Faculty of Mathematics and Informatics

VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
MODELLING AND DATA ANALYSIS
MASTER'S STUDY PROGRAMME

INVESTIGATION OF PERCEIVER NETWORKS FOR IMITATION LEARNING IN AUTONOMOUS DRIVING

IMITACINIS MOKYMASIS NAUDOJANT PERCEIVER TINKLUS AUTONOMINIAM VAIRAVIMUI

Master's thesis

Author: Augustas Žaltauskas
VU email address: augustas.zaltauskas@mif.stud.vu.lt
Supervisor: Dr. Virginijus Marcinkevičius

Vilnius

2022

Abstract

Imitation learning algorithms are widely applied in autonomous vehicles field. They reach good results in lane following and obstacle avoidance tasks. However current approaches still struggle in urban environments with multiple dynamic objects and complex traffic rules. We discuss that improving image encoding methods could help to alleviate the issues related to static or dynamic object detection, traffic light and stop sign infractions. We propose to use perceiver network as image encoder and show that it reaches lower loss compared to the state-of-the-art model when pre-trained image encoders are considered and no fine-tuning is done. In the same setting, the proposed model also shows higher road completion percentage and lower infraction rate when test runs are made in simulated environment.

Keywords: imitation learning, autonomous driving, perceiver, image encoding

Santrauka

Imitacinio mokymosi algoritmai yra dažnai taikomi autonominiuose automobiliuose. Šie algoritmai pasiekia gerus rezultatus kelio sekimo ir kliučių išvengimo užduotyse. Tačiau jų rezultatai krenta, kai algoritmai yra taikomi miesto aplinkose su dideliu eismo dalyvių skaičiumi ir sudėtingesnėmis eismo taisyklėmis. Vaizdo apdorojimo algoritmų tobulinimas gali padėti išspręsti problemas susijusias su statinių ir dinaminių objektų aptikimu, sumažinti eismo taisyklių pažeidimų kiekį ties šviesoforais bei stop ženklais. Šiame darbe siūlome naudoti vaizdo klasifikacijai apmokytą Perceiver tinklą vaizdo apdorojimo užduočiai. Parodome, kad pasiūlytas modelis pasiekia mažesnę paklaidą validacijos duomenyse lyginant su geriausius rezultatus pasiekiančiu modeliu, kai vaizdo apdorojimo dalis nėra papildomai apmokoma autonominiam vairavimui. Taip pat, testuojant šiuos modelius simuliuotoje aplinkoje, automobilis naudodamas pasiūlytą modelį vidutiniškai nuvažiuoja didesnį trasos atstumą bei padaro mažiau eismo taisyklių nusižengimų.

Raktiniai žodžiai: imitacinis mokymasis, autonominis vairavimas, perceiver tinklas, vaizdo apdorojimas

Contents

List of Figures

List of Tables

1 Introduction

1.1 Research Area

Autonomous vehicle is a vehicle that is able to sense environment and decide what actions to take with little to no human interaction. They are usually equipped with multiple modality sensors to cover all parts of environment, similarly to human senses. Devices like lidar, sonar or radar are used for distance measurement to surrounding objects. Visual data is taken from cameras that covers front view or multiple sides in multi camera setup. Cameras, as a relatively cheap option is usually the main and preferable source of input information for the models. In addition, Global Positioning systems can be used to determine approximate vehicle's position and calculate optimal path to the target destination. Furthermore, internal sensors are commonly used to measure vehicle's position, acceleration, velocity, and other metrics.

Over the years the number of cars on the road increased heavily, therefore such issues as traffic congestion, pollution, and road safety become critical. Autonomous driving is one of possible solutions to alleviate the severity of these problems. As a result, autonomous vehicles field has seen an increase in popularity in both research community and industry. Furthermore, autonomous vehicles are taking an increasingly larger part in automobile market which is only expected to grow.

Imitation learning methods became one of the to go approaches for developing autonomous vehicles. It allows end-to-end models, which approximate human's behaviour and maps observations (visual, distance data, vehicle measurements) to actions that car should take. Behaviour cloning (BC) is a branch of imitation learning, that focuses on off-policy model training in supervised learning approach that directly replicates desired behaviour from expert's examples. Together with observations, position of the destination point or high-level directions to it are provided for the model, to help ego vehicle reach destination using optimal path.

Imitation learning depend on expert's examples, therefore real world multiple hour data sets are used for training the models. However, collection of real-world dataset has important drawbacks: it requires human effort, not all measurements can be reliably taken, and it brings safety risks. In real world setting it is difficult to evaluate model performance and it only gets more complicated in safety critical situations. As a result, many authors investigate simulated data as a proxy to real world data. It gives flexibility in measurements, controlled environment, and simplicity in data collection for model training. It also provides possibility to evaluate models when infractions occur (either from ego vehicle or other traffic participants) without safety risks.

With many advancements in this area, there are still unsolved issues that needs to be addressed for autonomous vehicles to be performant in complex environments and real-world setting.

1.2 Problem Relevance

Recent studies [5] [17] [14] [9] indicate several main problematic areas for autonomous vehicles. Datasets collected from human driving in real world setting is known to be biased in a sense that they contain limited amount of examples for safety critical scenarios, like off road drifting, rules infractions, unexpected objects avoidance, etc. On the other hand, examples of simple behaviours like driving straight or road following can have overwhelming quantity. This negatively impacts learning of expert's

behaviour in complex scenarios. Bias in datasets can also create causal confusion [5], when spurious correlation can be mixed up with actual causation and thus model learns invalid behaviour in these situations.

Another problem is caused by formulation of behaviour cloning and it's underlying assumptions. During training, the examples are independent from each other and predictions does not affect the states seen during training. However, during evaluation, actions from the model will affect following states, this causes to break i.i.d. assumption made by most algorithms. As a result, this leads to distributional shift [17] [9] between training and testing phases. This in turn can lead to mistakes to be made by the model due to unfamiliar state distributions seen during testing. This issue can be addressed with iterative on-policy algorithms, however it is difficult to apply it in real world setting [14].

