Minimal Visual Transfer with Foundation Models for Sim2Real End-to-End Driving

Angela Liu
University of Maryland, College Park
Department of Computer Science
College Park, United States
aliu1237@umd.edu

Laura Zheng
University of Maryland, College Park
Department of Computer Science
College Park, United States
lyzheng@umd.edu

Ming Lin
University of Maryland, College Park
Department of Computer Science
College Park, United States
lin@umd.edu

***Abstract*—Autonomous driving has made unprecedented strides in recent years, thanks in large part to the wide use of computer simulations to generate large amounts of training data without the risk of real-life experiments. However, one of the biggest challenges in training self-driving systems is generalizing across the simulation to real world domain gap. The visual environments in the simulator differ from visual environments in reality in many ways, such as lighting, textures, and coloration. A self driving model is susceptible to overfitting on these unimportant traits and fail to replicate the same level of performance in deployment. While current simulators attempt to close this gap with photo-realistic rendering, another approach is via Transfer Learning. In this work, our goal is to learn the *minimum visual transfer* necessary to convert image data from simulation to perceptually-realistic images in the eyes of a robust, pre-trained foundation model. Our proposed method, Min2Real, enables models to leverage virtual data to improve performance in the real world, even if the policy only encounters specific scenarios in simulation.**

***Index Terms*—end-to-end driving, sim2real, generalization, autonomous driving**

I. Introduction

Simulators, such as CARLA, play a vital role in autonomous driving research. Autonomous driving is a domain that is difficult to train in the target domain; any single failure can lead to expensive costs as well as endanger human safety. Thus, it is not only important to be able to train a model in simulation, but also for the trained model to generalize to the real world, even if it only encountered certain situations in simulation. One clear example of this is the capability to generalize avoidance maneuvers to dangerous situations. The hope is that if the policy can navigate difficult successfully in simulation, that it can also do s0o in the real world.

In modular autonomous driving approaches, simulation can be fully confined to its respective modules. Perception simulation can be solely response for simulating changes in perception, and behavior simulation can be solely responsible for simulating changes in behavior. However, end-to-end driving frameworks are gaining relevance due to recent advancements in performance and efficiency. Since end-to-end approaches are more difficult to interpret in failure cases than modular approaches, bridging the sim-to-real gap also comes with greater challenges.

Our work leverages strong, pre-trained visual foundation models to act as “judges" of perceptual realism. This is in contrast to existing work in 3-D reconstruction for visual simulation, which assumes humans as judges of realism. While a highly photo-realistic render may also appear in-distribution to a foundation model, we hypothesize that the relationship between inputs and relevant features is not injective. Perhaps, we can learn a simpler function that maps a simulated image to the real-world image distribution in the eyes of a neural network.

Our contributions can be summarized as follows:

1. A novel end-to-end training approach which leverages minimal visual transfer from simulation to real world;

2. Ablation studies showing the effect of leveraging both real and simulated training data;

3. Results showing that Min2Real outperforms SOTA benchmarks on the Longest6 benchmark.

II. Related Works

A. Transfer Learning

Transfer learning enables knowledge from one domain to be transferred to another. One strategy to leverage both simulated and real world data for driving is through domain-agnostic learning, which extracts latent features common to both source and target domains [1]. Using a discriminator and generator , the authors were able to train their feature extractor select for features invariant to the visual domain, preventing overfitting to any single domain. Another recent work, SimScale [2], utilizes transfer learning to achieve first place on the NAVHARD public leaderboard [3]. However, this method achieves visual transfer through diffusion-based photo-realistic rendering; conversely, we believe that this computational expensive process can be avoided by leveraging gradients from robust, pre-trained foundation models.

B. End-to-End Driving

End-to-end driving is a popular task that is accessible to both large industry labs and independent researchers, unlike modular driving frameworks. Approaches for end-to-end driving now generally fall under two different paradigms: regression-based and diffusion-based.

Diffusion-based approaches leverage diffusion models to handle the inherent uncertainty and multi-mode nature of driving behaviors, and has recently emerged as a popular

method to do end-to-end driving. These models iteratively denoise within the trajectory space to obtain plausible plans for the driving policy. Despite being computationally expensive to run at a high frame rate, approaches such as DiffusionDrive [4] have enabled diffusion learning to run at 45 fps or more. While this is a promising paradigm for end-to-end driving, we focus on an efficient approach for learning across domains; diffusion policies have yet to generalize across the sim-to-real gap.

