

ReasonNet: End-to-End Driving with Temporal and Global Reasoning

Hao Shao¹ Letian Wang² Ruobing Chen¹
Steven L. Waslander² Hongsheng Li³ Yu Liu^{1,4*}

¹SenseTime Research ²University of Toronto ³CUHK MMLab
⁴Shanghai Artificial Intelligence Laboratory

Abstract

The large-scale deployment of autonomous vehicles is yet to come, and one of the major remaining challenges lies in urban dense traffic scenarios. In such cases, it remains challenging to predict the future evolution of the scene and future behaviors of objects, and to deal with rare adverse events such as the sudden appearance of occluded objects. In this paper, we present ReasonNet, a novel end-to-end driving framework that extensively exploits both temporal and global information of the driving scene. By reasoning on the temporal behavior of objects, our method can effectively process the interactions and relationships among features in different frames. Reasoning about the global information of the scene can also improve overall perception performance and benefit the detection of adverse events, especially the anticipation of potential danger from occluded objects. For comprehensive evaluation on occlusion events, we also release publicly a driving simulation benchmark DriveOcclusionSim consisting of diverse occlusion events. We conduct extensive experiments on multiple CARLA benchmarks, where our model outperforms all prior methods, ranking first on the sensor track of the public CARLA Leaderboard [53].

1. Introduction

Despite significant recent progress in the field of autonomous driving, truly large-scale deployment of autonomous vehicles (AVs) on public roads has yet to be established. The majority of the remaining issues lie in navigating dense urban traffic scenes, where a large number of different dynamic objects (e.g. vehicles, bicycles, pedestrians), complex road geometries and road user interactions are involved. In such circumstances, currently deployed or tested solutions could make incorrect or unexpected decisions, resulting in severe accidents or traffic infractions [4, 24, 53]. Two of the major challenges behind

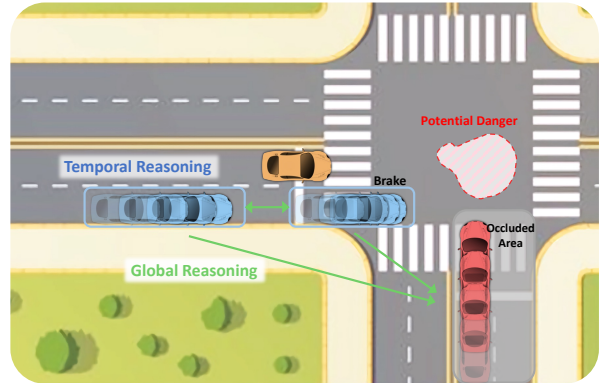


Figure 1. Temporal reasoning on the historic behaviors of surrounding objects can benefit the prediction of the scene evolution and objects’ future behaviors. Global reasoning on the interaction among objects and the environment allows for inference about unobservable space and occluded objects, anticipating potential danger and enhancing perception/driving performance.

such autonomous incompetence include 1) how to achieve a comprehensive understanding of the driving scene and, more importantly, how to make high-fidelity predictions on the future evolution of the driving scene; 2) how to deal with rare adverse events in long-tail distributions, such as undetected but relevant objects in occluded regions.

Comprehensive scene understanding and high-fidelity prediction of how objects in the scene will move in the future are vital for autonomous vehicles to take safe and reliable actions. Toward this end, modularized methods were proposed to decompose the task into three sequential sub-tasks: detection [37–40, 42], tracking [5, 65], and forecasting [6, 26, 28, 32, 33, 59–61]. While more interpretability is provided by developing each module independently, these sub-tasks are still regarded as open research questions and errors in each sub-task can propagate and accumulate, leading to unstable overall performance. In contrast, end-to-end driving methods [9, 10, 51] have recently emerged as a promising method to solve these subtasks in a monolithic manner directly. However, a high-fidelity future prediction necessitates sufficient **temporal reasoning** over the historic

*Corresponding author

information of the scene [22, 50], which is usually only somewhat considered in previous end-to-end driving methods if not completely ignored. For example, [10, 15] only exploited scene information in the current frame, and [54] simply concatenated features in historic frames for temporal reasoning. In such cases, the interactions and relationships amongst features in different frames and objects cannot be sufficiently modeled. Thus in this paper, we propose a temporal reasoning module to effectively fuse information from different frames for better driving performance.

On the other hand, rare adverse events in long-tail distributions remain a notoriously challenging issue on the way toward large scale deployment of autonomous vehicles. For example, one such challenge is the difficulty in detecting occluded but relevant objects in the scene. While a large amount of research has focused on improving perception performance [37, 57], the occluded objects essentially lie out of the scope of observable elements, and failure to consider such objects can result in either dangerous or overly cautious driving behavior. Our observation is that, while humans also suffer from similar limitations to autonomous vehicles regarding occluded objects, they are able to reason about these unobservable spaces by exploiting global information of the scene such as road geometry and driving interaction patterns, to anticipate potential danger even under occlusion. For example, when one human driver notices another vehicle braking abruptly, the driver may reason the presence of an occluded object (e.g., a pedestrian) ahead, reminding himself to drive cautiously. Thus, our insight is that, a safe and intelligent autonomous vehicle should also master the **global reasoning** capability to have a better perception of the scene. In this paper, we propose a transformer-based global reasoning module to sufficiently fuse information of the environment and objects, and analyze their interactions for better scene understanding. Such global reasoning capability not only benefits interaction modeling with occluded objects, but also improves overall perception performance. Examples of such performance gains include better traffic light status identification by reasoning over other vehicles' actions and more accurate future trajectory forecasting by reasoning over interactions among objects. Besides, considering the fact that the occlusion events lie in the long-tail distribution and have been rare in currently available datasets, we also construct a Driving in Occlusion Simulation benchmark (DOS) consisting of 4 occlusion scenarios, each with 25 cases, as a comprehensive occlusion event evaluation benchmark in the field of end-to-end autonomous driving.

In this paper, we propose a novel end-to-end driving framework named temporal and global reasoning network (ReasonNet), which provides enhanced reasoning on the temporal evolution and the global information of the scene, for better perception performance and driving quality. Our

contributions are three-fold:

- We propose a novel temporal and global reasoning Network (ReasonNet) to enhance historic scene reasoning for high-fidelity prediction of the scene's future evolution and improve global contextual perception performance even under occlusion.
- We present a new benchmark called **Driving in Occlusion Simulation** benchmark (DOS), which consists of diverse occlusion scenarios in urban driving for systematic evaluation in occlusion events, and make the benchmark publicly available.
- We experimentally validate our method on multiple benchmarks with complex and adversarial urban scenarios. Our model ranks first on the sensor track of the CARLA autonomous driving leaderboard.

2. Related work

End-to-end Autonomous Driving End-to-end autonomous driving in urban scenarios has become more studied recently thanks to the CARLA simulator and leaderboard [21]. Recent works mainly consist of reinforcement learning (RL) and imitation learning (IL) methods. The reinforcement Learning methods train the agents by constantly interacting with simulated environments and learning from these experiences. Latent DRL [54] first trains an embedding space as a latent representation of the environment observation, and then conducts reinforcement learning with the latent observation. Roach [66] utilizes an RL agent with privileged information of the environment to distill a model only with regular information (e.g. sensor) as the final agent. WOR [9] builds a model-based RL agent along with the world model and reward model. The final agent is distilled from the expert knowledge acquired from these pretrained models. Imitation learning methods aim at learning from an expert agent to bypass interacting with the environment. Early IL methods include CIL [17] and CILRS [18], which apply a conditional architecture with different network branches for different navigation commands. LBC [11] first trains an imitation learning agent with privileged information, which is then distilled into a model using sensor data. Transfuser [15, 47] designs a multi-modal transformer to fuse information between the front camera image and LiDAR data. LAV [10] exploits data of not only the ego vehicle but also surrounding vehicles for data augmentation by learning a viewpoint-invariant spatial intermediate representation. TCP [63] proposes a network with two branches which generates the control signal and waypoints respectively. An adaptive ensemble is applied to fuse the two output signals. InterFuser [51] uses a transformer to fuse and process multimodal multi-view sensors for comprehensive scene understanding.

