Is attention to bounding boxes all you need for pedestrian action prediction?

Lina Achaji, Julien Moreau, Thibault Fouqueray, Francois Aioun, François Charpillet

Is attention to bounding boxes all you need for pedestrian action prediction?

Lina Achaji[1,2], Julien Moreau[1], Thibault Fouqueray[1], Francois Aioun[1], and Francois Charpillet[2]
[1]*Stellantis Group, Technical center of Velizy 78140, France*
[2]*Inria, Nancy 54000, France*
{lina.achaji, julien.moreau, thibault.fouqueray, francois.aioun}@stellantis.com,
francois.charpillet@inria.fr

***Abstract*— The human driver is no longer the only one concerned with the complexity of the driving scenarios. Autonomous vehicles (AV) are similarly becoming involved in the process. Nowadays, the development of AV in urban places underpins essential safety concerns for vulnerable road users (VRUs) such as pedestrians. Therefore, to make the roads safer, it is critical to classify and predict their future behavior. In this paper, we present a framework based on multiple variations of the Transformer models to reason attentively about the dynamic evolution of the pedestrians' past trajectory and predict its future actions of crossing or not crossing the street. We proved that using only bounding-boxes as input to our model can outperform the previous state-of-the-art models and reach a prediction accuracy of 91% and an F1-score of 0.83 on the PIE dataset up to two seconds ahead in the future. In addition, we introduced a large-size simulated dataset (CP2A) using CARLA for action prediction. Our model has similarly reached high accuracy (91 %) and F1-score (0.91) on this dataset. Interestingly, we showed that pre-training our Transformer model on the simulated dataset and then fine-tuning it on the real dataset can be very effective for the action prediction task.**

I. INTRODUCTION

During every moment of our waking life, our brains are trying to predict what sights, sounds, and tactile sensations will be experienced next [1]. During motion continuation, viewers observe the trajectory of a moving object and simultaneously simulate its future trajectory, predicting where it is likely to move next. Knowledge about whether an object moves quickly or slowly affects these predicted trajectories accordingly [2]. The shift in our mobility system to the autonomous driving era is often regarded as adding an artificial layer of cognitive intelligence on top of basic vehicle platforms [3]. This layer of intelligence should be capable of not only perceiving the world, but also predicting and analyzing its future states. The predictive processing can be applied everywhere, especially when interacting with vulnerable road users (VRU) such as pedestrians. When dealing with pedestrians, we can consider prediction as a forecasting of the future state or states. For instance, these states can be the future positions of the pedestrian in the case of trajectory prediction. Also, we can formulate it as a higher-level semantic prediction such as the early anticipation of the future action of the pedestrian, for example, walking, running, performing hand gestures, or most importantly crossing or not crossing the street in front of the AV. Recently, trajectory and action prediction solutions have been proposed based on sequential reasoning that mainly use algorithms built on recurrent neural networks (i.e., RNN, LSTM) [4, 5, 6, 7]. However, it has recently become clear that LSTM lacks many capabilities to model sequential data. For instance, LSTM suffers from long-term prediction, often due to the vanishing gradient problem [8]. That leads to its inability to model the correlation between non-neighboring inputs in the sequence. Furthermore, during training, LSTM is not able to assign different weights to different tokens based on their relative importance to the output. This will force it to give equal attention to all inputs, even if the capacity of the model does not allow it. Hence, the attention coupled with LSTM [9] has enhanced the previously suggested solution by proposing a mathematical framework that can weigh each of the input tokens differently depending on their importance to the input sequence itself (self-attention) [10] or the output sequence (cross-attention). Nevertheless, attention mechanisms coupled with LSTM have limited the potential of "Attention" itself. We can see that using only attention "Transformer architecture" [11] can lead to better results. Transformers have first revolutionized the natural language processing problems by outperforming all the previously proposed solutions [12, 13]. However, it was only until recently that were proved equally efficient for non-NLP problems [14, 15, 16].

