2024 IEEE Intelligent Vehicles Symposium (IV)
June 2-5, 2024. Jeju Island, Korea

2024 IEEE Intelligent Vehicle Symposium (IV) | 979-8-3503-4881-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/IV55156.2024.10588479

Pix2Planning: End-to-End Planning by Vision-language Model for Autonomous Driving on Carla Simulator

Xiangru Mu, Tong Qin*, Songan Zhang, Chunjing Xu, and Ming Yang

***Abstract*— The end-to-end neural network has become a hot topic in recent years. Compared with traditional module-based solutions, the end-to-end paradigm is able to reduce the accumulated error and avoid information loss, so that it earns great attention in autonomous driving tasks. However, the current end-to-end network designs easily lose useful information during training due to the complexity of mapping high-dimensional visual observation to navigation waypoints. Since the future navigation point is reasoned from the former one, the planning task is like a sequence generation task. Inspired by the great power of the neural language model, we propose an end-to-end framework, which transfers the planning task as a language sequence generation task conditioned on pixel inputs. The proposed method firstly extracts and transforms the image feature from camera-view to bird-eye-view (BEV). Then the target navigation point is constructed into a text sequence, as the prompt of the visual-language transformer. Finally, the auto-regressive transformer decoder receives the BEV feature and the text sequences to generate sequential waypoints. Overall, our proposed method can make full use of the environmental information and express the planning trajectory as a language sequence to learn the correspondence between trajectory sequences and images. We have conducted extensive experiments on CARLA benchmarks and our model achieves state-of-the-art performance compared with other visual methods.**

I. Introduction

The typical module-based autonomous driving solutions usually consist of multiple parts, such as perception, prediction, decision-making, planning and control. These modules are independent and connected in series. The advantage of this cascade structure is that the system has a clear division of functions and every module takes clear responsibility. It performs well in simple scenarios where missions are fully under the control of each module. However, when the driving scenarios becomes complicated and tough, some modules cannot handle their own tasks well and the error accumulated and is amplified step by step. The downstream modules trust the noisy output from the upstream modules and make the wrong decision. Therefore, the system is inflexible to various scenarios and fails to generalize at scale.

Compared with module-based autonomous driving solutions, the end-to-end autonomous driving approaches [1] integrate different modules into a whole network, which has the ability to avoid cascade error and information loss between modules. The purpose of end-to-end autonomous driving is to build a completely differentiable neural network that directly maps the input of the raw sensor to the planning trajectory or control signal. Recently, the end-to-end paradigm has attracted attention widely from academia to the industry. However, due to the complexity and difficulty of learning the mapping from the raw sensor input to low-level planning waypoints, the end-to-end autonomous driving methods [2]–[4] often lack explainability and safety guarantee.

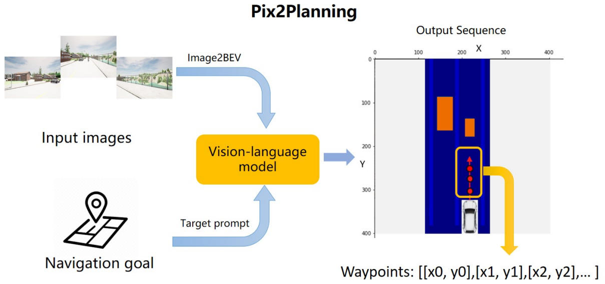


Fig. 1: Illustration of the proposed method Pix2Planning. This method transfers the end-to-end autonomous driving task to a language modeling task with pixel input. In detail, the navigation goal is constructed as language prompt. With the hidden BEV features from encoder, the transformer decoder expresses the planning trajectory as a sequence of discrete toksens. The video demonstration can be found at: `https://youtu.be/qz14aaRmutA`.

Due to the high price of LiDAR, commercial low-cost autonomous driving systems aim to rely on visual solutions, so our research focuses on visual end-to-end autonomous driving methods. Current visual end-to-end autonomous driving methods [5]–[13] usually use an encoder-decoder paradigm to design the whole framework. The encoder usually extracts visual feature from images, and the high-dimensional hidden feature is decoded by MLP-based or GRU-based decoder [8, 9, 14, 15] to obtain the waypoints or control signals for autonomous driving tasks. By adding some visual auxiliary supervision, such as depth, and semantic segmentation, the encoder is interpretable and trained by effective guidance. However, it's hard to explain the principle of mapping high-dimensional feature into discrete waypoints or control signals in MLP-based decoders. In addition, the GRU-based decoder generates waypoints one by one recurrently, which easily suffers from forgetting information and vanishing gradient in the long-range prediction. Decoders are hard to train unless overfitting.

Inspired by the exciting performance of transformer [16] used in natural language processing, we consider using the transformer as a decoder to generate a series of waypoints

Chunjing Xu is with IAS BU, Huawei Technologies, Shanghai, China. Xiangru Mu, Tong Qin, Songan Zhang, Ming Yang are with the Global Institute of Future Technology, Shanghai Jiao Tong University, Shanghai, China. `xuchunjing@huawei.com`, `{muxiangru, qintong, songanz, mingyang}@sjtu.edu.cn`. * is the corresponding author.

as language sequences. Naturally, the planning trajectory is a sequence consisting of multiple waypoints. Every point in the trajectory is like a word in the language sentence. The transformer can extract self-attention inside every waypoint to make the trajectory consistent. Moreover, the transformer can make full use of environmental information through cross-attention to make the planning task reasonable.

