This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3149642, IEEE Transactions on Multimedia

Live 360 Degree Video Delivery based on User Collaboration in a Streaming Flock

Liyang Sun, Yixiang Mao, Tongyu Zong, Yong Liu, *Fellow, IEEE,* and Yao Wang, *Fellow, IEEE*

***Abstract*—Streaming of live 360-degree video allows users to follow a live event from any view point and has already been deployed on some commercial platforms. However, the current systems can only stream the video at relatively low-quality because the entire 360-degree video is delivered to the users under limited bandwidth. Streaming video falling into user field of view (FoV) can improve bandwidth efficiency of 360-degree video delivery. In this paper, we propose to use the idea of "flocking" to simultaneously improve the accuracy of user FoV prediction and video delivery efficiency for live 360-degree video streaming. By assigning variable playback latencies to users in a streaming session based on their network conditions, a "streaming flock" is formed and led by "strong" users with low playback latencies in the front of the flock. We propose a long short-term memory (LSTM) based collaborative FoV prediction scheme where the FoV traces of users in the front of the flock are utilized to predict the FoV of users behind them. Given a predicted FoV, we develop an optimal rate allocation strategy to maximize the perceptual quality. By conducting experiments using real-world user FoV traces and LTE/5G network bandwidth traces, we evaluate the gains of the proposed strategies over several benchmarks. Our experimental results demonstrate that the proposed streaming system can increase the overall quality dramatically by about 10 dB compared with heuristic FoV prediction strategy. In addition, the network-aware flocking formation can further reduce the video freeze without influencing video quality.** [1]

I. Introduction

Live streaming of 360° video facilitates immersive view experience by allowing users to dynamically choose their view directions in live events. It has great potential to become popular in many fields, e.g., live concerts and sports, etc. However, the bandwidth requirement of 360° video is much higher than the traditional 2D-planar video. How to deliver high-quality 360° video with short playback latency over the global Internet has become a hot topic for both academia and industry. To address the bandwidth and latency challenges, there is a proven effective solution: *Field-of-View (FoV)* adaptive streaming [1]. Instead of streaming the whole 360° video, FoV streaming only streams a fraction of video within the predicted user FoV. It can significantly reduce the bandwidth requirement of 360° video streaming [2]. However, the user Quality-of-Experience (QoE) of FoV streaming largely hinges on the accuracy of user FoV prediction [3], [4]. Prediction strategies including deep neural network (DNN) based prediction and multi-user collaborative prediction algorithms have been proposed and demonstrated to be effective [5]. One unique aspect of the live 360° video streaming system is that, normally, users view the video of the same live event with different latencies. For example, a recent study has shown that the playback latencies in the commercial live streaming services on Over-the-Top (OTT) devices range from 10 to 30 seconds [6]. The latency gap among users creates the possibility to conduct *collaborative FoV prediction* and *network-aware latency control* for a live 360° video streaming session.

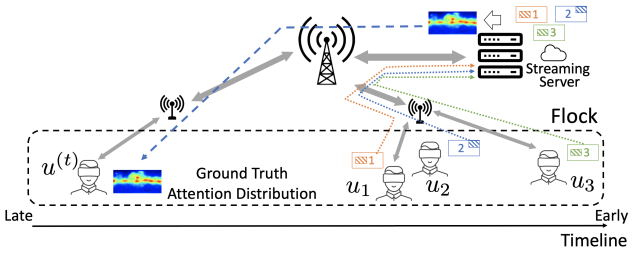


Fig. 1. User FoV sharing in flocking-based streaming.

Firstly, we investigate how the idea of "flocking" can be used to improve the efficiency of live 360° video streaming. We treat all users watching the same live event as a "streaming flock". While all the users play the video at the same speed, they can be engineered to have different playback latencies within a short range[2] based on their network conditions: users with good network conditions can stream with short video buffer to achieve low playback latency, while users with poor network conditions need longer video buffers (thus larger playback latency) to absorb bandwidth variations and sustain smooth video streaming. The relative position of a user in a streaming flock is therefore determined by her playback latency as shown in Fig. 1. The latency differences among users can be exploited to achieve "flocking gain" in both FoV prediction accuracy and playback smoothness. At a high level, the view directions of users in the front of a flock, i.e., with shorter playback latency, serve as valuable inputs to predict the view directions of users behind them, i.e., with longer playback latency. Leveraging on this, we develop an LSTM-based collaborative FoV prediction algorithm that predicts a target user's view direction for a video scene based on her own past FoV trajectory as well as the actual view directions (repre-

[1]L. Sun, Y. Mao, T. Zong, Y. Liu, and Y. Wang are with the Department of Electrical and Computer Engineering, Tandon School of Engineering, New York University, Brooklyn, NY 11201 USA (e-mail: ls3817@nyu.edu; yixiang.mao; tz1178@nyu.edu; yongliu@nyu.edu; yw523@nyu.edu).

[2]The acceptable user latency ranges vary from sub-second to tens of seconds for different types of live streaming services with different latency allowances. For the ease of streaming latency control and the clarity of the presentation, in this paper, we will focus on live streaming services where latency from a couple of seconds to ten seconds is acceptable. The flocking concept can also be applied to live streaming services with sub-second latency requirement, given that latency control can be effectively done in sub-second latency range.

sented by shared averaged attention distribution maps) of users who have watched the same video scene very recently. Given the predicted view direction (attention distribution), the rate allocation among different spatial regions in the 360 degree scope can be optimized to maximize the perceived quality. We formulate the spatial quality maximization problem and obtain a close-form solution.

The rest of the paper is organized as follows. Related work is overviewed in Section II. We formally introduce the concept of streaming flock and propose network condition aware user flocking strategies in Section III. Collaborative LSTM-based FoV prediction algorithms are developed in Section IV. Temporal and spatial rate allocation algorithms are presented in Section V, with a focus on spatial quality maximization. The proposed components are systematically evaluated to demonstrate their gains using real FoV and bandwidth traces in Section VI. The paper is concluded in Section VII.

II. Related Work

User FoV prediction plays an important role for all types of 360° video streaming applications including video-on-demand (VoD) [1], [7], [8], [9], [10], [11], [12], [13], [14], live 360° video streaming system [15], [16], [17] and interactive 360° video applications [18], [19], [20]. In [21], linear regression and deep neural network (DNN) based solutions are proposed to predict user future FoV center using historical FoV trajectory. Instead of only using the past FoV trajectory, video content features are also utilized to predict future FoV in [22]. In [5], the authors focused on FoV prediction over long time horizon for on-demand streaming, and multiple LSTM-based models are proposed. Auto Regressive Moving Average (ARMA) prediction and transition probability model are applied in [23] and [24], respectively. In [25], [5], collaborative FoV predictions based on other users' viewing directions are considered. However, these methods are proposed for VoD streaming and assume that there are always a large number of users who have watched the same video. Our proposed LSTM-based collaborative FoV prediction is specifically designed for live streaming, can work with any variable number of users and is light-weight.

Given a predicted bandwidth, tile rate allocation aims to maximize users' QoE through distributing the limited bandwidth budget to each tile. Normally, the tiles with a high probability to be watched should be assigned with a higher bitrate than the ones outside the FoV region. In [26], the tile rate allocation problem is modeled as a multiclass knapsack problem with a dynamic profit that is a function of the FoV and the buffer occupancy. DASH SRD-extension is used in [27] and the rate is allocated to tiles based on their priorities greedily. A hierarchical buffer based rate adaption algorithm is adopted in [11] in which the received tiles can still be updated if the buffer length is safe. Through subjective study, VMAF is proved to be an efficient way to evaluate 360° video quality in [28]. In [18], users' viewport quality is optimized by minimizing the distortion and variance based on the predicted viewing probabilities of all the tiles. In this paper, we also conduct rate allocation based on the predicted attention distribution (normalized viewing probabilities of tiles). But, differently, we predict the attention distribution directly using both the past attention distributions of the target user and other users' attention distributions. The formulated rate allocation problem is solved by maximizing the weighted sum of WS-PSNR of all the tiles within the predicted FoV of the user. For evaluation, we report the weighted sum of WS-PSNR of all the tiles within the user's actual FoV, noted as WS-PSNR-FOV.

