CoDa-4DGS: Dynamic Gaussian Splatting with Context and Deformation Awareness for Autonomous Driving Supplementary Material

Rui Song[*1,2], Chenwei Liang[*1], Yan Xia[2,3], Walter Zimmer[2], Hu Cao[2]
Holger Caesar[4], Andreas Festag[1,5], Alois Knoll[2]

[1]Fraunhofer IVI [2]TU Munich [3]USTC [4]TU Delft [5]TH Ingolstadt

https://rruisong.github.io/publications/CoDa-4DGS

In this supplementary material, we provide additional implementation details in Sec. A. In Sec.B, we showcase the plug-and-play functionality of our approach, demonstrating how CoDa-4DGS enhances the performance of both vanilla 4DGS [8] and S^3Gaussian [4]. Sec. C presents visualization results for 4D dynamic scene editing, highlighting the distinctions between our method and prior work that predominantly focuses on 3D. In Sec. D, we provide visual results for novel view synthesis, addressing scenarios with large ego-view angle shifts, thereby extending beyond previous evaluations that primarily consider small frame transitions in test sets. Additionally, Sec. E offers an in-depth conceptual comparison with recent works, while Sec.F provides experimental results on efficiency and explores potential improvements. Sec. G includes extended 4D visualizations to demonstrate the robustness and versatility of our approach.

A. Implementation details

In the implementation, for context awareness, we follow and use LSeg [6] to maintain 128-dimensional semantic features that link each Gaussian with temporal deformation. Thus, context awareness is represented by aggregated semantic features across all Gaussians, *i.e.* $\mathbf{f}_{con} \in \mathbb{R}^{N\times 128}$. Temporal deformation awareness is built on $\Delta\mathcal{G}$, such that $\mathbf{f}_{def} \in \mathbb{R}^{N\times 62}$, where for SH coefficients $k = 48$. Additionally, the frame information is binarized and encoded as a periodic function to generate a time embedding $\mathbf{f}_{time} \in \mathbb{R}^{N\times 64}$. Since our primary comparison is with S^3Gaussian [4], we adopted similar hyperparameters. We train for 50,000 steps, with a learning rate set to $1.6e^{-3}$, which decays to $1.6e^{-4}$. For our loss function, we assign weights for each as follows: $\lambda_{rgb} = 1$, $\lambda_{d\text{-}ssim} = 0.2$, $\lambda_{tv} = 1$, $\lambda_{depth} = 0.5$, $\lambda_f = 1$.

* Equal contribution
Corresponding author, email address: rui.song@ivi.fraunhofer.de

B. Plug-and-play

The core functionality of CoDa-4DGS lies in extracting temporal deformation awareness and context awareness, followed by Gaussian deformation compensation using DCN. This streamlined interface design makes CoDa-4DGS a plug-and-play method. When integrating CoDa-4DGS, we only need to focus on two aspects: acquiring temporal deformation awareness and context awareness. Since context awareness depends on components related to 2D foundation models, selecting an appropriate foundation model is essential to complement CoDa-4DGS effectively. For temporal deformation awareness, it is crucial to ensure that the embedded method can extract temporal deformation, such as the vanilla 4DGS.

To validate the plug-and-play functionality of CoDa-4DGS and its performance improvements over baseline methods, we conducted ablation studies using vanilla 4DGS and S^3Gaussian on Scene 22 and Scene 02, respectively. To ensure a fair comparison, we used identical hyperparameters, including learning rate, number of iterations, and the number of frames. As shown in Tab. 1, incorporating CoDa-4DGS led to performance improvements across all metrics for both vanilla 4DGS and S^3Gaussian, with approximately a 2% increase in global PSNR. Notably, the improvement in dynamic PSNR was even more significant, aligning with the findings in the main paper. These results demonstrate that CoDa-4DGS effectively enhances rendering performance for dynamic objects through deformation compensation, making it particularly beneficial for autonomous driving scenarios.

C. Scene editing

In CoDa-4DGS, each Gaussian is trained to encode 4D semantic features, enabling context awareness. This allows us to use text encoding to generate a corresponding refer-

Table 1. Enhancing scene reconstruction accuracy via Plug-and-Play integration of CoDa-4DGS for Scene 22 and Scene 02.