Upon these insights, we propose a visual end-to-end autonomous driving approach, which casts the planning to a language generation task. The task conversion leads the network to comprehend correspondences between trajectory and images rather than learn the trajectory directly from the complexity mapping. Specifically, the visual feature is extracted by the CNN-based backbone and projected into the BEV. Using the target navigation point as the task prompt, the language-based decoder makes full use of the BEV information to generate waypoints in an auto-regression way. We have conducted experiments on closed-loop CARLA autonomous driving benchmarks to validate the proposed method and it achieved state-of-the-art performance. The main contributions of this paper include:

- We proposed a novel end-to-end planning method, which casts planning as a language sequence generation problem conditioned on pixel input to learn the correspondence between trajectory sequences and images.
- We designed an auto-regressive language-based transformer decoder for waypoints generation, which has a larger receptive field and makes full use of the bev feature information.
- We conducted experiments on closed-loop autonomous driving benchmarks with CARLA and achieved state-of-the-art performance among visual-only methods. The source code will be open-sourced.

II. Related Work

A. End-to-end Autonomous Driving

The end-to-end autonomous driving solution has been a hot spot over recent years. Most methods learn the driving policy supervised by an experienced agent.

ChauffeurNet [17] is an imitation learning end-to-end method applied to real vehicles. ChauffeurNet established a series of network structures that work together, including FeatureNet, AgentRNN, and PerceptionRNN, to learn effective driving strategies from expert data. CIL [6] and CILRS [7] leveraged conditional imitation learning to directly map front-view image features to control signals. LBC [18] utilized a two-stage framework to improve the precision of imitative learning. Transfuser [14] and Interfuser [19] designed a multi-modal transformer to fuse information from the camera and LiDAR to get a full comprehension of the environment. NEAT [8] presents neural attention fields for joint BEV semantic prediction and trajectory planning in autonomous vehicles. TCP [9] proposed a fusion network that fuses the control signal with the waypoints prediction to benefit from the advantages of both branches. ST-P3 [11] built an interpretable end-to-end vision-based autonomous driving system, which uses a spatial-temporal network for perception, prediction and planning tasks simultaneously. ThinkTwice [20] used a scalable decoder for end-to-end autonomous driving that emphasizes the importance of enlarging the capacity of the decoder. CAT [10] proposed a distilled-based method for teaching a student to drive using supervision from a privileged teacher. ReasonNet [21] built a temporal and global reasoning network to enhance historic scene reasoning and improve global contextual perception performance under occlusion. Transfuser++ [22] identified two biases in current end-to-end methods. The first is lateral recovery guided by target navigation point. And the second bias is slowing down caused by longitudinal averaging of multimodal waypoint predictions. UniAD [13] presented a comprehensive system with a wide span of tasks. It leveraged unified query design across nodes to get state-of-the-art results in nuScenes Dataset.

The above methods have impressive performance on CARLA or nuScenes benchmarks. However, most of these methods use GRU-based decoder to get planning waypoints from high-dimension features recurrently, which suffer from vanishing gradient and forgetting information. Due to the large receptive field and global attention mechanism of transformer [16], we design an auto-regressive transformer decoder to obtain the planning waypoints from BEV features.

B. BEV Representation for Autonomous Driving

The bird-eye-view representation of the driving scene is a powerful tool for planning and control tasks. Compared to camera-view expression, the BEV representation contains 3D scene layout information without occlusion, which is better correlated with vehicle kinematics than 2D images.

State-of-the-art methods that convert camera view to BEV include LSS [23], BEVFormer [24], CVT [25] and others [26]. LSS [23] and BEVDepth [27] firstly learned a depth distribution to lift each pixel in 3D and then use the camera extrinsic and intrinsic to splat all frustums into the BEV grid. BEVDepth [27] added ground-truth supervision for the depth prediction to improve the prediction of depth distribution. BEVFormer [24] leveraged predefined grid-shaped BEV queries to look up spatio-temporal space and aggregate spatio-temporal information, achieving state-of-the-art perception performance. CVT [25] presented cross-view transformers, an efficient attention-based model to learn the mapping between views through a geometry-aware positional embedding from multiple cameras.

These camera-to-BEV algorithms were widely adopted in autonomous driving tasks. Transfuser [14], Transfuser++ [22], Interfuser [19] designed a multi-modal transformer to fuse the images and LiDAR data to generate the bird-eye-view perception. Neat [8] presented neural attention fields to convert image features from perspective view to bird-eye-view. ST-P3 [11] accumulated several past frames and aligned them with the current BEV feature, providing better BEV representation for subsequent tasks. The expert used in LAV [15], Roach [5], LBC [18], CAT [10] and ThinkTwice

[20] depended on privileged BEV information to learn the representation of 3D scenes.

With comprehensive consideration of precision and speed, we adopt LSS with depth supervision in our BEV encoder to generate the 3D representation of the scene.

C. Vision-language Model For Autonomous Driving

Vision-and-language models refer to models that can associate images and text, including image captioning, visual question-answering, visual grounding, and so on. Pix2seq [28] proposed to use a language modeling framework for object detection, which casts the representation of an object as a sequence of tokens.

Recently, the vision-language model has been applied to autonomous driving tasks. Kanishk Jain et al. [29] proposed a vision language navigation method that explicitly grounds the navigable regions corresponding to the textual command in dynamic scenes. LM-Nav [30] built a robotic navigation system that combines three large independently pre-trained models: a self-supervised robotic control model (VNM), a vision-language model that grounds images in text (VLM), and a large language model that can parse and translate text (LLM). ADAPT [31] is an end-to-end transformer-based architecture, which provides natural language narrations and reasoning for each decision-making step of autonomous vehicles.

