PRECEPT: Power-efficient Field-of-View Prediction for VR Video Streaming*

Ahmed Abdelreheem (*Member, IEEE)*, Darijo Raca, and Ahmed H. Zahran (*Senior Member, IEEE)*

***Abstract*—Field-of-view (FoV) prediction is critical for reducing device energy consumption and enhancing user quality of experience (QoE) in immersive streaming. To address the high computational and energy costs of standard DL-based FoV prediction, we propose PRECEPT, an energy-efficient, system-oriented two-stage framework. PRECEPT splits the prediction pipeline by adding a lightweight, CPU-based classifier to identify tile change. PRECEPT's classifier successfully filters approximately 80% of "no-change" events. PRECEPT activates the resource-intensive DL model only during identified tile change. This design reduces the average inference delay and energy consumption by up to 69% in a real mobile deployment. PRECEPT's two-stage design enables sustainable, high-performance FoV prediction on resource-constrained devices.**

***Index Terms*—Immersive video streaming, energy efficiency, field of view, prediction, feature engineering.**

I. Introduction

The Virtual Reality (VR) video market is projected to witness a Compound Annual Growth Rate (CAGR) of 25% from 2025 to 2033, reaching approximately 10 billion USD by 2033 [1]. However, the high traffic demand and the lack of pervasive consumer devices, e.g., Head Mounted Display (HMD), are identified as key challenges for VR markets. More critically, the user only sees a small portion of the panorama that overlaps with her Field of View (FoV). Hence, novel 360^o video streaming strategies embrace adaptive FoV strategies to optimise the traffic, energy, and user Quality of Experience (QoE) [2]–[4]. These streaming solutions adaptively select the quality of different video regions to efficiently utilize the wireless resources and improve user experience. Such adaptation is commonly enabled through tiled video encoding [5] that splits the projected panorama into independent regions. Adaptive algorithms can reduce traffic either aggressively, by streaming only the user's field of view, or more conservatively, by streaming a low-quality version outside the user FoV. These strategies could save up to 70% of the video traffic.

In such adaptive 360^o streaming systems, FoV prediction represents a fundamental component to avoid QoE degradation, e.g., stalls, emanating from user head movements. Hence, the design of advanced FoV prediction engines evolved as a crucial requirement. These engines would use historic data from HMD sensors that track head movement and eye gaze, if available, to predict users' future FoV. More complex engines would leverage other features, such as video motion data and historic heatmaps (saliency data) based on previous viewers behavior. These engines typically use Deep Learning (DL) models with different architectures, including Long-Short Term Memory (LSTM) [6], [7], transformer [8], and vision transformers [9]. Running such complex FoV prediction strains Graphical Processing Unit (GPU) resources. Hence, they contribute to higher energy consumption and video decoding latency.

This paper presents PRECEPT as a novel *power-efficient* FoV prediction engine for 360^o video streaming. This design is motivated by the importance of the tile change event, which represents the essence of FoV prediction to ensure timely request of missing tiles. More critically, advanced DL FoV prediction engines constitute a significant control overhead on the device GPU. Hence, we propose a novel two-stage FoV prediction framework. The first stage predicts tile change for future FoV and is designed to ensure fast and low-energy inference. Specifically, this stage is designed as a traditional Machine Learning (ML) classifier using *engineered features*. The second stage could be any typical DL FoV prediction model and is only executed when a change of tiles is anticipated by the first stage. Our extensive evaluation shows that the first stage detects 80%-88% of tile no-change events and 72%-82% of tile changes for all tested horizons (0.25s-2s). We show that PRECEPT's two-stage design reduces the average inference energy and latency by up to 69% in comparison to a typical LSTM baseline without compromising the prediction accuracy. To the best of our knowledge, PRECEPT is the first FoV prediction engine designed to reduce energy consumption and can be integrated with any FoV prediction engine.

II. Background and Related Work

The majority of work regarding adaptive bitrate streaming in 360-degree video focuses on tiled 360-degree video, allowing requesting tiles with different bitrate quality depending on viewer's viewport. Techniques used for developing these approaches broadly range from machine learning-based [10], [11], optimisation-based [12], [13], control-theoretic [14], and heuristic-based [15]–[18]. Alternative to the tile-based approach, Alhilal et al. [19] propose an approach where regions of a single frame are encoded in different quality levels unlike requesting separate tiles (files) with different quality. Predicting the viewer's viewport (i.e., FoV prediction) position represents a key requirement for successful adaptive bitrate streaming in 360-degree video. In addition to adaptive

*This research is supported by Research Ireland CONNECT Center.