In this paper, we will apply Transformer Networks for the pedestrian action prediction task of crossing or not-crossing the street. This network will predict the crossing-probability of a pedestrian in front of the ego-vehicle using an observation sequence of 0.5 seconds with a prediction horizon varying between one and two seconds in the future. To our knowledge, we believe that we are the first to apply Transformers for this task. We will propose multiple variants of the Transformer architecture: the Encoder-alone architecture, the Encoder coupled with pooling-layers, and the Encoder-Decoder architecture. Our model will use only bounding boxes as input features. Moreover, it will predict the crossing probability with accuracy and F1-score that will outperform the previous benchmarks even when using more features as input [6]. These results are evaluated on the PIE dataset [4], and on a simulated dataset (CP2A) that we generated using the CARLA simulator. The latter dataset is larger in size than PIE, and will equally confirm the high efficiency of our models. Additionally, we will show that the pre-trained weights on the CP2A dataset will increase the efficiency of our fine-tuned models on the PIE dataset.

II. Related Work

In this section, we review the recent advances in pedestrian action and trajectory prediction. Then, we discuss the work done using Transformers in various field of applications and the effect of Transfer learning on Transformer models.

Pedestrian Action Anticipation is a highly important problem for autonomous cars. Where the objective is to anticipate in advance if a pedestrian will cross the road in front of the ego-vehicle. This problem was addressed using multiple approaches. For instance, [5] used a network of convolutional neural networks and LSTM to detect the pedestrians and predict their actions up to 1.3 seconds ahead in the future and their estimated time-to-cross. [17] used a stacked GRU network composed of five GRUs each of which processes a concatenation of different features (i.e., the pedestrian appearance, the surrounding context, the skeleton poses, the coordinates of the bounding boxes, and ego-vehicle speed). [18] converted the human pose skeleton sequences into 2D image-like Spatio-temporal representations and then applied CNN-based models. [6] has published a benchmark to evaluate action prediction on the PIE [4] and JAAD [19] datasets. Additionally, they proposed a new state-of-the-art model that uses attention mechanisms to combine implicit and explicit features.

Pedestrian Trajectory Prediction is a closely related task to the action prediction. In contrast, the output sequence is a set of predicted positions in the future. In recent works, we see that intention and action prediction can be critical for trajectory prediction [4]. The M2P3 model [7] used an Encoder-Decoder RNN along with conditional variational auto-encoder (CVAE) to predict multi-modal trajectories from an ego-centric vehicle view. GRIP++ [20] builds a dynamic GNN (Graph Neural Network) to model the interaction between agents in the scene. The Trajectron [21] combines elements from CVAE, LSTM, and dynamic Spatio-temporal graphical structures to produce multimodal trajectories. Recently, a Transformer model was proposed by [15] to predict the future trajectories of the pedestrians conditioning on the previous displacement of each pedestrian in the scene. The transformer used has the same architecture as the Vanilla Transformer proposed in [11].

Transformers Networks are self-attention-based models proposed by [11] for machine translation and have since become the state-of-the-art method in many NLP tasks. Here we focus on the development of Transformers for modeling actions, motion dynamics, and visual context. For instance, [22] introduced an Action Transformer model for recognizing and localizing human actions in video clips. [16] proposed a Spatio-temporal transformer for 3D human motion modeling by learning the evolution of skeleton joints embeddings through space and time. Also, [23] proposed the use of transformer models for the prediction of dynamical systems representative of physical phenomena. Recently, [24] and [25] applied a Spatio-temporal transformer for video action recognition. Similar to the NLP-oriented Transformers, where Large Transformer-based models are often pre-trained on large corpora and then fine-tuned for a particular task, [24, 25] used a pre-trained Image Transformer [26] weights, bootstrap the weights, and then fine-tuned the model on a very large dataset [27] for action recognition. This training on the large dataset has remarkably increased the performances of Transformers for visual action recognition.

III. Method

The prediction and analysis block in autonomous car architectures receives its inputs from the perception block. We assume that the sets of observations of every agent are arranged in sequences. Every sequence describes the states on a time-ordered scale from a specific time-step $t = m$, to another time-step $t = M$: $O_{m:M} = \{O_m, \ldots, O_M\}$. Conditioning on the whole history of observations, the prediction block will answer the question of "what will be the future state (or states) at the time $t = A$". More formally, the objective is to calculate the following probability distribution:

$$p(\, P_{a:A} \mid O_{m:M} \,)$$

Where $P_{a:A} = \{P_a, \ldots, P_A\}$, and P_t is the predicted state of a particular object or agent at time t. For the case of trajectory prediction, a is equal to the next time-step $M + 1$ of the end-frame in the observation sequence. For the action early anticipation, a is equal to A, where A is the time that the pedestrian starts to cross, or the last frame the pedestrian is observable in case no crossing takes place. The time between the last observation frame M in the observation sequence and the critical time A is called the Time-To-Event (TTE).

