2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Looking Inside Out: Anticipating Driver Intent From Videos

Yung-Chi Kung*, Arthur Zhang*, Junmin Wang, *IEEE Fellow*, and Joydeep Biswas

***Abstract*— Anticipating driver intention is an important task when vehicles of mixed and varying levels of human/machine autonomy share roadways. Driver intention can be leveraged to improve road safety, such as warning surrounding vehicles in the event the driver is attempting a dangerous maneuver. In this work, we propose a novel method of utilizing both in-cabin and external camera data to improve state-of-the-art performance in predicting future driver actions. Compared to existing methods, our approach explicitly extracts object and road-level features from external camera data, which we demonstrate are important features for predicting driver intention. Using our handcrafted features as inputs for both a transformer and a long-short-term-memory-based architecture, we empirically show that jointly utilizing in-cabin and external features improves performance compared to using in-cabin features alone. Furthermore, our models predict driver maneuvers more accurately and sooner than existing approaches, with an accuracy of 87.5% and an average prediction time of 4.35 seconds before the maneuver takes place. We release our model configurations and training scripts on `https://github.com/ykung83/Driver-Intent-Prediction`.**

I. Introduction

The number of vehicles being driven is continuously increasing, but less than half of all drivers follow even basic safety conduct like turning on a blinker before performing a lane change [1]. To improve road safety, many safety-centric Advanced Driver Assistance Systems (ADAS) and Automated Driving Systems (ADS) [2, 3] have been designed to anticipate the actions of the driver and provide warnings or assistive actions. These approaches measure success using the prediction accuracy and average prediction time before the maneuver takes place (time-until-maneuver, TUM).

To predict driver intentions, both in-cabin and external information should be jointly utilized. It is well documented that cephalo-ocular cues are an excellent indicator of driver intent [4, 5]. However, the use of external data has been shown [6] to decrease accuracy and TUM. Consequently, follow-up work [7] purposely choose not to utilize external sensing, relying on internal camera and vehicle dynamics data. While there exist methods focused on the fusion of external data streams with internal data [8], they cannot match the state-of-the-art (SOTA) performance.

Despite these findings, we hypothesize that external sensing provides invaluable information for understanding driver intent. Vehicle surroundings provide context that may explain observed cephalo-ocular cues and communicate which maneuvers are possible. We propose explicitly extracting object and road level features from road camera videos instead of learning these features in an end-to-end manner. We employ two model architectures proficient in handling time dependencies to combine feature vectors from both in-cabin and external cameras. Both models surpass other methods when evaluated on a real-world dataset for predicting driver intentions [9]. A high-level overview of our proposed method is shown in Figure 1. In the remainder of the paper, we review prior work, describe the experiment setup, explain our model architectures, and conclude with an analysis of our results.

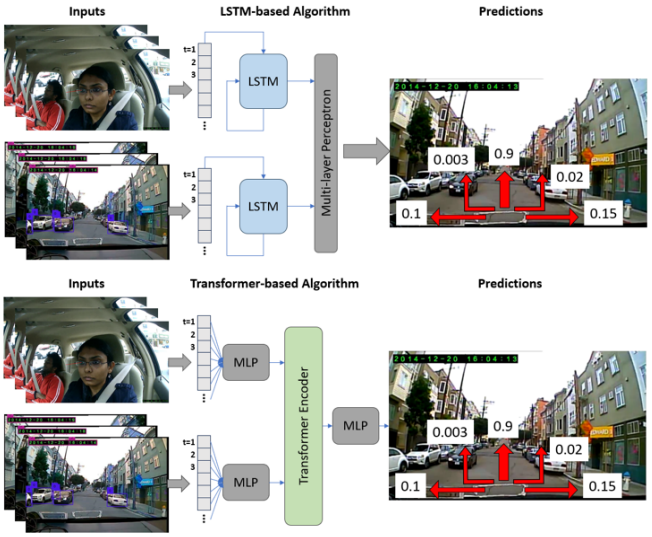

Fig. 1. **Predicting Driver Intent** Our proposed LSTM and transformer-based architectures use eye gaze, head pose, object detections, and road features from a real-world driving dataset to outperform state-of-the-art methods for driver intent prediction.