Live 360° video streaming poses more challenges for both end users and video server compared with on-demand 360° video streaming. In [29], the authors proposed a measurement platform to conduct measurement on the existing commercial live 360° video streaming platforms, e.g., Facebook and YouTube. Authors of [16] proposed a live 360° video streaming system which trade-offs between the bandwidth usage and video quality within user's FoV. As with VoD, tile-based video encoding and delivery is widely used to achieve FoV-adaptive live video streaming [30], [31]. In [32], tiles with different resolutions are aggregated into one High Efficiency Video Coding (HEVC) bitstream on-the-fly. Layered coding scheme is applied in [33], [34] to reduce the occurrences of video freezes without compromising the quality and bandwidth efficiency. We adopt the standard tile-based coding and streaming in our system. We take advantage of the fact that viewers are often interested in similar regions in the 360° scope. By intentionally assigning varying playback latencies to users based on their network conditions, we can improve the accuracy of collaborative FoV prediction, while reducing the likelihood of video freezing.

Our preliminary work on flocking-based streaming was published in [17]. This paper significantly improves over [17] with more accurate LSTM-based collaborative FoV prediction, optimal rate allocation among tiles, and systematical evaluation of various gains of flocking.

III. Network-Condition-Aware Flocking

Users in a live streaming session naturally have heterogeneous network conditions. Users with stable and high-speed networks can promptly download live video segments immediately after they are generated. A short video buffer can be employed to achieve low playback latency. On the contrary, "weak" users with unstable and low-speed networks have to use long buffers to avoid video freeze. A longer buffer leads to not only longer playback latency, but also longer FoV prediction interval for which lower FoV prediction accuracy is expected. The basic idea of flocking-based 360° video streaming is to allow users within the same session to help each other. We borrow the name from bird flocking. When birds fly as a flock, the birds in the front have to fight harder against headwind. It is therefore wise to have stronger birds lead a flock. As illustrated in Fig. 2, we will follow a similar strategy to place "strong" users with better network conditions in the front of a streaming flock (with small video buffers and short playback latencies) and let the relatively "weak" users stay in the rear (with larger video buffers and longer playback latencies). In particular, the "weak" users with longer playback latencies can benefit from the "strong" users' (with shorter

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3149642, IEEE Transactions on Multimedia

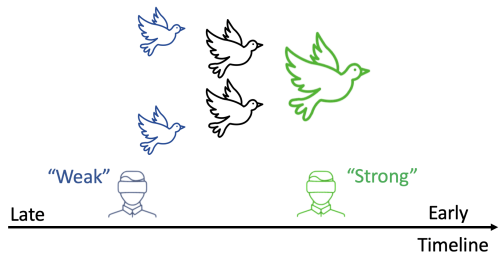


Fig. 2. Analogy between birds flock and live streaming flock. "Strong" birds or users are "flying" in the front so that the "weak" ones in the rear can benefit.

latency) ground-truth FoV information and thereby improve FoV prediction accuracy. In this section, we will elaborate how this can be naturally realized by manipulating the target playback latency and maximum buffer length on all users.

At any given time t, if a user's target latency is d, and the maximum buffer length is $B^{(u)} \leq d$, the user should be watching video generated at time $t-d$ and downloading video generated at $t-d+B+1$ where $B \leq B^{(u)}$ is the current buffer length. d and $B^{(u)}$ are two critical parameters for live 360° video streaming. Before downloading segment $t-d+B+1$, the user should first estimate its FoV based on her FoV trajectory up to $t-d$. Therefore $B^{(u)}+1$ determines the maximum FoV prediction temporal horizon, and the larger the $B^{(u)}$, the less accurate the trajectory-based self-prediction. On the other hand, for collaborative FoV prediction (to be described in Sec. IV), the user can leverage FoV information of users with shorter playback latency who have watched segment $t-d+B+1$. Therefore, the larger the d, the more potential for collaborative FoV prediction. Meanwhile, a streaming buffer is important to absorb network bandwidth oscillation, a larger $B^{(u)}$ is beneficial to achieve high quality.

In our proposed streaming flock, users at the front must have short playback latency, which means they have to assume small d and $B^{(u)}$. The immediate requirement is that they must have high bandwidth and stable network condition so that they don't run into video freeze or segment skip even with a short streaming buffer. Although they have no/low chance to benefit from collaborative FoV prediction, because $B^{(u)}$ and consequently the FoV prediction horizon is short, FoV prediction based on their own FoV history is generally more accurate. Meanwhile, for users with unstable network conditions, to maintain smooth streaming, a large $B^{(u)}$ is necessary. This naturally pushes them to the back of the flock. The negative impact of a long FoV prediction horizon resulting from large $B^{(u)}$ can be compensated by collaborative FoV prediction based on FoV information of users in the front. This cooperative flocking strategy can improve both the individual and overall user QoE.

In order to enable such network-aware assignment, before requesting the first video segment, a user operates a short-term monitoring on her current network condition. In details, through comparing her own network capability in terms of relative standard deviation (σ/μ) to the statistics of the current active users which can be downloaded from the server, a user can get her network capability rank. Then the user selects a latency group and sets her buffer upper bound based on her rank. To illustrate, in Fig. 3, users are divided into four groups with different target latencies $d_1 = G_1$, $d_2 = G_1+G_2$, $d_3 = G_1+G_2+G_3$, and $d_4 = G_1+...+G_4$. Since this is live streaming, the buffer upper bound of Group 1 $B_1^{(u)}$ should be less than G_1 [3]. Then, for latency Group 2, in order to benefit from Group 1 users' FoV information, the buffer length of the users in Group 2 should not exceed where users in Group 1 are watching. In other words, the buffer upper bound of Group 2, $B_2^{(u)}$, should not be greater than G_2. More generally, to enable Group k users to benefit from all the previous groups, its buffer upper bound should satisfy $B_k^{(u)} \leq G_k$. Overall, through intelligently assigning playback latencies and buffer upper bounds to users, they can benefit from either their own superior network condition or "stronger" users "flying" in the front of the flock.

IV. Flocking-based Collaborative FoV prediction

A user's view direction for 360° video is affected by both the distributions of the attractive objects in a video scene and her personal preferences to them. FoV prediction for future frames based on the FoV trajectories of the past frames is hard because the first appearance of a new object of interests is not predictable from the past. In this challenging scenario, knowing which areas other users (earlier viewers) have focused on for a "future" frame could greatly help the FoV prediction for the current user (later viewer). Even in the situation when no new objects appear in a frame, the distribution of the viewing areas of the earlier viewers can still help to predict the FoV of the later viewer, especially when the distribution is non-uniform and has one or a few focuses. In [25], [5], FoV prediction based on multiuser trajectories was proposed. With the help of information from other users, the prediction accuracy for a target user can be greatly improved. These studies were based on the user FoV traces collected from VoD streaming and assumed that an equal number of earlier viewers are always available and is relatively large. However, in live 360° video streaming, the number of earlier viewers for a target user is variable and dynamically changing. The flocking formation proposed in the previous section facilitates collaborative FoV prediction in live 360° video streaming. We develop a LSTM-based collaborative FoV prediction algorithm that adaptively combines the prediction from the user's own past trajectory and the ground truth information of users in the front of the flock.

A. Attention Distribution Prediction with LSTM

As shown in Fig. 4, we propose a LSTM-based collaborative FoV prediction algorithm. A video segment consists of multiple frames. A user's FoV varies for frames in the same segment. We characterize a user's FoV for a segment using attention distribution, which is a 2D map representing how a user's attention is distributed among all the tiles within an Equirectangular Projection (ERP). More specifically, for each

[3] If the user finishes downloading the latest video segment and the following video segment is still under encoding or transcoding by the server, the user has to wait for the server processing to be completed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3149642, IEEE Transactions on Multimedia

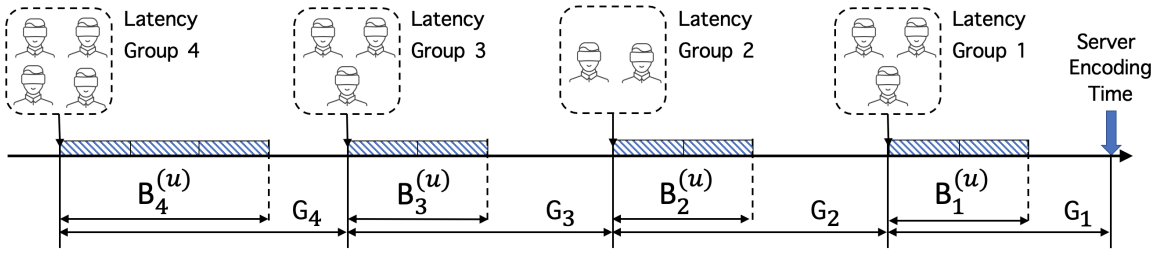


Fig. 3. Configuration of Playback Latency and Buffer Upper Bound to Achieve Network-condition-aware Flocking.

