Transformer-Based Domain Knowledge Transfer for End-to-End Autonomous Driving

Zeyu Dong[1], Yimin Zhu[2], Kevin Mahon[3], and Yu Sun[4]

***Abstract*— End-to-end autonomous driving focuses on training neural networks to directly output low-level control signals, reducing the need for intermediate environment labels. However, a significant challenge lies in ensuring these models generalize across diverse environments without requiring vast amounts of real-world driving data. Vehicle simulation like CARLA offers a cost-effective solution, generating large-scale, diverse datasets that include edge-case driving scenarios. Yet, transferring models trained in simulation to real-world environments remains difficult due to the gap between simulated and real-world data. This paper addresses that challenge, demonstrating that the Transformer architecture facilitates domain knowledge transfer from simulation to reality with minimal real-world data requirements. We use obstacle avoidance, a critical case in urban driving, to validate our approach. Collecting real-world driving data for all possible obstacle types is costly and impractical. To address this, we first pre-train a FastViT model on a large-scale, domain-randomized dataset generated in the CARLA simulator. We then fine-tune the model using a small, real-world obstacle avoidance dataset with limited obstacle types, freezing the pre-trained self-attention layers to aid in domain adaptation. Our real-world experiments show that the fine-tuned model successfully avoids unseen obstacle types that were not present in the real-world training dataset. Moreover, the pre-trained model demonstrates a significant boost in generalization compared to FastViT models without pre-training. Baseline models, such as ResNet and EfficientNet, even with pre-training, fail to generalize effectively, underscoring the importance of the Transformer architecture. These results confirm that pre-training a Transformer model in a simulation environment is crucial for successful domain knowledge transfer, enhancing the model's real-world performance in autonomous driving tasks.**

I. INTRODUCTION

Traditional autonomous driving follows a modular design, breaking down the task into components such as perception, prediction, planning, and control, where each module is trained separately using intermediate labels from the environment (e.g., lanes, traffic lights) or other vehicles (e.g., position, speed). In contrast, end-to-end autonomous driving uses a data-driven approach, training a neural network that takes sensor input and directly generates low-level trajectories or control signals from expert demonstrations, thereby removing the need for intermediate labels. Recently, end-to-end autonomous driving in simulators has achieved impressive results [1], [2], [3], as simulated environments can easily generate vast amounts of high-fidelity driving data, which are essential for data-intensive deep learning training tasks. However, challenges persist in real-world driving scenarios, where the scarcity of massive, high-quality datasets hinders the generalization of end-to-end models to complex road conditions, especially in unseen environments [4]. Domain adaptation aims to solve this by transferring driving knowledge across different domains, such as from simulation to reality. Unfortunately, directly transferring a driving policy from simulation to the real world is not feasible [5]. Most studies [6], [7], [8] on domain adaptation focus on translating domain-specific data or using deep learning methods to identify domain-invariant features within real-world object embeddings. These approaches, however, often involve intricate modeling of specific target domains, limiting their generalizability.

The Transformer architecture [9] has been adopted to address the generalization challenges. When trained on large-scale datasets from different domains, Transformers excel in generalizing across multiple datasets [10]. This creates new possibilities for domain adaptation in end-to-end autonomous driving. By pre-training on large-scale simulation datasets, Transformers can transfer knowledge from simulated environments to real-world scenarios.

In this paper, we demonstrate that pre-training Vision Transformers on large-scale simulation datasets with domain randomization [11] improves the model's ability to generalize in real-world end-to-end autonomous driving. Our focus is on a critical safety problem in urban driving: obstacle avoidance. In urban environments, encountering previously unseen obstacles, such as new types of objects, is inevitable. Human drivers can easily recognize various objects, like stationary or moving vehicles, as well as pedestrians in different actions (e.g., resting, walking, running). However, unknown obstacles with different characteristics can confuse autonomous systems. End-to-end models struggle to generalize to new environmental conditions, as it is impractical to label and train models for every possible obstacle [12]. Simulation environments help overcome this challenge by generating diverse and extensive datasets that cover a wide range of scenarios and obstacles.