II. Related Work

A considerable amount of work has been done on driver maneuver prediction. Early efforts [10, 11] predict three different driving maneuvers: straight-line driving, left lane-change, and right lane-change. Later attempts [6, 12, 7] expand the action space to include left turn and right turn. We analyze both 3- and 5-maneuver prediction methods but focus on 5-maneuver prediction as it more closely resembles the true action space available to drivers.

Z. Hao *et al.* [11] use a gated-recurrent unit (GRU) with an attention mechanism for the 3-driving maneuver problem. Their model uses solely vehicle dynamics to achieve high

*Equal Contribution.

Y. Kung and J. Wang are with the Walker Department of Mechanical Engineering and A. Zhang and J. Biswas are with Department of Computer Science at the University of Texas at Austin, Texas, USA. {yung-chi.kung}@utexas.edu {jwang}@austin.utexas.edu {arthurz, joydeepb}@cs.utexas.edu

accuracy and precision when predicting one second before the driving maneuver takes place. However, both accuracy and precision decrease substantially if asked to predict at an earlier time step. N. Zhao *et al.* [10] employ a Convolutional Neural Network (CNN) and a Long-Short-Term-Memory (LSTM) based network to interpret driving dynamics and roadway information provided by a simulator for 3-maneuver prediction. Their method gives good accuracy but does not provide information on how early the network can predict driver intent before it occurs and only evaluates on simulated driving data. P. Gebert *et al.* [6] utilize an end-to-end CNN and LSTM-based network for the 5-maneuver problem. Their approach feeds the raw interior camera data through an optical flow estimation algorithm. They provide this output to a CNN for classification and feature extraction and use these features in an LSTM for prediction. Their method has an accuracy of 83.12% and an average prediction time of 4.07 seconds using only interior camera data. When using external video data, the accuracy drops to 75.5%. The paper [6] also released their real-world driving dataset, which we use as a benchmark. The other SOTA method is from N. Khairdoost *et al.* [7]. They generate an LSTM-based network with driver gaze, head pose, and vehicle dynamics data as inputs. The work expresses gaze information as a histogram. Each bin in the histogram correlates with a region of space in the driver's field of view. Despite video footage of the exterior being available, the authors choose not to use it in their network. Their method has an accuracy of 84.2% and an average prediction time of 3.6 seconds.

The uni-modal study by L. Li and P. Li [13] shows that there are only significant correlations between vehicle dynamics and driving maneuvers 0.55 seconds before the maneuver takes place. This means that when predicting driver intent for longer TUM scenarios, vehicle dynamics will not play a significant role. M. Hofbauer *et al.* [4] found that regions of interest and situational awareness can be predicted from driver gaze. This may allow us to better understand the driver's priorities and infer the intended maneuver well before it is executed. A. Kar [5] provides a detailed list of different gaze types and their intent. Some examples include fixation, where the eye moves at less than 100 degrees per millisecond and is indicative of cognitive processing and attention; saccade, where the eye moves between 100 and 700 degrees per second and is indicative of moving between targets of interest; and smooth pursuit, where the eye moves at less than 100 degrees per second based on the targets speed and is indicative of target tracking. This suggests that a high-precision list of gaze locations ordered in time would provide valuable insights into the driver's intentions.

Despite the numerous and varied studies on this topic, there exists a gap on how the surroundings of the vehicle can be leveraged in tandem with the behavior of the driver to provide earlier and more accurate maneuver predictions.

III. METHODOLOGY

We aim to increase the driver intent prediction accuracy and extend the average TUM in this work. Since vehicle dynamics seem only useful for short term maneuver prediction [13], it is not used. Instead, we extract handcrafted features from the in-cabin and external camera data. We follow the evaluation procedure used by current SOTA methods to ensure a fair comparison of results.

A. Data

We train and evaluate our methods on the publicly available Brains4Cars [9] dataset, which contains a collection of naturalistic driving maneuvers for driver action prediction containing RGB videos of both vehicle cabin and external road views. While the original dataset reports 700 vehicle maneuver videos, a portion of the training data is missing and 634 videos are publicly available. These videos are comprised of 234 driving straight, 124 left lane-change, 58 left turn, 123 right lane-change, and 55 right turn maneuvers. Each video is 5 seconds and 150 frames long. Per [6], these five-second snippets are from 6 seconds to 1 second before the maneuver. Following standard convention, the time of maneuver is based on the time the vehicle crosses the lane line. Using this dataset allows direct result comparisons between our methods and the prior work [6, 9, 14] because they are all based on the same data. In addition to using autonomous vehicle (AV) datasets, we considered predicting robot operator intent on large-scale urban robotics datasets like CODa [15]. However, we leave this for future work as these datasets lack a driver focused camera.

