Supplementary Material for *OmniSCS: Omni Safety-Critical Scenario Synthesis for Autonomous Driving via a Fully Editable Driving World*

Anonymous submission

Abstract

In this supplementary document, we review related work in the field of autonomous driving data synthesis in Sec. **1**. Sec. **2** and Sec. **3** provide additional details on the framework, implementation, and experiments presented in the main paper. In Sec. **4**, we conduct further validation experiments to demonstrate the superiority of our method in challenging scenarios, such as crowded environments, high-speed driving, extreme weather conditions, and low-light settings. We also evaluate the speed of our method. Sec. **5** presents comprehensive visualized experimental results. Sec. **6** evaluates the individual modules of our method through additional ablation studies. These studies specifically verify the robustness and effectiveness of the depth-refinement background reconstruction module, while also assessing the advantages of our proposed joint adversarial loss in behavior-level safety-critical scenario synthesis.

1 Related Work

1.1 High-fidelity Data Synthesis

High-fidelity data for autonomous driving can be produced via three approaches: physical simulation using dedicated simulators, reconstruction from real-world data and generation using learned models. Based on the underlying techniques, mainstream methods can be categorized into Simulator-based Scene Synthesis, NeRF/3DGS-based Scene Reconstruction and Diffusion-based Scene Generation.

Simulator-based Scene Synthesis. Traditional simulators, such as CARLA (Dosovitskiy et al. 2017), AirSim (Shah et al. 2017) and LGSVL (Rong et al. 2020), provide open-source platforms for generating synthetic driving data. These simulators offer significant advantages, including extensive API integrations that enable easy customization and interfacing with autonomous driving algorithms. However, expanding to new simulation environments often requires substantial manual modeling efforts, which can be time-consuming and expensive. Moreover, the resulting models frequently lack photorealism, potentially limiting their effectiveness in training perception module and end-to-end (E2E) module. Additionally, for synthesizing safety-critical scenarios (SCS), these simulators typically necessitate integration with specialized adversarial or optimization methods to create challenging conditions effectively.

Nerf/3DGS-based Scene Reconstruction. Recent advances in neural rendering, particularly Neural Radiance Fields (NeRF) (Mildenhall et al. 2021) and 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023), have enabled photorealistic scene reconstruction for autonomous driving applications. These techniques excel at synthesizing novel viewpoints and reconstructing static environments with high fidelity. To support dynamic driving scenarios, studies such as StreetGaussian (Yan et al. 2024) and OmniRe (Chen et al. 2025), as well as driving simulators like HUGSIM (Zhou et al. 2024) and MARS (Wu et al. 2023), employ Neural Scene Graphs (NSGs) to decompose scenes into static backgrounds and dynamic agents. Specifically, dynamic objects are modeled using 3D bounding boxes, which act as spatial anchors for distinct neural representations, thereby facilitating efficient modeling and motion rendering. However, when agent behaviors significantly deviate from the captured data, these methods can produce blurring and ghosting artifacts, limiting their effectiveness in synthesizing SCS involving agent behavior changes.

Diffusion-based Scene Generation. Rather than reconstructing real-world scenes, another type of approach is to generate synthetic driving scenarios directly. Recent advances in diffusion models (Zhao et al. 2025; Yang et al. 2025; Mei et al. 2025; Wang et al. 2024; Gao et al. 2024; Amini et al. 2022) offer the ability to generate controllable and diverse synthetic video data. These models accept multimodal inputs, such as camera images, Bird's Eye View (BEV) layouts, and text prompts, to generate highly customizable driving scenarios. Existing works (Amini et al. 2022; Wang et al. 2024; Ma et al. 2024) demonstrate that these diffusion-generated videos can be used at the training stage of the perception and the E2E driving algorithms, thus enhancing their robustness. A key advantage of diffusion models is their progressive denoising process, which allows fine-grained manipulation of agent behaviors through BEV layout editing. Closed-loop autonomous driving simulation platforms like DriveArena (Yang et al. 2025) and DreamForge (Mei et al. 2025) exploit this capability to produce behavior-alterable, photorealistic video data for simulation and testing. These simulators enable comprehensive evaluations of autonomous driving systems in dynamic virtual environments. However, current diffusion models struggle to

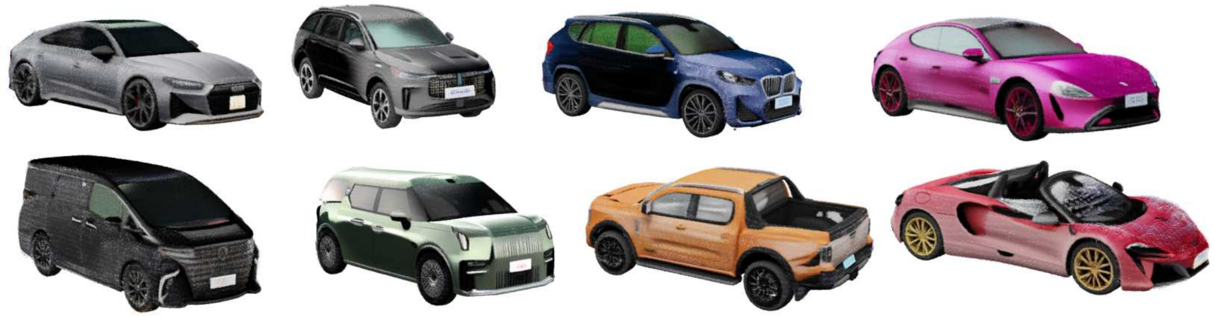


Figure 1: Examples of 3D vehicle models in the Model Pool. The surface of each model is densely covered with Gaussian blobs, offering rich geometric and appearance information.

control the fine-grained appearance of individual objects in dynamic scenes, limiting their effectiveness for appearance-level SCS synthesis.