We collect datasets using the open simulator CARLA [13], applying domain randomization to variables like town maps, obstacle types, and weather conditions. We use the FastViT model [14], a variant of (Vision Transformer) ViT [10]

[1]Zeyu Dong is with the Department of Applied Math and Statistics, Stony Brook University, 100 Nicolls Road, Stony Brook, NY 11794, USA Zeyu.Dong@stonybrook.edu

[2]Yimin Zhu is with the Department of Computer Science, Stony Brook University, 100 Nicolls Road, Stony Brook, NY 11794, USA yimzhu@cs.stonybrook.edu

[3]Kevin Mahon is with Sunrise Technology Inc., 1500 Stony Brook Rd, Stony Brook, NY 11790, USA k.mahon.dev@gmail.com

[4]Yu Sun is with Sunrise Technology Inc., 1500 Stony Brook Rd, Stony Brook, NY 11790, USA sunrisetechnology001@gmail.com

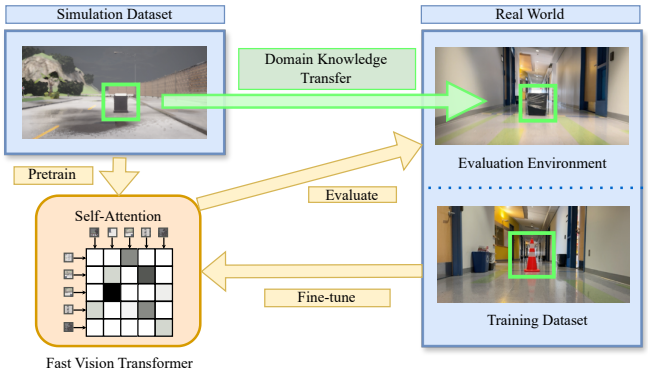

Fig. 1. Pretraining-finetuning on the vision transformer transfers the domain knowledge from simulation to reality.

optimized for speed, which runs efficiently on embedded platforms. FastViT uses a typical transformer architecture, with self-attention layers to transfer knowledge across various domains, including natural language processing [15], [16] and computer vision [10].

The self-attention mechanism plays a critical role in learning and transferring generic knowledge. Therefore, freezing pre-trained self-attention layers during fine-tuning enhances the model's ability to transfer knowledge to different domains [17]. Consequently, we apply this strategy, freezing the self-attention layers during fine-tuning to facilitate smoother domain knowledge transfer from the simulated environment to the real-world driving scenario.

Figure 1 illustrates the key idea of cross-domain transfer. We first pre-train the FastViT model on a large-scale simulated dataset generated by CARLA. Then, we fine-tune the model on a small real-world driving dataset, with the attention layers frozen. As shown in Figure 1, while the real-world dataset does not include examples of avoiding a trash can, the model still guides the vehicle to avoid it, as shown in the top left of the figure.

Experimental results show that the FastViT model trained using the pretrain-then-finetune approach effectively transfers obstacle avoidance knowledge from the simulated environment to the real world. This allows our model to generalize to various real-world obstacle types. We compared the FastViT model pre-trained on the CARLA dataset with a model trained solely on real-world data. The pre-trained model significantly outperformed the one trained without pre-training. Additionally, we included baselines using ResNet [18] and EfficientNet [19] with the same pretrain-then-finetune approach. Our results show that Transformers are crucial for effective domain knowledge transfer.

In summary, our contributions are as follows:

- We pre-train the FastViT model on the large-scale CARLA simulation dataset and then fine-tune it on a small-scale real-world dataset. This approach effectively transfers obstacle avoidance knowledge from the simulation to the real world, bridging a significantly large domain gap.
- We compare the performance of FastViT with ResNet and EfficientNet as baselines, demonstrating that the Transformer architecture is crucial for effective domain knowledge transfer from simulation to the real world.
- We evaluate FastViT's performance with and without pre-training on simulation data. The results show that pre-training on large-scale simulation datasets with domain randomization significantly enhances the model's generalization ability in real-world scenarios.

II. RELATED WORKS

A. End-to-end Autonomous Driving

Training end-to-end deep neural networks for autonomous driving presents the challenge of ensuring that models generalize well to out-of-sample distribution data. To address this, data augmentation techniques increase sensor data variety for robust driving [20]. Additionally, image translation is applied to balance the disparity between normal driving samples and drift recovery samples [21], [22], [23]. Another approach is to train the model on massive simulation data and transfer it to the real world. For instance, Müller et al. [5] trained a driving policy using modularity and abstraction of real-world scenes, such as segmentation and waypoints. However, this method differs from ours, as we rely on fine-tuning the network to align the simulation-trained model with real-world scenarios. In our approach, scene abstraction is learned implicitly during the pre-training and fine-tuning process, eliminating the need for manually crafted intermediate abstractions. Online learning algorithms, such as DAgger [24], are also adopted to collect data that lies outside of the expert distribution. For more complex driving tasks, driving simulators [13], [25], [26] produce large-scale datasets covering diverse driving scenarios. Another key challenge for robust end-to-end autonomous driving is ensuring that models generalize for similar types of tasks across different domains, e.g., varying surrounding environments, sensor characteristics, and task configuration. Several studies have been conducted for unseen road object detection and avoidance [6], [27] and driving in unseen weather conditions [28].