Inspired by Pix2seq [28], we aim to cast end-to-end planning as a language modeling task, which generates every waypoint as a word in sequence. This formulation bridges the gap between low-level planning and control with high-dimension of image features. Specially, we model the target navigation point as a language prompt to drive the waypoint generation.

III. Methodology

As illustrated in Fig. 2, the proposed framework consists of three main components: (1) A BEV encoder that extracts image features and transforms them to bird-eye-view, Sect.III-B. (2) A waypoint encoder that encodes the target navigation point and sequential waypoints into input sequences for transformer decoder, Sect.III-C. (3) An auto-regressive transformer decoder that receives the BEV feature and encoded sequence to generate the following waypoint, Sect.III-D. Moreover, several auxiliary tasks are added to enhance interpretability and accelerate the convergence of training, Sect.III-E. Finally, a safety controller maps the planning waypoints to control signals, Sect.III-F.

A. Input and Output Representations

Input Representations: There are three inputs for our framework, which are multi-view (front, right, left) images, a target navigation point. The target navigation point is a noisy goal point, $50m$ to $100m$ ahead of the vehicle, indicating the destination of the vehicle. The target navigation point serves as the prompt of the vision-language model. The waypoints used for supervision come from an experienced agent in the training time. In the Carla simulator, we use Autopilot as the experienced agent.

Output Representations: The network predicts the trajectory of the ego vehicle in the BEV space, which consists of several waypoints, $P_t = (x_t, y_t),\ t \in (0, T)$. The time interval of every waypoint is 0.5 second. T is 4 in current settings. Then the future waypoints are put into a safety controller to obtain control signals by a kinematic model. Besides, the auxiliary tasks are able to predict roads, vehicles, pedestrians, traffic light state, and depth.

B. BEV Encoder

Since it is convenient and stable to plan a trajectory in the BEV space, we transform the information from multiple cameras to a bird's-eye representation.

Firstly, we use an image backbone EfficientNet [32] on each camera-view image to obtain its feature map $f \in \mathbb{R}^{C\times H\times W}$. Then the image-to-bev method LSS [23] is adopted: We first learn a depth distribution $d \in \mathbb{R}^{D\times H\times W}$ of image features and lift each pixel in 3D space, C is the number of feature channels, D denotes the number of discrete depths. The predicted depth distribution d and image feature map f is multiplied to get the image feature with depth. Then we use camera extrinsic and intrinsics to splat all frustums into a rasterized BEV grid to generate BEV feature $f_{bev} \in \mathbb{R}^{C\times H_{bev}\times W_{bev}}$. Inspired by BEVDepth [27], we also add depth supervision for the depth prediction to improve the performance of the BEV encoder.

C. Waypoint Encoder

As mentioned before, we cast the planning task as a language modeling task conditioned on pixel inputs. Like the quantization scheme used in Pix2seq [28], we need to convert the target waypoint into a sequence of discrete coordinate tokens. The sequences served as input for language modeling tasks contain two parts. The first part is the target point, and the second part is the sequential waypoints from time 0 to $t-1$. The target navigation point, $[x^w_{tg}, y^w_{tg}]$, is a coarse point in world coordinate supplied by a global planner indicating the destination of the vehicle, playing an important role in guiding the planning tasks. The target point is transformed from the world coordinate to the vehicle coordinate by,

$$\begin{bmatrix} x^{veh} \\ y^{veh} \\ 1 \end{bmatrix} =^{veh} \mathbf{T}_{world} \cdot \begin{bmatrix} x^{w} \\ y^{w} \\ 1 \end{bmatrix} \tag{1}$$

All the waypoints are under the vehicle frame. Therefore, the sequence input is constructed to be:

$$[x^{veh}_{tg}, x^{veh}_{tg}, x^{veh}_0, y^{veh}_0, ..., x^{veh}_{t-1}, y^{veh}_{t-1}], \tag{2}$$

Next, the BEV space is rasterized into a N_{bin}xN_{bin} grid, with the length l in the world coordinate. All points in vehicle coordinate can be transformed into rasterized grid by the following quantization formula:

