A Comparison of Imitation Learning Pipelines for Autonomous Driving on the Effect of Change in Ego-vehicle

Noorsyamimi Abdur Ajak[1,2], Wee Hong Ong[1,2], and Owais Ahmed Malik[2]

***Abstract*— This paper presents a comparison of the effect of change in ego-vehicle in two different pipelines of imitation learning for autonomous driving (AD) between direct control-based and waypoint-based pipelines. Control-based pipeline involves predicting control signals directly to control the car, whereas a waypoint-based pipeline predicts the future trajectory of the car and uses a controller module to generate the control signals from the predicted waypoints. In this study, CIL++ was used for the control-based method whereas TransFuser was used for the waypoint-based method. In our experiments, we used CARLA simulator and deployed both imitation learning models, without retraining or re-tuning the controller parameters, on various cars different from the car used during training. We used Town05 from CARLA's Leaderboard benchmark to evaluate the performance based on driving score, the main metric used in the benchmark. Based on the experiment results, TransFuser is more robust in adapting to different ego-vehicles than CIL++. TransFuser performed better when deployed to different vehicles. However, the performance still suffered when there was a significant change in the car classes. The source code of this work is made publicly available at https://github.com/ailabspace/Comparison-of-Autonomous-Driving-IL-Pipeline-for-Ego-Vehicle-Changes.**

I. INTRODUCTION

Autonomous driving (AD) often refers to a system that enables a vehicle to drive by itself and navigate without human intervention. It holds the potential to transform the future of the transportation industry by enhancing safety, efficiency, and accessibility. Nowadays, there are various solutions to achieve AD, among them are classical modular approach and modern end-to-end learning. End-to-end learning or learning-based approach in which an ego-vehicle utilizes deep learning algorithms to control the car is the current trend for AD solution. Among different learning-based approaches, imitation learning is the most widely used. Imitation learning is when an agent tries to learn to mimic an expert demonstrator to do a specific task or action. For instance, an agent is learning to drive a car from the observation data of an expert driver. For imitation learning, there are several pipelines to achieve AD, each of them having different input and output modalities and even, neural network architecture. This paper will focus on comparing two different pipelines of imitation learning: (1) **Control-based imitation learning model** where the model learns to predict vehicle controls directly (2) **Waypoint-based imitation learning model** where the model predicts a sequence of waypoints or future trajectory of the ego-vehicle but to control the ego-vehicle, a controller like PID (Proportional-Integral-Derivative) controller, is used to generate the control signals. In this study, we have chosen CIL++ [1] and TransFuser [2] as the representative work for control-based and waypoint-based imitation learning approaches respectively. In their original studies, both models were tested on the default ego-vehicle (Lincoln MKZ 2017), the same car used to record the training data. We wondered what would happen if we changed the ego-vehicle to a different vehicle with different dynamics, without retraining with new training data. Which imitation learning pipeline is more robust to change in the ego-vehicle? Thus, in this paper:

- We evaluated both imitation learning pipelines, by training and testing using the default ego-vehicle in the same test conditions (same town, routes, scenario, and weathers) in CARLA [3].
- We conducted an empirical evaluation of the models obtained from the two imitation learning pipelines when deployed on new vehicle models of different car classes, which was not used for data collection and training.

II. LITERATURE REVIEW

End-to-end learning for autonomous driving (AD) has become an active research topic over the years, especially with the development of research tools and platforms like CARLA [3], a simulator that is built for autonomous driving research, which has allowed rapid growth in the field. Recent works of learning-based approach fall into two paradigms: (1) **Reinforcement learning** where this method lets the ego-vehicle learn from trial and error, exploring different actions and improving its driving policy. Significant works that utilize reinforcement learning for autonomous driving are ROACH [4], WOR [5], and Learning to Drive in a Day [6]. (2) **Imitation learning** is a method where the ego-vehicle learns from demonstrations of experts such as human drivers or privileged agents. This is supervised learning, where the model learns from the training data collected from the demonstrations. This method is the dominant paradigm in AD at the moment.