B. Interior Camera

Much like Leonhardt *et al.* [7], we extract gaze and head pose information from the interior camera. The difference is that in [7], they have a built-in, non-contact 3D gaze and head pose tracker running at 60 Hz while the dataset from Brains4Cars only provides an RGB video feed from the interior camera at 30 Hz. The work in [5] provides some references to extract eye gaze from a single stationary video.

We use MediaPipe to extract face landmarks from the driver in 2D and define a list of generic face landmarks with their coordinates in 3D. Using the solvePnP solver from OpenCV, we use these two lists of landmarks to estimate the projection of the rotation and translation of the driver's face onto a 2D plane using Eq. 1. S is an unknown scale factor, u and v are points from the 2D image, M is an estimate of the camera matrix, R is the rotation matrix, x, y, and z are from the tuned 3D model, and t is the translation vector.

$$S \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = M(R \begin{bmatrix} x \\ y \\ z \end{bmatrix} + t). \tag{1}$$

The rotation and translation vectors give us information on the direction the driver is facing. This is our approximation for the driver's head pose. Next, we obtain the 3D coordinates of the pupils by using the estimateAffline3D function from OpenCV to estimate the transformation between the estimated face coordinates and the model of a generic face. This gives us a 3D representation of the driver's pupils. Once we have the eye center and the pupil location, we can project

a line through those two points onto the same 2D plane used for the head pose to get the location of the gaze. This is illustrated in Figure 2.

The representation of the driver's head pose and gaze is a 4-dimensional vector containing the x and y coordinates of the intersection of the projected line from the driver's face and eyes with an imaginary 2D plane representing the windshield of the car. We choose this representation of the driver's gaze because it provides sufficient information for data-driven algorithms to distinguish fine-grained eye movements that provide different connotations regarding driver intent [5]. Prior histogram approaches [7] would not be precise enough to make this differentiation.

C. Exterior Camera

Fig. 2. Qualitative results of the gaze preprocessing algorithm on the Brains4Cars dataset. The red arrow projects their gaze to a point on the imaginary plane.

Our method uses the exterior camera to add lane and object-level information inputs to our model. This is the first method that directly incorporates object-level information into the model. Comprehending the context of objects allows methods to determine if maneuvers are unsafe due to nearby objects. We leverage Grounding Dino [16], a SOTA zero-shot 2D object detector to detect the following object classes: CAR, BICYCLE, PERSON, TRAFFIC SIGN, TRAFFIC LIGHT, and DATE. We omit the DATE class from the model and store the bounding box centers, height, width, and class ID as model inputs. Fig. 3 shows sample object detections on the Brains4Cars dataset. For computational reasons, we provide only the top 5 largest bounding boxes by area for each frame to the model.

For lane information, we use the ground truth lane labels in Brains4Cars, which contain the vehicle lane position, number of lanes, and whether the car is near a road intersection. This is useful because if there is no lane to the left of the driver, it should be a significant indicator that the driver is not attempting a left lane-change.

We use a 28-dimensional vector representation for the exterior camera, comprised of a 25-dimensional vector that describes the locations of surrounding objects and a 3-dimensional vector that represents the composition of lanes around the vehicle.

D. Evaluation

Like other SOTA methods, our method is evaluated on model accuracy, F1 score, and the average TUM that the model can correctly predict the driver's intentions. The accuracy and F1 scores are based on the performance of the models when trained and tested on the full 5 seconds of driving. We refer to this as Zero-time-to-maneuver. The average TUM is assessed by training the model on various time increments (1, 2, 3, 4, and 5 seconds) of driving data and computing the prediction accuracy at the corresponding time intervals before the maneuver actually occurs. We refer to this as Varying-time-to-maneuver. This is consistent with the approach taken by [6]. We train and evaluate with ten-fold cross-validation and report the average performance across all splits for each method.

