Multi-Modal Fusion Transformer for End-to-End Autonomous Driving

Aditya Prakash*[1] Kashyap Chitta*[1,2] Andreas Geiger[1,2]
[1]Max Planck Institute for Intelligent Systems, Tübingen [2]University of Tübingen
{firstname.lastname}@tue.mpg.de

Abstract

How should representations from complementary sensors be integrated for autonomous driving? Geometry-based sensor fusion has shown great promise for perception tasks such as object detection and motion forecasting. However, for the actual driving task, the global context of the 3D scene is key, e.g. a change in traffic light state can affect the behavior of a vehicle geometrically distant from that traffic light. Geometry alone may therefore be insufficient for effectively fusing representations in end-to-end driving models. In this work, we demonstrate that imitation learning policies based on existing sensor fusion methods under-perform in the presence of a high density of dynamic agents and complex scenarios, which require global contextual reasoning, such as handling traffic oncoming from multiple directions at uncontrolled intersections. Therefore, we propose TransFuser, a novel Multi-Modal Fusion Transformer, to integrate image and LiDAR representations using attention. We experimentally validate the efficacy of our approach in urban settings involving complex scenarios using the CARLA urban driving simulator. Our approach achieves state-of-the-art driving performance while reducing collisions by 76% compared to geometry-based fusion.

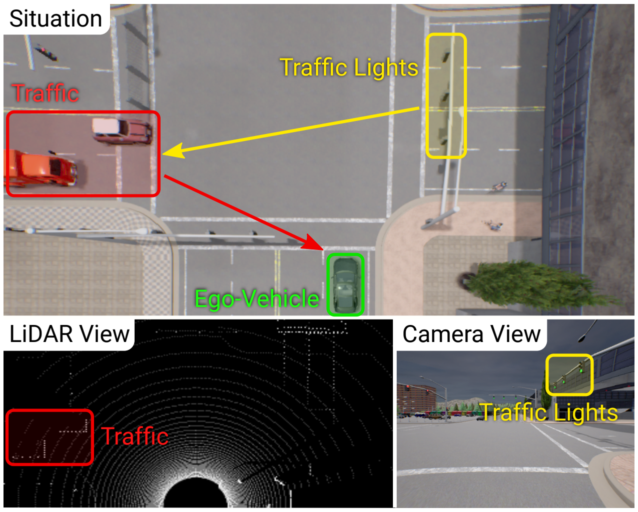


Figure 1: **Illustration.** Consider an intersection with oncoming traffic from the left. To safely navigate the intersection, the ego-vehicle (green) must capture the global context of the scene involving the interaction between the traffic light (yellow) and the vehicles (red). However, the traffic light state is not visible in the LiDAR point cloud and the vehicles are not visible in the camera view. Our TransFuser model integrates both modalities via global attention mechanisms to capture the 3D context and navigate safely.

1. Introduction

Image-only [16, 8, 41, 3, 42, 64, 53] and LiDAR-only [46, 23] methods have recently shown impressive results for end-to-end driving. However, these studies focus primarily on settings with limited dynamic agents and assume near-ideal behavior from other agents in the scene. With the introduction of *adversarial scenarios* in the recent CARLA [21] versions, e.g. vehicles running red lights, uncontrolled 4-way intersections, or pedestrians emerging from occluded regions to cross the road at random locations, image-only approaches perform unsatisfactory (Tab. 1) since they lack the 3D information of the scene required in these scenarios. While LiDAR consists of 3D information, LiDAR measurements are typically very sparse (in particular at distance), and additional sensors are required to capture information missing in LiDAR scans, e.g. traffic light states.

While most existing methods for end-to-end driving focus on a single input modality, autonomous driving systems typically come equipped with both cameras and LiDAR sensors [21, 47, 25, 59, 17, 26, 48, 1, 62]. This raises important questions: *Can we integrate representations from these two modalities to exploit their complementary advantages for autonomous driving? To what extent should we process the different modalities independently and what kind of fusion mechanism should we employ for maximum performance gain?* Prior works in the

[*] indicates equal contribution

field of sensor fusion have mostly focused on the perception aspect of driving, e.g. 2D and 3D object detection [22, 12, 66, 9, 44, 31, 34, 61, 33, 37], motion forecasting [22, 36, 5, 35, 63, 6, 19, 38, 32, 9], and depth estimation [24, 60, 61, 33]. These methods focus on learning a state representation that captures the geometric and semantic information of the 3D scene. They operate primarily based on geometric feature projections between the image space and different LiDAR projection spaces, e.g. Bird's Eye View (BEV) [22, 12, 66, 9, 44, 31, 34, 61, 33] and Range View (RV) [39, 37, 22, 38, 9, 51]. Information is typically aggregated from a local neighborhood around each feature in the projected 2D or 3D space.

While these approaches fare better than image-only methods, we observe that the locality assumption in their architecture design hampers their performance in complex urban scenarios (Tab. 1a). For example, when handling traffic at intersections, the ego-vehicle needs to account for interactions between multiple dynamic agents and traffic lights (Fig. 1). While deep convolutional networks can be used to capture global context within a single modality, it is non-trivial to extend them to multiple modalities or model interactions between pairs of features. To overcome these limitations, we use the attention mechanism of transformers [54] to integrate global contextual reasoning about the 3D scene directly into the feature extraction layers of different modalities. We consider single-view image and LiDAR inputs since they are complementary to each other and our focus is on integrating representations from different types of modalities. We call the resulting model *TransFuser* and integrate it into an auto-regressive waypoint prediction framework (Fig. 2) designed for end-to-end driving.

Contributions: (1) We demonstrate that imitation learning policies based on existing sensor fusion approaches are unable to handle adversarial scenarios in urban driving, e.g., unprotected turnings at intersections or pedestrians emerging from occluded regions. (2) We propose a novel Multi-Modal Fusion Transformer (TransFuser) to incorporate the global context of the 3D scene into the feature extraction layers of different modalities. (3) We experimentally validate our approach in complex urban settings involving adversarial scenarios in CARLA and achieve state-of-the-art performance. Our code and trained models are available at https://github.com/autonomousvision/transfuser.