1.2 Safety-critical Scenario Generation

Traditional safety-critical scenario generation approaches (Wang et al. 2021; Tan et al. 2023; Rempe et al. 2022) primarily focus on synthesizing adversarial behaviors of road agents to evaluate the planning performance of autonomous driving systems. These methods perturb agent trajectories while maintaining physical plausibility, thereby creating collision-prone interactions that challenge planning algorithms. For instance, (Tan et al. 2023; Rempe et al. 2022) demonstrate that optimized adversarial trajectories can expose vulnerabilities in motion planning, leading to dangerous behaviors under critical conditions. However, these approaches typically emphasize behavior-level simulation and lack the generation of corresponding photorealistic sensor data aligned with agent behaviors, rendering them incompatible with perception or E2E autonomous driving systems that rely on sensor inputs such as camera images and LiDAR point clouds.

2 Details of Framework and Implementation

2.1 3D Model Pool Construction.

To construct our comprehensive 3D model pool, we adopted the following multi-stage workflow:

1. **Data Collection:** We employed a web scraping tool to gather best-selling vehicle data from a public website (Autohome 2025). The data covers a diverse range of vehicle categories—including 367 sedans, 151 SUVs, 151 MPVs, 31 pickups, 31 racing cars, and 31 trucks—and contains images, sizes, and model designations.
2. **Data Cleaning:** To address incomplete and duplicate entries, we applied a Python script that verifies both the completeness and uniqueness of each record, discarding any invalid or redundant entries.
3. **3D Model Synthesis:** For each validated vehicle, we synthesized a 3D Gaussian model (denoted as G) using Trellis (Xiang et al. 2025), taking the collected vehicle images as input. Each resulting 3D model was then rendered into 30 keyframe images at 12° angular intervals, providing a comprehensive set of reference views.
4. **Coordinate System Alignment:** By employing the coordinate transformation method described in the "Scalable 3D Model Construction" section of the main paper, we aligned G to a standard vehicle coordinate system, such that the positive x-axis coincides with the vehicle's heading direction.
5. **Manual Quality Verification:** To ensure model fidelity, manual inspection was performed using visualization tools. Only models that passed this quality check were included in the final pool.

Through this workflow, as shown in Fig. 1, we constructed a high-quality pool of 440 3D vehicle models, each accompanied by corresponding keyframe images, forming the foundation for degraded agent representation in our system.

2.2 Visual Degradation-aware Object Matching Module.

The module contains a multi-scale feature extraction network (MSNet) and an object matching method. While the main paper describes our motivation, along with the network's training and inference procedures, this supplementary material provides detailed implementation information for MSNet.

MSNet consists of four stacked Omni-Scale Residual Blocks (Zhou et al. 2019) followed by a fully connected layer. Each residual block extracts and aggregates features across four scales ($3\times$, $5\times$, $7\times$, and $9\times$) and applies $2\times$ downsampling at the output. For an input image of dimensions $H \times W \times 3$, the network produces a feature map of $H/16 \times W/16 \times 512$ after the four blocks, which is then processed by the fully connected layer to yield a final 512-dimensional feature vector. The entire network has approximately 2.2 million parameters, significantly fewer than the 24 million in ResNet-50. To facilitate detailed inspection of the network architecture, we provide the ONNX-format network definition in the supplementary materials, which can be visualized using standard tools (Roeder 2025). To protect our work prior to publication, the weights in the ONNX

Layer	Kernel	Output Channels	Activation
Conv2d	3	32	LReLU + BN
Conv2d	3	64	LReLU + BN
Spatial Attention	7	64	Sigmoid
Conv2d	3	32	LReLU + BN
Conv2d	3	1	-

Table 1: Architecture of the proposed depth refinement network.

file have been anonymized, and the file is intended solely for viewing the network structure.

For training, we utilize the 3D Model Pool containing 440 objects, with 80% allocated to the training set and 20% to the test set. The network is trained using the AMSGrad optimizer (Reddi, Kale, and Kumar 2019) with an initial learning rate set to 0.0015 and a cosine learning rate scheduler. The weight decay coefficient is set to 0.0005, the batch size is 64, and the maximum number of training epochs is 250.

2.3 Depth Refinement Network.

The proposed depth refinement network consists of a series of convolutional layers and a spatial attention module, as outlined in Tab. 1. The input is a single-channel normalized relative depth map. After two convolutional blocks with batch normalization and LeakyReLU activations, a spatial attention module (Woo et al. 2018) is inserted to enhance spatially informative regions. The spatial attention module employs both average-pooling and max-pooling operations along the channel axis, followed by a convolution and sigmoid activation, allowing the network to adaptively highlight salient spatial regions and improve feature representation for depth refinement. The feature maps are then further refined by an additional convolutional block, followed by the final convolution layer to produce the output absolute depth map. During training, we select all regions in the scene that are occluded by large static objects as training data. The network is optimized using the Adam optimizer (Kingma and Ba 2017) with a learning rate of 0.01, and is trained for 100 epochs.

3 Details of Experiments

3.1 Datasets.

In our experiments, we comprehensively evaluated the effectiveness and generalization capabilities of our method using a variety of datasets and diverse scene selections. The details are as follows:

Photorealism Evaluation: We selected challenging driving scenarios from the NuScenes (Caesar et al. 2020), Waymo (Sun et al. 2020), and KITTI (Geiger, Lenz, and Urtasun 2012) datasets, as illustrated in Tab. 2. These scenarios encompass heavy traffic, high-speed conditions, low-light environments, and extreme weather, thereby ensuring a thorough assessment of our method's photorealistic scene editing performance.

Dataset	**Scene IDs**
NuScenes	scene-0003, scene-0105, scene-0127, scene-0399, scene-0405, scene-0443, scene-0514, scene-1098
Waymo	seg109239..., seg125050..., seg113555..., seg128560...
KITTI	2011_09_26_drive_0014_sync, 2011_09_26_drive_0017_sync, 2011_09_26_drive_0018_sync

Table 2: Selected scene IDs for photorealism evaluation.