The input to our model is composed of the bounding box coordinates of each pedestrian in the scene. The bounding box is defined as the four coordinates (x_up, y_left, x_down, y_right) around the pedestrian, without the use of images or visual context. Certainly, these bounding boxes were first extracted from images, either manually (i.e., in the case of annotated datasets) or using a state-of-the-art object detector.

To predict the future states, we shall first build a representative model of the world or the environment around the vehicle. In the following section, we will present our Transformer Network based model.

A. Input Embedding

Before feeding the input $O_{m:M} = X_T \in \mathbb{R}^{T \times 4}$ to our encoder, we first project the 4-dimensional bounding boxes into a D-dimensional space via a linear layer.

$$E_T = W_e X_T + b_e$$

Where $E_T \in \mathbb{R}^{T \times D}$. $W_e \in \mathbb{R}^{4 \times D}$ and $b_e \in \mathbb{R}^D$ correspond to the learnable weight and bias that project the bounding boxes vector into the embedding space.

Unlike RNNs or CNNs, the Transformer has no notion of ordering within a sequence. Following [11], we inject sinusoidal positional encoding into the input embeddings.

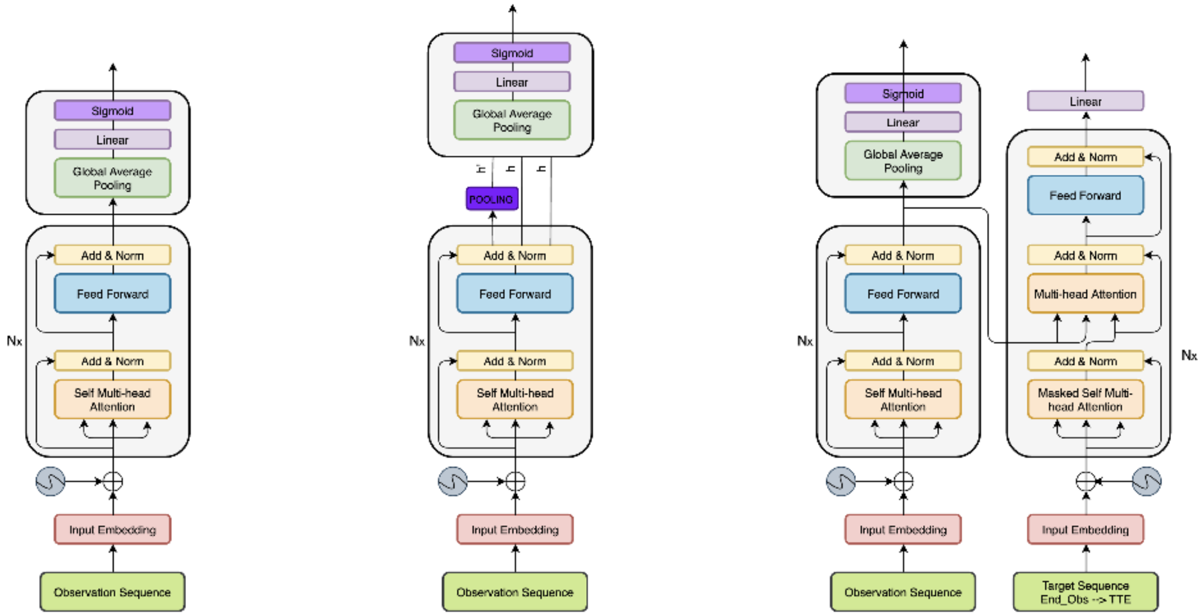

Fig. 1: (Left) Encoder Alone Architecture; (Middle) Pooled-Encoder Architecture; (Right) Encoder-Decoder Architecture.

B. Attention Layer

Given a set of items, the attention mechanism estimates the relevance of one item to other items. Basically, an attention layer updates each component of a sequence by aggregating global information from the complete input sequence. This is done by defining three learnable weight matrices to transform Queries ($W_Q \in \mathbb{R}^{D \times d_q}$), Keys ($W_K \in \mathbb{R}^{D \times d_k}$), and Values ($W_Q \in \mathbb{R}^{D \times d_v}$). In our model, $d_q = d_k = d_v = D$. The output of the attention layer is then given by:

$$Attention(Q, K, V, M) = softmax(QK^T/\sqrt{(d_k)} + M)V$$

Where the mask M prevents information leaking from future steps.

