# Multi-Modal End-To-End Autonomous Driving via Conditional Imitation Learning

Alex Zuzow and Charles Nimo

## I. Abstract

One of the key problems of autonomous vehicles is how do we represent the world around us in a digestible way. There are many modalities to choose from, each of which gives a different context for a situation, therefore it is important to choose the modalities which give the most important information gain. We explore the impact of adding and removing modalities with respect to early multimodal fusion paradigms in the context of conditional imitation learning. We test the impact of four modalities, RGB, LiDAR, optical flow, and velocity. Our model consists of two parts, a feature encoder, and a autoregressive waypoint predictor. There are two encoder architectures used in our experiments, the first is simply a pre-trained EfficientNet while the second is an EfficientNet that feeds into a transformer at the last block. We find that optical flow improves the model's performance although it becomes very unstable while training due to harsh augmentations of RGB images. Our conclusion is that optical flow provides key representation for end-to-end multimodal conditional imitation learning models; however, perturbations of RGB images drastically decrease model performance, effectively only adding noise to the model.

## II. Introduction

Crafting generalized decision-making rules for real-world Autonomous driving is difficult. Imitation Learning has become a widely used approach for training autonomous driving systems. However, this approach is suitable only when performing simple tasks such as lane-following or obstacle avoidance. For more complex tasks such as driving when there are several options available for the next action, imitation learning begins to become insufficient. For example, at a road intersection, the vehicle is unable to predict an optimal decision because the camera input alone is insufficient to decide whether to turn left, right, or continue straight. Even then if the vehicle determines some course of action, it may not be the desired action of the passenger; Thus the challenge here is to connect imitation learning and the commands of the passenger. Conditional imitation learning aims to address this problem.

Prior works have explored this approach in the context of conditional imitation learning with context given through RGB images and LiDAR but have not thoroughly investigated the benefit of additional modalities. Introducing additional modalities can add information to a model that is not captured from other modalities. These new modalities can compensate for the shortcomings of other modalities and give new insights into the reasoning behind an expert's actions.

In this project, the goal was to find the impact of optical flow as a modality in relation to other modalities such as RGB images and lidar. We utilize the Carla simulator which provides multiple modalities and we then create our own optical flow images by using a lightweight version of RAFT [24]. During our experimentation it is found that due to optical flow's highly susceptible nature to image perturbations it becomes an unreliable modality in all instances. Simple situations such as drastic weather changes or blur can lead to failures in optical flow. The requirement for robust and accurate models in autonomous driving limit the usefulness of optical flow in practical applications. RGB images and LiDAR tend to be competitive with the addition of optical flow when image degradation's are present. Thus our main contribution is the development of an end-to-end multimodal conditional imitation learning approach, of which we compare the influence of different modalities in relation to each other.

## III. Related Work

### A. Decision Making and Motion Planning

The typical workflow of an autonomous vehicle system seeks to process a stream of observations from the vehicle on-board sensors with high level routing plans to the executable control output such as steering angles, accelerations, and braking actions. At the behavioral layer,is a decision making system that decides the discrete state of mid-level driving actions such as lane-changing, car-following, and turning left or right. When a behavioral decision is made, the motion planning system is in charge of determining a safe, comfortable, and dynamically feasible continuous trajectory to achieve the driving action selected from the decision making system.

To this end, Deep Reinforcement learning has demonstrated significant success in the area of autonomous driving behavioral decision making, especially for the cases of highway scenarios and urban intersections. Moghadam et al.,[20] present an end-to-end continuous deep reinforcement learning trajectory planning approach towards motion planning that explores the driving corridors in surrounding moving vehicles and then generates spatiotemporal trajectories to safely navigate through traffic. In this work, lattice representations enable predictive planning based on the surrounding vehicles while also considering the kinematic constraints and vehicle motion limitations to generate optimal trajectories. It is quite common to add rule-based safety constraints that can determine unsafe actions before they are executed to mitigate the concerns of safety performance. Hoet et al, [11] explore training a Deep Q-Network (DQN) in a simulation environment to determine driving commands and compare the agent's performances to the effects of different neural network architectures. In this work, the Deep Q-Network is provided with the outputs from the perception system to label the unsafe behavioral decisions in unprotected turn scenarios. Some studies train a lane changing decision making system based on Deep Q-Networks for decision making and then utilizes a rule-based layer to determine the safety of a planned trajectory [25] [19] [4]. Contrarily, some other studies demonstrate learning from human demonstrations through imitation learning and then introduce perturbations to discourage undesirable behavior [2].