Existing imitation learning approaches are seen to be taking one of the two distinct learning pipelines in terms of output modality. One directly predicts control signals like steering angle, throttle, and brake as shown in Fig. 1, or another pipeline that predicts waypoints of the ego-vehicle as shown in Fig. 2. Besides the difference in the

[1]N. Abdur Ajak and W. H. Ong are with the Robotics and Intelligent Systems Lab (Robolab), School of Digital Science, Universiti Brunei Darussalam, Jalan Tungku Link, Brunei.

[2]All authors are with the School of Digital Science, Universiti Brunei Darussalam, Jalan Tungku Link, Brunei. syamimi.rajak@gmail.com, [weehong.ong, owais.malik]@ubd.edu.bn

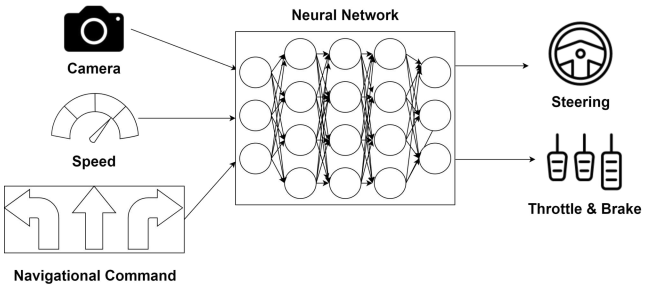

Fig. 1. Control-based imitation learning pipeline.

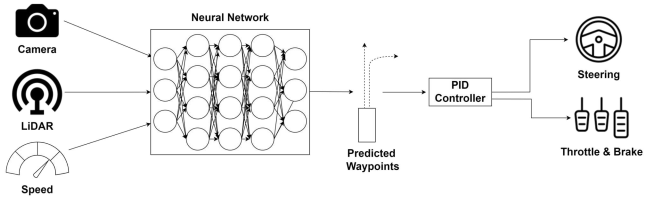

Fig. 2. Waypoint-based imitation learning pipeline.

output modalities of the imitation learning pipeline, there are also different input modalities used in both pipelines. Vision data is the most common input modality to train autonomous driving models. For example, CIL++ [1] uses multi-view cameras to perceive its surroundings, along with other measurement and navigational data to generate the control signals. Whereas, TransFuser [2] fuses both multi-view cameras and LiDAR information as their input representation for their transformer-based model. For direct control-based imitation learning pipeline, among the first to develop autonomous driving via end-to-end learning was DAVE-2 by NVIDIA [7] where they trained a model using convolutional neural network (CNN) to map image pixels to vehicle steering angle. Conditional Imitation Learning (CIL) [8] and its variants (CILRS [9], CIL++ [1]) learn to generate vehicles' control signals from the expert demonstrations on how to control the ego-vehicle based on what it had perceived. These variants use navigational commands like "go straight", "turn left", "follow lane" and "turn right" to direct the ego-vehicle on where to go and deep neural networks to generate the vehicle control signals to maneuver the car. Another significant work that predicts vehicle control signals based on the vehicle state is MILE [10] where their model learned from driving videos of an expert demonstrator to predict the vehicle actions. In all the works mentioned above, their models were evaluated with the same ego-vehicle used in the collection of the training data. Given that the training data is dependent on the physics of the vehicle used for data collection, we hypothesize the model will not perform well when deployed to a vehicle different from the vehicle used to collect the training data.

As for the waypoint-based imitation learning pipeline, this approach focuses on learning to generate a sequence of waypoints or future trajectories based on what it has seen. Subsequently, a controller module like a PID controller is used to generate the control signals from the predicted waypoint sequence. TransFuser [2] and its variant (TF++ [11]), NEAT [12], LAV [13] are among some of the works that predict waypoints to control the ego-vehicle. Compared to the control signals, the waypoint sequence is less dependent on the vehicle's physics. In addition to this, TCP [14] attempted to combine the two output modalities. It predicts both control and waypoints together to use the future trajectory as guidance to control the car.

