Target-point Attention Transformer: A novel trajectory predict network for end-to-end autonomous driving

Jingyu Du¹, Yang Zhao*1 and Hong Cheng¹

Abstract—In the field of autonomous driving, there have been many excellent perception models for object detection, semantic segmentation, and other tasks, but how can we effectively use the perception models for vehicle planning? Traditional autonomous vehicle trajectory prediction methods not only need to obey traffic rules to avoid collisions, but also need to follow the prescribed route to reach the destination. In this paper, we propose a Transformer-based trajectory prediction network for end-to-end autonomous driving without rules called Target-point Attention Transformer network (TAT). We use the attention mechanism to realize the interaction between the predicted trajectory and the perception features as well as target-points. We demonstrate that our proposed method outperforms existing conditional imitation learning and GRU-based methods, significantly reducing the occurrence of accidents and improving route completion. We evaluate our approach in complex closed loop driving scenarios in cities using the CARLA simulator and achieve state-of-the-art performance.

I. INTRODUCTION

With the continuous development of deep learning, endto-end autonomous driving is gradually becoming a hot topic[30], [19], [31], [15], [38]. End-to-end autonomous driving uses neural networks to directly map sensor inputs (cameras, LIDAR, IMU, etc.) to future trajectories[4], [8], [29], [11], [24], [9] or low-level control actions(e.g. throttle, brake and steering angle)[14], [26], [13], [28], [7], [42], [6], eliminating the need for complex rule base design. It is certainly exciting, as we know that rule bases are hardly designed to cover the full range of situations. Currently, the model for end-to-end autonomous driving can be thought of as an encoder-decoder structure[2], [9], [33], where the encoder is responsible for encoding the information around the vehicle, and the decoder uses the encoded information for future trajectory or low-level control action prediction. For the encoders, there have been developments from the original ResNet networks[14], [13] to the multi-stage fusion of images and radar called Transfuser[29], and the interpretable network called Interfuser[33]. However, there have been few improvements for decoders, especially the method of future trajectory prediction.

Unlike trajectory prediction tasks in other fields, for the autonomous driving task, the vehicle needs to make the correct decision at intersections, such as going straight, turning left, or turning right, based on advanced navigation commands(e.g. follow lane, turn right/left, change lane). Currently, there are two main types of decoder structures:

conditional imitation learning network (CIL)[14] and GRUbased[12] network. Codevilla et al. proposed CIL structure, which trains multiple decoders to be responsible for the straight, left or right turn of the vehicle according to the advanced navigation commands. The disadvantages of this approach are obvious, firstly, training multiple networks for similar functions results in significant redundancy and waste of computational resources. In addition, this type of approach often performs poorly when passing curves due to dataset bias, making the completion of routes often low. Prakash et al.[29] have proposed GRU-based methods that no longer use high-level navigation commands but instead use highlevel target-points for vehicle navigation (these points are sparse and they can be hundreds of metres apart). However, passing the perception features from the encoder to the GRU cell usually requires a dimensionality reduction of the features, this dimensionality reduction can result in a loss of important perceptual features, which can lead to suboptimal predictions and potentially dangerous driving behavior, such as driving off the road or colliding with other vehicles, as shown in Figure 1. The recent success of Transformer structures for pedestrian trajectory prediction[22], [41], [43], [32] demonstrates the potential of Transformer for trajectory prediction. Compared with RNN networks such as GRU, Transformer can input higher dimensional perception features and its attention mechanism is more conducive to smooth trajectory prediction. To this end, we propose the Target-point Attention Transformer network(TAT), witch uses the Transformer's attention mechanism to directly interact with the 2D perceptual features and predict the future trajectory. This approach preserves more of the relevant perceptual features and enables smoother, more accurate trajectory predictions. As a result, TAT significantly reduces the occurrence of accidents and improves the vehicle's ability to complete the route successfully. To the best of our knowledge, we are the first to propose using Transformer for end-to-end autonomous driving trajectory prediction.

The contributions can be summarized as follows:

- (1) The proposed Target-point Attention Transformer model utilizes attention mechanism to predict the future trajectory of autonomous vehicles, which significantly reduces the occurrence of collisions and improves route completion.
- (2) Quantitative experiments are conducted on the CARLA, different trajectory prediction methods are compared and analyzed. Experiment results demonstrate the effectiveness of the proposed method.