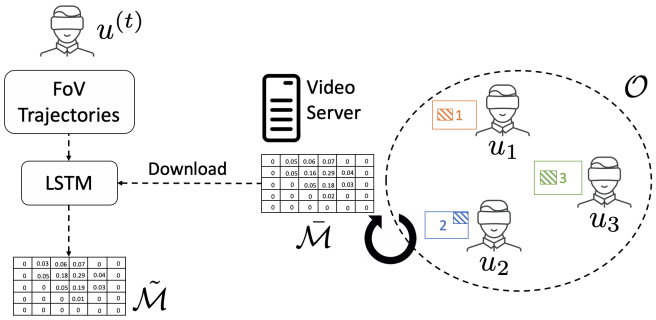


Fig. 4. LSTM-based collaborative FoV prediction. The target user $u^{(t)}$ predicts future attention distribution $\tilde{\mathcal{M}}$ by utilizing the average and variance attention maps $\bar{\mathcal{M}}$ of users in the front of the flock.

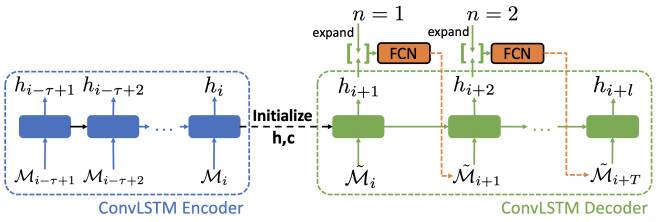


Fig. 5. LSTM$_s$: ConvLSTM + FCN based target users attention distribution prediction.

frame, given the target user's view direction, we can calculate the user's attention weights for all tiles of the frame: if a tile falls into the FoV, its attention weight is one; if a tile is outside the FoV, its weight is zero; for a tile partially overlapping with the FoV, its attention weight is the ratio of the overlapped area. The user's attention distribution on a segment is represented by the normalized average attention weights of all tiles over all the frames within the segment. The calculated attention distribution characterizes the fraction of time and size that a specific tile falls into the user's FoV over the whole duration of a segment.

When a target user $u^{(t)}$ predicts future attention distribution for a video segment (video segment index is omitted for simplicity), ground truth FoV information of some other users who have watched the requested video segment might be available. To utilize it, each user u_k uploads her past FoV trajectories of each viewed segment to the video server periodically along with request for future segments. All the uploaded FoV trajectories will be stored at the video server and can be converted to the attention distribution maps for all viewed segments. Then, while distributing Media Presentation Description (MPD) files, the average attention distributions $\bar{\mathcal{M}}_{i+n}$ of users who have watched segments $i+n$ with $n = 1, 2, \cdots, T$ are delivered to the target user who is watching segment i. To facilitate collaborative FoV prediction, variance maps of the attention distributions for these segments are also calculated and delivered to the target user. The variance map for a segment records the variance of attention weights at each tile among all earlier viewers for this segment. The shared average attention distribution $\bar{\mathcal{M}}$ and variance map can be calculated/updated periodically on the server when more users' FoV trajectories are uploaded. In this way, only the average and variance of the earlier viewers' attention distribution are sent to the target user so that the system scalability can be improved dramatically compared with [17].

Depending on whether the attention distributions about other users are available, two prediction models are developed: a single-user LSTM (LSTM$_s$) and a collaborative LSTM (LSTM$_c$). Details of these models are introduced below.

1) Self Prediction with LSTM$_s$: Firstly, for the target user $u^{(t)}$, let $\mathcal{O}$ denote the set of other users who have viewed the requested segment, if $\mathcal{O}$ is empty at time i, she can only predict $\tilde{\mathcal{M}}$ based on her own historical FoV trajectory. Illustrated in Fig. 5, the target user generates attention distribution $\mathcal{M}_{i-j}$ for each of the past τ segments (j from $\tau-1$ to 0) using her historical FoV trajectory. The sequence of $\mathcal{M}_{i-j}$ is fed into a Convolutional LSTM (ConvLSTM) encoder to generate a hidden state h_i and cell state c_i, which are used to initialize the LSTM decoder. Then the LSTM decoder recursively generates the predicted attention distribution for segment $i+n$, with $n = 1, 2, ...T$ where T is maximum the prediction horizon. Specifically, to predict the attention distribution for future time $i+n$, the predicted attention $\tilde{\mathcal{M}}_{i+n-1}$ (which equals to $\mathcal{M}_i$ when $n = 1$) and the previous hidden and memory states h_{i+n-1} and c_{i+n-1} are used to generate the hidden state h_{i+n} using a ConvLSTM decoder. In order for the model to be aware of the time lapse since the last ground-truth input in the decoder, the hidden state h_{i+n} is concatenated with the prediction step indicator n (which is replicated to form a matrix with the same spatial dimension as h_{i+n}), and used to predict $\tilde{\mathcal{M}}_{i+n}$ through a fully convolutional network (FCN) [5]. This predicted map will be used recursively to predict the attention distribution for the next segment, until the prediction for the target segment is determined. This model is called LSTM$_s$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3149642, IEEE Transactions on Multimedia

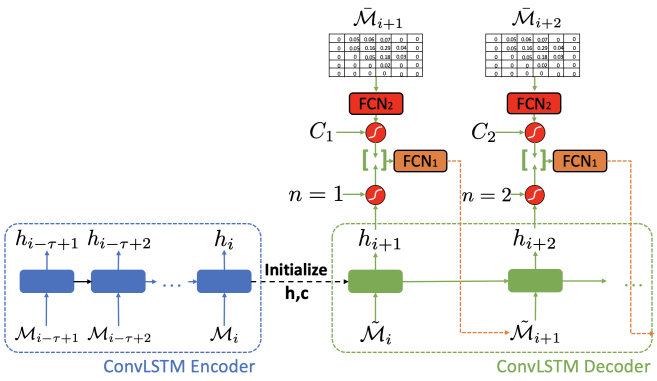


Fig. 6. LSTM$_c$: ConvLSTM + FCN based collaborative attention distribution prediction. $\bar{\mathcal{M}}_{i+n}$ represents the concatenation of the average and variance maps at segment $i+n$.

2) Collaborative Prediction with LSTM$_c$: As shown in Fig. 6, first of all, the target user $u^{(t)}$ still adopts ConvLSTM-based encoder to capture historical information from her own historical attention distribution. In addition, to predict the attention distribution for segment $i+n$, the statistics of other users' attention distribution $\bar{\mathcal{M}}_{i+n}$ is fed into FCN$_2$ to convert it to the same feature domain as the hidden state of the target user. Specifically, we concatenate the average and variance distribution maps, where the value for each tile is either the mean or the variance of the attention values among all collaborating viewers for segment $i+n$. The purpose of introducing variance is to take into consideration the variability of users' attention distributions for a particular tile. Before concatenating hidden state h_{i+n} and the shared attention $\bar{\mathcal{M}}_{i+n}$, these two maps are point-wise attenuated to take into account of the prediction step n and the number of collaborating viewers $C_n = |\mathcal{O}|_{i+n}$. In more detail, ① if the prediction horizon n is large, the prediction accuracy based on the past information is likely low. Therefore, we introduce a decay function $e^{-(\alpha_1 n+\beta_1)}$ to shrink the contribution from h_{i+n}. ② Besides, if $\bar{\mathcal{M}}_{i+n}$ is generated by a large group of users, it is likely a good reflection of the visual saliency of different regions in the scene and hence a better indication of the target user's potential attention distribution. Therefore, an exponentially increasing function $e^{\alpha_2 C_n+\beta_2}$ is used to adjust the impact of $\bar{\mathcal{M}}_{i+n}$ based on C_n. Given the two weighted attention distributions (h_{i+n} and ground truth $\bar{\mathcal{M}}_{i+n}$), FCN$_1$ is utilized to generate the final attention distribution $\tilde{\mathcal{M}}_{i+n}$. We call this collaborative FoV prediction model as LSTM$_c$.

3) LSTM$_s$ and LSTM$_c$ Model Design: Both the encoder and decoder contain 3 ConvLSTM layers with 64, 32 and 16 filters respectively. The output shape is same as the input attention distribution shape with spatial dimension of 16×32. For LSTM$_s$, the hidden state outputs from all the 3 ConvLSTM decoder layers are concatenated and fed into FCN. In addition, n is expanded into the same shape of the hidden state output h_{i+n} with 16 channels and concatenated with h_{i+n}. The FCN also has three layers, generating 32, 64 and 1 channel hidden-state feature maps, with the last one being the predicted attention map. In the LSTM$_c$ model, FCN$_1$ uses three layers of 64, 128 and 1 filters and FCN$_2$ consist of two layers with 32 and 64 filters. The parameters $(\alpha_1, \alpha_2, \beta_1, \beta_2)$ in the decay functions are trained together with other model parameters. KL divergence is utilized as the loss function during the training phase.