Moreover, high variance in learned policies was observed in [17], when the performance of trained model depends on initialization seed and the order of samples seen during training. The issue arises when training on longer demonstrations from expert which are dependant on previous actions and thus i.i.d. assumption does not hold in those cases.

Difficulty of adapting trained models to different environments and road conditions like, different lightning, weather conditions, driving on different types of road were already addressed in early autonomous driving papers [13], [2]. This issue can be alleviated by models with better generalization capability, training on already processed data or gathering datasets, that exposes models to as much environment variability as possible. However, performance of the state-of-the-art end-to-end models still visibly drops when applied in unseen conditions [5].

In addition, driving through urban areas introduces more dynamic objects and denser traffic, and the model should learn how to act in these scenarios. [5] shows that no current approach reliably handles scenarios with a lot of dynamic objects (e.g., other cars, pedestrians). Furthermore, urban areas presents multiple intersections with traffic lights, which models struggles to detect as shown in [16].

All of the discussed issues are critical for autonomous driving and needs to be addressed, before autonomous vehicles can be safely and confidently used in real world setting. While issues like dataset bias, high variance in learned policies or generalization in different environments can be at least partially addressed with different dataset generation methods. Issues like improving dynamic or static object detection for obstacle avoidance or reducing traffic light or stop sign infractions can be addressed by improving raw observations encoding and feature extraction procedures.

1.3 Research Object

- Autonomous driving
- Imitation learning
- Perceiver
- Behaviour Cloning
- Image encoding

1.4 Goal

Propose imitation learning method based on perceiver as image encoding model for autonomous driving.

1.5 Contribution

In this work we focus on image encoding methods. Improving the quality of image encodings can alleviate issues related to static or dynamic object detection, traffic light infractions and improve overall performance. We start by investigating current state-of-the-art algorithms. As the baseline we choose Auto-regressive IMage-based waypoint prediction network (AIM) from [16] as it uses single front facing camera as an input and reaches top results on simulated autonomous driving dataset. We propose to modify baseline model with perceiver based image encoder [8] which has showed good performance on image classification tasks and leverage waypoint prediction network from AIM model for policy learning. Finally, we compare modified model with state-of-the-art model and show that it achieves lower validation loss and gets higher scores on test runs in simulated environment when no additional fine-tuning is done.

1.6 Objectives

In this work we follow these objectives to reach the set goal:

- Research and investigate imitation learning algorithms for autonomous vehicles
- Investigate datasets and metrics for algorithm evaluation
- Investigate issues and short-comings of state-of-the-art imitation learning approaches for autonomous vehicles
- Modify chosen algorithm and evaluate performance
- Compare results with state-of-the-art approaches

2 Literature Review

This section is separated into two parts. First, we describe Behaviour Cloning - the branch of imitation learning that is widely used in autonomous driving. In the second part we focus on reviewing approaches that leverages Behaviour Cloning algorithms in autonomous driving field, present improvements made over time and describe state-of-the-art algorithms.

2.1 Behaviour Cloning

To better understand the work described in literature review section, we need to define the Behaviour Coning (BC). In this section we follow BC definitions and formulas presented in [12]. BC methods are widely used in autonomous driving as they allow learning direct mapping from states to actions without the need of defining the reward function. The aim of behaviour cloning in action-state learning is to learn a policy π that generates a action u for a given state x. To put into autonomous driving context, we can think about the state s as an input data that the autonomous vehicle uses to make the decisions, it can be such information as the view from the cameras, distance measures from lidar sensors, current vehicle speed or angle of steering wheel. In this case actions u would be the new angle of steering wheel, and the amount of breaking or acceleration. Next, we can define a dataset D that is composed of N action-state pairs:

$$D = \{(u_i, x_i)\}_i^N$$

Given the dataset D, a policy π and it's parameters θ can be learned as a mapping from states to actions $u = \pi(x, \theta)$. The learning problem can be formulated as supervised learning where we optimize policy parameters θ by minimizing the loss l between expert's and learner's actions:

$$\min_{\theta} \sum_{i=1}^{N} l(\pi(x_i, \theta), u_i)$$

Here, N - number of action-state pairs, l - loss, π - learner's policy, θ - learner policy's parameters, x_i - expert's state, u_i - expert's action.

2.2 Behaviour Cloning For Autonomous Vehicles

Imitation learning is widely applied in the field of autonomous vehicles. One of the earlier approaches were suggested in "Alvinn" model [13] where it was trained to follow the road. As an input the model used camera images and laser range finder and the output was the direction the car should travel to follow the road. 3 layer neural network was used to map states to actions. For training it used simulated data, however follow up tests indicated that this approach, given certain field conditions could be sufficient for real world road following task.

Following it multiple different approaches were proposed to further solve autonomous vehicle problem. Deep Neural Networks were proposed to model end-to-end solution. Bojarski [2] applied Convolutional Neural Network (CNN) models to the similar road following problem. In this case, proposed approach used only camera images as an input. The output was steering wheel angle. The model was trained on real world driving dataset. During the training the model learns to detect internal represen-

tations of road features such as outline of road without implicitly training for road detection. Compared to at the time popular modular approaches, proposed end-to-end model showed great improvements as it optimizes for lane markings detection, path planning, and steering control tasks simultaneously. The proposed network architecture is composed of normalization, 5 convolutional and 3 fully connected layers. Since model was trained on real world data, image augmentation approaches such as shifts and rotations were used to allow network to learn to recover from poor positions on the road. Model was evaluated in simulation and real-world settings. Autonomy metric was derived for evaluation, which show how long vehicle was driving autonomously. If the car started drifting from the road, driver would interfere. On average it took around 6 seconds for driver to recover. On real world evaluation car was autonomously driven for 98 - 100% of the time. With reaching high performance, it shows that road following can be successfully applied on streets. The author shows, that this approach allowed the car to drive autonomously in various settings: with and without traffic, local reads with and without lane markings as well as highways. Author also shows that learned policy is capable of driving in parking lots and unpaved roads. However it if worth mentioning, that this approach solves only road following task, and the turns at intersections or lane changes were performed by the driver and that time was not counted in the final score.