The structure of this paper is as follows. In Section II, we present a comprehensive review of the related works con-

¹Jingyu Du, Yang Zhao, and Hong Cheng are with School of Automation Engineering, University of Electronic Science and Technology of China, 611731 Chengdu, China.

^{*} Corresponding Author: vzhao@uestc.edu.cn

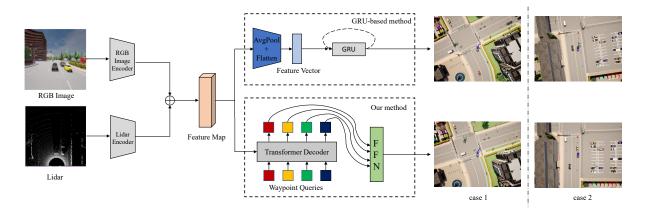


Fig. 1. The GRU-based trajectory prediction network commonly adopt an AvgPool layer and a Flatten layer to convert 2D perception feature maps into 1D feature vectors. This dimensionality reduction can result in a loss of important perceptual features,, which may cause the autonomous vehicle to be unable to complete the route normally (case 1 driving off the road, case 2 colliding with other vehicle, blue points are predicted waypoints, red points are target-points). Our method employs the Transformer's attention mechanism to directly interact with the 2D perceptual features and predict the future trajectory. Our method significantly reduces the occurrence of accidents and improves the completion of the route.

cerning end-to-end driving models and trajectory forecasting using Transformer. In Section III, we describe our proposed method in detail. Section IV outlines the experimental setup, and Section V presents the results. Finally, we conclude this paper in Section VI.

II. RELATED WORK

A. End-to-end Autonomous Driving

With the continuous development of deep learning, learning-based end-to-end autonomous driving has become an active research topic. Currently, research in this field can be broadly categorized into two areas: reinforcement learning (RL) and imitation learning (IL). The RL methods in automatic driving is often combined with IL method. Liang *et al.*[26] uses the supervision method to pretrain the model firstly, and then uses the Deep Deterministic Policy Gradient (DDPG) strategy for training. [34], [6], [44] use the pretrained perception model to perform the perception task in RL. While RL has the potential to address the issue of dataset distribution shifts[7], their training complexity often limits their practical applications.

Imitation learning realizes automatic driving by imitating the behavior of experts. The neural network is often trained under the supervision of experts' behaviors by collecting the data of cameras, laser radars and other sensors as input[4], [8], [29], [11]. And the data of experts behaviors typically has two forms, actions and trajectories. Nvidia proposed PilotNet[3], which directly maps the image pixels of a single forward facing camera to the steering command via CNN. Codevilla et al.[14] add a measurement encoder to fuse the vehicle status features, e.g. current speed and location. And it proposed conditional imitation learning with using multiple branches for different high level commands to drive to cross the intersection. Codevilla et al.[13] proposed an improved method based on CIL, it use a speed prediction head to improve the inertia problem of end-to-end automatic driving[36]. The above methods directly predict actions.

However, these methods usually can only predict the action of the current time step, which lead to the actions of vehicle more unstable and discontinuous. In addition, such methods are difficult for data augmentation. Wu *et al.*[37] presented a multi-task learning(MTL) method, it predicts the future trajectories first, and predict future actions through attention mechanism. It realized multi-step action prediction and achieved good results on the public CARLA Leaderboard[1].

For the trajectory methods, Chen et al.[8] introduced a knowledge distillation method. It utilizes privileged information (such as Bird's Eye View (BEV) map) to train a privileged agent, which generates a set of heatmaps. These heatmaps are passed through a soft-argmax layer (SA) to obtain waypoints for all commands. The sensorimotor agent is then trained using the privileged agent as a supervisor. Chitta et al.[11] proposed a method that uses attention maps to compress high dimensional 2D image features into a more compact BEV representation for autonomous driving. It achieves this by using a series of intermediate attention maps that iteratively process the input image. NEAT also employs a waypoint offset prediction map to transform the discretely predicted waypoints into a dense prediction task. Transfuser[29] use a multi-stage CNN-transformer to fuse the RGB image and LIDAR, then it uses a single GRU to autoregress waypoints. Learning from All Vehicles (LAV)[9] also adopted a temporal GRU module to auto-regress waypoints. Note that such methods usually require a PID controller to convert waypoints into low-level control actions. And the trajectory-based methods have the advantage of providing smoother and more continuous driving behaviors than action-based methods. Additionally, trajectory-based methods can be more easily extended to incorporate additional constraints[33], such as safety or efficiency requirements, by adjusting the trajectory planning algorithm.