### *B. Motion Perception*

The capacity to perceive environmental states, specifically the existence of surrounding objects and their motion behavior is crucial for autonomous driving. The estimation of environmental state requires perception, which aims to identify the locations and categories of objects in the surrounding environment. Depending on the input modality, existing works in motion perception can be categorized into three areas - 1) 2D object detection on images [22] [18] [17] [13][32] 2) 3D object detection on point clouds [30] [12] [29] [33] [28] and 3) fusion based detection [9] [5] [15] [14] [16]. Of the three areas of motion perception, our work focuses on multiple combinations of fusion based detection with RGB, LiDAR, and optical flow.

The application of optical flow relates to our research on using explicit perception systems to improve the performance of learned control policies. In [31], Zhou et al analyze the usefulness of different computer vision modalities on the performance of sensorimotor control of a vehicle. In this study, their results show that models equipped with explicit intermediate representations such as optical flow train faster, achieve higher task performance, and generalize better to previously unseen environments. For autonomous driving, the presence of moving objects in its environment substantially increases the difficulty of a control task, requiring fast reaction time and prediction of the future trajectories of the moving objects. Using deep reinforcement learning, a model of the environment's dynamics can be learned implicitly. In this paper we investigate the use of optical flow to help the network learn the use of motion features. Amiranashvili et al [1] explores how the use of optical flow to learn an explicit representation of motion improves the quality of the learned controller in dynamic scenarios. In this study, a small specialized optical flow network is derived from FlowNet [**?** ] and is run jointly with a reinforcement learning network while keeping computational requirements practical.

In our work, we analyze the usefulness of optical flow in tandem with other sensor modalities for autonomous driving. Optical flow can be used to provide estimates of the absolute velocity of the vehicle based on the flow of stationary objects such as the ground. Additionally, the absolute velocity of surrounding vehicles can also be estimated by using both the relative velocity and the absolute velocity of the ego-vehicle. In this paper, we use RAFT for predicting optical flow with a model that we trained on multiple datasets.

### *C. End-to-End Learning for Autonomous Driving*

The end to end learning approach for self driving vehicles has been explored since the 1980s. These legacy works have inspired new proposals based on imitation learning using CNNs. Modern self-driving cars have recently employed imitation learning

Recent studies have shown that introducing multimodal methods for end-to-end driving can add new information which complements the standard RGB images. [27] incorporates RGB, depth, and velocity into three different modality fusion architectures. They compare performance between early, intermediate, and late fusion; they find that early fusion returns the best results.

[10] proposes to use a network similar to [**?** ] which produces semantic segmentation and depth estimation images, they then concatenate intermediate features from an optical flow model to the other input images.

They note that providing more information to imitation learning models can lead to worser results, this is known as causal confusion. To combat this problem they add a high amount of noise to their inputs and train the network with a large dropout ratio.

In comparison to the previous works [21] doesn't opt for an early fusion multimodal scheme, instead, they design an intermediate fusion model based around transformers. They encode each of their modalities using different variants of ResNet, they then apply modality fusion at the end of each block by sending the outputs to small transformer networks.

The goal of this is to not only have a competent fusion method but to also be able to explain the effect of each modality by visualizing attention maps.

We base our work on a behavioral cloning method called conditional imitation learning. In the CIL paper [**?** ], they propose a method that enables the driver to be able to input high-level commands such as turn left, right, go straight, break; these high-level commands enable the model to deal with ambiguous situations such as intersections and also make it possible to navigate the world along a specified path. They propose to use separate branches in their network for each high-level command thus each sub-network deals with only a single command.