Other work [11] shows that end-to-end behaviour cloning algorithm can be applied to obstacle avoidance. Proposed approach uses 6-layer CNN, trained with real world examples from expert driving off-road in various surroundings and weather conditions. Author outlines the several advantages of end-to-end model and using only cameras as sensors - no need to compute depth maps from stereo cameras or use expensive depth sensors, eliminate the need of hand-crafted heuristics, instead allowing the model to learn policy directly from data.

For truly autonomous vehicle, avoiding obstacles and staying on road is not sufficient, it needs to be able to drive in urban environment with more complex traffic rules. This problem was addressed in [4]. The author notes that there are cases when only visual information is not enough to address autonomous vehicle problem, for example in the intersections the decision ambiguity arises of not knowing to which direction to turn. Mapping only visual input to vehicle control is no longer possible as there are multiple possible turns to take, and according to driving rules or expert's examples each of them is correct. The same issue is also mentioned in earlier works [13]. To allow vehicle to perform turns at intersection author suggests to provide high level directional commands for the model. Commands are defined as "turn right", "turn left", "go straight" or "follow the road", in the same way as they could be provided by a person or by a route planner. Encoded commands together with image data and additional vehicle metrics are provided as an input to the model. These changes allows to apply imitation learning for autonomous vehicles in wider range of environments, like urban driving. Authors describe learning with additional high level commands as Conditional imitation learning (CIL).

Conditional imitation learning objective can be derived from Behaviour Cloning. Let's say that for every expert's state x_i and there is an action u_i made by the expert. We can construct expert's generated dataset D from N pairs of actions and states $D = \{(u_i, x_i)\}_i^N$. We define it as supervised learning problem, where parameters θ of function approximator $\pi(x; \theta)$ should be optimized to fit the

mapping from states to actions, same as described in section 2.1.

$$\min_{\theta} \sum_{i=1}^{N} l(\pi(x_i, \theta), u_i)$$

Here, N - number of action-state pairs, l - loss, π - learner's policy, θ - learner policy's parameters, x_i - expert's state, u_i - expert's action.

Further author argues that the implicit assumption that expert's actions are fully explained by observations (states), or in other words that there exists such function π^* that maps expert's states to actions $u_i = \pi^*(x_i)$ is not always correct. If this assumption holds, function approximator should be capable to fit the function π^*. Success in previously described tasks like road following [2] [13] can be explained by this estimator. However the more complex tasks, this assumption might not be sufficient. For example in intersection, only observations are not enough to explain chosen road. Author argues that we should consider including expert's internal state, such as intended destination, into approximator, because the same observation, could lead to different actions based on expert's internal state. Author represents expert's internal state as vector h, which could contain such information as goal, destination, or prior knowledge. Internal state vector together with observation explains expert's actions: $u_i = \pi^*(x_i, h_i)$. Accordingly we can construct imitation learning objective with included expert's internal state h:

$$\min_{\theta} \sum_{i=1}^{N} l(\pi(x_i; \theta), \pi^*(x_i, h))$$

It is evident that expert's policy does not include information provided by vector h. Latent state h is exposed by additional command input $c = c(h)$. During training, command c is provided by the expert. At test time commands c can come from the human user or navigational module. Further we can construct dataset from observations, commands and actions: $D = \{(x_i, c_i, u_i)\}_i^N$ and update objective to:

$$\min_{\theta} \sum_{i=1}^{N} l(\pi(x_i, c_i, \theta), u_i)$$

CIL implementation [4] uses image data and vehicle metrics as observations (states) data. Action space is continues and 2 dimensional, composed of steering angle and speed. Author proposes two alternative architectures - to use command as input, or create a branched network. In the first case, each input is processed individually: image encoded using CNN, metrics and commands encoded using fully connected networks. Afterwards, outputs are concatenated and put as an input for fully connected network, which predicts actions. All models are trained simultaneously as end-to-end model. For the second approach, branched network, the commands are not used as inputs, but instead act as switches for branched networks. Authors assume discrete commands and for each command new separate branch is added. The command acts as switch and determines which branch is used.

Approach was evaluated on simulated data. Two different maps were selected, one for training and one for testing. Human driving was recorded and provided as training dataset. It comprises around 2h of driving. Evaluation was composed of 50 pairs of start and end locations at least 1 km apart. For evaluation, two metrics were chosen: Success rate - percentage of how many destinations were reached,

KM per infraction - on average how many kilometers were driven before infraction. Results show that branched network performs best with 64% success rate and 1.18 km per infraction.

Improved CIL approach was presented in [5]. The author proposes to use ResNet architecture [6] for image encoding instead of CNN, and arguments that it better generalizes at learning reactions to dynamic objects and visual ques like traffic lights. Author applies transfer learning and uses ResNet model that was pre-trained on ImageNet dataset [18]. Additionally the model is trained to predict the speed of the vehicle. Training for speed prediction forces network to learn speed related features, this way, dynamics of the scene can be observed from visual data and not only from input speed. Author also proposed to use L1 loss function.

Other papers also introduce variations of multitask training in their work. [22] uses privileged learning and employs semantic segmentation on input image data as an additional tasks. As a result, image encoder learns better representations of input data, which in turn helps with learning better policy. Author shows that additional segmentation task, improves performance in scenarios when it needs to focus on small objects, like traffic light or break lights off other vehicles. In another approach [21], VGG [19] network is applied for image encoding. To improve performance, image encoder is pre-trained on ImageNet classification task and is additionally trained on depth prediction and semantic segmentation tasks. Another work [10] proposes to use encoder-decoder architecture for learning representative lower dimension features. Single encoder with separate decoders is used for depth and segmentation tasks. During inference, only encoder is used to extract features, which are then passed to driving model for motion prediction.

[1] investigates the possibility of training imitation learning model on visually abstracted data. It applies semantic segmentation as a pre-processing step of raw images. It is showed that even 6 classes can be enough to reach good results in autonomous driving task and performance does not increase substantially by adding more classes. Author outlines that including semantic segmentation task in image encoder training can lead to overall improvements in autonomous driving tasks. Author shows that even a small dataset of segmented data (few hundred examples) is enough to increase performance.

