A Comparison of AI Models for Viewport Prediction in Immersive 360° Video Streaming

Muhammad Farooq[1], Gioacchino Manfredi[1], Maria Martini[2], Saverio Mascolo[1], Luca De Cicco[1]

***Abstract*— Streaming of 360° videos has gained popularity in recent years due to its immersive viewing experience and the diffusion of inexpensive Head Mounted Displays (HMDs). Such videos allow users consuming the content through an HMD to freely change the point of view of a scene by simply turning their head. Streaming these videos over the Internet is challenging due to bandwidth constraints and latency requirements. Only roughly 1/6th of the whole omnidirectional scene falls in the field of view – or viewport – of the user. Therefore, efficient viewport prediction plays a key role as it helps identifying user's regions of interest for both short-term and long-term periods. However, as the prediction horizon increases, the accuracy of viewport prediction tends to decrease. This paper presents a comparative analysis of Machine Learning (ML) models for viewport prediction, using a public dataset. Results show that with the selected configurations the Convolutional Neural Network (CNN) model performs better than all other models and achieves a viewport prediction accuracy of 93.93% for the next frame, while the Long Short-Term Memory (LSTM) model performs better when the viewport prediction horizon is increased up to 5 seconds.**

I. Introduction

In recent years, Virtual Reality (VR) has gained attention due to relevant improvements in Head-Mounted Display (HMD) and the opportunity to provide an immersive experience to users [1]. 360° or omnidirectional videos provide the user with a complete, uninterrupted visualization of content in all directions. Specialized cameras and software are used to capture video from multiple angles and seamlessly stitch the footage into a smooth 360° view. This process enables an immersive experience, allowing users to explore and interact with virtual environments in real time. These videos are used for instance in sports, entertainment and landscape to create a more realistic environment. Several projections have been proposed for mapping these videos on a 2D plane, such as Equirectangular Projection (ERP), pyramid and Cube-map Projection (CMP). Encoders such as H.264 Advanced Video Coding (AVC) and H.265 High-Efficiency Video Coding (HEVC) are typically used to compress them while maintaining their quality. While capturing 360° videos requires specialized equipment, advancements in technology have made it easy and seamless to view them on platforms such as mobile devices, desktops, and VR headsets [2]. In the context of 360° video streaming the term *viewport* is used to define the specific part of the video that is currently rendered in the Field of View (FoV) of the user. Two main approaches have been proposed in the literature to stream 360° videos: 1) *Viewport-independent* solutions stream the complete 360° video content to users, which results in high bandwidth requirements; 2) *Viewport-adaptive* solutions provide high resolution to the portion of video falling in the viewport, which results in reduced required bandwidth.

Tile-based video streaming is widely adopted in state-of-the-art approaches for delivering 360° video over the Internet. As the user's viewport contains a part of the 360° video, using this approach saves bandwidth and delivers high-resolution content in correspondence of the user's viewport tiles. Optimizing bandwidth usage and minimizing latency are key challenges in streaming 360° videos. Particularly, viewport prediction plays an important role in efficient tile-based video streaming.

Different machine learning and reinforcement learning approaches have been used for viewport prediction [3]–[5]. This work aims at providing a unified empirical comparison of various ML models under the same dataset, preprocessing, prediction horizons, tile-accuracy and complexity evaluation. Such a comparison is carried out using a public dataset and considering different time horizons for the prediction. Among the considered approaches, the one achieving the best prediction accuracy for the next frame is the CNN (93.93%), which outperforms the other tested ML models. Beside next frame prediction, we evaluated viewport prediction accuracy for different time horizons (1, 3, and 5 seconds). To ensure fairness of the comparison, the same hyperparameters were applied to all sequential models (LSTM, Recurrent Neural Network (RNN) and Gated Recurrent Unit (GRU)). We also examined the performance of non sequential models including Linear Regression (LR) and Multi-Layer Perceptron (MLP) along with CNN. Beyond accuracy, we also consider the computational complexity and real-time suitability of models, which provides a balanced perspective on both predictive performance and practical deployment. As explained later on, Figure 1 shows the proposed viewport prediction workflow adopted in this study.

