TRACK: A New Method from a Re-examination of Deep Architectures for Head Motion Prediction in 360° Videos

Miguel Fabián Romero Rondón, Lucile Sassatelli, Ramón Aparicio Pardo, and Frédéric Precioso

Abstract—

We consider predicting the user's head motion in 360° videos, with 2 modalities only: the past user's positions and the video content (not knowing other users' traces). We make two main contributions. First, we re-examine existing deep-learning approaches for this problem and identify hidden flaws from a thorough root-cause analysis. Second, from the results of this analysis, we design a new proposal establishing state-of-the-art performance. First, re-assessing the existing methods that use both modalities, we obtain the surprising result that they all perform worse than baselines using the user's trajectory only. A root-cause analysis of the metrics, datasets and neural architectures shows in particular that (i) the content can inform the prediction for horizons longer than 2 to 3 sec. (existing methods consider shorter horizons), and that (ii) to compete with the baselines, it is necessary to have a recurrent unit dedicated to process the positions, but this is not sufficient. Second, from a re-examination of the problem supported with the concept of Structural-RNN, we design a new deep neural architecture, named TRACK. TRACK achieves state-of-the-art performance on all considered datasets and prediction horizons, outperforming competitors by up to 20% on focus-type videos and horizons 2-5 seconds. The entire framework (codes and datasets) is online and received an ACM reproducibility badge https://gitlab.com/miguelfromeror/head-motion-prediction.

Index Terms—Modeling and prediction, Virtual reality, Neural nets, Machine learning, Kinematics and dynamics

1 Introduction

Immersive media are on the rise: the global market for Virtual Reality (VR) is projected to grow from US$9.2 Billion in 2020 to US$89.1 Billion by 2027 [2]. 360° videos are an important modality of VR, with applications in story-telling, journalism or remote education. Despite these exciting prospects, the development is persistently hindered by the difficulty to access immersive content through Internet streaming. Indeed, owing to the closer proximity of the screen to the eye in VR and to the width of the content (2π steradians in azimuth and π in elevation angles), the data rate is two orders of magnitude that of a regular video [3]. To decrease the amount of data to stream, a solution is to send in high resolution only the portion of the sphere the user has access to at each point in time, named the Field of View (FoV). To do so, recent works have proposed to either segment the video spatially into tiles and set the quality of the tiles according to their proximity to the FoV [4], [5], [6], or use projections enabling high resolutions of regions close to the FoV [7], [8]. These approaches however require to know the user's head position in advance, that is at the time of sending the content from the server (see Fig. 1). Failing to predict correctly the future user's positions can lead to a lower quality displayed in the FoV, which can impair the user's experience. It is therefore crucial for an efficient 360° video streaming system to embed an accurate head motion predictor to periodically inform where the user will be likely looking at, over a future horizon.

In this article, we consider the problem of predicting the user's head motion in 360° videos over a future horizon, based both and only on the past trajectory and on the video content. Various methods tackling this problem with deep neural networks have therefore been proposed in the last couple of years (e.g., [9], [10], [11], [12], [13]). We show that the relevant existing methods have hidden flaws, that we thoroughly analyze to overcome with a new proposal establishing state-of-the-art performance. We hence make two main contributions.

Contributions:

- **Uncovering hidden flaws of existing methods and performing a root-cause analysis**:

After a review and taxonomy of the most relevant and recent methods (PAMI18 [9], CVPR18 [10], MM18 [11], ChinaCom18 [12] and NOSSDAV17 [13]), we compare them to common baselines. First, comparing against the *trivial-static baseline*, we obtain the intriguing result that they all perform worse, on their exact original settings, metrics and datasets. Second, we show it is indeed possible to outperform the *trivial-static baseline* (and hence the existing methods) by designing a stronger baseline, named the *deep-position-only baseline*: it is an LSTM-based architecture considering only the positional information, while the existing methods are meant to benefit both from the history of past positions and knowledge of the video content. From there, we carry out a thorough root-cause analysis to understand why the existing methods perform worse than baselines that do not consider the content information. Looking into the metrics and the data, we show that: (i) evaluating only on some specific pieces of trajectories or specific videos, where the content is proved useful, does not change the comparison results, and that (ii) the content can indeed inform the head position prediction, but for

A preliminary version of this work has been published in [1]. The authors are with Université Côte d'Azur, CNRS, I3S, 06900 Sophia Antipolis, France. Lucile Sassatelli is also with Institut Universitaire de France. Frédéric Precioso is also with Inria. E-mail: {first.last}@univ-cotedazur.fr prediction horizons longer than 2 to 3 sec.. All these existing methods consider shorter horizons. Looking into the neural network architectures, we identify that: (iii) when the provided content features are the ground-truth saliency, the only architecture not degrading away from the baseline is the one with a Recurrent Neural Network (RNN) layer dedicated to the positional input, but (iv) when fed with saliency estimated from the content, the performance of this architecture degrades away from the *deep-position-only baseline* again.

• **Introducing a new deep neural architecture achieving state-of-the-art performance on all the datasets of compared methods and all prediction horizons (0-5 sec.)**:
To overcome this difficulty, we re-examine the requirements on how both modalities (past positions and video content) should be considered given the structure of the problem. We support our reasoning with the concept of Structural-RNN, modeling the dynamic head motion prediction problem as a spatio-temporal graph. We obtain a new deep neural architecture, that we name TRACK. TRACK establishes state-of-the-art performance on all the prediction horizons 0-5 sec. and all the datasets of the existing competitors. In the 2-5 sec. horizon, TRACK outperforms the second-best method by up to 20% in orthodromic distance error on focus-type videos, i.e., videos with low-entropy saliency maps.

Owing to the critical results and perspective we raise on the state-of-the-art, and in our concern for reproducibility, the experimental setup and datasets of each assessed method and all our codes, are provided online (detailed and illustrated) at [14]. This reproducible framework has already obtained an ACM reproducibility badge [15], and allows to easily test any future approach.

Sec. 2 formulates the exact prediction problem considered, and presents a taxonomy of the existing methods as well as a detailed description of each. Sec. 3 evaluates these methods against two baselines it introduces, the *trivial-static baseline* and the *deep-position-only baseline*. Sec. 5 presents the first part of the root-cause analysis by analyzing the data, introducing the *saliency-only baseline*. Sec. 6 completes the root-cause analysis by analyzing the architectural choices. Sec. 7 presents our reasoning to obtain our new prediction method, TRACK, which establishes state-of-the-art performance. Sec. 8 gives perspective and connects our work to most recent critical re-examinations of deep learning-based approaches for other application domains. Sec. 9 concludes the article.

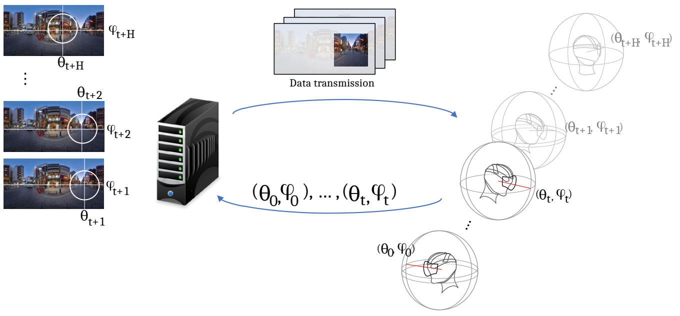

Fig. 1: **360° video streaming principle**. The user requests the next video segment at time t, if the future orientations of the user $(\theta_{t+1}, \varphi_{t+1})$, ..., $(\theta_{t+H}, \varphi_{t+H})$ were known, the bandwidth consumption could be reduced by sending in higher quality only the areas corresponding to the future FoV.

2 Review and taxonomy of existing head prediction methods

This section reviews the existing methods relevant for the problem we consider. We start by formulating the exact problem: it consists, at each video playback time t, in predicting the future user's head positions between t and $t + H$, as illustrated in Fig. 1 and represented in Fig. 2, with the only knowledge of this user's past positions and the (entire) video content. We therefore do not consider methods aiming to predict the entire user trajectory from the start based on the content and on the starting point as, e.g., targeted by the challenge in [16] or summarizing a 360° video into 2D [17], [18]. As well, and importantly, we consider that the users' statistics for the video are *not* known at test time, hence we do not consider methods relying on these per-video statistics, such as [19], [20]. Also, the domain of egocentric videos is related to that of 360° video. However, the assumptions are not exactly the same: only part of the scene and some regions likely to attract the users are available (video shot from a mobile phone), contrary to a 360° video. We therefore do not compare with such works. The problem we tackle is inherently dynamic and aims to help streaming decisions made along the playback. We then present the existing methods and classify them based on the choices of deep neural network architecture. Finally, we provide a detailed description of each method we analyze later in this article.

2.1 Problem formulation

Let us first define some notation. Let $\mathbf{P}_t = [\theta_t, \varphi_t]$ denote the vector coordinates of the FoV at time t. Let $\mathbf{V}_t$ denote the considered visual information at time t: depending on the models' assumptions, it can either be the raw frame with each RGB channel, or a 2D saliency map resulting from a pre-computed saliency extractor. Let T be the video duration. The prediction is not assessed over the first T_{start} seconds of video. To match the settings of the works we compare with, T_{start} is set to 0 sec. for all the curves generated in Sec. 3. In order to skip the exploration phase, as explained in Sec. 5.4, and be more favorable to all methods as they are not able to consider non-stationarity of the motion process, we set $T_{start} = 6$ sec. from Sec. 5 onward. We now refer to Fig. 2. Let H be the *prediction horizon*. We define the terms *prediction step* s, and video *time-stamp* t, such that: at every *time-stamp* $t \in [T_{start}, T]$, we run predictions $\hat{\mathbf{P}}_{t+s}$, for all *prediction steps* $s \in [0, H]$.

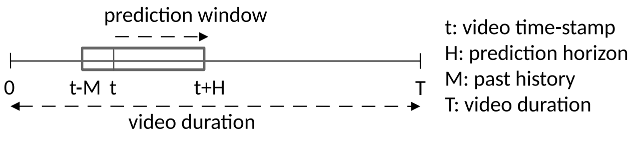

Fig. 2: For each time-stamp t, all the next positions from t until $t + H$ are predicted.

We formulate the problem of trajectory prediction as finding the best model $\mathbf{F}_H^*$ verifying:

$$\mathbf{F}_H^* = \arg\min \mathbb{E}_t \Big[D\Big([\mathbf{P}_{t+1}, \ldots, \mathbf{P}_{t+H}], \\ \mathbf{F}_H \big([\mathbf{P}_t, \mathbf{P}_{t-1}, \ldots, \mathbf{P}_0, \mathbf{V}_{t+1}, \ldots, \mathbf{V}_{t+H}] \big) \Big) \Big]$$

where $D(\cdot)$ is the chosen distance between the ground-truth series of the future positions and the series of predicted positions.

Except for the results in Fig. 6, for each s, we average the errors $dist(\hat{\mathbf{P}}_{t+s}, \mathbf{P}_{t+s})$ over all $t \in [T_{start}, T]$. As considered in the existing methods we compare with, we make H vary between 0.2 sec. and 2.5 sec., then extend H to 5 sec. as detailed from the analysis in Sec. 5.

2.2 Taxonomy

Various approaches to predict user motion in 360° video environments have been published in the last couple of years, and are organized in Table 1. First, for the sake of clarity, each considered method is named with the name of the conference or journal it was published in, appended with the year of publication, as represented in column 1 of Table 1 (starting from the left). They consider different objectives (col. 2), such as predicting the future head position, gaze position or tiles in the FoV. The prediction horizons (col. 3) also span a wide range, from 30ms to 2.5 sec.. Some articles share common datasets for experiments (col. 4), while generally not comparing with each other. Different types of input and input formats are considered (col. 5): some consider the positional information implicitly by only processing the content in the FoV (PAMI18), other consider the position separately, represented as a series of coordinates (e.g., CVPR18) or as a mask (e.g., MM18), with the last sample only (IC3D17) or various length of history, some extract features from the visual content by employing some pre-trained saliency extractors (e.g. NOSSDAV17, MM18) or training end-to-end representation layers made of convolutional and max-pooling layers (e.g., PAMI18). Finally, most of the methods but the first two in Table 1 rely on deep-learning approaches. A key aspect is the way they handle the combination of the positional information (if they consider it individually) with the video content information. As these two types of information are time series, those works all consider the use of deep Recurrent Neural Networks (RNN), and all use Long Short Term Memory (LSTM). However, whether the features are first extracted from each time series independently, or whether the time series samples are first concatenated then fed to a common LSTM, depends on each method. The positioning of the recurrent network in the whole architecture is the multimodality fusion criterion we have selected (col. 6) to order the rows in Table 1 (within each group, methods are ordered from the most recently published), thereby extracting 3 groups of methods:

- if the positional information is not explicitly considered, then no combination is made and a single LSTM processes the content of the FoV: PAMI18;
- combination is made after the single LSTM module in CVPR18: the LSTM processes past positions, and its output gets fused with the video features through a fully connected layer (see Fig. 3-Right);
- if the current saliency map extracted from the content is first concatenated with the current position information, then the LSTM module handles both pieces of information modalities simultaneously: NOSSDAV17, ChinaCom18, MM18 (see Fig. 3-Left).

The architectures tackling this dynamic head motion prediction problem have hence three main objectives: (O1) extracting attention-driving features from the video content, (O2) processing the time series of position, and (O3) combining (fusing) both information modalities to produce the final position estimate. We depict the modules in charge of (O2) and (O3) of methods MM18 and CVPR18 in Fig. 3. The existing methods are described more in detail next and those in bold are selected for comparison with the baselines presented in Sec. 3.