Recent work [16] introduces auto-regressive waypoint prediction network and multi-modal fusion algorithms and reaches state-of-the-art results in autonomous driving task on Carla No-crash benchmark in simulated environment. By using waypoint predictions network, model no longer predicts direct actions, but instead - expert's trajectory. Trajectory W is defined by a set of 2 dimensional waypoints x, y in Bird-Eye-View (BEV) space for each timestep t:

$$W = \{w_t = (x_t, y_t)\}_{t=1}^{T}$$

Waypoints are defined on ego vehicle's coordinate frame. States X consists of images from front facing camera. Following CIL definition, expert's internal state is represented as C, however it is given as a GPS position of goal location. Given states, trajectories and goal location, dataset D is constructed as a set of state, trajectory, goal location triplets of size N:

$$D = \{(W^i, X^i, C^i)\}_{i=1}^{N}$$

The policy π is trained in supervised manner using collected dataset of expert's examples D with the

loss function l.

$$\min_{\Theta} l(\pi(X, C, \theta), W)$$

As loss function L1 loss is used:

$$l = \sum_{t=1}^{T} |w_t - w_t^{gt}|$$

Here w - predicted waypoints, w^{gt} - waypoints generated by expert policy. Finally, using the inverse dynamics model implemented as PID controller I, waypoints are mapped to actions A (steering, throttle, break): $A = I(W)$

Using objective definition above, several models are implemented and tested on Carla benchmark. Using only frontal facing camera, state-of-the-art results are reached with Auto-regressive IMage-based waypoint prediction (AIM) model [16]. It consists of ResNet-34 network for image encoding, Multi-layer perceptron (MLP) and waypoint prediction network with Gated Recurrent Units (GRU). Best results are achieved using pretrained ResNet model on ImageNet classification task. High level model architecture is presented in Figure 2

In the same paper [16] author also presents multi-modal model. Similarly to AIM it uses waypoint prediction network, however instead of single image modality, depth maps from lidar sensor are also used as an input. Authors uses pretrained ResNet models for each modality encoding and suggests to use attentions, to fuse modalities at certain layers of both encoding networks. This allows model to reach top results on Carla Leaderboard benchmark [15].

As many proposed approaches achieve great results in a specific part of the problem, still there is no fully end-to-end approach which adapts well to different driving behaviours and would allow to run autonomous vehicles in real world setting, especially in dense traffic scenarios.

3 Methodology

We split Methodology into four sections. We begin by describing data, that was used in model training and evaluation phases. Next we describe the metrics that are used to evaluate models when running in simulated environment. In the third part we describe AIM [16] - a model, that reaches state-of-the-art results, when considering models with single frontal camera setup. We choose AIM model as our baseline, to which we compare the proposed models. In the fourth section, we describe two variants of proposed models.

3.1 Data

In this work we focus on simulated data as it allows easier experimentation with different models and evaluation process without the need of physical system. Data is generated from CARLA 0.9.10 simulator and the same approach is used as described in [16]. From total of 8 different towns, 7 are used for training and one (Town 5) is left for validation. Town 5 is chosen due to the large diversity of driving regions. RGB image is collected from front facing camera having a field of view of 100°. Images are collected at 2 frames per second rate. Weather conditions are changed every 0.5 seconds on each route, to have a uniform distribution of weathers across examples. Each image corresponds to four

waypoints that are 4-7 meters away from the car and shows the trajectory of traveling vehicle. Dataset is generated by collecting information from expert policy. Expert policy consists of A planner and 2 PID controllers for lateral and longitudinal control. Several heuristics based on the global position of the dynamic agents and traffic lights are used by the expert to avoid collisions and traffic violations. Expert policy is described more in depth in [16]. Collected dataset is around 20 GB in size and consists of 144k records, from which 8k are used for validation (Town 5). Some example images from Town 5 are shown in Figure 1

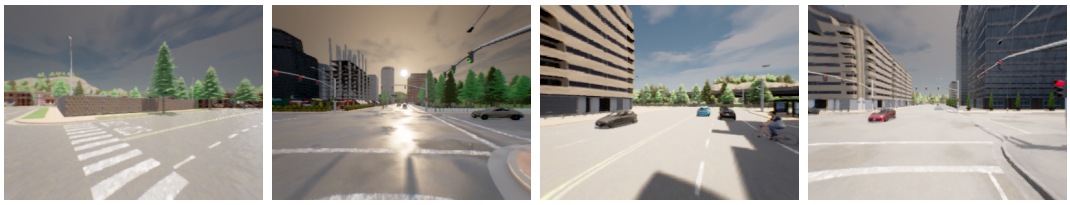

Figure 1: Examples of input images from Carla simulated environment with different weather conditions, obstacles, and position on road.

3.2 Metrics

Following [16] 3 metrics are used in this work to evaluate performance of autonomous driving in simulated environment:

Road Completion (RC) - percentage of route distance R_i completed on route i averaged over all routes N.

$$RC = \frac{1}{N} \sum_{i}^{N} R_i$$

Infraction multiplier (IM) - a multiplier composed of multiplication of penalty coefficients p_{ij} for each infraction type j in route i.

$$P_i = \prod_{j} p_{ij}$$

Different infractions have different pre-defined coefficients: 0.5 for collision with a pedestrian, 0.6 for collision with a vehicle, 0.65 for collision with infrastructure, 0.7 for red light violation, and 0.8 for stop sign violation. Ideal score is 1 and it is lowered with each infraction.

Driving Score (DS) - is a composite score of R_i and P_i. DS score of 100 means, that the route was fully completed without any infractions.

$$DS = \frac{1}{N} \sum_{i}^{N} R_i P_i$$

3.3 AIM Model

As the baseline model, we use AIM model [16]. It shows top results on Carla Autonomous Driving Leaderbord dataset and Carla Town 5 dataset [16] when compared to similar models that uses only image modality and single front facing camera. Limitations of only using single front facing camera brings benefits of cost optimization - no need for additional expensive and sensitive devices on board (e.g. lidar, multiple camera setting), and has faster processing times both in training and evaluation

steps. AIM model consists of ResNet network for image encoding, multilayer perceptron (MLP) (3 fully connected layers) and GRU network for waypoints prediction, high level model architecture is presented in Figure 2. Waypoint prediction network is described in depth in section 2.2. Best results are achieved using pre-trained ResNet model on ImageNet classification task[18].