II. Related Work

The growing interest in 360° video streaming has driven significant research efforts towards improving user experience while reducing bandwidth consumption [6], [7]. Unlike traditional video streaming, users only observe a limited portion of the panoramic video at any given time through their

This work has been partially supported by Regione Puglia through "Bando Reti" under the project ARIANNA, CUP B97H24001880007.

[1]Muhammad Farooq, Gioacchino Manfredi, Saverio Mascolo, Luca De Cicco are with the Department of Electrical and Information Engineering, Politecnico di Bari, Bari, Italy. E-mails: m.farooq1@phd.poliba.it, gioacchino.manfredi@poliba.it, saverio.mascolo@poliba.it, luca.decicco@poliba.it

[2]Maria Martini is with the Department of Engineering, Kingston University London, London, UK. E-mail: m.martini@kingston.ac.uk

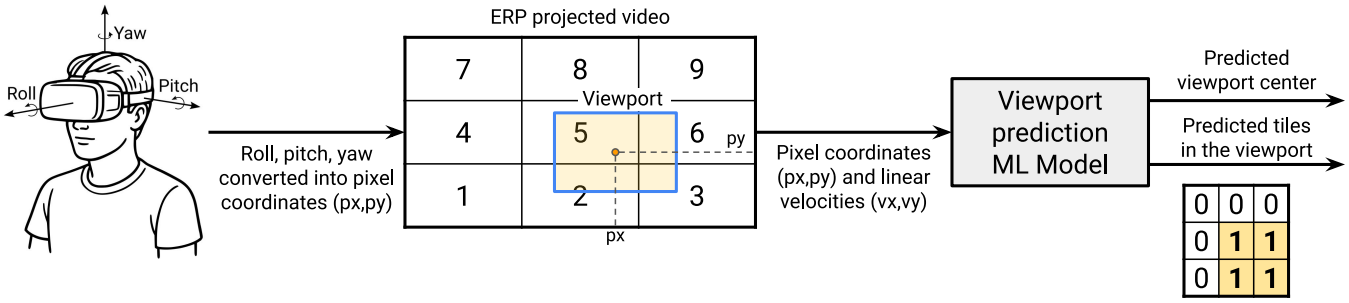

Fig. 1: Workflow for Viewport Prediction Using ML Models.

device or head-mounted display. As a result, transmitting the entire 360° frame at high quality leads to inefficient use of network resources [8]. Viewport prediction, whose goal is to forecast the specific viewing region of the 360° video displayed on the user device, plays a key role in addressing this challenge. By predicting the user's future viewing direction, streaming systems can allocate higher quality to the predicted viewport while transmitting the remaining regions at lower quality, hence improving bandwidth efficiency and overall streaming performance [3], [4].

Recent studies have proposed a variety of viewport prediction approaches based on machine learning and deep learning models to improve prediction accuracy and enhance Quality of Experience (QoE) in immersive video streaming systems [3], [4]. Several works incorporate video content information such as visual saliency, scene dynamics or motion features to better capture the regions that are most likely to attract users' attention [9], [10]. Other approaches combine multiple sources of information including user motion traces and visual content features to improve prediction robustness and accuracy in dynamic viewing scenes [11]. Studies focus on advanced architectures such as attention-based models, lightweight neural networks and transformer-based frameworks, which have been explored to capture complex temporal dependencies and improve viewport prediction performance in real-time streaming systems [12], [13].

Data used for predicting user viewing area is typically derived from three commonly used categories: 1) user head movement data (trajectory-based), which contains information about user pitch, yaw and roll angles that describe the viewing direction over time; 2) video content information, which analyzes visual features of the video to determine regions that may attract user attention; 3) a combination of both user head movement and video content features to improve prediction performance.

A. *Trajectory-based viewport prediction*

This includes recording head movement data such as pitch, yaw, and roll angles [14] while the user is watching the video. Then, the data gathered for a certain number of users is processed to apply a prediction strategy. In [15], authors report improved viewport prediction accuracy using a clustering-based approach. However, this study mainly focuses on short-term head movement prediction to evaluate prediction horizon up to 2 seconds. Transformers, i.e. attention-based deep learning models designed to capture long-range dependencies in sequential data, are also employed for viewport prediction by modeling complex temporal relationships through the attention mechanism [16], [17].