4) System Overhead Analysis: To support the collaborative prediction, users have to upload his/her true FoV trajectory to the server and download the shared (average and variance) distribution maps from the server. More specifically, in the uploading phase, if a user's FoV viewing direction is sampled in the format of (timestamp, yaw, pitch, roll) with frequency of 30 Hz, then, for each second, the uploading data overhead equals $30 \times 4 \times 4 = 480$ bytes (assuming each value of a sample takes 4 bytes). In the downloading part, users have to download two maps that are cut into 32×16 tiles. The total data size of the two maps would be $2 \times 32 \times 16 \times 4 = 4$ KBytes. Both the uploading and downloading of the FoV information can be piggybacked with the video segment request and video segment, respectively, without introducing extra latency. Compared with the video content data at rate of tens or hundreds of Mbps, these FoV information overheads are negligible.

V. Temporal and Spatial Rate Allocation

Similar to 2D-planar video streaming, to cope with dynamic network bandwidth, the streaming rate of live 360° video has to be adapted over time. It can be achieved by partitioning a 360° video into temporal segments with a chosen duration, e.g., one second, and encoding (at the server) each video segment to multiple versions with different rates. A streaming client will dynamically select the rate version for each requested video segment based on the network condition. Additionally, to cope with user view direction changes, rate allocation over different spatial regions on the 360° sphere within the same segment has to be adapted. In a tile-based design, each 360° video segment is spatially partitioned into multiple tiles in the Equirectangular Projection (ERP) format, and each tile is coded with multiple rates. The rate and consequently the quality chosen for a tile should be determined by the likelihood that this tile will be viewed by the user. The tiles within the predicted FoV should be allocated with more bits than the tiles around the boundary or outside of the predicted FoV. In addition, the impact of the rate difference between two spatially adjacent tiles on the user perceived video quality should also be considered.

We use segment+tile based design to achieve temporal and spatial rate adaption. Similar to DASH for 2D-planar video, we can select the video rate for a 360° video segment using buffer-based and/or rate-based algorithms. Each segment consists of multiple frames, and a user's view direction can change at the frame-level. A tile within a user's FoV at one frame may fall out of her FoV at the next frame. Instead of predicting one view direction for each segment, as presented in Section IV, we predict the *tile attention distribution* $\tilde{\mathcal{M}}$, i.e., the fraction of time that a specific tile falls into the user's FoV over the whole duration of a segment. Given the predicted attention distribution, we further solve the spatial

rate allocation problem using Lagrange multiplier so that the rendering quality within the user's predicted FoVs over the entire segment is maximized.

A. Bandwidth Prediction and Temporal Rate Allocation

Due to the dynamic network environment, adaptive rate control becomes crucial to video streaming especially for live video streaming with short buffer. Bandwidth prediction is one of the most important parts in adaptive rate control. In our system, bandwidth prediction is an independent component, and any effective bandwidth prediction algorithm can be applied. In the simulation results shown later, the harmonic mean of the download bandwidth of the past 10 video segments is calculated as the predicted bandwidth for the next segment. Instead of video rate, user QoE is also affected by the rate fluctuation between two adjacent video segments. In order to solve this problem, temporal rate adaption should be optimized. For example, model predictive control (MPC) [35], [36] can be applied to get the optimal temporal rate allocation for a given sequence of predicted bandwidth. In addition, model-free based solution, e.g., reinforcement learning (RL) [37], [38], [39], can also solve this problem efficiently through exploring the optimal rate allocation in the environment. In our current work, we operate temporal rate allocation based on one-step bandwidth prediction and will explore potential gain of different temporal rate adaption algorithms in future work.

B. Optimal Spatial Rate Allocation

As each 360° video segment is spatially divided into multiple tiles and requested independently, given a limited bandwidth budget, we want to allocate more bits for the tiles that have high likelihood of being viewed. Based on the study in [10], the quality-rate (Q-R) functions of tiles differs with tile position. For example, to obtain the same quality, the tiles near the equator and south pole require more bits than the tiles near the north pole. Therefore, we jointly consider the location-dependent weight (affected by the tile's vertical position) and the predicted attention distribution (viewing probability) of tiles while allocating the predicted bandwidth budget. Specifically, we formulate rate allocation problem for a video segment as following:

$$\begin{aligned} \max_{\{r_j\}} \quad & \mathcal{Q} = \sum_{j=1}^{J} p_j w_j Q_j(r_j) \\ \textbf{subject to} \quad & \sum_{j=1}^{J} r_j = c \\ & r_j \geq 0, \qquad j = 1, 2, ..., J \end{aligned} \tag{1}$$

where c is the total rate budget for the segment, r_j is the number of bits allocated for tile j, $Q_j(\cdot)$ is the Q-R function of tile j which is modeled by a logarithmic function $Q_j(r_j) = a_j \log(r_j) + b_j$, following the findings in [10]. Note that $Q_j(\cdot)$ is a tile-specific function which is fitted to the weighted-to-spherically-uniform peak-signal-to-noise ratio (WS-PSNR) of the jth tile under different rates. Therefore, $\mathcal{Q}$ represents the weighted average of the WS-PSNR of all the tiles within user's FoV. As shown in [10], tiles at different vertical positions in the ERP (corresponding to different latitudes in the 360° sphere) have quite different Q-R characteristics (with varying a_j and b_j). p_j is the predicted users' attention to the jth tile by the LSTM models, and w_j is a location-dependent weight of tile j, proportional to the area that this tile contributes in the 360° sphere. Specifically, the weight of a tile j is the average weight of all the pixels in tile j and the weight of a pixel is represented by $\cos\theta$ where θ is the latitude of the pixel. For example, a tile near the north or south pole corresponds to a smaller area than a tile near the equator.

To solve the constrained maximization problem, Lagrange multiplier can be introduced so that we can reformulate the problem as following:

$$\begin{aligned} \mathcal{Q}' &= \sum_{j=1}^{J} p_j w_j a_j \log(r_j) + \lambda(\sum_{j=1}^{J} r_j - c) + \sum_{j=1}^{J} p_j w_j b_j \\ &= \sum_{j=1}^{J} z_j \log(r_j) + \lambda(\sum_{j=1}^{J} r_j - c) + \sum_{j=1}^{J} p_j w_j b_j \end{aligned} \tag{2}$$

with $z_j = p_j w_j a_j$.
Through setting derivate of $\mathcal{Q}'$ to r_j to zero:

$$\begin{aligned} &\frac{\partial \mathcal{Q}}{\partial r_j} = \frac{z_j}{r_j} + \lambda = 0, \text{ for all } j \in [1, J] \\ &\sum_{j=1}^{J} r_j = c, \end{aligned} \tag{3}$$

we obtain the following the close-form rate allocation solution:

$$r_j = \frac{z_j}{\sum_{j=1}^{J} z_j} c, \text{ for all } j \in [1, J]. \tag{4}$$

The result reveals that the rate for a tile should not only depend on its viewing probability but also its location (which affects its quality-rate slope reflected by a_j and the spherical area it contributes indicated by w_j). In practice, each tile is only encoded into a finite set of bitrates. Therefore, we quantize the optimal rate allocation solution generated by Eq. (4) by selecting the highest encoded bitrate that is less than or equal to the optimal solution.

VI. Experiments and Evaluation

A. Experiments Configuration

The proposed live 360° streaming system is evaluated using real LTE/5G network bandwidth traces [40] and users' FoV dataset [41]. The bandwidth traces were collected under the mobile scenarios with cellular access mode switching between 5G and LTE. As the original bandwidth can be either too high (more than 100 Mbps in 5G scenario) or too low (being zero for several seconds during handover) for the live 360° video streaming, we filter the bandwidth traces with lower and upper bounds of 3 and 80 Mbps. As each of the videos in the FoV dataset [41] includes traces from different numbers of users, we filter the dataset by selecting videos with at least 31 users' traces. Finally, 55 videos are selected. FoV traces of 33 videos are used for the LSTM$_s$ and LSTM$_c$ training, and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3149642, IEEE Transactions on Multimedia

the other 22 are used for validation/testing. Both LSTM$_s$ and LSTM$_c$ models are trained in 1,000 epochs. In order to avoid overfitting, we adopt early stopping during the training so that when the validation loss keeps increasing for several epochs, the training will terminate.