Dataset	**Scene IDs**
NuScenes	scene-0092, scene-0109, scene-0521, scene-0522, scene-0523, scene-0637 scene-0627, scene-0909

Table 3: Selected scene IDs for SCS evaluation.

Algorithm Data Fidelity and Performance Optimization: In the experiment for "Data Fidelity for Algorithms" and "Algorithm Performance Optimization", to evaluate algorithm performance on our synthesized SCS data, we followed the setup of the CODA dataset (Li et al. 2022), which identifies a range of corner-case scenarios within NuScenes. To prevent data leakage during evaluation, we exclusively selected those corner cases from CODA that are included in the official NuScenes validation set as our test set (specific scene IDs are provided in Tab. 3. This strategy ensures a fair and rigorous evaluation of our method in safety-critical scenarios.

3.2 Baseline.

We compare with SOTA scenario reconstruction methods and generative models respectively. OmniRe (Chen et al. 2025) and StreetGS (Yan et al. 2024) use Gaussian scene graphs to separate objects and the background in the reconstructed scene, which can be used for scene editing. DeformableGS (Yang et al. 2024) represents scenes in a canonical space using Gaussians and models dynamics by employing a deformation network that predicts property offsets, enabling the Gaussians to deform and capture scene motion. PVG (Chen et al. 2023) introduces Periodic Vibration Gaussians, which are optimized in a self-supervised manner and achieve static-dynamic decomposition by categorizing Gaussians based on their life spans to represent dynamic scenes. WorldDreamer (Yang et al. 2025) and DreamForge (Mei et al. 2025) are data synthesis methods based on diffusion models. They can edit the scene by controlling the layout of objects in the scene. Moreover, (Yang et al. 2025; Mei et al. 2025) integrate their methods into the simulation

Method	Translation (FID)↓		Rotation (FID)↓		
	2.0m	3.5m	90°	180°	270°
StreetGS	67.70	74.04	81.50	68.65	82.92
Omnire	68.32	74.02	81.88	68.42	81.97
PVG	110.10	117.65	121.37	113.07	121.43
DeformGS	101.81	107.85	123.38	117.62	124.91
WorldDream	82.45	88.15	91.58	77.08	91.12
DreamForge	73.20	83.34	79.76	65.71	80.88
Ours	**64.50**	**68.52**	**68.10**	**56.95**	**66.34**

Table 4: Quantitative evaluation of scene editing performance on heavy traffic scenes. Best results are in bold, second best are underlined.

platform to conduct closed-loop testing for end-to-end autonomous driving algorithms. DeformableGS and PVG both model agents and the background in a unified representation, making it impossible to directly identify which Gaussian blobs belong to the agents. For a fair comparison, we use the 3D bounding boxes of the agents to select the corresponding blobs, and then apply operations such as rotation or translation to these agent-specific Gaussian blobs.

3.3 Experiments Setting.

In the Fully Editable Driving World (EDW) module, we introduce dual-strategy agent reconstruction and depth-refinement background reconstruction methods. For dual-strategy agent reconstruction, we employ an online reconstruction approach for objects whose 3D bounding boxes contain more than 50 points and whose projected 2D bounding boxes maintain both height and width greater than 100 pixels. In all other cases, we utilize a Visual Degradation-aware Object Matching Module to match objects from a pre-built 3D model pool. It is important to note that if an object's 3D bounding boxes contain fewer than 10 points throughout the entire driving scenario, we do not modify the agent node of that object within the Gaussian Scene Graphs (NSG). During roi-guided background node training, we load the Gaussian parameters from the original scene and update both the properties and the number of Gaussian spheres according to the default settings of Gaussian Splatting (Kerbl et al. 2023), performing 500 steps of background finetuning. For behavior-level SCS training, the driving scenes are segmented into sequences of 5 seconds. Each sequence is used for training to generate hazardous trajectories. The Adam optimizer is used with a learning rate of 0.05 and a batch size of 1. Experiments are conducted on a single NVIDIA RTX 3090 GPU alongside 12th Gen Intel(R) Core(TM) i9-12900K CPU. The software setup operated on Ubuntu 22.04, with PyTorch version 2.0.0 and CUDA version 11.7.

4 Additional Experiments

4.1 OmniSCS on Challenging Scenes.

To evaluate the effectiveness and robustness of OmniSCS, we conducted experiments across a diverse range of challenging scenes. The results demonstrate that OmniSCS consistently maintains photorealism following scene editing, even under the most challenging conditions. This high level of photorealism forms the foundation for reliable SCS synthesis and closed-loop simulation in these challenging scenarios. This section provides the quantitative performance of the compared methods.

Method	Translation (FID)↓		Rotation (FID)↓		
	2.0m	3.5m	90°	180°	270°
StreetGS	66.91	80.08	100.84	92.57	91.23
Omnire	65.41	79.12	102.96	87.83	89.79
PVG	93.48	99.50	98.25	88.46	88.66
DeformGS	91.12	94.09	90.96	89.34	90.57
Ours	**63.62**	**76.73**	**88.55**	**73.19**	**82.20**

Table 5: Quantitative evaluation of scene editing performance on high-speed scenes.

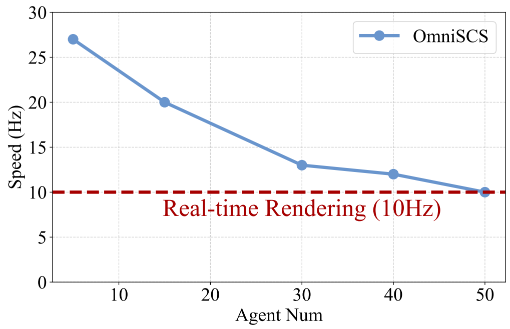


Figure 2: Efficiency evaluation analysis of OmniSCS.