B. Transformer in trajectory forecasting

Transformer first proposed in the field of Natural Language Processing[35], [16], [25], [40], [39], and it quickly

dominated this field with its unique attention mechanism. The ability of Transformer to parallelize computation and capture long-term dependencies has made it a popular choice for sequence modeling tasks. Recently the transformer architecture has also achieved success in the field of Computer Vision[18], [5], [21], [27], [20] and trajectory prediction[22], [41]. Giuliari *et al.*[22] use the basic transformer decoder architecture for pedestrian trajectory prediction, and achieve SOTA on multiple datasets. Compared to RNN, Transformer has the ability of parallel training, and can process higher dimensional features, these make the transformer performs better than RNN in the field of trajectory forecasting.

III. METHOD

In our work, we propose a novel waypoint prediction network for end-to-end autonomous driving. The following sections briefly introduce the Transfuser backbone and detail the design of our waypoint prediction network.

A. Problem Setting

The goal of whole network is to learn a policy π so that the vehicle can reach the destination u_G along the predefined route, $u_1^G \in (u_1,...,u_g,...,u_G), u_g \in \mathbb{R}^2$ and can obey the traffic rules and avoid collisions with other traffic participants.

For the perception backbone, the goal is to encode the high dimensional observations of environment, \mathcal{X} , to the lower perception features \mathcal{F} . Then the waypoint prediction network takes perception features \mathcal{F} and navigation points u in to generate the future waypoints $\mathcal{W} = \{w_i\}_{i=1}^Z$, Z is the number of prediction waypoints.

The policy π is trained in a supervised manner using the collected data D, with the loss function \mathcal{L} .

$$\underset{\pi}{argmin} \mathbb{E}_{(\mathcal{X}, \mathcal{W}, u) \sim \mathcal{D}}[\mathcal{L}(\mathcal{W}, \pi(\mathcal{X}, u))] \tag{1}$$

Assuming access to an inverse dynamics model implemented as a PID controller for low-level control, the system can generate the necessary steering, throttle, and brake commands given the predicted future waypoints \mathcal{W} . The actions can be determined as $a = \theta(\mathcal{W})$.

Input Representation: Following Tranfuser, our model takes an RGB front camera image and Lidar point cloud as input. For the image $x_t \in \mathbb{R}^{H \times W \times C}$ where H = W = 256, C = 3, we use the parameters on ImageNet for normalization. For the Lidar point cloud, it converted to a histogram pseudo-image $v_t \in \mathbb{R}^{H \times W \times D}$, where D = 2. Since we consider the points within 32m in front of the egovehicle and 16m to each of the sides, each grid represents $0.125m \times 0.125m$ area. And we use the horizontal plane as the boundary to divide the points on the horizontal plane in the first dimension, and other divisions in the second dimension.

Output Representation: We predict the future waypoints \mathcal{W} of the ego vehicle. $\mathcal{W} = \{w_i\}_{i=1}^Z$ is in the coordinate system with the ego vehicle as the origin, the front is the positive direction of the x-axis, and the left is the positive direction of the y-axis. Following [10], we take Z = 4.

B. Transfuser Backbone

Transfuser use ResNet[23] to process the image x_t and the LIDAR BEV v_t . During processing, several Transformer modules are used to fuse the intermediate feature maps between both modalities. The Transformer module takes the intermediate features as token, the output is split into two parts and respectively sum with the existing feature maps. Note that the Transformer modules work at 8×8 resolution, so there is a downsampling operation at the input and an upsampling operation at the output. For more details we refer to [29], [10].

The original Transfuser model uses the average pool layer and flatten layer to reduce the extracted 2D feature map to 1D feature vector so that the perception features can pass into GRU cell, this processing may lead to a large loss of information, which may cause the autonomous vehicle to be unable to complete the route normally. In the architecture illustrated in Figure 2, we remove the final avgpool layer and flatten layer to produce a full-scale perception feature map. This map is then fed into our Transformer decoder-based waypoint prediction network, which is described in detail in the following section.