State-of-the-art studies on viewport prediction investigate different prediction horizons, ranging from short-term predictions for the next frame or a few seconds ahead to longer horizons of up to 10 seconds [18], [19].

B. *Content-based viewport prediction*

Content-based viewport prediction relies on visual characteristics of the video to estimate the regions that users are most likely to watch. These approaches typically use saliency maps that highlight visually prominent areas in the frame and help to identify regions that naturally attract human attention. In addition to saliency information, these methods also consider dynamic scenes, object motion and motion vectors extracted from the video stream to capture temporal changes in visual importance. Analyzing these visual features, models can infer potential user viewing behavior even without extensive historical user trajectory information. Such techniques are especially useful when combined with user behavior data as they provide complementary information about where users are likely to focus within immersive 360° environment [9].

C. *Hybrid Approaches*

These approaches combine multiple models such as LSTM encoder–decoder architectures, LSTM combined with CNN or multi-fusion transformer-based models. In most cases, the primary input consists of the user's historical viewport trajectory (e.g., head movement or pitch and yaw angles). In addition to this core information, hybrid approaches often integrate additional modalities such as saliency maps, video frame features or semantic content extracted from the video to further improve prediction accuracy.

To address the challenges in viewport prediction accuracy, [20] combines HMD data with video contents. However, this model struggles to maintain accuracy in diverse and dynamic scenarios. Similarly, in [21] an HMD dataset with saliency maps is used, reporting improved accuracy, though their federated learning approach remains impractical for resource-constrained devices. Live VR streaming [22] faces major

challenges due to its high bandwidth demands and the lack of historical user data, making it difficult to scale effectively. To tackle this issue, a CNN-based viewport prediction model [23] was designed with a simpler architecture to better handle real-time streaming. However, the model's accuracy drops significantly when dealing with videos that have complex, dynamic scenes or longer durations.

To actually implement viewport-based adaptive streaming techniques with ERP, it must be possible to separate the portion of the scene watched by the user from the rest, to be sent at a lower resolution. To this purpose, every frame of the projected 2D video is divided into smaller portions so as to easily identify the part of the scene the user is interested in. The most widely adopted technique is called *tiling* [24], and consists of breaking down frames into smaller frames.

Previous comparative studies relied on simple hyperparameter settings for each model and did not consider model complexity or real-time suitability [25]. Moreover, their evaluation focuses on predicting viewing *tiles* instead of directly estimating the exact viewport center coordinates (e.g., pitch and yaw angles), as they provide a more precise representation of the user's viewing direction.

The comparative analysis presented in this paper maintains the computational complexity characteristics of each sequential model while it also includes non-sequential models, which enables a more comprehensive and balanced comparison across different model architectures. By considering both categories of models, this study provides a clearer understanding of how different learning approaches perform in predicting user viewing area tasks. In addition, the analysis investigates the accuracy of predicting the tiles that fall within the viewport as well as the prediction of the viewport center for each model. This allows the evaluation to capture both the spatial accuracy of tile-based predictions and the precision of the estimated viewport position. Therefore, performance of each model is assessed using both prediction accuracy and error distribution metrics, which provides a detailed comparison of their strengths and limitations.

III. System Model

A. Dataset description

We consider a publicly available dataset[1] containing 88 omnidirectional videos with viewing histories from an average of 45 users. User motion is represented either as Euler angles (pitch, yaw) or unit quaternions, with roll set to zero. Dataset consists of six sub-datasets, each originating from a different study reported in the corresponding articles [6]–[8], [26]–[28]. Among them, datasets 2, 3 and 5 represent head orientation using Euler angles (pitch, yaw and roll), while the remaining datasets use quaternion representations. Each data file corresponds to a specific video and contains head orientation traces of multiple users. The first line provides the sampling timestamps (in seconds) that represents the time instances at which the head orientation was recorded. The dataset considered is sampled at 10 Hz, therefore data is read at $t = [0, 0.1, 0.2, \ldots]$ seconds.

