Grounded Latents for Entity-Centric 4D Scene Generation (Supplementary Materials)

Jinhyung Park[1] Navyata Sanghvi[1] Erica Weng[1] Shawn Hunt[3] Shinya Tanaka[3]
Hironobu Fujiyoshi[2] Kris Kitani[1]
[1]Carnegie Mellon University [2]DENSO Corporation [3]DENSO International America, Inc.

A. Supplementary Overview

This supplementary provides additional implementation details, ablations, and qualitative analysis, together with a supplementary video.

Section B briefly describes the content of the video. Sections C–H then provide the technical details and extended results:

- Section C gives precise details on evaluation models and metrics.
- Section D outlines implementation details of the proposed model.
- Section E presents in-depth ablations on the feature dimension of the grounded latent space, the proposed push weight for outpainting, and conditioning signals for motion generation.
- Section F provides additional results: 4D occupancy forecasting and motion prediction metrics, vehicle geometry diversity analysis, and pedestrian generation examples.
- Section G provides qualitative visualizations of generated 3D scenes on the CarlaSC [4, 18] dataset.
- Section H extensively visualizes our grounded latents and reconstruction results on the Waymo [14, 15] dataset.

B. Supplementary Videos

The supplementary material includes two videos: one showing generations from DynamicCity and another showing generations from our 4D scene generation method. Our framework synthesizes more realistic 4D scenes:

- **Foreground coherence.** DynamicCity frequently fails to preserve vehicle and pedestrian identity, with actors flickering or merging across frames, while our actors remain consistent over time.
- **Ego motion and background stability.** Since DynamicCity encodes ego motion only through global changes, it struggles to separate ego and actor motion during turns, causing merges and unstable background geometry.
- **Grounded latent formulation.** Our method uses one grounded latent per foreground object, explicitly models ego transforms, and generates actor motion directly, yielding precise trajectories and more consistent foreground and background geometry.

C. Details on Evaluation Metrics

We evaluate generation quality by comparing feature distributions of real and generated scenes using pretrained 3D autoencoders [2], following prior work [1, 10, 11, 17]. Each scene is passed through the autoencoder's encoder, and we use its bottleneck features for evaluation. The network has four $2\times$ downsampling stages in x, y, z. A Waymo grid of $200\times200\times16$ with downsampling rate $d = 16$ produces a BEV bottleneck of 12×12 with channel dimension $C = 256$; the decoder then upsamples back to the original voxel grid. We evaluate using these $12\times12\times256$ BEV feature grids.

We use three autoencoders with the same architecture but different inputs/outputs and training objectives so that the bottleneck concentrates on different aspects of scene quality. **Geometry** autoencoder: the input is a sparse occupancy grid and the objective is to reconstruct occupancy. Because dense BEV features must drive the decoder to reconstruct a sparse 3D structure, the bottleneck has to retain shape cues that support recovering surfaces and topology independent of semantic classes. **Semantics** autoencoder: the input is a sparse voxel grid with semantic class labels and the objective is to reconstruct those labels after decoding back to the voxel grid. During decoding, the sparse occupancy layout is provided at each stage, so the model does not need to carry fine geometric detail in the bottleneck and instead captures which classes appear and where. This configuration is common in prior work [1, 10, 11, 17]. **Geometry+Semantics** autoencoder: the input is a sparse voxel grid with semantic class labels, the same as the Semantics autoencoder, but the objective is now to reconstruct *both*, yielding a holistic descriptor that balances geometry and semantics.

To assess fidelity of individual categories, we evaluate per class. For each class c, we assemble two sets of feature vectors taken from the BEV bottleneck: one set extracted from real testing scenes and one set from generated scenes.

A BEV cell contributes a feature to class c if the 3D region covered by that cell includes that class; a cell may contribute to multiple classes when several categories occur in that region. In addition, we form an *All* set by treating everything as a single class, taking feature vectors from every BEV cell, providing an overall scene measure that complements the class-balanced metric.