Attention for Autonomous Driving The attention mechanism has been demonstrated to be a powerful module in many areas of deep learning, including the context of driving. The classic attention-based Transformer architecture [55] was originally established in Natural Language Processing. Transformer (ViT) was then applied in Computer Vision (vision Transformer, ViT [20, 49]) and attains excellent performance on Imagenet classification. Later generations move on to generalize the attention mechanism to the driving domain, including motion forecasting [23, 36, 58], driver attention prediction [25, 34] and object tracking [45, 52]. In the field of end-to-end autonomous driving, TransFuser [15, 47] exploits several transformer modules for the fusion of data from the front view camera and LiDAR. NEAT [14] uses intermediate attention maps to iteratively compress 2D image features into a compact bird-eye-view (BEV) representation for driving. InterFuser [51] utilizes a transformer encoder and decoder to fuse information and decode the feature into interpretable embeddings.

Multi-task Learning Our end-to-end driving framework adopts multi-task learning, with a joint objective of object detection, occupancy forecasting, traffic sign prediction and waypoint prediction. MotionNet [62] proposes a spatio-temporal pyramid network for joint perception and motion prediction based on BEV maps. PnPNet [41] proposes a new object trajectory representation and multi-object tracker to handle occlusion and false positives. IntentNet [7] predicts the high-level intentions of each agent from semantic HD maps building. ST-P3 [29] proposes an egocentric-aligned accumulation technique to preserve geometry information in 3D space and utilize a dual pathway modeling to consider past motion variations.

3. Method

We aim at learning a driving policy π that generates raw control commands by taking multi-view multi-modal sensor readings, vehicle measurements, and navigation commands as inputs. As shown in Figure 2, the proposed ReasonNet consists of three parts: 1) a perception module that extracts bird’s-eye-view (BEV) features from LiDAR and RGB data; 2) a temporal reasoning module that processes temporal information and maintains a memory bank storing historic features; 3) a global reasoning module that captures the interaction/relationship amongst objects and the environment, to detect adverse events (e.g. occlusion) and improve overall perception performance. This section will introduce these modules in detail.

3.1. Perception Module

The perception module is responsible for processing and fusing different sensor data at the early stage of our framework, based on which temporal and global reasoning can be conducted by later modules. Specifically, five sensors

are utilized: four RGB cameras (left, front, right and rear) $I_{rgb} = I_{0,1,2,3}$ and one LiDAR sensor $I_{lidar} = I_4$. Four image inputs are obtained from the four cameras, and an additional focus-view image input is center-cropped from the front image to capture distant traffic lights. Point cloud data is retrieved from the LiDAR sensor. Our perception module includes a 2D backbone to embed image input into keys and values, a 3D backbone to embed LiDAR input into queries, and a BEV decoder that utilizes these keys, values, and queries to obtain features of the bird’s-eye view (BEV) map, waypoints, and traffic signs.

Image Input For each image input of I_{rgb} , a 2D CNN backbone ResNet [27] is applied to generate a feature map f_i . Then, we use a convolution layer to map the channels of f_i to C_v and flatten it to one-dimensional tokens. A sinusoidal positional encoding and learnable sensor embeddings are added to the tokens, so that the following network can distinguish them from different cameras and relative positions. Finally, tokens of different images are passed through a standard transformer encoder with K_e layers. Each layer consists of Multi-Headed Self-Attention [55], MLP blocks and layer normalization [3]. This image fusion operation can contribute to a better perception of global context from multi-view inputs, generating keys and values for the image-LiDAR fusion in BEV decoder.

LiDAR Input For the LiDAR input, we use PointPillars [35] as our 3D perception backbone to process points in the ego-vehicle-centered area $x \in [-H_b, H - H_b]$ and $y \in [-W/2, W/2]$. Specifically, we use a simplified version of PointNet [48] to encode the information of raw LiDAR points. Each pillar includes the points in a $0.25m \times 0.25m$ area. The extracted feature map is down-sampled to $C_v \times H \times W$ for computation reduction and then serves as BEV queries used in the BEV decoder and memory bank.

Sensor-Fusion BEV Decoder The BEV decoder follows a standard transformer architecture design with K_{bev} layers to fuse tokens from different sensors. Tokens from the RGB images are fed as values and keys into the decoder, and tokens from the LiDAR points are fed as the $H \times W$ queries to generate BEV features. In addition, two other kinds of queries for the prediction of traffic signs and waypoints w are also fed into the decoder. Following InterFuser [51], we use a 2-layer MLP as the traffic sign classifier to predict the traffic light state and whether there is a stop sign ahead; we then use a single-layer GRU [16] to auto-regressively generate consecutive waypoints $\{w_t\}_{t=1}^{T_f}$ conditioned on the goal location of the ego vehicle. T_f denotes the number of the predicted time steps. To pretrain the perception module in the first training stage, the generated BEV feature is passed through a one-stage CenterPoint [64] to generate the $H \times W \times 7$ BEV map covering an $Hm \times Wm$ spatial region, where the seven channels represent object existence

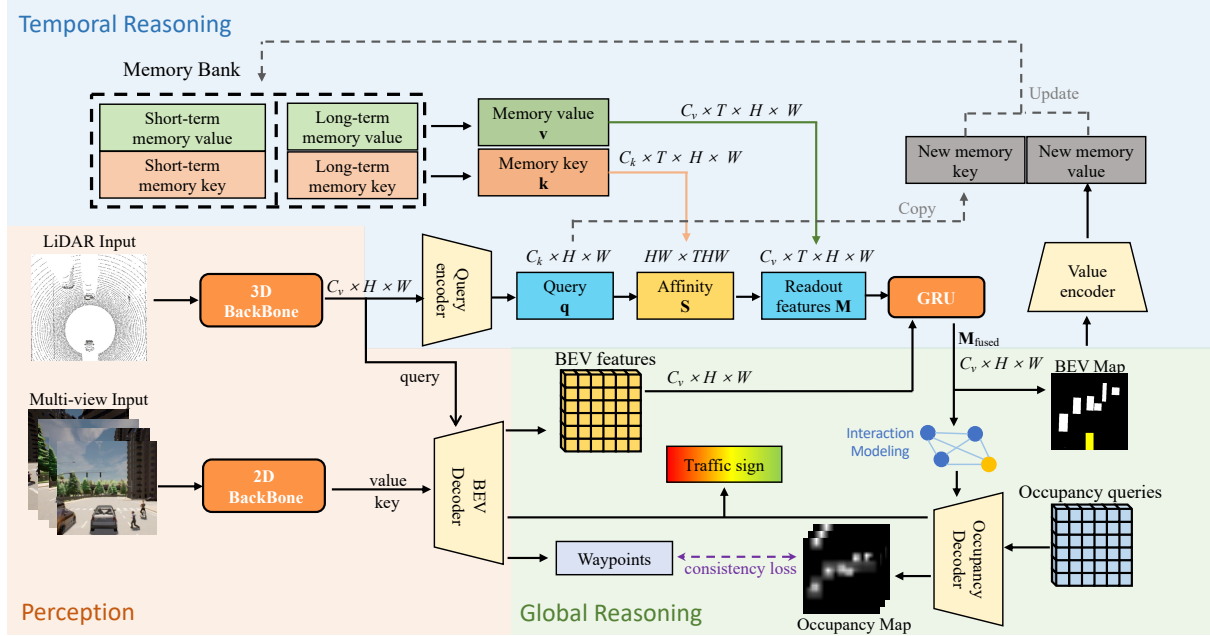


Figure 2. The proposed ReasonNet consists of three modules: 1) the perception module fuses different sensor data to generate the BEV feature, traffic sign feature, and waypoints in the early stage of our framework; 2) the temporal reasoning module processes current and historic features and maintains a memory bank to store historic features; 3) the global reasoning module models the interaction and relationship among objects and the environment to detect adverse events (e.g. occlusion) and improve overall perception performance.

probability, offset from grid center, bounding box extent, heading angle and velocity for objects at each grid cell.