B. Domain Randomization

Domain Randomization is well-known as an effective approach to domain adaptation [11], [29], specifically, for transferring models trained in simulated environments to real-world driving tasks. This technique has been successfully applied in various areas, including obstacle detection in robotics [30]. For end-to-end autonomous driving, domain randomization has been shown to improve real-world trajectory planning to avoid collisions as demonstrated by the ROADS [12]. Combined with reinforcement learning, domain randomization further facilitates simulation-to-reality transfer [31].

C. Transformer and Variants

The Transformer architecture [9] has demonstrated remarkable potential across various fields, including Natural

Language Processing, Computer Vision, and Reinforcement Learning. In computer vision, ViT [10] achieved state-of-the-art results in the ImageNet challenge, showcasing the effectiveness of training Transformers on large-scale datasets for generalization across multiple image recognition tasks. In end-to-end autonomous driving, ViT has been utilized to predict steering angles and throttle values or future waypoints [32], while more advanced transformer architectures incorporating self-attention mechanisms have been adopted in end-to-end driving systems [3], [1], [2]. However, the quadratic time complexity of self-attention poses challenges for real-time, closed-loop inference in autonomous vehicles. To address these limitations, researchers such as Trockman et al. and Guo et al. have combined Transformers with Convolutional Neural Networks (CNNs) to enhance performance efficiency [33], [34]. More recently, FastViT [14], which integrates ConvMixer [33] with Transformer, has achieved a breakthrough in balancing performance and latency, making it a promising architecture for real-time computer vision tasks.

III. OUR APPROACH

In this section, we detail our approach to training an end-to-end autonomous driving model for obstacle avoidance and transferring it from simulation to the real world. To enable efficient sim-to-real transfer, we use a small real-world dataset to fine-tune the model pre-trained on simulation data. This approach significantly reduces the amount of real driving data needed to train a complex driving model.

A. Problem Statement

The goal of the end-to-end model is to safely navigate the vehicle in real-world traffic by effectively avoiding obstacles. However, due to the complexity of real-world environments, it is nearly impossible to generate test data diverse enough to ensure the safety of driving models in all conditions [35]. This issue frequently arises in real-world driving, where drivers encounter previously unseen obstacles. Therefore, a model that can generalize and effectively avoid obstacles is highly valuable.

Our goal is to train a generalizable model for monocular vision-based obstacle avoidance in real-world environments with various obstacle types. While camera-based models have shown promising results [21], [36], [37], monocular systems face significant challenges. These challenges include generalizing to obstacle types not seen during training and lacking 3D depth information. In our scenario, the model generates appropriate steering and throttle commands to guide the vehicle around obstacles, without hitting the side barrier, or the curb of the road.

B. Dataset

To bridge the gap between simulation and the real world, we start with a model that has been trained on a large dataset from simulations. To improve its performance in real-world conditions, we fine-tune the model using a smaller dataset of real-world examples. This helps the model better handle

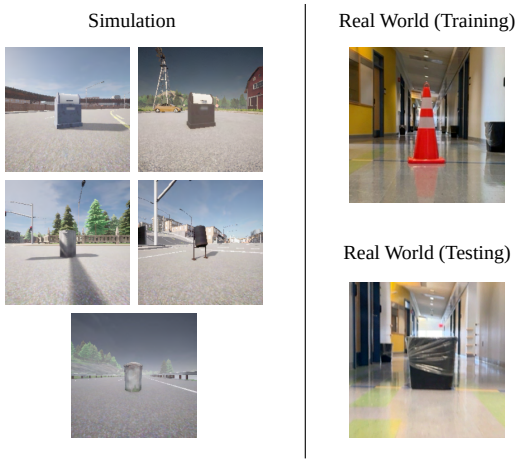

Fig. 2. Sample of different obstacle types used for training and testing.