We compare sets with squared Maximum Mean Discrepancy (MMD, also known as KID) [5, 8, 10, 11, 17]. With kernel k and feature sets $X = \{x_i\}_{i=1}^m$ from real scenes and $Y = \{y_j\}_{j=1}^n$ from generated scenes, the metric is

$$\mathrm{MMD}^2(X,Y) = \frac{1}{m(m-1)} \sum_{i \neq j} k(x_i, x_j) + \frac{1}{n(n-1)} \sum_{i \neq j} k(y_i, y_j) - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(x_i, y_j). \tag{1}$$

using the kernel $k(x,y) = \exp\big(-\|x-y\|^2/(2\sigma^2)\big)$. We use MMD/KID because it is more reliable than FID when feature embeddings are non-Gaussian as is the case for deep features [8]. We report **All** (MMD on the "all voxels" sets), **Per-class** (the mean of class-wise MMDs), and **Avg** $= 0.5\times$ All $+ 0.5 \times$ Per-class, which is the primary metric.

D. Model Implementation Details

In this section, we provide additional model and training details. Our framework consists of the grounded latent VAE model and three diffusion models G_L (layout), G_F (feature), and G_M (motion).

Grounded Latent VAE. The grounded latent VAE [9] takes as input a sparse semantic voxel grid, where each occupied voxel has a semantic class label. In Waymo, the grid is $200 \times 200 \times 16$ with 15 foreground classes, and in CarlaSC, the grid is $128 \times 128 \times 8$ with 10 foreground classes. Occupied voxel centers are encoded with Fourier positional embeddings, and class labels are embedded with a linear layer. These voxel embeddings are processed by a 6-layer transformer [16] (hidden size 384, 6 heads, MLP ratio 4) with 3D sliding window attention [12, 19, 20] and FlashAttention [3] for efficiency.

The embedded sparse voxel features are summarized into a sparse latent point-cloud representation. Foreground and background points are obtained separately. Each foreground instance (e.g., Vehicle, Pedestrian, Motorcycle) is assigned a single latent at its 3D center, which preserves one-to-one identity and enables accurate motion without merging or splitting. For background classes, we perform farthest point sampling per class to ensure even spatial coverage.[1] Foreground points inherit the orientation of their instances. We use 768 latent points on CarlaSC and 1024 on Waymo. Given this point set (positions, classes, orientations), we sample the transformer features at the selected locations to obtain a feature vector for each point, and map it with a linear layer to the mean and variance of the VAE bottleneck.

Table 5. Ablation on the latent feature dimension.

Feature dim	mIoU↑	Geo↓	Sem↓	Geo+Sem↓
32	90.94	9.90	**11.11**	6.96
64	91.86	9.67	**11.11**	**6.99**
128	**92.76**	**9.52**	11.23	7.01

The latent points and their features are passed through a decoder transformer with the same depth and dimensions (6 layers, hidden size 384, 6 heads, MLP ratio 4). Final point features are mapped by a linear layer to semantic Gaussian parameters [6, 7]. Each point predicts 32 Gaussians, each with an offset from the point, scale, rotation, and opacity. Each Gaussian inherits the class of its parent latent, and foreground Gaussians are rotated by the latent orientation as described in the main paper. We splat these semantic Gaussians to the voxel grid and supervise with cross-entropy and Lovasz losses (weights 1.0 and 1.0) together with the KL loss term (weight 0.0015). The model is trained for 20 epochs with a cosine learning rate schedule, global batch size 8, and learning rate 1×10^{-4}.

Layout Diffusion. As outlined in the main paper, G_L generates the grounded latent layout: 3D positions (x, y, z), foreground orientations $(\sin\theta, \cos\theta)$, and class bits. During diffusion, these noised attributes are embedded with a linear layer and passed through 12 DiT blocks (hidden size 384, 6 heads, MLP ratio 4) with AdaLN timestep conditioning [13]. We use the ϵ-prediction objective with 1000 denoising steps, global batch size 256, and learning rate 1×10^{-4}. Following DynamicCity [1], all diffusion models are trained for 1200 epochs with a constant learning rate schedule and exponential moving average. On Waymo, we temporally subsample the data to 2 Hz due to computational constraints.

Feature Diffusion. Given the generated layout from G_L, G_F produces a feature vector for each latent summarizing local geometry. The noisy feature tokens are added to the embedded layout attributes and passed through a diffusion transformer identical to G_L. Training and denoising settings are also the same as G_L (ϵ-prediction, 1000 steps, same optimizer and schedule). After feature generation, the grounded latents are decoded by the VAE decoder D to semantic Gaussians, which we splat to voxels for 3D scene

[1] In practice, to avoid running farthest-point sampling sequentially per background class (which would require preset per-class budgets), we concatenate to each voxel's (x, y, z) a class-indicator vector of length C_{bg} whose entry for the voxel's class is set to a large value and others to zero. Distances between different classes become effectively infinite, so a single sampling pass behaves the same as class-wise FPS.