Instead of providing high-level commands such as which direction to turn [7] expands on this idea by instead taking commands in the form of sparse goal locations, the CIL network is then converted into an auto-regressive waypoint prediction network. This is equivalent to modifying CIL to predict waypoints conditionally based on sparse goal locations instead of directly predicting vehicle actions conditioned on navigational commands. In recent years this auto-regressive waypoint prediction variant of CIL has become very popular; [3] [4] [6] adopt this as their method of choice, all of which are top contenders on the CARLA leaderboard challenge.

[4] is a knowledge distillation method that first trains a teacher who has access to ground truth LiDAR birds-eye view semantic maps, the teacher is trained using a supervised expert which is handcrafted in CARLA. A student network consisting of only RGB images as input is then trained using this teacher as a supervisor.

[6] approaches the autonomous driving problem by predicting waypoints in birds eye view scene coordinates based on intermediate attention fields generated through transformers. [6] learns directly from a birds eye view representation as opposed to [4] which learns an image to trajectory mapping.

Our work takes inspiration from [27] [21] [26] where we opt for a early modality fusion scheme and use a transformer at the end of our encoder network that learns an attention mask over features extracted from a 2D CNN

## IV. Data

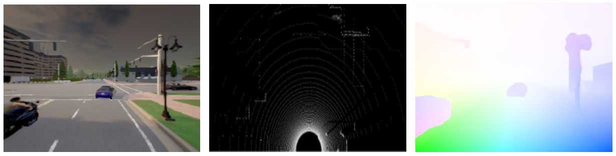
Our dataset is created using the CARLA simulator specifically version 0.9.10, this simulator provides 8 towns for use. 7 of the towns are used to create the training set and town 5 is used to evaluate on. We utilize the dataset made public by [21], they utilize a hand crafted expert agent that uses privileged information. This expert was created by [4], it is noted by both [4] [21] that the expert agent does have its faults and as such [21] has gone on to create their own expert agent which has not been made public at the time of writing.

The data collected from this expert is recorded at 2 fps and 256x256 resolution with a front facing RGB camera, birds eye view LiDAR, and various measurements such as where the ego vehicle will be in the next four frames. We then utilize the RGB images given to us to generate our own optical flow data. Optical flow is derived from our own lightweight version of RAFT [24], in all experiments using optical flow we transform the intermediate flow features into RGB images. Note that all images in the dataset have come pre-augmented with the same image augmentations as CIL including pixel dropout, blurring, Gaussian noise, and color perturbations. To add extra variation every other frame also experiences extreme weather shifts such as from sunny to heavy rain. Due to all RGB images being pre augmented the resulting optical flow images are in many cases of suboptimal quality. Our optical flow network is not robust to these perturbations which results in the derived flow being incomprehensible whenever there are drastic changes in augmentations from proceeding frames. The data generated for training and evaluation feature an abnormal amount of adversarial scenarios. It is a common occurrence in the dataset to find scenarios such as vehicles running red lights, uncontrolled 4-way intersections, or pedestrians emerging from occluded regions to cross the road at random locations. In total there is 60 GB of data used in training and 11 GB used in evaluation, this equates to 149k training frames and 29.6k evaluation frames used in evaluation