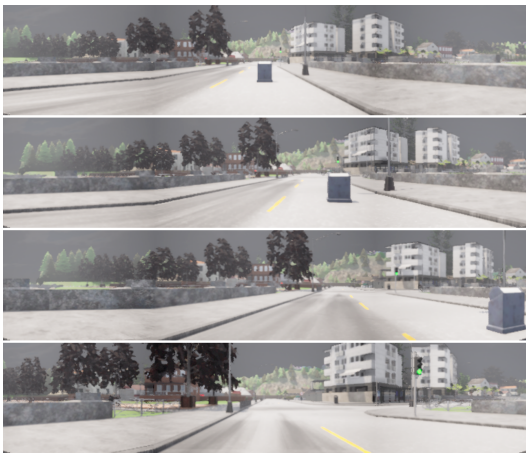

Fig. 3. Sample images in CARLA for obstacle avoidance.

the differences between simulations and actual environments. Simulation data is popular for its low training cost, and we use CARLA to generate the simulation dataset based on the scenario described in Section III-A. To mitigate differences between simulation and real-world settings, we apply domain randomization [11], which introduces variability to help extract domain-independent knowledge. The real-world dataset is collected by a human driver navigating obstacles in front of the ego vehicle in random locations of an office corridor, resembling the simulation. We limit the size and variety of obstacle types in the real-world dataset to verify the effectiveness of knowledge transfer from simulation.

C. Domain Randomization

1) Simulation Dataset: To ensure sufficient variation in the simulation data, we randomly select the driving map, weather conditions, time of day, and obstacle types for each run of the simulated scenario. The detailed environment settings are provided in Section IV-A.

The training dataset consists of approximately 250k front-view images from the ego vehicle, featuring five different

obstacle types. An example of different obstacles is shown in Figure 2. Figure 3 shows a sample scene from CARLA.

2) Real-world Dataset: In the real-world scene, we place obstacles in front of the ego vehicle with size and position similar to those in the simulated field of view. To introduce randomization, obstacles are placed in varying locations within the building.

The training dataset consists of approximately 5k images with a single obstacle type shown in Figure 2. For closed-loop evaluation in the real world, we use different obstacle types to assess the model's generalization ability. Detailed settings for real-world data collection and evaluation are provided in Section IV-B.

D. Model Architecture

The model utilizes, FastViT [14], a Transformer-based image encoding backbone and a decoder with fully connected layers for predicting steering and throttle. The image processing backbone acts as the "eyes and brain" of the autonomous system, enabling it to perceive the environment, identify objects, recognize signs, and make real-time decisions. Advances in deep learning and neural networks have transformed image processing, allowing vehicles to learn and adapt to dynamic road conditions. Figure 4 provides an overview of the model architecture.

1) FastViT Backbone: The ViT model [10] has demonstrated exceptional performance in generalizing and transferring across multiple tasks on large image datasets. However, due to its $O(N^2)$ time complexity for self-attention, where N represents the number of image patches, ViT becomes impractical for real-time inference on embedded systems compared to CNN models. To address this, we adopted FastViT, a hybrid model that combines the strengths of CNNs and Transformers, reducing memory access requirements and ensuring real-time inference on embedded platforms. FastViT achieves this by reparameterizing skip connections, replacing dense convolutions with linear train-time overparameterization, and computing self-attention token mixers using large-kernel convolutions for enhanced efficiency.

2) Steering Angle Prediction: To facilitate the transfer from the simulation environment to the real world, we use a multilayer perceptron to directly predict low-level hardware control signals–specifically, steering and throttle values. The steering value is normalized to the range $[-1, 1]$, while the throttle is normalized to $[0, 1]$. Compared to other low-level control signals, such as waypoints, this approach requires less tuning of domain-specific parameters.

E. Performance Metric

The loss function used for model training is the mean squared error between the predicted and ground truth values for both steering and throttle signals.

$$Loss = \frac{1}{n}\sum_{i=1}^{n}(y_s - \hat{y}_s)^2 + (y_t - \hat{y}_t)^2,$$

where y_s and $\hat{y_s}$ are actual and predicted steering angles, y_t and $\hat{y_t}$ are real and predicted throttles and n is the batch size.

We evaluate the model through closed-loop experiments in the real world, measuring the success rate by calculating the percentage of successful obstacle bypasses out of all attempts.

F. Training Procedure

We employ pre-training and fine-tuning in model training, a widely adopted approach for transformer-based models in both natural language processing [15], [16] and computer vision [10]. Fine-tuning helps align the model from the simulation environment to real-world scenarios, adjusting factors such as steering and throttle scales, while preserving the image and scene understanding capabilities learned during simulation.