2. Related Work

Multi-Modal Autonomous Driving: Recent multi-modal methods for end-to-end driving [58, 65, 51, 3] have shown that complementing RGB images with depth and semantics has the potential to improve driving performance. Xiao et al. [58] explore RGBD input from the perspective of early, mid and late fusion of camera and depth modalities and observe significant gains. Behl et al. [3] and Zhou et al. [65] demonstrate the effectiveness of semantics and depth as explicit intermediate representations for driving. In this work, we focus on image and LiDAR inputs since they are complementary to each other in terms of representing the scene and are readily available in autonomous driving systems. In this respect, Sobh et al. [51] exploit a late fusion architecture for LiDAR and image modalities where each input is encoded in a separate stream and then concatenated together. However, we observe that this fusion mechanism suffers from high infraction rates in complex urban scenarios (Tab. 1b) due to its inability to account for the behavior of multiple dynamic agents. Therefore, we propose a novel Multi-Modal Fusion Transformer that is effective in integrating information from different modalities at multiple stages during feature encoding and hence improves upon the limitations of the late fusion approach.

Sensor Fusion Methods for Object Detection and Motion Forecasting: The majority of the sensor fusion works consider perception tasks, e.g. object detection [22, 12, 66, 7, 44, 31, 34, 61, 33, 37] and motion forecasting [36, 5, 35, 63, 6, 19, 38]. They operate on multi-view LiDAR, e.g. Bird's Eye View (BEV) and Range View (RV), or complement the camera input with depth information from LiDAR by projecting LiDAR features into the image space or projecting image features into the BEV or RV space. The closest approach to ours is ContFuse [34] which performs multi-scale dense feature fusion between image and LiDAR BEV features. For each pixel in the LiDAR BEV representation, it computes the nearest neighbors in a local neighborhood in 3D space, projects these neighboring points into the image space to obtain the corresponding image features, aggregates these features using continuous convolutions, and combines them with the LiDAR BEV features. Other projection-based fusion methods follow a similar trend and aggregate information from a local neighborhood in 2D or 3D space. However, the state representation learned by these methods is insufficient since they do not capture the global context of the 3D scene which is important for safe maneuvers in adversarial scenarios. To demonstrate this, we implement a multi-scale geometry-based fusion mechanism, inspired by [34, 33], involving both image-to-LiDAR and LiDAR-to-image feature fusion for end-to-end driving in CARLA and observe high infraction rates in the complex urban setting (Tab. 1b). To overcome this limitation, we propose an attention-based Multi-Modal Fusion Transformer that incorporates global contextual reasoning and achieves superior driving performance.

Attention for Autonomous Driving: Attention has been explored in the context of driving for lane changing [13], object detection [11, 32] and motion forecasting [32, 50, 49, 28, 15, 30, 29, 56]. Chen et al. [11] employ a recurrent attention mechanism over a learned semantic map for predicting vehicle controls. Li et al. [32] utilize attention to capture temporal and spatial dependencies between actors by incorporating a transformer module into a recurrent neural network. SA-NMP [56] is a concurrent work that learns an attention mask over features extracted from a 2D CNN, operating on LiDAR BEV projections and HD maps, to focus on dynamic agents for safe motion planning. Chen et al. [13] utilize attention in a hierarchical deep reinforcement learning framework to focus on the surrounding vehicles for lane changing in the TORCS racing simulator. They incorporate a spatial attention module to detect the most relevant regions in the image and a temporal attention module to weight different time-step image inputs, which leads to smoother lane changes. However, none of these approaches considers multiple modalities or encodes the global context of the 3D scene which is necessary for safely navigating adversarial scenarios. In contrast, we demonstrate the effectiveness of attention for feature fusion between different modalities on challenging urban driving scenarios.

3. Method

In this work, we propose an architecture for end-to-end driving (Fig. 2) with two main components: (1) a Multi-Modal Fusion Transformer for integrating information from multiple modalities (single-view image and LiDAR), and (2) an auto-regressive waypoint prediction network. The following sections detail our problem setting, input and output parameterizations, and each component of the model.

3.1. Problem Setting

We consider the task of point-to-point navigation in an urban setting [23, 45, 46, 8, 16] where the goal is to complete a given route while safely reacting to other dynamic agents and following traffic rules.

Imitation Learning (IL): The goal of IL is to learn a policy π that imitates the behavior of an expert π^*. In our setup, a policy is a mapping from inputs to waypoints that are provided to a separate low-level controller to output actions. We consider the Behavior Cloning (BC) approach of IL which is a supervised learning method. An expert policy is first rolled out in the environment to collect a dataset, $\mathcal{D} = \{(\mathcal{X}^i, \mathcal{W}^i)\}_{i=1}^Z$ of size Z, which consists of high-dimensional observations of the environment, $\mathcal{X}$, and the corresponding expert trajectory, defined by a set of 2D waypoints in BEV space, i.e., $\mathcal{W} = \{\mathbf{w}_t = (x_t, y_t)\}_{t=1}^T$. This BEV space uses the coordinate frame of the ego-vehicle. The policy, π, is trained in a supervised manner using the collected data, $\mathcal{D}$, with the loss function, $\mathcal{L}$.

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

The high-dimensional observation, $\mathcal{X}$, includes a front camera image input and a LiDAR point cloud from a single time-step. We use a single time-step input since prior works on IL for autonomous driving have shown that using observation histories may not lead to performance gain [40, 55, 2, 57]. We use the L_1 distance between the predicted trajectory, $\pi(\mathcal{X})$, and the expert trajectory, $\mathcal{W}$, as the loss function. We assume access to an inverse dynamics model [4], implemented as a PID Controller $\mathbb{I}$, which performs the low-level control, i.e., steer, throttle, and brake, provided the future trajectory $\mathcal{W}$. The actions are determined as $\mathbf{a} = \mathbb{I}(\mathcal{W})$.

Global Planner: We follow the standard protocol of CARLA 0.9.10 and assume that high-level goal locations $\mathcal{G}$ are provided as GPS coordinates. Note that these goal locations are sparse and can be hundreds of meters apart as opposed to the local waypoints predicted by the policy π.