Ahmed Abdelreheem is with University College Cork, Cork, Ireland (aabdelreheem@ucc.ie)
Darijo Raca is with University of Sarajevo, Sarajevo, Bosnia and Herzegovina (draca@etf.unsa.ba)
Ahmed H. Zahran is with University College Cork, Cork, Ireland (ahmed.zahran@ucc.ie)

streaming algorithm, the authors propose their own techniques or use existing methods for FoV prediction.

Early FoV prediction solutions employ LSTM architectures to model temporal dependencies. For instance, Fan et al. [6] utilized sensor data (head orientation) combined with content features like saliency and motion maps to predict fixations up to one second ahead. To combat visual distortion emanating from spherical projections, "Overlapping Virtual Viewports" (OVV) were introduced to ensure proper computer vision algorithms [20]. To achieve long-term prediction (1 to 10 seconds), Li et al. [7] utilized sequence-to-sequence (seq2seq) models with "Attentive Mixture of Experts" to fuse a target user's history with the future trajectories of similar viewers. Nasrabadi et al. [21] employed clustering algorithms like DBSCAN with quaternion extrapolation to avoid the inaccuracies of Eulerian coordinates.

Recent models focus on refining input features and model architectures to handle dynamic user behaviors and varying data distributions. Recognizing that head orientation is a coarse proxy for attention, systems like SalientVR and the Gaze-based FoV Prediction (GFP) model incorporate precise eye-tracking data [22]. Cross Modal Multiscale Transformer (CMMST) [23] aligns video saliency with user trajectories to capture intricate modal dependencies. The DVP360 [24] model combines Temporal Convolutional Networks (TCN) and CNNs with knowledge transfer to adapt to data distribution shifts, maintaining a lightweight size of 17.3 MB for a 5-second horizon.

III. PRECEPT Design

A. PRECEPT Overview

PRECEPT is designed to minimize both the energy consumption and time required for FoV prediction through a two-stage architecture. Figure 1 compares PRECEPT's FoV prediction approach with a conventional single-stage method. The first stage of PRECEPT employs a fast, energy-efficient, and encoding-aware change detector to determine whether the tiles in the upcoming prediction horizon are likely to differ from the current FoV tiles. If no change is detected, the future FoV is assumed to be identical to the current one. If a change is anticipated, the system predicts the future Point of View (PoV) to identify the new set of tiles. This design introduces a filtering mechanism that activates the more resource-intensive prediction stage only when necessary, thereby conserving energy and reducing inference time. To maintain low energy consumption, the change detector—referred to as PRECEPT-C is developed as a binary classifier using traditional machine learning techniques, as elaborated in Section III-B. The second stage, PRECEPT-P, functions as the core FoV prediction engine and can be any state-of-the-art predictors.

B. PRECEPT-C Design

PRECEPT-C is designed as a binary classifier using supervised learning. This section details the data labeling methodology, feature engineering process, model development, and optimization strategies employed to achieve real-time inference performance.

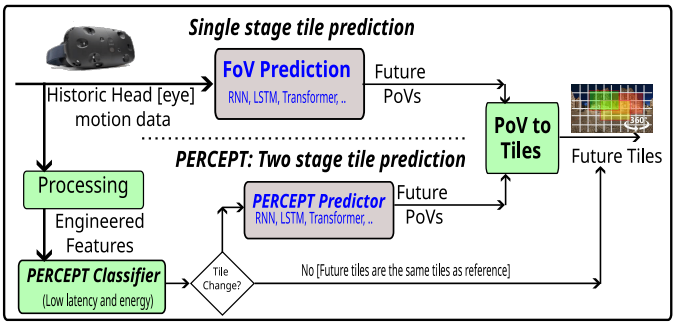

Fig. 1. Illustration of PRECEPT prediction versus traditional approaches.

1) Data Labeling: We process raw VR sensor data to establish ground-truth labels for supervised learning. The raw tracking data comprises six organic features recorded by the HMD: head orientation represented as a 3D Cartesian unit vector $\mathbf{h}_t = [h_x, h_y, h_z]_t$ indicating the viewing direction on the spherical projection, and eye gaze direction similarly represented as $\mathbf{e}_t = [e_x, e_y, e_z]_t$ capturing the user's visual attention within the HMD's field of view. We assume that the panorama is flattened using equirectangular projection with width and height denoted as W and H, respectively. The projection is partitioned into a grid of $N_h \times N_v$ tiles. We also assume that the HMD FoV is defined by θ_h and θ_v for the horizontal and vertical viewing angles, respectively.