3.2. Temporal Reasoning module

Compared to existing end-to-end driving methods that only exploit scene information of the current frame [10, 15] or simply concatenate features of historic frames [54], we propose a temporal reasoning module that can sufficiently store and fuse temporal information to benefit the motion forecasting of traffic participants and the tracking of intermittently occluded objects. As shown in Figure 2, our temporal reasoning module includes temporal processing to fuse current and historic features through an attention mechanism, and maintains a memory bank which stores historic short-term and long-term feature keys and values.

Temporal Processing Considering that information in different historic frames could have different relevance to the current scene, we apply an attention-based memory reading from the historic features. Specifically, for each historic frame t stored in the memory, we first measure its relevance by calculating the normalized similarity S between the historic-frame feature key $\mathbf{k} \in \mathbb{R}^{C_k \times T_h \times H \times W}$ ¹ and the current-frame feature query $\mathbf{q} \in \mathbb{R}^{C_k \times H \times W}$:

$$S(\mathbf{q}_{h,w}, \mathbf{k}_{t,i,j}) = \frac{(\mathbf{k}_{t,i,j} - \mathbf{q}_{h,w})^2}{\sum_{i=0, j=0}^{i=H-1, j=W-1} (\mathbf{k}_{t,i,j} - \mathbf{q}_{h,w})^2} \quad (1)$$

¹ C, H, W denotes the channel, height, and width of the feature respectively, T_h denotes the number of the frames stored in the memory bank.

We map every query element to a distribution over $H \times W$ memory elements and correspondingly aggregate their values v to obtain the readout feature $\mathbf{M} \in \mathbb{R}^{C_v \times T_h \times H \times W}$ for each frame t stored in the memory:

$$\mathbf{M}_{t,h,w} = \sum_{i=0, j=0}^{i=H-1, j=W-1} v_{t,i,j} S(\mathbf{q}_{h,w}, \mathbf{k}_{t,i,j}) \quad (2)$$

The aggregated features from all historic frames are then concatenated with the current-frame feature value to get $\mathbf{M}' \in \mathbb{R}^{C_v \times (T_h+1) \times H \times W}$, which is then passed through a GRU to progressively fuse temporal information and get $\mathbf{M}_{fused} \in \mathbb{R}^{C_v \times H \times W}$ as the final output of the module. Technically, we take the L2 similarity proposed in STCN [13] as the similarity measure function, which is more stable than the dot product [46]. The current-frame feature query \mathbf{q} is obtained by passing the features from the 3D backbone $\mathbf{F} \in \mathbb{R}^{C_v \times H \times W}$ through a query encoder (several convolution layers). The historic-frame feature key \mathbf{k} and value \mathbf{v} are taken from the temporal memory bank.

Memory Bank Maintaining As above, we have introduced the temporal processing at one single frame. After every τ frame, the obtained feature key and value at that frame will be used to update the memory bank. Specifically, the current-frame feature query \mathbf{q} is directly copied and fed into the memory bank as the memory key without extra computation. The final output \mathbf{M}_{fused} will first be encoded

to a BEV map \mathbf{M}_p . The BEV map \mathbf{M}_p will be concatenated with the final output \mathbf{M}_{fused} and passed through a value encoder to obtain the memory value \mathbf{v} , which is fed into the memory bank. With the above key-value pairs, the memory bank maintains two kinds of buffer: the short-term and long-term buffer. On the one hand, the new key-value pair will be appended to the short-term buffer, as a high-resolution memory of the scene in the past few seconds for accurate feature matching. Considering the limited GPU memory resources, we limit the buffer size and older key-value pairs will be discarded when the limit number T_s is reached. However, when these older features are discarded, the long-term behavior of the traffic participants is missing, which can be crucial for motion forecasting in complex traffic scenarios. Thus on the other hand, inspired by XMem [12], the memory bank also maintains a long-term buffer that selectively stores important/representative key-value feature pairs discarded by the short-term buffer. Considering the fact that the objects surrounding the ego vehicle are sparse most of the time², the long-term buffer selectively stores key-value features (\mathbf{k} and \mathbf{v}) which meet one of the two criteria: 1) their corresponding location in the BEV map \mathbf{M}_p has a high probability of object existence; 2) their usage frequency is in top- K of all candidate key-value features. The usage frequency is defined by its cumulative normalized similarity (Eq. 1). The features selected by the above criteria are appended to the last frame of memory. And if the last memory frame is full, we will initialize one new frame with the zero vector and set it as the last frame to store new features. When the number of frames reaches the limit T_l , the obsolete memory will be removed. Such a compact storing strategy can efficiently track long-term representative features and intermittently occluded objects, while balancing the resources required.

3.3. Global Reasoning module

Rare adverse events such as occluded objects are a notorious issue for the practical deployment of AVs. Our insight is that humans perceive their surroundings not only through sensors, but also by exploiting global information on the scene to reason over the unobservable spaces. For instance, when a vehicle performs an emergency stop without a clear reason, humans can infer that there is potentially an occluded object ahead of the vehicle and will drive more cautiously. Thus we propose the global reasoning module to capture the interaction and relationship between objects and the environment to detect adverse events (e.g. occlusion) and improve overall perception performance. The module consists of three parts: 1) an object-environment and object-object interaction modeling process; 2) an occupancy decoder to generate the occupancy map; 3) a consistency loss

²Based on the data collected from the CARLA simulator, only 7% of the ego-vehicle-centered BEV map area is occupied by active objects.

to encourage consistent prediction of waypoints and the occupancy map.

Interaction Modeling The object-environment and object-object interaction modeling process aims at reasoning about the relationship among objects and the environment. On the one hand, \mathbf{M}_{fused} features whose corresponding location in the BEV map \mathbf{M}_p has a high probability of object existence will be extracted to represent object features. On the other hand, \mathbf{M}_{fused} features will also be downsampled to represent the environment features. All object and environment features are used to construct a graph, which is passed through a graph attention network (GAT) [56] for interaction modeling.

Occupancy Decoder Taking the features outputted by the GAT as keys and values, and the learnable positional embeddings as queries, the occupancy decoder utilizes a transformer decoder with K_{opy} layers to generate: 1) the traffic sign feature, which is then concatenated with the traffic sign feature from the BEV decoder to generate the final traffic sign prediction; 2) the occupancy map feature, which is then applied with convolution operation to generate the occupancy map $\mathbf{O}_t \in \mathbb{R}^{T_f \times H \times W}$. At a future time t , each cell in the occupancy map contains a value in the range $[0,1]$ representing the probability that the cell is occupied.

Consistency Loss Currently, our framework predicts the waypoints and the occupancy map independently, which are not necessarily consistent. For example, the waypoints could overlap some obstacles in the occupancy map. Thus we propose a consistency loss to discourage waypoints' crossing the high-probability region of the occupancy map. Further, the consistency loss also encourages generating longer waypoint trajectories for efficient driving. Specifically, the consistency loss aims at minimizing the average object existence probability of the cells that cover the predicted waypoints, and maximizing the average l_2 length of the waypoint trajectory \mathbf{w} :

$$\mathcal{L}_{\text{consistency}} = \frac{1}{T_f} \left(\sum_{t=0}^{T_f} \frac{\sum_{i=0}^{N_c^t} \mathbf{O}_{t,i}}{N_c^t} - \lambda \sum_{t=0}^{T_f} \|\mathbf{w}_t\|_1 \right) \quad (3)$$

, where N_c^t denotes the number of covered cell at step t , $\mathbf{O}_{t,i}$ denotes the object existence probability at cell i at time t .