3.2. Input and Output Parameterization

Input Representation: Following [45, 23], we convert the LiDAR point cloud into a 2-bin histogram over a 2D BEV grid with a fixed resolution. We consider the points within 32m in front of the ego-vehicle and 16m to each of the sides, thereby encompassing a BEV grid of 32m × 32m. We divide the grid into blocks of 0.125m × 0.125m which results in a resolution of 256 × 256 pixels. For the histogram, we discretize the height dimension into 2 bins representing the points on/below and above the ground plane. This results in a two-channel pseudo-image of size 256 × 256 pixels. For the RGB input, we consider the front camera with a FOV of 100°. We extract the front image at a resolution of 400 × 300 pixels which we crop to 256 × 256 to remove radial distortion at the edges.

Output Representation: We predict the future trajectory $\mathcal{W}$ of the ego-vehicle in BEV space, centered at the current coordinate frame of the ego-vehicle. The trajectory is represented by a sequence of 2D waypoints, $\{\mathbf{w}_t = (x_t, y_t)\}_{t=1}^T$. We use $T = 4$, which is the default number of waypoints required by our inverse dynamics model.

3.3. Multi-Modal Fusion Transformer

Our key idea is to exploit the self-attention mechanism of transformers [54] to incorporate the global context for image and LiDAR modalities given their complementary nature. The transformer architecture takes as input a sequence consisting of discrete tokens, each represented by a feature vector. The feature vector is supplemented by a positional encoding to incorporate positional inductive biases.

Formally, we denote the input sequence as $\mathbf{F}^{in} \in \mathbb{R}^{N\times D_f}$, where N is the number of tokens in the sequence and each token is represented by a feature vector of dimensionality D_f. The transformer uses linear projections for computing a set of queries, keys and values ($\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$),

$$\mathbf{Q} = \mathbf{F}^{in}\mathbf{M}^q, \;\; \mathbf{K} = \mathbf{F}^{in}\mathbf{M}^k, \;\; \mathbf{V} = \mathbf{F}^{in}\mathbf{M}^v \tag{2}$$

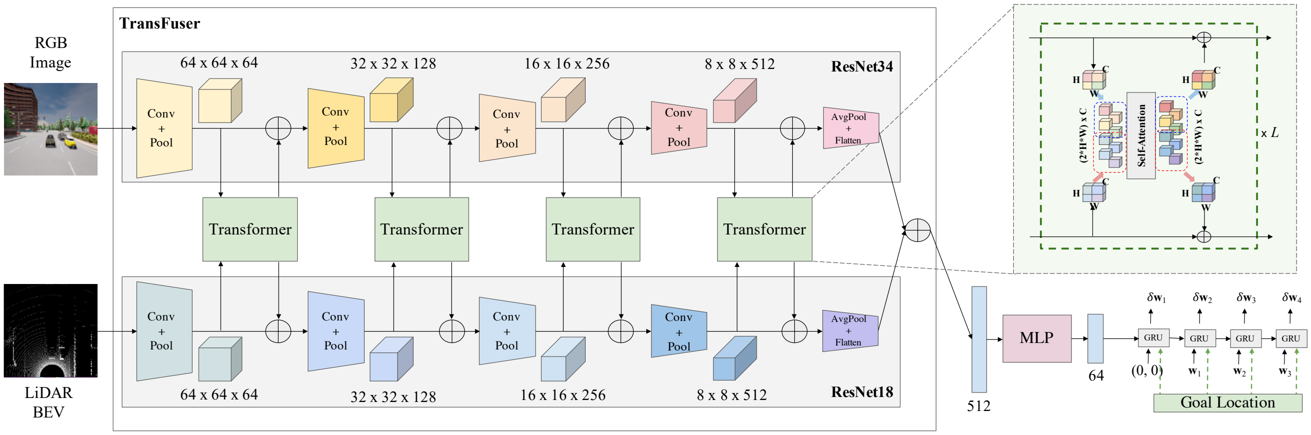


Figure 2: **Architecture.** We consider single-view RGB image and LiDAR BEV representations (Sec. 3.2) as inputs to our Multi-Modal Fusion Transformer (TransFuser) which uses several transformer modules for the fusion of intermediate feature maps between both modalities. This fusion is applied at multiple resolutions (64×64, 32×32, 16×16 and 8×8) throughout the feature extractor resulting in a 512-dimensional feature vector output from both the image and LiDAR BEV stream, which is combined via element-wise summation. This 512-dimensional feature vector constitutes a compact representation of the environment that encodes the global context of the 3D scene. It is then processed with an MLP before passing it to an auto-regressive waypoint prediction network. We use a single layer GRU followed by a linear layer which takes in the hidden state and predicts the differential ego-vehicle waypoints $\{\delta\mathbf{w}_t\}_{t=1}^{T}$, represented in the ego-vehicle's current coordinate frame.

where $\mathbf{M}^q \in \mathbb{R}^{D_f \times D_q}$, $\mathbf{M}^k \in \mathbb{R}^{D_f \times D_k}$ and $\mathbf{M}^v \in \mathbb{R}^{D_f \times D_v}$ are weight matrices. It uses the scaled dot products between $\mathbf{Q}$ and $\mathbf{K}$ to compute the attention weights and then aggregates the values for each query,

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_k}}\right)\mathbf{V} \tag{3}$$

Finally, the transformer uses a non-linear transformation to calculate the output features, $\mathbf{F}^{out}$ which are of the same shape as the input features, $\mathbf{F}^{in}$.

$$\mathbf{F}^{out} = \text{MLP}(\mathbf{A}) + \mathbf{F}^{in} \tag{4}$$

The transformer applies the attention mechanism multiple times throughout the architecture resulting in L attention layers. Each layer in a standard transformer has multiple parallel attention 'heads', which involve generating several $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ values per $\mathbf{F}^{in}$ for Eq. (2) and concatenating the resulting values of $\mathbf{A}$ from Eq. (3).

Unlike the token input structures in NLP, we operate on grid structured feature maps. Similar to prior works on the application of transformers to images [52, 10, 43, 20], we consider the intermediate feature maps of each modality to be a set rather than a spatial grid and treat each element of the set as a token. The convolutional feature extractors for the image and LiDAR BEV inputs encode different aspects of the scene at different layers. Therefore, we fuse these features at multiple scales (Fig. 2) throughout the encoder.