a) Mapping PoV to Tiles.: For each sensor sample t, we compute the set of tiles $\mathcal{T}_t$ overlapping with the user's FoV through a geometric transformation pipeline. The head orientation vector $\mathbf{h}_t$ is first normalized to unit length, and then converted to spherical coordinates:

$$\phi = \arctan 2(h_y, h_x), \quad \theta = \arccos(h_z) \tag{1}$$

where $\phi \in [0, 2\pi)$ represents the azimuthal angle (adjusted for negative values). The spherical coordinates are projected onto the equirectangular image plane:

$$x_{\text{pix}} = \left\lfloor \frac{W \cdot \phi}{2\pi} \right\rfloor, \quad y_{\text{pix}} = \left\lfloor \frac{H \cdot \theta}{\pi} \right\rfloor \tag{2}$$

The viewport dimensions in pixels are computed from the angular FoV as $w_{\text{vp}} = \frac{W}{2\pi} \cdot \theta_h$ and $h_{\text{vp}} = \frac{H}{\pi} \cdot \theta_v$. The FoV bounding box is then defined by its top-left corner $(x_{\text{pix}} - w_{\text{vp}}/2,\ y_{\text{pix}} - h_{\text{vp}}/2)$ and bottom-right corner $(x_{\text{pix}} + w_{\text{vp}}/2,\ y_{\text{pix}} + h_{\text{vp}}/2)$. All tiles within this bounding box are enumerated with horizontal wraparound handling for 360° continuity to form the tile set $\mathcal{T}_t$.

The tile change label y_t considers the past T_h historic samples to predict changes in the future T_f horizon. The label of sample t is computed as:

$$y_t = \mathbb{1}\left[\left|\mathcal{T}_t \triangle \bigcup_{\tau=t+1}^{t+T_f} \mathcal{T}_\tau\right| \geq 1\right] \tag{3}$$

where $\triangle$ denotes the symmetric difference operator. A label of $y_t = 1$ indicates that at least one tile will change (either added or removed from the FoV) within the prediction horizon in comparison to the current FoV tiles; $y_t = 0$ indicates the FoV tiles remain unchanged.

2) Feature Engineering: PRECEPT-C employs a carefully engineered feature set of 52 attributes that capture the orientation context through both organic sensor data and derived kinematic features. Note that conventional single-stage unimodal DL models would use at least $N_h\times$ the number of considered organic features. Hence, the considered 52 features represent a significant reduction in the model input. This approach enables the use of computationally efficient gradient boosting methods while maintaining predictive accuracy.

Notation. Let $\mathbf{p}_t = [p_x, p_y, p_z]_t$ denote a generic 3D tracking vector representing either head orientation ($\mathbf{p} = \mathbf{h}$) or eye gaze ($\mathbf{p} = \mathbf{e}$). Window sizes are denoted $w \in \mathcal{W}$, varying by feature type.

Organic Features (6). Head orientation $\mathbf{h}_t$ and eye gaze $\mathbf{e}_t$ as 3D Cartesian unit vectors.

Instantaneous Movement Dynamics (9). For $\mathbf{p} \in \{\mathbf{h}, \mathbf{e}\}$, velocity and speed are:

$$\dot{p}_{i,t} = p_{i,t} - p_{i,t-1}, \quad s_t^p = \|\dot{\mathbf{p}}_t\|, \quad i \in \{x, y, z\} \tag{4}$$

Head acceleration: $a_{i,t} = \dot{h}_{i,t} - \dot{h}_{i,t-1}$, $a_t = \|[a_{x,t}, a_{y,t}, a_{z,t}]\|$. This yields 5 head and 4 eye features.

Direction Change (1). Angular change in head velocity direction:

$$\alpha_t = \arccos\left(\dot{\mathbf{h}}_t \cdot \dot{\mathbf{h}}_{t-1} / \|\dot{\mathbf{h}}_t\| \|\dot{\mathbf{h}}_{t-1}\|\right) \tag{5}$$

Multi-Scale Velocity (9). Velocity slopes across windows $\mathcal{W}_v$:

$$v_{i,t}^{(w)} = (h_{i,t} - h_{i,t-w})/w, \quad w \in \mathcal{W}_v,\ i \in \{x, y, z\} \tag{6}$$

Angular Rates (3). Yaw (ψ), pitch (ϑ), roll (φ) rates with yaw wraparound:

$$\dot{\psi}_t = \text{wrap}(\psi_t - \psi_{t-1}), \quad \dot{\vartheta}_t = \vartheta_t - \vartheta_{t-1}, \quad \dot{\varphi}_t = \varphi_t - \varphi_{t-1} \tag{7}$$

where wrap$(\cdot)$ handles $[0°, 360°)$ discontinuity.