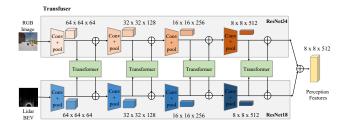


Fig. 2. We discard the last avgpool layer and flatten layer of Transfuser to output full scale perception feature map. In addition, we cancel the speed input during fusion because it is one of the reason for the inertia problem[10]

C. Target-point Attention Waypoint Prediction Network

Our model takes the full-scale perception features $\mathcal{F} \in \mathbb{R}^{H_0 \times W_0 \times d_{\mathrm{model}}}$ as the input to predict the future trajectory of the ego vehicle. Our key idea is to use Transformer's self-attention mechanism to build dependency between waypoints and target-points, and use cross attention mechanism to build dependency between waypoints and perception features. The overall structure of the our network is shown in the Figure 3.

Self-Attention: As we need to predict a set $\mathcal{W} = \{w_i \in \mathbb{R}^2\}_{i=1}^Z$ waypoints, we use a set of waypoints queries $\mathcal{Q}_l = \{e_{wi} \in \mathbb{R}^{dmodel}\}_{i=1}^Z$ to producing a new set \mathcal{Q}_{l+1} in the each layer. And for the target-points, we embedding them onto a higher d_{model} -dimensional space by a linear layer without bias, i.e, $e_t = Linear(u)$. Since vehicle trajectory prediction involves predicting a sequence of future positions, it is important to include a position encoding that captures the temporal information of each past and future time step. More formally, the input embedding e_w is time-stamped at time t by adding a positional encoding vector p^t of the same

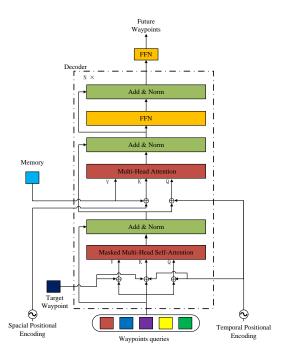


Fig. 3. Architecture of TAT, We added the target-point vector during the calculation of self-attention, so that the self-driving vehicle can pass the intersection correctly. And the cross-attention mechanism of Transformer can complete the interaction between waypoints and perception features.

dimensionality d_{model} : $E = p^t + e_w$. Following[22], we use sine/cosine functions to define p^t .

$$p^{t} = \{p_{t,d}\}_{d=1}^{d_{\text{model}}}$$

$$where \quad p_{t,d} = \begin{cases} \sin(\frac{t}{10000^{d/d_{\text{model}}}}) & \text{for } d \text{ even} \\ \cos(\frac{t}{10000^{d/d_{\text{model}}}}) & \text{for } d \text{ odd} \end{cases}$$

$$(2)$$

And for the target-point embedding, we added a learnable encoding $U = u + p^u$.

Linear projections are employed in the Transformer architecture to calculate a group of queries, keys, and values (Q, K, and V). In order to compute attention between the waypoints and the target-point, our self-attention calculation is different from other Transformer structures:

$$\begin{aligned} Q_a &= E \cdot M_a^q \\ K_a &= Concat(E,U) \cdot M_a^k \\ V_a &= Concat(e_w,u) \cdot M_a^v \end{aligned} \tag{3}$$

Where $M_a^q \in \mathbb{R}^{Z \times d_{\mathrm{model}}}, M_a^k \in \mathbb{R}^{(Z+1) \times d_{\mathrm{model}}}, M_a^v \in R^{(Z+1) \times d_{\mathrm{model}}}$ are weight matrices.

Then the transformer utilizes the dot product of scaled query (Q) and key (K) matrices to compute the attention weights, and subsequently multiplies the attention weights with the corresponding value (V) matrix to obtain the final output.

$$SA = softmax(\frac{Q_a K_a^T}{\sqrt{d_{\text{model}}}})V$$
 (4)

Cross-Attention: Our cross-attention mechanism is similar to VIT[18]. For the feature map output by Encoder $\mathcal{F} \in \mathbb{R}^{H_0 \times W_0 \times d_{\mathrm{model}}}$, we reshape it into a sequence of flattened 2D patches $memory \in \mathbb{R}^{N \times d_{\mathrm{model}}}$, where $N = H_0 \times W_0$ is the number of patches. And a learnable positional embedding is added to the patch to retain positional information. The calculation of Q, K, V matrices can be expressed by the following formula:

$$Q_c = SA \cdot M_c^q$$

$$K_c = memory \cdot M_c^k$$

$$V_c = memory \cdot M_c^v$$
(5)

Then the cross-attention are computed according to Eq(6). Similar to the GRU-based method, our model predicts the waypoints offset rather than directly predicting the waypoints. Finally, we use the a FFN layer to back project the output of the decoder to the Cartesian person coordinates.