Heavy Traffic. As summarized in Tab. 4, we compared FID scores for multiple methods on six heavy-traffic scenes from the Nuscenes dataset—each containing over 50 active agents. The evaluated scenes include scene-0003, scene-0105, scene-0127, scene-0399, scene-0443, and scene-0514. OmniSCS significantly surpasses all baselines, consistently achieving the lowest FID for both translation and rotation operations. This demonstrates its robustness in complex, agent-dense environments commonly encountered in urban autonomous driving.

High-Speed Scenarios. High-speed scenes pose considerable challenges for scene synthesis due to the presence of rapidly moving objects. Tab. 5 reports FID performance on the Waymo dataset's high-speed scenario (seg109239). OmniSCS achieves the best FID across all translation and rotation tasks. This highlights OmniSCS's superior ability to reconstruct and edit scenes with high object velocities, a critical capability for safe and reliable autonomous driving in real-world high-speed environments.

Adverse Weather. Rainy weather adds noise, occlusions, and reflections, complicating accurate scene editing. Tab. 6(a) highlights OmniSCS's robustness in NuScenes scene 0443, where it secures the lowest FID in all categories.

	(a) Rainy				**(b) Night**			
Method	Translation (FID) ↓	Rotation (FID) ↓			Translation (FID) ↓	Rotation (FID) ↓		
	3.5m	90°	180°	270°	3.5m	90°	180°	270°
StreetGS	89.37	108.66	91.20	113.23	95.46	128.10	127.60	130.99
Omnire	89.27	113.18	95.29	111.68	103.99	128.32	123.69	124.61
PVG	117.20	121.45	113.27	120.97	101.43	137.22	128.29	138.42
DeformableGS	114.29	135.13	132.54	137.41	127.59	133.55	127.08	136.53
Ours	**82.42**	**81.10**	**73.24**	**83.25**	**94.86**	**123.79**	**95.68**	**123.59**

Table 6: Quantitative evaluation of scene editing performance on challenging scenes.

Unlike baselines that degrade significantly, OmniSCS delivers consistent photorealism, making it ideal for autonomous driving in variable weather and enhancing reliability during storms or wet conditions.

Nighttime Scenes Low-light environments demand strong generalization to handle poor visibility. Tab. 6(b) evaluates performance in NuScenes scene 1098, with OmniSCS achieving the top overall FID for translations and rotations. This capability ensures dependable edits in dark settings, directly serving autonomous driving needs like nighttime operation and improving safety in low-illumination scenarios.

4.2 Efficiency Evaluation

We further evaluated the rendering speed of our method. Specifically, we measured the average rendering speed of OmniSCS in scenes with varying numbers of target objects. Fig. 2 demonstrates that OmniSCS can consistently achieve a rendering speed of 10 Hz even in complex scenes containing 50 objects, thereby meeting the requirements for real-time closed-loop simulation. These results clearly validate the efficiency and practicality of OmniSCS in AVs scenarios.

5 More Qualitative Results

In the following, we present qualitative visualizations to demonstrate the effectiveness of our OmniSCS method in scene editing and SCS synthesis.

In Fig. 3, RGB and depth visualizations reveal that the baseline method (OmniRe) exhibits noticeable appearance blurring and geometric inconsistencies. In contrast, OmniSCS preserves high-fidelity appearance and geometric continuity even under extreme rotation operations.

Fig. 4 shows that OmniSCS facilitates 1-to-N data synthesis by seamlessly compositing rare objects (e.g., deformed trucks) from 2D images into a 4D driving environment. The synthesized sequences in consecutive frames exhibit strong temporal and spatial consistency, illustrating our method's ability to generate high-quality, coherent data for SCS.

Fig. 5 further illustrates that OmniSCS supports 1-to-N SCS synthesis at the behavior level through the editing of agent trajectories. This allows for the generation of diverse, risky maneuvers and complex traffic interactions, showcasing the flexibility and practicality of our approach in creating varied behavior-level SCS.

In Fig. 6, We visualize the synthesis results under adverse weather and lighting conditions, such as rain and nighttime. OmniSCS produces images that remain highly consistent with the original scene across multiple frames and viewpoints, demonstrating strong generalization ability and robustness in challenging real-world scenarios.

6 Additional Ablation Study

6.1 Depth-refinement Background Reconstruction.

In this section, we conduct experiments by removing large static objects from the reconstructed world. Building on the results presented in the main paper, we provide additional renderings of the inpainted regions under novel trajectories. As shown in Fig. 7, our method maintains high visual quality in these novel views, producing realistic and coherent renderings without noticeable artifacts.

It is important to note that the primary goal of our background reconstruction is to prevent holes in the background that could appear after moving large static objects during behavior-level SCS synthesis and closed-loop testing. We do not aim to ensure perfect background consistency under drastic viewpoint changes—for instance, switching from driving on the left side of the inpainted area to the right side. Therefore, by leveraging precise depth constraints, our approach effectively achieves this targeted objective while remaining efficient and practical.

6.2 Behavior-level Safety-Critical Scenario Synthesis.

In this section, we further validate the effectiveness of the joint adversarial loss introduced during behavior-level SCS synthesis, which synergistically optimizes both the scene trajectory and the quality of rendered images. As illustrated in Fig. 8, without the $\mathcal{L}_{\text{app}}$ loss, modifications in the driving trajectory lead to severe data artifacts, rendering the generated data entirely unusable for downstream tasks. In contrast, as shown in Fig. 5, incorporating the $\mathcal{L}_{\text{app}}$ loss enables our method to produce artifact-free behavior-level SCS data that maintains continuity with the original driving scene.