Let the intermediate grid structured feature map of a single modality be a 3D tensor of dimension $H \times W \times C$. For S different modalities, these features are stacked together to form a sequence of dimension $(S * H * W) \times C$. We add a learnable positional embedding, which is a trainable parameter of dimension $(S * H * W) \times C$, so that the network can infer spatial dependencies between different tokens at train time. We also provide the current velocity as input by projecting the scalar value into a C dimensional vector using a linear layer. The input sequence, positional embedding, and velocity embedding are combined using element-wise summation to form a tensor of dimension $(S * H * W) \times C$. As shown in Fig. 2, this tensor is fed as input to the transformer which produces an output of the same dimension. We have omitted the positional embedding and velocity embedding inputs in Fig. 2 for clarity. The output is then reshaped into S feature maps of dimension $H \times W \times C$ each and fed back into each of the individual modality branches using an element-wise summation with the existing feature maps. The mechanism described above constitutes feature fusion at a single scale. This fusion is applied *multiple times* throughout the ResNet feature extractors of the image and LiDAR BEV branches at different resolutions (Fig. 2). However, processing feature maps at high spatial resolutions is computationally expensive. Therefore, we downsample higher resolution feature maps from the early encoder blocks using average pooling to a fixed resolution of $H = W = 8$ before passing them as inputs to the transformer and upsample the output to the original resolution using bilinear interpolation before element-wise summation with the existing feature maps.

After carrying out dense feature fusion at multiple resolutions (Fig. 2), we obtain a feature map of dimension $8 \times 8 \times 512$ from the feature extractors of each modality for an input of resolution 256×256 pixels. These feature maps are reduced to a dimension of $1 \times 1 \times 512$ by average pooling and flattened to a 512-dimensional feature vector. The feature vector of dimension 512 from both the image and the LiDAR BEV streams are then combined via element-wise summation. This 512-dimensional feature vector constitutes a compact representation of the environment that encodes the global context of the 3D scene. This is then fed to the waypoint prediction network which we describe next.

3.4. Waypoint Prediction Network

As shown in Fig. 2, we pass the 512-dimensional feature vector through an MLP (comprising 2 hidden layers with 256 and 128 units) to reduce its dimensionality to 64 for computational efficiency before passing it to the auto-regressive waypoint network implemented using GRUs [14]. We initialize the hidden state of the GRU with the 64-dimensional feature vector. The update gate of the GRU controls the flow of information encoded in the hidden state to the output and the next time-step. It also takes in the current position and the goal location (Sec. 3.1) as input, which allows the network to focus on the relevant context in the hidden state for predicting the next waypoint. We provide the GPS coordinates of the goal location (registered to the ego-vehicle coordinate frame) as input to the GRU rather than the encoder since it lies in the same BEV space as the predicted waypoints and correlates better with them compared to representing the goal location in the perspective image domain [8]. Following [23], we use a single layer GRU followed by a linear layer which takes in the hidden state and predicts the differential ego-vehicle waypoints $\{\delta \mathbf{w}_t\}_{t=1}^{T}$ for $T = 4$ future time-steps in the ego-vehicle current coordinate frame. Therefore, the predicted future waypoints are given by $\{\mathbf{w}_t = \mathbf{w}_{t-1} + \delta \mathbf{w}_t\}_{t=1}^{T}$. The input to the first GRU unit is given as (0,0) since the BEV space is centered at the ego-vehicle's position.

Controller: We use two PID controllers for lateral and longitudinal control to obtain steer, throttle and brake values from the predicted waypoints, $\{\mathbf{w}_t\}_{t=1}^{T}$. The longitudinal controller takes in the magnitude of a weighted average of the vectors between waypoints of consecutive time-steps whereas the lateral controller takes in their orientation. For the PID controllers, we use the same configuration as in the author-provided codebase of [8]. Implementation details can be found in the supplementary.

3.5. Loss Function

Following [8], we train the network using an L_1 loss between the predicted waypoints and the ground truth waypoints (from the expert), registered to the current coordinate frame. Let $\mathbf{w}_t^{gt}$ represent the ground truth waypoint for time-step t, then the loss function is given by:

$$\mathcal{L} = \sum_{t=1}^{T} ||\mathbf{w}_t - \mathbf{w}_t^{gt}||_1 \tag{5}$$

Note that the ground truth waypoints $\{\mathbf{w}_t^{gt}\}$ which are available only at training time are different from the sparse goal locations $\mathcal{G}$ provided at both training and test time.

4. Experiments

In this section, we describe our experimental setup, compare the **driving performance** of our approach against several baselines, conduct an **infraction analysis** to study different failure cases, **visualize** the attention maps of TransFuser and present an **ablation study** to highlight the importance of different components of our model.

Task: We consider the task of navigation along a set of predefined routes in a variety of areas, e.g. freeways, urban areas, and residential districts. The routes are defined by a sequence of sparse goal locations in GPS coordinates provided by a global planner and the corresponding discrete navigational commands, e.g. follow lane, turn left/right, change lane. Our approach uses only the sparse GPS locations to drive. Each route consists of several scenarios, initialized at predefined positions, which test the ability of the agent to handle different kinds of adversarial situations, e.g. obstacle avoidance, unprotected turns at intersections, vehicles running red lights, and pedestrians emerging from occluded regions to cross the road at random locations. The agent needs to complete the route within a specified time limit while following traffic regulations and coping with high densities of dynamic agents.

Dataset: We use the CARLA [21] simulator for training and testing, specifically CARLA 0.9.10 which consists of 8 publicly available towns. We use 7 towns for training and hold out Town05 for evaluation. For generating training data, we roll out an expert policy designed to drive using privileged information from the simulation and store data at 2FPS. Please refer to the supplementary material for additional details. We select Town05 for evaluation due to the large diversity in drivable regions compared to other CARLA towns, e.g. multi-lane and single-lane roads, highways and exits, bridges and underpasses. We consider two evaluation settings: (1) Town05 Short: 10 short routes of 100-500m comprising 3 intersections each, (2) Town05 Long: 10 long routes of 1000-2000m comprising 10 intersections each. Each route consists of a high density of dynamic agents and adversarial scenarios which are spawned at predefined positions along the route. Since we focus on handling dynamic agents and adversarial scenarios, we decouple this aspect from generalization across weather conditions and evaluate only on ClearNoon weather.