## V. Method

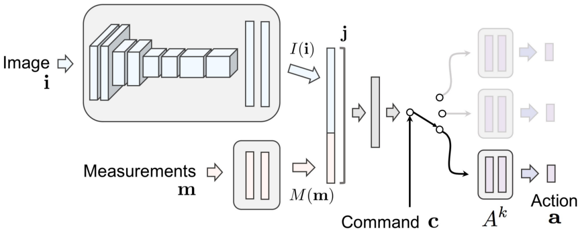
In this work we propose two new architectures for end-to-end driving, we then compare how different combinations of modalities affect autonomous driving performance on both networks. Each network consist of two components, an image encoder for early multimodal fusion and an autoregressive waypoint prediction network. The encoder section is then made up of a convolutional network that then leads into a transformer; we do this with the goal of bringing attention to the most important high level feature maps. The only difference between the two networks is the convolutional network leading into the transformer. The first network uses an efficientNet-b0 [23] while the second network uses a simple fully convolutional network. The output of these encoder networks are then processed in the autoregressive waypoint prediction network which predicts the next four positions the car should travel towards to reach its goal. These waypoints and the velocity are then converted into actions such as the steering angle, and whether or not to press the gas or brake by using a pid controller.

Since our goal was to find the effect that optical flow has on CIL as a modality with respect to autonomous driving, we tested four different combinations of modalities: 1) [RGB , LiDAR] 2) [Optical Flow, RGB] 3) [Optical Flow, LiDAR] 4) [Optical Flow, LiDAR, RGB] on both networks. We test on two separate networks to see if the modalities tested can be utilized to greater value with an encoder network that is highly engineered. We consider the task of autonomous driving where the goal is to complete a given route safely responding to adversarial agents and abiding by traffic laws. To achieve this we use CIL which is an extension of the imitation learning, in imitation learning we have a policy $\pi$ that learns to imitate the behavior of an experts policy $\pi^*$. In our problem an agents policy is mapping inputs to waypoints which direct the agent towards an end goal. To learn this policy CIL expands on the imitation learning algorithm known as behavioral cloning which frames imitation learning as a supervised learning problem. In the original CIL architecture they condition on high-level commands such as left, right, or straight, however we instead condition on high-level goal locations, these goal points are provided through GPS coordinates in CARLA. We create two networks that only differ in the initial encoder architecture. Network 1 begins with a simple fully convolutional network similar to that of the original CIL architecture while network 2 starts with the highly optimized efficientNet-b0. We use efficientNet because of its proven performance in comparison to resNet. Both networks result in a 320 feature maps of dimensions 320x8x8, we then create a learnable positional embedding and project the current velocity into vector using a linear layer. The flattened feature maps, positional embedding, and velocity projection are combined then using element-wise summation. This new tensor is then fed as input into the transformer which we borrow the base implementation from [21]. The transformer applies the attention mechanism to the resulting high-level intermediate feature maps and velocity projection. This transformer then returns an output which is the same dimensions as the input feature maps. This then leads into the last step of the encoder which reduces the feature map dimensions to 320x1x1 by average pooling and flattened into just a 320-dimensional feature vector.



(a) RGB (b) LiDAR BEV (c) Optical Flow

Fig. 1: An example of the three types of modalities used in this project.



(b)

Fig. 2: Original CIL branched architecture: vehicle actions are decided by action branches which return a triplet <steering angle, throttle, and brakes>. The branch used is decided by the high level command {turn-left,turn-right,go-straight,continue}. Originally only RGB images and the ego vehicles speed were considered.

This feature vector is a compact representation of the observations the agent has made at a time step. We then feed this vector into the waypoint prediction network. The waypoint prediction network is the same as Transfuser [21]. This is implemented by first sending the encoder output to a MLP which returns a feature vector of length 64 which is what we use to initalize the hidden state of the GRU.The update gate takes the current position and the goal location, The hidden state is then passed to a linear layer which predicts the waypoints for 4 future time steps. The input to the first GRU unit is given as (0,0) since the BEV space is centered at the ego-vehicle's position. We then use two pid controllers which are implemented by [4] for lateral and longitudinal control which allows us to obtain steer, throttle and brake values. We use the same configuration that they use in their work.