$$Q_x(x) = round(\frac{x}{l}) \cdot N_{bin}, \tag{3}$$

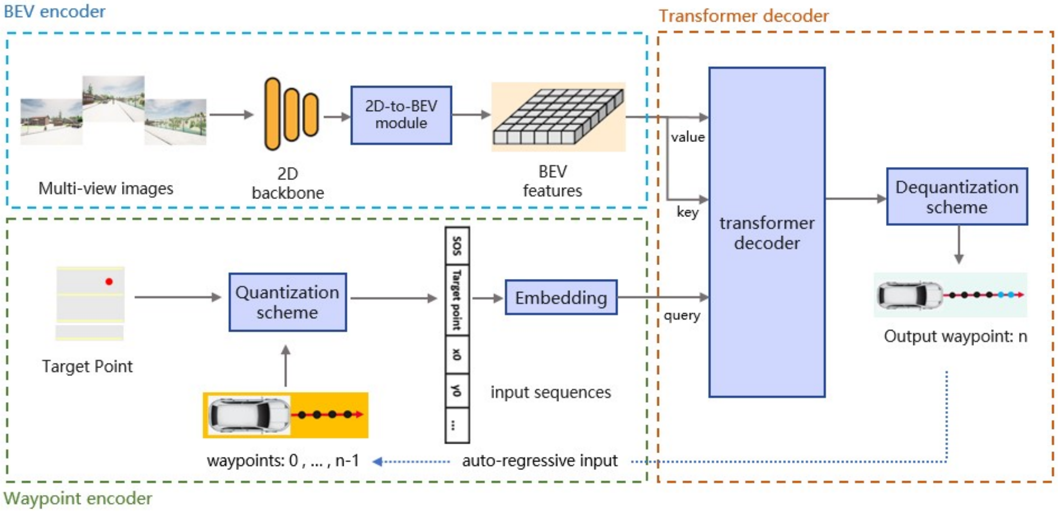


Fig. 2: Overview of the architecture of Pix2Planning. Firstly, raw images from multi-view cameras are processed by a 2D backbone. Then the image features are transformed into the bird's-eye-view (BEV) representation. We encode the target navigation point and previous waypoints as a text sequence. The text sequence and BEV features are sent to the self-attention and cross-attention modules in transformer. Finally, the auto-regressive decoder predicts next waypoints one by one. Planning waypoints are put in the controller to generate safe control signals. In addition, we design some auxiliary tasks on the image and BEV features to enhance interpretability and accelerate the convergence of training.

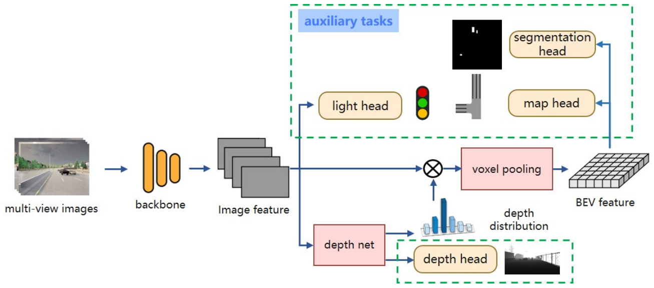


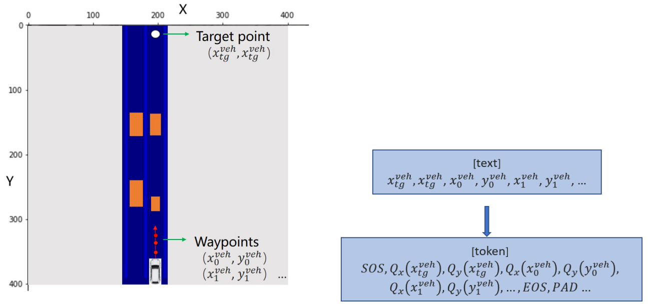
Fig. 3: Illustration of the BEV encoder architecture. This module transforms the camera-view images to bird-eye-view features.

$$Q_y(y) = N_{bin} - round(\frac{y}{l}) \cdot N_{bin}, \tag{4}$$

After quantization, the sequence is formed in the following representation:

$$[SOS, Q_x(x^{veh}_{tg}), Q_y(y^{veh}_{tg}), Q_x(x^{veh}_0), Q_y(y^{veh}_0), \\ ..., EOS, PAD, ...] \tag{5}$$

where SOS serves as a start flag to indicate the beginning of the sequence and the EOS is the end flag. PAD is the empty token to guarantee the same length of different sequences. An illustration of the quantization procedure is shown in Fig. 4. The effect of different levels of quantization on the waypoints is further discussed in the Ablation Study, IV-D.


X
0
100
200
300
400
Y
Target point
$(x^{veh}_{tg}, x^{veh}_{tg})$
Waypoints
(x^{veh}_0, y^{veh}_0)
(x^{veh}_1, y^{veh}_1) ...
[text]
$x^{veh}_{tg}, x^{veh}_{tg}, x^{veh}_0, y^{veh}_0, x^{veh}_1, y^{veh}_1, ...$
[token]
$SOS, Q_x(x^{veh}_{tg}), Q_y(x^{veh}_{tg}), Q_x(x^{veh}_0), Q_y(y^{veh}_0), Q_x(x^{veh}_1), Q_y(y^{veh}_1), ..., EOS, PAD ...$

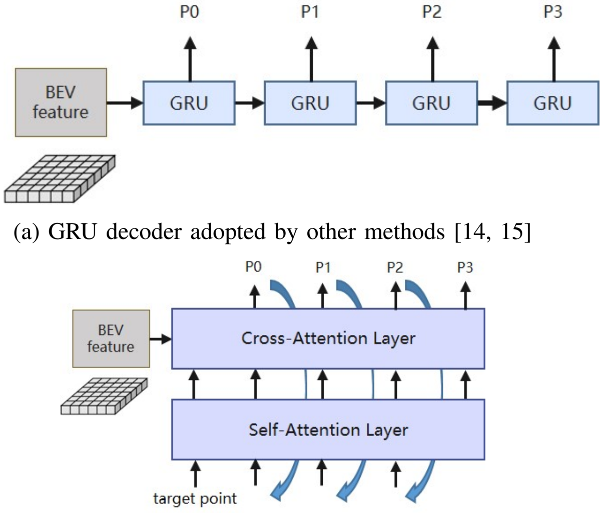
(a) (b)

Fig. 4: The input and output representations of quantization scheme in the waypoint encoder.

D. Auto-regressive Transformer Decoder

Current end-to-end autonomous driving approaches [14, 15] often adopt global pooling and MLP (Multilayer Perceptron) to convert image features into a high-dimension vector. Then, the high-dimension vector is sent to the GRU decoder to generate waypoints, as shown in Fig.5a. Due to global pooling and MLP operation, the spatial information in the BEV features is not reserved. In contrast, we propose to use the transformer as the decoder, which reserves the spatial information by position embedding and has a global receptive field.