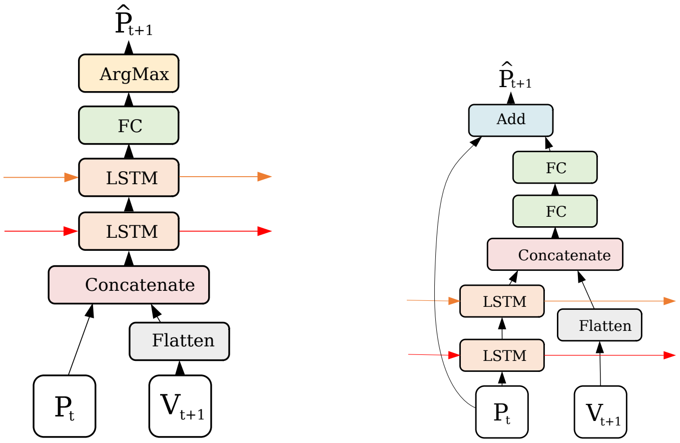

Fig. 3: The building blocks in charge, at each time step, of processing positional information P_t and content information V_t, that are visual features learned end-to-end or obtained from a saliency extractor module (omitted in this scheme). Left: MM18 [11]. Right: CVPR18. [10]

2.3 Detailed description of methods selected for comparison

PAMI18: Xu et al. in [9] design a Deep Reinforcement Learning model to predict head motion. Their deep neural network only receives the viewer's FoV as a 42 × 42 input image, and must decide to which direction and with which magnitude the viewer's head will move. Features obtained from convolutional layers processing each 360° frame cropped to the FoV are then fed into an LSTM to extract direction and magnitude. The training is done end-to-end. The prediction horizon is only one frame, i.e., 30ms. By only injecting the FoV, the authors make the choice not to consider the positional information explicitly as input. The PAMI18 architecture therefore does not feature any specific fusion module. The better performance of our *deep-position-only baseline* shown in Sec. 3 questions this choice.

IC3D17: The strategy presented by Aladagli et al. in [21] simply extracts saliency from the current frame with an off-the-shelf method, identifies the most salient point, and predicts the next FoV to be centered on this most salient point. It then builds recursively. We therefore consider that this method to be a sub-case of PAMI18, and that the comparison with PAMI18 is thus more relevant.

ICME18: Ban et al. in [22] assume the knowledge of the users' statistics, and hence assume more information than our problem definition, which is to predict the user motion only based on the user's position history and the video content. We therefore do not consider this architecture for comparison. A linear regressor is first learned to get a first prediction of the displacement, which it then adjusts by computing the centroid of the k nearest neighbors corresponding to other users' positions at the next time-step.

CVPR18: In [10], Xu et al. predict the gaze positions over the next second in 360° videos based on the gaze coordinates in the past second and the video content. As depicted in Fig. 3-Right, the time series of past head coordinates is processed by a doubly-stacked LSTMs. For the video information, spatial and temporal

Reference	Objective	Prediction horizon	Dataset	Inputs	RNN before/after concatenation of modalities
PAMI18 [9]	head coordinates	30ms	76 videos, 58 users	frame cropped to FoV	N/A (no fusion)
IC3D17 [21]	head coordinates	2s	16 videos, 61 users	Pre-trained sal. in FoV	N/A (no fusion, no LSTM)
ICME18 [22]	tiles in FoV	6s	18 videos, 48 users	Position history, users' distribution	N/A (no LSTM)
CVPR18 [10]	gaze coordinates	1s	208 videos, 30+ users	Video frame, position history as coordinates	before
MM18 [11]	tiles in FoV	2.5s	11 videos, 48+ users from [23], [24] with custom pre-processing	Pre-trained saliency, mask of positions	after
ChinaCom18 [12]	tiles in FoV	1s	NOSSDAV17's dataset	Pre-trained saliency, FoV tile history	after
NOSSDAV17 [13]	tiles in FoV	1s	10 videos, 25 users	Pre-trained saliency, FoV position or tile history	after

TABLE 1: Taxonomy of existing dynamic head-prediction methods. References in bold are considered for comparison in Sec. 3.

saliency maps are first concatenated with the RGB image, then fed to Inception-ResNet-V2 to obtain the "saliency features" denoted as $Vt+1$ in Fig. 3-Right. They formulate the gaze prediction problem the same way as the head prediction problem.

MM18: Nguyen et al. in [11] first construct a saliency model based on a deep convolutional network and named PanoSalNet. The so-extracted saliency map is then fed, along with the position encoded as a mask, into a doubly-stacked LSTM, as shown in Fig. 3-Left.

ChinaCom18: Li et al. in [12] present a similar approach as MM18, adding a correction module to compensate for the fact that tiles predicted to be in the FoV with highest probability may not correspond to the actual FoV shape (having even disconnected regions). This is a major drawback of the tile-based approaches as re-establishing FoV continuity may significantly impact final performance.

NOSSDAV17: Fan et al. in [13] propose two LSTM-based networks, predicting the likelihood that tiles pertain to future FoV. Visual features extracted from a pre-trained VGG-16 network are concatenated with positional information, then fed into LSTM cells for the past M time-steps, to predict the head orientations in the future H time-steps. Similarly to MM18 and as depicted in Fig. 3-Left, the building block of NOSSDAV17 first concatenates flattened saliency map and position, and feeds it to a doubly-stacked LSTM whose output is post-processed to produce the position estimate. An extended version of this work has been published in [25].

These methods therefore make for a wide range of deep network architectural choices. In particular the fusion problem (O3), defined in Sec. 2.2, may be handled differently. MM18 and CVPR18 are selected as representatives: combining both modalities before or after the recurrent (LSTM) unit, respectively. There is no pairwise comparison between any of the above works. From the above articles, the only works which provided their code and their deep neural networks for reproducibility are PAMI18 and MM18. However, we could obtain all the datasets to compare with all (the datasets not publicly available were kindly shared by the authors whom we have contacted).

2.4 Description of datasets from the literature for comparison

This information is summarized in Table 1, and detailed here:

PAMI18: This dataset contains both head movement and eye movement data of 58 subjects on 76 360° videos of variable duration, from 10 to 80 seconds (25 seconds in average).

CVPR18: It is made of 208 360° videos between 15 and 80 seconds (36s in average), each video is viewed by at least 31 participants.

NOSSDAV17: This dataset consists of 10 360° videos with a duration of 60 seconds, along with the identification number of the tiles that overlap with the FoV of the viewer according to the head orientation data (the tile size considered is 192 × 192). This dataset contains the traces of 50 participants, however, for the experiment performed in [13], the traces of 25 participants only were used.

MM18: The dataset used in MM18 consists on the post-processing of two publicly available datasets [23], [24]. The first dataset [24] includes 18 videos viewed by 48 users, from which 9 videos are selected. The second dataset [23] has five videos viewed by 59 users, from which 2 videos are used. From the chosen videos, a segment is selected such that there are one or more events that introduce a new salient region (e.g. a scene change).

MMSys18: We also considered the dataset presented by David et al. in [26] and referred to as MMSys18. It is made of 19, 360° videos of 20 seconds, along with the head positions of 57 participants starting their exploration at a random angular position.

3 Comparison against two baselines: trivial-static and deep-position-only

To compare the above recent proposals (PAMI18, CVPR18, MM18, ChinaCom18, NOSSDAV17) to a common reference, we first introduce the *trivial-static baseline*. First, we show that all these methods on their original settings, metrics and datasets, are outperformed by this trivial baseline. This is surprising and raises the question of whether it is actually possible to learn anything meaningful with these settings (datasets and prediction horizons).

To answer this question, we then introduce a *deep-position-only baseline*, that we design as a sequence-to-sequence LSTM-based architecture exploiting the time series of past positions only (disregarding the video content). We show this new baseline is indeed able to outperform the *trivial-static baseline* (establishing state-of-the-art performance). Later, Sec. 5 introduces a *saliency-only baseline*.

3.1 Definition of the trivial-static baseline

Different linear predictors can be considered as baselines. We consider here the simplest one which predicts no motion: $[\hat{\mathbf{P}}_{t+1}, \ldots, \hat{\mathbf{P}}_{t+H}] = [\mathbf{P}_t, \ldots, \mathbf{P}_t]$.

More complex baselines exist. For example in [27], a Linear Regressor and a Neural Network perform better than the *trivial-static baseline*. However, as we will see, all existing methods trying to leverage both the video content and the position to predict future positions perform worse than the *trivial-static baseline*, without exception.

3.2 Design of the deep-position-only baseline

We now present an LSTM-based predictor which considers positional information only. An LSTM enables non-linear shape of the motion and the memory effect due to inertia, as discussed in [10] and shown by the generated trajectories in [14]. We select a sequence-to-sequence (seq2seq) architecture because it has proven powerful at capturing complex dependencies and generating realistic sequences, as shown in text translation for which it has been introduced [28].

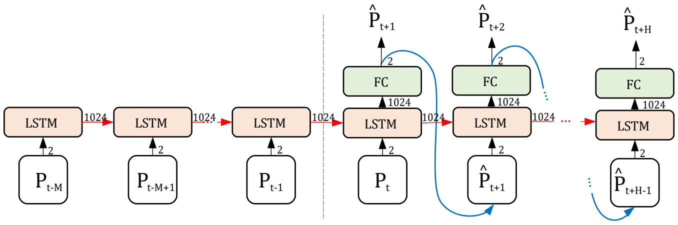

Fig. 4: The *deep-position-only baseline* based on an encoder-decoder (seq2seq) architecture.

As depicted in Fig. 4, a seq2seq framework consists of an encoder and a decoder. The encoder receives the *historic window* input (samples from $t-M$ to $t-1$ shown in Fig. 2) and generates an internal representation. The decoder receives the output of the encoder and progressively produces predictions over the target horizon, by re-injecting the previous prediction as input for the new prediction time-steps. This is a strong baseline (not only a trivial-static or linear predictor) processing the head coordinates only. We have optimized the *deep-position-only baseline* as described in [14] and [29, Sec. I]. This baseline has been inspired from the work of Martinez et al. in [30], which re-examined major deep networks as multi-modal fusion methods, combining video content and motion time series for 3D human skeleton motion prediction. Their findings that all state-of-the-art methods were worse than a simple baseline, have echoed and corroborate with our own findings for the problem of multi-modal fusion methods for head motion prediction as detailed below. We give more perspective on this aspect in Sec. 8.

Reproducibility: All the additional details of implementation are described in the supplemental material joined to the submission [29, Sec. II]. We emphasize that the entire reproducible framework, with all methods including baselines, homogeneized datasets and common metrics, has been published in [15], obtained an ACM reproducibility badge, and is available in full at [14].

3.3 Results

We now present the comparisons of the state-of-the-art methods presented in Sec. 2.2 with the *trivial-static baseline* and *deep-position-only baseline* defined above. We report the exact results of the original articles, along with the results of our baselines, the *deep-position-only baseline* being trained and tested on the exact same train and test subsets of the original dataset as the original method (there is no training for the *trivial-static baseline*). The benchmark metrics (related to predicting head or gaze positions, or FoV tiles) are those from the original articles, so are the considered prediction horizons H.

Results for PAMI18 are shown in Table 2, for CVPR18 in Fig. 5-Bottom, for MM18 in Fig. 5-Top, for ChinaCom18 in Table 3 and for NOSSDAV17 in Table 4. Let us mention that none of these methods considered baselines identical to the *trivial-static baseline* and *deep-position-only baseline* defined above. All perform worse than both our *trivial-static* and *deep-position-only baselines*. Specifically, all but one (CVPR18) perform significantly worse.

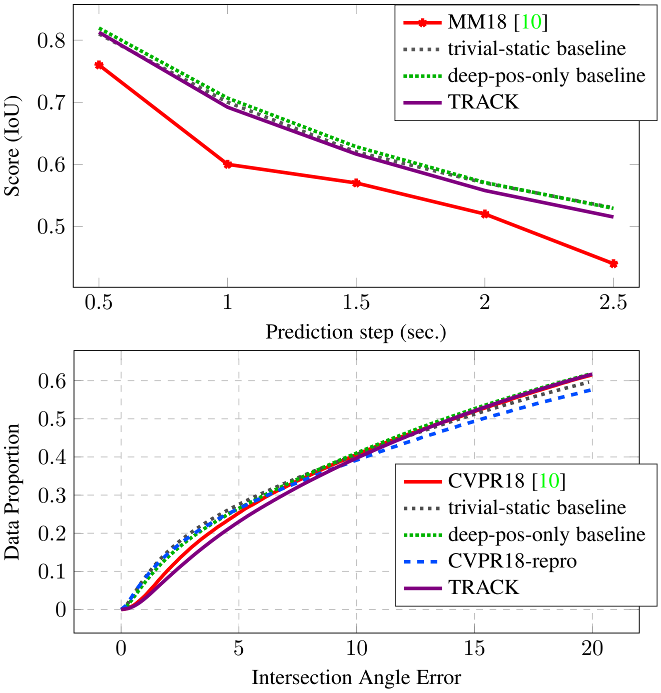

Fig. 5: **Top**: Comparison with MM18 [11], H =2.5 seconds. **Bottom**: Comparison with CVPR18 [10], prediction horizon $H = 1$ sec. CVPR18-repro is introduced in Sec. 4.1, the model TRACK in Sec. 6.2.

We define below the metrics used for every considered predictor:

• **NOSSDAV17** [13] considers the following metrics:

− *Accuracy*: ratio of correctly classified tiles to the union of predicted and viewed tiles.

− *Ranking Loss*: number of tile pairs that are incorrectly ordered by probability normalized to the number of tiles.