Regression-based approaches typically predict driving actions or goal waypoints directly. State-of-the-art regression-based approaches include Transfuser [3], Interfuser [5], Drive Adapter [6], and Carla Garage [7].

III. Method

There are 2 different type environments that the agent will be running in: the CARLA simulator [8] and the NAVSIM pseudo simulator [9]. In CARLA, each route contains a list of target points spaced around 30 meters apart and the agent will encounter manually designed traffic scenarios that test the agent's ability to avoid a crash. In NAVSIM, the ego agent will generate a 4-second trajectory based on sensor inputs and a driving command, scored via Extended Predictive Driver Model Score (EPDMS). In both environments, we consider the task of navigation from point A to B without incurring infractions. In this work, we present a novel architecture for end-to-end driving, built upon existing work with Transfuser [3] and Carla Garage [7]. A diagram can be seen in Figure 1. There are 4 main components: (1) The image augmentation applies a randomized gaussian noise ranging from 0-100, (2) the virtual adapter and real extractor pair that apply dimensional bottleneck on the rgb data, (3) the discriminator that tries to tell apart the real and virtual images, and (4) the trajectory predictor that determined the left, right or straight steering pattern.

A. Image Augmentation

Our data comes from 2 input streams: real and virtual. Virtual data is collected from the CARLA simulator and contain both rgb and lidar data. Real data is curated from the OpenScene [10] dataset and only contain rgb data. Currently, we have zeroed out the lidar vectors for the real data stream. Both are passed through the augmentation pipeline, which consists of a randomized level of gaussian noise ranging from 0 to 100 see Figure 3. Random noise will help prevent the model from overfitting to any particular domain. We compare the original baseline performance with the augmentation performance with no additional changes on the longest6 scenarios. As shown in Table III, we found an over 8 point increase to driving score and an over 8 percentage increase in route completion.

Earlier on, we explored alternative forms of image augmentation, including image segmentation and gray-scaling. After running a number of tests, we found a noticeable performance drop. We hypothesize that the alterations were too extreme and our model was unable to learn from them. Given how much Gaussian noise improved the model, there is potentially more to explore with augmentation.

B. V2R Adapter and Real Extractor

The V2R adapter is a bottleneck convolutional adapter that transforms input images for domain adaptation. Only virtual images are passed in. As shown in Figure 2, we pass in a 384 by 1024 image and reduce its dimensionality to 128 before passing it back out as an augmented image suitable for the extractor. DINOv2 is a powerful pretrained vision foundation model [11] that uses ViT as a feature extractor and is pretrained on real images. Unlike V2R, we pass in both real data and the adapted virtual data. The extractor is frozen during training. It takes the adapted image as the output and produces features for further processing.

C. Discriminator

We train a discriminator to separate real and virtual images from DINOv2 features. We run binary cross entropy on the prediction to get the discriminator loss, which gets passed back to the V2R Adapter as an inverted version. V2R learns what can fool the discriminator and will select features shared in both real and virtual images. This adversarial method is used to force the feature extractor to obtain as much domain-invariant information as possible.

D. Trajectory Predictor

We train a trajectory predictor to predict the steering direction. There are 3 classes: straight, left, and right. We pass in the DinoV2 features to a simple linear classifier and get cross entropy loss on the output. This serves as the an auxiliary task that will be helpful for analyzing later tests on real turning data.

IV. Results

We show results comparing our trained Min2Real model of various forms with the original baseline. In terms of hardware, all results were produced on 32 CPU cores, 120 GB of system memory, and 8 NVIDIA RTX A5000 graphics cards.

Overall, we find that adding our custom components improved model performance across all mediums. Evaluation metrics can be found in Table IV. Inference is run twice: one on the first 10 scenarios in the longest6 benchmark in CARLA simulator and one on 100 scenarios in NAVSIM with navhard_two_stage split.