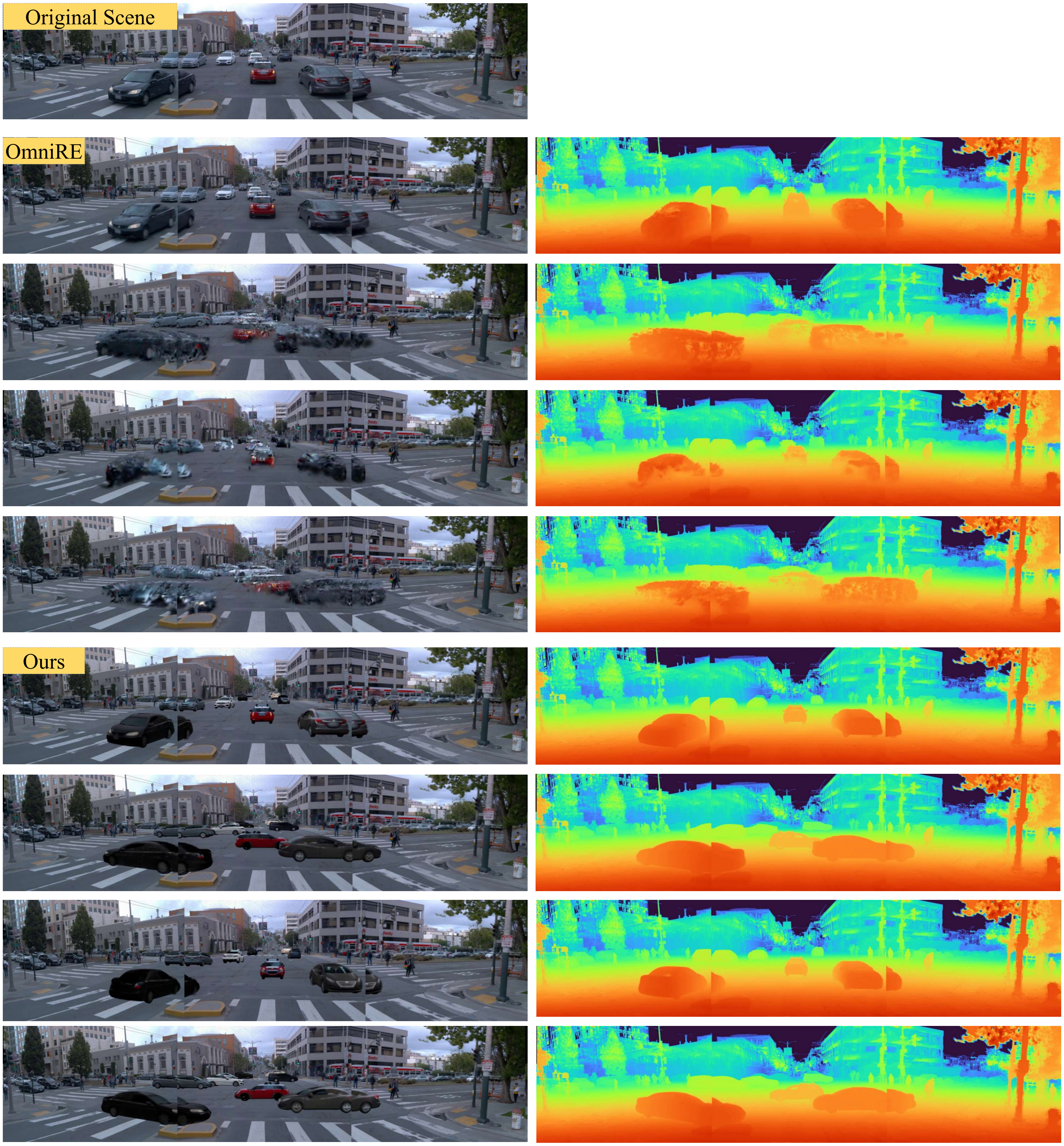


Figure 3: Qualitative comparison of scene editing results under extreme rotations. After scene editing, RGB visualizations reveal that the baseline method (OmniRe) suffers from noticeable appearance blurring. In the corresponding depth maps, geometric inconsistencies and missing structures are also observed.