Method	KingKong	SpaceWar2	StarryPolar	Dancing	Guitar	BTSRun	InsideCar	RioOlympics	Average
PAMI18 [9]	0.809	0.763	0.549	0.859	0.785	0.878	0.847	0.820	0.753
trivial-static baseline	0.974	0.963	0.906	0.979	0.970	0.983	0.976	0.966	0.968
deep-position-only baseline	**0.983**	**0.977**	**0.930**	**0.984**	**0.977**	**0.987**	**0.982**	**0.976**	**0.977**
TRACK	0.974	0.964	0.912	0.978	0.968	0.982	0.974	0.965	0.968
Method	SpaceWar	CMLauncher2	Waterfall	Sunset	BlueWorld	Symphony	WaitingForLove		Average
PAMI18 [9]	0.626	0.763	0.667	0.659	0.693	0.747	0.863		0.753
trivial-static baseline	0.965	0.981	0.973	0.964	0.970	0.968	0.978		0.968
deep-position-only baseline	**0.976**	**0.989**	**0.984**	**0.973**	**0.979**	**0.976**	**0.982**		**0.977**
TRACK	0.965	0.981	0.972	0.964	0.970	0.969	0.977		0.968

TABLE 2: Comparison with PAMI18 [9]: Mean Overlap scores of FoV prediction, prediction horizon $H \approx 30ms$ (1 frame). The model TRACK is introduced in Sec. 6.2.

	trivial-static baseline		*deep-position-only baseline*		ChinaCom18	
	Accuracy	F-score	Accuracy	F-score	Accuracy	F-score
Hog Rider	96.29%	0.8858	**96.97%**	**0.9066**	77.09%	0.2742
Driving with	95.96%	0.8750	**96.59%**	**0.9843**	77.34%	0.2821
Shark Shipwreck	95.23%	0.8727	**96.12%**	**0.8965**	83.26%	0.5259
Mega Coaster	97.20%	0.9144	**97.71%**	**0.9299**	88.90%	0.7011
Roller Coaster	96.99%	0.9104	**97.50%**	**0.9256**	88.28%	0.6693
Chariot-Race	97.07%	0.8802	**96.91%**	**0.9056**	87.79%	0.6040
SFR Sport	96.00%	0.8772	**96.91%**	**0.9054**	89.29%	0.7282
Pac-Man	96.83%	0.8985	**97.16%**	**0.9089**	87.45%	0.6826
Peris Panel	95.60%	0.8661	**96.54%**	**0.8947**	89.12%	0.7246
Kangaroo Island	95.35%	0.8593	**96.54%**	**0.8954**	82.62%	0.5308
Average	96.15%	0.8840	**96.90%**	**0.9063**	72.54%	0.5155

TABLE 3: Comparison with ChinaCom18 [12], prediction horizon $H = 1$ second.

Method	Accuracy	F-Score	Rank Loss
NOSSDAV17-Tile [13]	84.22%	0.53	0.19
NOSSDAV17-Orient. [13]	86.35%	0.62	0.14
trivial-static baseline	95.79%	0.87	0.10
deep-position-only baseline	**96.30%**	**0.89**	**0.09**
TRACK	95.48%	0.85	0.15

TABLE 4: Comparison with NOSSDAV17: Performance of Tile- and Orientation-based networks of [13] compared against our *deep-position-only baseline*, prediction horizon $H = 1$ second. The model TRACK is introduced in Sec. 6.2.

— *F-Score*: harmonic mean of *precision* and *recall*, where *precision* is the ratio of correctly predicted tiles by the total number of predicted tiles, and *recall* is the ratio of correctly predicted tiles by the number of viewed tiles.
Let us point out here that the tile data is not balanced, as more tiles pertain to class 0 (tile $\notin$ FoV) than to class 1 (tile $\in$ FoV) owing to the restricted size of the FoV compared to the complete panoramic size. If we predict all the tiles systematically in class 0, the accuracy already gets to 83.86%. The accuracy is indeed known to be a weak metric to measure the performance of such unbalanced datasets.

- **PAMI18** [9] uses as metric the Mean Overlap (MO) defined as:

$$MO = \frac{A(FoV_p \cap FoV_g)}{A(FoV_p \cup FoV_g)}$$

Where FoV_p is the predicted FoV, FoV_g is the ground-truth FoV, and $A(\cdot)$ is the area of a panoramic region.

- **CVPR18** [10] uses the Intersection Angle Error IAE for each gaze point (θ, φ) and its prediction $(\hat{\theta}, \hat{\varphi})$, defined as $IAE = arccos(\langle P, \hat{P} \rangle)$, where P is the 3D coordinate in the unit sphere: $P = (x, y, z) = (cos(\theta)cos(\varphi), cos(\theta)sin(\varphi), sin(\theta))$. Let us mention that CVPR18 also considers a deep-position-only baseline. However, ours appears stronger, likely due to the seq2seq architecture. We readily apply our different predictors on the gaze data available in the CVPR18-dataset.
- **MM18** [11] takes the tile with the highest viewing probability as the center of the predicted viewport, and assigns it and all the neighboring tiles that cover the viewport, with label 1. Tiles outside the viewport are assigned 0. Then, the score is computed on these labels as $IoU = TP/TT$ (True Positive TP, True Total TT), the intersection between prediction and ground-truth of tiles with label 1 (TP) over the union of all tiles with label 1 in the prediction and in the ground-truth (TT).
- **ChinaCom18** [12] uses the Accuracy and F-Score on the labels assigned to each predicted tile.

4 Root cause analysis: the metrics in question

We have shown that the existing methods assessed above, which try to leverage both positional information and video content to predict future positions, perform worse than a simple baseline assuming no motion, which in turn can be outperformed by the *deep-position-only baseline* (considering only positional information). This section and the next two (Sec. 5 & Sec. 6) aim to identify the reasons why the existing approaches perform worse than the baselines. In this part, we focus on the possible causes due to the evaluation, specifically asking:

Q1 Metrics: Can the methods perform better than the baselines for some specific videos or pieces of trajectories?

4.1 Evaluation Metrics

Let us first describe the losses and evaluation metrics considered from now on. The prediction of the FoV motion can be cast as a classification problem, where pixels or tiles are classified in or out of future FoV (as done in NOSSDAV17, MM18, ChinaCom18). However, this problem is inherently imbalanced. Therefore, for the analysis, we choose to keep the original formulation as a regression problem. The tracking problem on a sphere can be assessed by different distances. Given two points on the surface of the unit sphere $P_1 = (\theta_1, \varphi_1)$ and $P_2 = (\theta_2, \varphi_2)$, where θ is the longitude and φ is the latitude of the point, possible distances are:

- Mean squared error $= ((\theta_1 - \theta_2)^2 + (\varphi_1 - \varphi_2)^2)/2$

- Angular error $= \sqrt{\arctan(\sin(\Delta\theta)/\cos(\Delta\theta))^2 + (\varphi_1 - \varphi_2)^2}$, where $\Delta\theta = \theta_1 - \theta_2$
- Orthodromic distance $= \arccos\left(\cos\left(\varphi_1\right)\cos\left(\varphi_2\right)\cos\left(\Delta\theta\right) + \sin\left(\varphi_1\right)\sin\left(\varphi_2\right)\right)$ which is a reformulation of:

$$D(P_1, P_2) = \arccos\left(\vec{P_1} \bullet \vec{P_2}\right), \tag{1}$$

where $\bullet$ is the dot product operation, and $\vec{P_1}$ are the coordinates in the unit sphere of point P. Indeed, for a point $P_1 = (\theta_1, \varphi_1)$, the coordinates in the unit sphere are then given by $\vec{P_1} = (\cos\theta_1 \cos\varphi_1, \sin\theta_1 \cos\varphi_1, \sin\varphi_1)$.
The latter two metrics are able to handle the periodicity of the latitude, which the first one cannot. The difference between *angular error* and *orthodromic distance* is that the latter computes the distance on the surface of the sphere, while the *angular error* computes the error of each angle independently. Finally, owing to its adequacy to the tracking problem on the unit sphere, we choose the **orthodromic distance** as the test metric in our approach.

4.2 Q1: Can the methods perform better than the baselines for some specific pieces of trajectories or videos?

The metrics used in Sec. 3 are averages over time trajectories and videos. The question we ask is whether the methods can perform better than the baselines for some specific pieces of trajectories or videos.

4.2.1 *Specific pieces of trajectory*

To evaluate whether the existing methods perform better than the baselines in some specific pieces of the trajectory, we adopt the same approach as in [31, Sec. 4], introducing the *Average non-linear displacement error* as a metric to evaluate the error around the non-linear regions of the trajectory where most errors occur owing to human-content interactions. We therefore quantify the *difficulty of prediction* with the second derivative of the trajectory, i.e., the radius of curvature. To obtain detailed results (for each instant of time of each user and video pair), we re-implement CVPR18 with the exact same architectural and training parameters as those described in the article [10].[1] The curve CVPR18-repro in Fig. 5-Bottom shows that we obtain similar results on the original dataset (higher on the first half of the truncated CDF, then slightly lower on the second half of the truncated CDF). This confirms the validity of our re-implementation. Fig. 6-Left depicts the distribution of the prediction difficulty. Fig. 6-Right shows that for every difficulty range, CVPR18-repro is not able to improve the prediction over the baselines. Considering CVPR18 and MM18 the two representative and best performing methods in Sec. 3 (apart from the baselines), for the sake of space we also report the results for MM18 in the supplemental material in [29, Sec. IV]. We obtained similar qualitative results with MM18.

We conclude that for more difficult parts of the trajectory, the CVPR18-repro or MM18 methods are not able to improve over the baselines.

4.2.2 *Specific videos*

Fig. 6-Left shows that the majority of the data is in the 0-1 difficulty range, therefore, we can think the models have difficulty to pay attention to the rarer cases of trajectory pieces where the prediction difficulty is higher. To evaluate whether the existing methods perform better than the baselines when the dataset (train and test sets) is properly balanced with videos where the content is proved to help, we consider the dataset prepared in Sec. 6.1. The details on the usefulness of the content are given in Sec. 5. The performance of CVPR18-repro and MM18-repro on this dataset can be seen in Fig. 13 in average and per test video in [29, Sec. V]: they are never able to take advantage of the content as they are systematically outperformed by the *deep-position-only baseline* (even for the videos where the saliency is proved useful).

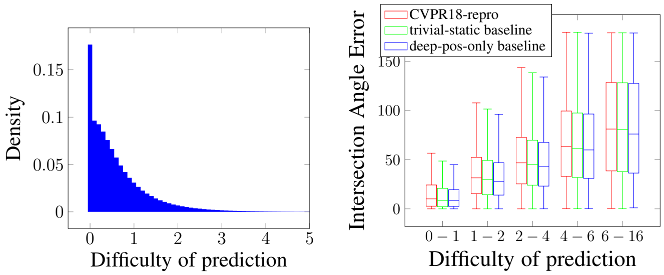

Fig. 6: Left: Distribution of difficulty in the CVPR18 dataset. Right: Error as a function of the difficulty for the CVPR18-repro model.

Answer to Q1: No, the methods considering the video content do not perform better than the *deep-position-only baseline* for specific pieces of trajectories or videos where the knowledge of the content should improve the prediction.

5 Root Cause Analysis: The Data in Question

In this section, we focus on the possible causes due to the data. In Sec. 6, we analyze the possible architectural causes. This section therefore aim to answer question Q2, whose answer is provided at the end of the section:

Q2 Data: Do the datasets (made of videos and motion traces) match the design assumptions the methods build on?
To answer Q2, we consider the assumptions at the core of the existing architectures attempting to leverage the knowledge of position history and video content, and break them down into :

- Assumption (A1): the position history can inform the prediction of future positions
- Assumption (A2): the visual content can inform the prediction of future positions

We identify whether these assumptions hold in the datasets and settings considered by the existing methods (Sec. 5.1 and 5.4), and introduce a new *saliency-only baseline* to do so (Sec. 5.2).

5.1 Assumption (A1): the position history is informative of future positions

The amount of information held by a process about another one can be quantified by the Mutual Information (MI). This in turns informs on the degree of predictability of the target process using the first process. MI has been used in [32] for inter-user analysis. Here, we define the MI between head positions of a given user at time t and $t+s$ by $I(P_t; P_{t+s}) = D_{KL}(Pr[P_t, P_{t+s}] || Pr[P_t] \otimes Pr[P_{t+s}])$, where $D_{KL}(\cdot)$ and $\otimes$ stand for the Kullback–Leibler

[1] We had to replicate the architecture of CVPR18 because we could not find any official code and the authors did not reply to our emails. Our reproduced code is available online at [14] and detailed in [29, Sec. III].

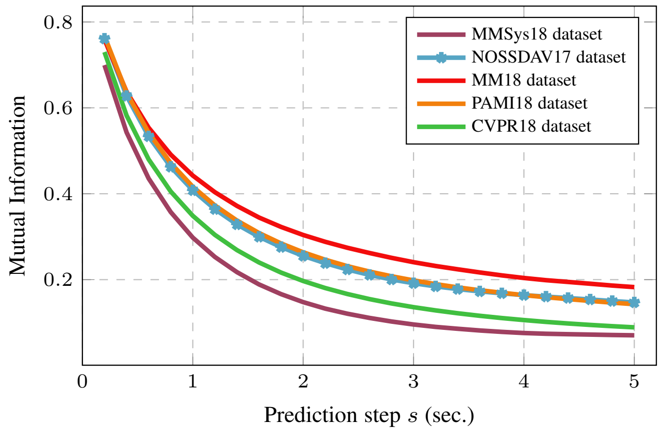

Fig. 7: Mutual information $I(P_t; P_{t+s})$ between P_t and P_{t+s} (averaged over t and videos) for all the datasets used in NOSSDAV17, PAMI18, CVPR18 and MM18, with the addition of MMSys18.

divergence and convolutional product, respectively. For each of the datasets considered in PAMI18, CVPR18, MM18, NOSSDAV17, and MMSys18, Fig. 7 represents MI normalized and averaged over all videos and time stamp t, as a function of $s \in [0, H = 5sec.]$. The 2D-coordinates have been discretized in 128 bins. This figure shows that position at time $t + s$ can be predicted to a significant degree by P_t when s is low (e.g., lower than 2 sec.). As expected, the further away the prediction step, the lowest the predictability of P_{t+s} from P_t. In [29, Sec. VI], we also relate MI with a more intuitive characterization of the datasets, showing that the amount of user's motion is generally low, except in the MMSys18 dataset. *Does Assumption (A1) hold?*: On the datasets and prediction horizons considered in the literature ($H \leq 2$ sec.), the position history is therefore strongly informative of the next positions. Another element supporting this observation is the best performance obtained by our baseline exploiting position only (see Sec. 3 above). A similar study was conducted in [27] showing that the viewer motion has a strong temporal auto-correlation.

