need to be specifically addressed due to the equirectangular projection [36, 37]. First, the horizontal cyclicity should be carefully handled. For instance, if the gaze point is in the leftmost edge in a planar frame, the rightmost part of the gaze region should be noticed. Second, pixel scales of the viewport and the gaze region should adapt to the latitude coordinate (the higher latitude, the larger pixel scale) due to the projection distortion [38, 39].

Content-driven method. Based on the effects of video content on viewer's attention, we design a 360° video-specific content-aware DNN, named TSMNet (Tile-level Saliency Map NETwork), to generate saliency maps.

The structure of TSMNet, as depicted in Figure 9, mainly includes the modules of PyramidCSA [32] and VGG Net [40]. *The TSMNet takes multiple consecutive raw frames together as input and outputs the tile-level saliency maps.* It firstly captures the object-level saliency information (i.e., binary images) by the PyramidCSA module. PyramidCSA, a state-of-the-art saliency object detection architecture, groups a set of constrained self-attention operations in a pyramid structure and can effectively perform the object-level saliency judgment with a low computation overhead. The widely-applied VGGNet-16 architecture follows to transform the object-level saliency information to the tile-level saliency information. Then, two convolution layers with small filters (i.e., 3×3 and 1×1 respectively) and a fully connected layer are added for learning panorama features of 360° videos and merging the features into the final output.

Based on model weights pre-trained on large-scale non-360° video datasets [41], we fine-tune[2] the TSMNet model on the dataset we build (will be described in §5) and the Gaze18 dataset [23]. Both are 360° video datasets with gaze labels collected by the gaze tracker in real immersive VR settings. We use 186 videos with almost 340,000 frames as the training data and other 45 videos with almost 85,000 frames as the test data. *The viewing data is not inputted into TSMNet but is used to generate the groundtruth labels (i.e., tile-level saliency maps) via the gaze-driven saliency judgment (§4.1).* We use a mean absolute error (MAE) loss function in training, expressed by $L_{MAE}(P,G) = \frac{1}{M \times N} \sum_{i=1}^{M \times N} |p_i - g_i|$, where P and G are the predicted and groundtruth saliency maps respectively, and $(p_i, g_i) \in (P, G)$. We apply an Adam optimizer [43] with the batch size 15 and the decaying learning rate from an initial value 10^{-6}. Data parallelism and multi-GPU support are used to speed up DNN training.

Comparison with alternative content-driven DNNs. We try other alternative DNN structures for acquiring saliency maps, as described in Table 1. These DNN structures are all content-driven solutions without requiring any viewing data as input, where SALI360 is a gaze-driven but object-level method, and FPN is a tile-level but head-driven method. Table 1 shows that TSMNet achieves a lower MAE compared to DNN-based alternatives. While we adopt the TSMNet and specific settings, the SalientVR design is not bound to any specific DNNs but benefits from their evolution. We will further compare the similarity-driven and content-driven methods, and evaluate the effects of fine-tuning and gaze information, with respect to *end-to-end QoE improvement*, in §7.4.

Algorithm	Description	MAE
Naive	a heuristic simple DNN method	0.26
SALI360 [18]	a gaze-driven DNN method	0.22
FPN [44]	a tile-level DNN method	0.20
TSMNet-WF	TSMNet w/o fine-tuning	0.19
TSMNet	our content-driven method	**0.15**
Similarity	our similarity-driven method	**0.12**

Table 1: Descriptions and mean absolute errors (MAE) of different saliency acquiring methods. All methods except Similarity are content-driven DNN structures.

Pragmatical adoption. The similarity-driven method (Similarity) gains better performances than the content-driven method (i.e., TSMNet) and requires far fewer computing resources obviously. However, enough viewing data are necessary for the Similarity but could hardly be collected in some cases. By contrast, the TSMNet detects saliency information from video content alone, leading to less dependence on user data. Thus, for pragmatism, two methods are adopted in SalientVR for different circumstances.

4.3 Saliency-Aware Quality Adaptation

Based on saliency maps, we design a DASH-compatible, tile-level quality adaptation algorithm and achieve a significant computation acceleration. We also design a motion-assisted online correcting mechanism by combining offline saliency information and online viewing data to correct saliency bias.

Saliency-driven quality adaptation as an optimization problem. The saliency map implies the probability of gazing at each tile for a new user. Thus, the tiles with larger saliency scores tend to be delivered in higher quality. We also set a minimum buffer threshold, γ, to reduce the risk of rebuffering. Specifically, we formulate the quality adaptation for the kth chunk as the following

[2]Fine-tuning [42] is a popular DNN training technique for further performance improvement on specific tasks.

maximization problem:

$$\max_{l_k(j)} \quad reward_k = Q_k - \alpha\, DC_k - \beta\, DT_k$$
$$s.t. \sum_j size_k(j) \le (buffer\_occupa - \gamma) \times bandwidth. \quad (1)$$

Q_k is the saliency-weighted average quality level of Chunk k and defined as the following:

$$Q_k = \sum_j Saliency_k(j) \times l_k(j),$$

where $Saliency_k(j)$ is the jth saliency score of the saliency map (acquired in §4.2) of Chunk k, and $l_k(j)$ is the video quality level (e.g., the function of qp) of Tile j of Chunk k.

DC_k and DT_k represent the Chunk k's saliency-weighted average quality differences between two consecutive chunks, and between neighboring tiles in an identical chunk, respectively, as defined in the following:

$$DC_k = \sum_j Saliency_k(j) \times Saliency_{k-1}(j) \times |l_k(j) - l_{k-1}(j)|$$

$$DT_k = \sum_j \sum_{r \in nei(j)} |l_k(j) - l_k(r)| \times Saliency_k(j)/|nei(j)|.$$

To facilitate computation, we only consider quality differences of *same-position* tiles of two successive chunks in DC_k. Admittedly, quality differences of *diverse-position* tiles may influence the QoE due to the cross-tile gaze movement in the junction of two successive chunks. In practice, computing the diverse-position differences in DC_k gains little but increases the computation complexity by tens of times.

Our saliency-driven quality adaptation design dramatically reduces the reliance on HMT predictions by viewport-free reward computation. Therefore, SalientVR enables the client player to keep a longish buffer to absorb network variations without worrying about the diminution in HMT's temporal correlation. Inspired by [8], we set a minimum buffer threshold, γ, which means that the buffer occupancy should always exceed γ during video downloading. Specifically, a chunk size constraint is added in Eq.1, where $size_k(j)$ is the file size of Tile j of Chunk k. If there is no quality combo satisfying the size constraint, SalientVR will adopt the most conservative quality adaptation scheme, i.e., fetching chunks in the lowest quality levels, to expand buffer as best it can.

Computation speed-up for quality adaptation. Huge search space of tile-level adaptation would incur the considerable calculation delay for exhaustive search, which would severely impair adaptation performances. Thus, we adopt two techniques to reduce the search space and speed up computation. First, a constraint is added that the quality level of each tile never increases as the saliency score drops off. Second, the simulated annealing algorithm [45, 46], a probabilistic technique for approximating the global optimum of a given function, is delicately applied. After using the above two techniques, SalientVR only needs to compute hundreds of paces to find the near-optimal solution (i.e., the quality allocation with the almost maximal reward). Table 2 lists the average time consumption and optimal reward of the exhaustive search scheme and our speed-up methods.

Motion-assisted online correction. To improve the QoE of outlier viewers with distinctive preferences, we design a motion-assisted online correcting mechanism combining offline saliency information and online viewing data to correct the unfrequent but improper quality allocations caused by saliency bias. During the playback, SalientVR constantly tracks the online head movement data of viewers. If the overlapping rate of the actual viewport and the top-four tiles (ranked in order of saliency scores; for 4×6 tile segmentation) is less than 25% for a 3-second duration, the *quality adapter* will enter the *correction mode* until the overlapping rate exceeds 50% for the same duration.