[1]https://github.com/360VidStr/A-large-dataset-of-360-video-user-behaviour/blob/main/README.md

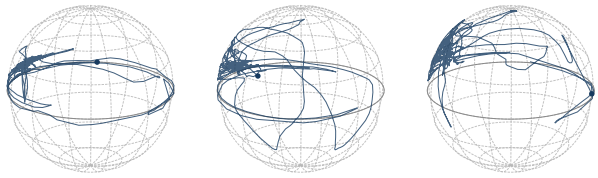

Fig. 2: Head movement trajectories

To further clarify, Figure 2 shows the head movement trajectories associated with three users in the considered dataset watching the same spherical video content. For processing purposes, as explained further on, such trajectories, along with the video, are then projected on the 2D plane leveraging some existing techniques, e.g. ERP.

B. Data preprocessing

As discussed above, the first line of each data file contains timestamps that indicates the time instances at which the head orientation measurements is recorded during video playback. Because the sampling time, i.e. timestamps, is the same for all users, the timestamp line is removed during preprocessing. Therefore, a fixed time step of $\Delta t = 0.1$ seconds is assumed for all users of each video. To process the viewport data in a planar domain, an ERP projection is adopted, which maps the spherical 360° video onto a two–dimensional rectangular frame. This projection allows the panoramic video to be represented on a 2D plane while preserving the angular information of the viewing direction. Video files considered in this study are provided in 8K resolution and viewport size is set to 1920×1080. Pitch and yaw angles describing the user's viewing direction are then converted into pixel coordinates $[p_x, p_y]$, which correspond to the horizontal and vertical positions of the viewport center on the projected 2D video frame. In other words, p_x represents the x-coordinate along the width of the ERP frame, while p_y represents the y-coordinate along the height of the frame. These coordinates indicate the exact location of the viewport center within the panoramic video frame. With positions p_x and p_y over time, horizontal and vertical viewport velocities $[v_x, v_y]$ can be computed as:

$$v_x = \frac{\Delta p_x}{\Delta t}, \quad v_y = \frac{\Delta p_y}{\Delta t}$$

where Δp_x and Δp_y represent the changes in position along the x and y axes, respectively, and Δt is the sampling time. Thus, the features we consider in input to the ML models are p_x, p_y, v_x, v_y.

For tile-based video streaming, each 360° video frame is divided into nine tiles arranged in a 3×3 grid (three rows and three columns). The height and width of each tile are computed as $H_{tile} = H_{video}/3$ and $W_{tile} = W_{video}/3$, respectively. Since the ERP video resolution used in this

study is 7680×3840 (8K), each tile has a width of $7680/3 = 2560$ pixels and a height of $3840/3 = 1280$ pixels.

The user's FoV may contain 1, 2 or up to 4 tiles depending on the head orientation while watching the 360° video. As the user changes their viewing direction, viewport may shift across tile boundaries, which causes multiple tiles to fall within the visible region. In such cases, all tiles that overlap within the FoV must be transmitted and rendered to ensure a seamless viewing experience. If the FoV covers even a small portion of a tile, that tile must be encoded and delivered at high resolution to avoid visible quality degradation within the user's viewport.

C. ML Models for viewport prediction

Various ML models are considered in this study to assess their performance for both short and long-term viewport prediction accuracy, with prediction horizons up to 5 seconds. The objective is to analyze how different model architectures perform when predicting the user's future viewing direction in immersive 360° video environments. In particular, these models are evaluated in terms of their ability to accurately predict the viewport center and the tiles that fall within the user's FoV. Figure 1 illustrates the overall system model for the tile-based viewport prediction. Hyperparameters follow common viewport-prediction settings [25]. All models use sigmoid output activation for multi-label tile prediction; CNN layers use ReLU while recurrent models use standard internal non-linearities.

