A.3 Traffic information loss function

When predicting the traffic information ($\mathcal{L}_{\text{tf}}$), we expect to recognize the traffic light status ($\mathcal{L}_l$), stop sign ($\mathcal{L}_s$), and whether the vehicle is at junction of roads ($\mathcal{L}_j$):

$$\mathcal{L}_{tf} = \lambda_l \mathcal{L}_l + \lambda_s \mathcal{L}_s + \lambda_j \mathcal{L}_j, \tag{12}$$

where λ balances the three loss terms, which are calculated by binary cross-entropy loss.

B Safety controller - desired speed optimization

The desired velocity is expected to: 1) drive the vehicle to the goal point as soon as possible. 2) ensure collision avoidance in a future horizon. 3) consider the dynamic constraint and actuation limit of the ego vehicle. Toward these goals, we set up a linear programming optimization problem, where we try to maximize the desired velocity while the other requirements are achieved by constraints:

$$\begin{aligned} \max_{v_d^1} \quad & v_d^1 \\ \text{s.t.} \quad & (v_0 + v_d^1)T \leqslant s_1 \\ & (v_0 + v_d^1)T + (v_d^1 + v_d^2)T \leqslant s_2 \\ & \left|v_d^1 - v_0\right| T \leqslant a_{max} \\ & \left|v_d^2 - v_d^1\right| T \leqslant a_{max} \\ & 0 \leqslant v_d^1 \leqslant v_{max} \\ & 0 \leqslant v_d^2 \leqslant v_{max} \end{aligned} \tag{13}$$

where we consider a horizon of 1 second, and two desired velocities v_d^1 and v_d^2 are set at 0.5 second and 1 second respectively. v_0 denotes the current velocity of the ego vehicle. T denotes the time step duration (0.5s). s_1 and s_2 denote the maximum safe distance for the ego vehicle to drive at the first step and the second step respectively. v_{max} and a_{max} denotes the constraint on the maximum velocity and acceleration. When determining the maximum safe distance s_1, we augment the shape of other objects for extra safety:

$$\begin{aligned} s_1 &= max(s_1' - \bar{s}, 0) \\ s_2 &= max(s_2' - \bar{s}, 0) \end{aligned} \tag{14}$$

where $\bar{s}$ denote the augmented distance for extra safety. s_1' and s_2' denote the maximum distance the ego vehicle can drive on the predicted route without collision with other objects. Note that in the optimization problem, we maximize the desired velocity at the first step v_d^1, while we set the desired velocity at the second step v_d^2 as a free variable. The constraint on the second step helps the optimization of v_d^1 looks into a future horizon, to avoid future safety intractability due to actuation limit and dynamic constraint.

C Implementation Details

All cameras have a resolution of 800×600 with a horizontal field of view (FOV) 100°. The side cameras are angled at $\pm 60^\circ$ away from the front. For the front view, we scale the shorter side of the raw front RGB image to 256 and crop its center patch of 224×224 as the front image input $\mathbf{I}_{\text{front}}$. For the two side views, the shorter sides of the raw side RGB images are scaled to 160 and a center patch of 128×128 is taken as the side image inputs $\mathbf{I}_{\text{left/right}}$. For the focusing-view image input $\mathbf{I}_{\text{focus}}$ by directly cropping the center patch of the raw front RGB image to 128×128 without scaling.

For the LiDAR point clouds, we follow previous works [36, 37, 8] to convert the LiDAR point cloud data into a 3-bin histogram over a 2-dimensional Bird's Eye View (BEV) grid. The first 2-bin of the histogram in each $0.125m^2$ grid represents the numbers of points above and below the ground plane respectively. The last bin represents the total numbers of points in each grid. This produces