	Time cost	Optimal reward
Exhaustive search	3.48s	0.045
Our speed-up	0.04s	0.044

Table 2: Our techniques achieve an 87× computation speed-up to find the near-optimal solution of quality adaptation, compared to the exhaustive search.

Under the correction mode, SalientVR performs the online viewport prediction using the motion-based method [10] while running the saliency-driven adaptation. The saliency-driven adaptation decides the quality allocations in the distant future based on offline saliency information and therefore enables the client player to keep a longish buffer occupancy. Simultaneously, the online corrector performs the short-term viewport prediction based on online viewing data to compensate the video tiles with improper quality levels in the buffer. Specifically, if the buffer occupancy exceeds 3γ, the low-quality buffer-in tiles inside the predicted viewport would be redownloaded and replaced with high-quality tiles. The redownloaded tiles are useless if they fail to arrive before the time they are displayed. Hence, SalientVR carries out tile correction only if the subtraction of estimated redownloading time from the buffer occupancy exceeds γ. Honestly, redownloading wastes a few network resources but still achieves respectable benefits (will be shown in §7.4). Hopefully, the scalable video coding (SVC) [47] could enable incremental downloading to reduce the overhead of correction, although it is not the point in this work.

5 DATASET AND QUALITY ASSESSMENT

For more solid evaluations, we constructed a gaze-annotated long 360° video dataset and a gaze-driven 360° video quality assessment (VQA) metric for mobile VR viewers.

Gaze data collection. We downloaded 23 long 360° videos (2-11 minutes, 11 genres, 4K resolution, 30-60 fps, and ~2 hours in total) from YouTube according to the popularity ranking. The genres include aerial, animal, cartoon, dance, diving, game, mixed, racing, roller coaster, scenery, and outer space. Then, 30 participants viewed these videos using the HTC VIVE PRO EYE VR headset with built-in Tobii's gaze tracking [48]. The gaze data of each participant were recorded by the Tobii Pro Lab software [49]. *VIVE Wireless Adapter [50] enabled viewers to explore freely without wired hindrance.* To alleviate fatigue, a short break was required every 10-minute watching. Before collection, every participant was required to watch a *warm-up video* to allay nervousness and a *calibration video* to calibrate gaze coordinates successively. Any participant that felt dizzy or other discomfort discontinued watching and the corresponding data were discarded. The participants experienced

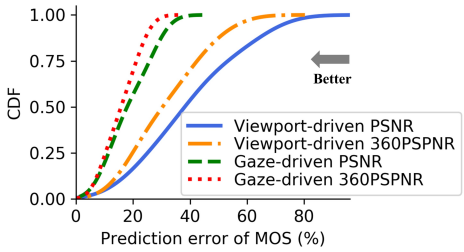


Figure 10: Cumulative distribution of the MOS prediction error. It demonstrates that the gaze-driven VQA metrics achieve more accurate MOS predictions.

the uniformly high video quality (without rebuffering) to eliminate the effect of spatially-uneven quality distribution on viewing behaviors (will be discussed in §8).

Gaze-driven 360° VQA metric. The gaze region occupies more visual acuity than the hollow viewport (the viewport except the gaze region). Therefore, based on the determination of gaze region (§4.1), we build a gaze-driven 360° VQA metric:

$$FrameQua = \frac{q(GazeRegion) + \epsilon \cdot q(HollowViewport)}{1 + \epsilon} \quad (2)$$

$$VideoQua = \sum_i FrameQua(i)/FrameNum, \quad (3)$$

where $q(X)$ is the image quality (e.g., PSNR [51]) in Region X of a frame, $FrameQua(i)$ is the weighted average quality of Frame i, and $\epsilon \in (0, 1)$ is the weight coefficient.

The VQA metric is built for more precise evaluations and is different from the reward metrics (e.g., Q_k) designed for quality adaptation (§4.3). Moreover, the gaze-driven VQA mainly focuses on the more precise perceptible scope with high visual acuity over gaze data and is combinative with conventional VQA criteria such as PSNR and 360PSPNR [11]. For instance, the gaze-driven PSNR denotes the $VideoQua$ where the $q(X)$ is specified by the PSNR value in Region X.

Validation of usefulness. We conducted a user study on 10 360° videos and 15 participants to verify the usefulness of gaze-driven VQA. The videos were encoded with uneven quality distributions, viewed on VR headsets, and rated by viewers. We combined the viewport-driven and gaze-driven VQA with PSNR and 360PSPNR to compute the objective quality. We then checked correlations between the combined VQA metrics and MOS (i.e., mean opinion score). Similarly as [11], we built four linear MOS predictors based on the different VQA metrics. Figure 10 shows the distributions of prediction errors (i.e., $|MOS_{predict} - MOS_{real}|/MOS_{real}$). Compared to the viewport-driven VQA, the gaze-driven VQA metrics achieve more accurate MOS predictions, which validates that the gaze-driven VQA metrics correlate with MOS more closely.

6 IMPLEMENTATION

We implemented a prototype of SalientVR, client running on a Mini PC, based on 8300 additional lines of JavaScript and Python code. The Mini PC, Dell OptiPlex 3080MFF, is equipped with a mobile-class Intel UHD Graphics 610 GPU, an Intel Pentium G6405T 1.8 GHz dual-core processor, 4 GB of RAM, and a 128 GB SSD. It is equivalent to a common commodity smartphone in computing resources. For instance, the Samsung Galaxy S20 phone [52] is equipped with a Qualcomm Adreno 650 GPU, a Qualcomm Snapdragon 865 1.8-2.84 GHz multi-core processor, 8 GB of RAM, and a 128 GB SSD, which is similar to our Mini PC settings. Thus, we elaborately chose the Mini PC to emulate the computing resources of a mobile VR device. A similar prototype implementation scheme was adopted by the previous work [53].

We used FFmpeg [54] and Kvazaar [55] to encode videos in High Efficiency Video Coding (HEVC) formats due to native support for tiled coding [56, 57], and used MP4Box [58] to package and dashify HEVC bitstreams. The video server was a web server over HTTP built in Ubuntu 18.04. We constructed an HTML5-based, DASH-compatible VR Player to stream and display HEVC-tiled 360° videos on the Microsoft Edge browser on top of the HEVC Tiles Merger [59]. The Player fetches and merges split tiles with different qualities according to a controllable *quality matrix*. To facilitate online correction, we set a JavaScript *Array* before the buffer queue in the Player. The downloaded video files are placed in the Array firstly and are deferred to enter the buffer queue. The videos in the Array maintain the independence of tiles without merging or decoding, which means that any tile replacement in the Array does not impact other tiles. The DNN structures in this work were constructed using the Pytorch repository.

For user studies on untethered mobile VR headsets, we used *A-Frame* [60], a web framework for building VR experiences, to render and project 360° video streams from the PC Player into the mobile VR headset (i.e., Oculus Quest 2 [2]). Simultaneously, we acquired the online viewing data by *VRFrameData.pose* in *A-Frame* and sent the viewing data to the online corrector. Due to support issues, we used *HTML5 Canvas* to capture the content in *video tag* in the PC Edge browser (supports HEVC but not A-Frame), and transmitted the content to the onboard browser in Oculus Quest 2 (supports A-Frame but not HEVC). The two browsers were on the same WiFi network (802.11ac at 5 GHz) with an imperceptible transmission delay.