For this study, we selected CIL++ [1] because the system is a good representation of the control-based algorithm as it is the current state-of-the-art for the control-based system. CIL++ is the latest best-performing model at the time of our work. As for waypoint-based, we selected TransFuser as an ideal representation of the waypoint-based system that is currently performing best. Even though TransFuser's successors, InterFuser [15] and ReasonNet [16] performed better than TransFuser in CARLA Online Leaderboard [17], they had added extra modules on top of the waypoint prediction which is not what we are interested to evaluate. TransFuser has a simpler system than the two models. What we are interested in evaluating is the impact on the performance of the imitation learning models based on waypoint-based against control-based pipelines when they are deployed on different cars. TCP [14] which combined both waypoint and control has not shown better performance than the waypoint-based model. Furthermore, most of the recent submissions on the CARLA Leaderboard are waypoint-based systems. However, none of the waypoint-based algorithms on the CARLA Leaderboard have been tested on a real car. On the other hand, one of the control-based approaches, CIL [8] has been deployed on a physical remote-controlled car. Thus, due to these reasons, we decided to select TransFuser [2] for the waypoint-based system and CIL++ [1] for the control-based system.

All the works in the literature have in common that they were tested on the same ego-vehicle that was used in training demonstration. In this work, we evaluated the performance of the two different imitation learning pipelines when the trained model was deployed on different ego-vehicles. The purpose is to see the impact on the performance when two models were deployed in car models different from the ego-vehicles used in collecting the training data.

III. LEARNING METHODS

In this section, we describe the two models evaluated in this work. In this study, for the waypoint-based pipeline, we have selected TransFuser [2] and for the control-based pipeline, Conditional Imitation Learning (CIL++) [1] was selected. Both models have distinct pipelines in terms of their neural network architecture, and input and output modalities.

A. TransFuser

TransFuser [2] introduced a novel method to integrate multiple sensor information from multi-view camera images and LiDAR by using Transformer architecture [18]. The model will then predict the sequence of waypoints or future

trajectories of the ego-vehicle to navigate safely around other dynamic agents to the destination while adhering to traffic rules. TransFuser addresses challenges in autonomous driving tasks by utilizing transformer-based sensor fusion techniques for improved performance and adaptability in complex driving scenarios.

TransFuser designed a rule-based expert agent that has access to privileged information from the simulator to control the vehicle for its training data. It is similar to CARLA traffic manager autopilot. The TransFuser system pipeline (Fig. 3) and the details are as follows:

Input and output: TransFuser uses multiple sensors which are RGB images of multi-view cameras and LiDAR bird-eye-view images for its inputs to the neural network. The image inputs are from three RGB cameras (left, front, and right). The point cloud data of LiDAR is transformed into a 2-bin histogram over a 2D bird-eye-view grid which results in a 3-channel bird-eye-view image, where in this image, the goal location information is included. As for the output, TransFuser generates a sequence of 2D waypoints (x,y) for four future timesteps.

Network architecture: ResNet [19] was used for feature extractions from the images from the RGB cameras (Image Branch) and LiDAR (BEV Branch). Self-attention structures of transformers are used to combine multi-modal information to generate a global 3D scene information of the environment. Next step, the information generated is used as inputs to GRU-based neural networks to predict the waypoint sequence of the ego-vehicle. Apart from learning to predict waypoints, TransFuser also incorporates multi-task learning to address the complex temporal and spatial scene configurations. The auxiliary tasks are depth prediction, HD map prediction, semantic segmentation, and vehicle detection.

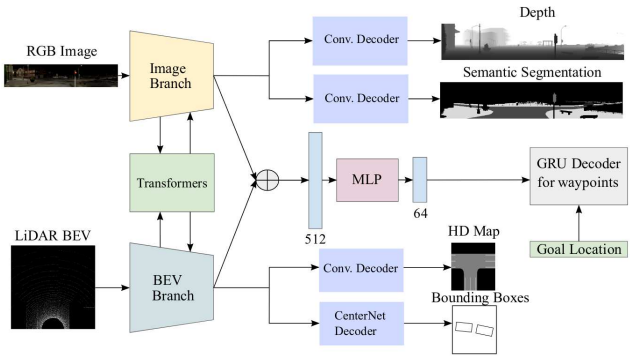

Fig. 3. The TransFuser system [2]

Controller: The predicted waypoint sequences are then fed into a classical PID controller (not shown in the figure) to generate control signals (steering angle, throttle and brake) to drive the vehicle. TransFuser utilizes two PID controllers for lateral and longitudinal controls. Apart from PID controllers, TransFuser applies a couple of rule-based controls, creeping and safety heuristics, to improve vehicle control.