In the CARLA simulator, we see general improvements in simulator only training. We choose to run the Longest6 benchmark as its length and variety of scenarios make it challenging for models to cheat through, as illustrated in Figure 4. As seen in Table I, the greatest performance boost comes from the augment model. However, the Min2Real run also shows a small improvement including the least infraction rate among all models. To rest of the models include both simulator and real data. Both the mixdata baseline and mixdata Min2Real have drops in score. As mixdata is simply baseline model with real data mixed in, the subpar results could be

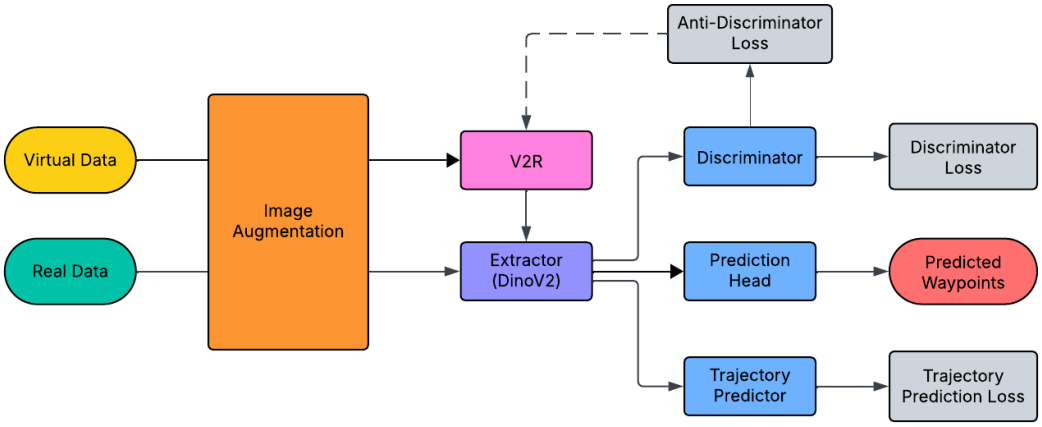


Fig. 1: **Minimal Visual Transfer with Simple Adapters (Min2Real) Architecture.** We consider a simulator data input stream and real data input stream. A random level of gaussian noise is applied to images from both streams. Only simulator data is passed through a V2R adapter, forcing it through a bottleneck and creating an adapter image. Both domains are passed through the pretrained and frozen DINOv2 feature extractor. Features are then passed to (1) a discriminator that distinguishes real and virtual domains which also provides adversarial feedback to the V2R Adapter, (2) a prediction head outputting waypoints, and (3) a trajectory predictor that predicts the trajectory direction. The V2R adapter is trained to produce domain-invariant features that fool the discriminator, while the downstream prediction heads are trained on both domains to learn driving policies.

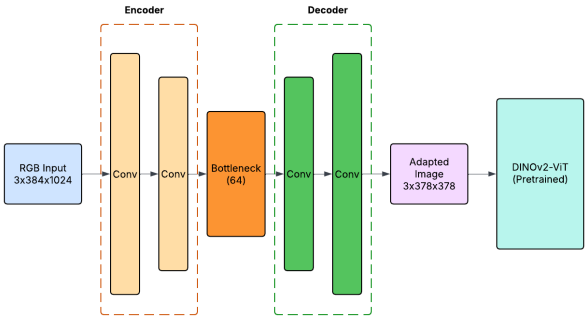


Fig. 2: V2R Adapter to Extractor Pipeline. We use an encoder decoder structure where raw image is compressed to bottleneck size before expanding into a fitting dimension for DINOv2.

due to how the original model is not built to handle real data. This is supported by the fact that mixed Min2Real achieves nearly twice the driving score. However, the full model, adding augmentation to mixed Min2Real, does not follow this trend and performs the worst. Overall, there needs to be more finetuning conducted to ensure that real and simulator data mesh correctly. Looking closer at the scenario footage, we see that fails in CARLA frequently result from agents running into fail states such as crashes or moving into incoming traffic following a turn. The turn process is successful up until the moment it reaches an anchor point. Then the waypoints abruptly lose their sense of direction and frequently, the model is unable to recover. This behavior is shared across models in varying degrees of severity. An ongoing effort looks to increase the weight of losses that occur during a turn.

In the NAVSIM pseudo-simulator, each run is much shorter, most occuring in a couple frames (see Figure 5). All the models pass over 90 percent of scenarios as shown in Table II. Still, we see from average score and infraction rate that regular Min2Real trained on exclusively simulator data performs the best, getting nearly double the score as baseline. Given that the DINOv2 extractor is pre-trained on real images, it is no surprise that Min2Real outperforms the base transfuser model. While an average score of 0.49 is an approximation of the actual pdm score, it shows promise that our method will be able to outperform the top ranked leaderboard methods.