1) Pre-training: By pre-training the FastViT model on the CARLA dataset with domain randomization across various obstacle types, the model becomes more adaptive at detecting and avoiding multiple obstacle types. We use Adam [38] as the optimizer, training the model for 100 epochs with a learning rate of 10^{-4}. After pre-training, we assess the model's performance in the CARLA simulator to determine the optimal epoch for the fine-tuning phase.

a) Fine-tuning: After pre-training the transformer on the large-scale CARLA dataset with domain randomization, we fine-tune the model on a small real-world dataset containing only one obstacle type. Our goal is to determine whether the model can generalize to new obstacle types without additional training on specific real-world obstacles. To maintain the model's ability to perceive the environment and identify objects, we apply the strategy from [17] by freezing the self-attention layers, which is crucial for learning generic knowledge and facilitating effective cross-domain transfer. We use the Adam optimizer with a batch size of 64, a learning rate of 10^{-4}, and train the model for 5 epochs.

IV. EXPERIMENTS

We conducted two sets of experiments to study the effectiveness of pre-training and the FastViT model. We use the CNN-based models, EfficientNetB4 and ResNet23, as the baseline. In the first experiment, all models were pre-trained using the CARLA dataset, and then fine-tuned on a real-world dataset to evaluate its capability for real-world adaptation. We labeled the pre-trained models with postfix PT (pre-training), i.e., FastViT+PT, EfficientNetB4+PT, and ResNet34+PT. In the second experiment, models were trained directly on the real-world dataset without pre-training, labeled with postfix NOPT (no pre-training), i.e., FastViT+NOPT, EfficientNetB4+NOPT, and ResNet34+NOPT.

A. CARLA Simulator Settings

To ensure sufficient domain randomization during data collection, we randomly selected the following configuration in each simulation run. The identifiers below are pre-defined labels in the CARLA simulator.

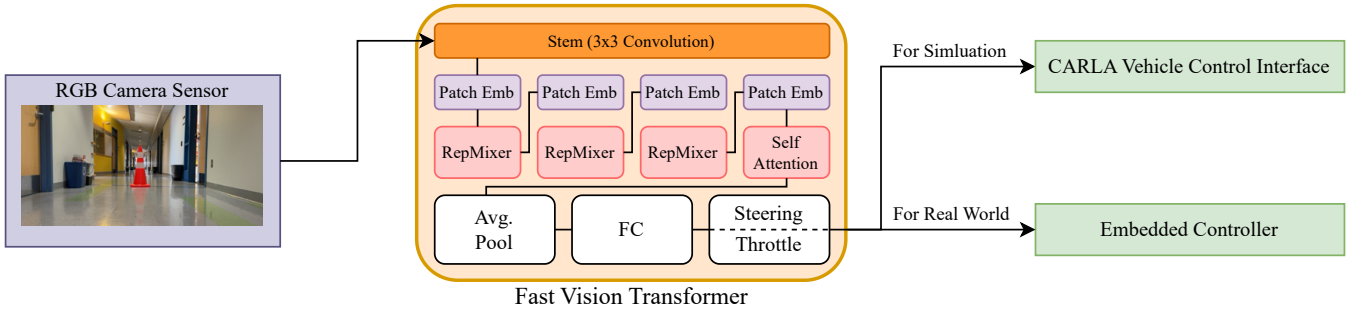

Fig. 4. The FastViT [14] model modified for steering wheel prediction with monocular RGB camera sensor as input. The input sensor image goes through four stages of token mixers and self-attention layers to produce the feature extraction of the current frame. The image feature then goes to a fully connected layer to output the steering and throttle signal.

Fig. 5. An obstacle from the training obstacle set is set seven feet from the vehicle.

- Map: `Town01` to `Town06`;
- Weather: `Clear`, `Cloudy`, `Wet`, `WetCloudy`;
- Time: `Sunset`, `Morning`, `Noon`;
- Obstacles: `transhcan01` to `transhcan05`.

For the closed-loop evaluation, we use the same randomization configuration for Map, Weather, and Time. For obstacles, the training obstacle set consists of the same obstacles as mentioned above, and the testing obstacle set includes all supported obstacle types other than the training set. This setup evaluates the model's ability to generalize to unseen obstacles under the same environmental configurations.

B. Real-world Settings

In the real-world environments, we designed a custom self-driving vehicle using a Traxxas radio-controlled racecar as the chassis and an iPhone as the camera and model inference device. For each trial, to replicate the simulator's view, the vehicle is placed seven feet from the obstacle, as shown in Figure 5.