Following [11], we use a multi-head attention (MHA) mechanism to project the D-dimensional representation into subspaces calculated by different attention heads $i \in \{1, \ldots, H\}$. We concatenate all the output vectors to get the final MHA output:

$$MHA(Q^i, K^i, V^i) = Concat(head_1, \ldots, head_h)W^O$$

Where Q^i , K^i, and $V^i \in \mathbb{R}^{T \times F}$. We set $F = D/H$. The $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$, and $W^O \in \mathbb{R}^{D \times D}$.

C. Transformer Architecture

We proposed to apply multiple variations of the Transformer model:

1) Encoder-Only Architecture: The encoder-only model takes the X_T as input, proceeds into the embedding layer, and then goes into a stack of N encoder layers. Each encoder layer consists of a multi-head self-attention layer and point-wise feed-forward neural networks (P-FFN). Both layers are wrapped by a residual connection and layer normalization. We define $h_n \in \mathbb{R}^{T \times D}$ as the output of the $n-$th encoder layer, $n \in \{1, \ldots, N\}$:

$$h_n = LayerNorm(h_{n-1} + MHA(Q = h_{n-1}, KV = h_{n-1}))$$

$$h_n = LayerNorm(h_n + P - FFN(h_n))$$

The output sequence length of the encoder layers remains the same as the input $\in \mathbb{R}^{T \times D}$. However, to predict the action of crossing or not crossing the street, we compress the output sequence in the temporal domain by applying a global average pooling layer at the top of the encoder stacks. Then we apply another embedding layer followed by a sigmoid function to get the action crossing probability.

2) Encoder-Pooling Architecture: Instead of applying the global average pooling at the top of the encoder stack of layers directly, we proposed pooling intermediate layers between the encoder blocks (Fig. 1 middle). We see this solution as a compact way to reduce the observation sequence length gradually from the input to the output as the layer gets deeper. This solution will prevent the network from directly transforming the last layer embedding size from $\mathbb{R}^{T \times D}$ to $\mathbb{R}^{1 \times D}$ to predict the one-dimensional action. Instead, the final sequence length will be T', a reduced transformation of the initial sequence size T. In fact, pooling layers have been used in CNN architectures to reduce the dimensions of the feature maps. Similarly, [28] proposed to adapt the pooling layers to achieve representation compression and computation reduction in transformer networks. At the output of each encoder layer, we will apply a strided mean pooling layer on the query vector to reduce its sequence length:

$$h'_{n-1} = pooling(h_{n-1}),$$

where $h' \in \mathbb{R}^{T' \times D}$ for $T' < T$, and the unpooled sequence h serves the role of the key and value vectors:

$$h_n = LayerNorm(h_{n-1} + MHA(Q = h'_{n-1}, KV = h_{n-1}))$$

3) Encoder-Decoder Architecture: The final architecture is the encoder-decoder. The encoder block is identical to the encoder-only architecture with the classification head on the top of the encoder stacks. For the decoder architecture, we input the target sequence $T_{a:A} = Y_C \in \mathbb{R}^{C \times 4}$, where a is equal to the next-observation time frame $M + 1$, and A is equal to the critical time TTE. Each decoder layer consists of a masked multi-head self-attention layer, a masked multi-head cross-attention layer, and point-wise feed-forward neural networks (P-FFN). All layers are wrapped by a residual connection and layer normalization. The cross-attention layer takes the query from the previous self-attention layer plus the keys and the values from the encoder memory vectors:

$$h_{dec_n} = LayerNorm(h_{dec_{n-1}} + MHA(Q = h_{dec_{n-1}}, KV = h_{enc_N}))$$

The output of the decoder block is the target input shifted by one representing the future trajectory of the pedestrian from time step $a + 1$ to $A + 1$. This architecture will jointly learn to classify the pedestrian's future action as well as its future trajectory. In the following sections, we will see that jointly learning to predict the action and the trajectory is beneficial to both parties, where knowledge of the trajectory prediction increases the performance of the action prediction and vice versa.