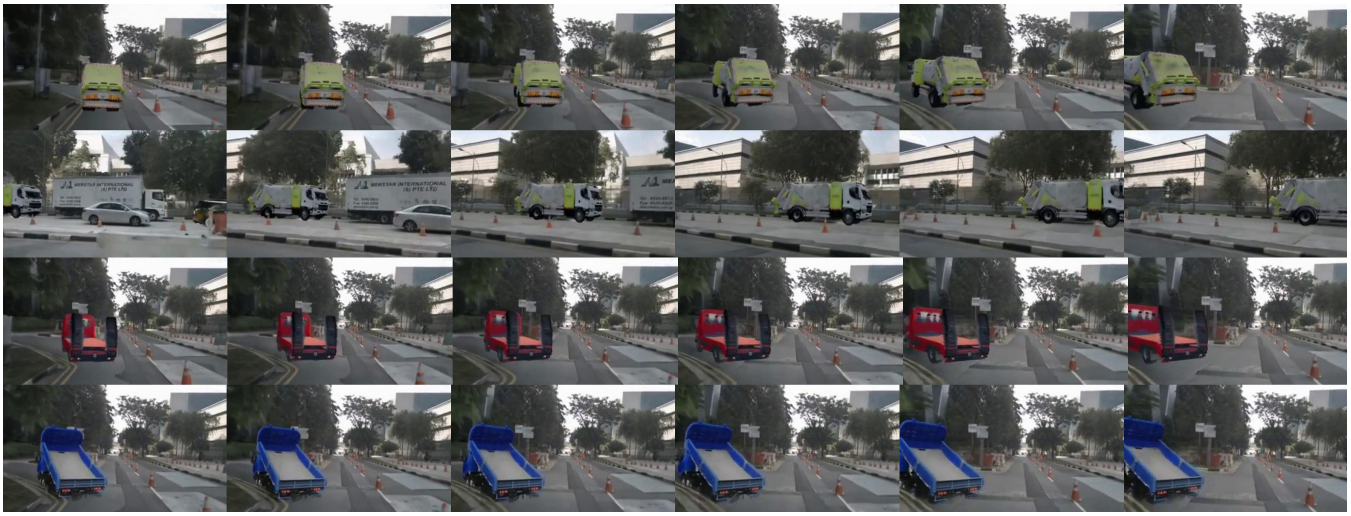
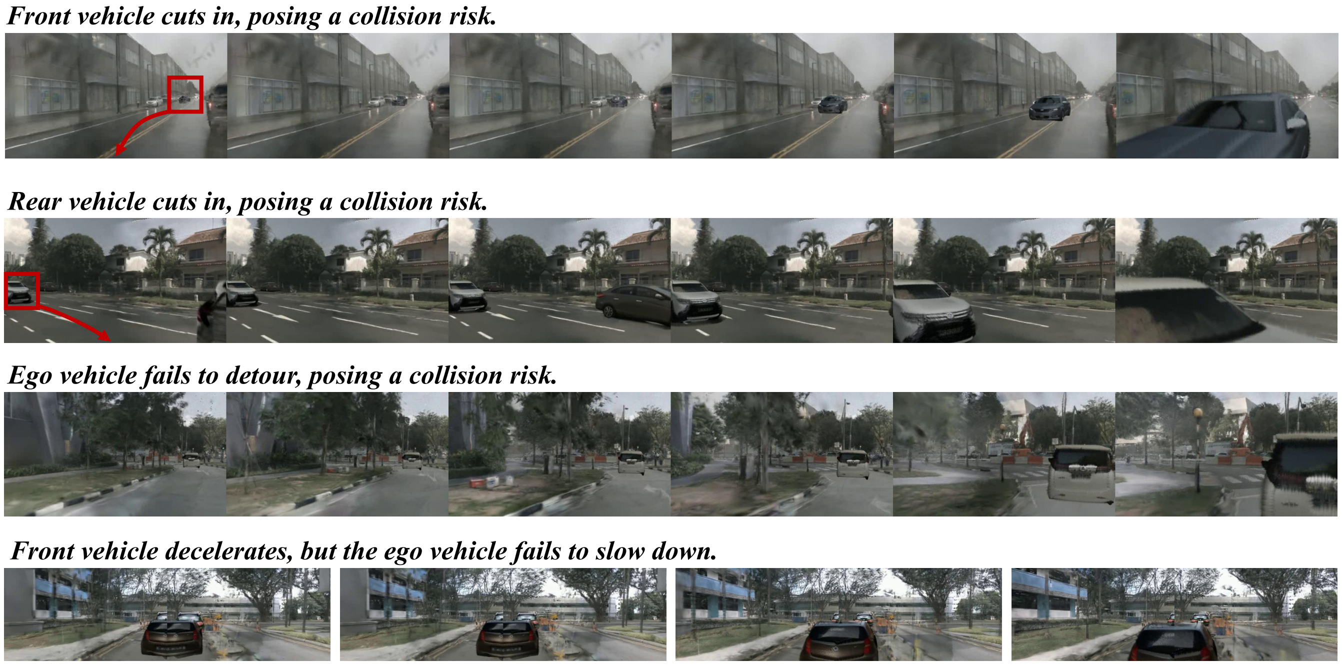


Figure 4: Appearance-level safety-critical scenario synthesis across consecutive frames. OmniSCS facilitates 1-to-N data synthesis by seamlessly compositing rare objects (e.g., deformed trucks) from 2D images into a 4D driving environment.



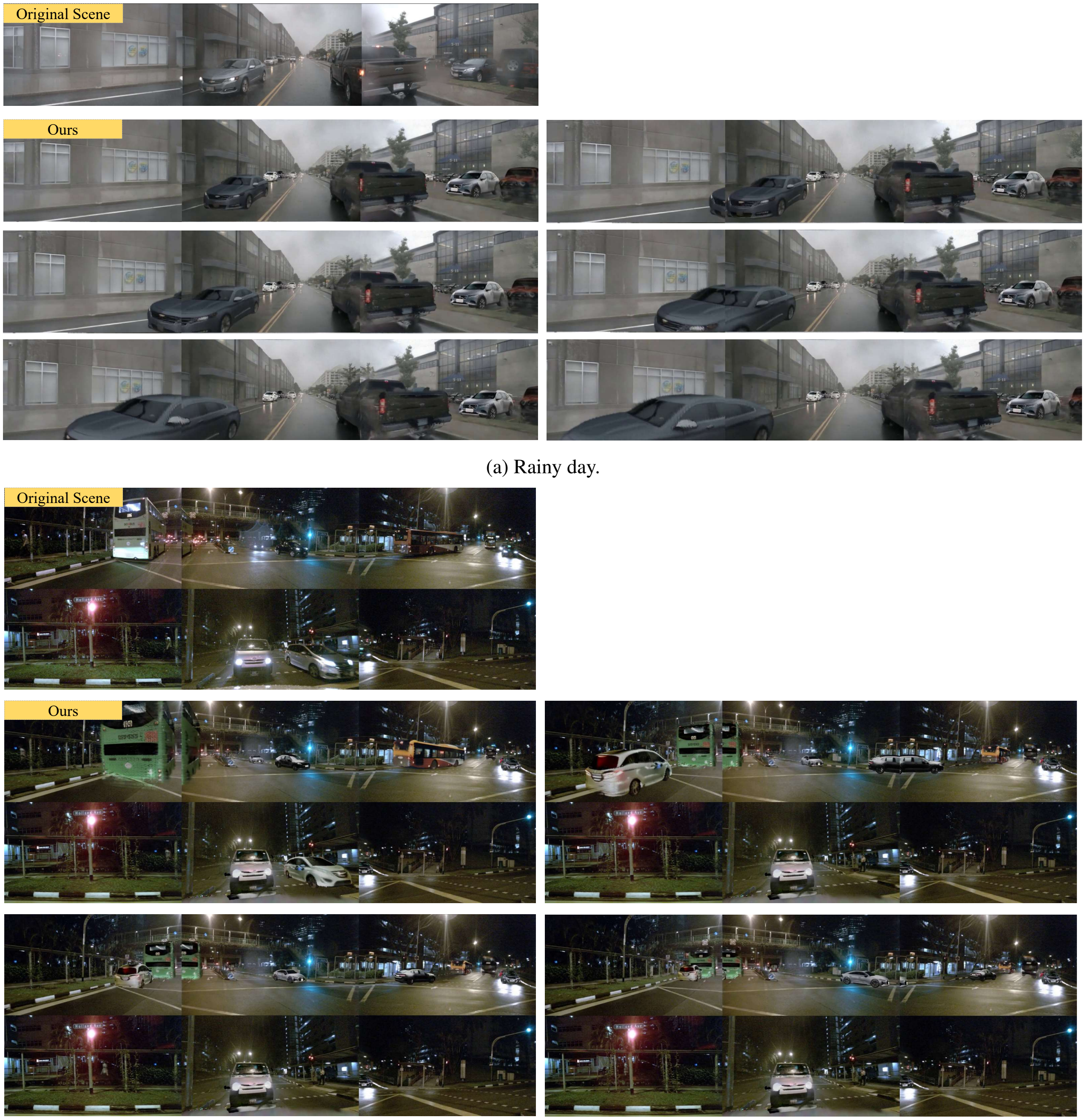
Front vehicle cuts in, posing a collision risk.