Upon these insights, we design an auto-regressive transformer decoder (shown in Fig.5b) to improve the performance of the decoder. Inputs are the BEV feature and the target point. Since the vision feature and language sequences are aligned, we use the target point serves as a prompt to boost the planning task. Through self-attention and cross-attention with the BEV feature, the transformer generates waypoints one by one in an auto-regressive way. The detailed design of the transformer is shown in Fig 6. The text sequence is first put into the embedding layer. Then the embedded vector serves as the key, value, and query in the self-attention layer. In the cross-attention, the BEV feature serves as the value and key, while the sequence vector serves as the query. After each iteration, the transformer outputs the next waypoint. The next waypoint will be added to the input sequence to predict the following one. This process will



(a) GRU decoder adopted by other methods [14, 15]

(b) Transformer decoder proposed by us

Fig. 5: The comparison of architecture about GRU decoder and the auto-regressive transformer decoder.

repeat N times. When the auto-regressive decoder meets the EOS token, the regression progress will stop. In this way, the decoder generates waypoints in an auto-regressive way.

E. Tasks and Losses

Due to the sparse supervision of waypoint prediction, the network suffers from training divergence and weak generalization ability. Therefore, it is important to add auxiliary tasks to enhance interpretability and accelerate convergence. The designed loss function for every task in our framework is shown as follows:

Trajectory Loss: The trajectory T_i $(i \in 0, ..., n)$ from an experienced agent is treated as the ground truth to supervise the network's output. The sequence of the target trajectory is first constructed to the text sequences. The cross entropy is used to compute the loss between predicted sequences and target sequences:

$$f_{ce}(\hat{y}, y) = -\sum_{i=N}^{i=1} y_i log(\hat{y}_i) \tag{6}$$

$$\mathcal{L}_{traj} = f_{ce}(\hat{T}_i, T_i) \tag{7}$$

Auxiliary tasks and Losses: As is shown in Fig. 3, we design auxiliary tasks on image and BEV features.

We propose a depth prediction task and a light state prediction task concerning the image feature. The depth prediction task is designed to guarantee a better depth estimation so that the image feature can be lifted correctly to construct the BEV grid. In addition, the traffic light state (red, yellow, and green) has a significant influence on safe passing at the intersection, so we explicitly learn it from the front-view image feature. Both depth prediction and traffic light are treated as classification tasks. Therefore, the loss functions of these tasks are:

$$\mathcal{L}_{dep} = f_{ce}(\hat{F}_{dep}, F_{dep}) \tag{8}$$

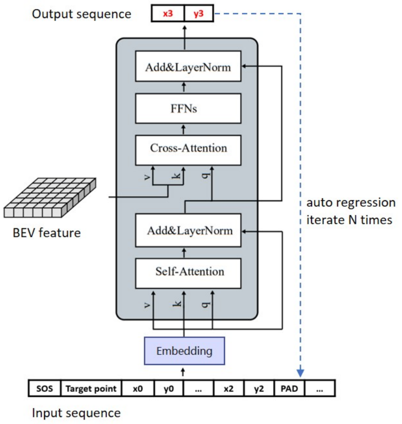


Fig. 6: Architecture of the auto-regressive transformer decoder of Pix2Planning.

$$\mathcal{L}_{light} = f_{ce}(\hat{F}_{light}, F_{light}). \tag{9}$$

Besides the auxiliary tasks on the image feature, we perform some auxiliary tasks on the BEV feature. The stable road structure (lane divider and drivable area) and dynamic traffic participants (vehicle, pedestrian, motorcycle, ...) are segmented separately. The loss functions of stable road structure and dynamic traffic participants are also cross entropy losses, which are widely used in the segmentation task:

$$\mathcal{L}_{road} = f_{ce}(\hat{F}_{road}, F_{road}) \tag{10}$$

$$\mathcal{L}_{obj} = f_{ce}(\hat{F}_{obj}, F_{obj}) \tag{11}$$

The overall loss is as follows, as weighted by λ_{traj}, λ_{dep}, λ_{light}, λ_{road}, and λ_{obj}, :

$$\begin{aligned}\mathcal{L} = \lambda_{traj} \cdot \mathcal{L}_{traj} + \lambda_{road} \cdot \mathcal{L}_{road} + \lambda_{obj} \cdot \mathcal{L}_{obj} + \\ \lambda_{dep} \cdot \mathcal{L}_{dep} + \lambda_{light} \cdot \mathcal{L}_{light}\end{aligned} \tag{12}$$

F. Safety Controller

After the neural network generates waypoints, the waypoints are converted to control signals which include steer, throttle, and brake. We propose a PID-based safety controller to get the control signals. Using the kinematics model, we compute the expected angle and velocity of the ego-vehicle from subsequent waypoints. Then, the PID controller adjusts the control signals to reach the expected angle and velocity. Although the waypoints from the neural network have the ability to obey the traffic rules and avoid collisions, we perform some hand-craft strategy to further enhance safety. An explicit collision-avoiding strategy is added, by leveraging BEV segmentation results to avoid the ego vehicle hitting others. In addition, the traffic light detection result is put into the controller to ensure that the ego vehicle will stop at the red light.

IV. Experiments

We used closed-loop simulator CARLA [33] to validate the end-to-end autonomous driving ability of the proposed method.