5.2 Definition of the saliency-only baseline

To analyze Assumption A2 in Sec. 5.4 and assess how much gain can the consideration of the content bring to the prediction, we first define a so-called *saliency-only baseline*. This baseline is defined from an attentional heat map, either extracted from the visual content (heat map then named Content-Based saliency) or directly from the position data of all the users (heat map then named Ground-Truth saliency). For either type of heat map, the *saliency-only baseline* provides an upper-bound on the prediction error that a more refined processing of the heat map, in combination with the past positions, would make. In this section, we only consider heat maps obtained from the users data, we therefore start by defining such heat maps. Only in Sec. 6.2 do we use the heat maps estimated from the video content.

5.2.1 Definition of the ground-truth saliency

To be independent from the imperfection of any saliency predictor fed with the visual content, we consider here the ground-truth saliency: it is the heat map (2D distribution) of the viewing patterns, obtained at each point in time from the users' traces. To compute the ground-truth saliency maps, we consider the point at the center of the viewport $P^t_{u,v}$ for user $u \in U$ and video $v \in V$ at time-stamp $t \in [0, T]$, where T is the length of the trace. For each head position $P^t_{u,v}$, we compute the orthodromic distance $D(\cdot)$ from $P^t_{u,v}$ to each point $Q_{x,y}$ at longitude x and latitude y in the equirectangular frame. Then, we use a modification of the radial basis function (RBF) kernel shown in Eq. 2 to convolve the points in the equirectangular frame and obtain the Ground-Truth Saliency (GT_Sal) for user u on video v at time t in image location (x, y):

$$GT\_Sal^t_{u,v,x,y} = \exp\left(-\frac{D(P^t_{u,v}, Q_{x,y})^2}{2\sigma^2}\right), \qquad (2)$$

where $D(P^t_{u,v}, Q_{x,y})$ is the orthodromic distance, computed using Eq. 1. A value of $\sigma = 6°$ is chosen so that the ground-truth saliency maps look qualitatively similar to those of PanoSalNet [11] used in Sec. 6.2. We compute saliency maps $GT\_Sal^t_{u,v}$ per user $u \in U$, video $v \in V$ and time-stamp t by convolving each head position $P^t_{u,v}$ with the modified RBF function in Eq 2. The saliency map at time t of video v is calculated as $GT\_Sal^t_v = \frac{1}{U}\sum_{u \in U} GT\_Sal^t_{u,v}$, where U is the total number of users watching this video.

5.2.2 Definition of the K-saliency-only baseline

We extract the K highest peaks of the heat map for every prediction step $t + s$ (for all t, for all $s \in [0, H]$). At every $t + s$, the *K-saliency-only baseline* predicts $\hat{\mathbf{P}}_{t+s}$ as the position of the peak, amongst the K peaks, which is closest to the last known user's position $\mathbf{P}_t$.

Fig. 8 and 9 show the prediction error of the *K-saliency-only baseline* for $K = 1, 2, 5$. For low s, we verify that the higher K, the lower the error close to time t, because the more the number of points of interest possibly considered. However, as the prediction step s increases and $t + s$ gets away from t, the error is lower for lower K. Indeed, if the user moves, then she is more likely to get closer to a more popular point of interest, that is to a higher-ranked peak.

5.2.3 Definition of the saliency-only baseline

As mentioned in the beginning of the section, each ***K-saliency-only baseline*** can be considered as an upper-bound on the error that the best predictor optimally combining position and content modality could get. Therefore, for a given κ, we define the *saliency-only baseline* as the minimum of these *K-saliency-only baseline*, for $K \in [1, \kappa]$ and for every s in $[0, H]$. In this article, we set $\kappa = 5$. The *saliency-only baseline* is shown in red in Fig. 12. From Fig. 12, we do not represent the *K-saliency-only baselines* anymore, but only the *saliency-only baseline*.

5.3 Background on human attention in VR

Before analyzing Assumption (A2), let us first provide some characteristics of the human attention in VR identified recently. It has been recently shown in [33] and [34] that, when presented with a new VR scene (the term "scene" is defined by Magliano and Zacks in [35] as a period of the video between two edits with space discontinuity), a human first goes through an exploratory phase that lasts for about 10 to 15 sec. ([34, Fig. 18], [33, Fig. 2]), before settling down on so-called Regions of Interest (RoIs), that are salient areas of the content. The duration and amplitude of exploration, as well as the intensity of RoI fixation, depend

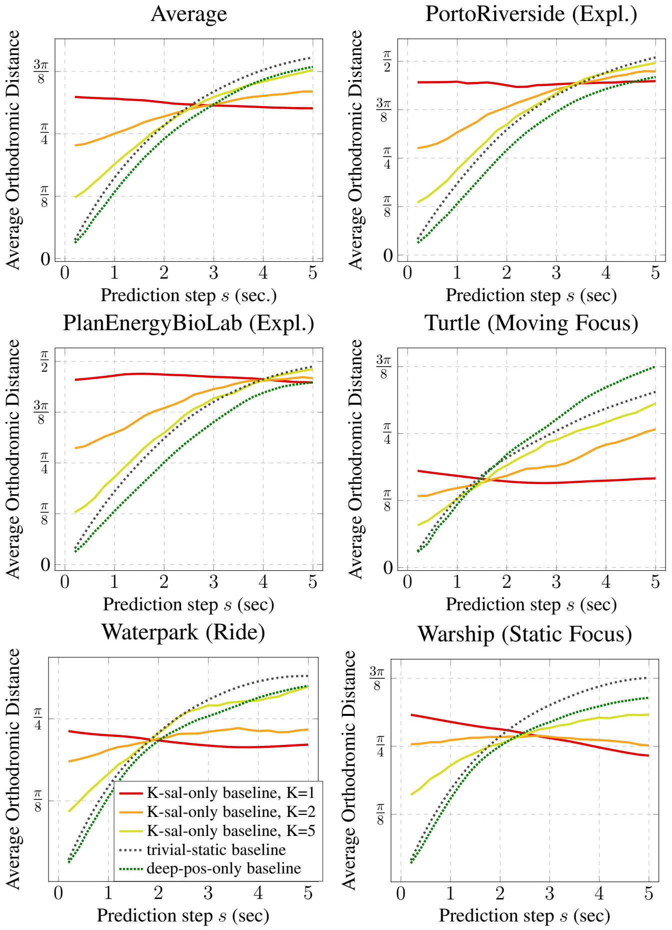

Fig. 8: Prediction error on the MMSys18 dataset. The *deep-position-only baseline* is tested on the 5 videos above, and trained on the others (see [29, Sec. I] or [14], [15]). Top left: Average results on all 5 test videos. Rest: Detailed result per video category (Exploration, Moving Focus, Ride, Static Focus). Legend is identical in all sub-figures.

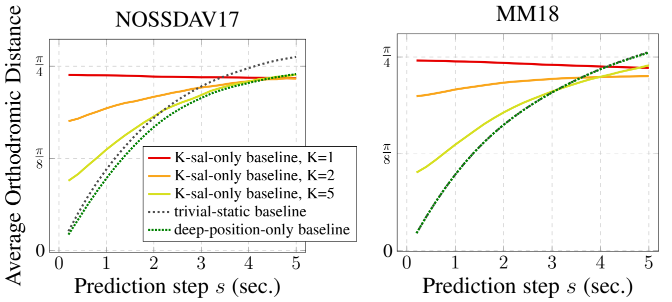

Fig. 9: Prediction error averaged on test videos of the datasets of NOSSDAV17 (left) and MM18 (right). We refer to the supplemental material [29, Sec. II] or [14] for the train-test video split used for the *deep-position-only baseline* (identical to original methods). Legend is identical in both sub-figures.

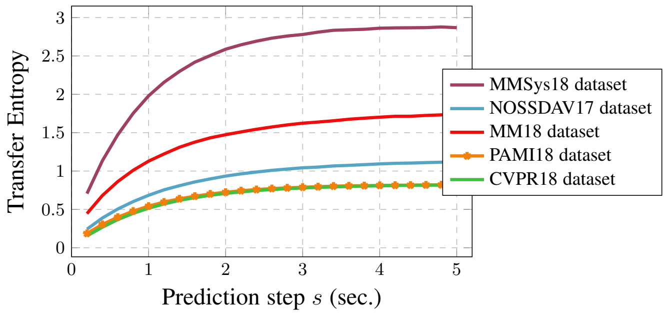

Fig. 10: Transfer Entropy (TE) $TE_{V\to P}(t,s)$ between V_{t+s} and P_{t+s} (averaged over t and videos) for all the datasets used in NOSSDAV17, PAMI18, CVPR18 and MM18, with the addition of MMSys18.

on the video content itself. Almquist et al. [34] have identified the following main video categories for which they could discriminate significantly different users' behaviors: *Exploration, Static focus, Moving focus* and *Rides*. In *Exploration* videos, the spatial distribution of the users' head positions tends to be more widespread, making harder to predict where the users will watch and possibly focus on. *Static focus* videos are made of a single salient object (e.g., a standing-still person), making the task of predicting where the user will watch easier in the focus phase. In *Moving focus* videos, contrary to *Static focus* videos, the RoIs move over the sphere and hence the angular sector where the FoV will be likely positioned changes over time. *Rides* videos are characterized by substantial camera motion, the attracting angular sector being likely that of the direction of the camera motion.

5.4 Assumption (A2): the visual content is informative of future positions

We now analyze whether this assumption (A2) holds, and for which settings (datasets, prediction horizons). As for (A1), we first quantify how much additional information can be gained on P_{t+s} by knowing the visual content V_{t+s} at time $t+s$, given we already know the past positions. This corresponds to the conditional MI $I(P_{t+s};V_{t+s}|P_t)$, also named Transfer Entropy (TE) and satisfying for every video: $TE_{V\to P}(t,s) = I(P_{t+s};V_{t+s}|P_t) = H(P_{t+s}|P_t) - H(P_{t+s}|P_t,V_{t+s})$, where $H(\cdot)$ denotes the entropy. TE has been used in [32] but not with saliency data. Fig. 10 represents $TE_{V\to P}(t,s)$ averaged over all time stamps t and videos of every dataset. The 2D-coordinates have been discretized in 128 bins and V_{t+s} is taken as the Content-Based saliency defined in Sec. 6.2, the probability values being discretized into 256 bins. The TE values cannot be compared across the datasets, but the important observation is that the TE value triples from $s = 0$ to $s = 5$ sec. It shows that the predictability of future positions from the content, conditioned on the position history, is initially low then increases with s. The results of MI in Fig. 7 and TE in Fig. 10 therefore show that short-term motion is mostly driven by inertia from t, while the content saliency may impact the trajectory in the longer-term. To cover both short-term and long-term, we set the prediction horizon $H = 5$ sec.. We confirm this and better quantify the durations of both phases for the different video categories in the next results. We analyze A2 on the datasets used in NOSSDAV17, MM18, CVPR18 and PAMI18. We also consider an additional dataset, referred to as MMsys18-dataset [26]. All these datasets are detailed in Sec. 2.4. In MMsys18-dataset, the authors show that the exploration phase in their videos lasts between 5 and 10s, and show that after this initial period, the different users' positions have a correlation coefficient reaching 0.4 [26, Fig. 4]. This dataset is made of 12

Exploration videos, 4 Static focus videos (Gazafisherman, Sofa, Mattswift, Warship), 1 Moving focus video (Turtle) and 2 Ride videos (Waterpark and Cockpit). Fig. 8, 9, 12 and 11 depict the prediction error for prediction steps $s \in [0, H = 5$ sec.$]$, obtained with the *deep-position-only baseline* and *saliency-only baseline* on the 4 previous datasets. We remind that each point for every given step s, is an average over all the users and all time-stamp $t \in [T_{start}, T]$, with T the video duration and $T_{start} = 6$ sec. from now on to skip the initial exploration phase (presented right above in the beginning of this Sec. 5.4) and ensure that the content can be useful for all time-stamps t. By analyzing the *saliency-only baseline* for every prediction step s (saliency baseline in red in Fig. 12), the same phenomenon can be observed on all the datasets: the *saliency-only baseline* has a higher error than the *deep-position-only baseline* for prediction steps s lower than 2 to 3 seconds. This means that there is no guarantee that the prediction error over the first 2 to 3 seconds can be lowered by considering the content. After 2 to 3 sec., on non-Exploration videos, we can see that relevant information can be exploited from the heat maps to lower the prediction error compared to the *deep-position-only baseline*. When we isolate the results per video type, e.g., in Fig. 8, for Exploration (PortoRiverside, PlanEnergyBioLab), a Ride (WaterPark) a Static focus (Warship) and a Moving focus (Turtle) videos, we observe that the saliency information can significantly help predict the position for prediction steps beyond 2 to 3 seconds.

We therefore conclude by answering

Q2 Data: Do the datasets (made of videos and motion traces) match the design assumptions the methods build on?