Method	Scene 22				Scene 02			
Metric	PSNR↑	SSIM↑	PSNR*↑	SSIM*↑	PSNR↑	SSIM↑	PSNR*↑	SSIM*↑
4DGS (CVPR 24)	32.54	0.9327	29.77	0.8767	29.14	0.8837	24.19	0.7965
4DGS+CoDa-4DGS	33.12	0.9437	30.53	0.8802	29.84	0.8902	24.96	0.8042
Improvement	+0.58	+0.0110	+0.76	+0.0035	+0.70	+0.0065	+0.77	+0.0077
S^3Gaussian (ECCV 24)	33.49	0.9367	29.79	0.8832	30.14	0.8998	24.78	0.8087
S^3Gaussian+CoDa-4DGS	34.09	0.9441	31.14	0.8935	30.37	0.9030	25.26	0.8150
Improvement	+0.60	+0.0074	+1.35	+0.0103	+0.23	+0.0032	+0.48	+0.0063

[1] * indicates that the metrics are calculated only for dynamic objects.

Figure 1. Object decomposition for scene editing. We extract an object from one scene (Scene 22) and integrate it into other scenes (Top: Scene 16 and Bottom: Scene 86). This process involves fusing two Gaussian models, leveraging instance decomposition in a 4D spatial-temporal space to achieve realistic synthesis and consistent object placement.

ence feature and, through cosine similarity and clustering, decompose the associated 4D Gaussians for a specific instance.

In Fig. 1, we demonstrate how context awareness can be used to extract an object from one scene and place it into another. Notably, the rendering of the edited scene is not achieved by directly rasterizing the object's corresponding 4D Gaussian onto the previous image. Instead, the 4D Gaussians are merged, allowing the object's Gaussian to deform in tandem with the temporal dynamics of the new scene. As shown in Figure 1, the spatial relationships inherent in the 4D Gaussians ensure that the newly added object can be partially occluded by other vehicles in the scene.

In Fig. 2, we showcase how the newly added synthetic 4D object can be manipulated within the new scene. By simultaneously rotating and translating the vehicle, the synthetic 4D object can be positioned with various poses in different locations. In the attached video, we provide demonstrations of these capabilities.

D. Novel view synthesis

Novel view synthesis involves rendering camera perspectives that were not included in the training data. In autonomous driving, this capability is essential for photorealistic closed-loop simulations, particularly in validating end-to-end autonomous driving systems. Current benchmarks address this challenge primarily in two ways: (*i*) utilizing simulator engines like CARLA [3], and (*ii*) leveraging Bird's Eye View (BEV) abstractions, as exemplified by the method proposed in NAVSIM [2], which serves as a benchmark for the CVPR 2024 Autonomous Grand Challenge. NAVSIM enables short closed-loop simulations built on the nuPlan dataset [1]. However, achieving photorealistic novel view synthesis remains a significant challenge.

This difficulty stems from the nature of real-world data collected for autonomous driving, which is typically captured using cameras mounted on vehicles. Camera movements are constrained by vehicle trajectories, often limited to simple, linear paths or curves. These restrictions in the