We use traffic cones as the training obstacle set, and trash bins and large containers as the testing obstacle set to simulate diverse real-world obstacles that differ in size and appearance. The training dataset was collected in the corridor of the office building, with the cones placed at random locations. To evaluate the performance of the model, the vehicle operated in a closed-loop in real-world obstacle avoidance scenarios on both training and testing obstacle sets at 17 locations. Each location featured a unique combination of background, surrounding objects, and lighting conditions. A successful run was defined as the ego vehicle goes around the obstable without collision or getting blocked.

C. Model Hyperparameter Configurations

To meet the needs for steering wheel prediction and inference in our mobile embedded systems, we modified the FastViT backbone hyperparameter settings. Specifically, our FastViT model contains four stages, incorporating the RepMixer block in the first three stages and self-attention block in the last stage. The number of layers in each stage is 4, 4, 12, and 4, respectively. The number of output features for each stage is 48, 96, 192, and 384, respectively. Under this setting, the total number of parameters in FastViT is 29M, which is comparable to ResNet34 (23M) and EfficientNetB4 (22M).

D. Performance and Analysis

Table I shows the results of the close-loop experiment in real-world scenarios. Among the pre-trained models, FastViT+PT performed the best on unseen obstacles types. This confirms FastViT's ability in sim-to-real transfer. It also demonstrates that FastViT+PT generalized better to out-of-distribution data than ResNet18+PT and EfficientNet+PT.

TABLE I
COMPARISON OF MODELS AND TRAINING APPROACHES.

Model	Success Rate (%)	
	Train	Test
FastViT+PT	88	**79**
ResNet34+PT	56	26
EfficientNetB4+PT	74	25
FastViT+NOPT	74	29
ResNet34+NOPT	91	29
EfficientNetB4+NOPT	**97**	9

*all values are rounded into decimal place.

The FastViT+PT experienced fewer collisions on the testing obstacle set compared to models without pre-training. It highlights the importance of CARLA pre-training for generalizing to new obstacle types. Training only on real-world obstacle avoidance led to failures in most scenarios. This can be attributed to: the model's limited ability to generalize its behavior across diverse driving environments and unseen obstacle types.

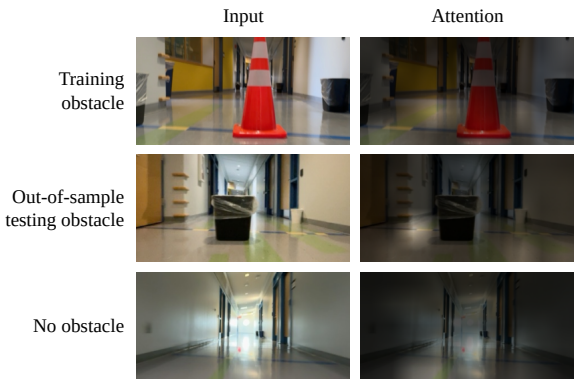

Fig. 6. Visualization of the last self-attention layer for FastVit+PT. The color indicates the average attention weight. The brighter the color is, the more attention it has.

The performance of the CARLA pre-trained model indicates the effectiveness of pre-training in minimizing the dependency on real driving data and the model's ability to transfer. While collecting real-world obstacle data is expensive and tedious, pre-training a model on enormous and automatically collected simulated data expedites the process and reduces the cost of obtaining the best-performing model on the self-driving task.

E. Visualization

In this section, we visualize the self-attention layer of the FastViT+PT model. We compute the regions where the last self-attention layer assigns the most weight for an image from the training dataset, an image with an out-of-sample obstacle type, and an image without obstacles. As shown in Figure 6, for the training image, our attention mechanism prioritizes the empty space, indicating that it focuses on the critical regions of the road. In the testing image, the model generalizes to the unseen obstacle types by similarly prioritizing the empty space on the road.

F. Ablation Study

In this section, we explored four variants of FastViT by scaling the number of output features in four stages from (48, 96, 192, 384) to (64, 128, 256, 512) and adjusting the number of layers in four stages from (2, 2, 4, 2) to (4, 4, 12, 4). We name the model as FastViT/#(feature size)/#(total layers) to reflect the configuration. The models were evaluated in a closed-loop driving simulator, where an episode was considered successful if the vehicle avoided the obstacle and reacheed the end-waypoint without collision. We ran 500 tests for each model and calculated the successful rate of obstacles avoidance.