5.5 Q2: Do the datasets (made of videos and motion traces) match the design assumptions the methods build on?

Answer to Q2:

- Study of MI for assumption A1 confirms that the level of predictability of short-term position from past position is significant, corresponding to the inertia effect and frequent low velocity in some datasets.
- Considering the ground-truth saliency (attentional heat maps), we conclude on A2 by stating that considering the content in the prediction can significantly help for non-Exploration videos if the prediction horizon is longer that 2 to 3 sec.. There is no guarantee it can significantly or easily help for shorter horizons. All the selected existing works considered prediction horizons lower than 2.5 sec., making it very unlikely to outperform the *deep-position-only baseline*.

Having shown it is difficult to outperform the *deep-position-only baseline* on these short horizons, next we investigate why most existing methods are however not able to match its performance.

6 Root cause analysis: the architectures in question

In Sec. 5, we have analyzed the possible causes for the weakness of the existing predictors, related to the metrics and the assumptions on the dataset. As they do not suffice to explain the counter-performance of the existing methods compared with single-modality baselines, in this section, we state and analyze the possible architectural causes. Let us recall the three main objectives a prediction architecture must meet, as stated in Sec. 2.2: (O1) extracting attention-driving features from the video content, (O2) processing the time series of position, and (O3) fusing both information modalities to produce the final series of position estimates. Note that this is a conceptual description, and does not necessarily correspond to a processing sequence: fusion (O3) can be performed from the start and O1 and O2 may not be performed in distinguishable steps or elements, as it is the case in NOSSDAV17 or MM18.

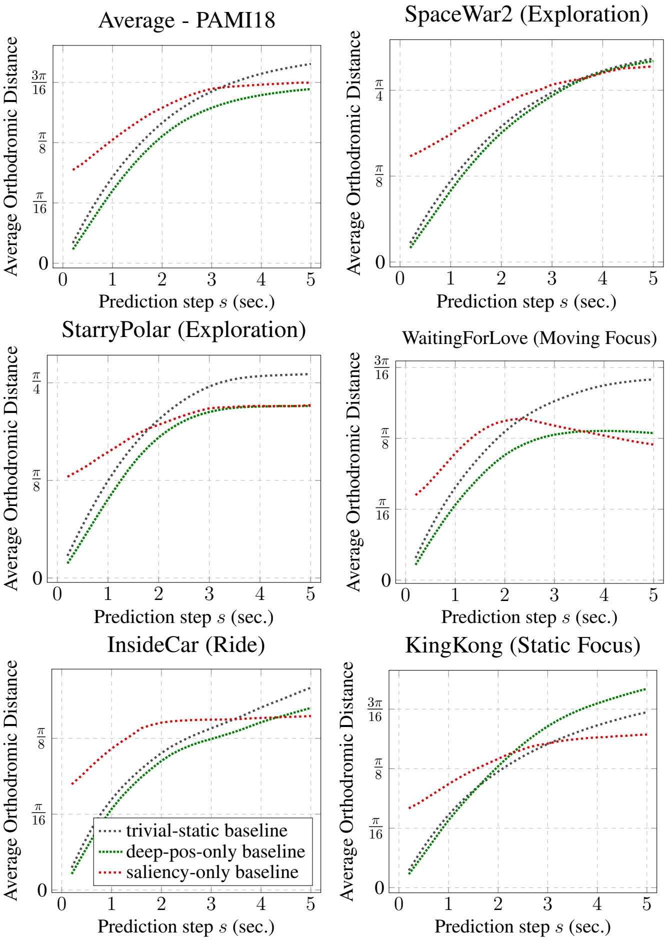

Fig. 11: Prediction error on the dataset of PAMI18. We refer to the supplemental material [29, Sec. II] or [14] for the train-test video split used for the *deep-position-only baseline* (identical to original method). Top left: Average on test videos. Rest: Results per video category (Exploration, Moving Focus, Ride, Static Focus). Legend is identical in all sub-figures.

The main interrogation is: Why does the performance (of existing predictors compared with baselines) degrade when both modalities are considered? To explore this question from the architectural point of view, we divide this in two intermediate questions *Q3* and *Q4*.

Q3 on ground-truth saliency: If O1 is solved perfectly by providing the ground-truth saliency, what are good choices for O2 and O3?

That is, in comparison with the baselines considering each modality individually, choices whose performance improves, or at least

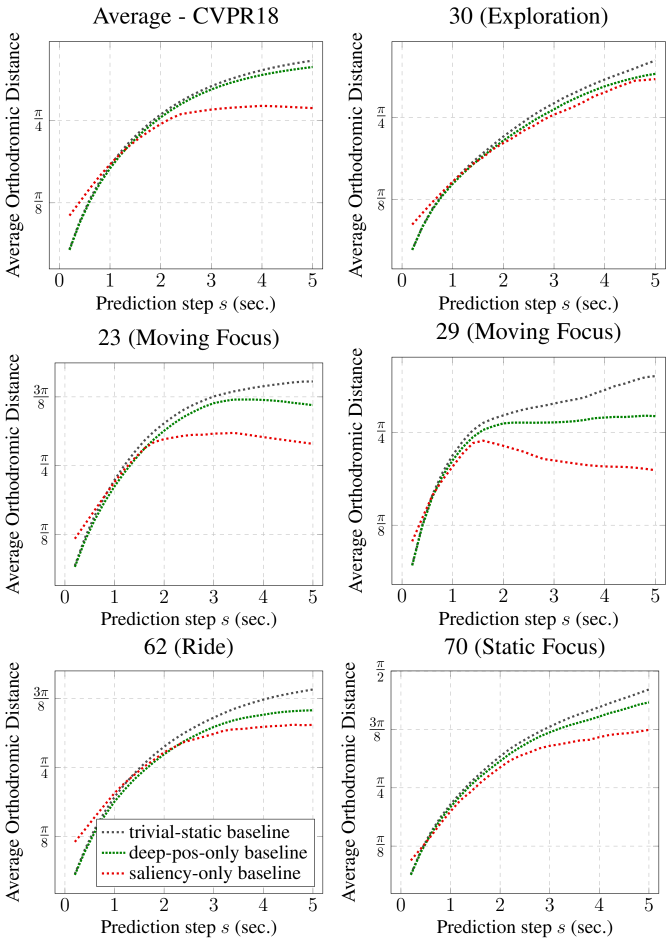

Fig. 12: Prediction error on the dataset of CVPR18. We refer to the supplemental material [29, Sec. II] or [14] for the train-test video split used for the *deep-position-only baseline* (identical to original method). Top left: Average on test videos. Rest: Results per video category (Exploration, Moving Focus, Ride, Static Focus). Legend is identical in all sub-figures.

does not degrade, when considering both information modalities.

6.1 Answer to Q3 - Analysis with ground-truth saliency

In our taxonomy in Sec. 2.2, we have distinguished the prediction methods that consider both input modalities, based on the way they handle the fusion: either both position and visual information are fed to a single RNN, in charge of at least O3 and O2 at the same time (case of MM18, ChinaCom18, NOSSDAV17), or the time series of positions are first processed with a dedicated RNN, the output of which then gets fused with visual features (case of CVPR18). To answer Q3, we consider their most recent representatives: the building blocks of MM18 and CVPR18 (see Fig. 3). We still consider that O1 is solved perfectly by considering the ground-truth saliency introduced in Sec. 5.4.

Prediction horizon: From the answer to Q2, we consider the problem of predicting head positions over a prediction horizon longer than the existing methods (see Table 1), namely 0 to $H = 5$ seconds. This way, both short-term where the motion is mostly driven by inertia at t, and long-term where the content saliency impacts the trajectory, are covered.

Dataset: Given the properties of MMSys18-dataset, where users move significantly more (see Sec. 5.1) and which comprises different video categories (introduced in Sec. 5.4), we select this dataset for the next experiments investigating the architectures. In particular, we draw a new dataset out of MMSys18-dataset, selecting 10 train and 4 test videos by making sure that the sets are balanced between videos where the content is helpful (Static focus, Moving focus and Rides) and those where it is not (Exploration). Specifically, the train set is made with 7 Exploration videos, 2 Static Focus and 1 Ride, while the test set has 2 Exploration, 1 Static focus and 1 Ride videos. This number of videos is equivalent to the dataset considered in MM18, ChinaCom18 and NOSSDAV17 (10). This dataset is therefore challenging but also well fitted to assess prediction methods aiming to get the best out of positional and content information.

Auto-regressive framework: Our re-implementation of CVPR18, named CVPR18-repro, has been introduced in Sec. 4.1. For MM18, we use the code provided by the authors in [36]. The evaluation metric is still the orthodromic distance as exposed in Sec. 5.4. We make three modifications to CVPR18 and MM18 (shown in Fig. 3), which we refer to as CVPR18-improved and MM18-improved, respectively. **First**, as for our *deep-position-only baseline*, we add a sequence-to-sequence auto-regressive framework to predict over longer prediction windows. We therefore embed each MM18 and CVPR18 building blocks into the sequence-to-sequence framework. It corresponds to replacing every LSTM cell in Fig. 4 with the building blocks represented in Fig. 3. **Second**, we train them with the mean squared error based on 3D Euclidean coordinates $(x, y, z) \in \mathbb{R}^3$. This helps the convergence with a seq2seq framework handling content, which is likely due to the removal of the discontinuity of having to use a modulo after each output in the training stage when Euler angles are considered. With 3D Euclidean coordinates, the projection back onto the unit sphere is made only at test time. We however retain the orthodromic distance as the benchmark metric. **Third**, instead of predicting the absolute position as done by MM18, we predict the displacement (motion). This corresponds to having a residual connection, which helps to reduce the error in the short-term, as also identified by [30]. Specifically for the MM18 block, we also change (1) the saliency map that we grow from 16×9 to 256×256, and (2) the output, i.e. the center of the FoV, which is defined by its (x, y, z) Euclidean coordinates.

Training: We train each model for 500 epochs, with a batch size of 128, with Adam optimization algorithm with a learning rate of 0.0005 and the mean squared error based on 3D Euclidean coordinates $(x, y, z) \in \mathbb{R}^3$ as loss function.

Results: Fig. 13 shows the improved models of MM18 and CVPR18 perform better than the original models. It also shows that MM18-improved is still not able to perform at least as well as the *deep-position-only baseline*. However, it is noticeable that CVPR18-improved is able to outperform the *deep-position-only baseline* for long-term prediction, approaching the *saliency-only baseline*. CVPR18-improved is also able to stick to the same performance as the *deep-position-only baseline* for short-term prediction. Fig. 14 provides the detailed results of CVPR18-improved over the different videos in the test set, associated with their respective category identified in [34]. While the average results show reasonable improvement towards the *saliency-only baseline*, we observe that CVPR18-improved significantly improves over the *deep-position-only baseline* for non-exploratory videos. Finally, we recall that the visual features provided to CVPR18-improved are the ground-truth saliency (i.e., the heat maps obtained from the users traces).

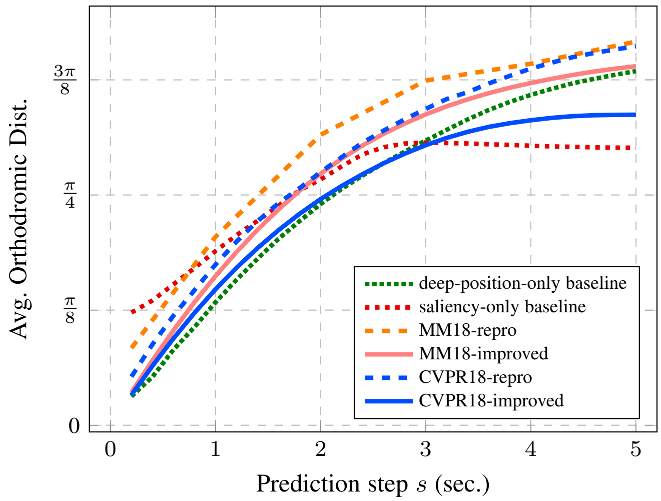

Fig. 13: Average prediction error of the original and improved models of MM18 and CVPR18, all fed with GT-sal, compared with baselines.

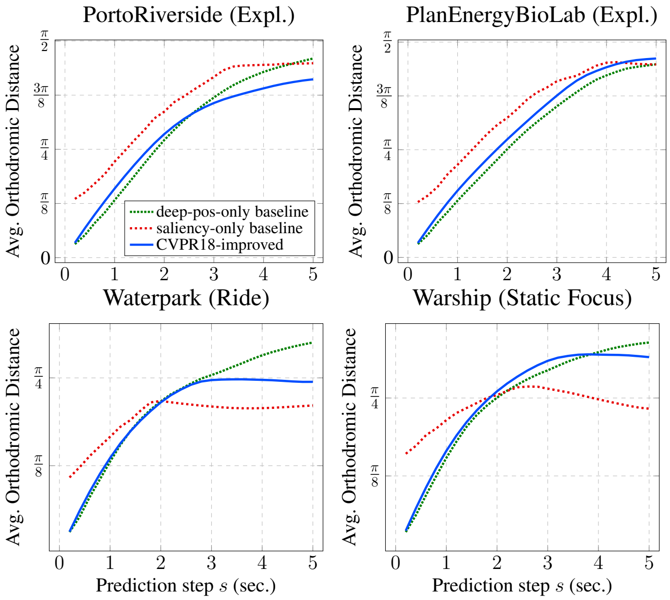

Fig. 14: Prediction error for CVPR18-improved. Detailed result for each type of test video. Legend is identical in all sub-figures.

Answer to Q3: If O1 is solved perfectly by providing the ground-truth saliency, then O2 and O3 are best achieved separately by having a dedicated recurrent unit to extract features from the positional information only, before merging them in subsequent layers with visual features, as CVPR18 does. If the same recurrent unit is both in charge of O2 and O3, as in MM18, it appears to prevent from reaching the performance of the *deep-position-only baseline*.

Therefore, we next analyze:

Q4 on content-based saliency: If O1 is solved approximately by providing a saliency estimate obtained from the video content only, do the good choices for Q3 still hold, or does the performance degrade away from the baselines again? If so, how to correct?