Fig. 3. Object detection pre-processing results on the Brains4Cars dataset [9]. The object classes Car, Bicycle, Person, Traffic Sign, and Traffic Light are mapped to classes 0, 1, 2, 3, and 4 in the images shown. Only these object classes are used.

IV. EXPERIMENTS

We propose two machine-learning algorithms to predict the driver's intent. The first is based on fusing multiple LSTM units which we refer to as the F-LSTM. The LSTM is a standard method of predicting driver intent and is used by both SOTA methods that our algorithms are evaluated against [7] [6]. This makes it a good baseline to compare with the SOTA to evaluate if the hand-tuned features we use as inputs improve the prediction accuracy. The second algorithm fuses multiple streams of data using a transformer architecture which we refer to as F-TF. This method will be evaluated against the F-LSTM to see if it can learn long-term dependencies that would be difficult to capture with an LSTM-based algorithm.

A. F-LSTM

The LSTM-based architecture is a natural choice given that the problem is inherently dependent on time. LSTMs store and interpret data as a hidden vector, and propagate this hidden vector along each time step. This characteristic is desirable because the inputs have different representations

and this hidden vector may be used to project their qualities into a common representation.

Figure 4 shows the architecture of the F-LSTM. Separate LSTMs are used to accept the inputs for head pose and gaze, vehicle objects, and lane detections. This architecture allows each LSTM to specialize in interpreting separate modes of data. LSTM 1 accepts gaze and head pose information and has a hidden dimension of 10. LSTM 2 accepts lane information and has a hidden dimension of 5. LSTM 3 accepts surrounding vehicle information and has a hidden dimension of 10. The outputs of all three LSTMs are then flattened and fully connected to a multi-layer perceptron (MLP). The MLP consists of a 100 dimension fully connected layer with ReLU activation followed by a 5 dimension fully connected layer with sigmoid activation. The output from the last fully connected layer is used to predict driver actions. Cross-entropy loss is used to train the model.

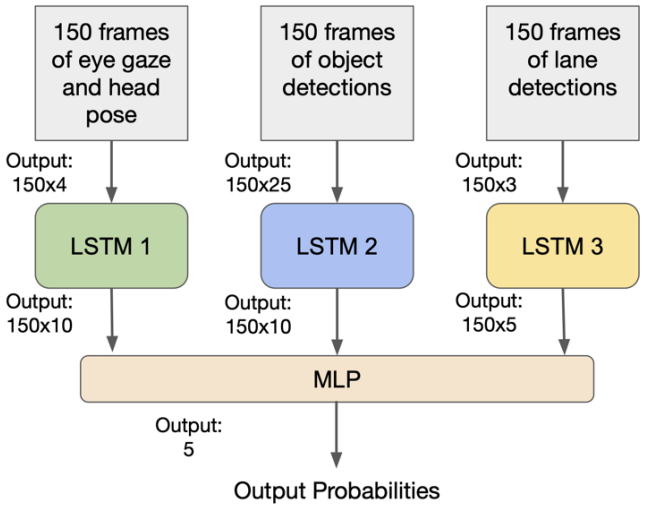

Fig. 4. LSTM-based architecture. The MLP is composed of a linear layer with ReLU activation followed by a linear layer with sigmoid activation. For the time-varying setup, we provide 30, 60, 90, 120, or 150 frames of the original data and pad the input to hold the number of input frames constant.

The same architecture is also used for the time-varying version of the problem. The only difference is that the training sequences are no longer the full 150 frames but a combination of the first 30, 60, 90, 120, and 150 frames of the original data. The extra spaces are padded with zeros. This keeps the time between predictions consistent with [6] to allow for a fair comparison.

B. F-TF

While LSTM-based architectures have proven successful for sequential tasks, prior works [17] demonstrate that they are adversely affected by long-range time dependencies due to increasing path length for signals. However, the self-attention module in transformer architectures reduces the path length, which can be leveraged to learn long-range correlations in sequential tasks.