Table II shows the results of the evaluation in CARLA. The simulation result suggests that increasing the number of layers in the FastViT model increases the success rate by approximately 3%, while increasing the number of features does not significantly improve performance. FastViT/384/24 achieved the best performance on the testing obstacle sets and was selected for fine-tuning and evaluation in real-world experiments.

TABLE II
FASTVIT CARLA EVALUATION RESULTS

Model	Success Rate (%)	
	Train	Test
FastViT/384/10	80.8	76.2
FastViT/384/24	86.1	**79.8**
FastViT/512/10	**90.7**	74.8
FastViT/512/24	86.5	78.3
ResNet34	39.3	13.5
EfficientNetB4	67.0	51.6

⋆all values are rounded into first decimal place.

V. CONCLUSIONS

The simulation environment is crucial for end-to-end autonomous driving due to its low cost and the ease of collecting high-quality data, particularly for rare-case scenarios. While simulators have demonstrated impressive results in autonomous driving models, transferring these models directly to real-world scenarios is hindered by the "reality gap" between the simulated and real environments.

In this work, we evaluate the domain adaptation of our Transformer-based model for end-to-end autonomous driving from simulation to real-world scenarios. We pre-train a FastViT model on a large-scale dataset collected in the CARLA driving simulator, using domain randomization, and then fine-tune it with a small real-world obstacle avoidance dataset. The fine-tuning process helps align the model from the simulation environment to real-world scenarios, adjusting factors such as steering and throttle scales while preserving the image and scene understanding capabilities learned during simulation. Our real-world experiments show that the model effectively transfers domain knowledge from simulation to real-world driving, successfully generalizing to avoid previously unseen obstacles. We compare our approach with a version of FastViT that was not pre-trained on the CARLA dataset, demonstrating that pre-training is key to the model's generalization capabilities. Additionally, we benchmark FastViT against ResNet and EfficientNet, confirming that FastViT is essential for successful domain knowledge transfer.

Our results suggest that pre-training on large-scale datasets with domain randomization facilitates the effective transfer of knowledge from simulation to reality. Furthermore, the Transformer architecture exhibits superior transferability across domains compared to CNNs, making our approach promising for tackling complex urban driving scenarios in fully autonomous driving systems.

REFERENCES

[1] H. Shao, L. Wang, R. Chen, S. L. Waslander, H. Li, and Y. Liu, "ReasonNet: End-to-End Driving with Temporal and Global Reasoning," May 2023, arXiv:2305.10507 [cs].

[2] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, "TransFuser: Imitation with Transformer-Based Sensor Fusion for Autonomous Driving," May 2022, arXiv:2205.15997 [cs].

[3] H. Shao, L. Wang, R. Chen, H. Li, and Y. Liu, "Safety-Enhanced Autonomous Driving Using Interpretable Sensor Fusion Transformer," Dec. 2022, arXiv:2207.14024 [cs].

[4] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, "End-to-end Autonomous Driving: Challenges and Frontiers," June 2023, arXiv:2306.16927 [cs].

[5] M. Müller, A. Dosovitskiy, B. Ghanem, and V. Koltun, "Driving Policy Transfer via Modularity and Abstraction," Dec. 2018, arXiv:1804.09364 [cs]. [Online]. Available: http://arxiv.org/abs/1804.09364

[6] C. Hu, S. Hudson, M. Ethier, M. Al-Sharman, D. Rayside, and W. Melek, "Sim-to-Real Domain Adaptation for Lane Detection and Classification in Autonomous Driving," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, June 2022, pp. 457–463.

[7] X. Pan, Y. You, Z. Wang, and C. Lu, "Virtual to Real Reinforcement Learning for Autonomous Driving," Sept. 2017, arXiv:1704.03952 [cs].

[8] A. Bewley, J. Rigley, Y. Liu, J. Hawke, R. Shen, V.-D. Lam, and A. Kendall, "Learning to Drive from Simulation without Real World Labels," Dec. 2018, arXiv:1812.03823 [cs].

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," *Advances in Neural Information Processing Systems*, vol. 2017-December, pp. 5999–6009, June 2017, publisher: Neural information processing systems foundation _eprint: 1706.03762.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," June 2021, arXiv:2010.11929 [cs].

[11] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept. 2017, pp. 23–30, iSSN: 2153-0866.

[12] S. Pouyanfar, M. Saleem, N. George, and S.-C. Chen, "ROADS: Randomization for Obstacle Avoidance and Driving in Simulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[13] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[14] P. K. A. Vasu, J. Gabriel, J. Zhu, O. Tuzel, and A. Ranjan, "FastViT: A Fast Hybrid Vision Transformer using Structural Reparameterization," Aug. 2023, arXiv:2303.14189 [cs].