Implement Details: Since the output dimension of Encoder is 512, we set the $d_{\rm model}$ to 512, and have 4 layers and 8 attention heads. For the activation function, because the network may input and output negative values, we tested different activation functions that can retain negative information and chose the best one: Leaky ReLU. Our FFN layer has 3 layers and also use the Leaky ReLU activation function. Our network was trained for 100 epochs using the AdamW optimizer with a learning rate of 0.0001. We applied a learning rate reduction of 0.1 at the 40th and 70th epochs.

D. Loss Function

we use the L2 loss between the predicted waypoint and the ground truth waypoint to train the network. For the time-step t, the loss function is given by:

$$\mathcal{L} = \sum_{t=1}^{T} \| w_t - w_t^{gt} \|_2 \tag{6}$$

IV. EXPERIMENTS

A. Task and Metrics

We conducted all of our experiments on the CARLA simulator[17], which provides a realistic urban driving environment. In this environment, the autonomous agent is required to follow a predefined route and navigate through various scenarios, including pedestrian crossings and obstacle avoidance, while receiving sparse goal locations in GPS coordinates (which we refer to as target-points) and discrete navigational commands.

To evaluate the performance of our method, we used the three metrics provided by the CARLA leaderboard: Driving Score (DS), Route Completion (RC), and Infraction Score (IS). RC represents the percentage of the route that the autonomous agent successfully completed, while IS measures the number of infractions made along the route, such as collisions with pedestrians, other vehicles, or traffic violations. The main metric, DS, is the product of RC and IS,

TABLE I
DRIVING PERFORMANCE COMPARISON

Method	Town05 Short			Town05 Long		
	Driving Score↑	Route Completion↑	Infraction Score↑	Driving Score↑	Route Completion↑	Infraction Score↑
TAT-RT(ours)	44.68	79.44	0.59	11.28	40.61	0.51
TAT-CT(ours) TAT-RR(ours)	48.21 10.26	92.53 67.43	0.53 0.15	11.31 5.23	69.08 40.24	0.29 0.14
TAT-CR(ours)	29.96	88.71	0.33	7.45	53.10	0.14
CILRS[13] AIM[29] Transfuser[29]	7.44 20.62 43.31	13.56 50.16 82.97	0.49 0.39 0.55	2.12 5.12 9.68	8.12 30.54 66.66	0.14 0.16 0.19

TABLE II
INFRACTIONS FREQUENCY COMPARISON

Method	Driving score	Collisions with pedestrians	Collisions with vehicles	Collisions with layout	Red lights infractions	Off-road infractions	Agent blocked
		#/Km↓	#/Km↓	#/Km↓	#/Km↓	#/Km↓	#/Km↓
TAT-RT(ours)	44.68	0.0	1.97	0.0	27.17	2.18	8.30
TAT-CT(ours)	48.21	0.48	3.82	0.0	26.75	2.33	1.43
Transfuser[29]	43.31	1.63	6.92	2.66	20.77	6.42	4.79

and it provides an overall evaluation of the performance of the autonomous driving system.

B. Dataset

As CARLA provide 8 towns, we use 7 towns for training and hold out Town05 for evaluation. We generate routes randomly but the weather is fixed to ClearNoon. The routes are divided into Short routes of 100-500m and Long routes of 1000-2000m, and we ran a rule-based expert that can access privileged information to collect data at 2 FPS. The entire dataset contains 160k data.

C. Baselines

We compare our model with the following baselines. (1) CILRS[13] predicts vehicle controls from a single front camera image while being conditioned on the navigational command. (2) AIM[29] use GRU-based waypoint prediction network with an image-based ResNet-34 encoder. It predicts future trajectory instead of vehicle controls, and it uses sparse goal locations as input instead of navigational commands. (3) Transfuser[29] use multi-stage Transformer to fuse image and lidar features at multi scales, and a GRU-based autoregression is used to generate future trajectory. It achieves SOTA on the CARLA leaderboard.