3.4. Control

Following [11], we use two PID controllers for latitudinal and longitudinal control, to track the heading and velocity of predicted waypoints respectively. If a red traffic light or stop sign is detected, the ego-vehicle will brake. Additionally, an emergency stop will also be performed if the ego vehicle's current bounding box crosses the area in the occupancy map that has a high object existence probability or if the future waypoints overlap with objects in the BEV map.

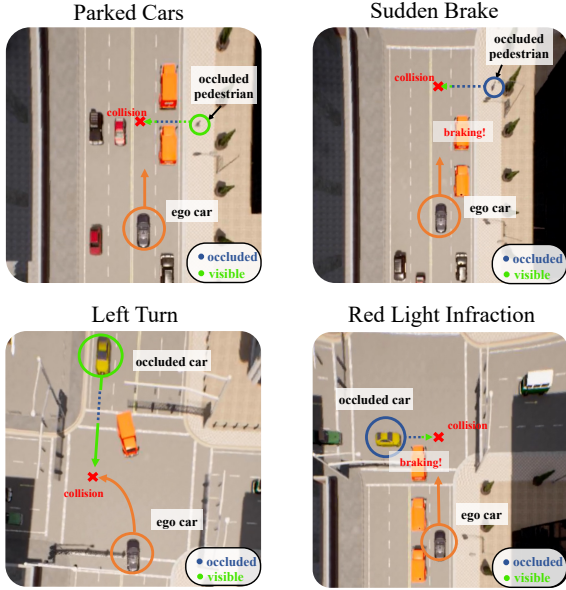


Figure 3. An illustration for the four types of occlusion scenarios included in the proposed DOS benchmark. The orange color denotes the ego car. The blue/green dots denote the occluded/visible trajectory of the occluded dangerous object.

3.5. Training Setup

The training of our framework consists of two stages. In the first stage, we train the perception module to predict BEV features, traffic sign features, and waypoints. Specifically, the loss of BEV features and traffic sign features is computed with additional prediction heads, which are discarded in the next stage. In the second stage, we freeze the perception module and train the other two modules. Five loss terms are considered: 1) the waypoints loss \mathcal{L}_w that minimizes the error between predicted waypoints and expert waypoints; 2) the BEV map loss \mathcal{L}_{BEV} that follows [10, 64] to minimize the current-frame BEV map prediction error; 3) the traffic sign loss \mathcal{L}_{sign} for the traffic regulation prediction; 4) occupancy map loss \mathcal{L}_{opy} that minimizes the occupancy prediction error in a future horizon; 5) the consistency loss $\mathcal{L}_{consistency}$ which encourages a consistent generation of the waypoints and occupancy map. These loss terms are balanced by corresponding loss weights.

4. Drive in Occlusion Sim (DOS) Benchmark

In order to address the issue that occlusion events are rare in existing datasets and benchmarks, we present the Drive in Occlusion Simulation benchmark (DOS), a CARLA-based framework providing diverse driving scenarios with occluded objects. As shown in Figure 3, the proposed DOS benchmark includes four types of challenging occlusion driving scenarios:

Parked Cars (#1) The ego vehicle is driving in a straight

Rank	Method	DS \uparrow	RC \uparrow	IS \uparrow
1	ReasonNet (Ours)	79.95	89.89	0.89
2	InterFuser [51]	76.18	88.23	0.84
3	TCP [63]	75.14	85.63	0.87
4	LAV [10]	61.85	94.46	0.64
5	TransFuser [15]	61.18	86.69	0.71
6	Latent TransFuser [15]	45.20	66.31	0.72
7	GRIAD [8]	36.79	61.85	0.60
8	TransFuser+ [1]	34.58	69.84	0.56

Table 1. Performance comparison on the public CARLA leaderboard [53] (accessed Nov 2022). For all three metrics, higher is better. Our method ranks first overall on the leaderboard, with the highest driving score (DS) and infraction score (IS), and the second highest route completion (RC).

lane with parked cars on the side. Pedestrians can first appear on the sidewalk (visible) and then suddenly emerge through the occluded areas between parked cars (occluded).

Sudden Brake (#2) The ego vehicle is driving in a straight lane along with other vehicles ahead. Pedestrians can suddenly emerge from the sidewalks, causing the other vehicles to brake while remaining invisible to the ego vehicle.

Left Turn (#3) The ego vehicle intends to perform an unprotected left turn at an intersection, but a truck in the opposite lane blocks the view of oncoming traffic, intermittently obscuring vehicles driving straight through the intersection.

Red Light Infraction (#4) The ego vehicle is crossing an intersection after some trucks. A left-to-right vehicle running a red light suddenly appears, forcing the trucks to brake promptly. But the ego vehicle’s view toward the running-light vehicle is blocked by the trucks, so it remains invisible to the ego vehicle.

Each of the four scenarios in the DOS benchmark comprises 25 different cases varying in the road environment and background traffic. Compared to a previous occlusion benchmark AUTOCASIM [19], the DOS benchmark: 1) includes occlusions of both vehicles and pedestrians, instead of only vehicles; 2) includes 100 cases of 4 scenarios, instead of only 3 cases of 3 scenarios; 3) considers specific occlusions that can potentially be resolved by temporal reasoning (intermittent occlusion, #1, #3) and global reasoning (constant occlusion but with interaction clues, #2, #4) about the scene, instead of random occlusions as in AUTOCASIM. Thus our scenarios can also serve as a good tracking-with-intermittent-occlusion benchmark and a People-as-Sensor [2, 31] benchmark.

5. Experiments

5.1. Experiment Setup

Implementation We implement and evaluate our approach on the open-source CARLA simulator with version

Setting		Town 05 Long						DOS			
T_s	T_l	DS \uparrow	RC \uparrow	IS \uparrow	CR \downarrow	Red \downarrow	Blocked \downarrow	SR#1 \uparrow	SR#2 \uparrow	SR#3 \uparrow	SR#4 \uparrow
0	0	66.7 \pm 3.8	97.6\pm2.7	0.68 \pm 0.03	0.18 \pm 0.03	0.05 \pm 0.02	0.03\pm0.03	22 \pm 1.6	28 \pm 3.4	26 \pm 2.1	25 \pm 1.6
1	0	67.9 \pm 3.4	96.8 \pm 2.3	0.70 \pm 0.02	0.16 \pm 0.04	0.04 \pm 0.03	0.05 \pm 0.02	30 \pm 3.6	38 \pm 3.6	32 \pm 2.8	32 \pm 3.4
2	0	68.1 \pm 3.1	96.9 \pm 3.4	0.70 \pm 0.03	0.16 \pm 0.03	0.04 \pm 0.02	0.05 \pm 0.03	28 \pm 5.5	48 \pm 4.1	38 \pm 4.4	52 \pm 3.9
2	1	70.9 \pm 2.0	95.7 \pm 3.1	0.74 \pm 0.02	0.13 \pm 0.02	0.04 \pm 0.02	0.06 \pm 0.04	55 \pm 4.4	57 \pm 4.1	48 \pm 4.1	55 \pm 5.5
4	0	70.5 \pm 2.1	96.4 \pm 2.5	0.73 \pm 0.04	0.14 \pm 0.03	0.03 \pm 0.02	0.06 \pm 0.03	32 \pm 5.4	58 \pm 4.4	40 \pm 5.5	55 \pm 4.9
4	2	73.2\pm1.9	95.9 \pm 2.3	0.76\pm0.03	0.11\pm0.02	0.03\pm0.01	0.07 \pm 0.03	63\pm4.2	73\pm3.6	80\pm4.2	70\pm5.5

Table 2. Ablation study on different short-term buffer size T_s and long-term buffer size T_l , on the Town 05 Long benchmark and the proposed DOS benchmark. Performance is evaluated over three runs. CR: Collision rate, Red: Red light violation, Blocked: Vehicle blocked, SR: Success rate. SR#1 denotes the first kind of scenario in the DOS benchmark. As the two buffer sizes increase, improvement is witnessed in all metrics but the road completion.