Metrics: We report results on 3 metrics. (1) **Route Completion (RC)**, percentage of route distance completed, (2) **Driving Score (DS)**, which is route completion weighted by an infraction multiplier that accounts for collisions with pedestrians, vehicles, and static elements, route deviations, lane infractions, running red lights, and running stop signs, and (3) **Infraction Count**. Additional details regarding the metrics and infractions are provided in the supplementary.

Baselines: We compare our TransFuser model to several baselines. (1) **CILRS** [16] is a conditional imitation learning method in which the agent learns to predict vehicle controls from a single front camera image while being conditioned on the navigational command. We closely follow the author-provided code and reimplement CILRS for CARLA 0.9.10 to account for the additional navigational commands compared to CARLA 0.8.4. (2) **LBC** [8] is a knowledge distillation approach where a teacher model with access to ground truth BEV semantic maps is first trained using expert supervision to predict future waypoints followed by an image-based student model which is trained using supervision from the teacher. It is the current state-of-the-art approach on CARLA 0.9.6. We use the latest author-provided codebase for training on CARLA 0.9.10, which combines 3 input camera views by stacking different viewpoints as channels. (3) Auto-regressive IMage-based waypoint prediction **(AIM)**: We implement our auto-regressive waypoint prediction network with an image-based ResNet-34 encoder which takes just the front camera image as input. This baseline is equivalent to adapting the CILRS model to predict waypoints conditioned on sparse goal locations rather than vehicle controls conditioned on navigational commands. The image encoder used for this is the same as CILRS and our model. (4) **Late Fusion**: We implement a version of our architecture where the image and the LiDAR features are extracted independent of each other using the same encoders as TransFuser but without the transformers (similar to [51]), which are then fused through element-wise summation and passed to the waypoint prediction network. (5) **Geometric Fusion**: We implement a multi-scale geometry-based fusion method, inspired by [34, 33], involving both image-to-LiDAR and LiDAR-to-image feature fusion. We unproject each 0.125m × 0.125m block in our LiDAR BEV representation into 3D space resulting in a 3D volume. We randomly select 5 points from the LiDAR point cloud lying in this 3D volume and project them into the image space. We aggregate the image features of these points via element-wise summation before passing them to a 3-layer MLP. The output of the MLP is then combined with the LiDAR BEV feature of the corresponding 0.125m × 0.125m block at multiple resolutions throughout the feature extractor. Similarly, for each image pixel, we aggregate information from the LiDAR BEV features at multiple resolutions. This baseline is equivalent to replacing the transformers in our architecture with projection-based feature fusion.

We also report results for the expert used for generating our training data, which defines an upper bound for the performance on each evaluation setting. We provide additional details regarding all the baselines in the supplementary.

Implementation Details: We use 2 sensor modalities, the front camera RGB image and LiDAR point cloud converted to BEV representation (Sec. 3.2), i.e., $S = 2$. The RGB image is encoded using a ResNet-34 [27] which is pre-trained on ImageNet [18]. The LiDAR BEV representation is encoded using a ResNet-18 [27] which is trained from scratch. In our default TransFuser configuration, we use 1 transformer per resolution and 4 attention heads for each transformer. We select D_q, D_k, D_v from $\{64, 128, 256, 512\}$ for the 4 transformers corresponding to the feature embedding dimension D_f at each resolution. For each of our baselines, we tested different perception backbone and chose the best: ResNet-34 for CILRS and AIM, ResNet-50 for LBC, ResNet-34 as the image encoder and ResNet-18 as the LiDAR BEV encoder for each of the sensor fusion methods. Additional details can be found in the supplementary.

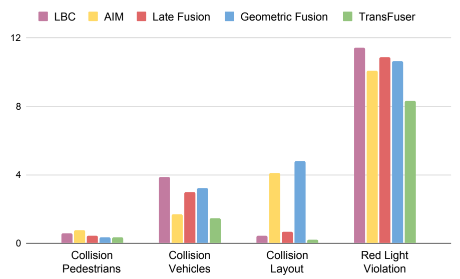
4.1. Results

Performance of CILRS and LBC: In our first experiment, we examine to what extent the current image-based methods on CARLA scale to the new 0.9.10 evaluation setting involving complex multi-lane intersections, adversarial scenarios, and heavy infraction penalties. From the results in Tab. 1a we observe that CILRS performs poorly on all evaluation settings. This is not surprising since CILRS is conditioned on discrete navigational commands whose data distribution is imbalanced as shown in the supplementary. While the original LBC [8] architecture uses only the front camera image as input, the authors recently released an updated version of their architecture with 2 major modifications, (1) multi-view camera inputs (front, 45° left, and 45° right), (2) target heatmap as input (instead of navigational command) which is formed by projecting the sparse goal location in the image space. We train their updated model on our data and observe that LBC performs significantly better than CILRS on the short routes (Tab. 1a), which is expected since it is trained using supervision from the teacher model which uses ground truth BEV semantic labels. However, LBC's performance drops drastically when evaluated on the long routes where it achieves 32.09 RC but suffers multiple infractions resulting in a low DS of 7.05. This is due to the frequent red light infractions and collision with vehicles (Tab. 1b) resulting in large multiplicative penalties on the DS. These results show that CILRS and LBC are unable to handle the complexities of urban driving.