We adopted the PC-based implementation approach for our client player, largely due to abundant support of the well-developed, compatible software stack and open-source tools on the Windows platform, e.g., browser support for Media Source Extensions (MSE) API and hardware HEVC decoding. Our implemented prototype properly emulates the computing resources of mobile VR devices, and fully achieves the free, untethered, immersive, and interactive viewing experience of 360° video streaming with a commodity mobile VR headset. Therefore, we believe that the experiments through our implementation are valid enough to demonstrate the advantages of SalientVR. That being said, lack of an implementation totally built on commodity mobile devices adds a complication to the practical deployment and application in industry.

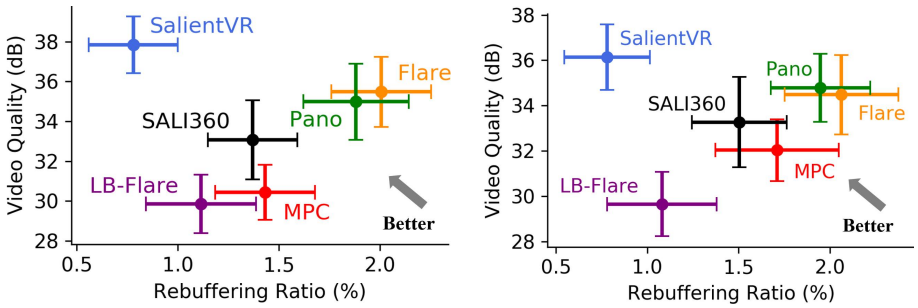
7 EVALUATION

7.1 Experimental Setup

Videos. We used totally four-hour gaze-annotated 360° videos in our evaluations, including 80 short videos (20-60 seconds) in the Gaze18 dataset [23] and 23 long videos (2-11 minutes) in our dataset (§5). These videos were viewed with commodity VR headsets by 45 participants in total. The videos exhibit a large diversity in terms of genres, including aerial, documentation, scenery, *etc.* There are both videos shotted from fixed cameras and videos captured with a moving camera, which probably introduces different variances

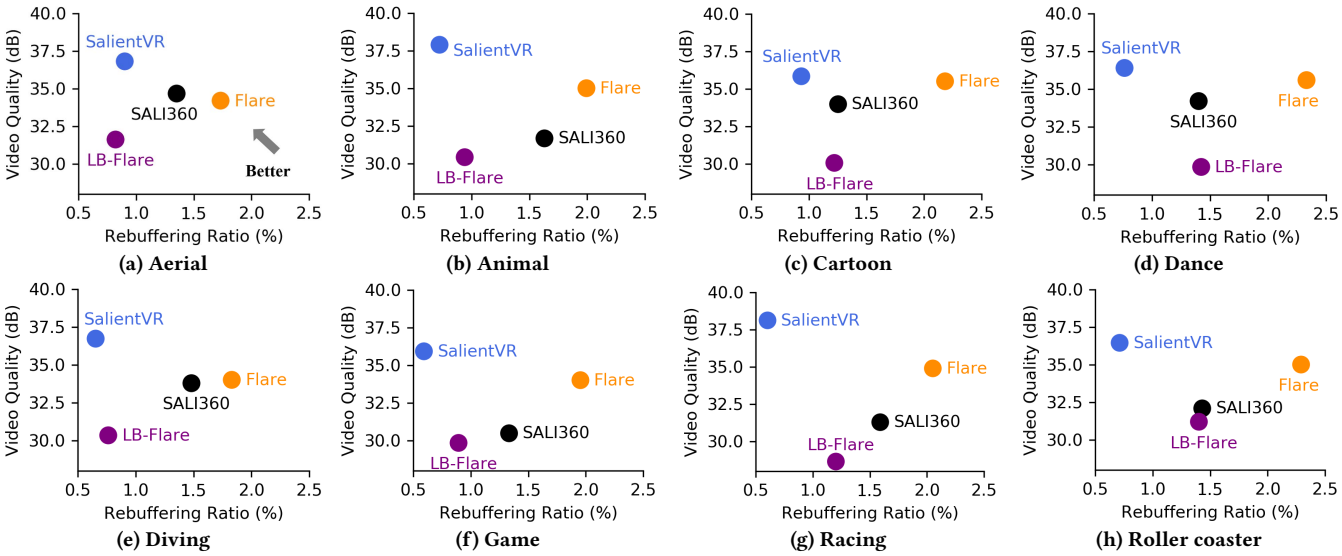
ALGO	Description
Flare	using HMT-driven viewport predictions
Pano	using HMT-driven 360° quality models
SALI360	a method using gaze and saliency
MPC	a method designed for non-tiled videos
LB-Flare	a variant of Flare with a long buffer

Table 3: A summary of algorithms for comparison.



(a) Gaze18 dataset [23]. (b) Gaze-annotated dataset we build (§5).

Figure 11: Overall results in video quality and rebuffering ratio by gaze-annotated dataset-based evaluations (both Gaze18 dataset and our dataset) over the LTE network emulation. (Error bars show 95% confidence intervals.)



(a) Aerial (b) Animal (c) Cartoon (d) Dance
(e) Diving (f) Game (g) Racing (h) Roller coaster

Figure 12: Separate results in video quality and rebuffering ratio across eight typical video genres by dataset-based evaluations over the LTE network emulation.

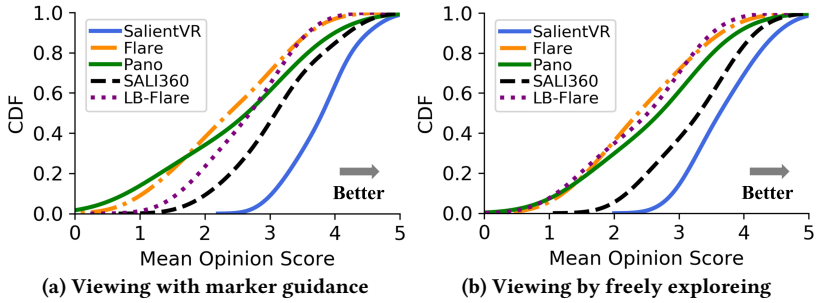
in head/gaze trajectories of viewers. Further, some videos possess relatively more complex content scenes (e.g., numerous foreground objects in a frame) while the scenes in some videos are simple, which would influence the difficulty of saliency analysis.

Each video was encoded into 2.13-second chunks with 4×6 tiling and five quality levels (QP=22,27,32,37,42). The videos were randomly divided into two parts to be adopted two saliency acquiring methods (§4.2), respectively. For each video with the similarity-driven saliency acquiring method, we randomly selected the viewing data of 75% users for offline saliency acquirement (also used for TSMNet training) and used the other viewing data for streaming system evaluations. As for the content-driven saliency acquirement, we only randomly selected 25% viewing data for evaluations due to the assumption of viewing data unavailability (§4.2).

Network conditions. We picked 10 real-world throughput traces from the Belgium 4G/LTE dataset [61], replayed by the Mahimahi network emulation tool [62]. The traces were captured using 4G/LTE-connected smartphones over multiple transportation types including foot, bus, train, *etc.* We also used five remote cloud servers on different cities as video servers through a residential commodity WiFi link to evaluate systems in the wild.

Algorithms for comparison. We compared SalientVR with Flare [10], Pano [11], SALI360 [18], MPC [63] and long-buffered Flare (LB-Flare), as summarized in Table 3. Flare and Pano are typical HMT-driven methods. SALI360 notices the potential of gaze-driven saliency in 360° video streaming. However, it mainly addresses the issues about cube map encoding and does not really discuss the tile-level streaming adaptation, which is the major concern in SalientVR. MPC is a control-theoretic adaptation approach designed for non-tiled video streaming. LB-Flare, a variant of Flare, expands the buffer size from three seconds to five seconds and tends to keep a longer buffer than Flare.