0.9.10.1 [21]. We use ResNet-50 pretrained on ImageNet as the 2D backbone and PointPillars trained from scratch as the 3D backbone. We predict $T_f = 4$ time steps for the waypoints and occupancy map, and the interval between each time step is 0.5 seconds. The memory bank maintains $T_s = 4$ frames in the short-term buffer and $T_l = 2$ frames in the long-term buffer. The memory bank is updated every $\tau = 2$ frame. We refer readers to Appendix A for more details.

Dataset Collection We collect an expert dataset of 2M frames by running a rule-based expert agent on all 8 public towns and 21 types of weather, with the access to the privileged information in the CARLA simulator. We randomly set routes, spawn dynamic objects and adversarial scenarios provided in [47], to diversify the collected data. To ensure the temporal continuity of collected data, the data are collected at a high frequency of 10HZ.

Metrics We consider three major metrics introduced by the CARLA LeaderBoard: route completion ratio (RC), infraction score (IS), and driving score (DS). The route completion ratio is the percentage of the route completed. The infraction score measures infractions triggered. When collisions or traffic rule violations occur, the infraction score will decay by a discount factor. The driving score is the product of the route completion ratio and the infraction score, describing both driving progress and safety, and thus is the primary ranking metric in the CARLA Leaderboard.

5.2. Comparison to the state of the art

Table 1 shows the top 8 entries on the public CARLA Leaderboard. Readers can refer to Sec 2 for descriptions of these methods. Our method outperforms all prior methods, with the highest driving score and infraction score, and the second highest route completion. The previous leading method InterFuser uses a transformer for sensor fusion but lacks temporal and global reasoning. Compared to InterFuser, our method improved the driving score, road completion, and infraction score by 5%, 2%, and 6% respectively.

5.3. Ablation study

We investigate the effect of the temporal and global reasoning modules on the Town05 Long benchmark and the DOS benchmark. For each scenario in DOS, we take 5 cases for training and 20 cases for evaluation. In addition to the three metrics mentioned earlier, we also present four more metrics for detailed analysis: collision rate (CR), red light violation (Red), ego vehicle blocked frequency (Blocked), and success rate (SR). The first three metrics are normalized by the driven distance (km). Visualizations of how the temporal reasoning and global reasoning work can be found at Figure 4 and Figure 5 respectively.

Memory Size Table 2 studies the effect of different short-term buffer size T_s and long-term buffer size T_l . The overall observation is that, as the two buffer sizes increase, improvement is witnessed in all metrics but road completion. Specifically, when the long-term memory is removed ($T_l = 0$), the average success rates drop sharply from 71.5 to 36 on DOS scenarios that require keeping track of intermittently occluded objects (#1 and #3). If we remove the temporal reasoning module ($T_s = T_l = 0$), the driving score on the Town05 benchmark drops by 9%, and the average success rate on the DOS benchmark drops by 46%. We hypothesize that the drop in performance is because 1) it can be really hard to accurately estimate the objects' future motion based only on single-frame data; 2) temporal information can help keep track of objects that are intermittently occluded; 3) the global reasoning module may also work poorly when historic information is missing.

Long-Term Memory Selection Strategy Table 3 studies the performance of different long-term memory selection strategies. Specifically, the proposed strategy in Sec 3.2 includes two selection criteria. So here we ablate the effect of the two criteria by 1) only selecting the short-term feature with top- K usages (usage-based); 2) only selecting the feature with a high probability of the existence of an object (object-based). Besides, we also compared a random selection strategy. As in Table 3, the random selection strategy has the poorest performance especially on the DOS benchmark, as random selection could miss important and repre-

Setting	Town 05 Long						DOS			
	DS \uparrow	RC \uparrow	IS \uparrow	CR \downarrow	Red \downarrow	Blocked \downarrow	SR#1 \uparrow	SR#2 \uparrow	SR#3 \uparrow	SR#4 \uparrow
Random	71.2 \pm 5.4	96.6\pm2.4	0.74 \pm 0.04	0.13 \pm 0.04	0.03 \pm 0.01	0.06 \pm 0.02	33 \pm 4.4	55 \pm 6.2	42 \pm 5.5	53 \pm 5.4
Usage-based	72.0 \pm 3.9	95.9 \pm 2.2	0.75 \pm 0.04	0.12 \pm 0.03	0.03 \pm 0.01	0.06 \pm 0.02	45 \pm 4.2	62 \pm 3.4	53 \pm 2.8	62 \pm 3.9
Object-based	72.2 \pm 3.7	96.1 \pm 3.0	0.75 \pm 0.03	0.12 \pm 0.03	0.03 \pm 0.01	0.05\pm0.02	57 \pm 4.1	65 \pm 3.6	73 \pm 4.4	60 \pm 3.7
Full (Ours)	73.2\pm1.9	95.9 \pm 2.3	0.76\pm0.03	0.11\pm0.02	0.03\pm0.01	0.07 \pm 0.03	63\pm4.2	73\pm3.6	80\pm4.2	70\pm5.5

Table 3. Ablation study on different long-term memory selection strategies. Our proposed strategy considering both the usage and object criteria outperforms the random selection strategy and the two methods with only one criteria, especially on the DOS benchmark.

Setting	Town 05 Long						DOS			
	DS \uparrow	RC \uparrow	IS \uparrow	CR \downarrow	Red \downarrow	Blocked \downarrow	SR#1 \uparrow	SR#2 \uparrow	SR#3 \uparrow	SR#4 \uparrow
No global reasoning	68.9 \pm 4.6	97.4\pm2.9	0.71 \pm 0.04	0.15 \pm 0.04	0.05 \pm 0.02	0.05\pm0.02	28 \pm 2.8	34 \pm 3.4	29 \pm 2.0	27 \pm 3.6
No consistency loss	72.2 \pm 3.4	96.1 \pm 3.2	0.75 \pm 0.03	0.12 \pm 0.02	0.03\pm0.02	0.06 \pm 0.03	60 \pm 4.1	72 \pm 3.9	77 \pm 4.9	68 \pm 4.2
No traffic sign prediction	71.1 \pm 2.7	96.0 \pm 4.1	0.74 \pm 0.03	0.11\pm0.03	0.05 \pm 0.03	0.07 \pm 0.03	62 \pm 4.4	72 \pm 4.0	82\pm2.8	70\pm4.1
Full (Ours)	73.2\pm1.9	95.9 \pm 2.3	0.76\pm0.03	0.11\pm0.02	0.03\pm0.01	0.07 \pm 0.03	63\pm4.2	73\pm3.6	80 \pm 4.2	70\pm5.5

Table 4. Ablation study on the global reasoning module. The performance would drop when 1) the entire global reasoning module is removed; 2) the consistency loss is not applied; 3) the traffic sign feature from the reasoning module is not utilized.

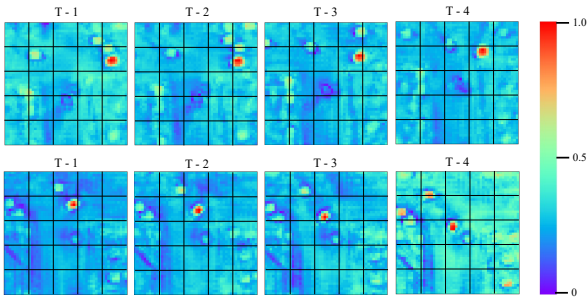


Figure 4. Visualization of the attention map between one object's current-frame feature query and the historic-frame feature stored in the short-term buffer, in two cases. The object's current-frame feature consistently attends to its corresponding region in the historic feature map.