Head-Eye Coordination (2). Distance and divergence rate:

$$d_t^{he} = \|\mathbf{h}_t - \mathbf{e}_t\|, \quad \dot{d}_t^{he} = d_t^{he} - d_{t-1}^{he} \tag{8}$$

Movement Consistency (6). Velocity standard deviation over windows $\mathcal{W}_\sigma$:

$$\sigma_{i,t}^{(w)} = \text{std}(\{\dot{h}_{i,k}\}_{k=t-w+1}^{t}), \quad w \in \mathcal{W}_\sigma,\ i \in \{x, y, z\} \tag{9}$$

Quaternion Angular Velocity (1). Rotation magnitude from quaternions:

$$\omega_t^q = \|\text{rotvec}(\mathbf{q}_t \cdot \mathbf{q}_{t-1}^{-1})\| \tag{10}$$

Historical Tile Change (3). Windowed change counts over $\mathcal{W}_c$:

$$c_t^{(w)} = \sum_{k=t-w}^{t-1} \mathbb{I}[y_k = 1], \quad w \in \mathcal{W}_c \tag{11}$$

Temporal Smoothing (4). Rolling mean velocities over $\mathcal{W}_s$:

$$\tilde{v}_{i,t}^{(w)} = \text{mean}(\{\dot{h}_{i,k}\}_{k=t-w+1}^{t}), \quad w \in \mathcal{W}_s,\ i \in \{x, z\} \tag{12}$$

Position Variance (6). Spatial extent over windows $\mathcal{W}_p$:

$$\text{var}_{i,t}^{(w)} = \text{var}(\{h_{i,k}\}_{k=t-w+1}^{t}), \quad w \in \mathcal{W}_p,\ i \in \{x, y, z\} \tag{13}$$

Additional Derived (5). Time since last tile change τ_t; rolling speed mean $\bar{s}_t^{(w_a)}$; jerk $j_t = a_t - a_{t-1}$; head-eye velocity angle $\beta_t = \arccos(\dot{\mathbf{h}}_t \cdot \dot{\mathbf{e}}_t / \|\dot{\mathbf{h}}_t\| \|\dot{\mathbf{e}}_t\|)$; movement entropy $\eta_t^{(w_e)} = \text{std}(\{\alpha_k\}_{k=t-w_e+1}^{t})$.

All features use only historical information up to frame t for causality. Missing boundary values are zero-filled.

3) Model Development: We consider XGBoost (eXtreme Gradient Boosting) for PRECEPT-C due to its computational efficiency, strong performance on structured tabular data, and suitability for real-time inference. We also evaluated random forest (RF) but report only XGBoost due to its superior performance and page space limits.

XGBoost builds an ensemble of decision trees sequentially through gradient boosting, minimizing a regularized objective function expressed as:

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} \ell(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{14}$$

where ℓ is the logistic loss for binary classification, $\hat{y}_i$ is the predicted probability, y_i is the true label, and $\Omega(f_k)$ represents a regularization term defined as:

$$\Omega(f_k) = \gamma L_k + \frac{1}{2}\lambda \sum_{j=1}^{L_k} w_{k,j}^2 \tag{15}$$

where L_k is the number of leaves in tree k, $w_{k,j}$ are the leaf weights, γ controls tree complexity, and λ is the L2 regularization coefficient. The prediction after K boosting iterations is:

$$\hat{y}_i^{(K)} = \sum_{k=1}^{K} f_k(\mathbf{x}_i) = \hat{y}_i^{(K-1)} + \eta \cdot f_K(\mathbf{x}_i) \tag{16}$$

where η is the learning rate controlling the contribution of each tree and $\mathbf{x}_i$ is the feature vector.

In this evaluation, we consider [25] that includes 50 users, 15 videos whose duration range from 140s to 352s and all HMD sensors are sampled at 120Hz. The dataset includes diverse content categories, such as sports, landscapes, cartoons, and narrative, exhibiting various watching dynamics. We set the HMD viewing angle to $\theta_h = 110°$ and $\theta_v = 90°$, which represents the specs of Meta Quest 3. The tiling configuration is set to $N_h = 8 \times N_v = 6$. We also consider a 1s historic window (120 sample) and we trained models for different horizons (T_f) (0.25s, 0.5s, 1s, and 2s). We employ 80-20 user-based data split. Hence, any user data remains entirely within either the training or test set across all videos. This approach prevents memorizing user-specific viewing patterns and ensures the classifier generalizes to unseen users. The windows considered for multi-scale features are set as

TABLE I
PRECEPT-C PERFORMANCE FOR PREDICTION HORIZONS.