1) ***LSTM Model:*** LSTM model is particularly suitable for learning temporal dependencies in sequential data, making it effective for predicting user viewing behaviour over time. The input sequence length for the LSTM model is set to 30 frames for next frame and 1-second FoV prediction. This configuration allows the model to learn the temporal relationship between recent head movement patterns and the immediate future viewport position. For longer prediction horizons, the sequence length is increased to capture a larger history of user motion. In particular, for 3-second and 5-second FoV predictions, the sequence length is set to 90 frames for 3-second prediction and 150 frames for 5-second prediction. This setting enables the model to use sufficient historical information for better estimation of future viewport positions over longer time intervals.

The LSTM model contains 3 layers with 200, 100 and 50 neurons, respectively. In the case of next frame or 1-second FoV prediction, as there are 4 features, i.e. pixel coordinates (p_x, p_y) and velocity (v_x, v_y) with sequence length 30, the input shape given to this model is (sequence length, number of features), i.e. (30,4). A dropout rate layer of 0.5 with batch-normalization layer are applied with a sigmoid activation function in the final layer.

2) ***RNN:*** RNN is used with three hidden layers of 200, 100 and 50 neurons, respectively. A learning rate of 0.0001 is used during training and a sigmoid activation function is applied at the output layer of the model. This model is trained for a maximum of 150 epochs with an early stopping condition implemented using a patience value of 10 epochs to prevent overfitting. A batch size of 32 is used during training and the Binary Cross Entropy (BCE) loss function is employed to optimize the model parameters.

3) ***GRU:*** GRU model is implemented using the same hyper-parameter configuration adopted for the LSTM and RNN models. In particular, the learning rate, batch size, number of training epochs and the early stopping patience level are kept identical to those used in the previous models. This allows a fair comparison among the different recurrent architectures while isolating the effect of the model structure on the viewport prediction performance.

4) ***CNN model:*** The first layer of this CNN model contains 1D convolutions with 16 filters, along with Rectified Linear Unit (RELU) activation function to add some non-linearity. The input to the CNN is defined as (batch-size,1,features), with the middle dimension representing a one-dimensional image. To reduce the feature map size, a max-pooling layer is employed. The second CNN layer is characterized by a filter size equal to 32 followed by the same RELU activation. An adaptive average pooling layer is used for fixed size output and the final layer is fully connected. Learning rate, number of epochs, loss function, batch size and early stop patience level are kept the same as in LSTM, RNN and GRU models.

5) ***LR model:*** Following [25], LR model is used as a lightweight feed-forward baseline with fully connected layers of 1024, 512 and 256 units, rather than as a classical single-layer linear regression model. It is trained using an Adam optimizer with BCE loss up to 150 epochs with a batch size of 32 and early stopping patience set to 10.

6) ***MLP model:*** MLP model used for this study consists of three fully connected (linear) layers with 200, 100 and 50 neurons, respectively. A sigmoid activation function is applied at the output layer. This model is then trained for up to 150 epochs with an early stopping mechanism, where the patience level is set to 5 epochs to avoid overfitting. The BCE loss function is used during training. CNN model achieves the best performance for next-frame viewport prediction, while the GRU model performs better for longer prediction horizons of up to 5 seconds as discussed in the following sections.

IV. Results and Discussion

Accuracy is measured by comparing the tiles actually viewed by the user with those predicted by the ML model as follows:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Predictions}} \tag{1}$$

Among the available motion traces in the dataset, videos with indices 10, 14, 55 and 58 demonstrated higher viewport prediction accuracy as compared to the remaining traces during the initial evaluation. These users motion traces performed more consistent head movement patterns, which made them more suitable for reliable viewport prediction analysis. In this study, we selected motion trace 55 as the primary dataset for further experimentation as it contains viewing data from 48 users, which provides a sufficiently

TABLE I: Complexity and Real-Time Performance of Different Models

Model	Params (#)	Model Size (MB)	Latency (ms)	Throughput (seq/s)	Peak GPU Memory (MB)	Real-Time @30fps
LSTM	317,312	1.22	8.763	3,711.6 @ batch=64	72.8	Yes
RNN	80,312	0.32	3.950	13,412.6 @ batch=64	71.7	Yes
LR	20,356	0.09	0.654	116,594.0 @ batch=64	68.9	Yes
MLP	26,711	0.11	1.298	39,190.0 @ batch=64	69.4	Yes
CNN	76,562	0.30	2.798	21,164.7 @ batch=64	70.3	Yes
GRU	238,312	0.92	7.052	8,416.9 @ batch=64	81.1	Yes

large number of samples for model training and evaluation. Motion trace 55 was split into 70% training, 10% validation and 20% testing data. A multi-trace evaluation including 10, 14 and 58 traces is left to future work.