TABLE I: Performance on Longset6 Benchmark

Methods	RGB	LiDAR	DS ↑	RC ↑	IS ↑	Ped ↓	Veh ↓	LC ↓	Red ↓	Dev ↓	TO ↓	Blk ↓
NEAT [8]	✓	✗	24±3	60±1	0.49±0.02	0.01	0.71	0.21	0.86	0.00	0.02	2.83
TCP [9]	✓	✗	54±2	78±2	0.69±0.03	0.01	0.65	0.20	0.19	0.02	0.03	0.30
CAT [10]	✓	✗	58±2	78±2	**0.77±0.02**	**0.00**	**0.20**	0.02	**0.05**	0.00	0.04	0.44
Pix2Planning	✓	✗	**59±2**	91±3	0.66±0.03	0.082	0.63	**0.00**	0.242	0.026	**0.00**	0.131
Transfuser [14]	✓	✓	47±6	**93±1**	0.50±0.06	0.03	2.45	0.07	0.16	**0.00**	0.06	0.10
Interfuser [19]	✓	✓	47±6	74±1	0.63±0.07	0.06	1.14	0.11	0.24	0.00	0.52	**0.006**

*The ↑ means the value of this metric is the higher the better, while ↓ means the value of this metric is the lower the better.

A. Data Collection

For closed-loop training and evaluations, we took CARLA Autopilot as the privileged expert agent, from which our model learns the driving policies. The CARLA autopilot takes the perfect perception of the road structure and traffic participants to generate safe and efficient routes. We took routes in 8 towns with 21 weather and collected about 180K frames with 2 FPS for the training. In the data collection process, we collected three camera views (front, left, and right), corresponding depth images, ego-vehicle state, BEV information, and target navigation point provided by the global planner. In addition, the waypoints for supervision were sampled from the expert trajectory by a time interval of 0.5s.

B. Training Details

For input representations, the resolution of the input image is 600×800. Three images from different views are fed into the network at once. The network output is four subsequent waypoints $[[x_0, y_0], ..., [x_3, y_3]]$ in the BEV, where x belongs to $[-25.0m, 25.0m]$, and y belongs to $[0.0m, 50.0m]$. The length and width of BEV gird is 50m with the ego vehicle located in the bottom-center. In the waypoint encoder, the resolution of BEV grid is 0.1m, and N_{bin} is 500. The RGB image is encoded by EfficientNet-B4 [32] which is pre-trained on ImageNet [34].

C. Evaluation Benchmarks and Metrics

Evaluation Benchmarks: We selected the Town05 and Longest6 scenarios as benchmarks to evaluate our and other methods. The Town05 benchmark consists of 32 short routes and 10 long routes on different scenarios which are spawned at predefined positions. Besides, we also added Longest6 benchmark evaluations, which have dense traffic, diverse towns, and routes from Town01 to Town06. In these benchmarks, the ego vehicle needs to follow the predefined routes to drive under different weathers without collision, blocking for a long time, and violating traffic rules.

Evaluation Metrics: We consider three common metrics introduced by the CARLA Leaderboard: route completion score (RC), infraction score (IS), and driving score (DS). The route completion score (RC) is the percentage of the route distance completed by the agent before it deviates from the route or gets blocked. The infraction score (IS) is a cumulative penalty for every infraction such as collision or traffic rules violation. The driving score (DS) is computed by the RC and the IS, the higher score means the routes are completed more safely. Additional infraction metrics (pedestrian collisions (Ped), vehicle collisions (Veh), layout collisions (LC), red light violations (Red), route deviation (Dev), route timeouts (TO), agent blocked (Blk)) are also adopted.

TABLE II: Performance on Town05 Benchmark

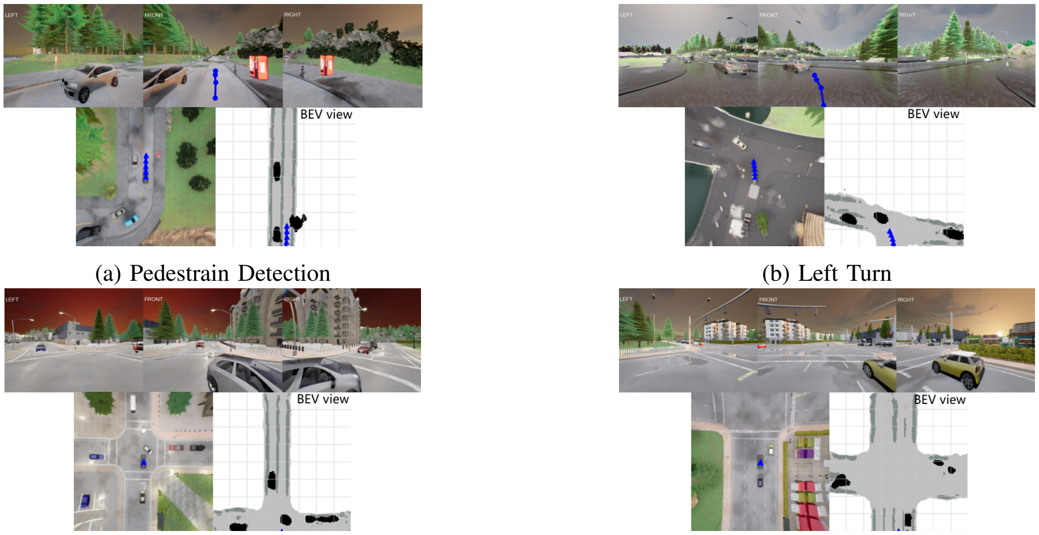
Methods	SENSOR		Town05 Short		Town05 Long	
	RGB	LiDAR	DS ↑	RC ↑	DS ↑	RC ↑
ST-P3 [11]	✓	✗	55.14	86.74	11.45	83.15
NEAT [8]	✓	✗	58.70	77.32	37.72	62.13
TCP [9]	✓	✗	75.74	88.44	57.2	80.4
Pix2Planning	✓	✗	83.26	**99.81**	63.59	**98.06**
Transfuser [14]	✓	✓	54.52	78.41	33.15	56.36
Interfuser [19]	✓	✓	**94.95**	95.19	**68.31**	94.97