6.2 Answer to Q4 - Analysis with content-based saliency

We first summarize the findings of the root-cause analysis so far. In Q1, we found that even though averaging the prediction error over the trajectory might benefit the baselines, it does not and it is not a cause for the worse performance. In Q2, we have shown that the design assumption of the predictors are met if the dataset is made of non-exploratory videos with sufficient motion, and the prediction horizon is greater than 2 to 3 sec.. In Q3, on horizons and datasets verifying the latter conditions, we have found that when the visual information is represented by ground-truth saliency (O1 is perfectly solved), only the architecture of CVPR18 is able to exploit this modality without degrading compared with the baselines.

In this section, we do not consider O1 perfectly solved anymore. We consider the saliency information (i.e., heat map) is estimated from the video content only, not obtained from the users' statistics anymore. Our goal is not to find the best saliency extractor for O1, but instead to uncover the impact of less accurate saliency information onto the architecture's performance, to then overcome this impact if necessary.

In the remainder of the paper, when the heat map fed to a method is obtained from the video content (not from the users traces), the name of the method is prefixed with CB-sal (for Content-Based saliency). Also, CB *saliency-only baseline* denotes the *saliency-only baseline* defined in Sec. 5.2.3 when the heat map is obtained from the content, and not from the users traces. Conversely, when the heat map fed to a method is obtained from the users traces (and not estimated from the video content), the name of the method is prefixed with GT-sal (for Ground-Truth saliency, defined in Sec. 5.2.1). The GT *saliency-only baseline* denotes the *saliency-only baseline* defined in Sec. 5.2.3 when the heat map is obtained from the users traces.

Saliency extractor: We consider PanoSalNet [11], [36], also considered in MM18. The architecture of PanoSalNet is composed by nine convolution layers, the first three layers are initialized with the parameters of VGG16 [37], the following layers are first trained on SALICON [38], and finally the entire model is re-trained on 400 pairs of video frames and saliency maps in equirectangular projection. We exemplify the resulting saliency on a frame in [15, Fig. 6].

Results of CVPR18-improved: First, Fig. 15 shows the expected degradation using the content-based saliency (obtained from PanoSalNet) compared with the ground-truth saliency: the CB *saliency-only baseline* (dashed red line) is much less accurate than the GT *saliency-only baseline* (solid red line).

Second, we observe that, despite performing well with ground-truth saliency, CVPR18-improved fed with content-based saliency degrades again away from the *deep-position-only baseline*. Specifically, two questions arise:

- Why does CB-sal CVPR18-improved degrades from GT-sal CVPR18-improved for horizons $H \leq 2$ sec., where the best to achieve is the *deep-position-only baseline* according to Fig. 13?
  The training losses are the same. The only difference is in the input values representing the saliency. We can show that the saliency CB-sal is less sparse than GT-sal, hence there are more non-zero inputs, which are also less

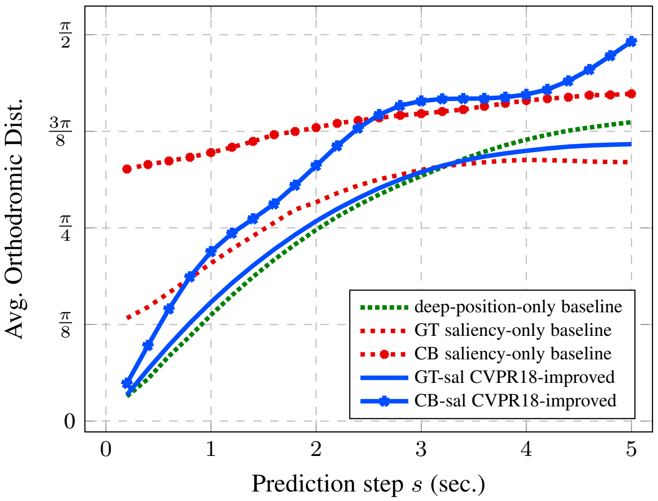

Fig. 15: Prediction error of CB-sal CVPR18-improved (with Content-Based saliency) against GT-sal CVPR18-improved (with Ground-Truth saliency) and baselines.

accurate (obviously, compared to the GT). Therefore, the contribution of the CB-sal inputs should be nullified by the weights of the fully-connected layer in charge of the fusion. It is simple to verify that when fully connected layers have to cancel out part of their inputs acting as noise for the classification task, the convergence of the training error degrades with the number of such inputs. Such wrong performance in training indicates a sub-optimal architecture for the problem at hand.

- Why does CB-sal CVPR18-improved degrade from original CVPR18 for $H \in [0s, 1sec.]$?
  The first difference is the training loss, defined over a longer horizon for CB-sal CVPR18-improved ($H \in [0$ sec.,5 sec.$]$), while it is only for $H = 1$ sec. in original CVPR18. The former loss is therefore likely more difficult to explore and minimize. The second difference is the presence, in original CPVR18, of convolutional and pooling layers processing various visual inputs including saliency before the fusion. Such layers can help decrease the input level into the fusion layer. However, they are not sufficient to enable a fully-connected layers to predict over $[0s, Hs]$ for $H \geq 3$ sec., as discussed in the next section.

Partial answer to Q4: If O1 is solved approximately by providing a saliency estimate obtained from the video content only, the good choice for Q3 (CVPR18-improved) is not sufficient anymore.

7 TRACK: A NEW ARCHITECTURE FOR CONTENT-BASED SALIENCY

We now first complete the root-cause analysis by examining more detailedly the architectural reasons for CVPR18-improved to degrade away again from the baselines with CB-sal. We then propose our new deep architecture, TRACK, stemming from this analysis. Its evaluation shows superior (once equal) performance on all the datasets of considered competitors and wider prediction horizons.

7.1 Analysis of the problem with CVPR18-improved and content-based saliency (CB-sal)

The fundamental characteristic of the problem at hand is: over the prediction horizon, the relative importance of both modalities (past positions and content) varies. Indeed, we expect the motion inertia to be more prominent first, and only then the content to possibly attract attention and change the course of the motion. It is therefore crucial to have a way of combining both modality features in a time-dependent manner to produce the final prediction. However, in the best-performing architecture so far, CVPR18-improved, we notice that the single RNN component enables this time-dependent modulation only for the positional features, while the importance of the content cannot be modulated over time. Replacing the ground-truth saliency with content-based saliency, the saliency map becomes much less correlated with the positions to predict. It is therefore important to be able to attenuate its effect in the first prediction steps, and give it more importance in the later prediction step.

7.2 Designing TRACK

From the latter analysis, **a key architectural element to add is a RNN processing the visual features (such as CB-sal), before combining it with the positional features.** Furthermore, this analysis connects with the seminal work of Jain et al., introducing Structural-RNN in [39]. It consists in casting a spatio-temporal graph describing a problem's structure into a rich RNN mixture following well-defined steps. Though the connection with head motion prediction is not direct, we can formulate our problem structure in the same terms. First, two contributing factor components are involved: the user's FoV and the video content. We can therefore express the spatio-temporal graph of a human watching a 360° video in a headset as shown in Fig. 16. Second, these two components are semantically different, and are therefore associated with: (i) an edgeRNN and a nodeRNN for the FoV, (ii) an edgeRNN for the video (only one input to the node), resulting in the architectural block shown in purple in Fig. 17. Embedded into a sequence-to-sequence framework, we name this architecture **TRACK**.

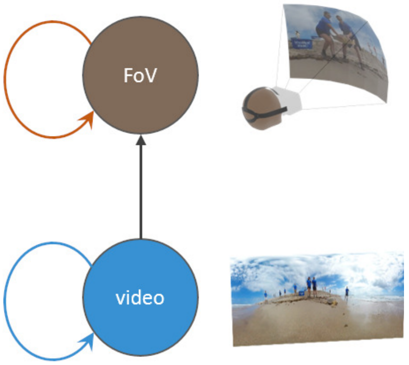

Fig. 16: The dynamic head motion prediction problem cast as a spatio-temporal graph. Two specific edgeRNN corresponds to the brown (inertia) and blue (content) loops, a nodeRNN for the FoV encodes the fusion of both to result into the FoV position.

Components of TRACK: The modules of TRACK are represented by (i) a doubly-stacked LSTM with 256 units each, processing the flattened CB-saliency map pre-generated for each time-stamp; (ii) another set of doubly-stacked LSTM with 256 units to process the head orientation input; (iii) a third set of doubly-stacked LSTM with 256 units to handle the multimodal fusion; and finally (iv) a FC layer with 256 and a FC layer with 3 neurons is used to predict the (x,y,z) coordinates, as described in Sec. 6.

7.3 Evaluation of TRACK

7.3.1 Comparison with GT-sal CVPR18-improved

On the MMSys18 dataset introduced in Sec. 6.1 (with higher user motion, and balanced video categories) and for prediction horizons up to 5 sec., Fig. 18 compares the results of TRACK with both CB-sal CVPR18-improved and GT-sal CVPR18-improved. Indeed, GT-sal CVPR18-improved is considered as a lower-bound on the error of CVPR18, which does not use PanoSalNet (and whose implementation is not available online nor was communicated on request). We observe that TRACK outperforms CB-sal CVPR18-improved (as expected), and equates to GT-sal CVPR18-improved, which is remarkable. This confirms the importance of the additional architectural elements of TRACK, able to exploit the (approximated) CB-saliency.

7.3.2 Comparison with all methods on their original metrics and $H \leq 2.5$ sec.

For the sake of space, the results of TRACK against all the considered existing methods, on their original metrics and prediction horizons, are presented in Sec. 3.3. It can be seen that on every dataset, TRACK (always with CB-saliency) establishes state-of-the-art performance: Fig. 3-Left shows that it outperforms MM18 (which also uses PanoSalNet), Table 2 shows that it significantly outperforms PAMI18, as does Table 4 for NOSSDAV17. ChinaCom18 is trained with the leave-one-out strategy, and the dataset is the same as NOSSDAV17. The results of TRACK listed against NOSSDAV17 in Table 4 are thus a lower-bound to TRACK's performance if it were trained with the leave-one-out strategy, already outperforming ChinaCom18 by more than 30%. As expected from the answer to Q2 in Sec. 5.5, for such short prediction horizons ($H \leq 2.5$ sec.), TRACK does not outperform the *deep-position-only baseline*. Its slightly inferior performance is due to the fact that we did not do any hyperparameter tuning for TRACK, while we did for the *deep-position-only baseline* which is smaller (tuning the number of layers and neurons). When training for $H = 5$ sec., the next results in Fig. 19 and [29, Sec. VIII] show that, for $s \leq 3$ sec., TRACK is similar to or even outperforms the *deep-position-only baseline* for 4 datasets in 5.

7.3.3 Exhaustive cross-comparison with all methods on all datasets with the orthodromic distance and $H = 5$ sec.

Average results: Fig. 19 presents the performance, on all 5 datasets (CVPR18, PAMI18, MMSys18, MM18, NOSSDAV17) of every (re-)implemented method, all with CB-sal: TRACK, CVPR18-improved, MM18-improved, *deep-position-only baseline*, *trivial-static baseline*. The results are averaged over the videos in the respective test sets made of 42 videos for CVPR18, 16 for PAMI18, 4 for MMSys18, 2 for MM18 and 2 for NOSSDAV17.

- For prediction steps $s \geq 3$ sec., TRACK outperforms all methods on all five datasets, except for the NOSSDAV17 dataset where it equates to the best (likely because the *saliency-only baseline* does not outperform the *deep-position-only baseline* on the NOSSDAV17 dataset, as shown in Fig. 9).
- For $s \leq 3$ sec., TRACK equates to the best method which is the *deep-position-only baseline*, except on the CVPR18 dataset where it has a slightly inferior performance but equates to the other methods.

Gains on video categories: The results in Sec. 5.4 have shown that the gains that can be expected from a multimodal architecture over the *deep-position-only baseline* are different depending on the video category: whether it is a *focus-type* or an *exploratory* video. The results in Fig. 19, averaged over all the videos of a test set, are therefore not entirely representative of the gains. To analyze the gains of TRACK over different video categories, we proceed as follows. First, we only focus on the CVPR18, PAMI18 and MMSys18 datasets to have a sufficient number of videos in the test set. Then, for MMsys18, we group the test videos into a Focus category (with Waterpark and Warship) and an Exploration category (with Portoriverside and Energybiolab), as done in Sec. 5.4. Finally for CVPR18 and PAMI18, in order to apply this binary categorization Focus vs Exploratory, we rely on the users behavior. Indeed, the more the users tend to have a focusing behavior, the lower the entropy of the GT saliency map[2]. Thus we consider the entropy of the GT saliency map of each video to assign the video to one category or the other. We sort the videos of the test set in increasing entropy, and we represent in Fig. 20 the results averaged over the bottom 10% (focus-type videos) and top 10% (exploratory videos).

- On the low-entropy/focus-type videos and for $s \geq 3$ sec., TRACK significantly outperforms the second-best method: by 16% for PAMI18 to 20% for both CVPR18 and MMSys18 at $s = H = 5$ sec.. TRACK performs similarly or better for $s < 3$ sec..
- On the high-entropy/exploratory videos, the gains of TRACK are much less significant: TRACK often performs similarly or slightly worse than the *deep-position-only baseline*, yet never degrading significantly away from this baseline, as the other methods do. Such results are expected from the observations drawn in Sec. 5.4 (Fig. 8,11,12) showing that the *saliency-only baseline* does not outperform the *deep-position-only baseline* on exploratory videos.

Qualitative examples: In Fig. 21, we exemplify the results on two low-entropy videos, also showing a representative frame with a user's future trajectory and the prediction of TRACK. More examples are in [29, Sec. VIII]. On focus-type videos, TRACK outperforms significantly the second-best method: by up to 25% in the examples.