Rear vehicle cuts in, posing a collision risk.

Ego vehicle fails to detour, posing a collision risk.

Front vehicle decelerates, but the ego vehicle fails to slow down.

Figure 5: Behavior-level safety-critical scenario synthesis. OmniSCS enables 1-to-N SCS generation by editing agents' trajectories to produce risky behavioral maneuvers.



(a) Rainy day.

(b) Nighttime scenes.

Figure 6: Multi-view data synthesis across consecutive frames in challenging scenes.

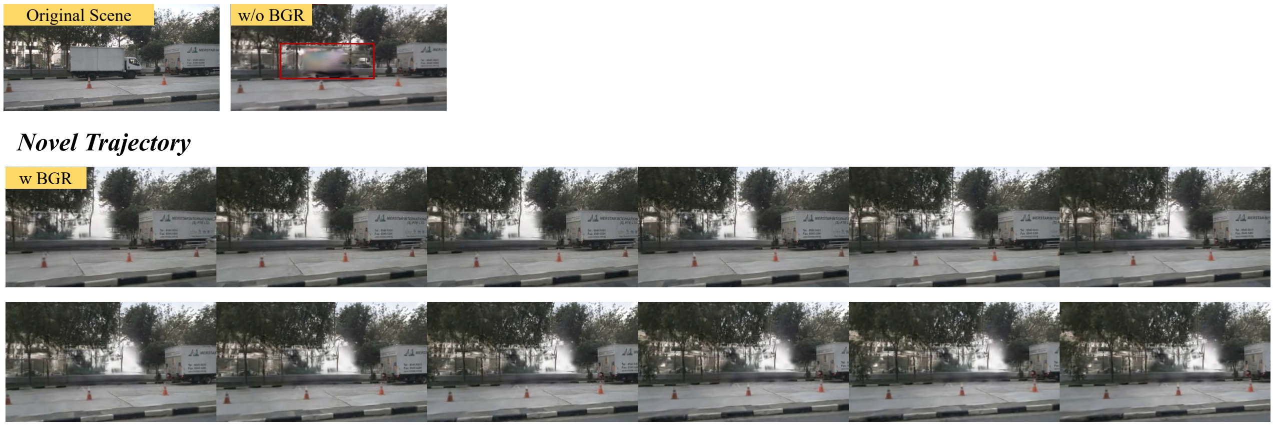


Figure 7: Ablation of depth-refinement background reconstruction (BGR). Our method maintains high visual quality in novel views, producing realistic and coherent renderings without noticeable artifacts.

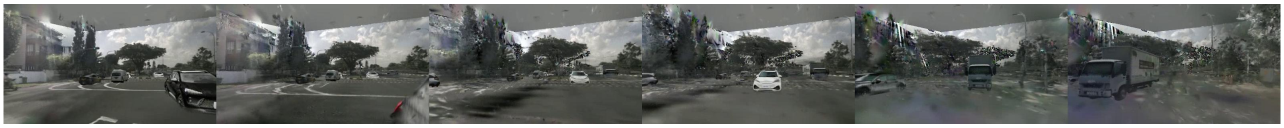


Figure 8: Ablation of the adversarial appearance loss $\mathcal{L}_{\text{app}}$. Without the $\mathcal{L}_{\text{app}}$ loss, modifications in the driving trajectory lead to severe data artifacts, rendering the generated data entirely unusable for downstream tasks.

References

Amini, A.; Wang, T.-H.; Gilitschenski, I.; Schwarting, W.; Liu, Z.; Han, S.; Karaman, S.; and Rus, D. 2022. VISTA 2.0: An Open, Data-driven Simulator for Multimodal Sensing and Policy Learning for Autonomous Vehicles. In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE.

Autohome. 2025. Autohome. https://www.autohome.com.cn.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631.

Chen, Y.; Gu, C.; Jiang, J.; Zhu, X.; and Zhang, L. 2023. Periodic Vibration Gaussian: Dynamic Urban Scene Reconstruction and Real-time Rendering. *arXiv:2311.18561*.