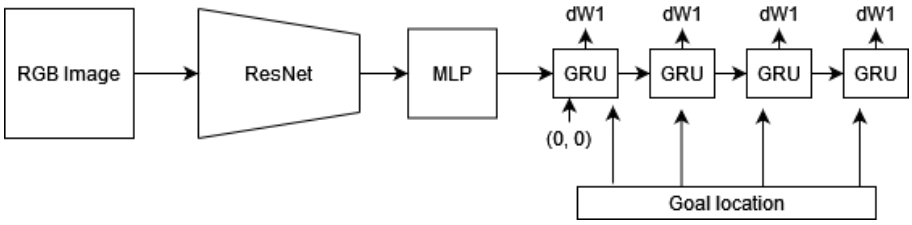

Figure 2: Architecture of AIM [16] model.

3.4 Proposed Model

For the proposed model we leverage the waypoint prediction network from [16], described in detail in section 2.2. Similarly to AIM model it uses raw images from frontal facing camera as inputs, however we modify image encoder part and MLP to accommodate the changes to the encoder. For the encoder we propose to use perceiver network, described in [8] and [7].

Perceiver is based on Transformer [20] networks. Transformers gained large popularity and reached state-of-the-art results in multiple fields including Natural Language Processing (NLP) and computer vision (CV). Transformers are universal models, that can be used with various inputs, however they also scale quadratically with the number of inputs in both memory and computation. Perceiver authors proposes to use cross attention module to project input to latent space. Then process latent arrays using stack of self-attention blocks. Perceiver iteratively attends to input array by alternating cross attention and self attention blocks. This way, most of computation is made in latent space, which is defined by a hyper parameter and does not depend on the input size. Also model complexity no longer scales linearly with input size. Additionally, attention make very little assumptions about input data and authors shows that it performs well in various tasks, such as ImageNet classification, pointcloud classification and audio event classification as well as multi-modal video-audio classification tasks.

We compare baseline AIM model with two variations of the modified model. First variation (further referred to as PMW-mean) is composed of perceiver as image encoder, multi-layer perceptron and waypoint prediction network, see Figure 3. As pre-processing step, raw images are resized and normalized to the size of 224x224. Perceiver network that was pre-trained on ImageNet classification task is used as the encoder. Perceiver consists of 8 repeating blocks. Each block is composed of 6 stacked self-attentions and attention weights are shared between all blocks. After final block latent space vectors are averaged and the mean latent vector of size 1024 is retrieved. It is passed to 4 layer MLP with 512, 256, 128, 64 sizes accordingly. As final step 64 size vector is used in waypoint prediction network. 2 waypoints are predicted and mapped to steering angle, throttle and break amounts using PID controllers. Method of averaging Perceiver latent space is similar to approach described in [8].

The second variation (further referred to as PMW-query) uses attention with a trainable query to extract encoding matrix from Perceiver's latent space. This approach is based on findings in [7]. We use

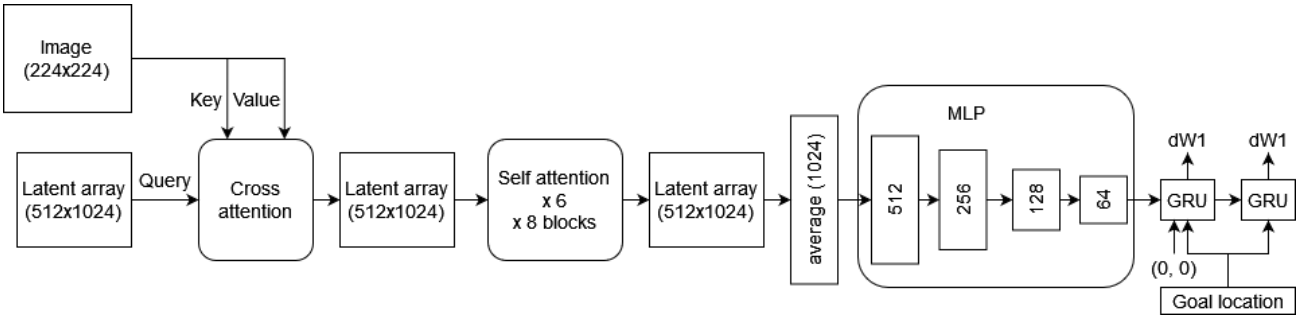

Figure 3: Architecture of proposed PMW-mean model

trainable query matrix of size 32x128. Applying cross attention with latent space, matrix of the same size is extracted. Matrix is flattened to 4096 vector and passed further to 3 layer MLP network with 1024, 256 and 64 sizes. The waypoint predictions network is kept the same as in AIM and PMW-mean model. Architecture is given in Figure 4.

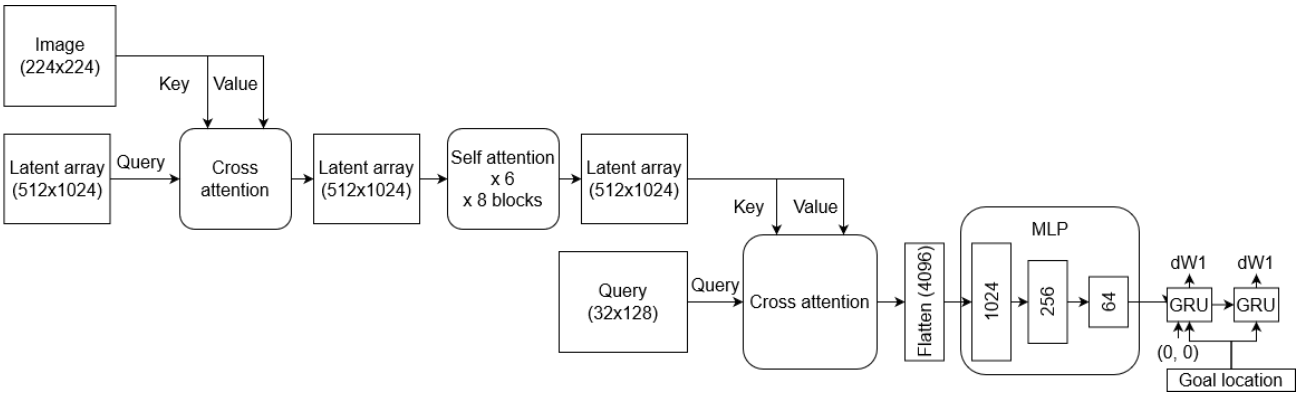

Figure 4: Architecture of proposed PMW-query model

4 Experiment Results

Results section is divided in three parts. First, model training is explored and optimal hyper-parameters are selected. In the second part the results of PMW and AIM models are compared. Finally, models are evaluated on runs in simulated Carla environment.