D. Training and Inference

We train the encoder-only and the encoder-pooling models by optimizing the classification loss between the predicted and target class. For the encoder-decoder model, we use a combined weighted loss of the classification binary cross entropy (BCE) and the l_2 distance between the predicted and target sequence:

$$L = \lambda_{cls}\, BCE + \lambda_{reg}\, l_2(Y_{C+1}, \hat{Y_{C+1}})$$

Where λ_{cls} and λ_{reg} are the classification and regression hyperparameters to be tuned.

IV. Experiments

In this section, We evaluate our proposed models on two datasets following the evaluation protocols in [6]. We compare our model results with the baselines based on different features choices. We fix The observation length for all models at 16 frames (i.e., 0.5 seconds) and the Time-To-Event (TTE) between 30 and 60 frames (i.e., 1 and 2 seconds). We examine different settings of model structure (ablation studies) and explore the effect of changing the temporal prediction horizon on the results. Also, we investigate the effect of transfer learning on our models.

TABLE I: Comparison of the CP2A dataset with other pedestrian action prediction datasets.

Dataset	Running Time	# Peds	S/R
PIE [4]	6 hours	740k	Real
JAAD [29]	1.5 hours	391k	Real
STIP [30]	15.4 hours	3.5M	Real
CP2A (ours)	5.5 days	232.7M	Simulated

A. Datasets

1) Pedestrian Intention Estimation (PIE) dataset: The PIE dataset [4] provides 6 hours of continuous footage recorded at 30 frames per second (FPS) in clear weather conditions. The dataset represents a wide variety of pedestrian behaviors at the crossing point and includes locations with high pedestrian traffic and narrow streets. For each pedestrian who can potentially interact with the ego vehicle driver, it provides the coordinates of the bounding boxes, the critical time where each pedestrian will cross, as well as their actions of crossing or not crossing the street.

2) CARLA Pedestrian Action Anticipation (CP2A) dataset: In this paper, we present a new simulated dataset for pedestrian action anticipation collected using the CARLA simulator [31]. Indeed, Transformer networks have shown an advantage over other types of algorithms when a large amount of training data is available [26]. Moreover, pre-training Transformer networks on a large corpus of data and then fine-tune it on pre-text tasks has proven to be very beneficial [24]. To this end, we have simulated a very large dataset (see Table I) that can be automatically and freely labeled using the simulator itself. Using such a dataset, we can first evaluate our model and then improve the overall performance using transfer learning on the real collected scenarios such as the PIE dataset. To generate this dataset, we place a camera sensor on our ego-vehicle and set the hyper-parameters to those of the camera used to record the PIE dataset (i.e., 1920x1080, 110° FOV). Then, we compute bounding boxes for each pedestrian interacting with the ego vehicle as seen through the camera's field of view. We stored the bounding boxes information, along with the current semantic label of the pedestrians, and their critical time to cross the street. We generated the data in two urban environments available in the CARLA simulator: Town02 and Town03. Samples from the CP2A dataset are shown in (Fig. 2). We used an Nvidia 1080 Ti GPU for data simulation with a generation rate of about 600 pedestrian sequences in one hour, which corresponds to the size of the PIE dataset in 2 hours.

B. Baseline Models

Recently, [6] have published a benchmark on the PIE dataset for evaluating pedestrian action prediction. They unified multiple baseline models under the same evaluation procedures and analyzed their performance with respect to various properties of the data. Also, they introduced a new

Fig. 2: Samples from the simulated CP2A dataset in different scenarios.

state-of-the-art model called PCPA:

1) Static: It is a baseline model that uses the VGG16 [32] backend to predict the action using only the last frame in the observation sequence.

2) Multi-stream RNN (MultiRNN) [33]: it is composed of separate GRU streams independently processing the following feature types: Bounding boxes coordinates, the pose skeletons, the ego-vehicle speed, and the local box frame cropped around the bounding box that is processed by a VGG16 backend.

3) Inflated 3D (I3D) network [27]: it takes as input a stack of RGB frames or Optical Flow and generates final prediction using a fully connected layer.

4) PCPA [6]: it is composed of multiple RNN branches to encode non-visual features (e.g., bounding boxes coordinates, the pose skeletons, and the ego-vehicle speed) along with a C3D network to encode each pedestrian's local context. The outputs of the branches are then fed into modality attention layers.