V. RESULTS

A. Comparison with baselines

TABLE I presents the driving performance comparison between our method and the baselines we introduced earlier. Meanwhile, TABLE II shows the comparison between our model and Transfuser in terms of infractions frequency. We evaluated four variants of our model: TAT-RT uses Transfuser as the backbone and auto-regressively predicts trajectory, TAT-CT uses Transfuser as the backbone and predicts trajectory as classification, TAT-RR uses ResNet34 as the

backbone and auto-regressively predicts trajectory, and TAT-CR uses ResNet34 as the backbone and predicts trajectory as classification. The CIL-based method CILRS was found to be unsuitable for complex urban scenes, as it exhibited an extremely low driving score and route completion rate. We suspect that the main reason is the imbalance of the dataset. On the other hand, the ResNet network struggled to complete the perception task of autonomous driving, resulting in low driving scores for all methods. Nevertheless, our method outperformed AIM, with a 45.30% improvement in driving score and a 76.85% increase in route completion rate in the short route, and a 45.51% improvement in driving score and a 73.87% increase in route completion rate in the long route. It's important to note that a higher route completion rate indicates a higher chance of accidents, which can lead to a decline in driving score.

When compared to the original Transfuser model, our method increased the driving score by 11.31% and the route completion rate by 11.52% in the short route, and by 16.84% and 3.63%, respectively, in the long route. As for infractions frequency, our method reduced collision by 61.64%, off-road infractions by 63.71%, and the occurrence of agent blockages by 70.15%. These results show that our method can make more effective use of perception features than GRU-based method, thus making more reasonable trajectory prediction. However, our method had a higher incidence of red light infractions. This could be due to the Transfuser perception model struggling to recognize traffic lights, as the original Transfuser model also had a high rate of running red lights.

B. Classification Vs Auto-regression

In the field of trajectory forecasting, a recurring question is whether to approach the problem as an auto-regression or a classification task. The auto-regressive method generates future trajectory points one at a time, conditioning on the

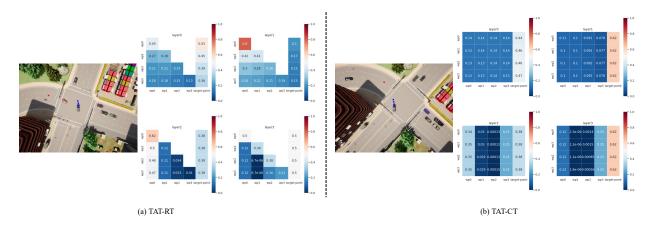


Fig. 4. Self-Attention visualization of TAT-RT and TAT-CT, blue points are predicted waypoints, red points are target-points. The dependence of the auto-regressive method on the predicted waypoints leads to insufficient attention to the target-point, which may be the reason for the insufficient steering of the auto-regressive method

previously generated points, and is beneficial to the trajectory's continuity but has the problem of error accumulation. On the other hand, the classification method predicts all path points at once. In pedestrian trajectory prediction, the auto-regressive approach has been shown to be superior[22]. However, in end-to-end autonomous driving, our research has reached the opposite conclusion.

In the Town05 short evaluation setting, compared to TAT-RT, TAT-CT has a 7.90% higher driving score and a 16.48% higher route completion rate, but has an 11.32% lower infractions score. Specifically, the auto-regressive method can better avoid collisions (54.19% lower collision rate), but the occurrence of agent blockages increased by 82.78%. Figure 4 shows that insufficient steering is the main reason for the auto-regressive method's agent blocked, and this may be due to insufficient attention to the target point or error accumulation.

C. Ablation study

In our default configuration, we use Leaky ReLU activation function and add 1D sine/cosine positional embedding to waypoint queries and add learnable 2D positional embedding to memory. In this section, we present ablation on target-point attention architecture, activation function and the positional embedding, all within the context of the Town05 Short evaluation setting.

Is target-point attention necessary? To assess the effectiveness of the target-point attention structure, we compared it with an alternative approach that embeds the target point information into the waypoints queries (TET). However, we found that TET struggles to complete the route successfully. Notably, TET fails to navigate intersections accurately, suggesting that simply encoding target-points as position embeddings may not be sufficient to provide the network with enough target-point information. Our findings underscore the importance of the target-point attention structure for achieving successful end-to-end autonomous driving.