Chen, Z.; Yang, J.; Huang, J.; de Lutio, R.; Esturo, J. M.; Ivanovic, B.; Litany, O.; Gojcic, Z.; Fidler, S.; Pavone, M.; Song, L.; and Wang, Y. 2025. OmniRe: Omni Urban Scene Reconstruction. In *The Thirteenth International Conference on Learning Representations*.

Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; and Koltun, V. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*, 1–16. PMLR.

Gao, R.; Chen, K.; Xie, E.; Hong, L.; Li, Z.; Yeung, D.-Y.; and Xu, Q. 2024. MagicDrive: Street View Generation with Diverse 3D Geometry Control. In *International Conference on Learning Representations (ICLR)*.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4).

Kingma, D. P.; and Ba, J. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980.

Li, K.; Chen, K.; Wang, H.; Hong, L.; Ye, C.; Han, J.; Chen, Y.; Zhang, W.; Xu, C.; Yeung, D.-Y.; Liang, X.; Li, Z.; and Xu, H. 2022. CODA: A Real-World Road Corner Case Dataset for Object Detection in Autonomous Driving. In Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; and Hassner, T., eds., *Computer Vision – ECCV 2022*, 406–423. Cham: Springer Nature Switzerland.

Ma, E.; Zhou, L.; Tang, T.; Zhang, Z.; Han, D.; Jiang, J.; Zhan, K.; Jia, P.; Lang, X.; Sun, H.; et al. 2024. Unleashing generalization of end-to-end autonomous driving with controllable long video generation. *arXiv preprint arXiv:2406.01349*.

Mei, J.; Hu, T.; Yang, X.; Wen, L.; Yang, Y.; Wei, T.; Ma, Y.; Dou, M.; Shi, B.; and Liu, Y. 2025. DreamForge: Motion-Aware Autoregressive Video Generation for Multi-View Driving Scenes. arXiv:2409.04003.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Reddi, S. J.; Kale, S.; and Kumar, S. 2019. On the Convergence of Adam and Beyond. arXiv:1904.09237.

Rempe, D.; Philion, J.; Guibas, L. J.; Fidler, S.; and Litany, O. 2022. Generating Useful Accident-Prone Driving Scenarios via a Learned Traffic Prior. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Roeder, L. 2025. Netron. Visualizer for neural network, deep learning, and machine learning models.

Rong, G.; Shin, B. H.; Tabatabaee, H.; Lu, Q.; Lemke, S.; Možeiko, M.; Boise, E.; Uhm, G.; Gerow, M.; Mehta, S.; Agafonov, E.; Kim, T. H.; Sterner, E.; Ushiroda, K.; Reyes, M.; Zelenkovsky, D.; and Kim, S. 2020. LGSVL Simulator: A High Fidelity Simulator for Autonomous Driving. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 1–6. IEEE Press.

Shah, S.; Dey, D.; Lovett, C.; and Kapoor, A. 2017. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics: Results of the 11th international conference*, 621–635. Springer.

Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2446–2454.

Tan, S.; Ivanovic, B.; Weng, X.; Pavone, M.; and Kraehenbuehl, P. 2023. Language Conditioned Traffic Generation. In *7th Annual Conference on Robot Learning*.

Wang, J.; Pun, A.; Tu, J.; Manivasagam, S.; Sadat, A.; Casas, S.; Ren, M.; and Urtasun, R. 2021. Advsim: Generating safety-critical scenarios for self-driving vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9909–9918.

Wang, Y.; He, J.; Fan, L.; Li, H.; Chen, Y.; and Zhang, Z. 2024. Driving into the Future: Multiview Visual Forecasting and Planning with World Model for Autonomous Driving. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Woo, S.; Park, J.; Lee, J.-Y.; and Kweon, I. S. 2018. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, 3–19.

Wu, Z.; Liu, T.; Luo, L.; Zhong, Z.; Chen, J.; Xiao, H.; Hou, C.; Lou, H.; Chen, Y.; Yang, R.; Huang, Y.; Ye, X.; Yan, Z.; Shi, Y.; Liao, Y.; and Zhao, H. 2023. MARS: An Instance-aware, Modular and Realistic Simulator for Autonomous Driving. *CICAI*.

Xiang, J.; Lv, Z.; Xu, S.; Deng, Y.; Wang, R.; Zhang, B.; Chen, D.; Tong, X.; and Yang, J. 2025. Structured 3D Latents for Scalable and Versatile 3D Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yan, Y.; Lin, H.; Zhou, C.; Wang, W.; Sun, H.; Zhan, K.; Lang, X.; Zhou, X.; and Peng, S. 2024. Street Gaussians for Modeling Dynamic Urban Scenes. In *ECCV*.

Yang, X.; Wen, L.; Ma, Y.; Mei, J.; Li, X.; Wei, T.; Lei, W.; Fu, D.; Cai, P.; Dou, M.; Shi, B.; He, L.; Liu, Y.; and Qiao, Y. 2025. DriveArena: A Closed-loop Generative Simulation Platform for Autonomous Driving.

Yang, Z.; Gao, X.; Zhou, W.; Jiao, S.; Zhang, Y.; and Jin, X. 2024. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 20331–20341.

Zhao, G.; Ni, C.; Wang, X.; Zhu, Z.; Zhang, X.; Wang, Y.; Huang, G.; Chen, X.; Wang, B.; Zhang, Y.; Mei, W.; and Wang, X. 2025. DriveDreamer4D: World Models Are Effective Data Machines for 4D Driving Scene Representation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhou, H.; Lin, L.; Wang, J.; Lu, Y.; Bai, D.; Liu, B.; Wang, Y.; Geiger, A.; and Liao, Y. 2024. HUGSIM: A Real-Time, Photo-Realistic and Closed-Loop Simulator for Autonomous Driving. arXiv:2412.01718.

Zhou, K.; Yang, Y.; Cavallaro, A.; and Xiang, T. 2019. Omni-scale feature learning for person re-identification. In *Proceedings of the IEEE/CVF international conference on computer vision*, 3702–3712.