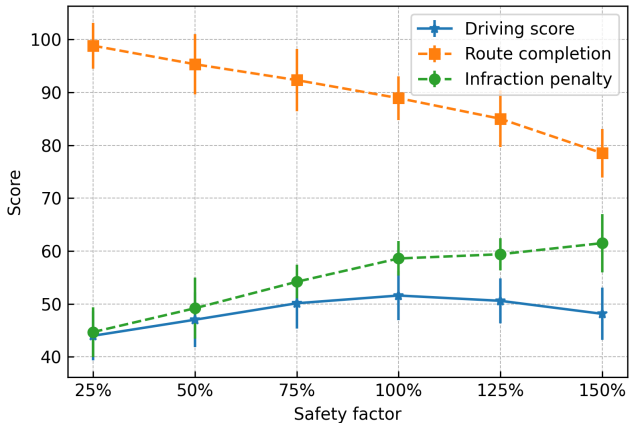

Figure 4: The driving preference varies when different safety factor is assigned to the safety controller. 100 % safety factor refers to the setting $\bar{s} = 2$ and $v_{max} = 6.5$, and 150 % safety factor refers to the setting $\bar{s} = 2 \times 150\%$ and $v_{max} = 6.5/150\%$. The Town05 Long with adversarial events benchmark is used here.

a three-channel LiDAR bird-view projection image input $\mathbf{I}_{\text{lidar}}$, covering the point cloud about 28 meters in front of the ego vehicle and 14 meters to the ego vehicle's two sides.

The backbone for encoding information from multi-view RGB images is Resnet-50 pretrained on ImageNet [56], and the backbone for processing LiDAR BEV representations is ResNet-18 trained from scratch. We take the output of stage 4 in a regular ResNet as the tokens fed into the downstream transformer encoder. The number of layers $\mathcal{K}$ in the transformer decoder and the transformer encoder is 6 and the feature dim d is 256. We train our models using the AdamW optimizer [57] and a cosine learning rate scheduler [58]. The initial learning rate is set to $5e^{-4} \times \frac{BatchSize}{512}$ for the transformer encoder & decoder, and $2e^{-4} \times \frac{BatchSize}{512}$ for the CNN backbones. The weight decay for all models is 0.07. All the models are trained for a maximum of 35 epochs with the first 5 epochs for warm-up [38]. For data augmentation, we used random scaling from 0.9 to1.1 and color jittering. The parameters used in the object density map and the safety controller is listed in Table 6.

The training time of the Interfuser is about 30 hours on 8 Tesla V100 32G graphic cards. The Interfuser consists of 52,935,567 parameters. The inference time of Interfusuer is about 0.04 second per frame on GeForce GTX 1060 (a low-end GPU), and about 0.02 second per frame on GeForce GTX 1080 Ti (a medium-end GPU). In the future, we can apply the tools of model acceleration or model quantization to further reduce the inference time to be less than 0.01 second per frame.

D Benchmark details

Leaderboard The CARLA Autonomous Driving Leaderboard [49] 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 as a starting point for training and verifying agents and contains a secret set of 100 routes to evaluate the driving performance of the submitted agents. However, the evaluation on the online CARLA leaderboard usually takes about 150 hours and each team is restricted to using this platform for only 200 hours per month. Therefore, we use the CARLA leaderboard for the state-of-the-art comparison, and use the following Town05 benchmark for quick development and detailed ablation studies.

Town05 benchmark In the Town05 benchmark, we use Town05 for evaluation and other towns for training. Following [8], 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 has a wide variety of road types, including multi-lane roads, single-lane roads, bridges, highways and exits. The core challenge of the benchmark is how to handle dynamic agents and adversarial events.

CARLA 42 routes benchmark The CARLA 42 routes benchmark [17] considers six towns covering a variety of areas such as US-style intersections, EU-style intersections, freeways, roundabouts, stop