Which activation function should be selected? Since our model deals with negative values in both input and

TABLE III
ABLATION STUDY

Method	Driving	Route	Infraction	
	Score↑	Completion↑	Score↑	
TAT(default)	48.21	92.53	0.53	
TET	21.33	30.96	0.70	
TAT-gelu	36.51	64.60	0.59	
TAT-tanh	46.63	88.57	0.53	
TAT-No Query PE	38.31	69.55	0.58	
TAT-No Memory PE	36.52	64.60	0.60	
TAT-2D Memory PE	37.89	75.92	0.55	

output, selecting the proper activation function is crucial. We evaluated three activation functions, namely tanh, Gelu, and Leaky ReLU, and observed that Leaky ReLU has the best performance. We found that the performance of Gelu significantly decreased, possibly due to less retention of negative value information. Meanwhile, the performance of tanh was slightly lower than that of Leaky ReLU, which might be caused by gradient disappearance and other issues.

Is the positional embedding useful? We expect that positional embedding can help the model understand the order between waypoints and the spatial dependency of the surrounding environment of the vehicle. TABLE III shows that positional embedding is indeed effective. No query PE or memory PE will result in significant performance degradation, and the performance of 2D memory PE is lower than that of learnable PE.

VI. CONCLUSION

In this paper, we proposed a novel trajectory prediction network for end-to-end autonomous driving. We demonstrate that the existing GRU-based trajectory prediction network fails to fully leverage the available perception features. To address this limitation, we propose a novel trajectory prediction network that leverages Transformer's attention mechanism to directly interact with high-dimensional perception features, and achieve state-of-the-art performance on CARLA. Our

method is versatile and adaptable, and we plan to investigate further improvements by exploring new perception networks, such as separate traffic light detection networks to mitigate the issue of running red lights.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (U1964203), National Key Research and Development Program of China (No.2022YFB2503004), Sichuan Science and Technology Program (NO.2022YFG0342).

REFERENCES

- Carla autonomous driving leaderboard. https://leaderboard.carla.org/, 2022.
- [2] Mayank Bansal, Alex Krizhevsky, and Abhijit S. Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. ArXiv. abs/1812.03079, 2018.
- [3] Mariusz Bojarski and David W. del Testa et al. End to end learning for self-driving cars. ArXiv, abs/1604.07316, 2016.
- [4] Mariusz Bojarski and Chenyi Chen et al. The nvidia pilotnet experiments. ArXiv, abs/2010.08776, 2020.
- [5] Nicolas Carion and Francisco Massa et al. End-to-end object detection with transformers. ArXiv, abs/2005.12872, 2020.
- [6] Raphael Chekroun and Marin Toromanoff et al. Gri: General reinforced imitation and its application to vision-based autonomous driving. ArXiv. abs/2111.08575, 2021.
- [7] Di Chen, Vladlen Koltun, and Philipp Krähenbühl. Learning to drive from a world on rails. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pages 15570–15579, 2021.
- [8] Dian Chen and Brady Zhou et al. Learning by cheating. ArXiv, abs/1912.12294, 2019.
- [9] Dian Chen and Philipp Krähenbühl. Learning from all vehicles. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 17201–17210, 2022.
- [10] Kashyap Chitta and Aditya Prakash et al. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. IEEE transactions on pattern analysis and machine intelligence, PP, 2022.
- [11] Kashyap Chitta, Aditya Prakash, and Andreas Geiger. Neat: Neural attention fields for end-to-end autonomous driving. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pages 15773– 15783, 2021.
- [12] Kyunghyun Cho and Bart van Merrienboer et al. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In Conference on Empirical Methods in Natural Language Processing, 2014.
- [13] Felipe Codevilla and Eder Santana et al. Exploring the limitations of behavior cloning for autonomous driving. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 9328–9337, 2019.
- [14] Felipe Codevilla and Matthias Müller et al. End-to-end driving via conditional imitation learning. 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 1–9, 2017.
- [15] Alexander Cui and Abbas Sadat et al. Lookout: Diverse multi-future prediction and planning for self-driving. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pages 16087–16096, 2021.
- [16] Jacob Devlin and Ming-Wei Chang et al. Bert: Pre-training of deep bidirectional transformers for language understanding. ArXiv, abs/1810.04805, 2019.
- [17] Alexey Dosovitskiy and Germán Ros et al. Carla: An open urban driving simulator. ArXiv, abs/1711.03938, 2017.
- [18] Alexey Dosovitskiy and Lucas Beyer et al. An image is worth 16x16 words: Transformers for image recognition at scale. ArXiv, abs/2010.11929, 2020.
- [19] Angelos Filos and Panagiotis Tigas et al. Can autonomous vehicles identify, recover from, and adapt to distribution shifts? 2020.
- [20] Valentin Gabeur and Chen Sun et al. Multi-modal transformer for video retrieval. In European Conference on Computer Vision, 2020.
- [21] Peng Gao and Minghang Zheng et al. Fast convergence of detr with spatially modulated co-attention. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pages 3601–3610, 2021.