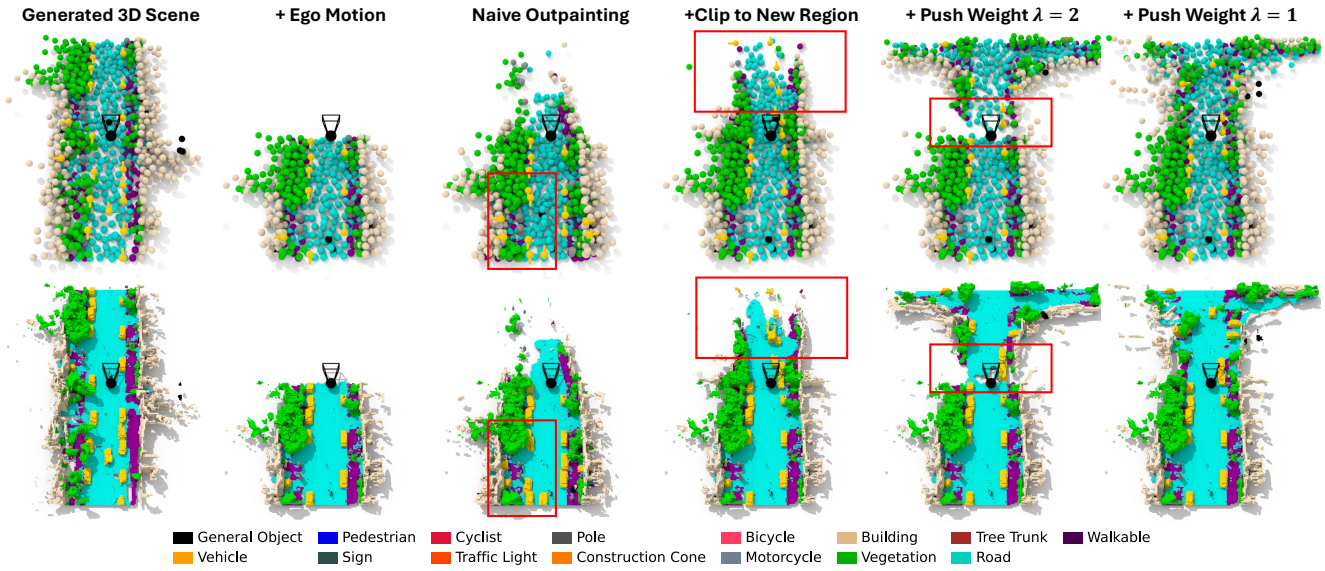

Figure 7. **Qualitative ablation on push weight for outpainting.** From left to right: initial generation, forward shift (new region lacks generation), naive outpainting without clipping or push, clipping only, clipping with λ=2.0 (over-pushed, boundary gap), and clipping with λ=1.0 (even coverage and stable prior content).

generation.

Motion Diffusion. To lift 3D to 4D, G_M predicts future motion for all dynamic foreground actors (including the ego agent), outputting 20 future steps at 10 Hz via waypoints and orientations, conditioned on a 10-step history ($\approx$1 s) with a 10% history-drop rate for unconditional generation. At a future timestep t, each agent token is initialized as the sum of its embedded grounded latent, the embedded future timestep, and the initial noised waypoint and orientation. The architecture mirrors the DiT above, with two differences. First, DiT conditioning attends to the full set of grounded latents at the current timestep (foreground and background) for scene context. Second, AdaLN is applied to agent tokens using a conditioning vector derived from both the denoising timestep embedding as well as the agent's embedded 10-step history. After denoising, the 3D latent layout is moved according to the generated motion while latent features remain unchanged to preserve local geometry. We decode the moved latents at each timestep with the VAE decoder D to obtain the generated 4D scene.

E. Additional Ablations and Analysis

Ablation on latent feature dimension. In addition to the ablation on the number of latents in Table 3 of the main paper, we ablate the per-latent feature dimension in Table 5. This ablation uses 512 latents in CarlaSC [4, 18]. Overall, performance is notably less sensitive to the feature dimension than to the number of grounded latents. As the feature dimension increases, reconstruction mIoU and geometric generation quality improve. By contrast, the semantics-only metric remains largely stable, since coarse class layout is affected mainly by G_L (layout diffusion) and only weakly by G_F (feature diffusion) which emphasizes local geometry. Based on the holistic geometry+semantics metric, we select a feature dimension of 64.