Using motion trace 55, the CNN model achieved the best viewport prediction accuracy: 93.93% for next-frame prediction, indicating its strong capability in capturing short-term user viewing behavior.

The Cumulative Distribution Function (CDF) of distortion provides a clearer view of the distribution of prediction errors across all samples. For next-frame viewport prediction, the CDF was computed using 16,685 test samples, with the distortion measures derived from the viewport center error vectors. These error vectors represent the difference between the predicted and ground-truth viewport center positions in terms of their horizontal and vertical coordinates.

Due to space constraints, we report only the curves based on Root Mean Squared Error (RMSE),which reflects the distance between the ground truth and the estimated viewport center:

$$RMSE_j = \sqrt{\frac{1}{2}[(x_j - \hat{x}_j)^2 + (y_j - \hat{y}_j)^2]}.$$

Here, y_j represents the actual test value, $\hat{y}_j$ represents the predicted test value, and j is the index for the different test viewport positions.

Figure 3 shows the CDF plot of RMSE for the considered ML models in the case of next frame viewport prediction. The curve that lies further to the left indicates that the model makes smaller errors more often, which reflects higher prediction accuracy. From the figure it is possible to observe that CNN model achieves the best results compared to the other models, indicating superior regression performance.

Figure 4 represents the accuracy in the tile selection as a consequence of viewport prediction for various ML models at different prediction horizons (next frame, 1, 3, and 5 seconds). It can be pointed out that the LSTM model shows a satisfactory trend in general, proving its effectiveness both for short- and long-term prediction. However, in the case of next frame prediction, the CNN model outperforms all other approaches, though presenting a substantial decrease in accuracy for longer horizons. This happens because CNN captures short-term patterns and has limited temporal memory. On the other hand, the best result for 5-second prediction is achieved by GRU, with 37.93% accuracy.

Finally, Table I summarizes the computational complexity

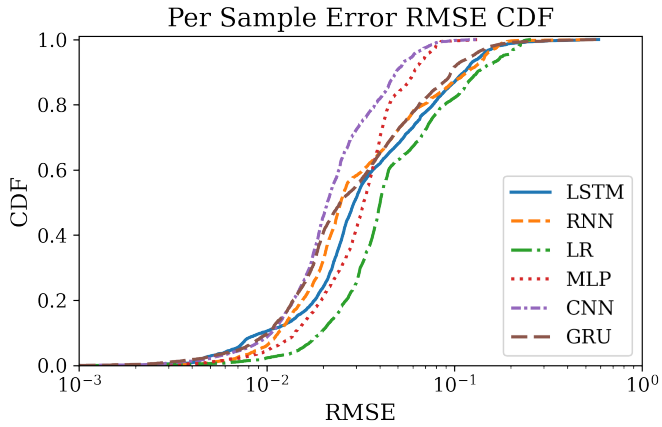

Fig. 3: RMSE CDF obtained by the considered models.

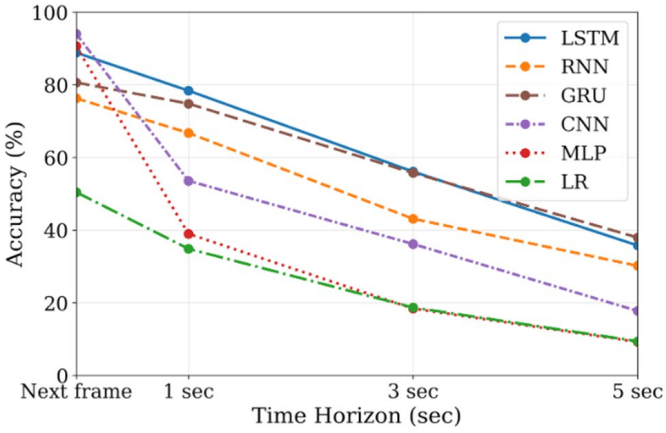

Fig. 4: Comparison of tile selection accuracy across models.