7.4 Ablation study of TRACK

To confirm the analysis that led us to introduce this new architecture TRACK for dynamic head motion prediction, we perform an ablation study of the additional elements we brought compared to CVPR18-improved: we either replace the RNN processing the CB-saliency with two FC layers (line named AblatSal in Fig. 22), or replace the fusion RNN with two FC layers (line named AblatFuse).

Fig. 18 and 22 confirm the analysis in Sec. 7.1: the removal of the first extra RNN (not present in CVPR18) processing the saliency input has more impact: AblatSal degrades away from the *deep-position-only baseline* in the first time-steps. The degradation is not as acute as in CVPR18-improved as the fusion RNN can still modulate over time the importance of CB-saliency.

[2] The entropy of the 2D map is computed per frame, then averaged over all the frames for $t \geq 6$ sec. to skip the exploratory phase.

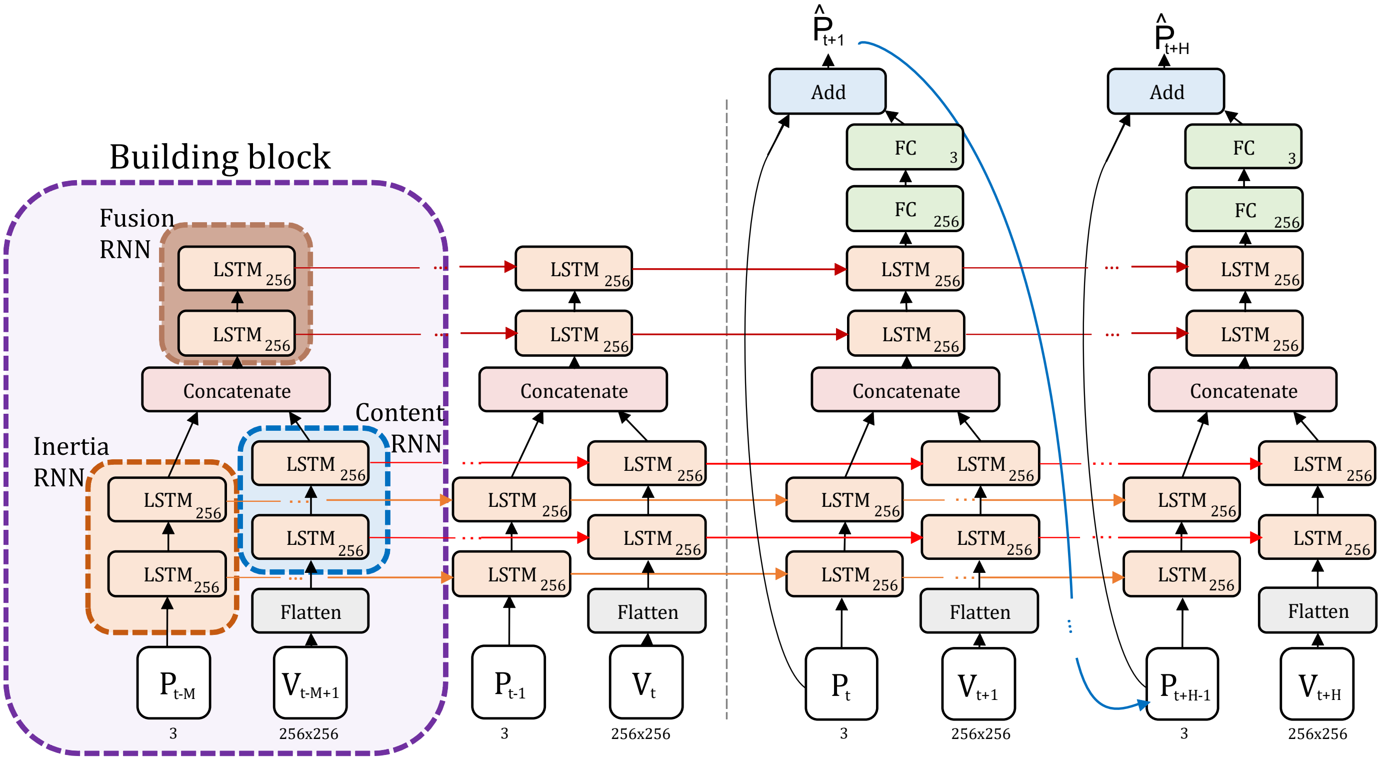

Fig. 17: The proposed TRACK architecture. The colors refer to the components in Fig. 16: the building block (in purple) is made of a an Inertia RNN processing the previous position (light brown), a Content RNN processing the content-based saliency (blue) and a Fusion RNN merging both modalities (dark brown).

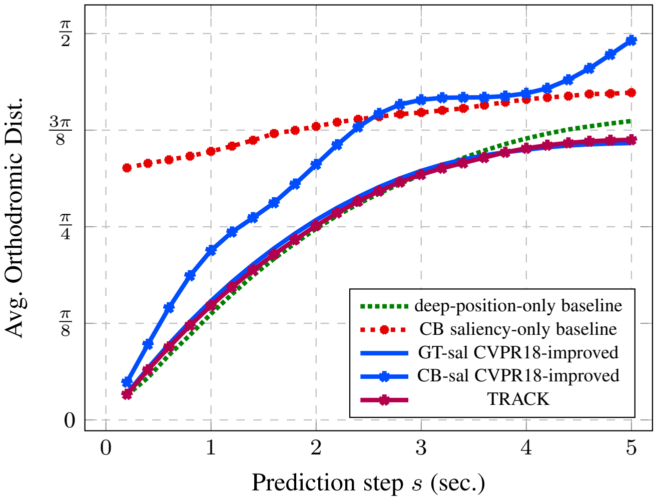

Fig. 18: Comparison, on the MMSys18 dataset, of TRACK with baselines and both CB-sal CVPR18-improved and GT-sal CVPR18-improved.

However, it seems this fusion RNN cancels most of its input (position and saliency features), as the performance of AblatSal is consistently similar to that of the *trivial-static baseline* (not plotted for clarity). The AblatFuse line shows that the impact of removing the fusion RNN is less important.

Answer to Q4: If O1 is solved approximately by providing a saliency estimate obtained from the video content only, the good choice (CVPR18-improved) for Q3 is not sufficient anymore. A RNN dedicated to processing the saliency must be added to prevent the prediction in the first time-steps from degrading away from the *deep-position-only baseline*. Our new deep architecture, named TRACK, achieves state-of-the-art performance on all considered datasets and prediction horizons.

8 Discussion

It is interesting to note that only a few architectures have been designed in the same way as TRACK, and none for head motion prediction. Indeed, following up on [39], Sadeghian et al. in [40] proposed a similar architecture to predict a pedestrian's trajectory based on the image of the environment, the past ego trajectory and the trajectories of others. Let us also mention that the CVPR18 block is similar to an early architecture proposed for visual question answering in 2015 [41], and PAMI18 is similar to Komanda proposed in 2016 for autonomous driving [42].

This article brings a critical analysis to existing deep architectures aimed to predict the user's head motion in 360° videos from past positions and video content. As we exhibit the weaknesses of the evaluation scenarios considered by previous works (dataset and competitor baselines), it is important to mention that other such critical analyses have been made for other application domains of deep learning very recently. Indeed, besides Martinez et al. mentioned earlier who showed in [30] the weakness of existing architectures for 3D-skeleton pose prediction, Ferrari Dacrema et. al. performed an analysis of recommendation systems in [43]. Not only did they show the difficulty to reproduce the evaluated algorithms, but also that the state-of-the-art methods could not outperform simple baselines. Similarly, the meta-analysis of Yang

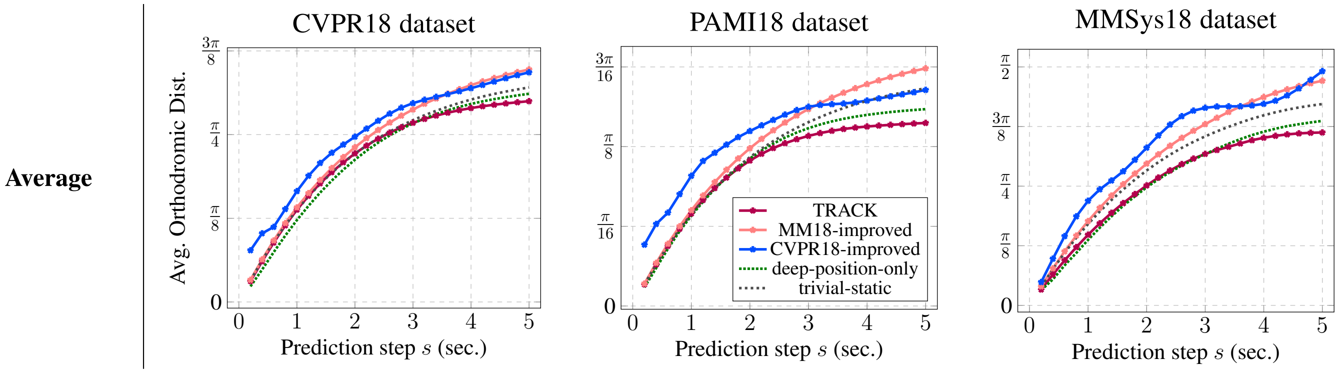

Fig. 19: Evaluation results of the methods TRACK, MM18-improved, CVPR18-improved and baselines, averaged over all test videos for the datasets of CVPR18, PAMI18 and MMSys18. Results for the NOSSDAV17 and MM18 datasets are shown in [29, Fig. 4]. Legend and axis labels are the same in all figures.

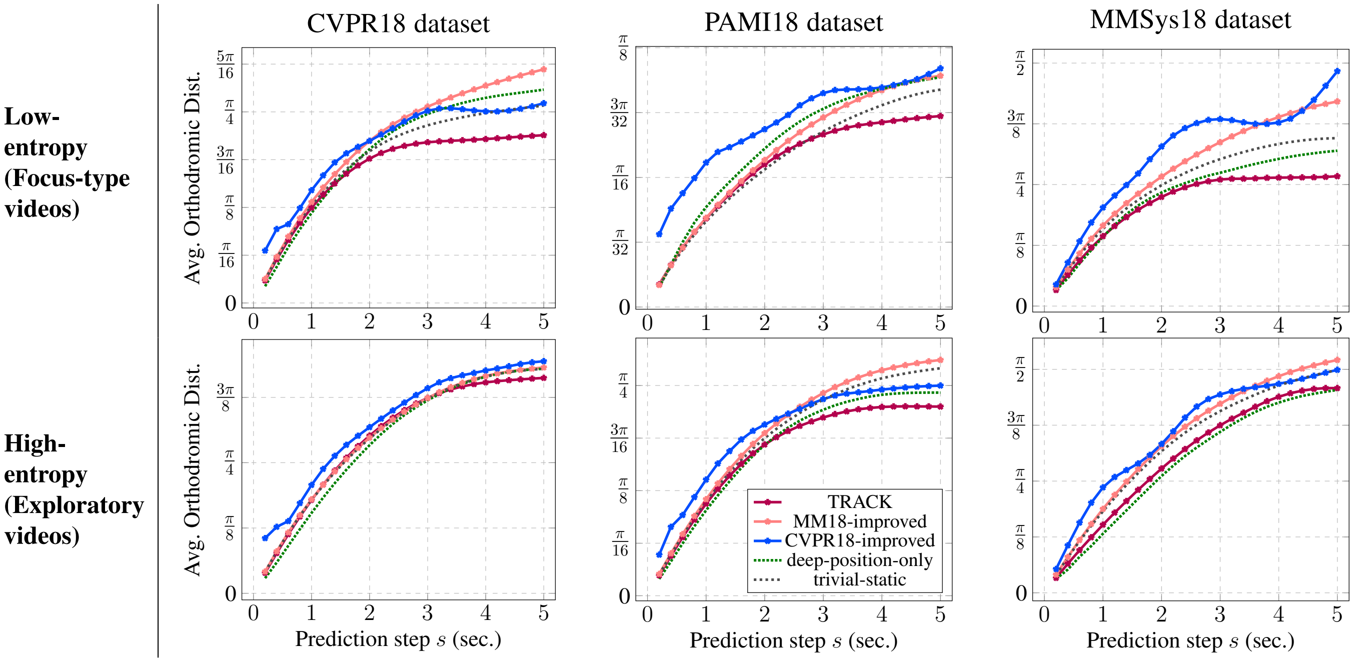

Fig. 20: Top row (resp. bottom row): results averaged over the 10% test videos having lowest entropy (resp. highest entropy) of the GT saliency map. For the MMSys dataset, the sorting has been made using the Exploration/Focus categories presented in Sec. 5.4. Legend and axis labels are the same in all figures.

et. al. [44] for information retrieval, and Musgrave et. al. [45] for loss functions, show that, contrary to the claims of the authors of multiple recent papers, there has been no actual improvement in several years of proposed neural networks to solve the problem in each of these fields.

In [46], Blalock et. al. show that the difficulty to reproduce, measure and compare the performances of different algorithms makes it difficult to determine how much progress has been made in a field, and this difficulty grows when each work uses different datasets, different performance metrics and different baselines. In the present article, we have faced the same difficulties. From an entire reproducible framework [14] we have made to enable replication and comparison, we could perform a critical and constructive analysis.

Our approach and findings are therefore aligned with other critical re-examinations of existing works in other application domains of deep learning.