The results are shown in Table I and Table II. We compared our method with other state-of-the-art methods. As is shown in Table I, our visual method got the best DS and RC among all visual methods in Longset6 benchmark, even better than the multi-modality fusion method Transfuser [14] and Interfuser [19] in DS. In addition, in the Town05 benchmark, shown in Table II, we achieved state-of-the-art performance compared to other visual methods. And our performance was only slightly lower than the sensor fusion method Interfuser [19] in DS.

D. Ablation Study

We also conducted experiments to analyze the influence of different kinds of decoders, auxiliary tasks design, and the resolution of the waypoint encoder.

Decoder Design: In Table III, we compared the performance between different kinds of decoders. The GRU decoder, shown in Fig. 5a, has the lowest DS and RC among the three methods. ConvGRU decoder adopted by ThinkTwice [20] has raised 4.6 points improvement in DS and 41 points in RC. The reason for the improvement is that the coarse-to-fine strategy could use the future BEV feature to predict the waypoint offset. Our method, a transformer-based decoder, gained the highest DS and RC scores, about 40 points higher in DS than ConvGRU. Due to the cross-attention architecture in the transformer, our method can



(a) Pedestrain Detection

(b) Left Turn

(c) Vehicle Avoidance

(d) Stop at Red Lights

Fig. 7: Visualization of model input and output representation in different scenarios. The top row of every subfigure shows the input RGB images in CARLA. In addition, the topdown-view and the output bird-eye-view of model are shown in the bottom row of each figure. The blue arrows show the planning waypoints predicted by our framework.

TABLE III: Ablation Study on Decoder

Methods	DS ↑	RC ↑
GRU decoder [14]	46.24	57.92
ConvGRU decoder [20]	52.80	98.50
Our auto-regressive transformer decoder	**92.39**	**99.60**

TABLE IV: Ablation Study on Auxiliary Tasks

Methods	DS ↑	RC ↑
baseline	**92.39**	**99.60**
w/o depth supervision	81.14	89.71
w/o light state prediction	63.75	95.24
w/o BEV supervision	44.87	66.86

make full use of the BEV feature to have global attention on the environment.

Auxiliary Tasks Design: We also discussed the necessity of various auxiliary tasks in Table IV. As mentioned before, depth supervision improved the quality of the BEV feature. Therefore, without depth supervision, the DS decreased by about 11 points in DS and 10 points in RC. Light state detection also played an important role in autonomous driving systems. We used the front camera to detect the light state, and the second line in the table shows that the DS decreased by about 30 points due to the insensibility of the traffic light. The most important auxiliary task was the BEV supervision task, which resulted in drops of 49 points and 33 points on the DS and the RC without this task.

The Number of Quantization Bins in Waypoint Encoder: The number of quantization bins affect the quantization accuracy of the waypoint encoder. We tested different resolutions, 200, 500, and 1000.

As shown in Table V, resolutions 500 and 1000 had similar performance in DS and RC. However, if the resolution was set to 200, the DS and RC would drop about 30 points due to the large quantization error. In addition, when the navigation waypoints are directly put in embedding layer without quantization, the DS and RC is dropped due to the misalignment between sequences and BEV feature. The results show that resolution of 500 achieved a good balance of the precision and efficiency.

TABLE V: Ablation Study on Resolution of Waypoint Encoder

Methods	DS ↑	RC ↑
w/o quantization	35.29	55.48
200	66.74	68.79
500 (baseline)	92.39	**99.60**
1000	**92.45**	98.70

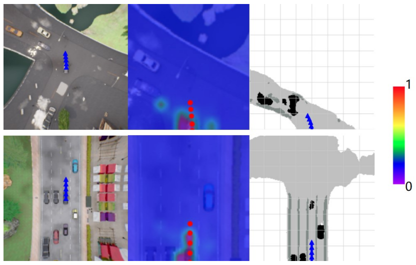


Fig. 8: Visualization of attention maps of models in two CARLA scenarios. In each image, from left to right is a topdown view in CARLA, attention maps, and the output of BEV auxiliary tasks.

E. Visualization

The visualization of the evaluations on the CARLA benchmark was shown in Fig. 7. In Fig. 7, the ego vehicle could deal with different kinds of scenarios using the planning waypoints. In addition, we visualized the attention map from the auto-regressive transformer, which is shown in Fig. 8. The attention map indicated the interested regions where the network focused on in the planning task.

V. Conclusion

In this paper, we proposed Pix2Planning, a vision-language modeling framework for end-to-end autonomous driving, which casts planning tasks as the language sequence generation tasks. With the prompt from the waypoint encoder, we leveraged an auto-regressive transformer decoder to make full use of the BEV feature from the BEV encoder. Our method has achieved state-of-the-art visual performance on CARLA benchmarks through designed evaluation experiments. In the future, we will further extend our method to predict the control signals directly from the network and conduct real world experiments.

References

[1] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, "End-to-end autonomous driving: Challenges and frontiers," 2023.