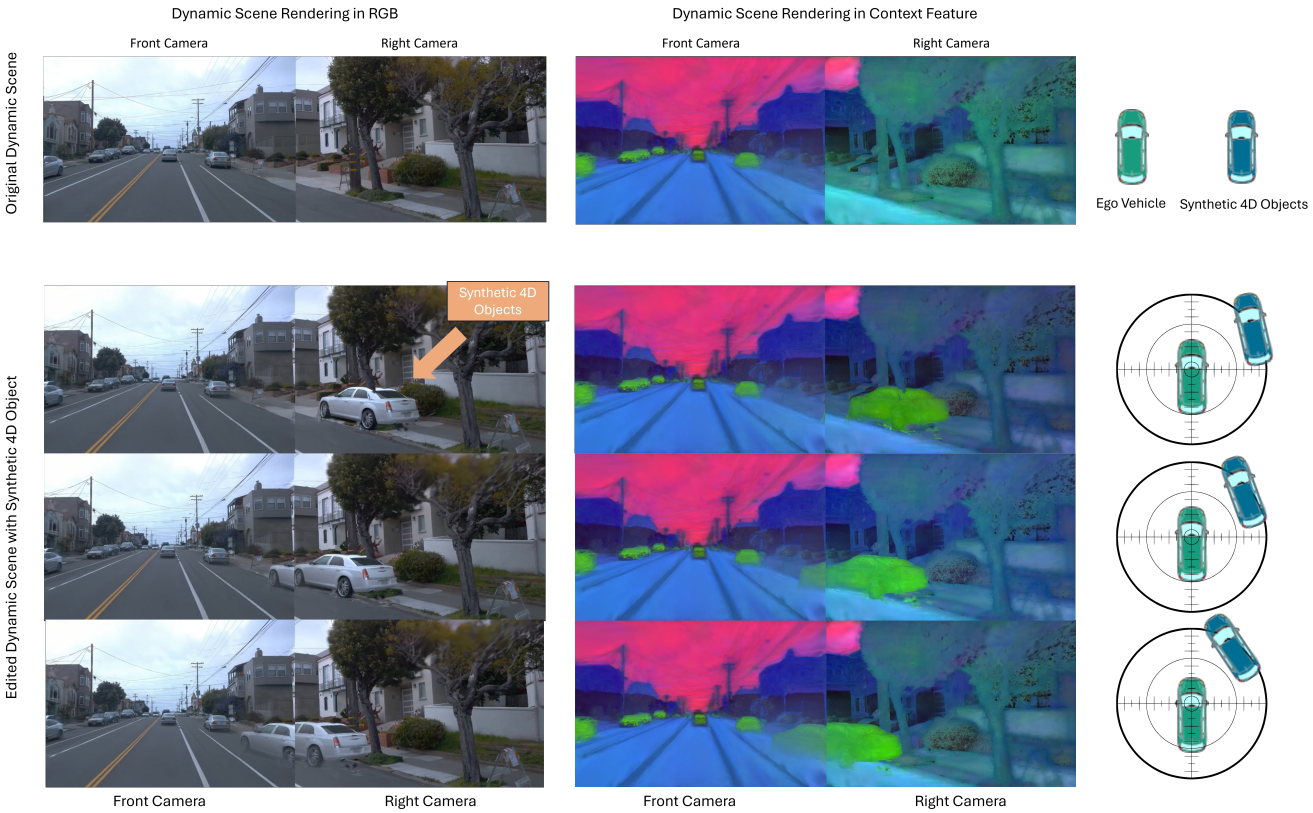

Figure 2. Manipulating synthetic 4D objects for scene editing by adjusting their poses and positions. This is achieved through translation and rotation of the object's Gaussian representations.

training data result in scene reconstructions that struggle to meet the requirements for novel view synthesis.

In studies such as EmerNeRF [11], MARS [9], and StreetGaussian [10], novel view synthesis is evaluated by dividing a scene's frames into training and test sets. For instance, EmerNeRF designates every 10th frame as the test set and the rest as the train set. While this benchmark method provides ground truth for benchmarking, and we adopt the same approach for quantitatively evaluating novel view synthesis performance, it falls short in meeting the requirements of closed-loop simulation, where novel views must be generated under diverse ego poses and dynamic trajectories.

Toward photorealistic closed-loop simulation, we showcase the capabilities of CoDa-4DGS in Fig. 3. By making slight adjustments to the ego camera's angles in both positive and negative directions, CoDa-4DGS generates novel views that do not exist in the dataset. Additionally, it supports novel view synthesis for 4D semantic segmentation, offering a versatile tool for various scenarios. A detailed demonstration of this capability is included in the attached video.

E. Conceptual comparison

To incorporate the latest advancements, we compare our method with S^3Gaussian [4] and StreetGaussian [10]. Unlike StreetGaussian, which requires ground truth for training, our approach is built upon 4DGS and S^3Gaussian and leverages self-supervised learning. While ground truth offers precise tracking priors for dynamic objects in a scene, the self-supervised approach enhances scalability by eliminating the need for labeled bounding boxes.

Additionally, it is worth emphasizing that our method can serve as a plug-and-play module to enhance other frameworks based on Gaussian temporal deformation. As demonstrated in Sec B, we focus on leveraging context awareness and deformation awareness to compensate for inaccuracies in vanilla Gaussian deformation predictions, thereby improving overall performance. For instance, a related method, DN-4DGS [7], also serves as a plug-and-play module and improves PSNR by 1.4% for vanilla 4DGS. In comparison, our approach achieves an improvement of approximately 2%. Moreover, DN-4DGS is not explicitly designed for autonomous driving scenes.

Using 2D foundation models to distill features has proven effective in 3DGS tasks [12, 13], especially with se-