Qualitative ablation on push weight for outpainting. To supplement the quantitative ablation on the push weight for outpainting, we present a qualitative analysis of why clipping and a push term during the diffusion process are necessary in Figure 7. In the first column, we visualize the initially generated 3D scene. In the second, we move the scene forward, revealing a newly uncovered forward region that is missing 3D generation. In the third, we try a naive outpainting strategy where latents in the earlier half are held fixed and only new latents are denoised. Unlike grid-aligned generators, which can simply shift the window, freeze the earlier cells, and denoise only the incoming half, our grounded latents can move freely. As such, first, new latents drift back and add to previously generated regions, reducing temporal consistency, and second, the newly exposed area remains under-covered.

To address the first issue, we clip new latents to the forward region during the diffusion process as shown in the fourth column. This preserves the previously generated half as intended, but new latents cluster near the boundary and still fail to fill the forward area. We therefore introduce a push gradient during diffusion, controlled by a push weight λ, which encourages new latents to occupy the forward region while leaving the rear unchanged. With $\lambda = 2.0$ (fifth

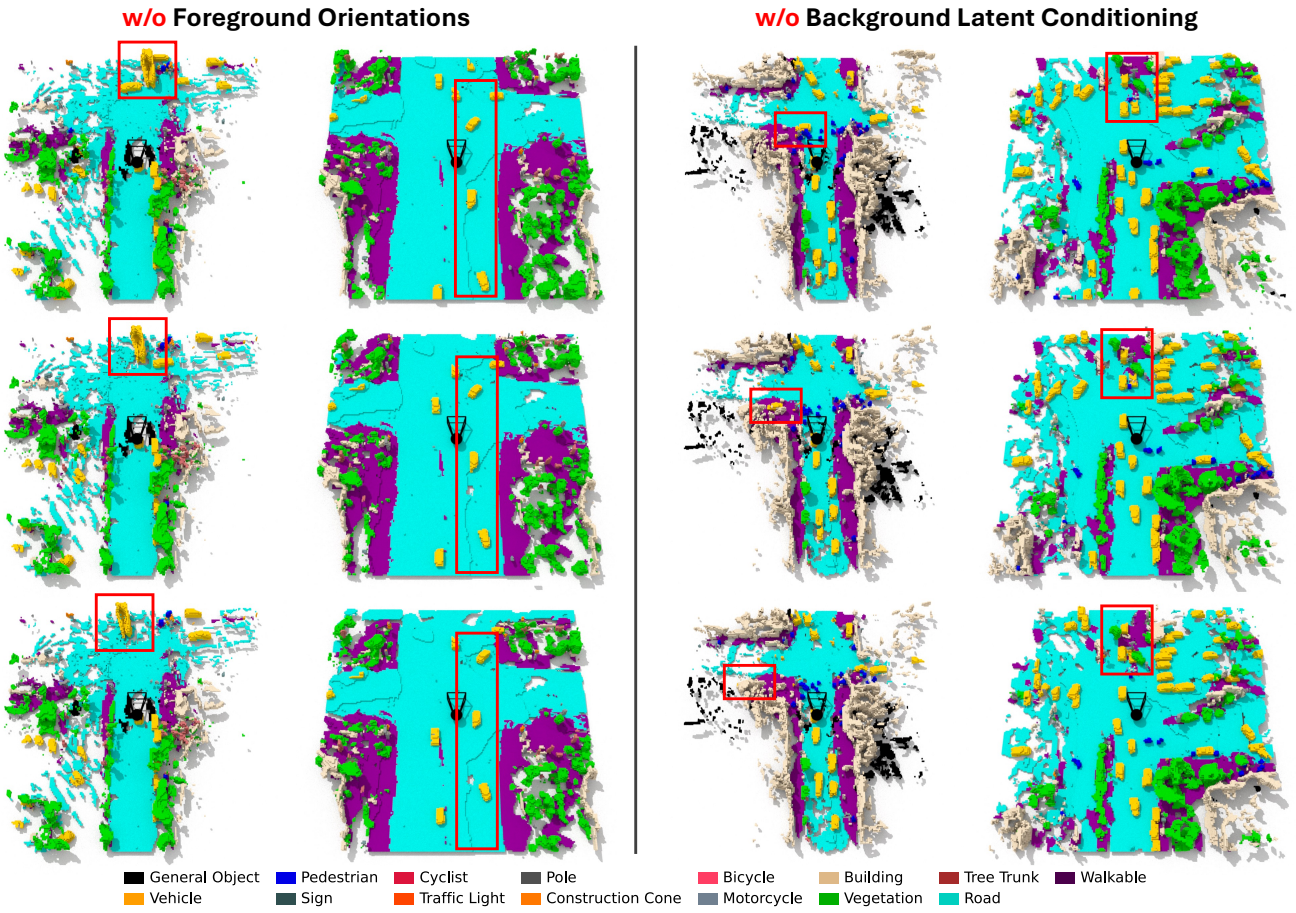

Figure 8. **Conditioning signals for motion.** Two rollouts per setting. *Left:* no explicit orientation in foreground latents leads to incorrect headings. *Right:* conditioning G_M only on foreground causes trajectories to leave drivable regions.

column), coverage improves but a visible gap appears at the boundary because the push is too strong. With $\lambda = 1.0$ (final column), coverage is even and the scene remains visually homogeneous across the transition without disturbing earlier content.

Conditioning signals for motion generation. We analyze two factors that influence motion generation: explicit orientation in the foreground latents and whether G_M is conditioned on all latents (foreground and background) or only foreground. We present two example rollouts for each setting in Figure 8. Without explicit orientation in the foreground latents we find that trajectories have unrealistic headings. Vehicles drift or rotate incorrectly, failing to align their yaw properly as they move. When conditioning G_M only on foreground entities, without context from background structure (road, sidewalk, vegetation, buildings), trajectories frequently stray off drivable surfaces or cut through non-drivable regions. These examples show that orientation in the foreground latent representation and scene-wide context are both necessary for coherent 4D motion.