Metrics and settings. We mainly used two objective QoE metrics, gaze-driven PSNR (§5), and the rebuffering ratio (i.e., the ratio between the total rebuffering duration and total watching time). MOS was adopted as the subjective QoE metric. Video quality variations, viewport-driven PSNR, and structural-similarity-index-measure (SSIM) [64] were also discussed. We empirically set α =



(a) Viewing with marker guidance (b) Viewing by freely exploreing

Figure 13: Overall improvement in real user rating (i.e., MOS) by the survey-based user study with two kinds of viewing methods (i.e., marker guidance and freely exploring) over the LTE network emulation.

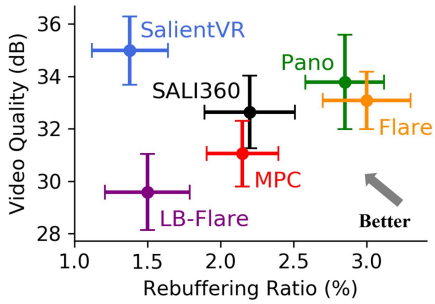


Figure 14: SalientVR's improvement in video quality and rebuffering ratio in the wild over residential WiFi.

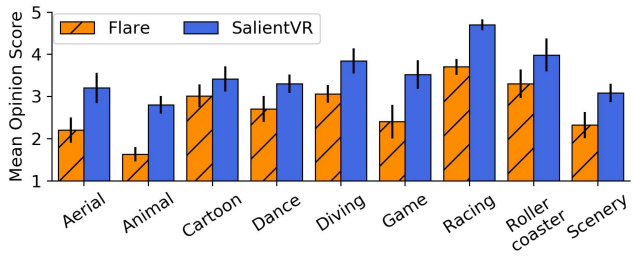


Figure 15: Separate MOS results of SalientVR compared to Flare across nine typical video genres.

0.1, $\beta = 0.5$, $\gamma = 2.5$, and $\epsilon = 0.3$ in SalientVR, which is independent of network trace selections. For other algorithms, we used the same parameter settings as their papers. For fairness, all algorithms used the same throughput estimation method [63].

Survey-based user study. 30 participants viewed 10 videos by mobile VR headsets [2] to evaluate the subjective QoE performances of different algorithms. After watching a video, the viewer was required to rate the experience between 1~5. We adopted two kinds of viewing methods. In the first group, we picked real gaze trajectories and marked the gaze point on each video frame. The viewers were asked to gaze at the markers to ensure the same viewing trajectory for the same video across different algorithms and viewers. This method of user study was widely adopted by previous works [11] although it does not provide actual experiences as viewers freely explore. In the second group, participants were able to freely explore the video content without any marker guidance. The random order and the interval between watching were required to reduce the influences of repeated watching.

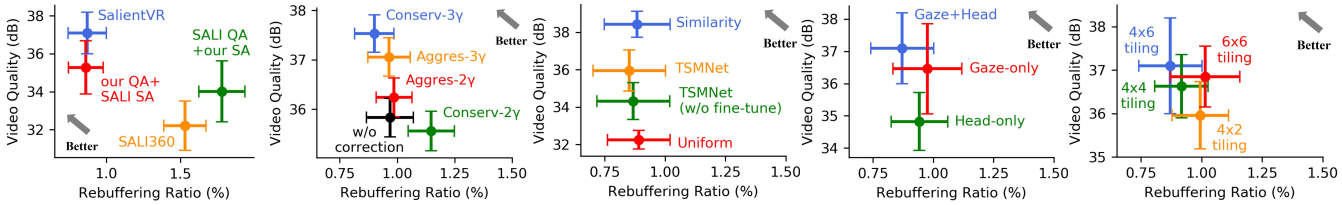
7.2 Improvement over LTE Emulations

Dataset-based evaluation. The collected viewing data are serially ingested by the client player to simulate the viewing process. As shown in Figure 11, overall, SalientVR achieves a 1.36-7.39 dB video quality improvement and reduces the rebuffering ratio by 1.64-3.07×, compared to existing approaches. Also, Figure 12 depicts the separate evaluation results across eight typical video genres to demonstrate the generalizability of SalientVR gains across genres.

For Flare and Pano, the LR-based HMT predictions are not accurate enough to support the video delivery with high perceived quality even keeping a short *pw*. Moreover, we found that the accuracy of HMT predictions varies with the content complexity, leading to different qualities across videos for the HMT-driven approaches. For some genres of videos like cartoon, the number of foreground objects is relatively low and the objects are relatively motionless for most clips. The accuracy of HMT predictions for these *non-complex* videos is higher than that for videos with complex contents, therefore, with higher qualities for Flare. In terms of rebuffering, the small buffer size in the HMT-driven solutions brings more rebuffering under the fluctuant wireless network bandwidth. In contrast, SalientVR performs better by virtue of more accurate attention estimations and a larger buffer size to absorb network variations. LB-Flare attempts to reduce rebuffering by simply enlarging the buffer size, but suffers from the distinct quality degradation due to the ineffective HMT prediction.

SALI360 utilizes saliency to achieve a better tradeoff between quality and rebuffering compared to HMT-driven methods, but is still far from tapping the full potential of saliency, resulting in inferior QoE performances over SalientVR. First, the object-level saliency detection (SD) in SALI360 is less precise than the tile-level SD in SalientVR, which removes some of benefits of saliency-based solutions. Second, SALI360 suffers from poor QoE performances in under-exposed video scenes (e.g., the genre of outer space) due to the luminance-sensitive SD approach used in SALI360. SalientVR avoids the above SD issues in dim scenes by luminance-free SD approaches. Third, the image-based SD method adopted in SALI360 experiences severe SD errors in some 360° video-unique background-centric scenes. For instance, for the genre of racing, most volunteers would like to gaze at the background region (e.g., racing track) from the perspective of a single image, but the foreground object (e.g., racing driver) tends to be detected as the high-salient object for image-based SD methods. It is an inevitable limitation of image-based SD methods due to the lack of motion information. In contrast, our TSMNet model adopts the video-based SD approach, i.e., taking multiple consecutive frames as input to capture the 360° video-specific motion information. Thus, SalientVR overcomes the limitation of image-based SD methods and achieves a more accurate attention prediction. Fourth, more importantly, in comparison to SALI360, we design a more sophisticated and robust quality adaptation algorithm fully leveraging saliency information, contributing to superior QoE results.

Survey-based user study. Figure 13 compares the MOS of SalientVR and other algorithms, over the two kinds of viewing methods (§7.1).



(a) Quality adaptation. (b) Online correction. (c) Saliency acquirement. (d) Gaze information. (e) Tile segmentation.

Figure 16: Microbenchmarks: component-wise analysis for end-to-end improvement in quality and rebuffering.

The results over marker guidance and freely exploring both describe that SalientVR gains a significantly higher user rating, with a 43.68% MOS improvement on average, compared to alternatives. The user feedbacks indicate that the rebuffering is more annoying than the quality degradation for the majority of participants when experiencing mobile 360° video streaming. For SalientVR, the remarkable decrease of rebuffering significantly boosts viewer QoE, compared to the HMT-driven methods. To further explain the generalizability of SalientVR, we compare SalientVR and Flare in MOS across nine typical video genres, as shown in Figure 15, and we see that the gains of SalientVR vary across videos with different genres (will be discussed in §8).