and real-time performance metrics of all evaluated models. This comparison highlights the trade-off between model capacity and computational efficiency. All models satisfy the 30 fps processing requirement, which indicates that they are suitable for practical viewport prediction in 360° video streaming systems. Among them, LR is the lightest model in terms of computational cost, while LSTM and GRU provide higher modeling capacity, as displayed in Figure 4, at the expense of increased latency.

V. Conclusion

Various ML models are tested for viewport prediction at different time horizons: next frame, 1, 3 and 5 seconds with 30 fps. Simulative results show that CNN model performs best at next frame prediction but its accuracy decreases

rapidly as time horizon is increased. On the other hand, sequence based models such as LSTM, RNN, and GRU perform well for long-term predictions as compared to non-sequence based models such as MLP and LR. Future work aims at improving prediction accuracy of these models using saliency maps, which indicate users' interest regions, along with HMD data. Another interesting direction consists of extending to a broader set of video contents in order to analyze how different scene characteristics influence prediction performance. Finally, through transformer-based models, hyperparameter optimization and sequence-aware metrics, the prediction horizon will be extended to 10 seconds to analyze the feasibility and accuracy of long-term viewport prediction in immersive 360° video streaming environments.

REFERENCES

[1] V. H. Nguyen, D. T. Bui, T. L. Tran, C. T. Truong, and T. H. Truong, "Scalable and resilient 360-degree-video adaptive streaming over http/2 against sudden network drops," *Computer Communications*, vol. 216, pp. 1–15, 2024.

[2] B. A. Zulkiewicz, V. Boudewyns, C. Gupta, A. Kirschenbaum, and M. A. Lewis, "Using 360-degree video as a research stimulus in digital health studies: Lessons learned," *JMIR Serious Games*, vol. 8, no. 1, 2020.

[3] X. Chen, B. Cao, I. Ahmad, and X. Xue, "Lightweight neural network-based viewport prediction for live vr streaming in wireless video sensor network," *Mob. Inf. Syst.*, vol. 2021, Jan. 2021.

[4] J. Vielhaben, H. Camalan, W. Samek, and M. Wenzel, "Viewport forecasting in 360° virtual reality videos with machine learning," in *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, 2019, pp. 74–747.

[5] J. Adhuran and M. G. Martini, "Efficient viewport prediction and tiling schemes for 360 degree video streaming," in *Proceedings of the 15th ACM Multimedia Systems Conference*, ser. MMSys '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 374–380.

[6] 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)*, 2016, pp. 1161–1170.

[7] 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*, ser. SIGCOMM '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 394–407. [Online]. Available: https://doi.org/10.1145/3341302.3342063

[8] X. Corbillon, F. De Simone, and G. Simon, "360-degree video head movement dataset," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, ser. MMSys'17. New York, NY, USA: Association for Computing Machinery, 2017, p. 199–204. [Online]. Available: https://doi.org/10.1145/3083187.3083215

[9] Y. Li, H. Wang, H. Liu, and B. Chen, "A study on content-based video recommendation," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 4581–4585.

[10] S. Wang, S. Yang, H. Li, X. Zhang, C. Zhou, C. Xu, F. Qian, N. Wang, and Z. Xu, "Salientvr: saliency-driven mobile 360-degree video streaming with gaze information," in *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, ser. MobiCom '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 542–555. [Online]. Available: https://doi.org/10.1145/3495243.3517018

[11] X. Feng, V. Swaminathan, and S. Wei, "Viewport prediction for live 360-degree mobile video streaming using user-content hybrid motion tracking," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 3, no. 2, Jun. 2019. [Online]. Available: https://doi.org/10.1145/3328914

[12] Z. Pan, Y. Zhang, T. Lin, and J. Yan, "Liveae: Attention-based and edge-assisted viewport prediction for live 360° video streaming," in *Proceedings of the 2023 Workshop on Emerging Multimedia Systems*, ser. EMS '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 28–33. [Online]. Available: https://doi.org/10.1145/3609395.3610597