F. Additional Results

F.1. 4D Occupancy Forecasting and Motion Prediction

To further evaluate the quality and temporal consistency of our generated scenes, we evaluate 4D occupancy forecasting on the Waymo validation set. Given 1 second of history, we predict future semantic occupancy at multiple horizons.

Table 6 reports per-class IoU↑ at 0s, 1s, and 2s into the future. Even at 2s, our model achieves **60.1 mIoU**, with strong foreground accuracy (Veh. 60.7 IoU, Ped. 50.6 IoU), showing that our grounded latent representation produces accurate and temporally consistent rollouts rather than just plausible-looking generations. These results validate that the motion diffusion model generates realistic trajectories consistent with the generated 3D scene.

We also evaluate actor trajectory quality under the same

Table 6. **Waymo occupancy forecasting.** Future semantic occupancy mIoU↑ (All) and per-class IoU↑ at different horizons.

t	mIoU	IoU	Gen.	Vehicle	Ped.	Sign	Cyclist	Traff.	Pole	Constr.	Bicycle	Motor.	Bldg.	Veg.	Tree	Road	Walk.
0s	96.8	84.8	99.2	80.0	98.7	99.4	97.9	99.1	99.1	98.9	96.9	95.3	98.7	98.9	97.4	94.8	98.3
1s	73.1	48.3	91.0	70.5	68.8	66.0	56.8	77.1	69.1	51.3	70.5	53.3	90.0	92.9	68.1	88.3	83.0
2s	60.1	34.1	83.7	60.7	50.6	51.2	34.4	72.6	52.4	32.8	54.3	33.6	82.8	86.9	50.8	83.1	72.0

setting, obtaining **ADE 1.03m** and **FDE 2.32m** for vehicle motion prediction, confirming that our grounded motion diffusion model produces accurate future trajectories in addition to plausible occupancy.

F.2. Foreground Class Diversity

Vehicle geometry diversity. The per-latent features capture subtle but crucial local geometry. Figure 9 shows length, width, and height histograms for generated vs. GT vehicles on Waymo. The distributions closely match across all three axes, spanning both small cars and large trucks. This matters in driving because modeling these scale differences is critical for planning in tight scenes. This is also consistent with the main paper Figure 4, where two generations under the same layout latents still vary in vehicle size and background details such as foliage.

Pedestrians and other foreground classes. Our framework generates and tracks all dynamic foreground classes, including vehicles, pedestrians, and cyclists. Figure 10 shows a crowded street scene where individual pedestrians (blue) are temporally consistent across frames, avoiding the merging and splitting artifacts that arise in dense voxel methods. This is a direct consequence of our entity-centric design: each pedestrian is assigned a single grounded latent, so its identity is preserved throughout the rollout.