4.1 Model Training

Experiment consists of training AIM, PMW-mean and PMW-query models on Carla dataset as described above (see section 3.1). Image encoders (ResNet for baseline and perceiver for proposed models) are pre-trained on ImageNet classification task. During training image encoders' weights are fixed and not fine-tuned. This decision is made due to the lack of computational capabilities (limitation of GPU memory size) to fine-tune perceiver networks. To compare performance with state of the art results, AIM model is also trained without fixing weights. Following [3], all models uses L1 loss, which can be interpreted as mean absolute error between ground truth and predicted waypoints. Loss on validation dataset is computed every epoch and models are trained until validation loss no longer

decreases for at least 5 epochs. 15 epochs was enough for all the models to converge. Training and validation losses are shown in Figure 5 and Figure 6 respectively. Each model was trained 10 times using a different random seed during initialization to investigate the consistency of the results. In Figure 7 we can see the distribution of validation loss from all the models after 15 epochs.

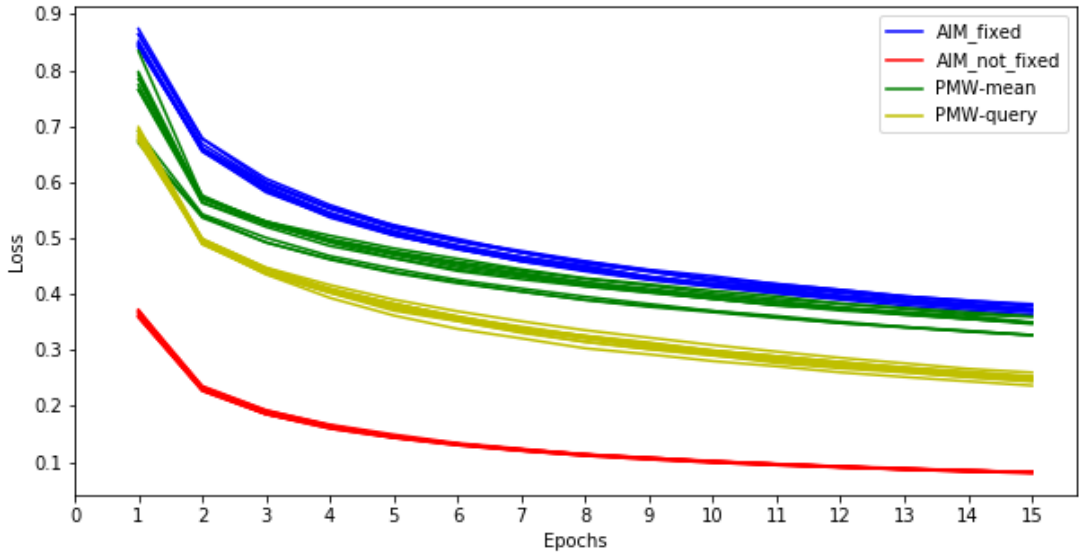

Figure 5: Training losses for baseline model AIM and proposed models: PMW-mean and PMW-query. Each model was run multiple times with different seed at initialization.

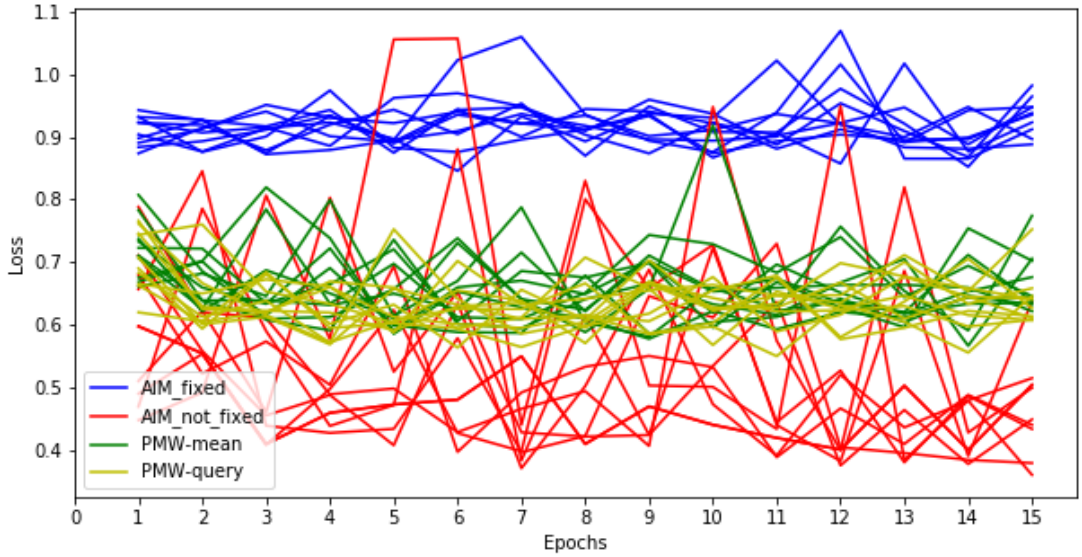

Figure 6: Validation losses for baseline model AIM and proposed models: PMW-mean and PMW-query. Each model was run multiple times with different seed at initialization.