The Q-R curve is generated by performing video coding on a JVET 360° video testing sequence [42]. HEVC reference software under JVET common test conditions is used for the video coding. The Q-R curve is fitted to the WS-PSNR with different quantization parameters (QP). The entire 360° video (8K resolution) is divided into 32×16 tiles with 256×256 pixels in each tile [20]. The user FoV span size is assumed to be $90° \times 90°$. To allow adaptive streaming, the original 360° video is coded into multiple bitrates as: {3, 5, 10, 20, 40, 70, 100} Mbps. Each video segment is of 1 second.

We use tile overlap ratio (TOR) [5] between the actual attention map and the predicted attention map to evaluate the FoV prediction performance. Firstly, for each video segment, we find all the N tiles with non-zero values in the ground truth attention distribution. Then, in the predicted attention distribution, we sort the tiles on their predicted viewing probabilities. The overlap ratio between the N largest tiles in the predicted attention distribution and the ground truth is defined as the tile overlap ratio.

B. Evaluation of different FoV prediction methods

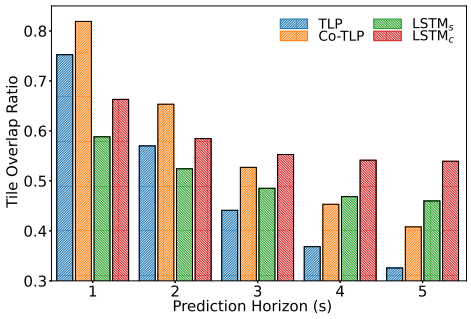


Fig. 7. Tile overlap ratio of different heuristic and LSTM based prediction algorithms for different prediction horizons.

1) Comparison of LSTM and TLP based FoV prediction: We compare the LSTM based FoV predictions (LSTM$_s$ and LSTM$_c$) with heuristic FoV prediction algorithms: truncated linear prediction (TLP) and Collaborative truncated linear prediction (Co-TLP) proposed in [17]. For each of the algorithms, we run FoV prediction for all the 31 users for 10 videos in the testing dataset. At each step, attention distributions of future 5 seconds are predicted. For LSTM based prediction, the distributions of the past 10 seconds are considered as the input. TLP first truncates the past FoV trajectory so that the remaining segment trajectory is monotonic and then predicts the FoV centers of the future seconds by linearly extrapolating the truncated trajectory using linear regression. The attention map $\check{\mathcal{M}}$ is then generated from the predicted FoV centers. Co-TLP additionally combines the predicted attention $\check{\mathcal{M}}$ with a weighted average distribution $\hat{\mathcal{M}}$ of other users' ground truth attention distributions as the final prediction. To generate the weighted average distribution $\hat{\mathcal{M}}$, the trajectory distance between each other user and the target user is calculated based on their historical FoV and the weight is reversely proportional to the trajectory distance [17]. For the FoV prediction of each target user, we assume the future ground-truth FoV of all the other 30 users are available in Co-LTP and LSTM$_c$.

As shown in Fig. 7, without benefiting from FoV information of other users, TLP leads to lower TOR than Co-TLP. In addition, the TOR of TLP decreases rapidly for long prediction horizons. With Co-TLP, by collaboratively utilizing other users' ground truth attention distribution, the prediction performance is improved significantly compared to TLP, especially for long horizons. For LSTM$_s$ which adopts deep neural network to predict future attention distribution, higher TOR can be achieved for long prediction interval, e.g. $\geq$ 3 seconds. By adopting LSTM$_c$, the highest TOR can be achieved for horizons larger than two seconds. However, while predicting one or two seconds ahead, Co-TLP still performs the best. Overall, LSTM based predictions perform better than TLP heuristic algorithms for long prediction horizons. In addition, a collaborative prediction (Co-TLP or LSTM$_c$) always achieves higher TOR than a self prediction (TLP or LSTM$_s$), suggesting that the flocking-based collaborative FoV prediction is always helpful.

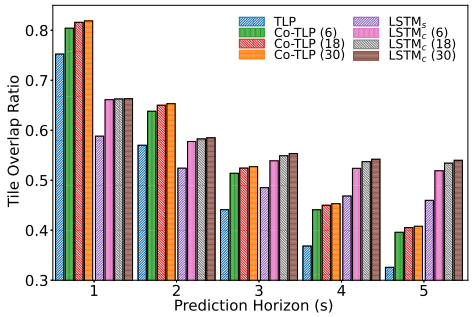


Fig. 8. FoV prediction accuracy with different numbers of other users with shorter latency.

2) Impact of number of available collaborating users: Generally, for different users, the numbers of users in front of them in the flock vary corresponding to their relative latencies. Therefore we evaluate how the collaborative FoV prediction is affected by the number of users in the front of the target user. Fig. 8 shows the average TOR for all the 31 users for the 10 testing videos with different numbers of collaborating users. Compared with TLP, with just 6 collaborating users, the TOR of Co-TLP can be improved significantly. With more users available, FoV prediction becomes more accurate, but the gain is small. For LSTM based prediction, similar trend is observed. In addition, for longer prediction interval, the relative improvement brought by having more collaborating users becomes more significant.

3) Comparison of different LSTM input lengths: We compare the performance of the LSTM$_s$ FoV prediction model

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3149642, IEEE Transactions on Multimedia

with different lengths of users' past attention distribution as input. We use the identical experiment settings to Sec. VI-B. The results in Fig. 9 demonstrate that longer past attention distribution can improve the prediction accuracy. Especially when the input length is increased from 5s to 10s, the improvement is the most significant. However, the FoV prediction performance of 10s and 15s input length are roughly the same for prediction horizons of 1s or 2s. For longer prediction horizons, 15s past attention distribution leads to slightly better performance than 10s input length. During the training phase, utilizing 15s input length takes 638 epochs to converge which is almost 1.5 times as long as using 10s input length (434 epochs). Overall, we consider using the past 10s attention distribution as the best tradeoff between FoV prediction accuracy and training time.

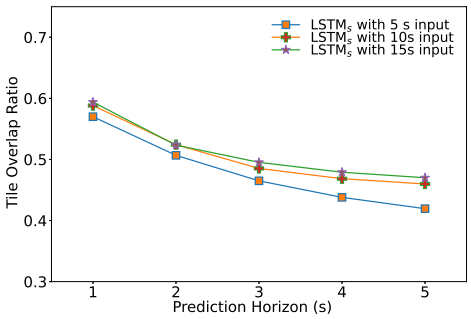


Fig. 9. Tile overlap ratio of LSTM$_s$ prediction model for taking different lengths of users' past attention distribution as input.

C. System Performance Evaluation

To evaluate the gains from different components of the proposed "flocking" system, we simulate and compare the system performance under different settings:

- **Non-Flocking** which is a non-flocking streaming strategy. The future FoV attention distribution is predicted only based on the target user's own past FoV trajectory using LSTM$_s$. Playback latency and buffer upper bound are randomly assigned.
- **Flocking** which is the proposed flocking strategy in this paper. LSTM$_c$ is utilized if other users' ground truth FoV information is available. Network condition based latency and buffer upper bound assignment are adopted. Rate allocation is determined by optimizing the quality, as explained in Sec. V-B.
- Flocking with Random Latency and Buffer Assignment (**Flocking-rand**) which uses LSTM$_c$ for collaborative FoV prediction without adopting network condition based latency and buffer upper bound assignment. The latency and buffer upper bound are randomly assigned to each user. All the other configurations are identical to the proposed Flocking strategy.
- Flocking with Proportional Rate Allocation (**Flocking-prop**) which is the same as Flocking, except that it does not use quality optimized rate allocation for tiles. Rate is directly assigned to each tile proportional to the predicted viewing probability.

TABLE I
ABLATION STUDY: SYSTEM SETTINGS

Algorithms	FoV Prediction	Co-Prediction	Lat/Buff Assign	Rate Allocation
Non-Flocking	LSTM$_s$	✗	✗	✓
Flocking	LSTM$_s$/LSTM$_c$	✓	✓	✓
Flocking-rand	LSTM$_s$/LSTM$_c$	✓	✗	✓
Flocking-prop	LSTM$_s$/LSTM$_c$	✓	✓	✗
Flocking-TLP	TLP/Co-TLP	✓	✓	✓

- Flocking with TLP FoV prediction (**Flocking-TLP**) which is same as the Flocking system, except that it uses the Co-TLP for FoV prediction, which includes TLP as a special case when there are no collaborating users available.

Table I summarizes the algorithm settings of these benchmark systems. Table II reports the performances of these systems in terms of three QoE metrics: FoV prediction accuracy measured by TOR, duration of freeze in seconds, and the rendering quality (WS-PSNR-FOV) in terms of the weighted average of the WS-PSNR of tiles covered by users' FoV where the weights are the actual user attention. We report the performances for different latency groups as well as the average performance among all users. The initial latency and buffer upper bound of each group is defined in Table II.