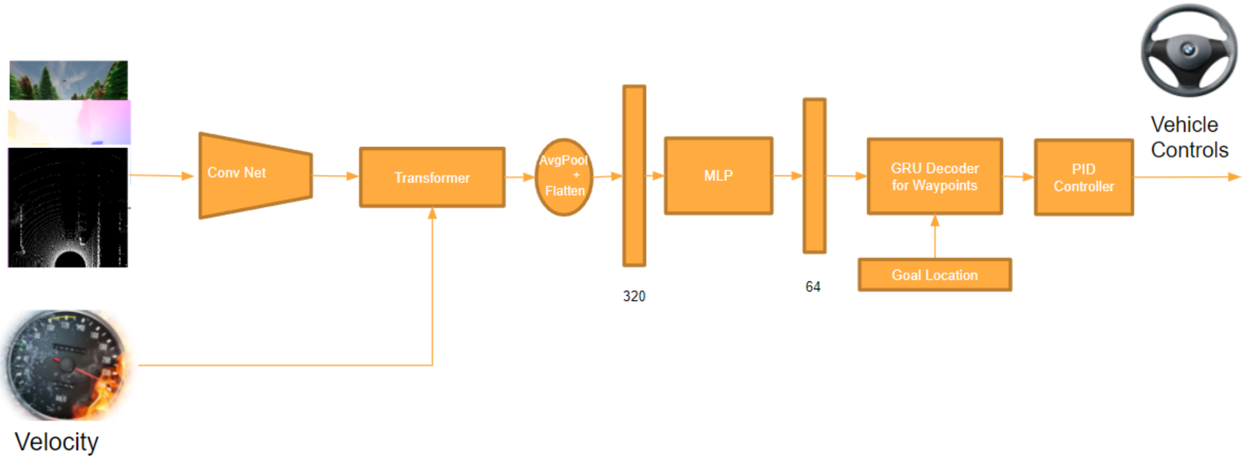


Fig. 3: Our Architectures: We consider multiple combinations of a single view RGB image, optical flow,and LiDAR BEV, as inputs to our network. We compare two architectures that differ only in the conv net section of the network. In network 2 we use the highly engineered EfficentNet-b0 while in network 1 we create a simple fully convolutional network similar to that of the original CIL architecture.

We follow [21] [4] and train using a $\ell_1$ loss function which compares the predicted waypoints and ground point waypoints. Let $W_t^p$ represent the predicted waypoints and $W_t^{gt}$ represent the ground truth waypoints at time-step $t$, then the loss function is defined as

$$\ell = \sum_{t=1}^{T} \|W_t - W_t^{gt}\|_1 \tag{1}$$

| Network 1 | Modalities | Town 5 Short | | Town 5 Long | |
|---|---|---|---|---|---|
| | | DS | RC | DS | RC |
| | I, F, L | - | - | - | - |
| | I, L | - | - | - | - |
| | I, F | - | - | - | - |
| | L, F | 40.44 ± 30.16 | 80.54 ± 33.72 | 18.17 ± 17.97 | 43.83 ± 27.35 |
| **Network 2** | **Modalities** | **Town 5 Short** | | **Town 5 Long** | |
| | | **DS** | **RC** | **DS** | **RC** |
| | I, F, L | 10.19 ± 11.70 | 33.10 ± 15.68 | 3.53 ± 10.87 | 21.35± 7.72 |
| | I, L | 38.56 ± 24.74 | 65.11 ± 35.84 | 14.92 ± 16.13 | 33.76 ± 14.67 |
| | I, F | 27.23 ± 31.15 | 50.14 ± 24.67 | 10.32 ± 15.83 | 26.49 ± 11.33 |
| | L, F | 42.48 ± 33.20 | 85.08 ± 31.01 | 17.17 ± 20.63 | 40.83 ± 24.92 |

TABLE I: Here we present the mean and standard deviation DS and RC of our networks on town 5 short and town 5 long, each test comprises of 10 different routes on town 5. We compare the performance of three modalities tested RGB Images: I, LiDAR: L, and Optical Flow: F