7.3 Improvement in the Wild over WiFi

To test SalientVR in the wild, we used five remote cloud servers (alleged 15-25 Mbps download bandwidth and 20-50 ms RTT) as video servers over a residential WiFi link during different periods (e.g., peak hours and non-peak hours). The commodity WiFi AP supports 802.11ac, 2.4 GHz, and the 2,976 Mbps maximum rate. We simultaneously ran multiple client players with different algorithms and downloaded videos from the same video server to keep the network bandwidth as equivalent as possible for different algorithms. Figure 14 shows that SalientVR achieves a 1.15-5.64 dB quality improvement and reduces the rebuffering ratio by 1.33-2.14×, compared to other approaches. This validates that SalientVR manages to make a better tradeoff between the video quality and the rebuffering over dynamic network conditions in the wild by the accurate attention estimation and the long buffer mechanism.

7.4 Microbenchmarks

In this study, we altered some key components in SalientVR one-by-one to specify the reasons of these settings and better understand their contributions to end-to-end QoE improvements for mobile 360° video streaming.

Effects of our quality adaptation algorithm. Figure 16a compares different combinations of saliency acquirement (SA) and quality adaptation (QA) methods to evaluate the effects of our QA algorithm. *SALI QA+our SA* combined the QA method used in SALI360 and the SA method designed in SalientVR. We observe that the QoE performance of SalientVR (i.e., *our QA+our SA*) is obviously better than *SALI QA+our SA*. Similarly, *our QA+SALI SA* performs better than SALI360 (i.e., *SALI QA+SALI SA*). It demonstrates the superiority of our QA algorithm *per se* even using other alternative SA approaches. Moreover, by comparing SalientVR and *our QA+SALI SA*, we see that the tile-level SA of SalientVR performs better than the more coarse-level SA of SALI360 due to the more precise and specific saliency analysis adopted by us.

Effects of online correction. To evaluate the effects of online correction (§4.3), we chose 15 real viewing trajectories of viewers that preferred to gaze at low-salient regions classified by collectively-determined saliency inference (§4.2). Figure 16b compares four different configurations of online correction. The Conserv and Aggres refer to the conservative and aggressive correction settings, i.e., entering the correction mode when the overlapping rate is less than 25% and 50%, respectively. The 2γ and 3γ refer to performing tile correction only when the buffer occupancy exceeds 2γ and 3γ, respectively. Hence, the 2γ also implies a more aggressive correction method compared to the 3γ. We see that the online correcting mechanism effectively enhances the video quality by replacing the improper quality allocations (i.e., low-quality tiles) for outlier viewers. Moreover, the relatively aggressive correction fails to achieve better performances than the relatively conservative method. For the aggressive correction, the frequent redownloading limits the enlargement of buffer occupancy, causing conservative (i.e., low-quality) quality allocations. Further, the low-quality allocations are prone to activating the correction that could have been avoided. Actually, the online correction is hardly triggered for most viewers because the saliency-driven attention estimation is accurate and effective enough for most users.

Different saliency acquiring methods. Figure 16c shows the end-to-end QoE performances of similarity-driven (i.e., the Similarity) and content-driven (i.e., the TSMNet) saliency acquiring methods (§4.2). The TSMNet (w/o fine-tune) adopted the non-finetuned TSMNet to acquire saliency maps. The Uniform set saliency scores of all tiles equal in saliency maps. We see that the Similarity achieves a larger QoE improvement than the TSMNet. Moreover, the adoption of fine-tuning training techniques enhances the accuracy of the TSMNet model and consequently improves the video quality.

Effects of gaze information. Figure 16d evaluates the effects of gaze data on SalientVR in contrast to head direction data. SalientVR used both gaze information and head information for saliency judgement and acquirement. The Gaze-only used the gaze data only, i.e., taking the non-gaze-region as non-salient region, and the Head-only used the head data only, i.e., taking the entire viewport as high-salient region (§4.1). The adoption of more precise gaze information supports the more accurate saliency map acquirement and the more aggressive quality allocation strategy, resulting in better QoE performances than Head-only. Even so, head direction information is also beneficial for QoE improvement, which is indispensable.

For online correcting (§4.3), we used online head directions to predict viewport rather than performing gaze trajectory predictions due to the lower accuracy of trajectory-driven gaze predictions. Figure 17 shows that more frequent abrupt gaze movements make

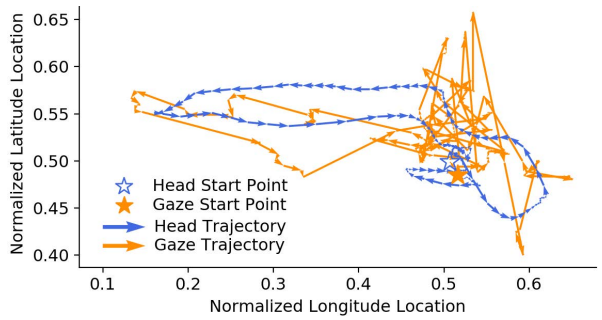


Figure 17: Simultaneous trajectory curves of head movements and gaze movements. The head trajectory is more smooth than the gaze trajectory.

gaze trajectories bumpier than head trajectories. A highly non-smooth gaze trajectory curve increases the difficulty of trajectory-driven gaze predictions and consequently degrades the prediction accuracy. Note that the saliency-driven attention estimation in SalientVR is not affected by non-smooth gaze trajectories due to not relying on the temporal correlation of gaze movements.

Different tile segmentation methods. Figure 16e compares different tile segmentation methods (i.e., 2×4, 4×4, 4×6, and 6×6) in SalientVR. The 4×6 tiling method gains the highest video quality and the lowest rebuffering ratio. A more fine-grained 6×6 tiling method increases the flexibility of quality adaptation but limits the network utilization. The transmission time of video tiles does not scale linearly with tile sizes. Smaller tile sizes lead to lower throughput rates due to the slow-start-restart behavior in the TCP protocol [66, 67]. More fine-grained tile splitting also increases the time consumption of quality adaptation due to the larger search space. Compared to the 4×6 tiling method, the 6×6 tiling method requires a 4× more solving time of the optimization problem (Eq.1). Overall, the 4×6 tiling method makes a better tradeoff among the flexibility of quality adaptation, network utilization, and the time consumption of quality adaptation.

7.5 Improvement in Other Quality Metrics

Besides gaze-driven PSNR, there are some other video quality assessment metrics that are worthy of measurement and discussion, and therefore, we further quantify the video quality on these metrics with different algorithms.

Quality variations. Figure 18 compares the temporal and spatial video quality variations (the lower the better) of different algorithms. The temporal quality variation quantifies the quality change between neighboring frames in the video, computed by $\sum_i \left|FrameQua(i) - FrameQua(i-1)\right|/num$. The spatial quality variation quantifies the quality difference across tiles in viewport, computed by $\sum_j \text{StdDev}_j\{q(Tile_i) : Tile_i \in Viewport\ of\ Frame\ j\}/num$, which is a unique metric for 360° videos due to tiling [10]. The $FrameQua$ and $q(\cdot)$ are defined in §5. SalientVR achieves lower quality variations (both temporal and spatial) than other tile-based algorithms. Moreover, SalientVR can further reduce the quality variations by tuning the penalty coefficients α and β in Eq.1, although some researchers claim that the quality variations would incur limited QoE degradation [10, 68].

Viewport-driven PSNR. We evaluated different algorithms using the conventional viewport-driven PSNR (i.e., the PSNR value of entire viewport), although it is less precise than the gaze-driven PSNR (§5). From Figure 19, SalientVR likewise achieves positive gains compared to alternatives.