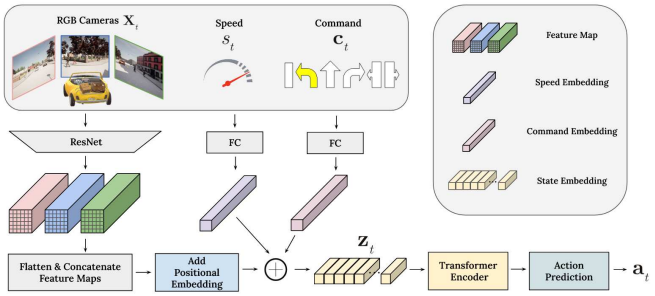

Fig. 4. The neural network architecture of CIL++ [1]

B. CIL++

Conditional Imitation Learning (CIL++) [1] is an imitation learning algorithm that directly predicts control signals (steering angle, throttle, and brake) of the vehicle. Comparing this model with TransFuser, this model pipeline does not need a PID controller to obtain the control signal values. For their training data, CIL++ uses ROACH [4], a reinforcement learning trained expert agent. The overview of the CIL++ system (Fig. 4) is as follows:

Input and output: For its input to the neural network, CIL++ uses multi-view RGB camera images (left, central and right) only. The model does not use LiDAR. Instead, it additionally uses measurement information like forward speed and navigational commands like ”follow lane”, ”turn right”, and ”change lane to left” as inputs. Navigational command acts as ”conditions” to direct the car on what to do in that instance. These commands are generated by a rule-based navigation module. The role of CIL++ is to generate control signals directly from the inputs with the neural network. Thus, the outputs are steering angle and acceleration. The acceleration value is used to derive the throttle and brake values. These outputs are enough to control the vehicle. Thus, it does not need a classical or rule-based controller to obtain the control signal values.

Network architecture: CIL++ also uses ResNet to extract features from the multi-view RGB images. Extracted features of the images are flattened and their feature maps are concatenated. Positional embedding is applied to the resulting feature maps of the images. As for the measurement information inputs, the ego-vehicle’s forward speed and navigational command were subjected to linear projection using a fully connected layer respectively. The input representation for the transformer-based neural network is the state embedding which is comprised of the concatenation of processed input information earlier which are forward speed, navigational command, and the set of images from the multi-view cameras. Like TransFuser, CIL++ also uses the attention mechanism of transformers to combine multiple information. The transformers’ output consequently led to predictions of the actions, which are the control signals. This imitation learning pipeline does not incorporate any auxiliary tasks for its model to learn. Hence, it is more lightweight than TransFuser.

IV. EXPERIMENTS

In this section, we describe the experiments conducted. We used CARLA, an open-source simulation for autonomous driving research that provides a platform for data collection, testing, and evaluation.

Fig. 5. Car models used for the experiments [3]

A. Car Models

In this work, we selected five car models of different types like sedan, compact, and SUV as shown in Fig. 5 to compare the performance between TransFuser and CIL++ when deployed to cars different from the default ego-vehicle used in the training demonstration. With respect to the car types, the selected car models are as follows: (1) 4-door Sedan: Lincoln MKZ (used in training) and Tesla Model 3. (2) 4-door Compact: Citroen C3 and Toyota Prius. (3) Sport utility vehicle (SUV): Nissan Patrol. Each car has different vehicle physics and dynamics. The details are given in tables Table I and Table II. All information was obtained from CARLA. For vehicle size, there is no exact information available. For this, we have used the bounding box size of each car model to report its size.