C. Results on PIE dataset

Table II shows the comparison of the results of our proposed models with the state-of-the-art baselines on the PIE dataset. Our Transformer models TEO (Transformer Encoder-only), TEP (Transformer Encoder-Pooling), and TED (Transformer Encoder-Decoder) based only on Bounding Boxes features (BB) outperformer the state-of-the-art baselines that use BB, PoseSkeletons (P), Ego-vehicle speed (S), and RGB images features in term of accuracy. The TED model is the best in terms of ACC (92 %), F1-score (0.86), and AUC (0.9). This result exceeds the baselines' results by a large margin without using any of the other features. It emphasizes the impact of learning to predict the future position states and actions jointly and end-to-end. Additionally, we show outstanding results when using Transformers by comparing the PCPA model based on attention modules coupled with LSTM layers using only the BB features. In the latter case, the PCPA model results will drop dramatically and prevent the model from learning any significant pattern about the action prediction task.

TABLE II: Accuracy (ACC), F1-score, and Area Under Curve (AUC) comparison with baseline models. TEO: Transformer Encoder-only, TEP: Transformer Encoder-Pooling, TED: Transformer Encoder-Decoder, FTEO: Fine-tuned Transformer Encoder-only, BB: Bounding Boxes, P: Pose-Skeletons, S: Ego-vehicle Speed. ***: Ours**. ****:** Trained on the CP2A dataset.

Model Name	Architecture	Features	ACC	AUC	F1-score
Static	CNN	RGB	0.71	0.60	0.41
I3D	CNN	RGB	0.80	0.73	0.62
I3D	CNN	Optical flow	0.81	0.83	0.72
MultiRNN	GRU	BB, P, S, RGB	0.83	0.8	0.71
PCPA	LSTM + ATT	BB, P, S, RGB	0.87	0.86	0.77
PCPA	LSTM + ATT	BB	0.48	0.42	0.57
TEO*	Trans	BB	0.88	0.85	0.77
TEP*	Trans	BB	0.88	0.87	0.77
TED*	Trans	BB	**0.91**	**0.91**	0.83
FTEO*	Fine-tuned Trans	BB	0.89	0.89	**0.88**
CP2A**	Trans	BB	0.9	0.9	0.9

D. Ablations

Figure 3 shows the performance of our TEO model when trained with different numbers of layers and heads. We obtained the best performance for the 4-layer and 8-head configurations (shown in Table II) and 8-layer, 2-head configurations. For the TEP and TED, the best model goes for using the 8-layer and 8-head settings. We experiment with different hyperparameters for the regression and classification values in the TED loss function settings. We obtained the best model (Fig. 4) using the (1.8, 0.8) combination for the regression and classification parameters respectively. In our experiments, we use an embedding dimension D of 128, a

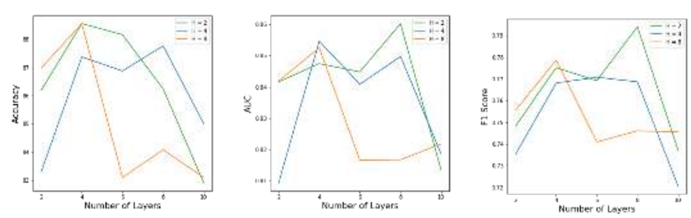

Fig. 3: Performance of the TEO model when trained with different number of attention heads and attention layers.

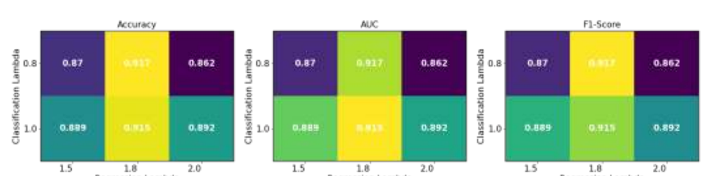

Fig. 4: Performance of the TED model when trained with different regression and classification hyper-parameters.

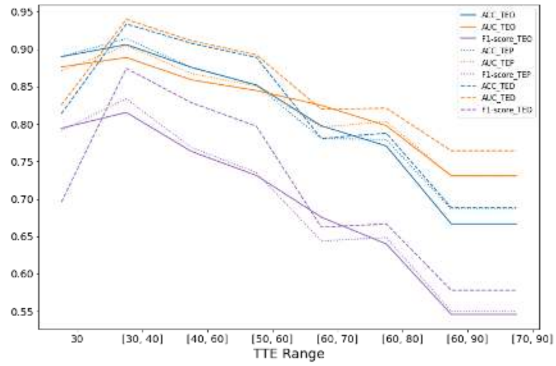

Fig. 5: Changing the prediction time horizon range for the models trained on the [30, 60] frames TTE interval.