	(300,8,0.1)				**(150,8,0.2)**				**(150,4,0.25)**				**(75,8,0.3)**				**(75,4,0.35)**			
T_f	30	60	120	240	30	60	120	240	30	60	120	240	30	60	120	240	30	60	120	240
Acu	0.87	0.817	0.773	0.745	0.868	0.815	0.771	0.744	0.847	0.803	0.765	0.735	0.864	0.813	0.769	0.74	0.84	0.797	0.762	0.733
TNR	0.885	0.851	0.823	0.796	0.884	0.849	0.821	0.792	0.861	0.837	0.826	0.82	0.878	0.847	0.82	0.798	0.854	0.83	0.82	0.822
TPR	0.821	0.761	0.728	0.72	0.82	0.759	0.727	0.721	0.805	0.748	0.712	0.695	0.821	0.757	0.725	0.712	0.796	0.744	0.711	0.691
PPV	0.701	0.758	0.824	0.882	0.698	0.755	0.822	0.88	0.654	0.738	0.822	0.891	0.688	0.752	0.82	0.882	0.641	0.729	0.817	0.891
F1	0.756	0.759	0.773	0.793	0.754	0.757	0.771	0.792	0.721	0.743	0.763	0.781	0.749	0.754	0.769	0.788	0.71	0.736	0.76	0.779
Util	0.302	0.4	0.492	0.578	0.303	0.4	0.492	0.579	0.316	0.403	0.482	0.552	0.307	0.401	0.491	0.571	0.318	0.405	0.484	0.55
ID_{95}^{CPU}	0.138	0.138	0.148	0.171	0.123	0.123	0.123	0.124	0.116	0.114	0.144	0.115	0.117	0.117	0.117	0.117	0.113	0.114	0.119	0.114
Size	5000	4800	4700	4400	2500	2400	2300	2200	257	257	257	257	1300	1200	1200	1100	129	129	129	129

$\mathcal{W}_s = \{5, 10, 20\}$, $\mathcal{W}_c = \{10, 30, 60\}$, $\mathcal{W}_p = \{20, 40\}$, $\mathcal{W}_\sigma = \{10, 20, 30, 40\}$, and $\mathcal{W}_v = \{3, 5, 10, 20, 30\}$. The model training and evaluation is conducted on a DELL machine equipped with 32GB RAM, Intel(R) Core(TM) Ultra 7 processor 265KF (20-Core, 66MB Total Cache, 3.3GHz to 5.5GHz) CPU, and NVIDIA(R) GeForce RTX(TM) 5080 16GB memory.

We trained multiple XGBoost models identified by (number of trees, maximum depth, learning rate). We explored different combinations of these parameters to investigate design tradeoff across accuracy, inference latency (ms), and model sizes (Universal Binary JSON format in KB). Note that reducing the number of trees decreases sequential evaluations, decreasing the tree depth reduces nodes per tree, and increasing the learning rate enables faster convergence. To handle class imbalance between change and no-change records, a scale weight $= n_0/n_1$, where n_0 and n_1 are the number of records for change and no-change classes, is applied. We consider 500 iterations with early stopping (patience = 30 rounds). In all configurations, we maintain a consistent regularization ($\gamma = 0.1$, $\alpha = 0.1$, $\lambda = 1.0$) for fair comparison.

PRECEPT-C Performance. Our evaluation considers standard machine learning metrics including accuracy (Acu), true positive rate (TPR), true negative rate (TNR), positive predictive value (PPV), and F1-Score (F1)[1]. In our problem, TPR measures change detectability, TNR indicates a lower bound on possible energy savings, and PPV measures the model efficiency for using the GPU when running stage 2. Additionally, we consider inference delay statistics, denoted as ID_x, where x represents the statistical percentile (e.g., 50 for median), model size, and GPU utilization (Util). Let TC denote the ratio of tile change event. Util is then estimated as $TC * TPR + (1 - TC)(1 - TNR)$, where the first term corresponds to detected TC while the second corresponds to false detected changes.

Table I highlights the performance of different PRECEPT-C models for different horizons and model parameters. PRECEPT-C achieves a striking low CPU inference delay, an ID_{95}^{CPU} less than 0.18ms for all tested scenarios. Hence, PRECEPT-C delivers realtime performance without relying on the system GPU. PRECEPT's GPU utilization is as low as 30% for short prediction horizon and it grows to only 58% for the 2-second horizon. This low utilization is mainly driven by the high TNR (0.79-0.89) across all horizons. PRECEPT-C achieves a TPR of 0.72-0.82 indicating strong ability to detect tile change events and promptly prefetch needed tiles. PRECEPT-C achieves a PPV range of 0.7-0.82 suggesting an increasing GPU usage efficiency as the prediction horizon increases. This proportional increase is attributed to the higher frequency of tile change events. PRECEPT-C performance promises significant reduction in GPU utilization, power, and inference delay for PRECEPT's two stage design. Such reduction would free the GPU for other demanding operations, such as video decoding.