Overall, these results suggest strong improvement gains with components on simulator only training. However, when incorporating both real and simulator data, the results are murkier. We see performance drops for all the mixdata models but mixdata Min2Real does achieve higher scores than regular mixdata. There is also the conflicting results for augmentation improving scores for simulator only training while hurting scores for combined training. We suspect that rgb processing for real images needs to be adjusted. We will also need to rework how lidar data is handled for real data. By zeroing out the lidar values, it is possible the discriminator may have found a shortcut without actually learning how to classify images. More testing and retraining will be needed.

V. Ablation Studies

We conducted several ablation studies to test and individually assess the effectiveness of each component. We ran different versions of the model that only contains the specified component for the first 3 scenarios on the longest6 benchmark,

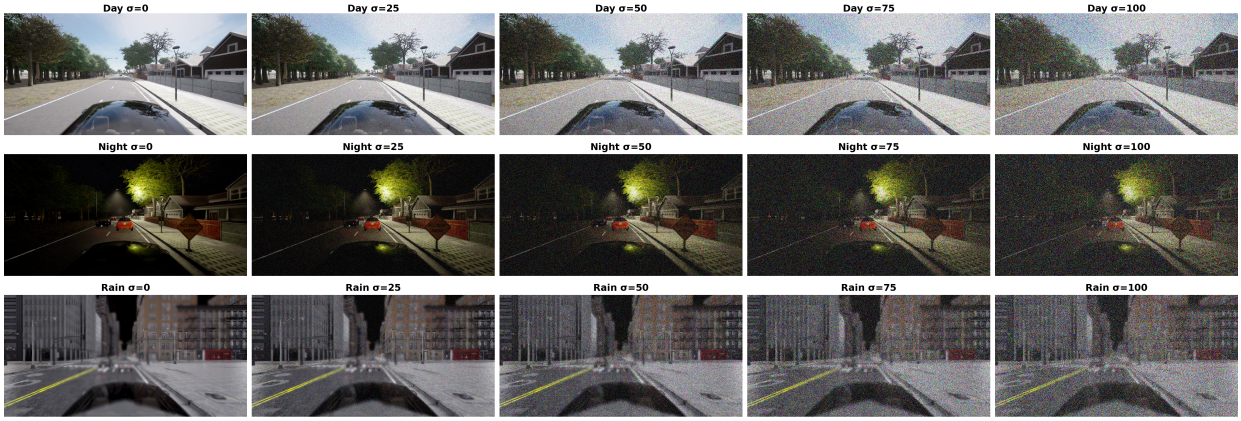
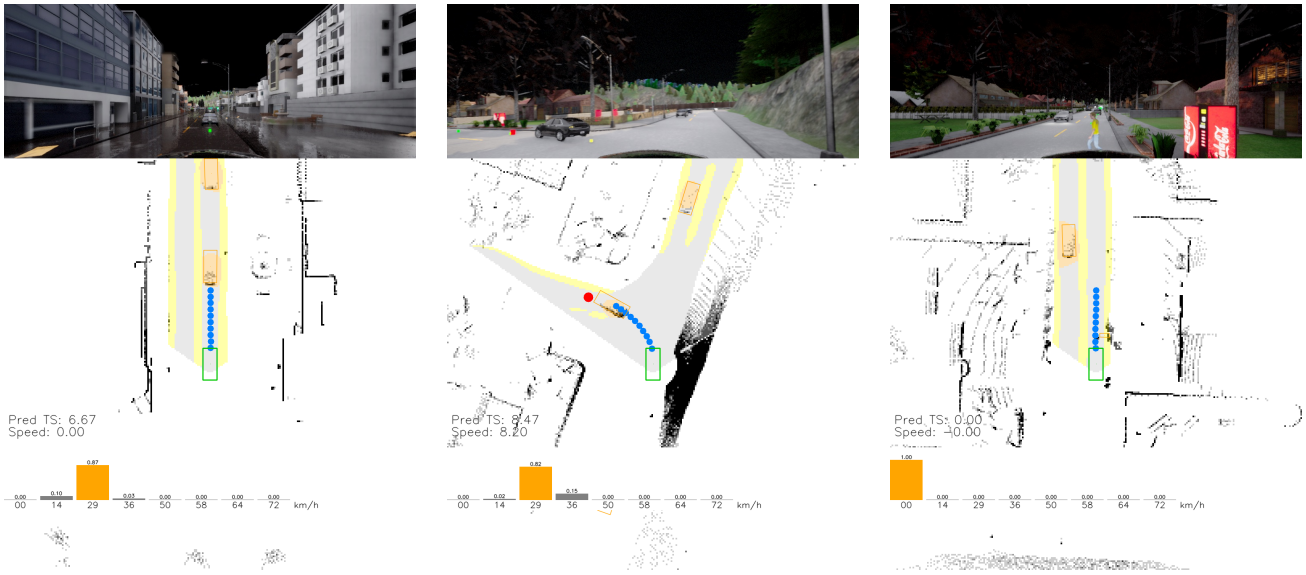