AIM is a strong baseline: Since the performance of CILRS and LBC drops significantly on the long routes, we focus on

Method	Town05 Short		Town05 Long	
	DS ↑	RC ↑	DS ↑	RC ↑
CILRS [16]	7.47 ± 2.51	13.40 ± 1.09	3.68 ± 2.16	7.19 ± 2.95
LBC [8]	30.97 ± 4.17	55.01 ± 5.14	7.05 ± 2.13	32.09 ± 7.40
AIM	49.00 ± 6.83	81.07 ± 15.59	26.50 ± 4.82	60.66 ± 7.66
Late Fusion	51.56 ± 5.24	83.66 ± 11.04	31.30 ± 5.53	68.05 ± 5.39
Geometric Fusion	54.32 ± 4.85	**86.91** ± 10.85	25.30 ± 4.08	**69.17** ± 11.07
TransFuser (Ours)	**54.52** ± 4.29	78.41 ± 3.75	**33.15** ± 4.04	56.36 ± 7.14
Expert	*84.67 ± 6.21*	*98.59 ± 2.17*	*38.60 ± 4.00*	*77.47 ± 1.86*

(a) **Driving Performance.** We report the mean and standard deviation over 9 runs of each method (3 training seeds, each seed evaluated 3 times) on 2 metrics: Route Completion (RC) and Driving Score (DS), in Town05 Short and Town05 Long settings comprising high densities of dynamic agents and scenarios.



(b) **Infractions.** We report the mean value of the total infractions incurred by each model over the 9 evaluation runs in the Town05 Short setting.

Table 1: **Results.** We compare our TransFuser model with CILRS, LBC, auto-regressive image-based waypoint prediction network (AIM), and sensor fusion methods (Late Fusion of image and LiDAR features, Geometric feature projections between image and LiDAR BEV space) in terms of driving performance (Tab. 1a) and infractions incurred (Tab. 1b).

designing a strong image-based baseline next. Towards this goal, we replace the learned controller of CILRS with our auto-regressive waypoint prediction network. We observe that AIM significantly outperforms CILRS on all evaluation settings (Tab. 1a), achieving nearly 7 times better performance. This is likely because AIM uses our inverse dynamics model (PID controller) for low-level control and represents goal locations in the same BEV coordinate space in which waypoints are predicted. In contrast, LBC's goal locations are represented as heatmaps in image space. Furthermore, AIM uses an auto-regressive GRU-based waypoint prediction network which enables the processing of these goal locations directly at the final stage of the network. This provides a prior that simplifies the learning of behaviors that follow the path to the goal location which could make the encoder prioritize information regarding high-level semantics of the scene, e.g. traffic light state, rather than features relevant for low-level control. AIM outperforms LBC by 58.21% on the short routes and 275.89% on the long routes. The red light violations of LBC lead to a compounding of other infractions (e.g. collisions with vehicles), which rapidly drops its DS compared to AIM.

Sensor Fusion Methods: The goal of this experiment is to determine the impact of the LiDAR modality on the driving performance and compare different fusion methods. For this, we compare our TransFuser to two baselines, Late Fusion (LF) and Geometric Fusion (GF). We observe that LF outperforms AIM on all evaluation settings (Tab. 1a). This is expected since LiDAR provides additional 3D context which helps the agent to better navigate urban environments. Furthermore, we observe even better performance on the short routes when replacing the independent feature extractors of image and LiDAR branches with multi-scale geometry-based fusion encoder. However, both LF and GF suffer from a sharp drop in DS compared to their RC. We hypothesize that this occurs because they do not incorporate global contextual reasoning which is necessary to safely navigate the intersections, and focus primarily on navigation to the goal at all costs while ignoring obstacles which leads to several infractions Tab. 1b. This has a compounding effect on the long routes due to the exponential nature of the infraction penalty, resulting in a rapid drop in DS. In contrast, our TransFuser model outperforms GF by 31.02% on DS with an 18.52% lower RC on Town05 Long. It also achieves a 51.58% reduction compared to LF and 76.11% reduction compared to GF in collisions and 23.5% reduction compared to LF and 21.93% reduction compared to GF in red light violations. This shows that our model drives cautiously and focuses on dynamic agents and traffic lights. This indicates that attention is effective in incorporating the global context of the 3D scene which allows for safe driving. We provide driving videos in the supplementary.

Limitations: We observe that all fusion methods struggle with red light violations (Tab. 1b). This is because detecting red lights is very challenging in Town05 since they are located on the opposite side of the intersection and are barely visible in the input image. Unlike some existing methods [53], we do not use any semantic supervision for red lights which furthers exacerbates this issue since the learning signal for red light detection is very weak. We expect the red light detection performance of the fusion approaches to improve when incorporating such additional supervision.

4.2. Attention Map Visualizations

The transformer takes in 64 image feature tokens and 64 LiDAR feature tokens as input where each token corresponds to a 32 × 32 patch in the input modality. We consider 1000 frames from Town05 intersections and examine the top-5 attention weights for the 24 tokens in the 2^{nd}, 3^{rd} and 4^{th} rows of the image feature map and the 24 tokens in the 4^{th}, 5^{th} and 6^{th} rows of the LiDAR feature map. We select these tokens since they correspond to the intersection

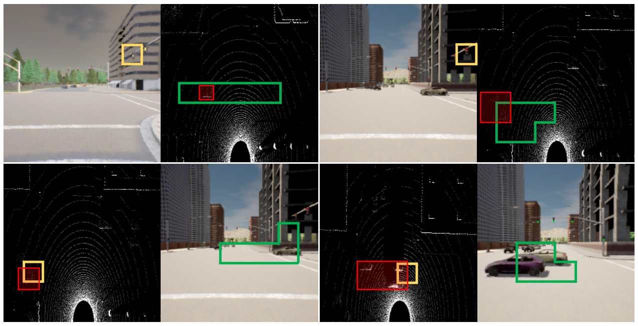


Figure 3: **Attention Maps.** For the yellow query token, we show the top-5 attended tokens in green and highlight the presence of vehicles in the LiDAR point cloud in red. TransFuser attends to the vehicles and traffic lights at intersections, albeit at a slightly different location.

region in the input modality and contain traffic lights and vehicles. We observe that for 62.75% of the image tokens, all the top-5 attended tokens belong to the LiDAR and for 88.87%, at least one token in the top-5 attended tokens belong to the LiDAR. Similarly, for 78.45% of the LiDAR tokens, all the top-5 attended tokens belong to the image and for 98.95%, at least one token in the top-5 attended tokens belong to the image. This indicates that TransFuser is effective in aggregating information from image and LiDAR. We show four such frames in Fig. 3. We observe a common trend in attention maps: TransFuser attends to the vehicles and traffic lights at intersections, albeit at a slightly different location in the image and LiDAR feature maps. Additional visualizations are provided in the supplementary.