Figure 5 describes our transformer architecture. We use three input representations: object detections, road/intersection data, and driver gaze with head pose. This gives feature vectors $\mathbb{R}^{B\times F\times P}$, where B is the batch size, F is the number of images in the sequence, and P is the number of dimensions in each feature vector. Fig. 5 states the size of P for each input representation: 4 features for gaze and head pose; 25 features for object detections; and 3 features for lane information. Each processed vector is linearly projected to a common representation and added with a 1D sinusoidal positional embedding before they are all concatenated to form a unified latent vector. The positional embedding is varied with the time dimension to retain temporal information. We use separate trainable linear projections for each processed vector because each vector is different in both scale and resolution. Similar to the F-LSTM architecture, we find that using a single linear projection results in worse performance than using three separate trainable linear projections. The MLP output vector sizes are 32, 16, and 16 for the in-cabin, object, and road, which provides the same representational power between the in-cabin and external information.

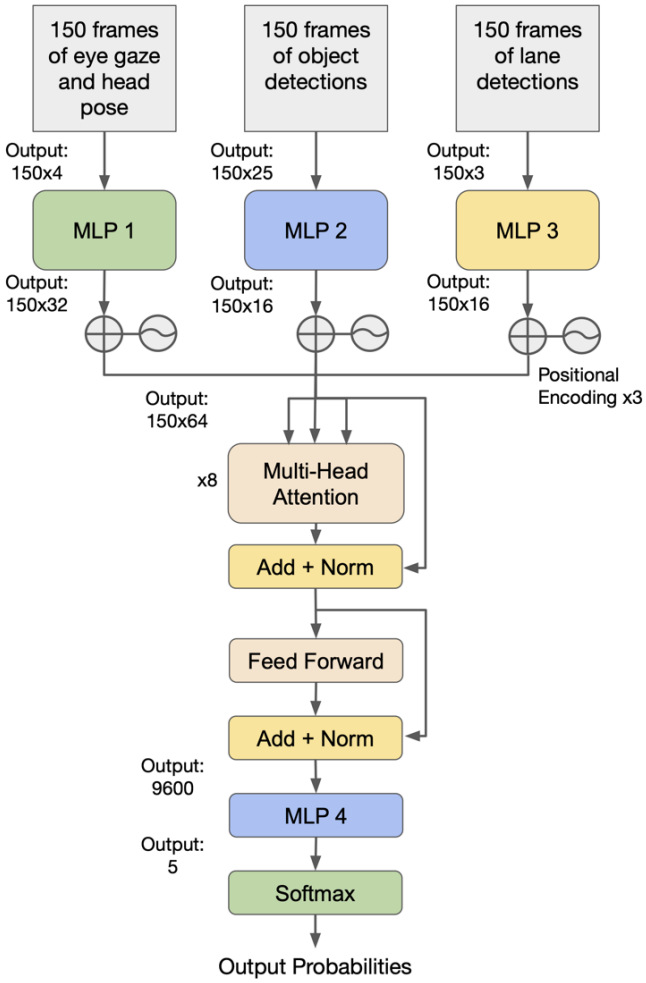

Fig. 5. Transformer-based architecture. All MLPs are composed of a Linear, ReLU, and Linear layer. For the time-varying setup, we provide 30, 60, 90, 120, or 150 frames of the original data and pad the input to hold the number of input frames constant.

We perform self-attention between all latent vectors for a single image sequence before feeding them to a standard feedforward module [18]. This allows our model to represent long- and short-term time dependencies with the same path length. Finally, we flatten the feedforward output to a 9600 dimensional vector and project this with a classification head, implemented as an MLP with one hidden layer. The MLP outputs a 5-dimensional feature vector to a softmax function that represents the driver intent probability vector. For the zero-time-to-maneuver experiments, we train using the full 150 frames of data. In the time-varying benchmark, we follow the same training setup as the F-LSTM for consistency.

C. Ablative Testing

Additional ablative testing is conducted on the F-LSTM and F-TF models to measure the performance contribution of the exterior camera's handcrafted features. The ablative tests for the F-LSTM and the F-TF are named F-LSTM-A and F-TF-A, respectively. These tests compare the performance of our F-LSTM and F-TF models with and without the external camera features. The original models are trained and tested using both modes of data while the ablative models are trained and tested using only features from the interior camera. The results are covered in the following section.