## VI. Experiments

In this section we compare the performance of different combinations of modalities on two separate networks, to compare performances we utilize the CARLA simulators scenario runner which creates deterministic situations, we then compare the number of infractions each model makes to each other. Our experiments in the CARLA simulator consist of navigating a predefined route in a wide variety of landscapes such as freeways, cities, and residential areas. Each route then has multiple different scenarios which add different adversarial situations. This allows us to test the ability of our models to react to many kinds of situations such as obstacle avoidance, unprotected turns at intersections, vehicles running red lights, and pedestrians emerging from occluded regions to cross the road at random locations. Each situation has a specified amount of time the agent is given to complete the task while needing to navigate the route while reacting to dynamic agents. Performance is measured using three metrics, (1) route completion, Percentage of the route distance completed by an agent, (2) infraction count, the amount of traffic infractions the agent accumulates ,(3) driving score: the product between the route completion and a infractions penalty score, each infraction has its own penalty score which is summed at the end of testing to create the infraction penalty score. We follow [21] and evaluate the performance of our networks on town 5 on 10 short and long routes. We cannot make a fair comparison between our networks and previous works which were submitted to the CARLA Autonomous Driving Leaderboard [3] [6] [21] due to not having fully trained any of our models and because we did not submit our models to the CARLA leaderboard due to time constraints. Unfortunately we also were unable to finish training our network all on modality combinations in time. We

| Method | Town 5 Short | | Town 5 Long | |
|---|---|---|---|---|
| | DS | RC | DS | RC |
| LBC | 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 |
| TransFuser | 54.42 ± 4.29 | 78.41 ± 3.75 | 33.15 ± 4.04 | 56.36 ± 7.14 |
| Ours | 42.48 ± 33.20 | 85.08 ± 31.01 | 17.17 ± 20.63 | 40.83 ± 24.92 |

TABLE II: Here we compare our best performing model against 3 different baselines, each baseline is picked due to their performance in the CARLA online leaderboard challenge and having tested on the same routes with similar adversarial situations. It should be noted that this is only a small subset of tests, a better comparison of our best network to theirs would be to submit our model to the complete CARLA online leaderboard which consists of 100 different secret routes

compare our best model against a few baselines which have been tested on similar situations. In Table I We find that network 2 with LiDAR and optical flow performs the best while network 2 with RGB images and optical flow performs the worst. We were unable to test all modalities combinations on network 1 however we hypothesis that network 1 with RGB images and optical flow would perform the worst out of all networks. Given this information we are able to compare our best model to previous CARLA Leaderboard contenders [21] [4] who have tested their models on the same routes but using different scenarios than ours. We find that our network which uses only LiDAR and optical flow is able to have comparable results with the baselines, however, the standard deviation of our network is extremely high in comparison. It seems as though if no adversarial agents throw the ego vehicle into unforeseen situations it is then able to navigate through the routes well. Unfortunately our agent is unable to recover from these adversarial situations in a robust manner.

We observe that our model is far more likely to collide with other vehicles in comparison to the baselines, during testing we noticed that the ego vehicle tends to get caught on other vehicles which adds multiple vehicle infractions. The agent is also unable to handle red lights very well with any combination of modalities. This is exaggerated with our network which uses only LiDAR and optical flow due to it needing to interpret the light change using only optical flow. Surprisingly our network which utilized all three modalities performed the worst, we believe this is because of causal confusion [8]. It is also noted that the intermediate values of optical flow can be a better feature than optical flow images [10].

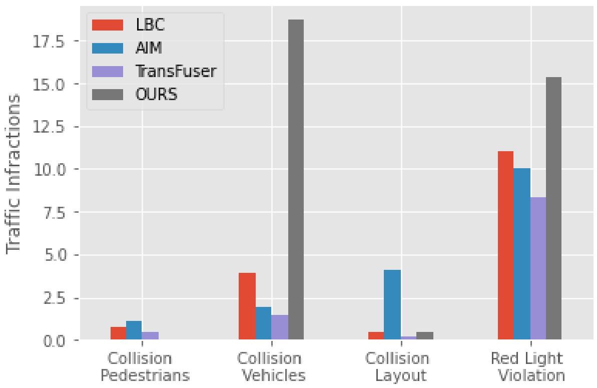


Fig. 4: We show the average number of infractions our car accumulates during our evaluation on town 5 short, this is not a exact comparison to the baselines due to us not testing on the same scenarios