Fig. 3: Different levels of gaussian noise on clear, night, and rainy weather.



(a) Straight Drive (b) Intersection Turn (c) Stop for Pedestrian

Fig. 4: Driving scenarios of running Min2Real in CARLA simulator.

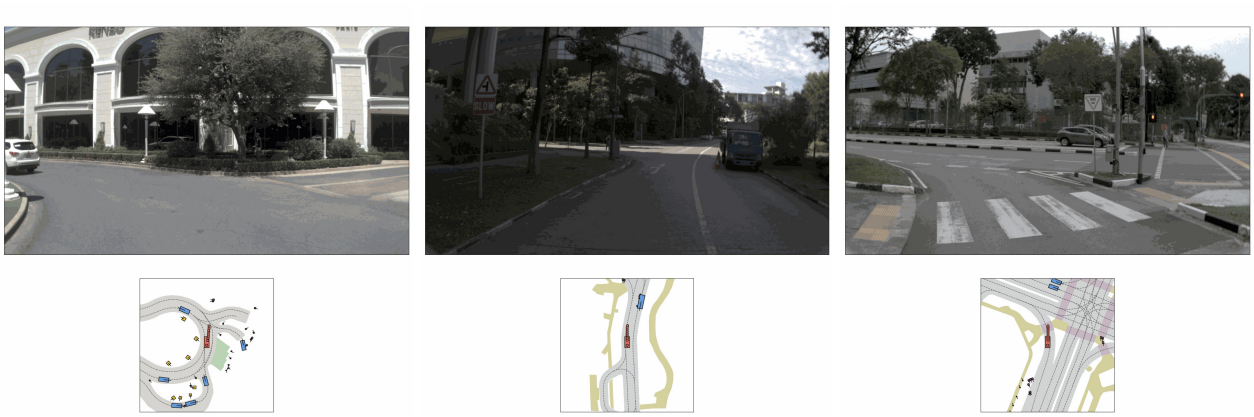


Fig. 5: Driving scenarios of running in model Min2Real in NAVSIM.

TABLE I: Longest6 Scores Across Scenarios 1-10.

Model	DS Mean ↑	DS Std	RC Mean ↑	RC Std	Infracts.
Trained on $\{S\}$					
baseline	20.73	14.37	61.51	36.56	0.34
augment	**28.82**	25.2	**69.82**	36.81	0.39
Min2Real	22.41	30.65	52.25	37.63	**0.33**
Trained on $\{S, R\}$					
mixdata	3.88	4.37	22.55	24.36	0.35
mix Min2Real	7.06	11.89	43.51	37.12	0.34
full	3.68	4.45	15.39	19.77	0.29

TABLE II: NAVSIM Scores on navhard_two_stage

Model	Avg. Score↑	Infraction Rate↓	Valid Scenarios↑
Trained on $\{S\}$			
Baseline	0.26	0.6	0.92
Min2Drive	**0.49**	**0.3**	0.91
Trained on $\{S, R\}$			
Baseline	0.34	0.47	0.93
Min2Drive	0.38	0.47	0.9
Mine2Drive+Aug	0.22	0.64	**0.93**

TABLE III: Effect of Augmentation on Driving Score and Route Completion Percentage

Model	Avg. driving score	Avg. route completion
Baseline	20.73	61.51
Baseline+Aug	**28.82**	**69.82**