[13] Y. Tian, Y. Zhong, Y. Han, and F. Chen, "Viewport prediction with cross modal multiscale transformer for 360° video streaming," *Scientific Reports*, vol. 15, no. 1, p. 30346, 2025. [Online]. Available: https://doi.org/10.1038/s41598-025-16011-7

[14] S. Petrangeli, G. Simon, and V. Swaminathan, "Trajectory-based viewport prediction for 360-degree virtual reality videos," in *2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, 2018, pp. 157–160.

[15] 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*, ser. ATC '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1–6.

[16] F.-Y. Chao, C. Ozcinar, and A. Smolic, "Transformer-based long-term viewport prediction in 360° video: Scanpath is all you need," in *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*, 2021, pp. 1–6.

[17] J. Li, Z. Zhao, Q. Li, Z. Li, P. Y. Zhou, Z. Liu, H. Zhou, and Z. Li, "VPFormer: Leveraging Transformer with Voxel Integration for Viewport Prediction in Volumetric Video," *ACM Trans. Multimedia Comput. Commun. Appl.*, Apr. 2025.

[18] C. Wu, R. Zhang, Z. Wang, and L. Sun, "A spherical convolution approach for learning long term viewport prediction in 360 immersive video," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 14 003–14 040, Jun. 2020.

[19] 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)*, 2019, pp. 297–302.

[20] L. Chopra, S. Chakraborty, A. Mondal, and S. Chakraborty, "Parima: Viewport adaptive 360-degree video streaming," in *Proceedings of the Web Conference 2021*, ser. WWW '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 2379–2391.

[21] S. M. Haseeb Ul Hassan, A. Brennan, G.-M. Muntean, and J. McManis, "User profile-based viewport prediction using federated learning in real-time 360-degree video streaming," in *2023 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2023, pp. 1–7.

[22] S. Park, Y. Cho, H. Jun, J. Lee, and H. Cha, "Omnilive: Super-resolution enhanced 360° video live streaming for mobile devices," in *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*, ser. MobiSys '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 261–274.

[23] X. Feng, Z. Bao, and S. Wei, "Exploring cnn-based viewport prediction for live virtual reality streaming," in *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, 2019, pp. 183–1833.

[24] C. Concolato, J. Le Feuvre, F. Denoual, F. Mazé, E. Nassor, N. Ouedraogo, and J. Taquet, "Adaptive streaming of hevc tiled videos using mpeg-dash," *IEEE transactions on circuits and systems for video technology*, vol. 28, no. 8, pp. 1981–1992, 2017.

[25] M. Mahmoud, S. R. R. Stamatia, A. S. Panayides, P. I. Lazaridis, N. V. Kantartzis, G. K. Karagiannidis, and Z. D. Zaharis, "A comparative analysis of viewing prediction techniques for 360° video streaming applications," in *2024 Panhellenic Conference on Electronics Telecommunications (PACET)*, 2024, pp. 1–4.

[26] W.-C. Lo, C.-L. Fan, J. Lee, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "360° video viewing dataset in head-mounted virtual reality," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, ser. MMSys'17. New York, NY, USA: Association for Computing Machinery, 2017, p. 211–216. [Online]. Available: https://doi.org/10.1145/3083187.3083219

[27] C. Wu, Z. Tan, Z. Wang, and S. Yang, "A dataset for exploring user behaviors in vr spherical video streaming," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, ser. MMSys'17. New York, NY, USA: Association for Computing Machinery, 2017, p. 193–198. [Online]. Available: https://doi.org/10.1145/3083187.3083210

[28] A. T. Nasrabadi, A. Samiei, A. Mahzari, R. P. McMahan, R. Prakash, M. C. Q. Farias, and M. M. Carvalho, "A taxonomy and dataset for 360° videos," in *Proceedings of the 10th ACM Multimedia Systems Conference*, ser. MMSys '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 273–278. [Online]. Available: https://doi.org/10.1145/3304109.3325812