[15] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, and others, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," May 2019, arXiv:1810.04805 [cs].

[17] T. Zhou, P. Niu, X. Wang, L. Sun, and R. Jin, "One Fits All:Power General Time Series Analysis by Pretrained LM," Oct. 2023, arXiv:2302.11939 [cs].

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 2015, arXiv:1512.03385 [cs].

[19] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," Sept. 2020, arXiv:1905.11946 [cs, stat].

[20] I. Sobh, L. Amin, S. Abdelkarim, K. Elmadawy, M. Saeed, O. Abdeltawab, M. Gamal, and A. E. Sallab, "End-To-End Multi-Modal Sensors Fusion System For Urban Automated Driving," Oct. 2018.

[21] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to End Learning for Self-Driving Cars," Apr. 2016, arXiv:1604.07316 [cs].

[22] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end Driving via Conditional Imitation Learning," Mar. 2018, arXiv:1710.02410 [cs].

[23] T. V. Samak, C. V. Samak, and S. Kandhasamy, "Robust Behavioral Cloning for Autonomous Vehicles using End-to-End Imitation Learning," *SAE International Journal of Connected and Automated Vehicles*, vol. 4, no. 3, Oct. 2020, publisher: SAE International _eprint: 2010.04767v4.

[24] S. Ross, G. J. Gordon, and J. A. Bagnell, "A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning," *Journal of Machine Learning Research*, vol. 15, pp. 627–635, Nov. 2010, _eprint: 1011.0686.

[25] S. Akhauri, L. Y. Zheng, and M. C. Lin, "Enhanced Transfer Learning for Autonomous Driving with Systematic Accident Simulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 5986–5993, iSSN: 2153-0866.

[26] A. Amini, I. Gilitschenski, J. Phillips, J. Moseyko, R. Banerjee, S. Karaman, and D. Rus, "Learning Robust Control Policies for End-to-End Autonomous Driving From Data-Driven Simulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1143–1150, Apr. 2020, conference Name: IEEE Robotics and Automation Letters.

[27] G. Li, Z. Ji, and X. Qu, "Stepwise Domain Adaptation (SDA) for Object Detection in Autonomous Vehicles Using an Adaptive CenterNet," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 17 729–17 743, Oct. 2022, conference Name: IEEE Transactions on Intelligent Transportation Systems.

[28] W. Wu, X. Deng, P. Jiang, S. Wan, and Y. Guo, "CrossFuser: Multi-Modal Feature Fusion for End-to-End Autonomous Driving Under Unseen Weather Conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 12, pp. 14 378–14 392, Dec. 2023, conference Name: IEEE Transactions on Intelligent Transportation Systems.

[29] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-Real Transfer of Robotic Control with Dynamics Randomization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 3803–3810, arXiv:1710.06537 [cs].

[30] D. Horváth, G. Erdős, Z. Istenes, T. Horváth, and S. Földi, "Object Detection Using Sim2Real Domain Randomization for Robotic Applications," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 1225–1243, Apr. 2023, conference Name: IEEE Transactions on Robotics.

[31] G. D. Kontes, D. D. Scherer, T. Nisslbeck, J. Fischer, and C. Mutschler, "High-Speed Collision Avoidance using Deep Reinforcement Learning and Domain Randomization for Autonomous Vehicles," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Sept. 2020, pp. 1–8.

[32] I. Sonata, Y. Heryadi, A. Wibowo, and W. Budiharto, "End-to-End Steering Angle Prediction for Autonomous Car Using Vision Transformer," *CommIT (Communication and Information Technology) Journal*, vol. 17, no. 2, pp. 221–234, Sept. 2023, number: 2.

[33] A. Trockman and J. Z. Kolter, "Patches Are All You Need?" Jan. 2022, arXiv:2201.09792 [cs].

[34] J. Guo, K. Han, H. Wu, Y. Tang, X. Chen, Y. Wang, and C. Xu, "CMT: Convolutional Neural Networks Meet Vision Transformers," June 2022, arXiv:2107.06263 [cs].

[35] K. Pei, Y. Cao, J. Yang, and S. Jana, "Deepxplore: Automated whitebox testing of deep learning systems," in *proceedings of the 26th Symposium on Operating Systems Principles*, 2017, pp. 1–18.

[36] 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.

[37] 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.

[38] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Jan. 2017, arXiv:1412.6980 [cs].