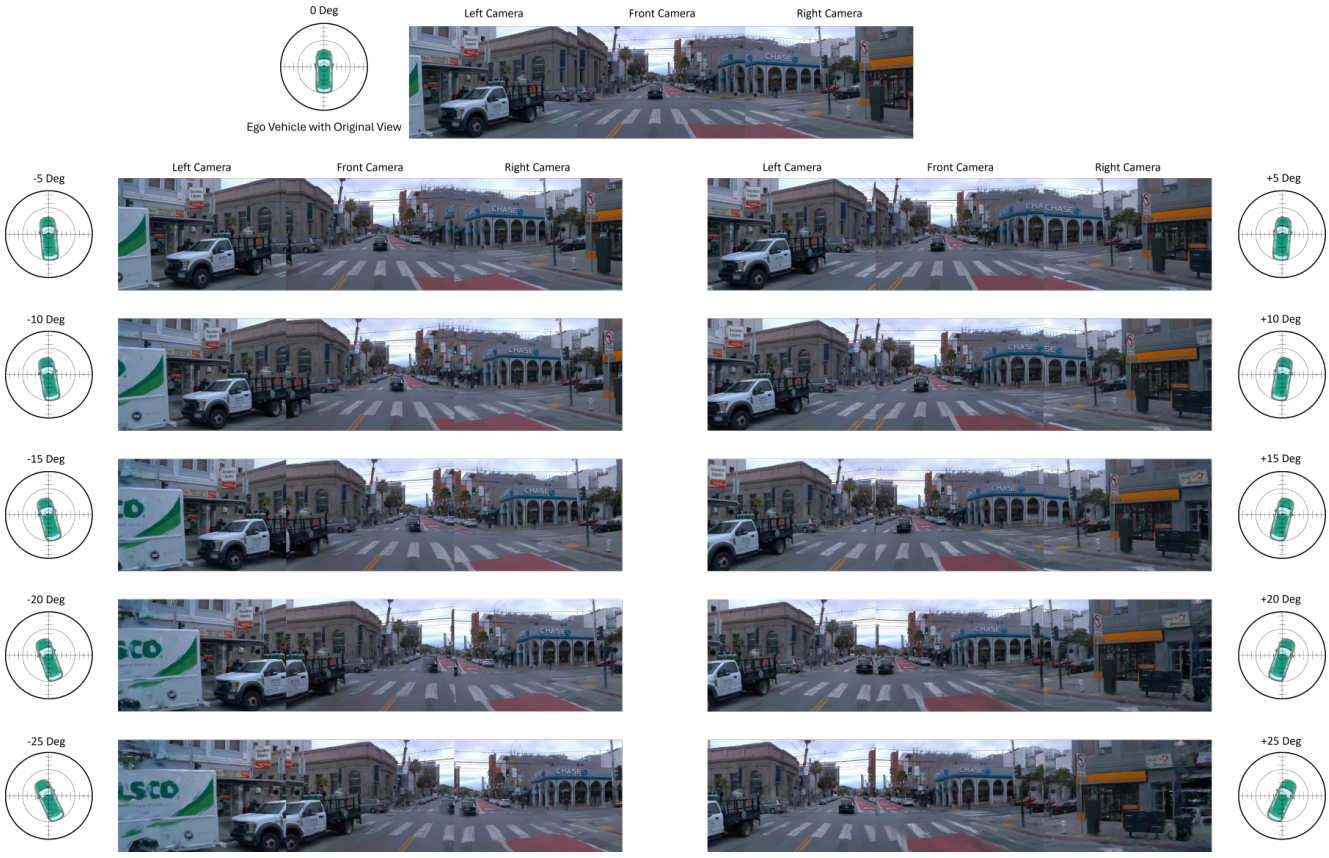

Figure 3. Novel view synthesis by freezing the scene in time and altering the cameras' perspective. The resulting views are synthetic images that do not exist in the dataset and have never been observed before.

Table 2. Conceptual comparison of very recent advancements in GS-based scene rendering approaches.

Approach	4D Scene	Autonomous Driving	Self-Supervised	Feature Distillation	Plug-and-Play
Feature-3DGS [12]			✓	✓	
FMGS [13]			✓	✓	
StreetGaussian [10]	✓	✓			
S^3Gaussian [4]	✓	✓	✓		
DN-4DGS [7]	✓		✓		✓
CoDa-4DGS (Ours)	✓	✓	✓	✓	✓

mantic information, which significantly aids scene understanding [5]. Semantic feature-supported 3DGS can enable downstream tasks like scene editing in static scenarios. We extend this idea to 4DGS, which is not straightforward and involves additional complexity. In 4DGS, each Gaussian undergoes temporal deformation, and the associated semantic features must also be transformed accordingly to maintain consistency in context after deformation. For example, if a Gaussian represents different objects at different time steps, its semantic features should adapt to reflect the new semantics, ensuring alignment between the rendered results and the foundation model's inferences. This consistency is essential for enabling CoDa-4DGS to perform semantic-aware applications, such as scene editor.

Furthermore, by incorporating semantic features as context-awareness inputs into the Deformable Compensation Network (DCN), we can constrain Gaussian deformation in spatial dimensions, thereby improving training outcomes. For instance, a Gaussian representing a road surface at one time step should remain consistent as a road surface

after temporal deformation rather than erroneously transforming into a car due to proximity in 3D space. This is because the semantic feature distance between the two would be significantly larger despite their spatial proximity.