The performance of optimized PRECEPT-C models suggests that maintaining the depth and reducing the number of trees has insignificant impact on the ML metrics. In comparison to the parameter-rich configuration (300, 8, 0.1), the performance degradation of (75,8,0.3) remains less than 0.5% for all metrics; but reduces the model size by 75% and the inference delay by up to 32%. Hence, optimized models can reduce the system resource utilization without a significant impact on the system performance.

Feature Importance Analysis. To understand which motion characteristics most strongly predict tile changes, we analyze XGBoost feature importance across tested prediction horizons. Table II presents the top-10 features for each horizon.

TABLE II
TOP-10 FEATURE IMPORTANCES (%) ACROSS PREDICTION HORIZONS

Rank	Feature	H=30	H=60	H=120	H=240
1	$\sigma_{x,t}^{(w)}$	14.1	18.5	24.5	22.8
2	s_t^h	13.1	13.3	9.5	7.4
3	$\sigma_{x,t}^{(w')}$	11.8	10.9	6.5	5.7
4	ω_t^q	6.6	7.5	6.2	6.7
5	$\bar{s}_t^{(w_a)}$	3.8	5.4	11.9	7.9
6	h_z	4.9	3.6	2.2	2.1
7	a_t	3.2	3.0	2.2	2.0
8	τ_t	2.8	2.4	2.1	3.4
9	$\tilde{v}_{z,t}^{(w)}$	2.5	1.9	1.3	–
10	$\text{var}_{x,t}^{(w)}$	–	–	3.8	5.2

Consistent Top Features. Six features appear in the top-10 across all four prediction horizons:

1) **Movement consistency** $\sigma_{x,t}^{(w)}$: Dominates all horizons (14–25%), indicating that variability in horizontal head

[1]TPR, TNR, and PPV are also widely known as recall, specificity, and precision respectively.

velocity is the strongest predictor of tile changes.
2) **Movement speed** s_t^h: Consistently ranks 2nd–3rd, confirming instantaneous motion magnitude as a key indicator.
3) **Movement consistency** $\sigma_{x,t}^{(w')}$: The shorter-window variant complements $\sigma_{x,t}^{(w)}$, capturing rapid fluctuations.
4) **Quaternion angular velocity** ω_t^q: Maintains 4th–5th rank, validating rotation-based features.
5) **Acceleration magnitude** a_t: Appears in positions 7–9, indicating motion dynamics matter.
6) **Time since last change** τ_t: Consistently important, reflecting temporal autocorrelation in viewing behavior.

Horizon-Dependent Patterns. As the prediction horizon increases, position variance $\text{var}_{x,t}^{(w)}$ gains importance, appearing in top-10 only for $H \geq 120$. This suggests that spatial extent of movement becomes more predictive for longer-term forecasting. Conversely, raw position features (h_z, h_x) decrease in relative importance at longer horizons.

The dominance of x-axis (horizontal) features across all metrics reflects the prevalence of lateral head movements in VR content exploration, consistent with the natural tendency for users to scan horizontally within 360° video environments.

C. Precept Predictor Design

Conventional prediction agents are typically trained on all dataset training records. On the contrary, the PRECEPT predictor (PRECEPT-P) is a specialized FoV prediction agent trained on records featuring tile change. *Our hypothesis is that PRECEPT-P will improve the prediction accuracy in comparison to models trained on all records*. To validate this hypothesis, we consider a baseline trained on all records and PRECEPT-P trained on a subset of training records encountering a change. Both models use the same architecture and history-horizon windows. Specifically, PRECEPT-P uses sequence-to-sequence architecture consisting of encoder and decoder models. Furthermore, the architecture is augmented with a Bahdanau-style additive attention mechanism to capture the non-linear temporal dynamics of user movement. The encoder consists of two bidirectional LSTM layers with 128 and 64 hidden units, respectively. To counter the overfitting challenge, layers employ dropout with 0.2 rate. Finally, for the decoder, we utilise LSTM layer with 64 hidden units.

Our key comparison metrics include
- *tile accuracy* (TA) representing the percentage of accurately predicted tiles.
- *tile overhead* (TO) represents the percentage of tiles requested outside the actual FoV.

Table III compares the TA and TO of the baseline and PRECEPT-P. The table shows that PRECEPT-P improves the accuracy by 1-2 % but it also increases the overhead by 1-3 %. With this clear trade-off, we do not envision a winning prediction training methodology.

TABLE III
PREDICTION PERFORMANCE FOR TEST RECORDS FEATURING TILE CHANGE.