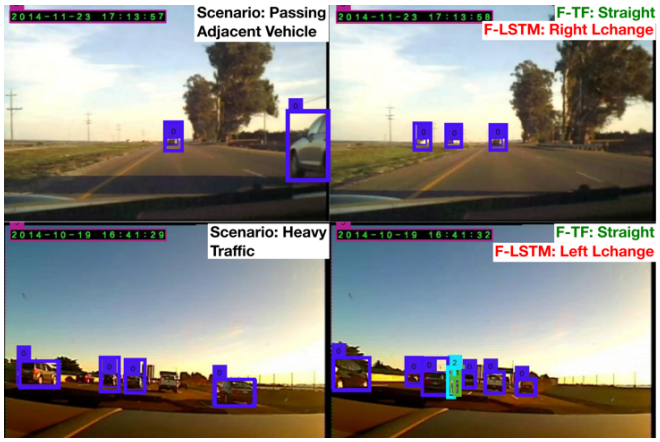

Fig. 6. Driver intent prediction scenarios that require long-term dependency understanding. Our transformer and LSTM based architecture are abbreviated as F-TF and F-LSTM. Green and red text indicate correct and incorrect predictions respectively.

V. Results

Table I compares our methods' results against other approaches in the literature. We observe that both the F-LSTM and F-TF algorithms outperform other SOTA methods by a significant margin.

A. Zero-time-to-maneuver

The F-LSTM can be directly compared with the methods from [6] and [7] since they all use an LSTM-based architecture. We conclude that our selected exterior features improve the model performance.

The performance of the F-TF is expected due to the ability of the self-attention module to attend to long-range dependencies across each video sequence. Fig. 6 qualitatively supports this property. In the upper scenario, the model must understand the adjacent vehicle's relative speed to infer that it may still be on the driver's right despite not being visible in the road camera. In the bottom situation, the model discerns that the driver refrained from changing lanes when there was an opportunity in the past. This suggests that the driver is less likely to make a lane change in the future, especially when traffic is heavier. In both of these situations, the F-TF accurately forecasts that the driver will continue straight, a prediction that the F-LSTM fails to make correctly.

However, the F-TF has a high standard deviation across the validation splits and is not significantly better than our F-LSTM architecture. We believe this can be attributed to the lack of training data available in the Brains4Cars dataset. It is well understood that computer vision transformer architectures [19] require internet-scale amounts of data to significantly surpass CNN architectures. We claim that because the transformer assumes no prior information about the sequential nature of the data, it requires far more data to learn this property and attain good performance. On the other hand, LSTM-based architectures are designed to leverage prior knowledge about the temporal relationship between frames, thus making it more data efficient. We postulate that our F-TF architecture would scale more effectively than some other methods if provided with far more training data.

TABLE I
ZERO TIME-TO-MANEUVER ACCURACY AND F1 SCORE RESULTS.

Method	Inside	Outside	Acc [%]	σ	F1 [%]	σ
Baseline Methods						
Chance	-	-	20	-	20	-
Prior	-	-	39	-	-	-
Methods from [9] and [14]						
IOHMM	X	X	-	-	72.7	-
AIO-HMM	X	X	-	-	74.2	-
S-RNN	X	X	-	-	74.4	-
F-RNN-UL	X	X	-	-	78.9	-
F-RNN-EL	X	X	-	-	80.6	-
Methods from [6]						
Outside	-	X	53.2	0.5	43.4	0.9
Inside	X	-	83.1	2.5	81.7	2.6
Two-stream	X	X	75.5	2.4	73.2	2.2
Method from [7]						
Interior+VD	X	-	84.2	-	82.9	-
Our Methods						
F-LSTM	X	X	**87.2**	2.3	**85.6**	3.4
F-TF	X	X	**87.5**	4.9	**86.3**	4.5

B. Varying-time-to-maneuver

Figure 7 compares our proposed model architectures' performance against the top performing prior approach [6] on the varying time-to-maneuver benchmark. Both the F-LSTM and the F-TF outperform the SOTA methods in this category. The F-LSTM and F-TF have an average prediction time of

4.34 and 4.35 seconds respectively compared to the 4.07 seconds from [6] and the 3.56 seconds from [7]. For a fair comparison with SOTA methods, we adopt their evaluation procedures. The accuracy of each method is compared using only the amount of data the method would have received at that time. At 5 seconds before the maneuver, each algorithm would have the first 30 frames of data of interior and exterior data. At 4 seconds before the maneuver, each algorithm would have access to the first 60 frames of data, and so on. Across all times before maneuver, our proposed methods outperform the state-of-art by at least 12%.