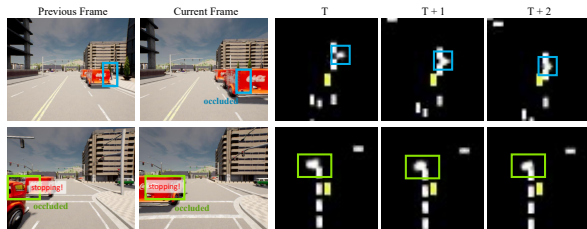


Figure 5. We show two cases of how our framework reasons the presence of the occluded object. In the first case, a pedestrian first appeared on the sidewalk (visible) and then emerges between two parked cars (occluded). In the second case, a vehicle runs the red light, forcing trucks to brake abruptly. But the ego vehicle's view toward the running-light vehicle is blocked by the front trucks, so the running-light vehicle remains invisible to the ego vehicle. The rectangles mark the occluded objects.

sentative features on the scene. Compared to our strategy utilizing both criteria, the two ablations omitting one of the criteria have a performance drop, especially on the DOS benchmark. The usage-based strategy performs worse than the object-based, showing that the features of objects could be more informative for capturing historic behaviors.

Global Reasoning Design Table 4 studies the performance when different designs of the global reasoning module are applied. First, we remove the entire module and observe a significant drop in all metrics but the road completion. For instance, the average success rate on the DOS benchmark dropped from 71.5 to 29.5. This demonstrates the effectiveness of global reasoning, especially in occlusion events. Second, we ablated the consistency loss, which could alleviate the sub-optimal issues in collected expert data. A removal of consistency loss leads to a lower driving score and higher collision rate on the Town 05 benchmark and a lower success rate on the DOS benchmark. Third, excluding the traffic sign feature from the global reasoning model results in an increase on the red light violation. One explanation is that the traffic sign feature from the global reasoning module could help reason the distant traffic light state according to other road participants' behavior.

6. Conclusion

We present ReasonNet, a novel end-to-end autonomous driving framework including two major components: a temporal reasoning module and a global reasoning module. The temporal reasoning module processes the historic information on the driving scene for high-fidelity forecasting of other road participants and dynamically maintains a temporal memory bank. The global reasoning module models the interaction and relationship among the objects and environment to detect adverse events, especially occlusion, and improve overall perception performance. Our method pushes the state-of-the-art performance of the CARLA leaderboard by a considerable margin. Moreover, we also publicly release a new benchmark DOS consisting of diverse occlusion scenarios, to facilitate the study of occlusion detection in the field of end-to-end autonomous driving.

References

- [1] Expert drivers for autonomous driving. [url=https://kait0.github.io/files/master_thesis_bernhard_jaeger.pdf](https://kait0.github.io/files/master_thesis_bernhard_jaeger.pdf), 2021. **6, 13**
- [2] Oladapo Afolabi, Katherine Driggs-Campbell, Roy Dong, Mykel J Kochenderfer, and S Shankar Sastry. People as sensors: Imputing maps from human actions. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2342–2348. IEEE, 2018. **6**
- [3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. **3**
- [4] Neal E. Boudette. Teslas self-driving system cleared in deadly crash. *The New York Times*, 2017. **1**
- [5] Jinkun Cao, Xinshuo Weng, Rawal Khirodkar, Jiangmiao Pang, and Kris Kitani. Observation-centric sort: Rethinking sort for robust multi-object tracking. *arXiv preprint arXiv:2203.14360*, 2022. **1**
- [6] Sergio Casas, Cole Gulino, Simon Suo, Katie Luo, Renjie Liao, and Raquel Urtasun. Implicit latent variable model for scene-consistent motion forecasting. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 624–641. Springer, 2020. **1**
- [7] Sergio Casas, Wenjie Luo, and Raquel Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *Conference on Robot Learning*, pages 947–956. PMLR, 2018. **3**
- [8] Raphael Chekroun, Marin Toromanoff, Sascha Hornauer, and Fabien Moutarde. Gri: General reinforced imitation and its application to vision-based autonomous driving. *arXiv preprint arXiv:2111.08575*, 2021. **6, 13**
- [9] Dian Chen, Vladlen Koltun, and Philipp Krährenbühl. Learning to drive from a world on rails. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15590–15599, 2021. **1, 2, 13, 14**
- [10] Dian Chen and Philipp Krährenbühl. Learning from all vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17222–17231, 2022. **1, 2, 4, 6, 13**
- [11] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krährenbühl. Learning by cheating. In *Conference on Robot Learning*, pages 66–75. PMLR, 2020. **2, 5, 13, 14**
- [12] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision*, pages 640–658. Springer, 2022. **5**
- [13] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *Advances in Neural Information Processing Systems*, 34:11781–11794, 2021. **4**
- [14] Kashyap Chitta, Aditya Prakash, and Andreas Geiger. Neat: Neural attention fields for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15793–15803, 2021. **3, 12, 13, 14**
- [15] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *arXiv preprint arXiv:2205.15997*, 2022. **2, 3, 4, 6, 13**
- [16] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. **3**
- [17] Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4693–4700. IEEE, 2018. **2**
- [18] Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9329–9338, 2019. **2, 13, 14**
- [19] Jiaxun Cui, Hang Qiu, Dian Chen, Peter Stone, and Yuke Zhu. Coopernaut: End-to-end driving with cooperative perception for networked vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17252–17262, 2022. **6**
- [20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. **3**
- [21] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. **2, 7, 13**
- [22] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. **2**
- [23] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11525–11533, 2020. **3**
- [24] Samuel Gibbs. Ubers self-driving car saw the pedestrian but didnt swerve—report. *The Guardian*, 2018. **1**
- [25] Chao Gou, Yuchen Zhou, and Dan Li. Driver attention prediction based on convolution and transformers. *The Journal of Supercomputing*, 78(6):8268–8284, 2022. **3**
- [26] Junru Gu, Chen Sun, and Hang Zhao. Densent: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15303–15312, 2021. **1**
- [27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. **3, 12**