4.3. Ablation Study

In our default configuration, we use 1 transformer per resolution, 8 attention layers and 4 attention heads for each transformer and carry out fusion at 4 resolutions. In this experiment, we present ablations on number of scales, attention layers, shared or separate transformers and posi- tional embedding, in the Town05 Short evaluation setting.

Is multi-scale fusion essential? We show results on scales 1 to 4 where 1 indicates fusion at a resolution of 8×8 in the last ResNet layer, 2 indicates fusion at 8×8 and 16×16 in the last and the penultimate ResNet layers respectively and similarly for scales 3 and 4. We observe an overall degradation in performance when reducing the number of scales from 4 to 1 (Tab. 2). This happens because different convolutional layers in ResNet learn different types of features regarding the input, therefore, multi-scale fusion is effective in integrating these features from different modalities.

Are multiple transformers necessary? We test a version of our model which uses shared parameters for the transformers (Shared Transformer in Tab. 2) and observe a significant drop in DS. This is intuitive since different convolutional layers in ResNet learn different types of features due to which each transformer has to focus on fusing different types of features at each resolution.

Parameter	Value	DS ↑	RC ↑
Scale	1	41.94	56.09
	2	52.82	74.70
	3	52.41	71.40
Shared Transformer	-	55.36	77.54
Attention layers	1	50.46	**96.53**
	4	51.38	79.35
No Pos. Embd	-	52.45	93.64
Default Config	-	**59.99**	74.86

Table 2: **Ablation Study.** We report the DS on Town05 Short setting for different TransFuser configurations.

Are multiple attention layers required? We report results for 1-layer and 4-layer variants of our TransFuser in Tab. 2. We observe that while the 1-layer variant has a very high RC, its DS is significantly lower. However, when we increase the number of attention layers to 4, the model can sustain its DS even with an 18% lower RC. This indicates that the model becomes more cautious with additional attention layers. As we further increase L to 8 in the default configuration, DS also increases. This shows that multiple attention layers lead to cautious driving agents.

Is the positional embedding useful? Intuitively, we expect the learnable positional embedding to help since modeling spatial dependencies between dynamic agents is crucial for safe driving. This is indeed apparent in Tab. 2 where we observe a significant drop in DS in the absence of positional embedding even though RC increases by 25%.

5. Conclusion

In this work, we demonstrate that IL policies based on existing sensor fusion methods suffer from high infraction rates in complex driving scenarios. To overcome this limitation, we present a novel Multi-Modal Fusion Transformer (TransFuser) for integrating representations of different modalities. The TransFuser uses attention to capture the global 3D scene context and focuses on dynamic agents and traffic lights, resulting in state-of-the-art performance on CARLA. Given that our method is flexible and generic, it would be interesting to explore it further with additional sensors, e.g. radar, or apply it to other embodied AI tasks.

Acknowledgements: This work was supported by the BMWi in the project KI Delta Learning (project number: 19A19013O) and the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039B. Andreas Geiger was supported by the ERC Starting Grant LEGO-3D (850533) and the DFG EXC number 2064/1 - project number 390727645. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Kashyap Chitta.

References

[1] Waymo open dataset: An autonomous driving dataset. https://www.waymo.com/open, 2019.

[2] Mayank Bansal, Alex Krizhevsky, and Abhijit S. Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. In *Proc. Robotics: Science and Systems (RSS)*, 2019.

[3] Aseem Behl, Kashyap Chitta, Aditya Prakash, Eshed Ohn-Bar, and Andreas Geiger. Label efficient visual abstractions for autonomous driving. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[4] Richard Bellman. *Adaptive Control Processes - A Guided Tour*, volume 2045. Princeton University Press, 2015.

[5] Sergio Casas, Cole Gulino, Renjie Liao, and Raquel Urtasun. Spagnn: Spatially-aware graph neural networks for relational behavior forecasting from sensor data. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2020.

[6] Sergio Casas, Wenjie Luo, and Raquel Urtasun. Intentnet: Learning to predict intention from raw sensor data. In *Proc. Conf. on Robot Learning (CoRL)*, 2018.

[7] Can Chen, Luca Zanotti Fragonara, and Antonios Tsourdos. Roifusion: 3d object detection from lidar and vision. *arXiv.org*, 2009.04554, 2020.

[8] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Proc. Conf. on Robot Learning (CoRL)*, 2019.

[9] Ke Chen, Ryan Oldja, Nikolai Smolyanskiy, Stan Birchfield, Alexander Popov, David Wehr, Ibrahim Eden, and Joachim Pehserl. Mvlidarnet: Real-time multi-class scene understanding for autonomous driving using multiple views. *arXiv.org*, 2006.05518, 2020.

[10] Mark Chen, A. Radford, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *Proc. of the International Conf. on Machine learning (ICML)*, 2020.

[11] Shi-tao Chen, Songyi Zhang, Jinghao Shang, Badong Chen, and Nanning Zheng. Brain inspired cognitive model with attention for self-driving cars. *arXiv.org*, 1702.05596, 2017.

[12] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[13] Yilun Chen, Chiyu Dong, Praveen Palanisamy, Priyantha Mudalige, Katharina Muelling, and John M. Dolan. Attention-based hierarchical deep reinforcement learning for lane change behaviors in autonomous driving. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[14] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[15] Chiho Choi and Behzad Dariush. Looking to relations for future trajectory forecast. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019.

[16] Felipe Codevilla, Eder Santana, Antonio M. López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019.

[17] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[18] Jia Deng, Wei Dong, Richard Socher, Li jia Li, Kai Li, and Li Fei-fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[19] Nemanja Djuric, Henggang Cui, Zhaoen Su, Shangxuan Wu, Huahua Wang, Fang-Chieh Chou, Luisa San Martin, Song Feng, Rui Hu, Yang Xu, Alyssa Dayan, Sidney Zhang, Brian C. Becker, Gregory P. Meyer, Carlos Vallespi-Gonzalez, and Carl K. Wellington. Multixnet: Multiclass multistage multimodal motion prediction. *arXiv.org*, 2006.02000, 2020.