[2] Z. Li, T. Motoyoshi, K. Sasaki, T. Ogata, and S. Sugano, "Rethinking self-driving: Multi-task knowledge for better generalization and accident explanation ability," 2018.

[3] A. Filos, P. Tigkas, R. Mcallister, N. Rhinehart, S. Levine, and Y. Gal, "Can autonomous vehicles identify, recover from, and adapt to distribution shifts?" in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 13–18 Jul 2020, pp. 3145–3153.

[4] M. R. Samsami, M. Bahari, S. Salehkaleybar, and A. Alahi, "Causal imitative model for autonomous driving," 2021.

[5] Z. Huang, J. Zhang, R. Tian, and Y. Zhang, "End-to-end autonomous driving decision based on deep reinforcement learning," in *2019 5th International Conference on Control, Automation and Robotics (ICCAR)*, 2019, pp. 658–662.

[6] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 4693–4700.

[7] F. Codevilla, E. Santana, A. M. Lopez, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[8] K. Chitta, A. Prakash, and A. Geiger, "Neat: Neural attention fields for end-to-end autonomous driving," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 15 773–15 783.

[9] P. Wu, X. Jia, L. Chen, J. Yan, H. Li, and Y. Qiao, "Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline," in *Advances in Neural Information Processing Systems*, vol. 35. Curran Associates, Inc., 2022, pp. 6119–6132.

[10] J. Zhang, Z. Huang, and E. Ohn-Bar, "Coaching a teachable student," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 7805–7815.

[11] S. Hu, L. Chen, P. Wu, H. Li, J. Yan, and D. Tao, "St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning," *ArXiv*, vol. abs/2207.07601, 2022.

[12] A. Hu, G. Corrado, N. Griffiths, Z. Murez, C. Gurau, H. Yeo, A. Kendall, R. Cipolla, and J. Shotton, "Model-based imitation learning for urban driving," in *Advances in Neural Information Processing Systems*, vol. 35. Curran Associates, Inc., 2022, pp. 20 703–20 716. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/827cb489449ea216e4a257c47e407d18-Paper-Conference.pdf

[13] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, L. Lu, X. Jia, Q. Liu, J. Dai, Y. Qiao, and H. Li, "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[14] A. Prakash, K. Chitta, and A. Geiger, "Multi-modal fusion transformer for end-to-end autonomous driving," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 7073–7083.

[15] D. Chen and P. Krähenbühl, "Learning from all vehicles," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 17 201–17 210.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[17] M. Bansal, A. Krizhevsky, and A. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," 2018.

[18] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 100. PMLR, 30 Oct–01 Nov 2020, pp. 66–75.

[19] H. Shao, L. Wang, R. Chen, H. Li, and Y. Liu, "Safety-enhanced autonomous driving using interpretable sensor fusion transformer," in *Proceedings of The 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 205. PMLR, 14–18 Dec 2023, pp. 726–737.

[20] X. Jia, P. Wu, L. Chen, J. Xie, C. He, J. Yan, and H. Li, "Think twice before driving: Towards scalable decoders for end-to-end autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 21 983–21 994.

[21] H. Shao, L. Wang, R. Chen, S. L. Waslander, H. Li, and Y. Liu, "Reasonnet: End-to-end driving with temporal and global reasoning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 13 723–13 733.

[22] B. Jaeger, K. Chitta, and A. Geiger, "Hidden Biases of End-to-End Driving Models," *arXiv e-prints*, p. arXiv:2306.07957, Jun. 2023.

[23] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *Computer Vision – ECCV 2020*. Cham: Springer International Publishing, 2020, pp. 194–210.

[24] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *Computer Vision – ECCV 2022*. Cham: Springer Nature Switzerland, 2022, pp. 1–18.

[25] B. Zhou and P. Krähenbühl, "Cross-view transformers for real-time map-view semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 13 760–13 769.

[26] Y. Liu, T. Wang, X. Zhang, and J. Sun, "Petr: Position embedding transformation for multi-view 3d object detection," in *Computer Vision – ECCV 2022*. Cham: Springer Nature Switzerland, 2022, pp. 531–548.

[27] Y. Li, Z. Ge, G. Yu, J. Yang, Z. Wang, Y. Shi, J. Sun, and Z. Li, "Bevdepth: Acquisition of reliable depth for multi-view 3d object detection," 2022.

[28] T. Chen, S. Saxena, L. Li, D. J. Fleet, and G. E. Hinton, "Pix2seq: A language modeling framework for object detection," *ArXiv*, vol. abs/2109.10852, 2021.

[29] K. Jain, V. Chhangani, A. Tiwari, K. M. Krishna, and V. Gandhi, "Ground then navigate: Language-guided navigation in dynamic scenes," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 4113–4120.

[30] D. Shah, B. Osiński, b. ichter, and S. Levine, "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *Proceedings of The 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 205. PMLR, 14–18 Dec 2023, pp. 492–504. [Online]. Available: https://proceedings.mlr.press/v205/shah23b.html

[31] B. Jin, X. Liu, Y. Zheng, P. Li, H. Zhao, T. Zhang, Y. Zheng, G. Zhou, and J. Liu, "Adapt: Action-aware driving caption transformer," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7554–7561, 2023.

[32] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 09–15 Jun 2019, pp. 6105–6114.

[33] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 78. PMLR, 13–15 Nov 2017, pp. 1–16. [Online]. Available: https://proceedings.mlr.press/v78/dosovitskiy17a.html

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.