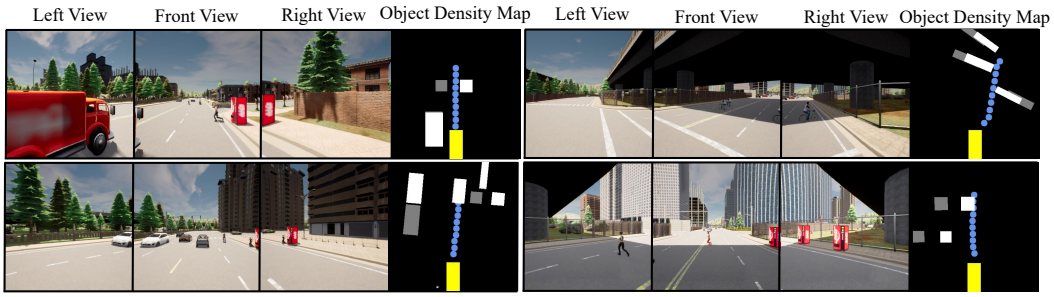

Figure 5: Four cases of how our method predicts waypoints and recover the traffic scene. Blue points denote predicted waypoints. Yellow rectangle represents the ego vehicle, and white/grey rectangles denote the current/future positions of detected objects.

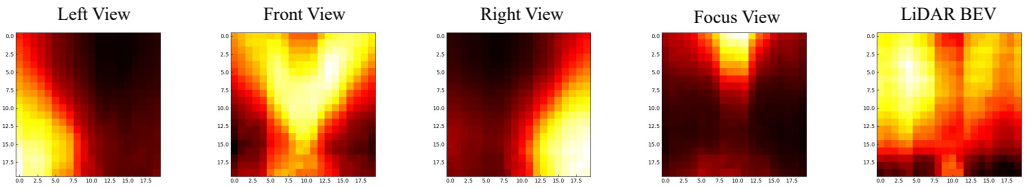

Figure 6: Visualization of attention weights between the object density map queries and features from different views.

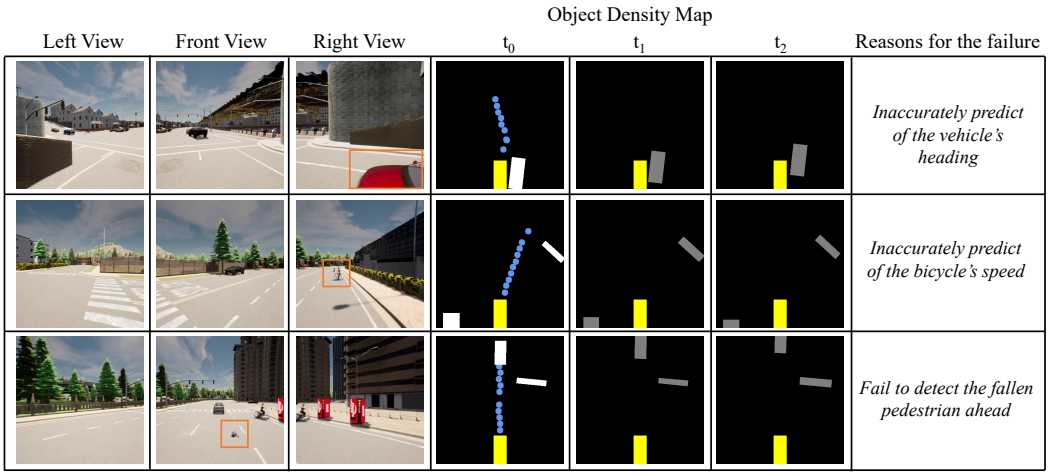

Figure 7: Visualization of failure cases with three RGB views and the predicted object density map. The orange boxes show the objects where the ego-vehicle is about to collide. t_0 of object density map denotes the predicted current traffic scenes, t_1 and t_2 denotes the predicted future traffic scenes after 1 second and 2 seconds.

signs, urban scenes and residential districts. The traffic density of each town is set to be comparable to busy traffic setting. We use the same benchmark configuration open-sourced by [8] to evaluate all methods.