- [28] Anthony Hu, Zak Murez, Nikhil Mohan, Sofia Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. Fiery: future instance prediction in bird’s-eye view from surround monocular cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15273–15282, 2021. [1](#)
- [29] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII*, pages 533–549. Springer, 2022. [3](#)
- [30] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. [12](#)
- [31] Masha Itkina, Ye-Ji Mun, Katherine Driggs-Campbell, and Mykel J Kochenderfer. Multi-agent variational occlusion inference using people as sensors. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4585–4591. IEEE, 2022. [6](#)
- [32] Xiaosong Jia, Li Chen, Penghao Wu, Jia Zeng, Junchi Yan, Hongyang Li, and Yu Qiao. Towards capturing the temporal dynamics for trajectory prediction: a coarse-to-fine approach. In *Conference on Robot Learning*, pages 910–920. PMLR, 2023. [1](#)
- [33] Xiaosong Jia, Penghao Wu, Li Chen, Hongyang Li, Yu Liu, and Junchi Yan. Hdgt: Heterogeneous driving graph transformer for multi-agent trajectory prediction via scene encoding. *arXiv preprint arXiv:2205.09753*, 2022. [1](#)
- [34] Jinkyu Kim, Teruhisa Misu, Yi-Ting Chen, Ashish Tawari, and John Canny. Grounding human-to-vehicle advice for self-driving vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10591–10599, 2019. [3](#)
- [35] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. [3](#)
- [36] Lingyun Luke Li, Bin Yang, Ming Liang, Wenyuan Zeng, Mengye Ren, Sean Segal, and Raquel Urtasun. End-to-end contextual perception and prediction with interaction transformer. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5784–5791. IEEE, 2020. [3](#)
- [37] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022. [1](#), [2](#)
- [38] Qing Lian, Peiliang Li, and Xiaozhi Chen. Monojsj: Joint semantic and geometric cost volume for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1070–1079, 2022. [1](#)
- [39] Qing Lian, Yanbo Xu, Weilong Yao, Yingcong Chen, and Tong Zhang. Semi-supervised monocular 3d object detection by multi-view consistency. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VIII*, pages 715–731. Springer, 2022. [1](#)
- [40] Qing Lian, Botao Ye, Ruijia Xu, Weilong Yao, and Tong Zhang. Exploring geometric consistency for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1685–1694, 2022. [1](#)
- [41] Ming Liang, Bin Yang, Wenyuan Zeng, Yun Chen, Rui Hu, Sergio Casas, and Raquel Urtasun. Pnpnet: End-to-end perception and prediction with tracking in the loop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11553–11562, 2020. [3](#)
- [42] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird’s-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022. [1](#)
- [43] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. [12](#)
- [44] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. [12](#)
- [45] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8844–8854, 2022. [3](#)
- [46] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. *Advances in neural information processing systems*, 34:12493–12506, 2021. [4](#)
- [47] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7077–7087, 2021. [2](#), [3](#), [7](#), [12](#), [13](#), [14](#)
- [48] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. [3](#), [12](#)
- [49] Shengju Qian, Hao Shao, Yi Zhu, Mu Li, and Jiaya Jia. Blending anti-aliasing into vision transformer. *Advances in Neural Information Processing Systems*, 34:5416–5429, 2021. [3](#)
- [50] Hao Shao, Shengju Qian, and Yu Liu. Temporal interlacing network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11966–11973, 2020. [2](#)
- [51] Hao Shao, Letian Wang, Ruobing Chen, Hongsheng Li, and Yu Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. *arXiv preprint arXiv:2207.14024*, 2022. [1](#), [2](#), [3](#), [6](#), [13](#), [14](#)
- [52] Peize Sun, Jinkun Cao, Yi Jiang, Rufeng Zhang, Enze Xie, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple object tracking with transformer. *arXiv preprint arXiv:2012.15460*, 2020. [3](#)

- [53] CARLA team. Carla autonomous driving leaderboard. <https://leaderboard.carla.org/>, 2020. Accessed: 2021-02-11. 1, 6, 12, 13
- [54] Marin Toromanoff, Emilie Wirbel, and Fabien Moutarde. End-to-end model-free reinforcement learning for urban driving using implicit affordances. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7153–7162, 2020. 2, 4, 13
- [55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3
- [56] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *stat*, 1050:20, 2017. 5
- [57] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4604–4612, 2020. 2
- [58] Letian Wang, Yeping Hu, Liting Sun, Wei Zhan, Masayoshi Tomizuka, and Changliu Liu. Hierarchical adaptable and transferable networks (hatn) for driving behavior prediction. *arXiv preprint arXiv:2111.00788*, 2021. 3
- [59] Letian Wang, Yeping Hu, Liting Sun, Wei Zhan, Masayoshi Tomizuka, and Changliu Liu. Transferable and adaptable driving behavior prediction. *arXiv preprint arXiv:2202.05140*, 2022. 1
- [60] Letian Wang, Liting Sun, Masayoshi Tomizuka, and Wei Zhan. Socially-compatible behavior design of autonomous vehicles with verification on real human data. *IEEE Robotics and Automation Letters*, 6(2):3421–3428, 2021. 1
- [61] Bob Wei, Mengye Ren, Wenyuan Zeng, Ming Liang, Bin Yang, and Raquel Urtasun. Perceive, attend, and drive: Learning spatial attention for safe self-driving. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4875–4881. IEEE, 2021. 1
- [62] Pengxiang Wu, Siheng Chen, and Dimitris N Metaxas. Motionnet: Joint perception and motion prediction for autonomous driving based on bird’s eye view maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11385–11395, 2020. 3
- [63] Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. *arXiv preprint arXiv:2206.08129*, 2022. 2, 6, 13
- [64] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021. 3, 6
- [65] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *European Conference on Computer Vision*, pages 1–21. Springer, 2022. 1
- [66] Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-end urban driving by imitating a reinforcement learning coach. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision, pages 15222–15232, 2021. 2, 13, 14

A. Implementation Details

Model Details The feature dimension of all decoders in our framework is set as 256. We use $K_e = 1$, $K_{bev} = 3$, $K_{copy} = 3$, $C^K = 64$, $C^V = 256$ for the feature dimensions mentioned in Sec 3. The feature of the 5th stage in Resnet was used as the feature map f_i in the 2D backbone. We use Fully Connected Layer and Batch Normalization [30] to construct a simplified version of PointNet [48] to encode the information of raw LiDAR points in the 3D backbone.

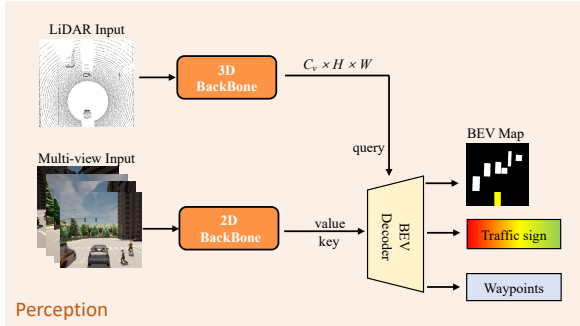


Figure 6. Overview of our pipeline for pretraining the perception module in the first training stage.

Training We train our models using the AdamW optimizer [44] and a cosine learning rate scheduler [43]. In the first training stage, the initial learning rate is set to $5e^{-4} \times \frac{BatchSize}{512}$ for the transformer encoder and the 3D backbone, and $2e^{-4} \times \frac{BatchSize}{512}$ for the 2D backbones. The weight decay is 0.07. We train the models for 35 epochs with the first 5 epochs for warm-up [27]. We used random scaling from 0.9 to 1.1 and color jittering to augment the collected RGB images. The overview of the first-stage framework can be found in Figure 6. In the second training stage, we freeze the perception module. The training schedule of the other two modules are similar to that in the first stage.

Sensors The RGB images are collected and cropped from one front-facing camera, two side-facing cameras, and one back-facing camera with a resolution of 800×600 . Each camera has a 100° horizontal field of view (FOV), and the side cameras are angled at 60° . For the front image, we scale the shorter side of the front camera input to 256 and crop its center patch of 224×224 . For the focusing-view image, we directly crop the center of the front camera input to get a 128×128 patch. For the other images, the shorter side of the camera input is scaled to 160 and a center patch of 128×128 is taken.

Other hyper-parameter values Some other hyper-parameter values used in ReasonNet are listed in Table 8.

B. Benchmark details

We evaluate our method on the CARLA public leaderboard [53], Town05 benchmark [47], and our proposed DOS benchmark. Adversarial events³ are included in the first two benchmarks, and occlusion events are included in the last benchmark. In these benchmarks, the ego vehicle is required to complete a given route without collision or traffic rules violation.

CARLA Leaderboard The CARLA Autonomous Driving Leaderboard [53] is to evaluate the driving proficiency of autonomous agents in realistic traffic situations with a variety of weather conditions. The CARLA leaderboard provides a set of 76 routes for training and verifying agents and contains a secret set of 100 routes to evaluate the driving performance of the submitted agents.

Town05 benchmark In this benchmark, we use Town05 for evaluation and other towns for training. Following [47], the benchmark includes two evaluation settings: 1) Town05 Short: 10 short routes of 100-500m, each comprising 3 intersections, 2) Town05 Long: 10 long routes of 1000-2000m, each comprising 10 intersections. Town05 is a complex town with multi-lane roads, single-lane roads, bridges, highways and exits. The core challenge of the benchmark is how to handle dynamic dense agents and adversarial events.