7.6 System Overhead

Tiled encoding. We used a Ubuntu Server with NVIDIA TITAN X GPU to encode our videos. Figure 20 depicts the average overhead and time cost for encoding a 2.13-second video chunk with different tiling methods. We measure the overhead using the chunk size in units of the size of baseline (i.e., a 2.13-second, non-tiled, HEVC-encoded chunk). The time cost is also expressed in units of the baseline consumption for a clearer comparison. The HEVC-encoded 4×6 tiling increases the overhead by 5% only but reduces the encoding time by 10× compared to the non-tiled HEVC encoding.

Client-side overhead. We used a weak Dell Mini PC (§6) as the client. The quality adapter averagely costs 57 ms per quality allocation of one chunk, which is negligible compared to the chunk transmission time. SalientVR applies the monotonicity constraint and the simulated annealing algorithm to remarkably reduce computation complexity and time consumption of quality adaptation. The HTML5-based Player averagely consumes 16% CPU usage, 210 MB memory, and 19.4 ms duration per decoding and tile merging of one chunk.

8 LIMITATIONS AND DISCUSSIONS

Effect of quality distribution on viewer attention. We observed the effect of spatially uneven video quality distribution on viewer attention. For example, more viewers gaze at the high-quality region than the low-quality region though both regions are deemed high-salient. That is, the quality allocation scheme would unintentionally lead viewer's attention to tiles with allocated high quality. In our future work, we will study this effect which is a common issue faced by quality-uneven video streaming. Note that this effect does not impair our algorithm evaluations due to the uniformly high-quality viewing in our dataset collection (§5).

Different gains across videos. We found that SalientVR gains less for saliency-uniform 360° videos compared to saliency-uneven videos. For saliency-uniform videos, the saliency scores in different regions are almost same. Therefore, the saliency map would fail to provide meaningful prior information for estimating a new viewer's attention. Fortunately, in light of our observations and experiments, most of 360° videos are saliency-uneven to some extent and significantly benefit from SalientVR.

Separate evaluation metrics. The correlations between the quality variation and the subjective QoE metric (e.g., MOS) are unexplored, especially considering gaze information. Moreover, we separately measure the video quality, the rebuffering, and the quality variation in our evaluations, but do not quantify the relationship between different metrics and how they jointly deduce the user-perceived quality.

Advanced wireless network. More advanced wireless network (e.g., 5G) would relieve the bandwidth pressure. However, the bandwidth would be still insufficient for indiscriminately high-quality video deliveries as 360° videos with higher resolution (e.g., 8K or 16K) are emerging. Moreover, the inherent volatility of wireless

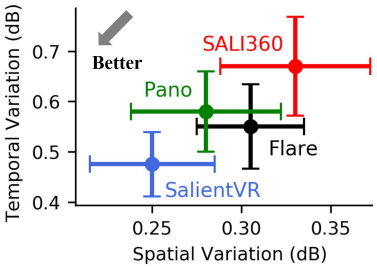


Figure 18: Temporal and spatial video quality variations of different algorithms.

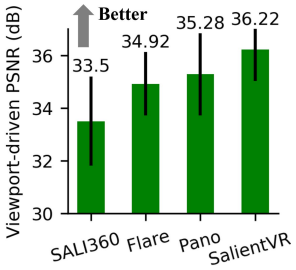


Figure 19: Viewport-driven PSNR values of different algorithms.

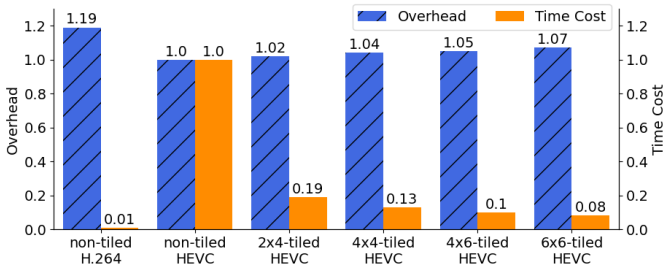


Figure 20: Overhead and time cost of different encoding methods. x264 [65], Kvazaar [55] encoders are used for H.264 and HEVC encoding, respectively.

networks has not been solved. Thus, the clash between high-quality viewing demands and inadequate bandwidth is a permanent issue.

9 RELATED WORK

360° video streaming. Most of past works are built upon LR-based [10–14] or DNN-based [23, 44, 69, 70] HMT predictions, limited by the temporal correlation (TC) assumption. Some studies explore the opportunities of cross-user similarity [70–73], scalable video coding [74–76], content-aware saliency analysis [19, 44, 70, 72, 77], and super resolution [34]. By contrast, SalientVR combines the advantages of cross-user similarity, saliency analysis, and HMT predictions, and introduces the new studies and designs for 360° video streaming by elaborately merging gaze information and saliency.

SALI360 [18] proposes a cubemap-based 360° video compression algorithm over saliency. Unlike SalientVR, it does not consider the dynamic quality adaptation problem during playback. Moreover, SALI360 focuses on image-based, encoding-oriented saliency analysis. By contrast, SalientVR presents the video-based, adaptation-oriented saliency acquiring approach. Xu *et al.* [70] and Li *et al.* [72] propose multiple viewport prediction (VP) approaches using the HMT, cross-user similarity, heatmaps, and saliency, but do not discuss how to use their VP results for quality adaptation of 360° video streaming. The accuracy of these VP methods is likewise highly sensitive to the TC of viewing data. By contrast, SalientVR designs TC-free saliency acquiring methods over gaze data and the corresponding quality adaptation algorithm. Huang *et al.* [19] design a multicast resource allocation algorithm combining saliency and viewport for 360° video streaming in wireless multi-RAT networks. By contrast, SalientVR cares for the client-side quality selection for a single user rather than the base station (BS)-side resource allocation across multiple users. Moreover, SalientVR is designed for general wireless network circumstances and is independent on wireless multi-RAT networks.

Saliency object detection. Salient object detection (SOD) aims at locating the most attention-grabbing objects from images [78–80] or videos [24–26, 32, 33]. Specially, Xu *et al.* [23] and Fan *et al.* [44] put forward different DNN-based gaze fixation prediction models for 360° videos. Unfortunately, the previous efforts cannot be used straightforwardly for tile-level saliency map acquirement that is required by quality adaptation of 360° video streaming. Therefore, the gaze-driven, tile-level saliency is considered in SalientVR, which is new from the perspective of saliency detection.

Adaptive bitrate. Despite numerous efforts on adaptive bitrate (ABR) [8, 63, 67, 81, 82], the saliency-driven quality adaptation for mobile 360° video streaming is still underexplored. Lee *et al.* [77] experiment the saliency-driven rate adaptation using a motion-constrained tile set technique. By contrast, we formulate the saliency-aware adaptation problem and achieve the significant computation acceleration. We also design a novel online correction algorithm by combining offline saliency information and online viewing data.

Gaze information. Gaze information has been used in viewport predictions [23, 44], saliency analysis [16, 18, 22, 25, 83], and video quality assessment (VQA) [35, 84], as the human attention proxy. By contrast, we investigate the gaze-driven saliency judgment and 360° VQA for mobile VR viewers, and design a new integrated 360° video streaming system based on gaze information. Recent works [23, 27, 28] contribute several gaze datasets that are built using short-duration 360° videos only. By contrast, we construct a gaze-annotated long 360° video dataset.

10 CONCLUSION

We design and implement SalientVR, a saliency-driven mobile 360° video streaming system integrated with gaze information to improve the quality of experience (QoE) under the insufficient wireless network bandwidth. SalientVR takes full advantage of gaze information to address three fundamental challenges in saliency-based mobile 360° video streaming, including precise saliency judgment, pragmatic saliency acquirement, and robust saliency-aware quality adaptation. Through extensive prototype experiments and user studies, compared to the state-of-the-art approaches, SalientVR achieves an up to 7.39 dB improvement in video quality and reduces the rebuffering ratio by up to 3.07× over wireless network conditions, which boosts viewer QoE by 43.68%. Moreover, we constructed a public gaze-annotated 360° video dataset and a gaze-driven quality assessment metric for mobile VR viewers, which are not only used in our evaluations but would also facilitate precise quality assessment and attention behavior analysis for the 360° video streaming research.