E Success and Failing Cases Discussion

In Figure 5, we provided additional 4 good cases where our method can well understand the driving scene with the intermediate outputs. In Figure 7, we provided some failure cases and analyzed the failure causes. Specifically, we collected failing cases and statistically analyzed the failing conditions and causes. As a detailed statistics, 45% percent of the failing cases are due to failing in detecting objects (vehicles, bicycles, etc.); 15% percent of the failing cases are due to inaccurate detection

	Town05 Short		Town05 Long	
Method	DS ↑	RC ↑	DS ↑	RC ↑
CILRS [14]	7.47 ± 2.51	13.40 ± 1.09	3.68 ± 2.16	7.19 ± 2.95
LBC [5]	30.97 ± 4.17	55.01 ± 5.14	7.05 ± 2.13	32.09 ± 7.40
TransFuser [8]	54.52 ± 4.29	78.41 ± 3.75	33.15 ± 4.04	56.36 ± 7.14
NEAT [17]	58.70 ± 4.11	77.32 ± 4.91	37.72 ± 3.55	62.13 ± 4.66
Roach [19]	65.26 ± 3.63	88.24 ± 5.16	43.64 ± 3.95	80.37 ± 5.68
WOR [55]	64.79 ± 5.53	87.47 ± 4.68	44.80 ± 3.69	82.41 ± 5.01
InterFuser	**94.95** ± 1.91	**95.19** ± 2.57	**68.31** ± 1.86	**94.97** ± 2.87

Table 4: Comparison of our InterFuser with six state-of-the-art methods in Town05 benchmark. Metrics: driving score (DS), Road completion (RC). Our method outperformed other strong methods in all metrics and scenarios.

Method	**Driving Score ↑**	**Road Completion ↑**	**Infraction Score ↑**
CILRS [14]	22.97±0.90	35.46±0.41	0.66±0.02
LBC [5]	29.07±0.67	61.35±2.26	0.57±0.02
AIM [8]	51.25±0.17	70.04±2.31	0.73±0.03
TransFuser [8]	53.40±4.54	72.18±4.17	0.74±0.04
NEAT [17]	65.17±1.75	79.17±3.25	0.82±0.01
Roach [19]	65.08±0.99	85.16±4.20	0.77±0.02
WOR [55]	67.64±1.26	90.16±3.81	0.75±0.02
InterFuser	**91.84±2.17**	**97.12±1.95**	**0.95±0.02**

Table 5: Comparison of our InterFuser with other methods in CARLA 42 routes benchmark. Metrics: Road completion (RC), infraction score (IS), driving score (DS). Our method outperformed other strong methods in all metrics and scenarios.

results (speed, heading, etc); 15% percent of the failing cases are due to misrecognition of the traffic lights.

F Additional Experimental Results

Table 4 and Table 5 additionally compares the driving score, road completion and infraction score of the presented approach (InterFuser) to prior state-of-the-art on the CARLA Town05 benchmark [8] and CARLA 42 routes [17].

G The hyper-parameter values

The hyper-parameter values used in InterFuser are listed in Table 6. Cyclists and pedestrians are rendered larger than their actual sizes when we reconstruct the scene from the object density map, this adds extra safety when dealing with these road objects.

Notation	Description	Value
	Object Density Map and Safety-Enhanced Controller	
threshold_1	Threshold for filtering objects	0.9
threshold_2	Threshold for filtering objects	0.5
a_{max}	Maximum acceleration	1.0 m/s
v_{max}	Maximum velocity	6.5 m/s^2
R	Size of the object density map	20×20
	Size of the detected area	20 meter $\times$ 20 meter
	Scale factor for bounding box size of pedestrians and bicycles	2
	Learning Process	
	Number of epochs	35
	Number of warm-up epochs	5
λ_l	Weight for the traffic light status loss	0.2
λ_s	Weight for the stop sign loss	0.01
λ_j	Weight for the junction loss	0.1
λ_{pt}	Weight for the waypoints loss	0.4
λ_{map}	Weight for the object density map loss for GAE	0.4
λ_{tf}	Weight for the traffic information loss	1.0
	Max norm for gradient clipping	10.0
	Weight decay	0.05
	Batch size	256
	Initial learning rate for the transformer	2.5e-4
	Initial learning rate for the CNN backbone	1e-4

Table 6: The parameter used for InterFuser.