1) Benefit from collaborative FoV prediction: First, we evaluate the flocking-based FoV prediction using the user FoV and LTE/5G network bandwidth traces discussed in Sec. VI-A. For this experiment, 8 videos are selected from the testing set. For each video, there are 31 users and each of them is assigned a unique FoV trajectory. The results show the average of the 8 videos. For this study, the latency assignment is not based on the network conditions as discussed in Sec. III, rather users are assigned to the 4 latency groups randomly with 8 users in each group (except for the last group which has 7 users). The same assignment is used for the evaluation of different FoV prediction methods. We choose to have 4 latency groups, with latency of 3s, 8s, 13s and 19s, respectively. We set the actual initial latency of each user to be slightly different from the group average latency by adding a random noise. The buffer upper bounds for groups 1 to 4 are set as: 2s, 3s, 4s and 5s respectively, following Fig. 3.

Fig. 10 illustrates the average TOR between the predicted and the ground truth tile attention distribution for all video segments. Self-prediction means that only LSTM$_s$ is used. However, in collaborative prediction, if other users' ground-truth is available, LSTM$_c$ is utilized to predict future tile attention distribution. For latency group 1, as they seldom have earlier viewers to enable the use of LSTM$_c$, LSTM$_s$ and LSTM$_c$ achieve almost the same TOR. However, the improvement brought by collaborative prediction increases as the group latency increases. The results confirm our hypothesis that the users with long latency can benefit from front users regarding the FoV prediction. While comparing the TOR for each user, we find that collaborative prediction always

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3149642, IEEE Transactions on Multimedia

TABLE II
QOE METRICS FOR DIFFERENT SYSTEMS

Algorithms	Group 1 (3s, 2s)			Group 2 (8s, 3s)			Group 3 (13s, 4s)			Group 4 (19s, 5s)			Overall		
	TOR	Freeze	Quality	TOR	Freeze	Quality	TOR	Freeze	Quality	TOR	Freeze	Quality	TOR	Freeze	Quality
Non-Flocking	0.42	1.74	44.4	0.46	1.07	44.6	0.47	**0.7**	45.1	0.43	0.75	43.8	0.44	1.06	44.5
Flocking	**0.48**	0.73	**45.9**	**0.52**	0.79	**45.4**	0.52	0.79	45.7	0.48	0.72	44.6	0.5	0.76	**45.4**
Flocking-rand	**0.48**	1.74	45.4	0.51	1.08	45.3	**0.55**	**0.7**	**45.8**	**0.52**	0.77	**45.0**	**0.52**	1.07	**45.4**
Flocking-prop	**0.48**	0.7	45.7	**0.52**	0.72	44.9	0.52	0.73	45.0	0.48	**0.71**	43.3	0.5	0.72	44.7
Flocking-TLP	0.39	**0.61**	21.5	0.43	**0.71**	29.6	0.43	0.76	42.5	0.37	0.72	42.2	0.4	**0.7**	34.0

[*]Initial latency and buffer upper bound are shown after each group index. For example, initial latency and buffer upper bound for group 1 are 3s and 2s. TOR is the tile overlap ratio defined in Sec. VI-A. Freeze represents the time of video freeze/stall in seconds. Quality (WS-PSNR-FOV) is the weighted average WS-PSNR of all the tiles within user's FoV for all the frames in dB.

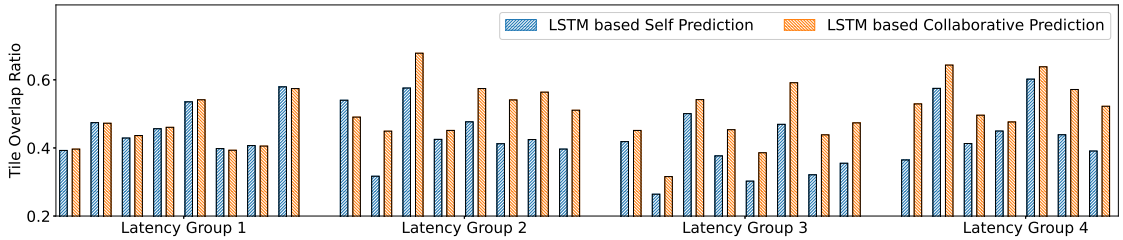


Fig. 10. Tile overlap ratio comparison between self prediction and collaborative prediction based on LSTM$_s$ and LSTM$_c$ respectively.

performs better than or equally to self-prediction (except for one user in latency group 2), even for the users with short latency, which suggests that leveraging other users' attention distributions is always beneficial.

In addition, in Table II, comparing the Non-Flocking and Flocking-rand systems, using LSTM$_s$ and LSTM$_c$, respectively, we see that collaborative prediction improves the TOR for all latency groups, leading to 18.2% higher overall TOR than Non-Flocking (using self prediction). Consequently, the overall averaged quality is also improved by 1 dB. The results demonstrate that, FoV prediction accuracy and video quality can be significantly improved by adopting collaborative predicting. Because both of these systems assigned the users randomly into the latency groups, compared to the proposed Flocking system, both suffer from higher freeze for users in groups with short latency/buffer upper bounds.

2) Benefit from latency and buffer upper bound assignment: For the network condition based latency and buffer upper bound assignment, the relative standard deviation (RSD) of user's bandwidth is calculated. Through comparing it with the predefined RSD thresholds, a user's network condition is classified so that the user can be assigned to a latency group accordingly. In order to make a fair comparison, the bandwidth traces are pre-selected to guarantee that, even with network aware latency group assignment, the number of users assigned into each latency group is the same [4].

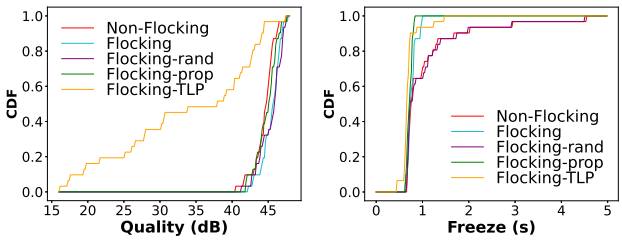
Compared to Flocking-rand (with random latency and buffer upper bound assignment), the Flocking system (with network aware assignment) can significantly reduce the freeze duration for the first two groups with short latencies without degrading the TOR and more importantly the quality in all users. For example, the overall average freeze is reduced from 1.07s to 0.76s over the entire streaming period of 200s, while the average rendering quality remained the same. So, we can draw the conclusion that both the individual and overall user experience can be improved with appropriate latency and buffer upper bound assignment.

3) Benefit from quality-optimized rate allocation: As described in Sec. V-B, given a predicted tile attention distribution and a bandwidth budget, the rate allocation among tiles should be optimized so that the overall quality can be maximized. To demonstrate the effectiveness of such quality-optimized rate allocation, we contrast the performance of the Flocking system with the Flocking-prop system, which assign rate to tiles proportional to the predicted viewing probability, rather than using the optimized rate allocation. We find that, compared to Flocking-prop, even with the same FoV prediction accuracy, Flocking can achieve 0.7 dB higher quality. For each latency group, quality-optimized rate allocation always leads higher video quality.

4) Heuristic vs. LSTM based FoV prediction: Finally, we evaluate the gain from using LSTM-based FoV prediction vs. heuristic prediction by comparing the Flocking and Flocking-TLP systems. As shown in Table II, we find that the overall average TOR is improved dramatically by adopting LSTM based FoV prediction for all latency groups. Note that even though TLP/CO-TLP has better prediction for prediction horizon up to 2 seconds, as shown in Fig 7, Group 1 users tend to reach the buffer upper bound of 2 seconds to minimize video freeze, leading to a prediction horizon of 3 seconds, for which LSTM$_s$ or LSTM$_c$ outperform than TLP or Co-TLP, respectively. Other latency groups will have even longer

[4]For all the experiments, the same set of selected FoV and bandwidth traces are utilized.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3149642, IEEE Transactions on Multimedia



(a) Video Quality (b) Freeze Time

Fig. 11. CDF of quality and freeze time for all the 31 users.

prediction horizons and can greatly benefit from LSTM$_c$ compared to Co-TLP. Overall, Flocking with LSTM based prediction achieves consistent gain in TOR than Flocking-TLP, leading to significantly higher quality.