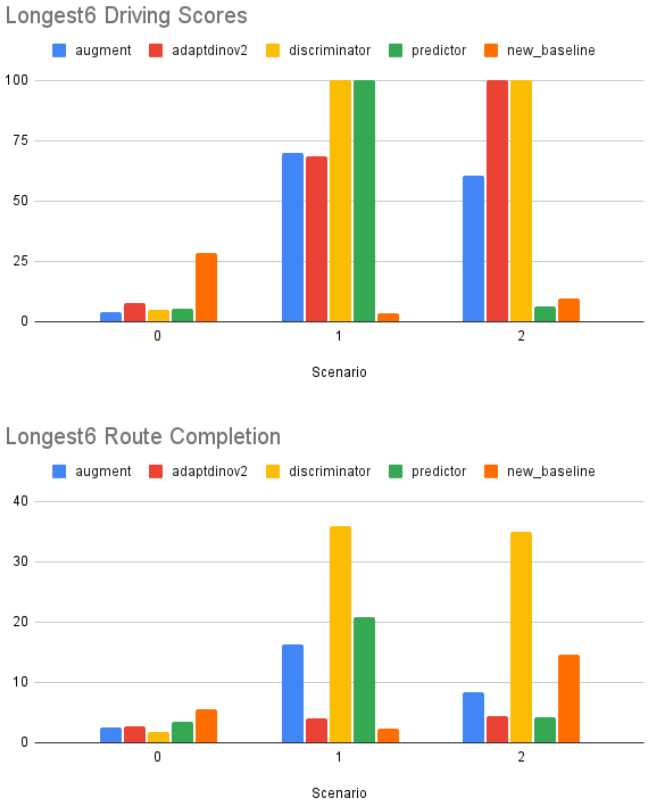


Fig. 6: Driving Scores and Route Completions for each ablation study (with baseline) when running on the first 3 scenarios of longest6 benchmark.

see Figure 6. We see that including each component gives an overall performance boost over the baseline. However, more results need to be collected to ensure that all the components work together effectively.

VI. Conclusion

Our work is ongoing as we continue to finetune our model and improve performance where possible. Nevertheless, there are a few limitations of our current study. Firstly, we notice that the baseline model and all others that were built upon it have unstable navigation after turning at an intersection. We hypothesize that turns are underrepresented in the training data, making it easy for the model to lose its sense of direction. Future work can explore replacing waypoint prediction with action prediction to soften the driving trajectory or incorporating reinforcement learning to give better error handling.

Secondly, there can be more measurement specificity in how our models handle the real data. The navsim average score is more of a report on general overall performance. It would be illuminating to see how models perform in certain situations or in specific location types. There is the question as to whether certain tasks are easier to transfer over other.

A promising direction of future study is analyzing how our trained adapter adapts an image to determine what makes a virtual image photorealistic. Because of the adversarial learning between the discriminator and V2R adapter, V2R will have learned how to adapt a given image to best fool the discriminator. This could drive further research in the nature of computer vision and enable better methods for tying together real and simulator data.

References

[1] Y. Shen, L. Zheng, T. Zhou, and M. C. Lin, "Task-driven domain-agnostic learning with information bottleneck for autonomous steering," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6858–6865, IEEE, 2024.

[2] H. Tian, T. Li, H. Liu, J. Yang, Y. Qiu, G. Li, J. Wang, Y. Gao, Z. Zhang, L. Wang, *et al.*, "Simscale: Learning to drive via real-world simulation at scale," *arXiv preprint arXiv:2511.23369*, 2025.

[3] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, "Transfuser: Imitation with transformer-based sensor fusion for autonomous driving," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 11, pp. 12878–12895, 2022.

[4] B. Liao, S. Chen, H. Yin, B. Jiang, C. Wang, S. Yan, X. Zhang, X. Li, Y. Zhang, Q. Zhang, *et al.*, "Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12037–12047, 2025.

[5] 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*, pp. 726–737, PMLR, 2023.

[6] X. Jia, Y. Gao, L. Chen, J. Yan, P. L. Liu, and H. Li, "Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7953–7963, 2023.

[7] B. Jaeger, K. Chitta, and A. Geiger, "Hidden biases of end-to-end driving models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8240–8249, 2023.

[8] 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*, pp. 1–16, 2017.

[9] D. Dauner, M. Hallgarten, T. Li, X. Weng, Z. Huang, Z. Yang, H. Li, I. Gilitschenski, B. Ivanovic, M. Pavone, *et al.*, "Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking," *Advances in Neural Information Processing Systems*, vol. 37, pp. 28706–28719, 2024.

[10] O. Contributors, "Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving." https://github.com/OpenDriveLab/OpenScene, 2023.

[11] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.