G. Qualitative Results on CarlaSC

In Figure 11 we visualize sampled 3D scenes generated by our method trained on the CarlaSC dataset. Our framework is able to accurately capture the real distribution of scenes, simulating complex traffic interaction scenarios with various foreground actors.

H. Qualitative Results on Waymo Reconstruction

In Figure 12, we visualize the underlying latent layout, the VAE reconstruction, and the ground-truth voxels on the held-out validation set. We do this to test if our VAE

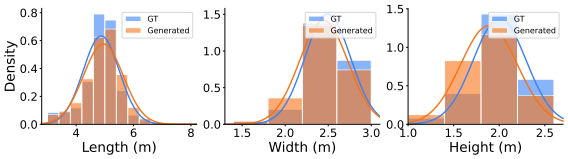

Figure 9. Vehicle sizes in Waymo GT vs our generations.

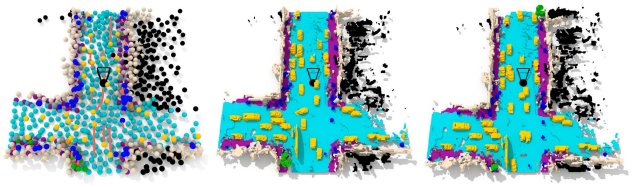
Figure 10. Our framework precisely models pedestrians (blue).

can generalize and to verify that our grounded latent representation captures fine detail even though it is not grid-aligned. We observe that the VAE reconstructs challenging scenes with complex background structure and crowded foreground actors, indicating that the latent representation preserves both detailed background regions and per-actor attributes, which in turn enables high-fidelity generations.

We also note that in many cases the ground truth is suboptimal. Ideally, the ground-truth voxels would cover the full extent of the scene, but real-world LiDAR sparsity and occlusion can leave portions incomplete (e.g., road or vehicle regions as in row 4). Our generative model inherits such artifacts by matching the data distribution. A similar concern exists in the CarlaSC dataset, which simulates LiDAR before voxelizing the scene. In future work, we plan to mitigate this by rendering synthetic data without LiDAR as an intermediary, then training jointly on real and synthetic data.