In Tab. 2, we summarize the above discussions, highlighting the key distinctions between our approach and other very recent methods.

F. Efficiency improvement

Due to the increased feature dimensionality required by CoDa-4DGS for supporting Gaussian deformation, its inference speed is reduced compared to the vanilla 4DGS. To improve efficiency and better accommodate potential real-time applications, we conduct additional experiments using mixed-precision computation and graph compilation, achieving a 2.47× speedup. Specifically, on an NVIDIA A100, the per-frame inference time is 0.037/0.038 seconds on the Waymo/KITTI datasets, corresponding to 27.02/26.45 FPS, respectively. On an NVIDIA RTX 3090, the inference time increases to 0.078/0.079 seconds, corresponding to 12.75/12.66 FPS. Furthermore, additional optimization strategies such as sparsification and quantization hold promise for further improving the efficiency of CoDa-4DGS.

G. Further visual results

To provide a more intuitive demonstration of CoDa-4DGS's performance on dynamic scenes, we present the RGB rendering results, semantic feature rendering results, and ground truth for each frame in chronological order. CoDa-4DGS consistently achieves exceptional 4D scene rendering across diverse scenarios, as shown in Scene 86 (Fig.4), Scene 80 (Fig.5), Scene 03 (Fig.6), and Scene 22 (Fig.7).

References

[1] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *Computer Vision and Pattern Recognition (CVPR) 2021 ADP3 workshop*, 2021. 2

[2] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2

[3] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 2

[4] Nan Huang, Xiaobao Wei, Wenzhao Zheng, Pengju An, Ming Lu, Wei Zhan, Masayoshi Tomizuka, Kurt Keutzer, and Shanghang Zhang. S3gaussian: Self-supervised street gaussians for autonomous driving. *arXiv preprint arXiv:2405.20323*, 2024. 1, 3, 4

[5] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12871–12881, 2022. 4

[6] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *International Conference on Learning Representations (ICLR)*, 2022. 1

[7] Jiahao Lu, Jiacheng Deng, Ruijie Zhu, Yanzhe Liang, Wenfei Yang, Tianzhu Zhang, and Xu Zhou. Dn-4dgs: Denoised deformable network with temporal-spatial aggregation for dynamic scene rendering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 3, 4

[8] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024. 1

[9] Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao Chen, Runyi Yang, et al. Mars: An instance-aware, modular and realistic simulator for autonomous driving. In *CAAI International Conference on Artificial Intelligence*, pages 3–15. Springer, 2023. 3

[10] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians for modeling dynamic urban scenes. In *European Conference on Computer Vision*, 2024. 3, 4

[11] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, and Yue Wang. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023. 3

[12] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024. 3, 4

[13] Xingxing Zuo, Pouya Samangouei, Yunwen Zhou, Yan Di, and Mingyang Li. Fmgs: Foundation model embedded 3d gaussian splatting for holistic 3d scene understanding. *International Journal of Computer Vision*, pages 1–17, 2024. 3, 4