4.2 Model comparison

From results on validation dataset (Figure 6) we can see that both proposed PMW variations reaches lower validation losses compared to baseline model (AIM-fixed) and that out of the box, perceiver has learned better latent features for autonomous vehicle problem. AIM model performance when allowing encoder weights to train (AIM-not-fixed) is also presented in Figures 5 and 6. In this case, AIM-not-fixed loss is the lowest and it outperforms both proposed models. This indicates that encoders fine-tuned on autonomous driving task learn better suited features. Statistics of training and validation loss after 15 epochs are provided in Table 1.

Table 1: Comparison of training and validation loss statistics between models after the 15th epoch of training

Data type	Statistics	AIM-fixed	AIM-not-fixed	PMW-mean	PMW-query
Training	Mean	0.3729	**0.0814**	0.3442	0.2486
	Standard dev.	0.005827	**0.000615**	0.012971	0.006256
	Minimum	0.363263	**0.080425**	0.326023	0.236400
	1st quartile	0.369251	**0.081104**	0.331783	0.245755
	2nd quartile	0.371936	**0.081577**	0.348830	0.248966
	3rd quartile	0.377706	**0.081904**	0.349356	0.251274
	Maximum	0.382063	**0.082114**	0.360301	0.259832
Validation	Mean	0.9335	**0.4718**	0.6671	0.6404
	Standard dev.	**0.0293**	0.0791	0.0477	0.0432
	Minimum	0.8881	**0.3598**	0.6215	0.6069
	1st quartile	0.9149	**0.4348**	0.6355	0.6112
	2nd quartile	0.9369	**0.4737**	0.6441	0.6336
	3rd quartile	0.9476	**0.5037**	0.6954	0.6446
	Maximum	0.9820	**0.6374**	0.7740	0.7522

Comparison of validation loss between converged models (after the last epoch of training) is presented in Figure 7. We can see that PMW-query and PMW-mean provides consistently lower loss compared to AIM with fixed ResNet weights. AIM model when allowed to train the image encoder weights, shows high variation in loss, however in most cases still reaches lowest loss from the tested models.

When comparing both proposed models we can see that PWM-query on average reaches slightly better validation loss and shows more consistent results, see Table 1. This can be explained by the differences in their architectures. PMW-mean averages latent feature matrix to extract feature vector and thus loses some of learned information in the process. PMW-query uses attention with trainable query, this gives a more universal way of extracting latent information for given task.

Next, we test if results of converged models are statistically different on validation dataset. Shapiro-Wilk test is used to test if results in each group are normally distributed. Zero hypothesis that the data is normally distributed is rejected for PMW-mean and PMW-query groups, however accepted for AIM-fixed and AIM-not-fixed models, see Table 2. Since not all groups are not normally distributed, we test if group medians are different. We use Moods median test, to check if samples come from populations with the same median and Kruskal-Wallis H-test to tests the null hypothesis that the population median of all of the groups are equal. When comparing PMW-mean and PMW-query groups, both tests results

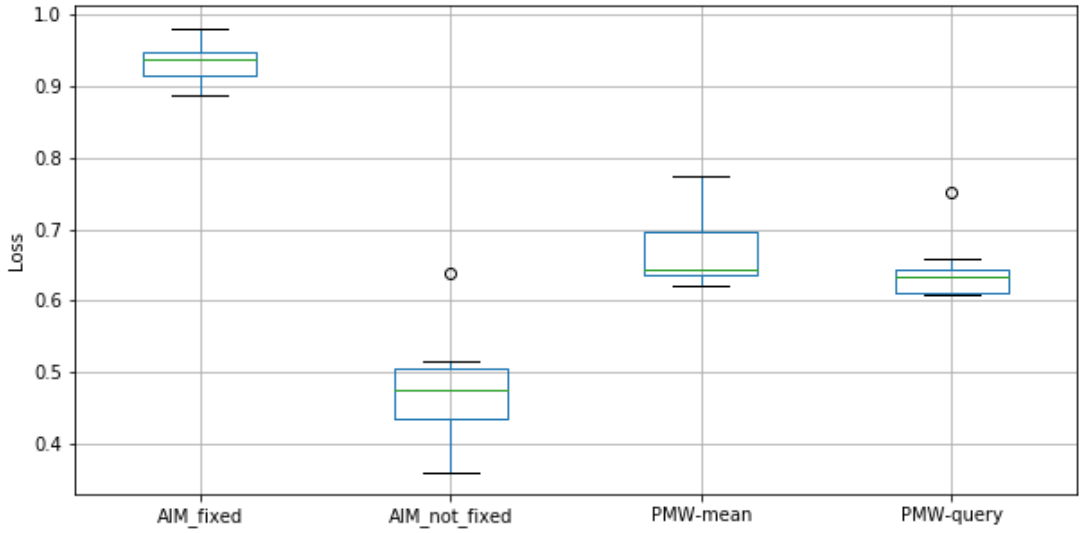

Figure 7: Comparison of validation loss between models after the 15th epoch of training

in P-value greater than 0.05, therefore we cannot reject the null hypothesis and conclude that PMW-mean and PMW-query medians are not statistically significantly different. However, test shows that difference between medians in proposed models (PMW-mean and PMW-query) and baseline models (AIM-fixed, AIM-not-fixed) is statistically significant. Both proposed models outperforms AIM-fixed baseline model. Since there is no proof of median difference between proposed models, to reduce the amount of computations, we choose only a single model - PMW-query - for test runs in Carla environment.