	Baseline				PRECEPT-P			
T_f	30	60	120	240	30	60	120	240
TA	0.89	0.85	0.80	0.74	0.9	0.87	0.81	0.76
TO	0.03	0.04	0.06	0.08	0.04	0.06	0.07	0.11

TABLE IV
PREDICTION PERFORMANCE FOR ***all test records***.

	T_f	TA	TO	ID	IE
Baseline	30	0.96	0.03	10.45	28.8
	60	0.92	0.04	11.1	35.3
	120	0.87	0.06	12.3	38.7
	240	0.80	0.09	15.2	44.3
PRECEPT	30	0.96	0.042	3.3	9.1
	60	0.93	0.073	4.5	14.3
	120	0.88	0.108	6.1	19.1
	240	0.81	0.150	8.7	25.4

IV. PERFORMANCE EVALUATION

This section compares the performance of PRECEPT (integrated PRECEPT-C (75,8,0.3) and PRECEPT-P) to the baseline LSTM predictor for the entire test dataset. In addition to the TA and TO, we also compare the average inference delay (ms) and energy consumption (mJ) using a real mobile device (Samsung Tab S8 Ultra featuring an Octa-core Qualcomm SM8450 Snapdragon 8 Gen 1 and Adreno 730 GPU). PRECEPT-C is implemented in Android using ONNX runtime framework[2] and prediction models were cross-compiled from PyTorch into TensorFlow Lite FlatBuffers using LiteRT[3]. The inference delay is measured for every instance while the energy efficiency is measured using the *Android BatteryManager API* in a batch of ten thousand samples as recommended in [26] for accurate energy measurement.

Table IV shows these metrics for both PRECEPT and baseline for all prediction horizons. The result shows that PRECEPT achieves a significant reduction across both inference delay and energy consumption. PRECEPT's ID drops to 31%-57% of the LSTM baseline, respectively. Larger inference delay savings are attained for smaller horizons (e.g., $T_f = 30,\ 60$) due to a larger percentage of no-tile change events. Hence, a significant portion of the prediction workload is handled by PRECEPT's first stage, i.e., PRECEPT-C. Similarly, PRECEPT's achieves inference energy savings, with IE dropping to 43%-69% of the LSTM baseline, respectively. Both energy and delay improvement are driven by PRECEPT-C negligible CPU-based inference delay ($\tilde{0}.5\%$ of the prediction stage ID) and insignificant inference energy (less than 1% of prediction stage IE) PRECEPT achieves these performance gains while slightly improving the tile accuracy that is accompanied by a slight increase in the tile overhead.

[2]https://onnxruntime.ai/
[3]https://github.com/google-ai-edge/LiteRT

V. Conclusions and Future Work

This paper introduces PRECEPT, a novel two-stage framework for energy-efficient FoV prediction in immersive streaming. By leveraging an efficient CPU-based first stage, PRECEPT significantly reduces both the average inference delay and energy consumption by up to 69% in a real mobile deployment without compromising the prediction accuracy. Hence, PRECEPT enables efficient real-time FoV tracking to maintain optimal VR streaming quality. By adopting the proposed two-stage strategy, i.e., pairing PRECEPT-C with advanced DL-based FoV prediction models, similar improvements can be attained by other state-of-the-art-models. Our future work targets developing a universal change detector (PRECEPT-C) that is independent of tiling and HMD configurations. Such universal model is envisioned to integrate additional features representing tiling and HMD FoV specifications relative to the projected panorama.

Acknowledgments

This research was supported by the Federal Ministry of Education and Science, BiH.

References

[1] Data Insights Market, "VR Video Content in Emerging Markets: Analysis and Projections 2025-2033," Jun 2025. Accessed: 1 Dec 2025.

[2] Y. Guan, C. Zheng, X. Zhang, Z. Guo, and J. Jiang, "Pano: Optimizing 360^o video streaming with a better understanding of quality perception," *SIGCOMM 2019 - Proceedings of the 2019 Conference of the ACM Special Interest Group on Data Communication*, no. 1, pp. 394–407, 2019.

[3] Y. Hu, Y. Liu, and Y. Wang, "VAS360: QoE-driven viewport adaptive streaming for 360 video," *Proceedings - 2019 IEEE International Conference on Multimedia and Expo Workshops, ICMEW 2019*, vol. 42, pp. 324–329, 2019.

[4] S. Ahsan, A. Hourunranta, I. D. Curcio, and E. Aksu, "Frisbe: Adaptive bit rate streaming of immersive tiled video," *Proceedings of the 2020 Packet Video Workshop, PV 2020 - Part of MMSys 2020*, pp. 28–34, 2020.

[5] C. L. Fan, W. C. Lo, Y. T. Pai, and C. H. Hsu, "A survey on 360^o video streaming: Acquisition, transmission, and display," *ACM Computing Surveys*, vol. 52, no. 4, 2019.