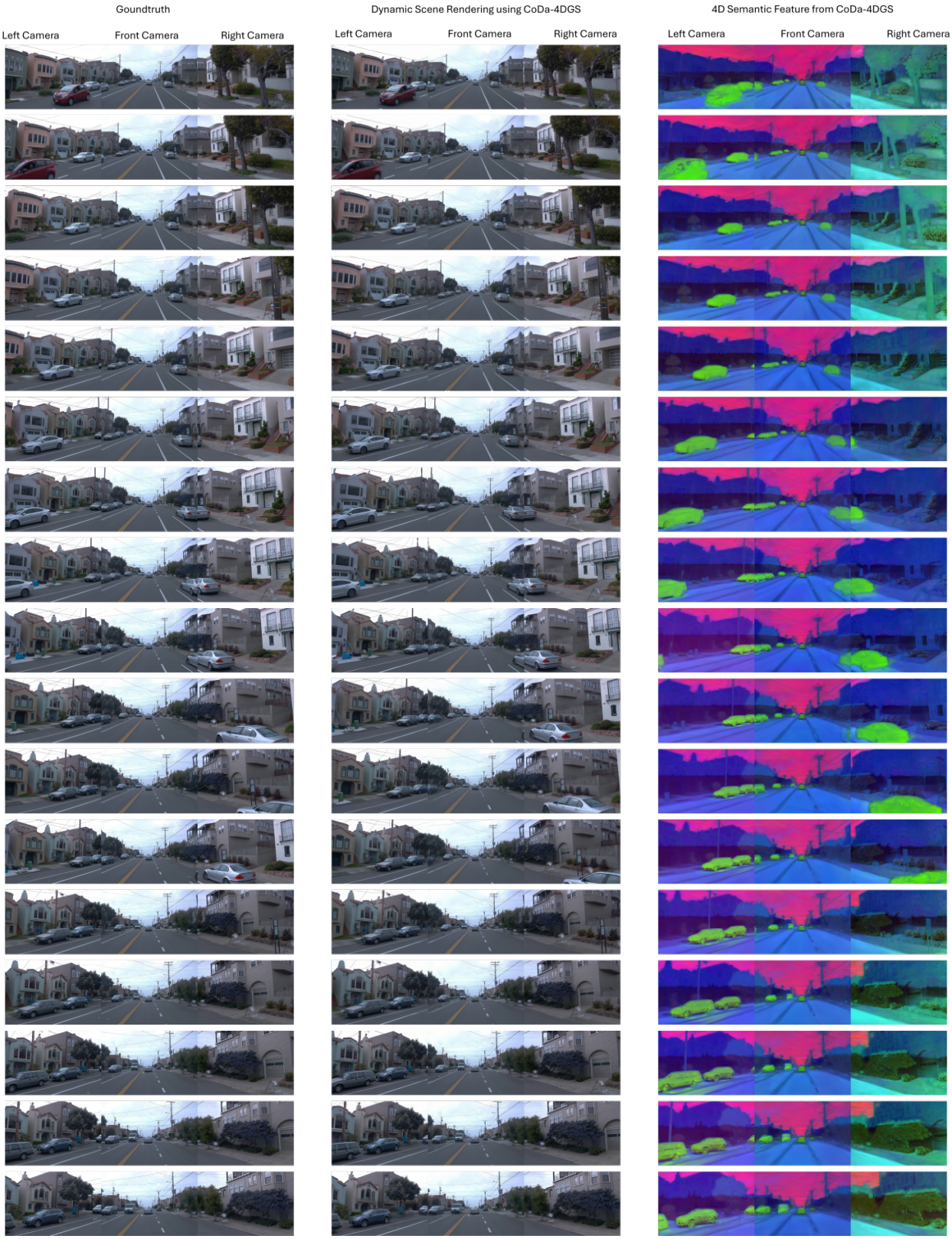

Figure 4. 4D scene rendering for scene 86.