P-FFN hidden dimension Dff of 256, and batch size of 32. We use the Adam [34] optimizer with a learning rate of 10^{-4}. Conforming with the findings in [35], we noticed that using stochastic optimizers for our Transformer models was not stable, and the results will change remarkably from one experiment to another using the same hyper-parameters.

We study how the prediction performance changes when we vary the temporal prediction horizon (equivalent to TTE time). We set the length of the observation sequence to 16 frames (i.e., 0.5 s) and varied the TTE time from 0.5 seconds to 3 seconds in the future (90 frames). These results were reported on the test set by taking the same models that were trained on the original 30-60 frame TTE interval. The graphs (Fig. 5) show that our models reach their upper limit when predicting the upcoming 1 to 1.3 s interval with an accuracy of 93% for the TED model. Furthermore, the model still performs reasonably well even when all prediction horizons are 2 to 2.3 seconds, with an accuracy of about 80% for all three models. We performed all our experiments on an Nvidia 1080 Ti GPU. We reported the inference time (Table III) of the three proposed models using their best hyperparameters scenarios. We should note here that we can use only the encoder block of the TED model in the inference phase. This strategy reduces the inference time since we predict the future action without the future pedestrian trajectories.

E. Results on CP2A

Previously presented results on using Transformer models with only bounding boxes as features and outperforming the models even using other features have been controversial because, in general, action prediction is related to understanding the visual context. The first assumption we can make here is that the PIE dataset is biased. A biased dataset is one where we can estimate the output independently of the input but rather based on some constant patterns that are uniquely related to that dataset. Although previous models were not able to detect these patterns, Transformers was able to find this bias. To test this hypothesis, we trained our models on the CP2A dataset, which is 43x larger than the PIE dataset and is highly unlikely to have the same bias that can be exhibited in the PIE dataset. The raw CP2A in Table II shows the performance of the 8-layer, 8-head TEO model on the simulated dataset. It achieved 90% accuracy and 0.9 for AUC and F1-score, demonstrating the effectiveness of using transformers with bounding boxes, regardless of the choice of dataset.

TABLE III: Inference Time on Nvidia 1080 Ti GPU.

Model Name	Number of Layers	Inference Time (ms)
TEO	4	1.63
TEP	8	2.85
TED	8	2.76

1) Effect of Transfer Learning: Interestingly, another result we obtained is that using the pre-trained weights from the CP2A model and fine-tuning them on the TEO model using the PIE dataset also has an advantage over the PIE results (FTEO model in Table II). In particular, we can see the improvement in terms of F1 score, where we reached 0.88, which outperforms the TED model trained from scratch on PIE, and slightly outperforms the baseline TEO model in terms of accuracy and AUC. In brief, the transfer of knowledge about action prediction from CP2A to PIE was effective. The results obtained here are consistent with the advantage observed in computer vision [24] when applying transfer learning from larger datasets.

V. CONCLUSION

In this paper, we presented transformer-based models for predicting pedestrians' action to cross or not cross the road in front of vehicles from an egocentric perspective. We have shown that using a simple and lightweight type of inputs (i.e., bounding boxes) with Transformers networks achieves high performance and outperforms the state-of-the-art in predicting the future up to 2 seconds in advance. The Encoder-decoder Transformer was the best model in terms of accuracy (91%) and F1-score (0.83) when we train from scratch on the PIE dataset. These findings make it clear that jointly predicting action and trajectory can be beneficial for both parties. We also introduced the CP2A simulated dataset that confirmed our results on the PIE dataset with accuracy and an F1-score of 0.9. Also, applying Transfer learning on the CP2A dataset from simulated scenarios to real cases improved our models by increasing the F1-score to 0.88 in the Encoder-alone architecture. In future work, we will apply Transformer models to visual context directly instead of bounding boxes. The goal of using visual features will not be to improve model performance in terms of accuracy, AUC, and F1-score, as it is difficult to beat the results obtained. However, it will help us focus on other types of criteria such as the explainability and interpretation of the model.

ACKNOWLEDGEMENT

This work was carried out in the framework of the OpenLab "Artificial Intelligence" in the context of a partnership between INRIA institute and Stellantis company.