TABLE I
VEHICLE SIZE AND MASS

Car Model	Bounding box size (m) $(X \times Y \times Z)$	Mass (kg)
Lincoln MKZ	$4.90 \times 2.13 \times 1.51$	2404
Tesla Model3	$4.79 \times 2.16 \times 1.49$	1845
Citroen C3	$3.99 \times 1.85 \times 1.62$	1365
Toyota Prius	$4.51 \times 2.01 \times 1.52$	1775
Nissan Patrol	$4.60 \times 1.93 \times 1.85$	2355

B. Evaluation

We used the CARLA Offline Leaderboard as a benchmark to evaluate the performance of TransFuser and CIL++ on different car models in completing the tasks. CARLA Offline Leaderboard comprises a list of routes and scenarios of multiple towns for its training and testing set. From the testing list, we have used 10 routes in Town05 in our experiment. Town05 is an urban environment with many wide multi-lane roads and intersections and it is linked to the elevated highway network which also functions as a ring road. This is a suitable test environment for both imitation learning pipelines given the complexity of the environment. Each run of each car was tested in the same test conditions of the same set of routes and weather settings. Basically, each run had one set of experiments comprised of five selected cars, with all running the same route set and weather conditions. Each test set was conducted on the two models. We run the evaluation with three runs for each experiment set and report the average in the results. Each model was tested on multiple car models with no retraining of the imitation learning models or re-tuning of the parameters for the classical controller used.

TABLE II
VEHICLE PHYSICS AND DYNAMICS

Car Model	Maximum RPM	Forward gear ratio	Wheel radius (cm)
Lincoln MKZ	5800	3.59	35.5
Tesla Model3	15000	2.50	37.0
Citroen C3	3750	3.46	34.0
Toyota Prius	5200	2.60	37.0
Nissan Patrol	3600	4.56	39.0

We note that the comparison is not about the driving performance of TransFuser versus CIL++. The comparison is about the impact on the performance of each imitation learning model when the ego-vehicle is different from that used in collecting training data.

C. Metrics

To compare the two imitation learning pipelines across different car models, we used the scoring metrics set utilized by the CARLA Leaderboard for evaluation: route completion (RC), infraction score (IS), and driving score (DS). Route completion is the percentage of route distance covered by an ego-vehicle. The infraction score compiles penalty scores for different types of infractions caused by the ego-vehicle, each infraction with its own specific penalty score, and aggregates them using a geometric series. As for the driving score, it is the product of route completion and infraction score and this will be the main metric to evaluate the driving performance of an ego-vehicle.

V. RESULTS & DISCUSSION

TABLE III
CIL++ RESULTS

Car model	Driving score (DS) ↑	Route completion (RC) ↑	Infraction score (IS) ↑
Lincoln MKZ	60.73 ± 2.57	88.56 ± 1.70	0.67 ± 0.02
Tesla Model3	3.73 ± 0.79	18.43 ± 3.06	0.32 ± 0.07
Citroen C3	4.02 ± 0.65	10.80 ± 1.32	0.39 ± 0.01
Toyota Prius	4.12 ± 0.37	11.55 ± 0.27	0.40 ± 0.00
Nissan Patrol	2.53 ± 0.65	7.61 ± 1.64	0.39 ± 0.02

The results are presented in Fig. 6 to Fig. 8 and Table III to Table V. For CIL++, as we can see from both Table III