5) Overall Comparison: The CDF of video quality and freeze time of all the 31 users are illustrated in Fig. 11. Fig. 11(a) shows that all the streaming systems using LSTM-based FoV prediction achieve much higher quality than Flocking-TLP which adopts TLP or Co-TLP. This demonstrates the effectiveness of the proposed LSTM based FoV prediction. While zooming into the range between 40-45 dB, Flocking can achieve higher quality than Non-Flocking (due to collaborative prediction) and Flocking-prop (due to optimized rate allocation). In addition, as shown in Fig. 11(b), Flocking can also stream with the lowest freeze together with Flocking-TLP and Flocking-prop. For example all users in the Flocking system have a freeze duration less than 1.2 seconds. On the contrary, close to 40% of users in the Non-Flocking and Flocking-rand systems suffer from freeze duration greater than 1.2 seconds, and over 10% users have freeze longer than 2 sec. These are likely users who have poor network conditions but are randomly assigned short latencies and buffer upper bounds. Overall, the proposed Flocking strategy which integrates LSTM based collaborative FoV prediction, network aware latency and buffer upper bound assignment and optimized rate allocation, can achieve the highest quality with minimal freeze time.

In practice, the flocking concept works for any live streaming system as long as there exists "workable" latency gap between the front and rear users. Meanwhile, for ultra-low latency live streaming, due to its very short streaming buffer, it will be more challenging to accurately control user playback latency to form a stable (latency-based) flocking structure and optimize the overall flock performance, e.g., maximizing the chance to conduct collaborative FoV prediction for the rear users and reducing the freeze time for the front users. Flocking for ultra-low and sub-second services will be an interesting topic for future study.

6) Superiority over benchmarks: We compare the performance of our proposed Flocking system with two benchmarks, uniform viewport quality (UVP) [43] and hierarchical resolution degrading (HRD) [7]. Both systems employ tile-based streaming. UVP divides the tiles into the viewport and non-viewport regions. The lowest bitrate is selected for all the non-viewport tiles. For the viewport tiles, the bitrate is uniformly increased based on the available bandwidth budget. In HRD system, titles are grouped into three regions: viewport, viewport surrounding, and non-viewport. The bitrate is assigned based on the priority of each region.

TABLE III
QoE Metrics Comparison with Benchmarks.

Systems	TOR	Freeze (s)	Quality (dB)
UVP	0.4	**0.66**	34.0
HRD	0.41	0.76	35.4
UVP with LSTM$_c$	**0.5**	0.74	44.5
HRD with LSTM$_c$	**0.5**	0.88	45.1
Flocking	**0.5**	0.76	**45.4**

Detailed QoE metrics of all the benchmarks and Flocking system are shown in Table III. Both UVP and HRD system adopt TLP based FoV prediction. In addition, to prove the efficiency of LSTM based collaborative FoV prediction, we enhance the benchmarks by replacing the TLP with the LSTM$_c$ attention prediction model while keeping all other parts identical to the original settings. The experiment settings are identical to those in Sec. VI-C. The results show that UVP and HRD with TLP based FoV prediction achieve about 0.4 TOR leading to just 34.0 dB and 35.4 dB quality in terms of WS-PSNR-FOV, which is the weighted average WS-PSNR of all tiles within user FoV. If the FoV prediction method is replaced by LSTM$_c$, TOR can be improved significantly to 0.5, which results in much higher quality to 44.5 dB and 45.1 dB for UVP and HRD. With the same FoV prediction strategy, Flocking still outperforms all the benchmarks by achieving 45.4 dB quality. UVP performs the best in terms of the video freeze time. All the other strategies result in roughly the same stall time except HRD with LSTM$_c$, which suffers the longest freeze time of 0.88 s. Overall, Flocking achieves the best performance among all the benchmarks by generating the highest video quality and moderate video freeze time.

VII. Conclusion

In this paper, we demonstrated that the idea of "flocking" can be used to improve the efficiency of live 360° video streaming from the aspects of both FoV prediction and video freeze avoidance. By assigning users to different latency groups and making use of the actual FoV attention distributions of the front users who have watched the same video segment, the FoV prediction accuracy for a latter user can be improved, leading to a significant increase of the video quality. The proposed LSTM based collaborative FoV prediction algorithms are also shown to improve the FoV prediction accuracy, especially for long prediction intervals, compared with heuristic prediction, leading to an overall quality gain of 10 dB, compared to using the TLP-based approach in our prior work [17]. In addition, the optimized rate allocation can increase the overall video rendering quality by 0.7 dB compared with simply assigning the rate proportional to the predicted viewing probability. Furthermore, by assigning users into groups with different latencies and buffer upper bounds based on their network conditions, the seemly conflicting goals

of low video freeze ratio (requiring long streaming buffer) and high FoV prediction accuracy (requiring short streaming buffer) on individual users can be simultaneously achieved, improving QoE for all users. The flocking idea can also benefit content caching and transcoding, as revealed in our preliminary work in [17]. We will further explore the gain of flocking-based 360° streaming in our future work.

ACKNOWLEDGEMENT

The project was partially supported by USA NSF under award number CNS-1816500.

REFERENCES

[1] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," in *2017 IEEE international conference on communications (ICC)*. IEEE, 2017, pp. 1–7.

[2] C. Ozcinar, A. De Abreu, and A. Smolic, "Viewport-aware adaptive 360 video streaming using tiles for virtual reality," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 2174–2178.

[3] A. Yaqoob and G.-M. Muntean, "A combined field-of-view prediction-assisted viewport adaptive delivery scheme for 360 videos," *IEEE Transactions on Broadcasting*, vol. 67, no. 3, pp. 746–760, 2021.

[4] A. T. Nasrabadi, A. Samiei, and R. Prakash, "Viewport prediction for 360 videos: a clustering approach," in *Proceedings of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2020, pp. 34–39.

[5] C. Li, W. Zhang, Y. Liu, and Y. Wang, "Very long term field of view prediction for 360-degree video streaming," in *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 2019, pp. 297–302.

[6] MUX. (2019) The low latency live streaming landscape in 2019. [Online]. Available: https://mux.com/blog/the-low-latency-live-streaming-landscape-in-2019/

[7] M. Hosseini and V. Swaminathan, "Adaptive 360 vr video streaming: Divide and conquer," in *2016 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2016, pp. 107–110.

[8] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*. ACM, 2016, pp. 1–6.

[9] L. Sun, F. Duanmu, Y. Liu, Y. Wang, Y. Ye, H. Shi, and D. Dai, "Multi-path multi-tier 360-degree video streaming in 5g networks," in *Proceedings of the 9th ACM Multimedia Systems Conference*. ACM, 2018, pp. 162–173.

[10] L. Sun, F. Duanmu, Y. Liu, Y. Wang, Y. Ye, H. Shi, and D. Dai, "A two-tier system for on-demand streaming of 360 degree video over dynamic networks," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 43–57, 2019.

[11] Z. Jiang, X. Zhang, W. Huang, H. Chen, Y. Xu, J.-N. Hwang, Z. Ma, and J. Sun, "A hierarchical buffer management approach to rate adaptation for 360-degree video streaming," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2157–2170, 2019.

[12] Y. Guan, C. Zheng, X. Zhang, Z. Guo, and J. Jiang, "Pano: Optimizing 360 video streaming with a better understanding of quality perception," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2019, pp. 394–407.

[13] X. Hou, S. Dey, J. Zhang, and M. Budagavi, "Predictive adaptive streaming to enable mobile 360-degree and vr experiences," *IEEE Transactions on Multimedia*, vol. 23, pp. 716–731, 2020.

[14] M. Qiao, M. Xu, Z. Wang, and A. Borji, "Viewport-dependent saliency prediction in 360 video," *IEEE Transactions on Multimedia*, vol. 23, pp. 748–760, 2020.

[15] R. Silva, B. Feijó, P. B. Gomes, T. Frensh, and D. Monteiro, "Real time 360 video stitching and streaming," in *ACM SIGGRAPH 2016 Posters*. ACM, 2016, p. 70.

[16] C. Griwodz, M. Jeppsson, H. Espeland, T. Kupka, R. Langseth, A. Petlund, P. Qiaoqiao, C. Xue, K. Pogorelov, M. Riegler *et al.*, "Efficient live and on-demand tiled hevc 360 vr video streaming," in *2018 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2018, pp. 81–88.

[17] L. Sun, Y. Mao, T. Zong, Y. Liu, and Y. Wang, "Flocking-based live streaming of 360-degree video," in *Proceedings of the 11th ACM Multimedia Systems Conference*, 2020, pp. 26–37.

[18] X. Xie and X. Zhang, "Poi360: Panoramic mobile video telephony over lte cellular networks," in *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*. ACM, 2017, pp. 336–349.