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers and shepherd for their thoughtful suggestions. This work was supported by the National Key Research and Development Program of China under Grant 2020YFA0713900; the National Natural Science Foundation of China under Grant 61772410, Grant 61802298, Grant 62172329, Grant U1811461, Grant 11690011, Grant U21A6005, Grant 62022005, and Grant 62061146001; and the China Postdoctoral Science Foundation under Grant 2020T130513 and Grant 2019M663726.

REFERENCES

[1] Grand View Research. Virtual reality market share and trends report, 2021-2028. https://www.grandviewresearch.com/industry-analysis/virtual-reality-vr-market.

[2] Oculus. Oculus quest 2 - untether your expectations. https://www.oculus.com/quest-2/.

[3] Samsung. Gear vr. https://www.samsung.com/global/galaxy/gear-vr/.

[4] YouTube. Virtual reality channel - youtube. https://www.youtube.com/channel/UCzuqhhs6NWbgTzMuM09WKDQ.

[5] VRGear. Netflix vr. https://vrgear.com/news/netflix-vr/.

[6] Internet connection speed recommendations. https://help.netflix.com/en/node/306.

[7] Verizon. 4g lte speeds vs. your home network. https://www.verizon.com/articles/4g-lte-speeds-vs-your-home-network/.

[8] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. A buffer-based approach to rate adaptation: Evidence from a large video streaming service. In *SIGCOMM*, pages 187–198, 2014.

[9] Tony Driscoll, Suzanne Farhoud, Sean Nowling, et al. Enabling mobile augmented and virtual reality with 5g networks. Technical report, 2017.

[10] Feng Qian, Bo Han, Qingyang Xiao, and Vijay Gopalakrishnan. Flare: Practical viewport-adaptive 360-degree video streaming for mobile devices. In *MobiCom*, pages 99–114, 2018.

[11] Yu Guan, Chengyuan Zheng, Xinggong Zhang, Zongming Guo, and Junchen Jiang. Pano: Optimizing 360 video streaming with a better understanding of quality perception. In *SIGCOMM*, pages 394–407, 2019.

[12] Lan Xie, Zhimin Xu, Yixuan Ban, Xinggong Zhang, and Zongming Guo. 360probdash: Improving qoe of 360 video streaming using tile-based http adaptive streaming. In *MM*, pages 315–323, 2017.

[13] Liyang Sun, Fanyi Duanmu, Yong Liu, Yao Wang, Yinghua Ye, Hang Shi, and David Dai. Multi-path multi-tier 360-degree video streaming in 5g networks. In *MMSys*, pages 162–173, 2018.

[14] Feng Qian, Lusheng Ji, Bo Han, and Vijay Gopalakrishnan. Optimizing 360 video delivery over cellular networks. In *AllThingsCellular*, pages 1–6, 2016.

[15] Chao Zhou, Mengbai Xiao, and Yao Liu. Clustile: Toward minimizing bandwidth in 360-degree video streaming. In *INFOCOM*, pages 962–970, 2018.

[16] Vincent Sitzmann, Ana Serrano, Amy Pavel, Maneesh Agrawala, Diego Gutierrez, Belen Masia, and Gordon Wetzstein. Saliency in vr: How do people explore virtual environments? *IEEE transactions on visualization and computer graphics*, 24(4):1633–1642, 2018.

[17] Ali Borji, Dicky N Sihite, and Laurent Itti. What stands out in a scene? a study of human explicit saliency judgment. *Vision research*, 91:62–77, 2013.

[18] Duin Baek, Hangil Kang, and Jihoon Ryoo. Sali360: design and implementation of saliency based video compression for 360° video streaming. In *MMSys*, pages 141–152, 2020.

[19] Wei Huang, Lianghui Ding, Hung-Yu Wei, Jenq-Neng Hwang, Yiling Xu, and Wenjun Zhang. Qoe-oriented resource allocation for 360-degree video transmission over heterogeneous networks. *arXiv preprint arXiv:1803.07789*, 2018.

[20] Zhiming Hu, Congyi Zhang, Sheng Li, Guoping Wang, and Dinesh Manocha. Sgaze: A data-driven eye-head coordination model for realtime gaze prediction. *IEEE transactions on visualization and computer graphics*, 25(5):2002–2010, 2019.

[21] Anjul Patney, Marco Salvi, Joohwan Kim, Anton Kaplanyan, Chris Wyman, Nir Benty, David Luebke, and Aaron Lefohn. Towards foveated rendering for gaze-tracked virtual reality. *ACM Transactions on Graphics (TOG)*, 35(6):179, 2016.

[22] Conor Kelton, Jihoon Ryoo, Aruna Balasubramanian, and Samir R Das. Improving user perceived page load times using gaze. In *NSDI*, pages 545–559, 2017.

[23] Yanyu Xu, Yanbing Dong, Junru Wu, Zhengzhong Sun, Zhiru Shi, Jingyi Yu, and Shenghua Gao. Gaze prediction in dynamic 360 immersive videos. In *CVPR*, pages 5333–5342, 2018.

[24] Wenguan Wang, Jianbing Shen, and Ling Shao. Consistent video saliency using local gradient flow optimization and global refinement. *IEEE transactions on image processing (TIP)*, 24(11):4185–4196, 2015.

[25] Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen. Shifting more attention to video salient object detection. In *CVPR*, pages 8554–8564, 2019.

[26] Guanbin Li, Yuan Xie, Tianhao Wei, Keze Wang, and Liang Lin. Flow guided recurrent neural encoder for video salient object detection. In *CVPR*, pages 3243–3252, 2018.

[27] Ioannis Agtzidis, Mikhail Startsev, and Michael Dorr. 360-degree video gaze behaviour: A ground-truth data set and a classification algorithm for eye movements. In *MM*, pages 1007–1015, 2019.

[28] Erwan J David, Jesús Gutiérrez, Antoine Coutrot, Matthieu Perreira Da Silva, and Patrick Le Callet. A dataset of head and eye movements for 360 videos. In *MMSys*, pages 432–437, 2018.

[29] Wikipedia. Mean opinion score. https://en.wikipedia.org/wiki/Mean_opinion_score.

[30] Cristina Perfecto, Mohammed S Elbamby, Javier Del Ser, and Mehdi Bennis. Taming the latency in multi-user vr 360°: A qoe-aware deep learning-aided multicast framework. *IEEE Transactions on Communications*, 68(4):2491–2508, 2020.

[31] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *CVPR*, pages 280–287, 2014.

[32] Yuchao Gu, Lijuan Wang, Ziqin Wang, Yun Liu, Ming-Ming Cheng, and Shao-Ping Lu. Pyramid constrained self-attention network for fast video salient object detection. In *AAAI*, pages 10869–10876, 2020.

[33] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *ECCV*, pages 715–731, 2018.

[34] Mallesham Dasari, Arani Bhattacharya, Santiago Vargas, Pranjal Sahu, Aruna Balasubramanian, and Samir R Das. Streaming 360-degree videos using super-resolution. In *INFOCOM*, pages 1977–1986. IEEE, 2020.

[35] Zhicheng Li, Shiyin Qin, and Laurent Itti. Visual attention guided bit allocation in video compression. *Image and Vision Computing*, 29(1):1–14, 2011.

[36] Apple Support. Equirectangular projection. https://support.apple.com/en-lamr/guide/final-cut-pro/vera81de1b2e/mac.