## VII. Conclusion

During this project we demonstrated that optical flow is a useful modality to include in end-to-end autonomous systems, however the true benefit is hard to gauge Due to many compounding factors such as not fully training any models, using pre-augmented data, and training the agent using a sub optimal expert. Given all of these negatives, utilizing optical flow and LiDAR produced the best results for which does lead us to believe it is a modality rich with information that is utilizable in autonomous driving. There are many things that could be done to improve this work, the simplest would be to continue training the networks to their full extent and then compare them using the CARLA online leaderboard challenge, this would give a better understanding as to how each modality truly interacts with one another and is affected by architecture changes. Another extension would be to use the intermediate flow as a feature instead of converting it to RGB. It would also be interesting to see how we could improve the inclusion of multiple modalities without harming performance this would allow us to be able to include semantic segmentation which is also a key modality to explore, we believe including this would drastically improve the red light violations.

## References

[1] Artemij Amiranashvili, Alexey Dosovitskiy, Vladlen Koltun, and Thomas Brox. Motion perception in reinforcement learning with dynamic objects, 2019.

[2] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst, 2018.

[3] Dian Chen, Vladlen Koltun, and Philipp Krähenbühl. Learning to drive from a world on rails, 2021.

[4] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating, 2019.

[5] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving, 2017.

[6] Kashyap Chitta, Aditya Prakash, and Andreas Geiger. Neat: Neural attention fields for end-to-end autonomous driving, 2021.

[7] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[8] Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning, 2019.

[9] 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, 2021.

[10] Jeffrey Hawke, Richard Shen, Corina Gurau, Siddharth Sharma, Daniele Reda, Nikolay Nikolov, Przemyslaw Mazur, Sean Micklethwaite, Nicolas Griffiths, Amar Shah, and Alex Kendall. Urban driving with conditional imitation learning, 2019.

[11] Carl-Johan Hoel, Krister Wolff, and Leo Laine. Automated speed and lane change decision making using deep reinforcement learning. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2018.

[12] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven Waslander. Joint 3d proposal generation and object detection from view aggregation, 2018.

[13] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints, 2019.

[14] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection, 2020.

[15] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection, 2020.

[16] 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, 2020.

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

[18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *Lecture Notes in Computer Science*, page 21–37, 2016.

[19] Branka Mirchevska, Christian Pek, Moritz Werling, Matthias Althoff, and Joschka Boedecker. High-level decision making for safe and reasonable autonomous lane changing using reinforcement learning. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2156–2162, 2018.

[20] Majid Moghadam, Ali Alizadeh, Engin Tekin, and Gabriel Hugh Elkaim. An end-to-end deep reinforcement learning approach for the long-term short-term planning on the frenet space. *CoRR*, abs/2011.13098, 2020.

[21] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving, 2021.

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[23] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.

[24] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow, 2020.

[25] Junjie Wang, Qichao Zhang, Dongbin Zhao, and Yaran Chen. Lane change decision-making through deep reinforcement learning with rule-based constraints, 2019.

[26] Bob Wei, Mengye Ren, Wenyuan Zeng, Ming Liang, Bin Yang, and Raquel Urtasun. Perceive, attend, and drive: Learning spatial attention for safe self-driving, 2021.

[27] Yi Xiao, Felipe Codevilla, Akhil Gurram, Onay Urfalioglu, and Antonio M. Lopez. Multimodal end-to-end autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, page 1–11, 2020.

[28] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding

box estimation, 2018.

[29] Bin Yang, Ming Liang, and Raquel Urtasun. Hdnet: Exploiting hd maps for 3d object detection, 2020.

[30] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds, 2019.

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

[32] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points, 2019.

[33] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection, 2017.