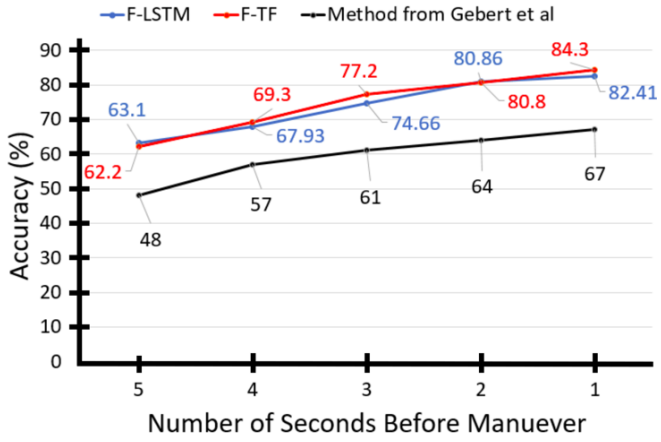

Fig. 7. Comparison of our methods with state of the art for varying-time-to-maneuver. Higher accuracy indicates that the algorithm is able to predict the driver action more accurately.

These findings corroborate our hypothesis that sequential driver intention prediction benefits from having access to a good external feature representation. Compared to our method, which has a prediction accuracy of about 63% 5 seconds before the maneuver takes place, we see that the prediction accuracy of the method proposed in [6] is as low as 48%. Our method outperforms the SOTA at every time interval. It is also worth noting that the decrease in performance when switching from a zero-time-to-maneuver problem to a varying-time-to-maneuver is much smaller for our proposed algorithms than the one proposed in [6]. This decrease can be quantified by comparing the accuracy of the varying-time-to-maneuver model at 1 second with the zero-time model. The accuracy of the F-LSTM decreases by 5.5% and the accuracy of the F-TF decreases by 3.7%. The method proposed by [6] degrades by 19.4% in accuracy from 83.1% to approximately 67.0% between the zero-time-to-maneuver and varying-time-to-maneuver problem. This would suggest that our methods, particularly the F-TF, are more capable of forecasting driver intent.

C. Ablative Tests

Ablative tests were also conducted to determine the contribution of the exterior information on the performance of our models. Table II demonstrates that the F-LSTM and F-TF significantly outperform their counterparts that were only trained with in-cabin features. This reinforces the idea that the hand-crafted feature space we designed to describe the exterior view is helpful for predicting driver intent.

TABLE II
ABLATIVE EXPERIMENTS ON THE EFFICACY OF THE HANDCRAFTED FEATURES OF THE EXTERIOR.

Driver Maneuver	F-LSTM-A	F-LSTM	F-TF-A	F-TF
Straight driving	67.3	87.0	68.7	**90.9**
Left turn	69.2	**86.8**	81.0	85.1
Left lane change	64.3	**91.8**	77.9	91.4
Right turn	63.2	85.6	**86.0**	83.8
Right lane change	65.5	84.0	79.0	**86.3**
Overall accuracy	66.2	87.2	78.1	**87.5**

VI. Conclusions

In this work, we proposed a novel method to predict driver intentions across 5 driving maneuvers that fuses hand-crafted feature representations of the in-cabin and exterior cameras. Since driver intentions are, in general, difficult to predict, we show that prediction accuracy can be improved by incorporating multiple sources of information that the driver is likely considering instead of limiting the model to somatic information from the driver and dynamic information from the vehicle. We illustrate that our selection of external features complements the in-cabin features, which is different from previous methods that rely on learned exterior features. Our approaches substantially surpass the state-of-the-art methods in several key metrics. The F-LSTM and F-TF architectures achieve an accuracy of 87.2% and 87.5% and are able to correctly predict the driver intention an average of 4.34 and 4.35 seconds before the maneuver occurs. This provides a key insight about what features are important for understanding driver intent. Interesting future directions to improve performance include developing better data augmentation strategies for additional data diversity, leveraging prior knowledge from pre-trained LSTM architectures to boost transformer learning efficiency, and expanding the driver action space.

Acknowledgements

This work is partially supported in part by NSF (EFMA-2318065, IIS-1954778, CAREER-2046955).

References

[1] S. E. Lee, E. C. B. Olsen, and W. W. Wierwille. “A comprehensive examination of naturalistic lane changes”. In: *Report DOT HS 809 702, Washington DC: NHTSA, U.S. Department of Transportation* (2004).