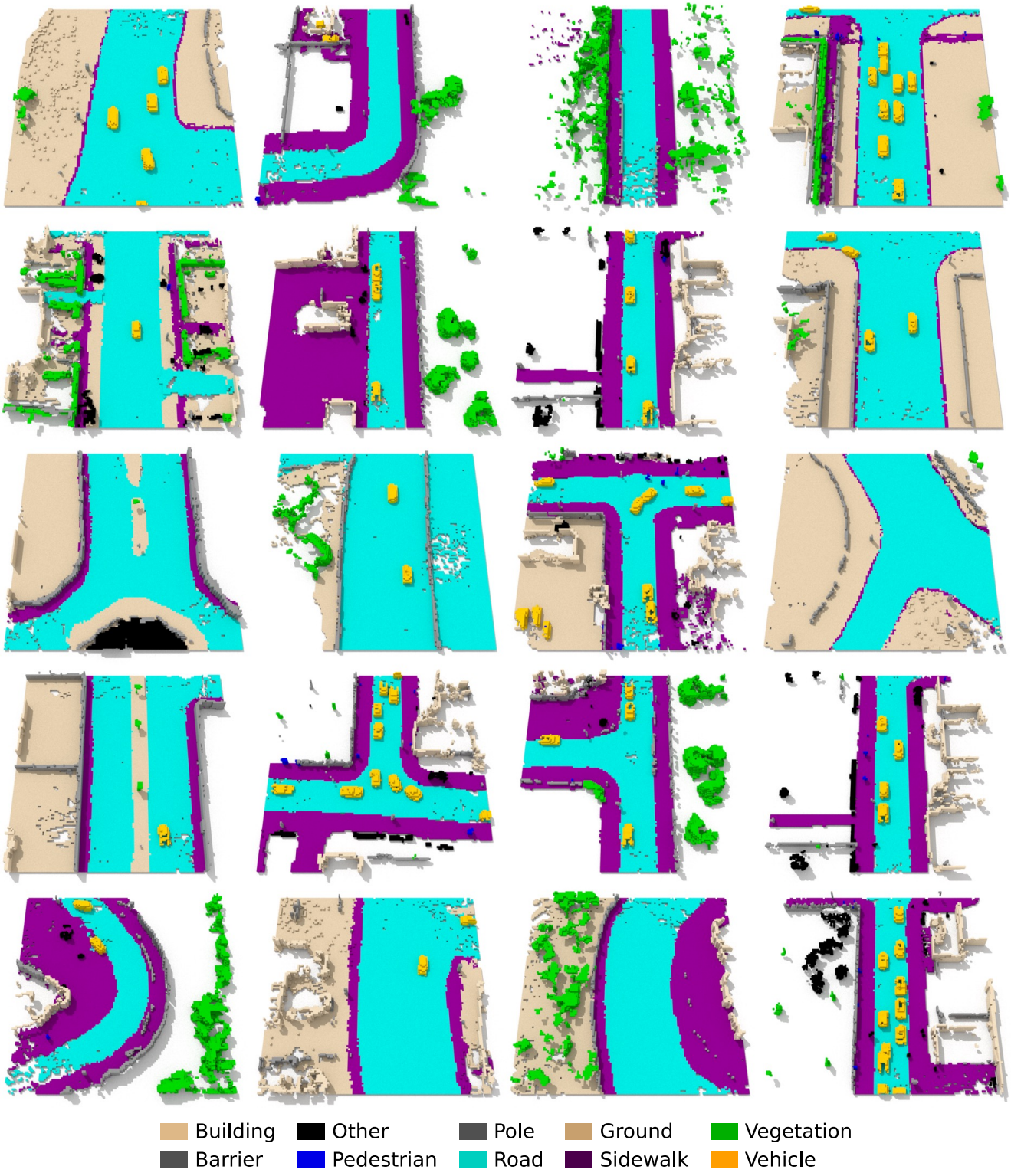

Figure 11. Example 3D scene generations of our framework on the CarlaSC dataset.