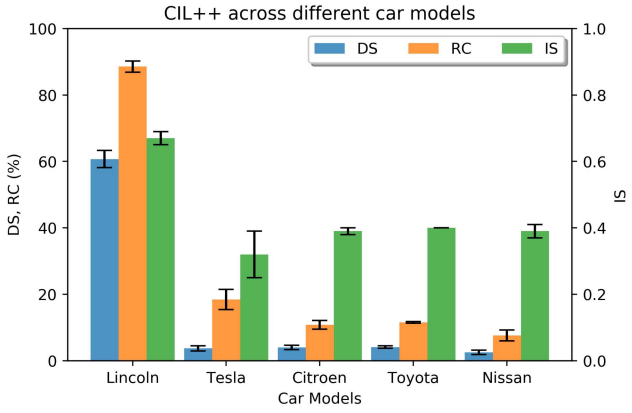

Fig. 6. CIL++ results of different car models

and Fig. 6, the Lincoln MKZ, the default ego vehicle has achieved the highest values in all the metrics; driving score, route completion, and infraction score. This was expected because the training data for the CIL++ model was acquired using the Lincoln MKZ and the same car was used for evaluation in the test conditions. Once we changed the vehicle to a car different from Lincoln MKZ, the model failed to control the car. This happened to all car models, including cars of similar car type as Lincoln MKZ, which is a sedan. This can be seen from Fig. 6, where the driving score for all vehicles other than the ego-vehicle was consistently less than 5% (decreased by more than 50%) which was contributed by the low values of route completion and infraction score.

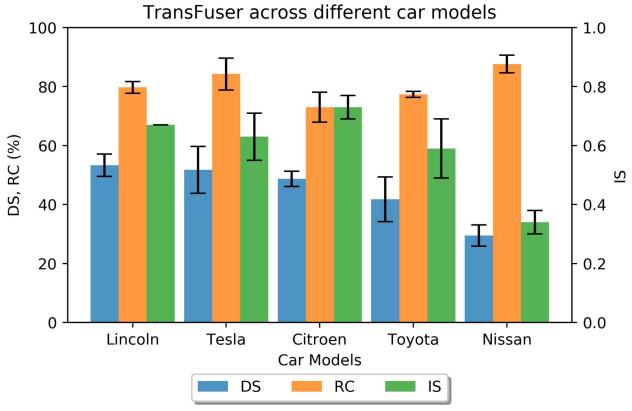

Fig. 7. TransFuser results of different car models

As for TransFuser, similar to CIL++, the driving score of the Lincoln MKZ was the highest compared to the rest of the vehicles as shown in Fig. 7 and Table IV. However, the driving score across different car models was more stable than CIL++ with only 5%-25% lower than with the Lincoln MKZ. Sedan cars like Lincoln MKZ and Tesla Model3, have quite similar driving score with only 2% difference. When we used the same TransFuser model on compact cars, Citroen C3 and Toyota Prius, the driving score decreased but by only less than 5% difference. However, if the car features are

TABLE IV
TRANSFUSER RESULTS

Car model	Driving score (DS) ↑	Route completion (RC) ↑	Infraction score (IS) ↑
Lincoln MKZ	53.29 ± 3.78	79.72 ± 2.00	0.67 ± 0.00
Tesla Model3	51.77 ± 7.94	84.26 ± 5.41	0.63 ± 0.08
Citroen C3	48.68 ± 2.57	73.03 ± 5.09	0.73 ± 0.04
Toyota Prius	41.77 ± 7.58	77.33 ± 1.02	0.59 ± 0.10
Nissan Patrol	29.51 ± 3.60	87.64 ± 3.01	0.34 ± 0.04

significantly different as in the case of an SUV car, Nissan Patrol, the driving score declined by 25% when compared to the training ego-vehicle's driving score.

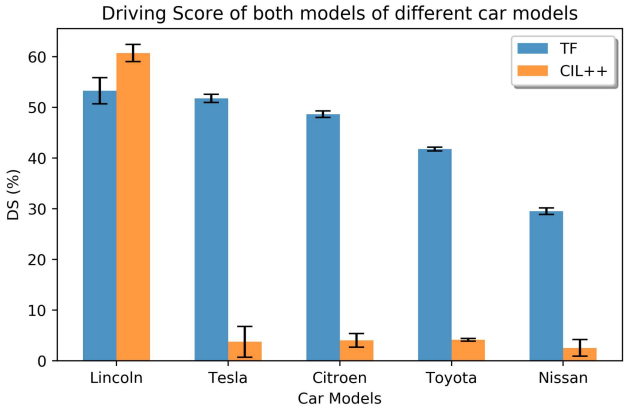

Fig. 8. Driving score of two imitation learning models

TABLE V
DRIVING SCORE (DS) OF TRANSFUSER AND CIL++

Car model	TransFuser	CIL++
Lincoln MKZ	53.29 ± 3.78	60.73 ± 2.57
Tesla Model3	51.77 ± 7.94	3.73 ± 0.79
Citroen C3	48.68 ± 2.57	4.02 ± 0.65
Toyota Prius	41.77 ± 7.58	4.12 ± 0.37
Nissan Patrol	29.51 ± 3.60	2.53 ± 0.65