[37] GOOGLE AR and VR. Bringing pixels front and center in vr video. https://blog.google/products/google-ar-vr/bringing-pixels-front-and-center-vr-video/.

[38] Ned Greene. Environment mapping and other applications of world projections. *IEEE Computer Graphics and Applications*, 6(11):21–29, 1986.

[39] Keisuke Tateno, Nassir Navab, and Federico Tombari. Distortion-aware convolutional filters for dense prediction in panoramic images. In *ECCV*, pages 707–722, 2018.

[40] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[41] Pyramidcsa project. https://github.com/guyuchao/PyramidCSA.

[42] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NIPS*, pages 3320–3328, 2014.

[43] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[44] Ching-Ling Fan, Jean Lee, Wen-Chih Lo, Chun-Ying Huang, Kuan-Ta Chen, and Cheng-Hsin Hsu. Fixation prediction for 360 video streaming in head-mounted virtual reality. In *NOSSDAV*, pages 67–72, 2017.

[45] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[46] David S Johnson, Cecilia R Aragon, Lyle A McGeoch, and Catherine Schevon. Optimization by simulated annealing: An experimental evaluation; part i, graph partitioning. *Operations research*, 37(6):865–892, 1989.

[47] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand. Overview of the scalable video coding extension of the h. 264/avc standard. *IEEE Transactions on circuits and systems for video technology*, 17(9):1103–1120, 2007.

[48] Htc vive pro eye. https://www.vive.com/uk/product/vive-pro-eye/.

[49] Tobii pro lab. https://www.tobiipro.com/product-listing/tobii-pro-lab/.

[50] HTC VIVE. Vive wireless adapter - untethered virtual reality is here. https://www.vive.com/us/accessory/wireless-adapter/.

[51] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010.

[52] Samsung. Samsung galaxy s20. https://www.samsung.com/us/mobile/galaxy-s20-5g.

[53] Kevin Boos, David Chu, and Eduardo Cuervo. Flashback: Immersive virtual reality on mobile devices via rendering memoization. In *MobiSys*, pages 291–304, 2016.

[54] Ffmpeg. https://github.com/FFmpeg/FFmpeg.

[55] Marko Viitanen, Ari Koivula, Ari Lemmetti, Arttu Ylä-Outinen, Jarno Vanne, and Timo D. Hämäläinen. Kvazaar: Open-source hevc/h.265 encoder. In *MM*, 2016.

[56] Moesen Gert. Viewport dependent mpeg-dash streaming of 360 degree natively tiled hevc video in web browser context.

[57] Cyril Concolato, Jean Le Feuvre, Franck Denoual, Frédéric Mazé, Eric Nassor, Nael Ouedraogo, and Jonathan Taquet. Adaptive streaming of hevc tiled videos using mpeg-dash. *IEEE transactions on circuits and systems for video technology*, 28(8):1981–1992, 2017.

[58] Jean Le Feuvre, Cyril Concolato, and Jean-Claude Moissinac. Gpac: Open source multimedia framework. In *MM*, page 1009–1012. Association for Computing Machinery, 2007.

[59] Chrille89. Hevc tiles merger. https://github.com/Chrille89/DashPlayer, 2019.

[60] A-Frame. A web framework for building virtual reality experiences. https://github.com/aframevr/aframe.

[61] Belgium 4g/lte bandwidth logs. https://users.ugent.be/~jvdrhoof/dataset-4g/, unknown.

[62] Ravi Netravali, Anirudh Sivaraman, Somak Das, Ameesh Goyal, Keith Winstein, James Mickens, and Hari Balakrishnan. Mahimahi: Accurate record-and-replay for http. In *ATC*, pages 417–429, 2015.

[63] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. A control-theoretic approach for dynamic adaptive video streaming over http. In *SIGCOMM*, pages 325–338, 2015.

[64] 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.

[65] VideoLAN. x264, the best h.264/avc encoder. https://www.videolan.org/developers/x264.html.

[66] Pensieve project. https://github.com/hongzimao/pensieve.
[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 *NSDI*, pages 495–511, 2020.
[68] Hui Wang, Vu-Thanh Nguyen, Wei Tsang Ooi, and Mun Choon Chan. Mixing tile resolutions in tiled video: A perceptual quality assessment. In *NOSSDAV*, pages 25–30, 2014.
[69] Yanan Bao, Tianxiao Zhang, Amit Pande, Huasen Wu, and Xin Liu. Motion-prediction-based multicast for 360-degree video transmissions. In *SECON*, pages 1–9. IEEE, 2017.
[70] Tan Xu, Q Fan, and Bo Han. Content assisted viewport prediction for panoramic video streaming. In *Workshop on Computer Vision for AR/VR*, 2019.
[71] Yixuan Ban, Lan Xie, Zhimin Xu, Xinggong Zhang, Zongming Guo, and Yue Wang. Cub360: Exploiting cross-users behaviors for viewport prediction in 360 video adaptive streaming. In *ICME*, pages 1–6, 2018.
[72] Chenge Li, Weixi Zhang, Yong Liu, and Yao Wang. Very long term field of view prediction for 360-degree video streaming. In *MIPR*, pages 297–302, 2019.
[73] Miao Hu, Jiawen Chen, Di Wu, Yipeng Zhou, Yi Wang, and Hong-Ning Dai. Tvg-streaming: Learning user behaviors for qoe-optimized 360-degree video streaming. *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.
[74] Afshin Taghavi Nasrabadi, Anahita Mahzari, Joseph D Beshay, and Ravi Prakash. Adaptive 360-degree video streaming using scalable video coding. In *MM*, pages 1689–1697, 2017.
[75] Jian He, Mubashir Adnan Qureshi, Lili Qiu, Jin Li, Feng Li, and Lei Han. Rubiks: Practical 360-degree streaming for smartphones. In *MobiSys*, pages 482–494, 2018.
[76] Xiaoyi Zhang, Xinjue Hu, Ling Zhong, Shervin Shirmohammadi, and Lin Zhang. Cooperative tile-based 360 panoramic streaming in heterogeneous networks using scalable video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(1):217–231, 2018.
[77] Soonbin Lee, Dongmin Jang, JongBeom Jeong, and Eun-Seok Ryu. Motion-constrained tile set based 360-degree video streaming using saliency map prediction. In *Proceedings of the 29th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 20–24, 2019.
[78] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. Salient object detection: A benchmark. *IEEE transactions on image processing (TIP)*, 24(12):5706–5722, 2015.
[79] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE Transactions on Pattern analysis and machine intelligence (TPAMI)*, 33(2):353–367, 2010.
[80] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604, 2009.
[81] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. Neural adaptive video streaming with pensieve. In *SIGCOMM*, pages 197–210, 2017.
[82] Zahaib Akhtar, Yun Seong Nam, Ramesh Govindan, Sanjay Rao, Jessica Chen, Ethan Katz-Bassett, Bruno Ribeiro, Jibin Zhan, and Hui Zhang. Oboe: auto-tuning video abr algorithms to network conditions. In *SIGCOMM*, pages 44–58, 2018.
[83] Amrita Mazumdar, Brandon Haynes, Magdalena Balazinska, Luis Ceze, Alvin Cheung, and Mark Oskin. Vignette: perceptual compression for video storage and processing systems. *arXiv preprint arXiv:1902.01372*, 2019.
[84] Jeroen van der Hooft, Maria Torres Vega, Stefano Petrangeli, Tim Wauters, and Filip De Turck. Quality assessment for adaptive virtual reality video streaming: A probabilistic approach on the user's gaze. In *ICIN*, pages 19–24. IEEE, 2019.