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

[7] 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)*, pp. 297–302, 2019.

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

[9] M. Cokelek, H. Ozsoy, N. Imamoglu, C. Ozcinar, I. Ayhan, E. Erdem, and A. Erdem, "Spherical vision transformers for audio-visual saliency prediction in 360° videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 48, no. 1, pp. 329–345, 2026.

[10] Y. Lu, Y. Zhu, and Z. Wang, "Personalized 360-degree video streaming: A meta-learning approach," in *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, (New York, NY, USA), p. 3143–3151, Association for Computing Machinery, 2022.

[11] S. Wang, S. Yang, H. Su, C. Zhao, C. Xu, F. Qian, N. Wang, and Z. Xu, "Robust saliency-driven quality adaptation for mobile 360-degree video streaming," *IEEE Transactions on Mobile Computing*, vol. 23, no. 2, pp. 1312–1329, 2024.

[12] J. Chen, Z. Luo, Z. Wang, M. Hu, and D. Wu, "Live360: Viewport-aware transmission optimization in live 360-degree video streaming," *IEEE Transactions on Broadcasting*, vol. 69, no. 1, pp. 85–96, 2023.

[13] X. Wei, M. Zhou, and W. Jia, "Toward low-latency and high-quality adaptive 360° streaming," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 5, pp. 6326–6336, 2023.

[14] R. Xu, C. Liu, M. Hu, S. Qian, Y. Zhang, and T. Lin, "Omms: Multiple control based adaptive 360° video streaming," in *Proceedings of the 15th ACM Multimedia Systems Conference*, MMSys '24, (New York, NY, USA), p. 429–434, Association for Computing Machinery, 2024.

[15] 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.

[16] D. Sheng, B. Gao, G. Liu, Q. Qi, and J. Wang, "Efficient tile-based adaptive streaming for 360-degree video-on-demand systems," in *Proceedings of the 15th ACM Multimedia Systems Conference*, MMSys '24, (New York, NY, USA), p. 435–440, Association for Computing Machinery, 2024.

[17] A. Yaqoob and G.-M. Muntean, "Advanced predictive tile selection using dynamic tiling for prioritized 360° video vr streaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 20, Aug. 2023.

[18] Z. Ye, Q. Li, X. Ma, D. Zhao, Y. Jiang, L. Ma, B. Yi, and G.-M. Muntean, "Vrct: A viewport reconstruction-based 360° video caching solution for tile-adaptive streaming," *IEEE Transactions on Broadcasting*, vol. 69, no. 3, pp. 691–703, 2023.

[19] A. Alhilal, Z. Wu, Y. H. Tsui, and P. Hui, "Fovoptix: Human vision-compatible video encoding and adaptive streaming in vr cloud gaming," in *Proceedings of the 15th ACM Multimedia Systems Conference*, MMSys '24, (New York, NY, USA), p. 67–77, Association for Computing Machinery, 2024.

[20] C.-L. Fan, S.-C. Yen, C.-Y. Huang, and C.-H. Hsu, "Optimizing fixation prediction using recurrent neural networks for 360° video streaming in head-mounted virtual reality," *IEEE Transactions on Multimedia*, vol. 22, no. 3, pp. 744–759, 2020.

[21] 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*, NOSSDAV '20, (New York, NY, USA), p. 34–39, Association for Computing Machinery, 2020.

[22] 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*, MobiCom '22, (New York, NY, USA), p. 542–555, Association for Computing Machinery, 2022.

[23] Y. Tian, Y. Zhong, Y. Han, and F. Chen, "Viewport prediction with cross modal multiscale transformer for 360^o video streaming," *Scientific Reports*, vol. 15, p. 30346, Aug. 2025.

[24] J. Lv, Y. Chen, X. Xu, and X. Qin, "Deep viewpoint prediction in 360º video with data distribution adaptation," in *2024 16th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1318–1323, 2024.

[25] Y. Xu, J. Du, J. Wang, Y. Ning, S. Zhou, and Y. Cao, "Panonut360: A head and eye tracking dataset for panoramic video," in *Proceedings of the 15th ACM Multimedia Systems Conference*, MMSys '24, (New York, NY, USA), pp. 319–325, Association for Computing Machinery, 2024.

[26] X. Tu, A. Mallik, D. Chen, K. Han, O. Altintas, H. Wang, and J. Xie, "Unveiling energy efficiency in deep learning: Measurement, prediction, and scoring across edge devices," in *Proceedings of the Eighth ACM/IEEE Symposium on Edge Computing*, SEC '23, (New York, NY, USA), p. 80–93, Association for Computing Machinery, 2024.