CARLA 42 routes benchmark The CARLA 42 routes benchmark was proposed in NEAT [14], including six towns covering a variety of areas such as US-style intersections, EU-style intersections, freeways, roundabouts, stop signs, urban scenes and residential districts. The traffic density of each town is set to be comparable to busy traffic setting. We take the same configuration open-sourced by [47] when we evaluated the methods.

C. More Experimental results

In this section we report additional experimental results, including the CARLA leaderboard and two other benchmarks.

C.1. CARLA leaderboard

Table 5 shows the detailed comparison between our method and the baselines on the CARLA public Leaderboard [53]. Our method also leads the vehicle collision and offroad infraction numbers among all the methods.

C.2. Town05 and CARLA 42 routes

Table 6 and Table 7 additionally compare the driving score, road completion, and infraction score of the presented approach to prior state-of-the-art on the CARLA

³Adversarial events include unexpected agents rushing into the road from occluded regions, vehicles running red traffic lights, etc. Please refer to <https://leaderboard.carla.org/scenarios/> for detailed descriptions.

Rank	Method	Driving Score	Route Completion	Infraction Score	Vehicle Collisions	Pedestrian Collisions	Layout Collisions	Red light Violations	Offroad Infractions	Blocked Infractions
1	ReasonNet (Ours)	79.95	89.89	0.89	0.13	0.02	0.01	0.08	0.04	0.33
2	InterFuser [51]	76.18	88.23	0.84	0.37	0.04	0.14	0.22	0.13	0.43
3	TCP [63]	75.14	85.63	0.87	0.32	0.00	0.00	0.09	0.04	0.54
4	LAV [10]	61.85	94.46	0.64	0.70	0.04	0.02	0.17	0.25	0.10
5	TransFuser [15]	61.18	86.69	0.04	0.71	0.81	0.01	0.05	0.23	0.43
6	Latent TransFuser [15]	45.20	66.31	0.72	1.11	0.02	0.02	0.05	0.16	1.82
7	GRIAD [8]	36.79	61.85	0.60	2.77	0.00	0.41	0.48	1.39	0.84
8	TransFuser+ [1]	34.58	69.84	0.56	0.70	0.04	0.03	0.75	0.18	2.41
9	Rails [9]	31.37	57.65	0.56	1.35	0.61	1.02	0.79	0.96	0.47
10	IARL [54]	24.98	46.97	0.52	2.33	0.00	2.47	0.55	1.82	0.94
11	NEAT [14]	21.83	41.71	0.65	0.74	0.04	0.62	0.70	2.68	5.22

Table 5. Comparison of our method and the state-of-the-art on the public CARLA leaderboard [53] (accessed Nov 2022). Methods are ranked by the driving score as the main metric. Driving Score, Route Completion, Infraction Score are higher the better, and the other metrics are lower the better. We outperform all other methods by a wide margin. We also lead the vehicle collision, offroad infraction numbers among all the methods.

Method	Town05 Short		Town05 Long	
	Driving Score \uparrow	Road Completion \uparrow	Driving Score \uparrow	Road Completion \uparrow
CILRS [18]	7.47 \pm 2.51	13.40 \pm 1.09	3.68 \pm 2.16	7.19 \pm 2.95
LBC [11]	30.97 \pm 4.17	55.01 \pm 5.14	7.05 \pm 2.13	32.09 \pm 7.40
TransFuser [47]	54.52 \pm 4.29	78.41 \pm 3.75	33.15 \pm 4.04	56.36 \pm 7.14
NEAT [14]	58.70 \pm 4.11	77.32 \pm 4.91	37.72 \pm 3.55	62.13 \pm 4.66
Roach [66]	65.26 \pm 3.63	88.24 \pm 5.16	43.64 \pm 3.95	80.37 \pm 5.68
WOR [9]	64.79 \pm 5.53	87.47 \pm 4.68	44.80 \pm 3.69	82.41 \pm 5.01
InterFuser [51]	94.95 \pm 1.91	95.19 \pm 2.57	68.31 \pm 1.86	94.97 \pm 2.87
ReasonNet (Ours)	95.71\pm1.88	96.23\pm3.17	73.22\pm1.91	95.88\pm2.31

Table 6. Comparison of our ReasonNet with six state-of-the-art methods in Town05 benchmark. Our method outperformed other strong methods in all metrics and scenarios.



Figure 7. Different types of weather in our dataset.

Town05 benchmark [47] and CARLA 42 routes benchmark [14].

D. Data statistics

We describe the detailed statistics for each town and their corresponding maps in Table 9. In Figure 7, we show six types of weathers among our dataset. For the submission for the online leaderboard, the model is trained in all eight

towns. For the ablation studies, we train the models on five towns (Town01, Town03, Town04, Town06, and Town07).

E. License of Assets

We use the open-source CARLA driving simulator [21]. CARLA is released under the MIT license. Its assets are under the CC-BY license. The pretrained ResNet model is under the MIT license. The source code for our work will be publicly available once accepted and they are under the CC-BY-NC 4.0 license.

Method	Driving Score \uparrow	Road Completion \uparrow	Infraction Score \uparrow
CILRS [18]	22.97 \pm 0.90	35.46 \pm 0.41	0.66 \pm 0.02
LBC [11]	29.07 \pm 0.67	61.35 \pm 2.26	0.57 \pm 0.02
AIM [47]	51.25 \pm 0.17	70.04 \pm 2.31	0.73 \pm 0.03
TransFuser [47]	53.40 \pm 4.54	72.18 \pm 4.17	0.74 \pm 0.04
NEAT [14]	65.17 \pm 1.75	79.17 \pm 3.25	0.82 \pm 0.01
Roach [66]	65.08 \pm 0.99	85.16 \pm 4.20	0.77 \pm 0.02
WOR [9]	67.64 \pm 1.26	90.16 \pm 3.81	0.75 \pm 0.02
InterFuser [51]	91.84 \pm 2.17	97.12\pm1.95	0.95 \pm 0.02
ReasonNet (Ours)	93.25\pm2.91	96.84 \pm 2.17	0.96\pm0.02

Table 7. Comparison of our ReasonNet with other methods in CARLA 42 routes benchmark. Our method outperformed other strong methods in driving score and infraction score.

Notation	Description	Value
BEV Map and Controller		
a_{max}	Maximum acceleration	1.0 m/s
v_{max}	Maximum velocity	7.5 m/s ²
H, W	Size of the BEV map	50, 50
	Size of the BEV area	50 meter \times 50 meter
H_b	The detection range for the backward of the ego vehicle	20
	Scale factor for bounding box size of pedestrians and bicycles	2
Learning Process		
	Number of epochs	35
	Number of warm-up epochs	5
λ_{sign}	Weight for the traffic sign loss	0.2
λ_w	Weight for the waypoints loss	0.4
λ_{BEV}	Weight for the BEV map loss	0.4
λ_{opy}	Weight for the occupancy map loss	0.2
$\lambda_{consistency}$	Weight for the consistency loss	0.05
	Max norm for gradient clipping	10.0
	Weight decay	0.07
	Batch size	256

Table 8. The parameter used for ReasonNet.

Town Name	#Frames	Description
Town01	342846	A basic town layout consisting of ‘‘T junctions’’
Town02	197240	Similar to Town01, but smaller
Town03	469115	The most complex town, with a 5-lane junction, a roundabout, unevenness, a tunnel, and more
Town04	429979	An infinite loop with a highway and a small town
Town05	297140	Squared-grid town with cross junctions and a bridge. It has multiple lanes per direction.
Town06	148495	Long highways with many highway entrances and exits. It also has a Michigan left
Town07	55299	A rural environment with narrow roads, barns and hardly any traffic lights
Town10	69039	A city environment with different environments such as an avenue or promenade

Table 9. Detailed statistics of the number of frames and a brief description of each town.