9 Conclusion

This article has brought two main contributions. First, we carried out a critical and principled re-examination of the existing deep learning-based methods to predict head motion in 360° videos, with the knowledge of the past user's position and the video content. We have shown that all the considered existing methods are outperformed, on their datasets and with their test metrics, by baselines exploiting only the positional modality. To understand why, we have analyzed the datasets to identify how and when should the prediction benefit from the knowledge of the content. We have analyzed the neural architectures and shown there is only one whose performance does not degrade compared with the baselines, provided that ground-truth saliency information is provided, and none of the existing architectures can be trained

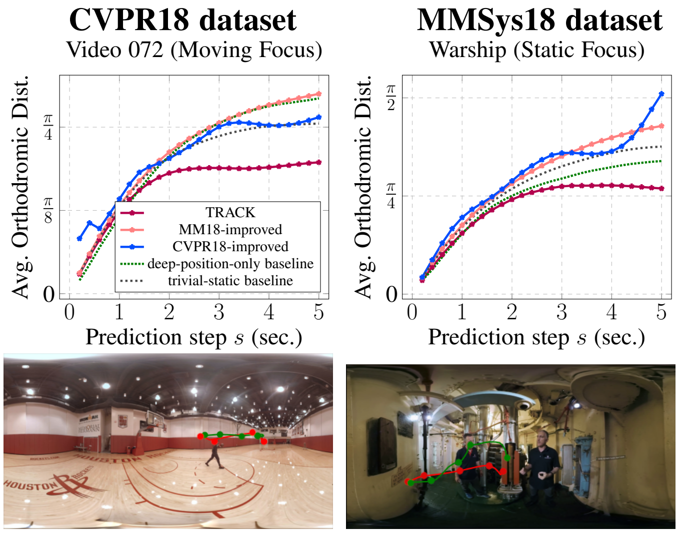

Fig. 21: Example of performance on two individual test videos of type Focus. On the frame, the green line represents the ground truth trajectory, and the corresponding prediction by TRACK is shown in red. More examples on other datasets are provided in the supplemental material [29, Fig. 5].

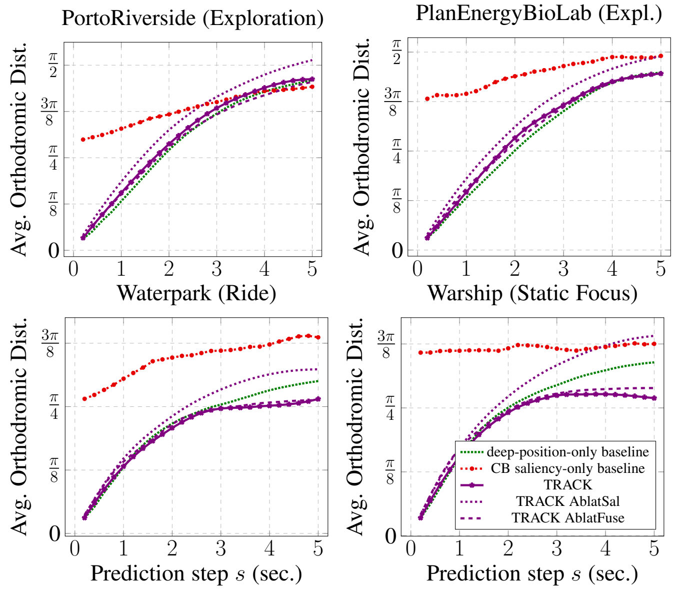

Fig. 22: Per-video results of TRACK and ablation study. The legend is identical for all sub-figures.

to compete with the baselines over the 0-5 sec. horizon when the saliency features are extracted from the content. Second, decomposing the structure of the problem and supporting our analysis with the concept of Structural-RNN, we have designed a new deep neural architecture, named TRACK. TRACK establishes state-of-the-art performance on all the prediction horizons $H \in [0$ sec.,5 sec.$]$ and all the datasets of the existing competitors. In the 2-5 sec. horizon, TRACK outperforms the second-best method by up to 20% on focus-type videos, i.e., videos with low-entropy saliency maps.

The experimental setup and datasets (whose formats we homogenized) of each assessed method and all our codes, are illustrated and provided online at [14]. This reproducible framework has already obtained an ACM reproducibility badge [15], and allows the community to easily test any predictor.

In future works, we will investigate deep attention mechanisms to refine the time- and space-varying fusion of modalities, as well as consider variational approaches (with VRNN) to also obtain confidence on the prediction, which is crucial for decision-making.

Acknowledgments

This work has been partly supported by the French government, through the UCA JEDI and EUR DS4H Investments in the Future projects ANR-15-IDEX-0001 and ANR-17-EURE-0004. This work was partly supported by EU Horizon 2020 project AI4Media, under contract no. 951911 (https://ai4media.eu/).

References

[1] M. F. R. Rondón, L. Sassatelli, R. A. Pardo, and F. Precioso, “track,” in *IEEE Int. Conf. on Image Processing (ICIP)*.

[2] Global Industry Analysts, Inc., “Virtual Reality (VR) - Global Market Trajectory & Analytics,” Industry report, 2020.

[3] J. Park, P. A. Chou, and J.-N. Hwang, “Rate-utility optimized streaming of volumetric media for augmented reality,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 149–162, 2019.

[4] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, “Optimizing 360 video delivery over cellular networks,” in *Workshop on All Things Cellular: Operations, Applications and Challenges*. ACM, 2016, pp. 1–6.

[5] M. Xiao, C. Zhou, V. Swaminathan, Y. Liu, and S. Chen, “BAS-360: Exploring spatial and temporal adaptability in 360° videos over HTTP/2,” in *IEEE INFOCOM*, 2018.

[6] 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.

[7] H. Hristova, X. Corbillon, G. Simon, V. Swaminathan, and A. Devlic, “Heterogeneous spatial quality for omnidirectional video,” in *IEEE MMSP*, Aug 2018, pp. 1–6.

[8] C.-W. Fu, L. Wan, T.-T. Wong, and C.-S. Leung, “The rhombic dodecahedron map: An efficient scheme for encoding panoramic video,” *IEEE Transactions on Multimedia*, vol. 11, no. 4, pp. 634–644, 2009.

[9] M. Xu, Y. Song, J. Wang, M. Qiao, L. Huo, and Z. Wang, “Predicting head movement in panoramic video: A deep reinforcement learning approach,” *IEEE Trans. on PAMI*, 2018.

[10] Y. Xu, Y. Dong, J. Wu, Z. Sun, Z. Shi, J. Yu, and S. Gao, “Gaze prediction in dynamic 360° immersive videos,” in *IEEE CVPR*, 2018, pp. 5333–5342.

[11] A. Nguyen, Z. Yan, and K. Nahrstedt, “Your attention is unique: Detecting 360° video saliency in head-mounted display for head movement prediction,” in *ACM Int. Conf. on Multimedia*, 2018, pp. 1190–1198.

[12] Y. Li, Y. Xu, S. Xie, L. Ma, and J. Sun, “Two-layer FoV prediction model for viewport dependent streaming of 360° videos,” in *EAI Int. Conf. on Communications and Networking (ChinaCom)*, Chengdu, China, Oct. 2018.

[13] 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 *ACM NOSSDAV*, 2017, pp. 67–72.

[14] M. F. Romero-Rondon, L. Sassatelli, R. Aparicio-Pardo, and F. Precioso, “head-motion-prediction,” https://gitlab.com/miguelfromeror/head-motion-prediction/tree/master, 2020.

[15] M. F. Romero-Rondón, L. Sassatelli, R. Aparicio-Pardo, and F. Precioso, “A unified evaluation framework for head motion prediction methods in 360° videos,” in *ACM Int. Conf. on Multimedia Systems (MMSys)*, 2020, pp. 279–284.

[16] J. Gutiérrez, E. J. David, A. Coutrot, M. P. Da Silva, and P. Le Callet, “Introducing un salient360! benchmark: A platform for evaluating visual attention models for 360° contents,” in *IEEE Int. Conf. on Quality of Multimedia Exp. (QoMEX)*, 2018.

[17] Y. Su and K. Grauman, “Making 360 video watchable in 2d: Learning videography for click free viewing,” in *IEEE CVPR*, July 2017, pp. 1368–1376.

[18] Y. Yu, S. Lee, J. Na, J. Kang, and G. Kim, “A deep ranking model for spatio-temporal highlight detection from a 360 video,” in *AAAI*, 2018.

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

[20] S. Rossi, F. De Simone, P. Frossard, and L. Toni, "Spherical clustering of users navigating 360° content," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 4020–4024.

[21] A. D. Aladagli, E. Ekmekcioglu, D. Jarnikov, and A. Kondoz, "Predicting head trajectories in 360° virtual reality videos," in *IEEE Int. Conf. on 3D Immersion (IC3D)*, 2017.

[22] 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 *IEEE Int. Conf. on Multimedia and Expo (ICME)*, 2018.

[23] X. Corbillon, F. De Simone, and G. Simon, "360° video head movement dataset," in *ACM Int. Conf. on Multimedia Systems (MMSys)*, 2017, pp. 199–204.

[24] C. Wu, Z. Tan, Z. Wang, and S. Yang, "A dataset for exploring user behaviors in VR spherical video streaming," in *ACM Int. Conf. on Multimedia Systems (MMSys)*, 2017, pp. 193–198.

[25] 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*, 2019.

[26] E. J. David, J. Gutiérrez, A. Coutrot, M. P. Da Silva, and P. L. Callet, "A dataset of head and eye movements for 360° videos," in *ACM Int. Conf. on Multimedia Systems (MMSys)*, 2018, pp. 432–437.

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

[28] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Adv. in Neural Info Proc. Sys. (NEURIPS)*, 2014, pp. 3104–3112.

[29] M. F. Romero-Rondon, L. Sassatelli, R. Aparicio-Pardo, and F. Precioso, "Track: A new method from a re-examination of deep architectures for head motion prediction in 360° videos," *Supplemental material to the submission*, 2021.

[30] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *IEEE CVPR*, 2017, pp. 4674–4683.

[31] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *IEEE CVPR*, 2016.

[32] S. Rossi and L. Toni, "Understanding user navigation in immersive experience: An information-theoretic analysis," in *ACM Int. Workshop on Immersive Mixed and Virtual Environment Systems*, ser. MMVE '20, 2020, p. 19–24.

[33] V. Sitzmann, A. Serrano, A. Pavel, M. Agrawala, D. Gutierrez, B. Masia, and G. Wetzstein, "Saliency in VR: How do people explore virtual environments?" *IEEE Trans. on Visualization and Computer Graphics*, vol. 24, no. 4, pp. 1633–1642, 2018.

[34] M. Almquist, V. Almquist, V. Krishnamoorthi, N. Carlsson, and D. Eager, "The prefetch aggressiveness tradeof in 360 video streaming," in *ACM Int. Conf. on Multimedia Systems (MMSys)*, 2018.

[35] J. P. Magliano and J. M. Zacks, "The impact of continuity editing in narrative film on event segmentation," *Cognitive Science*, vol. 35, no. 8, pp. 1489–1517, 2011.

[36] A. Nguyen, Z. Yan, and K. Nahrstedt, "PanoSalNet - source code for Your Attention is Unique: Detecting 360° video saliency in head-mounted display for head movement prediction," https://github.com/phananh1010/PanoSalNet, 2019.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[38] X. Huang, C. Shen, X. Boix, and Q. Zhao, "Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks," in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2015, pp. 262–270.

[39] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: Deep learning on spatio-temporal graphs," in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2016, pp. 5308–5317.

[40] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," in *IEEE ICCV*, 2017, pp. 300–311.

[41] A. Agrawal, J. Lu, S. Antol, M. Mitchell, C. L. Zitnick, D. Batra, and D. Parikh, "VQA: Visual Question Answering," in *IEEE ICCV*, 2015.

[42] 1st Rank in Udacity Challenge, "Team komanda," https://github.com/udacity/self-driving-car/tree/master/steering-models/community-models/komanda, 2016.

[43] M. F. Dacrema, P. Cremonesi, and D. Jannach, "Are we really making much progress? A worrying analysis of recent neural recommendation approaches," in *ACM Conf. on Recommender Systems*, 2019, pp. 101–109.

[44] W. Yang, K. Lu, P. Yang, and J. Lin, "Critically examining the "neural hype" weak baselines and the additivity of effectiveness gains from neural ranking models," in *ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2019, pp. 1129–1132.

[45] K. Musgrave, S. Belongie, and S.-N. Lim, "A metric learning reality check," *arXiv preprint arXiv:2003.08505*, 2020.

[46] D. Blalock, J. J. G. Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?" *arXiv preprint arXiv:2003.03033*, 2020.

Miguel Fabián Romero Rondón is a PhD student from the Université Côte d'Azur in France since 2017. His PhD thesis titled 'Streaming Virtual Reality: Learning for Attentional Models and Network Optimization,' is in the field of Machine Learning and Networking to identify how to structure the VR content in the cloud and optimally decide what to transmit to maximize the streaming quality. His research interests include Deep Learning, Multimedia Streaming, Virtual Reality and related applications.

Lucile Sassatelli has been an associate professor with Université Côte d'Azur since 2009, and she is a junior fellow of Institut Universitaire de France (IUF). She obtained a PhD from Université of Cergy Pontoise, France, and her professorial habilitation in 2019. She focuses on the problems of multimedia transmission of virtual and augmented reality (VR and AR), specifically on machine learning-based streaming approaches.

Ramón Aparicio Pardo is an associate professor at Université Côte d'Azur (UCA), France, since 2015. He received a MSc and a PhD from Universidad Politécnica de Cartagena (UPCT), Spain, in 2006 and 2011, respectively. His Ph.D. thesis was distinguished with Telefonica Award for Best Thesis in Networking. His research interests are in optimal design and management of communication networks, and more recently on machine learning-driven network control.

Frédéric Precioso received his PhD from Université Nice Sophia Antipolis in 2004. From 2005 to 2011, he had been an Associate Professor with engineering school ENSEA, Cergy, France. Since 2011, he has been a Full Professor at Université Côte d'Azur (UCA). Since 2018, Prof. Precioso is the Scientific and Program Officer on AI for the National Research Agency. His research interests are on Machine Learning and Deep Learning, specifically for computer vision, audio and linguistics applications, with a strong focus on interpretability and fairness.