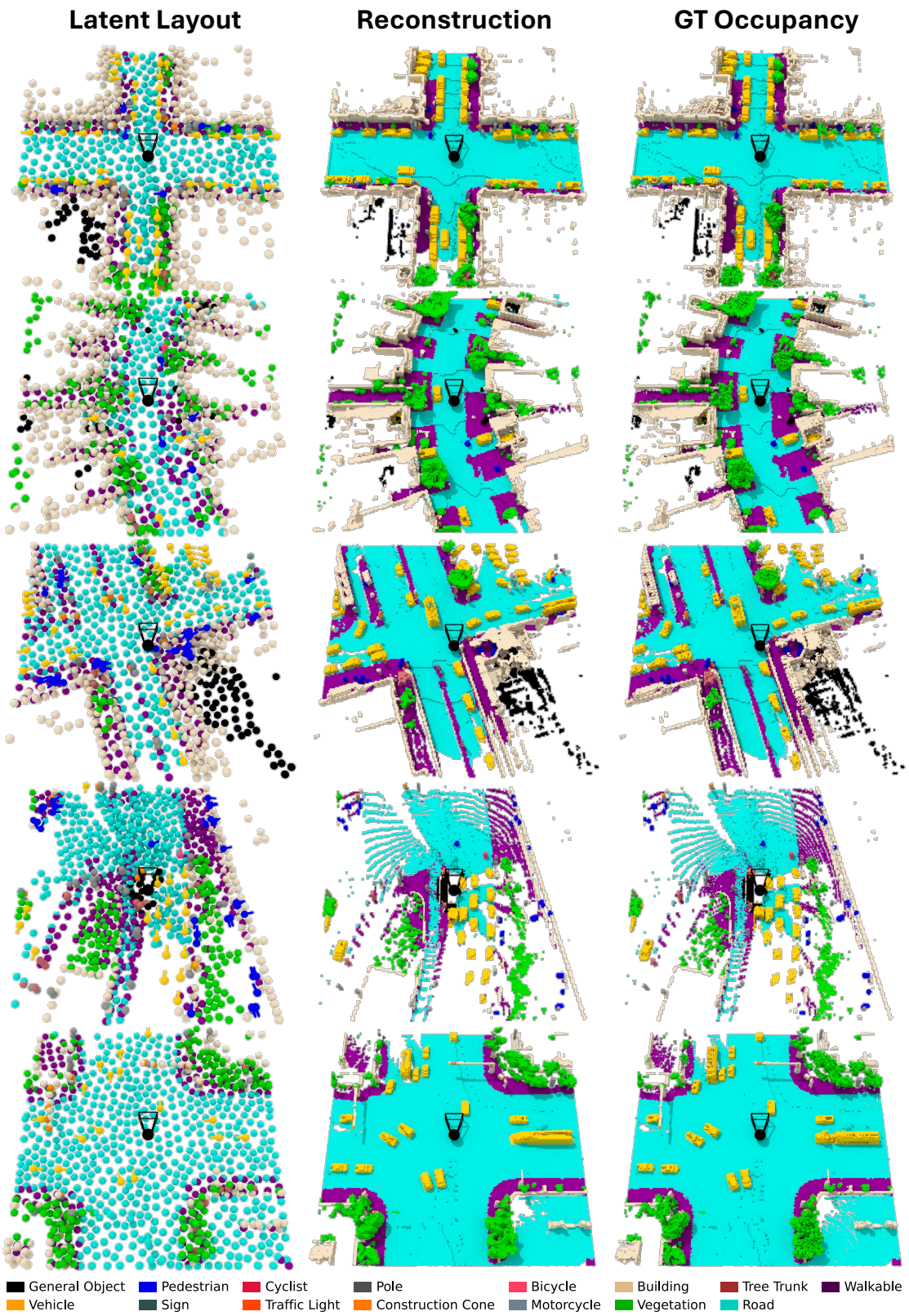

Figure 12. Reconstruction on validation scenes: latent layout, VAE reconstruction, and ground truth. Our latent VAE reconstructs the entire scene with high detail.

References

[1] Hengwei Bian, Lingdong Kong, Haozhe Xie, Liang Pan, Yu Qiao, and Ziwei Liu. Dynamiccity: Large-scale 4d occupancy generation from dynamic scenes. In *Proc. Int. Conf. Learn. Represent.*, 2025. 1, 2

[2] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 1

[3] Tri Dao, Daniel Y Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *arXiv preprint arXiv:2205.14135*, 2022. 2

[4] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conf. Robot Learn.*, pages 1–16, Mountain View, CA, USA, 2017. PMLR. 1, 3

[5] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proc. IEEE Int. Conf. Comput. Vis.*, 2021. 2

[6] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Gaussianformer: Scene as gaussians for vision-based 3d semantic occupancy prediction. *arXiv preprint arXiv:2405.17429*, 2024. 2

[7] Yuanhui Huang, Amonnut Thammatadatrakoon, Wenzhao Zheng, Yunpeng Zhang, Dalong Du, and Jiwen Lu. Probabilistic gaussian superposition for efficient 3d occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 2

[8] Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9307–9315, 2024. 2

[9] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[10] Jumin Lee, Sebin Lee, Changho Jo, Woobin Im, Juhyeong Seon, and Sung-Eui Yoon. Semcity: Semantic scene generation with triplane diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024. 1, 2

[11] Yuheng Liu, Xinke Li, Xueting Li, Lu Qi, Chongshou Li, and Ming-Hsuan Yang. Pyramid diffusion for fine 3D large scene generation. *arXiv preprint arXiv:2311.12085*, 2023. 1, 2

[12] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2

[13] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 4195–4205, 2023. 2

[14] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 1

[15] Xiaoyu Tian, Tao Jiang, Longfei Yun, Yucheng Mao, Huitong Yang, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. *Advances in Neural Information Processing Systems*, 36:64318–64330, 2023. 1

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Adv. Neural Inf. Process. Syst.*, 2017. 2

[17] Lening Wang, Wenzhao Zheng, Yilong Ren, Han Jiang, Zhiyong Cui, Haiyang Yu, and Jiwen Lu. Occsora: 4d occupancy generation models as world simulators for autonomous driving. *arXiv preprint arXiv:2405.20337*, 2024. 1, 2

[18] Joey Wilson, Jingyu Song, Yuewei Fu, Arthur Zhang, Andrew Capodieci, Paramsothy Jayakumar, Kira Barton, and Maani Ghaffari. MotionSC: Data set and network for real-time semantic mapping in dynamic environments. *IEEE Robotics and Automation Letters*, 7(3):8439–8446, 2022. 1, 3

[19] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21469–21480, 2025. 2

[20] Yu-Qi Yang, Yu-Xiao Guo, Jian-Yu Xiong, Yang Liu, Hao Pan, Peng-Shuai Wang, Xin Tong, and Baining Guo. Swin3d: A pretrained transformer backbone for 3d indoor scene understanding, 2023. 2