- [22] Francesco Giuliari and Irtiza Hasan et al. Transformer networks for trajectory forecasting. 2020 25th International Conference on Pattern Recognition (ICPR), pages 10335–10342, 2020.
- [23] Kaiming He and Xiangyu et al. Zhang. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.
- [24] Bernhard Jaeger. Expert drivers for autonomous driving. Master's thesis, University of Tübingen, 2021.
- [25] Zhenzhong Lan and Mingda Chen et al. Albert: A lite bert for self-supervised learning of language representations. ArXiv, abs/1909.11942, 2019.
- [26] Xiaodan Liang and Tairui Wang et al. Cirl: Controllable imitative reinforcement learning for vision-based self-driving. ArXiv, abs/1807.03776, 2018.
- [27] Ying-Hao Liu and Tiancai Wang et al. Petr: Position embedding transformation for multi-view 3d object detection. In European Conference on Computer Vision, 2022.
- [28] Eshed Ohn-Bar and Aditya Prakash et al. Learning situational driving. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 11293–11302, 2020.
- [29] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 7073–7083, 2021.
- [30] Nicholas Rhinehart, Rowan McAllister, and Sergey Levine. Deep imitative models for flexible inference, planning, and control. ArXiv, abs/1810.06544, 2018.
- [31] Abbas Sadat and Sergio Casas et al. Perceive, predict, and plan: Safe motion planning through interpretable semantic representations. In European Conference on Computer Vision, 2020.
- [32] Khaled Saleh. Pedestrian trajectory prediction using contextaugmented transformer networks. ArXiv, abs/2012.01757, 2020.
- [33] Hao-Chiang Shao and Letian Wang et al. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. ArXiv, abs/2207.14024, 2022.
- [34] Marin Toromanoff, Émilie Wirbel, and Fabien Moutarde. End-to-end model-free reinforcement learning for urban driving using implicit affordances. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 7151–7160, 2019.
- [35] Ashish Vaswani and Noam M. Shazeer et al. Attention is all you need. ArXiv, abs/1706.03762, 2017.
- [36] Chuan Wen and Jierui Lin et al. Fighting copycat agents in behavioral cloning from observation histories. ArXiv, abs/2010.14876, 2020.
- [37] Peng Wu and Xiaosong Jia et al. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. ArXiv, abs/2206.08129, 2022.
- [38] Huazhe Xu and Yang Gao et al. End-to-end learning of driving models from large-scale video datasets. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3530–3538, 2016.
- [39] Zhilin Yang and Zihang Dai et al. Xlnet: Generalized autoregressive pretraining for language understanding. In *Neural Information Pro*cessing Systems, 2019.
- [40] Tom Young and Devamanyu Hazarika et al. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13:55–75, 2017.
- [41] Cunjun Yu and Xiao Ma et al. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In European Conference on Computer Vision, 2020.
- [42] Zhejun Zhang and Alexander Liniger et al. End-to-end urban driving by imitating a reinforcement learning coach. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pages 15202–15212, 2021
- [43] Jingwen Zhao and Xuanpeng Li et al. Spatial-channel transformer network for trajectory prediction on the traffic scenes. ArXiv, abs/2101.11472, 2021.
- [44] Yinuo Zhao and Kun Wu et al. Cadre: A cascade deep reinforcement learning framework for vision-based autonomous urban driving. In AAAI Conference on Artificial Intelligence, 2022.