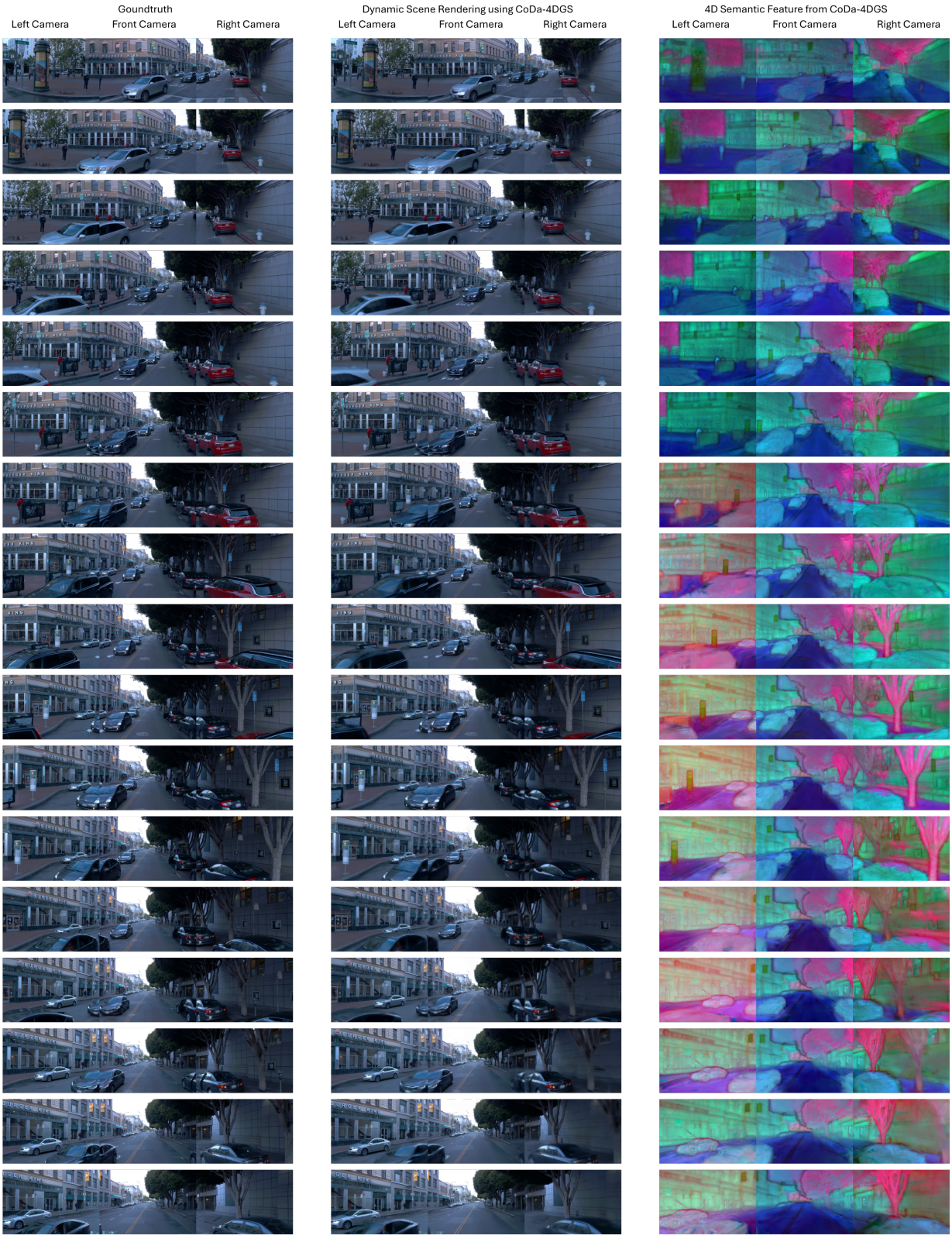

Figure 5. 4D scene rendering for scene 80.