[19] M. Berning, T. Yonezawa, T. Riedel, J. Nakazawa, M. Beigl, and H. Tokuda, "parnorama: 360 degree interactive video for augmented reality prototyping," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, 2013, pp. 1471–1474.

[20] Y. Mao, L. Sun, Y. Liu, and Y. Wang, "Low-latency fov-adaptive coding and streaming for interactive 360 video streaming," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 3696–3704.

[21] Y. Bao, H. Wu, T. Zhang, A. A. Ramli, and X. Liu, "Shooting a moving target: Motion-prediction-based transmission for 360-degree videos," in *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 2016, pp. 1161–1170.

[22] C.-L. Fan, J. Lee, W.-C. Lo, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "Fixation prediction for 360 video streaming in head-mounted virtual reality," in *Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, 2017, pp. 67–72.

[23] V. R. Gaddam, M. Riegler, R. Eg, C. Griwodz, and P. Halvorsen, "Tiling in interactive panoramic video: Approaches and evaluation," *IEEE Transactions on Multimedia*, vol. 18, no. 9, pp. 1819–1831, 2016.

[24] G. Cheung, Z. Liu, Z. Ma, and J. Z. Tan, "Multi-stream switching for interactive virtual reality video streaming," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 2179–2183.

[25] Y. Ban, L. Xie, Z. Xu, X. Zhang, Z. Guo, and Y. Wang, "Cub360: Exploiting cross-users behaviors for viewport prediction in 360 video adaptive streaming," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2018, pp. 1–6.

[26] P. K. Yadav and W. T. Ooi, "Tile rate allocation for 360-degree tiled adaptive video streaming," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 3724–3733.

[27] J. Le Feuvre and C. Concolato, "Tiled-based adaptive streaming using mpeg-dash," in *Proceedings of the 7th International Conference on Multimedia Systems*, 2016, pp. 1–3.

[28] M. Orduna, C. Díaz, L. Muñoz, P. Pérez, I. Benito, and N. García, "Video multimethod assessment fusion (vmaf) on 360vr contents," *IEEE Transactions on Consumer Electronics*, vol. 66, no. 1, pp. 22–31, 2019.

[29] X. Liu, B. Han, F. Qian, and M. Varvello, "Lime: understanding commercial 360 live video streaming services," in *Proceedings of the 10th ACM Multimedia Systems Conference*. ACM, 2019, pp. 154–164.

[30] P. R. Alface, J.-F. Macq, and N. Verzijp, "Interactive omnidirectional video delivery: A bandwidth-effective approach," *Bell Labs Technical Journal*, vol. 16, no. 4, pp. 135–147, 2012.

[31] A. Zare, A. Aminlou, M. M. Hannuksela, and M. Gabbouj, "Hevc-compliant tile-based streaming of panoramic video for virtual reality applications," in *Proceedings of the 24th ACM international conference on Multimedia*. ACM, 2016, pp. 601–605.

[32] R. Skupin, Y. Sanchez, C. Hellge, and T. Schierl, "Tile based hevc video for head mounted displays," in *2016 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2016, pp. 399–400.

[33] A. T. Nasrabadi, A. Mahzari, J. D. Beshay, and R. Prakash, "Adaptive 360-degree video streaming using scalable video coding," in *Proceedings of the 25th ACM international conference on Multimedia*. ACM, 2017, pp. 1689–1697.

[34] A. TaghaviNasrabadi, A. Mahzari, J. D. Beshay, and R. Prakash, "Adaptive 360-degree video streaming using layered video coding," in *2017 IEEE Virtual Reality (VR)*. IEEE, 2017, pp. 347–348.

[35] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[36] L. Sun, T. Zong, Y. Liu, Y. Wang, and H. Zhu, "Optimal strategies for live video streaming in the low-latency regime," in *2019 IEEE 27th International Conference on Network Protocols (ICNP)*. IEEE, 2019, pp. 1–4.

[37] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. ACM, 2017, pp. 197–210.

[38] L. Sun, T. Zong, S. Wang, Y. Liu, and Y. Wang, "Towards optimal low-latency live video streaming," *IEEE/ACM Transactions on Networking*, 2021.

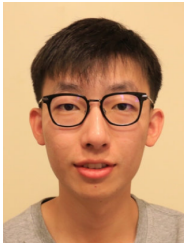
[39] L. Sun, T. Zong, S. Wang, Y. Liu, and Y. Wang, "Tightrope walking in low-latency live streaming: Optimal joint adaptation of video rate and playback speed," in *Proceedings of the 12th ACM Multimedia Systems Conference*, ser. MMSys '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 200–213. [Online]. Available: https://doi.org/10.1145/3458305.3463382

[40] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan, "Beyond throughput, the next generation: a 5g dataset with channel and context metrics," in *Proceedings of the 11th ACM Multimedia Systems Conference*, 2020, pp. 303–308.

[41] Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu, and S. Gao, "Gaze prediction in dynamic 360 immersive videos," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5333–5342.

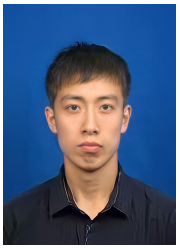
[42] I. Kim, K. McCann, K. Sugimoto, B. Bross, W. Han, and G. Sullivan, "High efficiency video coding (hevc) test model 14 (hm 14) encoder description. document: Jctvc-p1002," *JCT-VC, Jan*, 2014.

[43] J. V. d. Hooft, M. T. Vega, S. Petrangeli, T. Wauters, and F. D. Turck, "Tile-based adaptive streaming for virtual reality video," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 15, no. 4, pp. 1–24, 2019.



Liyang Sun received the B.E. in Optical and Electronic Information Engineering from Huazhong University of Science and Technology, Wuhan, China, in 2013, and the M.S. in Computer Engineering and Ph.D. in Electrical Engineering from New York University, NY, USA, in 2016 and 2021, respectively. He joined ByteDance as a video algorithm engineer in 2021. His research interests include 360 degree video streaming, live video streaming, reinforcement learning and multipath technologies.

Yixiang Mao received the B.S. degree in physics from Peking University, Beijing, China, in 2016. He is currently working toward the Ph.D. degree in electrical engineering at Tandon School of Engineering, New York University. His research interests include immersive media (360 video and point cloud) coding and streaming, video processing, computer vision, and machine learning.



Tongyu Zong is a Ph.D. candidate in Electrical and Engineering at New York University, NY, USA, from 2018. He received the B.S. and the M.E. in Microelectronics from Fudan University, Shanghai, China, in 2016 and 2018, respectively. His research interests include edge caching, video streaming and reinforcement learning.

Yong Liu is a professor at the Electrical and Computer Engineering department of New York University. He joined NYU-Poly as an assistant professor in March, 2005. He received his Ph.D. degree from Electrical and Computer Engineering department at the University of Massachusetts, Amherst, in May 2002. He received his master and bachelor degree in the field of automatic control from the University of Science and Technology of China, in July 1997 and 1994 respectively. His general research interests lie in modeling, design and analysis of communication networks. His current research include multimedia networking, network measurement, online social networks, and recommender systems. He is the winner of ACM/USENIX Internet Measurement Conference (IMC) Best Paper Award in 2012, IEEE Conference on Computer and Communications (INFOCOM) Best Paper Award in 2009, and IEEE Communications Society Best Paper Award in Multimedia Communications in 2008. He is a Fellow of IEEE.

Yao Wang received the B.S. and M.S. in Electronic Engineering from Tsinghua University, Beijing, China, in 1983 and 1985, respectively, and the Ph.D. degree in Electrical and Computer Engineering from University of California at Santa Barbara in 1990. Since 1990, she has been on the faculty of Electrical and Computer Engineering, Tandon School of Engineering of New York University (formerly Polytechnic University, Brooklyn, NY). Her current research areas include video communications, multimedia signal processing, and medical imaging. She is the leading author of a textbook titled "Video Processing and Communications", and has published over 250 papers in journals and conference proceedings. She has served as an Associate Editor for IEEE Transactions on Multimedia and IEEE Transactions on Circuits and Systems for Video Technology. She received New York City Mayor's Award for Excellence in Science and Technology in the Young Investigator Category in year 2000. She was elected Fellow of the IEEE in 2004 for contributions to video processing and communications. She is also a co-winner of the IEEE Communications Society Leonard G. Abraham Prize Paper Award in the Field of Communications Systems in 2004, and a co-winner of the IEEE Communications Society Multimedia Communication Technical Committee Best Paper Award in 2011. She was a keynote speaker at the 2010 International Packet Video Workshop.