REFERENCES

[1] Jeff Hawkins, Dileep George, and Jamie Niemasik. Sequence memory for prediction, inference and behaviour. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1521):1203–1209, 2009. 1

[2] Lawrence W Barsalou. Simulation, situated conceptualization, and prediction. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1521):1281–1289, 2009. 1

[3] Sagar Behere and Martin Törngren. A functional reference architecture for autonomous driving. *Information and Software Technology*, 73:136 – 150, 2016. 1

[4] Amir Rasouli, Iuliia Kotseruba, Toni Kunic, and John K Tsotsos. Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6262–6271, 2019. 1, 2, 4

[5] Dănuţ Ovidiu Pop, Alexandrina Rogozan, Clement Chatelain, Fawzi Nashashibi, and Abdelaziz Bensrhair. Multi-task deep learning for pedestrian detection, action recognition and time to cross prediction. *IEEE Access*, 7:149318–149327, 2019. 1, 2

[6] Iuliia Kotseruba, Amir Rasouli, and John K Tsotsos. Benchmark for evaluating pedestrian action prediction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1258–1268, 2021. 1, 2, 4, 5

[7] Atanas Poibrenski, Matthias Klusch, Igor Vozniak, and Christian Müller. M2p3: multimodal multi-pedestrian path prediction by self-driving cars with egocentric vision. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pages 190–197, 2020. 1, 2

[8] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014. 1

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 1

[10] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017. 1

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 1, 2, 3

[12] Klemens Lagler, Michael Schindelegger, Johannes Böhm, Hana Krásná, and Tobias Nilsson. Gpt2: Empirical slant delay model for radio space geodetic techniques. *Geophysical research letters*, 40(6):1069–1073, 2013. 1

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1

[14] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *arXiv preprint arXiv:2102.00719*, 2021. 1

[15] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer networks for trajectory forecasting. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10335–10342. IEEE, 2021. 1, 2

[16] Emre Aksan, Peng Cao, Manuel Kaufmann, and Otmar Hilliges. Attention, please: A spatio-temporal transformer for 3d human motion prediction. *arXiv preprint arXiv:2004.08692*, 2020. 1, 2

[17] Amir Rasouli, Iuliia Kotseruba, and John K Tsotsos. Pedestrian action anticipation using contextual feature fusion in stacked rnns. *arXiv preprint arXiv:2005.06582*, 2020. 2

[18] Joseph Gesnouin, Steve Pechberti, Guillaume Bresson, Bogdan Stanciulescu, and Fabien Moutarde. Predicting intentions of pedestrians from 2d skeletal pose sequences with a representation-focused multi-branch deep learning network. *Algorithms*, 13(12):331, 2020. 2

[19] Amir Rasouli, Iuliia Kotseruba, and John K Tsotsos. Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 206–213, 2017. 2

[20] Xin Li, Xiaowen Ying, and Mooi Choo Chuah. Grip: Graph-based interaction-aware trajectory prediction. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3960–3966. IEEE, 2019. 2

[21] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2375–2384, 2019. 2

[22] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2019. 2

[23] Nicholas Geneva and Nicholas Zabaras. Transformers for modeling physical systems. *arXiv preprint arXiv:2010.03957*, 2020. 2

[24] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021. 2, 4, 6

[25] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*, 2021. 2

[26] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,

Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 4

[27] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 2, 5

[28] Zihang Dai, Guokun Lai, Yiming Yang, and Quoc V Le. Funnel-transformer: Filtering out sequential redundancy for efficient language processing. *arXiv preprint arXiv:2006.03236*, 2020. 3

[29] Iuliia Kotseruba, Amir Rasouli, and John K Tsotsos. Joint attention in autonomous driving (jaad). *arXiv preprint arXiv:1609.04741*, 2016. 4

[30] Bingbin Liu, Ehsan Adeli, Zhangjie Cao, Kuan-Hui Lee, Abhijeet Shenoi, Adrien Gaidon, and Juan Carlos Niebles. Spatiotemporal relationship reasoning for pedestrian intent prediction. *IEEE Robotics and Automation Letters*, 5(2):3485–3492, 2020. 4

[31] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017. 4

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[33] Apratim Bhattacharyya, Mario Fritz, and Bernt Schiele. Long-term on-board prediction of people in traffic scenes under uncertainty. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4194–4202, 2018. 5

[34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[35] Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. Understanding the difficulty of training transformers. *arXiv preprint arXiv:2004.08249*, 2020. 6