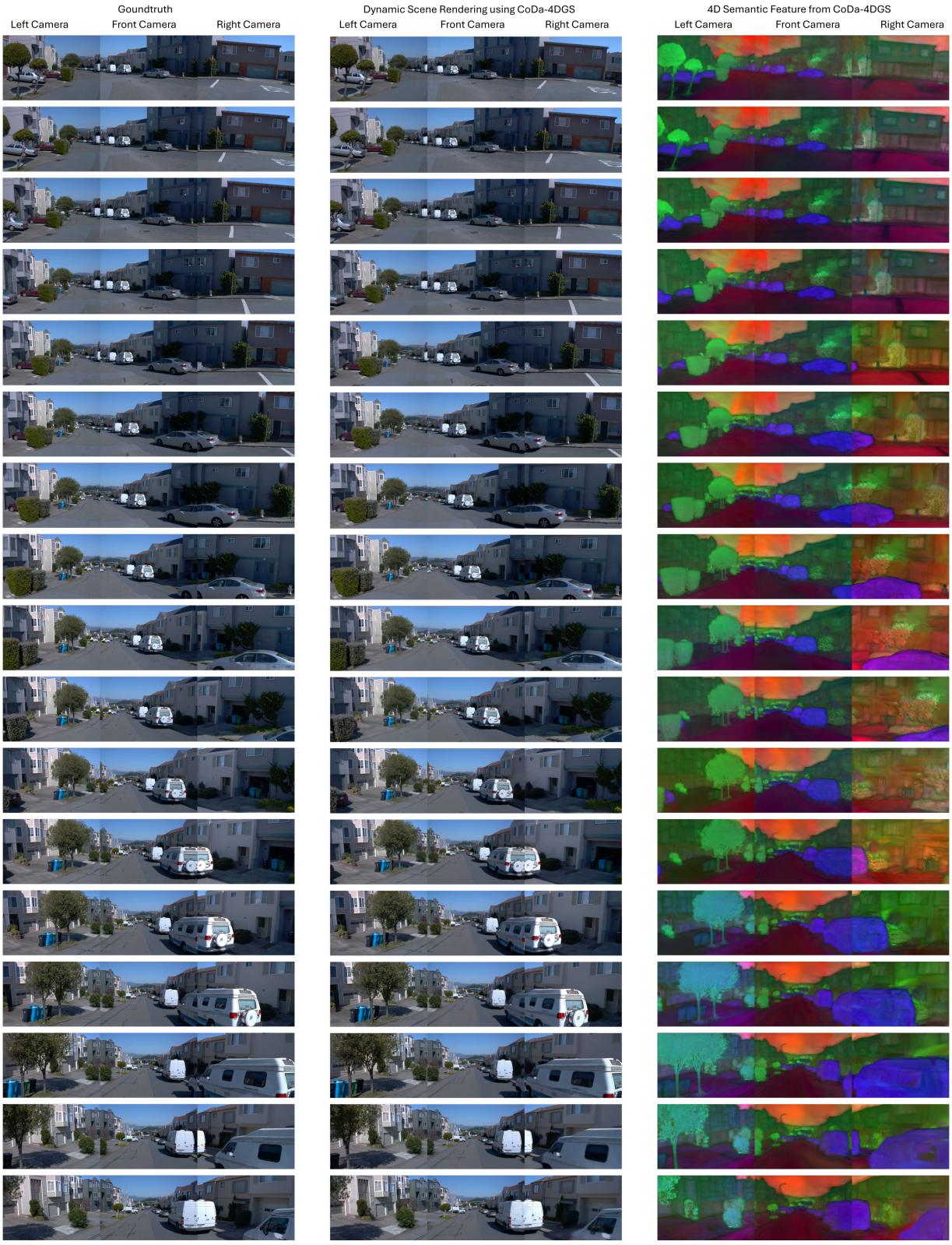

Figure 6. 4D scene rendering for scene 03.

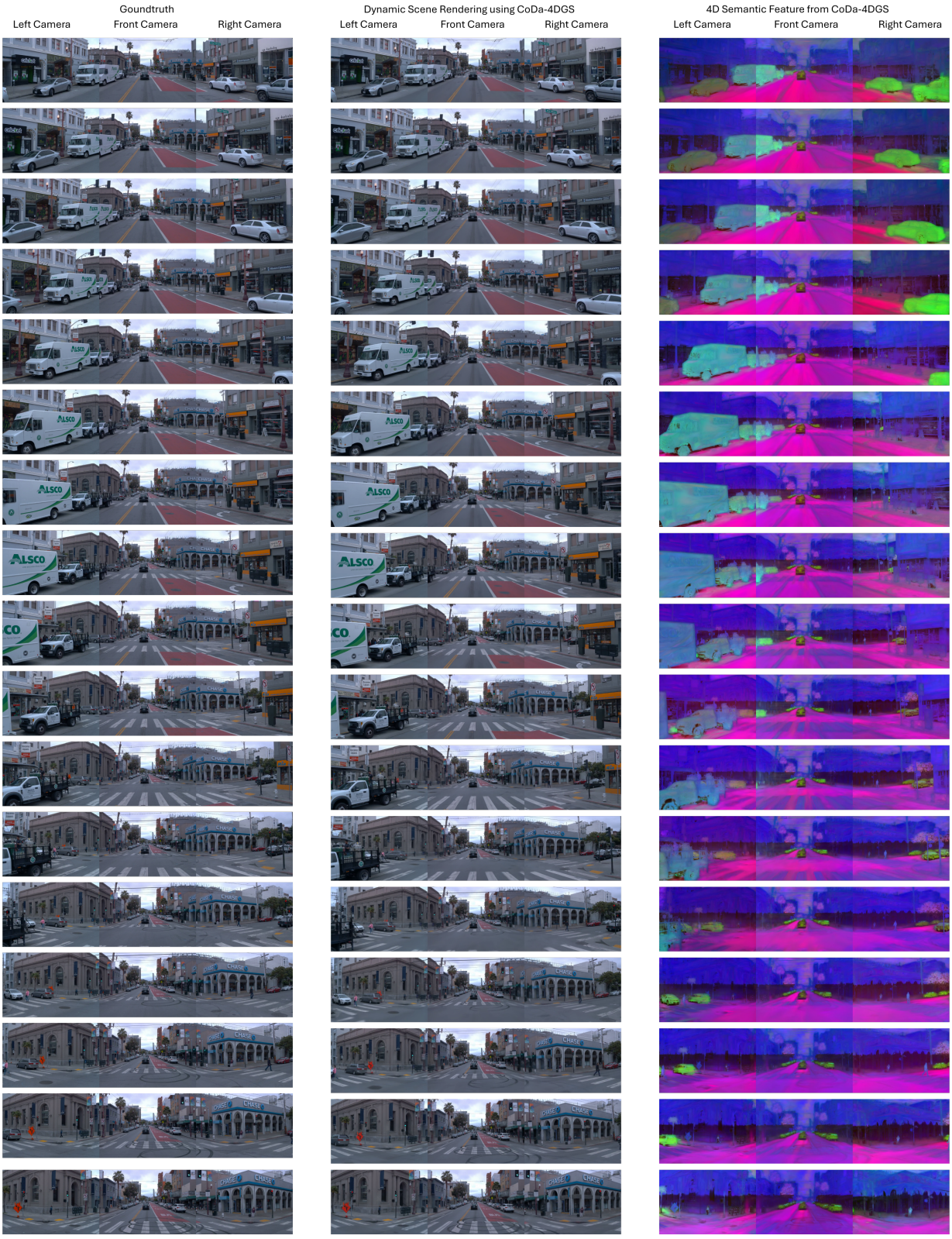

Figure 7. 4D scene rendering for scene 22.