Table 2: Statistical tests and results for losses from PMW-mean and PMW-query models

Test	Groups	P-value
Shapiro-Wilk	PMW-mean	**0.0399**
	PMW-query	**0.0025**
	AIM-fixed	0.9566
	AIM-not-fixed	0.4566
Moods median	AIM-fixed, PMW-mean	**5.7e-05**
	AIM-fixed, PMW-query	**5.7e-05**
	AIM-not-fixed, PMW-mean	**0.0017**
	AIM-not-fixed, PMW-query	**0.0017**
	PMW-mean, PMW-query	0.6547
Kruskal-Wallis H-test	AIM-fixed, PMW-mean	**0.0002**
	AIM-fixed, PMW-query	**0.0002**
	AIM-not-fixed, PMW-mean	**0.0003**
	AIM-not-fixed, PMW-query	**0.0007**
	PMW-mean, PMW-query	0.1305

4.3 Runs on Carla environment

To further evaluate performance of models, test runs were made in simulated Carla environment. Weights with lowest validation error are chosen for the evaluation for each model. During the test runs, ego vehicle is required to reach destination without infractions and within time limit. Average results over the 10 routes of Town 5 are presented in Table 3. Metrics are described in depth in section 3.2. Similarly to the results on validation dataset, we can see that proposed PMW-query model outperforms AIM-fixed model in all metrics. As expected, fine-tuning helps AIM model (AIM-not-fixed) to reach best results on evaluation and heavily outperforms proposed model on Road completion and Driving score metrics. In contrast, considering infraction penalty PMW-query model shows comparable results to AIM-not-fixed. This indicates, that even without fine-tuning PMW model already learns good image embeddings, which allows to reduce overall amount of infractions.

Table 3: Evaluation results on Carla simulated environment

Model	Avg. driving score	Avg. route completion	Avg. infraction penalty
AIM-fixed	18.69	24.53	0.7994
AIM-not-fixed	**52.43**	**61.83**	**0.9059**
PMW-query	26.32	30.18	0.9027

In simulated environment most commonly ego vehicle does not finish the route due to stopping and not starting to drive, see Figure 8 for examples. This could indicate dataset bias, or lack of visual queues from traffic lights due to their distance from the vehicle as discussed in [16]. However, this issue still needs to be investigated further.

Figure 8: Example images where ego vehicle (gray) correctly passes intersection (1st and 2nd image from the left) and where it fails to start driving (3rd and 4th image). Examples are taken from test runs with PMW-query model.

Discussion. Proposed PMW encoders were not fine-tuned for autonomous driving task, however still showed to be capable of autonomous driving and on average completed around 30% of routes, with minimal amount of infractions. The performance would be expected to rise with fine-tuning and is left for the future work to explore. Furthermore, Perceiver originally was introduced as a model that is not dependant on data modality. While it produces good results on single modalities, it was shown that it is an efficient way to fuse modalities as well. Multi-modal learning was not addressed in this work due to computational complexity of the model and lack of resources, however it could be considered as a possible alternative to current state-of-the-art multi-modal model on Carla Leaderboard dataset [16] which uses image and lidar data.

5 Conclusions

In this work we have reviewed imitation learning algorithms for autonomous vehicles and presented state-of-the-art approaches. We have discussed the main problems in autonomous vehicles field, importance of image encoding algorithms and proposed Perceiver based approach as a possible solution to object detection related problems. We have implemented two variants of proposed model: PMW-mean and PMW-query. We have trained baseline and proposed models and ran the trained models in Carla simulated environment to evaluate their performance. Finally, analysis of the results shows that:

- Difference between validation loss medians from baseline AIM and proposed PMW-mean and PMW-query models is statistically significant. Proposed models do improve performance of autonomous vehicles when no fine-tuning is considered.
- Difference between validation loss medians of PMW-mean and PMW-query is not statistically significant and there is no proof that proposed query mechanism allows better convergence compared to averaging of latent space.
- Difference between medians of fine-tuned model and not fine-tuned models is statistically significant. Fine-tuning should be used to improve the models performance given that enough computational resources are available.
- Evaluation on runs in Carla environment shows that while PMW-query is heavily outperformed by fine-tuned AIM model in route completion and driving score metrics, it shows comparable results on infraction penalty metric. This indicates, that PMW model already learns sufficient image embeddings, which allows to reduce overall amount of infractions.

References

[1] Aseem Behl, Kashyap Chitta, Aditya Prakash, Eshed Ohn-Bar, and Andreas Geiger. Label efficient visual abstractions for autonomous driving. *CoRR*, abs/2005.10091, 2020.

[2] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *CoRR*, abs/1604.07316, 2016.

[3] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Conference on Robot Learning*, pages 66–75. PMLR, 2020.

[4] Felipe Codevilla, Matthias Müller, Alexey Dosovitskiy, Antonio M. López, and Vladlen Koltun. End-to-end driving via conditional imitation learning. *CoRR*, abs/1710.02410, 2017.

[5] Felipe Codevilla, Eder Santana, Antonio M. López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. *CoRR*, abs/1904.08980, 2019.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[7] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J. Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver IO: A general architecture for structured inputs & outputs. *CoRR*, abs/2107.14795, 2021.

[8] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver: General perception with iterative attention. *CoRR*, abs/2103.03206, 2021.

[9] Sampo Kuutti, Richard Bowden, Yaochu Jin, Phil Barber, and Saber Fallah. A survey of deep learning applications to autonomous vehicle control. *CoRR*, abs/1912.10773, 2019.

[10] Zhihao Li, Toshiyuki Motoyoshi, Kazuma Sasaki, Tetsuya Ogata, and Shigeki Sugano. Rethinking self-driving: Multi-task knowledge for better generalization and accident explanation ability. *CoRR*, abs/1809.11100, 2018.

[11] Urs Muller, Jan Ben, Eric Cosatto, Beat Flepp, and Yann L Cun. Off-road obstacle avoidance through end-to-end learning. In *Advances in neural information processing systems*, pages 739–746. Citeseer, 2006.

[12] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *arXiv preprint arXiv:1811.06711*, 2018.

[13] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA ARTIFICIAL INTELLIGENCE AND PSYCHOLOGY . . . , 1989.

[14] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[15] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Supplementary material for multi-modal fusion transformer for end-to-end autonomous driving.

[16] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. *CoRR*, abs/2104.09224, 2021.

[17] Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. No-regret reductions for imitation learning and structured prediction. *CoRR*, abs/1011.0686, 2010.

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[21] Qing Wang, Long Chen, and Wei Tian. End-to-end driving simulation via angle branched network. *CoRR*, abs/1805.07545, 2018.

[22] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. End-to-end learning of driving models from large-scale video datasets. *CoRR*, abs/1612.01079, 2016.