Fig. 8 and Table V compare the driving scores of the two imitation learning models when deployed on each of the cars. From Fig. 8, when deployed on the same ego-vehicle as the trained model, CIL++ performed better than TransFuser as seen by its higher driving score. However, CIL++ failed to control the car when it was deployed on a vehicle different from the ego vehicle it was trained on. TransFuser was more robust to vehicle model change as the waypoint prediction is less dependent on the vehicle's physics. Although it still suffered if there was a significant difference between the characteristics of the vehicle used in training the model and the vehicle where the model is deployed, the waypoint-based model is superior since the prediction provides an intermediate representation of the driving task, and this is less dependent on the specific dynamics of the training vehicle compared to the direct

prediction of control signals, as observed in the control-based model. Essentially, this approach focuses on reaching the destination rather than executing specific maneuvers. The generation of control signals for maneuvering is left to the classical PID controller not fitted to the ego-vehicle in the training data. The PID controller can potentially be tuned to the physical properties of a vehicle irrespective of traveling routes.

VI. CONCLUSIONS

Waypoint-based imitation learning pipeline is more robust than a control-based pipeline in learning an autonomous driving agent for deployment to different car models. Nevertheless, the performance of the waypoint-based model still deteriorates when there is a significant difference in the vehicle's physical properties. The potential solution to this problem could be tuning PID parameters for each vehicle model while maintaining the rest of the imitation learning pipeline components. Another solution could be replacing classical PID controllers with another approach such as learning-based controllers. The fine-tuning of PID controllers should require significantly less data and computation than re-training the whole pipeline in the case of a control-based approach.

REFERENCES

[1] Y. Xiao, F. Codevilla, D. Porres, and A. M. López, "Scaling vision-based end-to-end autonomous driving with multi-view attention learning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 1586–1593.

[2] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, "Transfuser: Imitation with transformer-based sensor fusion for autonomous driving," *Pattern Analysis and Machine Intelligence (PAMI)*, 2023.

[3] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[4] Z. Zhang, A. Liniger, D. Dai, F. Yu, and L. Van Gool, "End-to-end urban driving by imitating a reinforcement learning coach," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 15 222–15 232.

[5] D. Chen, V. Koltun, and P. Krähenbühl, "Learning to drive from a world on rails," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 590–15 599.

[6] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8248–8254.

[7] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[8] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 4693–4700.

[9] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9329–9338.

[10] A. Hu, G. Corrado, N. Griffiths, Z. Murez, C. Gurau, H. Yeo, A. Kendall, R. Cipolla, and J. Shotton, "Model-based imitation learning for urban driving," *Advances in Neural Information Processing Systems*, vol. 35, pp. 20 703–20 716, 2022.

[11] B. Jaeger, K. Chitta, and A. Geiger, "Hidden biases of end-to-end driving models," in *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2023.

[12] K. Chitta, A. Prakash, and A. Geiger, "Neat: Neural attention fields for end-to-end autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 793–15 803.

[13] D. Chen and P. Krähenbühl, "Learning from all vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 222–17 231.

[14] P. Wu, X. Jia, L. Chen, J. Yan, H. Li, and Y. Qiao, "Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline," *Advances in Neural Information Processing Systems*, vol. 35, pp. 6119–6132, 2022.

[15] H. Shao, L. Wang, R. Chen, H. Li, and Y. Liu, "Safety-enhanced autonomous driving using interpretable sensor fusion transformer," in *Conference on Robot Learning*. PMLR, 2023, pp. 726–737.

[16] H. Shao, L. Wang, R. Chen, S. L. Waslander, H. Li, and Y. Liu, "Reasonnet: End-to-end driving with temporal and global reasoning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 723–13 733.

[17] "Carla autonomous driving leaderboard," https://leaderboard.carla.org/leaderboard/, (Accessed on 01/16/2024).

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.