[2] A. Doshi and M. Trivedi B. Morris. “On-road prediction of driver’s intent with multimodal sensory cues”. In: *IEEE Pervasive Computing* 10.3 (2011), pp. 22–34. DOI: 10.1109/MPRV.2011.38.

[3] N. Zhao et al. “Safety Prompt Advanced Driver-Assistance System with Lane-Change Prediction and Free Space Detection”. In: *2022 IEEE 17th International Conference on Control & Automation (ICCA)* (2022), pp. 734–739. DOI: 10.1109/ICCA54724.2022.9831811.

[4] M. Hofbauer et al. “Measuring Driver Situation Awareness Using Region-of-Interest Prediction and Eye Tracking”. In: *2020 IEEE International Symposium on Multimedia (ISM)* (2020), pp. 91–95. DOI: 10.1109/ISM.2020.00022.

[5] A. Kar and P. Corcoran. “A Review and Analysis of Eye-Gaze Estimation Systems, Algorithms and Performance Evaluation Methods in Consumer Platforms”. In: *IEEE Access* 5 (2017), pp. 16495–16519. DOI: 10.1109/ACCESS.2017.2735633.

[6] P. Gebert et al. “End-to-end Prediction of Driver Intention using 3D Convolutional Neural Networks”. In: *2019 IEEE Intelligent Vehicles Symposium (IV)* (2019), pp. 969–974. DOI: 10.1109/IVS.2019.8814249.

[7] N. Khairdoost, M. A. Bauer M. Shirpour, and S. S. Beauchemin. “Real-Time Driver Maneuver Prediction Using LSTM”. In: *IEEE Transactions on Intelligent Vehicles* 5.4 (2020), pp. 714–724. DOI: 10.1109/TIV.2020.3003889.

[8] V. Leonhardt, T. Peck, and G. Wanielik. “Data fusion and assessment for maneuver prediction including driving situation and driver behavior”. In: *2016 19th International Conference on Information Fusion (FUSION)* (2016), pp. 1702–1708.

[9] A. Jain et al. “Brain4Cars: Car That Knows Before You Do via Sensory-Fusion Deep Learning Architecture”. In: *CoRR* abs/1601.00740 (2016). arXiv: 1601.00740. URL: http://arxiv.org/abs/1601.00740.

[10] N. Zhao et al. “Direction Convolutional LSTM Network: Prediction Network for Drivers’ Lane-Changing Behaviours”. In: *2022 IEEE 17th International Conference on Control & Automation (ICCA)* (2022), pp. 752–757. DOI: 10.1109/ICCA54724.2022.9831900.

[11] Z. Hao et al. “Attention -Based GRU for Driver Intention Recognition and Vehicle Trajectory Prediction”. In: *2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI)* (2020), pp. 86–91. DOI: 10.1109/CVCI51460.2020.9338510.

[12] S. M. Zabihi, S. S. Beauchemin, and M. A. Bauer. “Real-time driving manoeuvre prediction using IO-HMM and driver cephalo-ocular behaviour”. In: *2017 IEEE Intelligent Vehicles Symposium (IV)* (2017), pp. 875–880. DOI: 10.1109/IVS.2017.7995826.

[13] L. Li and P. Li. “Analysis of Driver’s Steering Behavior for Lane Change Prediction”. In: *2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)* (2019), pp. 71–75. DOI: 10.1109/IHMSC.2019.10112.

[14] A. Jain et al. “Recurrent neural networks for driver activity anticipation via sensory-fusion architecture”. In: *IEEE Internation Conference on Robotics and Automation* (2016), pp. 3118–3125.

[15] Arthur Zhang et al. “Towards robust robot 3d perception in urban environments: The ut campus object dataset”. In: *arXiv preprint arXiv:2309.13549* (2023).

[16] Shilong Liu et al. *Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection*. 2023. arXiv: 2303.05499 [cs.CV].

[17] Fakultit Informatik et al. “Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies”. In: *A Field Guide to Dynamical Recurrent Neural Networks* (Mar. 2003).

[18] Ashish Vaswani et al. “Attention Is All You Need”. In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

[19] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *CoRR* abs/2010.11929 (2020). arXiv: 2010.11929. URL: https://arxiv.org/abs/2010.11929.