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv.org*, 2010.11929, 2020.

[21] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proc. Conf. on Robot Learning (CoRL)*, 2017.

[22] Sudeep Fadadu, Shreyash Pandey, Darshan Hegde, Yi Shi, Fang-Chieh Chou, Nemanja Djuric, and Carlos Vallespi-Gonzalez. Multi-view fusion of sensor data for improved perception and prediction in autonomous driving. *arXiv.org*, 2008.11901, 2020.

[23] Angelos Filos, Panagiotis Tigas, Rowan McAllister, Nicholas Rhinehart, Sergey Levine, and Yarin Gal. Can autonomous vehicles identify, recover from, and adapt to distribution shifts? In *Proc. of the International Conf. on Machine learning (ICML)*, 2020.

[24] Chen Fu, Chiyu Dong, Christoph Mertz, and John M. Dolan. Depth completion via inductive fusion of planar LIDAR and monocular camera. *arXiv.org*, 2009.01875, 2020.

[25] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[26] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[28] Ying-Fan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019.

[29] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019.

[30] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian D. Reid, Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[31] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L. Waslander. Joint 3d proposal generation and object detection from view aggregation. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[32] 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 *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[33] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[34] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018.

[35] 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 *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[36] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[37] Gregory P. Meyer, Jake Charland, Darshan Hegde, Ankit Laddha, and Carlos Vallespi-Gonzalez. Sensor fusion for joint 3d object detection and semantic segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019.

[38] Gregory P. Meyer, Jake Charland, Shreyash Pandey, Ankit Laddha, Carlos Vallespi-Gonzalez, and Carl K. Wellington. Laserflow: Efficient and probabilistic object detection and motion forecasting. *arXiv.org*, 2003.05982, 2020.

[39] Gregory P. Meyer, Ankit Laddha, Eric Kee, Carlos Vallespi-Gonzalez, and Carl K. Wellington. Lasernet: An efficient probabilistic 3d object detector for autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[40] Urs Muller, Jan Ben, Eric Cosatto, Beat Flepp, and Yann LeCun. Off-road obstacle avoidance through end-to-end learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2005.

[41] Eshed Ohn-Bar, Aditya Prakash, Aseem Behl, Kashyap Chitta, and Andreas Geiger. Learning situational driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[42] Aditya Prakash, Aseem Behl, Eshed Ohn-Bar, Kashyap Chitta, and Andreas Geiger. Exploring data aggregation in policy learning for vision-based urban autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[43] Di Qi, L. Su, Jia Song, E. Cui, Taroon Bharti, and Arun Sacheti. Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data. *arXiv.org*, 2001.07966, 2020.

[44] Xiaojuan Qi, Qifeng Chen, Jiaya Jia, and Vladlen Koltun. Semi-parametric image synthesis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[45] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. PRECOG: prediction conditioned on goals in visual multi-agent settings. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019.

[46] Nicholas Rhinehart, Rowan McAllister, and Sergey Levine. Deep imitative models for flexible inference, planning, and control. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020.

[47] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016.

[48] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[49] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive GAN for predicting paths compliant to social and physical constraints. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[50] Amir Sadeghian, Ferdinand Legros, Maxime Voisin, Ricky Vesel, Alexandre Alahi, and Silvio Savarese. Car-net: Clairvoyant attentive recurrent network. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018.

[51] Ibrahim Sobh, Loay Amin, Sherif Abdelkarim, Khaled Elmadawy, M. Saeed, Omar Abdeltawab, M. Gamal, and Ahmad El Sallab. End-to-end multi-modal sensors fusion system for urban automated driving. In *Advances in Neural Information Processing Systems (NeurIPS) Workshops*, 2018.

[52] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019.

[53] Marin Toromanoff, Emilie Wirbel, and Fabien Moutarde. End-to-end model-free reinforcement learning for urban driving using implicit affordances. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[55] Dequan Wang, Coline Devin, Qi-Zhi Cai, Philipp Krähenbühl, and Trevor Darrell. Monocular plan view networks for autonomous driving. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[56] Bob Wei, Mengye Ren, Wenyuan Zeng, Ming Liang, Bin Yang, and Raquel Urtasun. Perceive, attend, and drive: Learning spatial attention for safe self-driving. *arXiv.org*, 2011.01153, 2020.

[57] Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting copycat agents in behavioral cloning from observation histories. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[58] Yi Xiao, Felipe Codevilla, Akhil Gurram, Onay Urfalioglu, and Antonio M. López. Multimodal end-to-end autonomous driving. *arXiv.org*, 1906.03199, 2019.

[59] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. End-to-end learning of driving models from large-scale video datasets. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[60] Yan Xu, Xinge Zhu, Jianping Shi, Guofeng Zhang, Hujun Bao, and Hongsheng Li. Depth completion from sparse lidar data with depth-normal constraints. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019.

[61] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark E. Campbell, and Kilian Q. Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020.

[62] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling. *arXiv.org*, 1805.04687, 2018.

[63] Zhishuai Zhang, Jiyang Gao, Junhua Mao, Yukai Liu, Dragomir Anguelov, and Congcong Li. Stinet: Spatio-temporal-interactive network for pedestrian detection and trajectory prediction. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[64] Albert Zhao, Tong He, Yitao Liang, Haibin Huang, Guy Van den Broeck, and Stefano Soatto. Lates: Latent space distillation for teacher-student driving policy learning. *arXiv.org*, 1912.02973, 2019.

[65] Brady Zhou, Philipp Krähenbühl, and Vladlen Koltun. Does computer vision matter for action? *Science Robotics*, 4(30), 2019.

[66] Yin Zhou, Pei Sun, Yu Zhang, Dragomir Anguelov, Jiyang Gao, Tom Ouyang, James Guo, Jiquan Ngiam, and Vijay Vasudevan. End-to-end multi-view fusion for 3d object detection in lidar point clouds. In *Proc. Conf. on Robot Learning (CoRL)*, 2019.