Gaussian-based World Model: Gaussian Priors for Voxel-Based Occupancy Prediction and Future Motion Prediction
Supplemental Material

Tuo Feng, Wenguan Wang*, Yi Yang
ReLER, CCAI, Zhejiang University
https://github.com/FengZicai/GWM

To improve the clarity and organization of the main paper, this supplementary material is organized as follows. §A provides a review of additional related works on 3D occupancy prediction and 3D Gaussian splatting. In §B, we provide detailed implementation information and describe the training losses used in our framework. §C offers additional qualitative and quantitative results. Lastly, §D discusses the limitations of our approach and its societal implications, highlighting areas for future research, ethical considerations, and the potential societal impact of deploying autonomous driving systems.

A. Additional Related Works

3D Occupancy Prediction. The safety and efficiency of autonomous driving systems critically hinge on precise 3D occupancy prediction to ensure safe navigation. 3D Occupancy Prediction focuses on determining whether each voxel in a 3D space is occupied and assigning it a semantic label [11, 23, 24, 26, 28, 37]. LiDAR-based methods [7, 15, 22, 30] excel in capturing spatial depth information but lack rich semantic details, while vision-based methods [4, 11, 36] provide abundant semantic information yet struggle to accurately handle dynamic scenes. While most methods focus on static 3D occupancy, they often overlook temporal dynamics crucial for autonomous driving [11, 28, 37]. Recent advances include pipelines that generate dense occupancy labels [27], and models that leverage 2D data to reduce dependence on costly 3D annotations [19]. Challenges such as computational inefficiency in grid-based methods and information loss in bird's-eye-view (BEV) based perception persist [10, 14, 24], with newer object-centric approaches like GaussianFormer [12] aiming to address these issues by optimizing resource allocation and adapting to object scale and complexity [37].

Due to reasons of expressiveness, efficiency, and versatility, occupancy-based world models are superior to those based on bounding boxes and segmentation maps [34]. Compared to voxel representations, Gaussian representations offer advantages such as adaptive allocation of computational and storage resources, preservation of details, and explicit semantic meaning [12]. GWM integrates Gaussian representations into voxel-based scene representations, preserving these advantages. It not only accurately captures spatial details but also, through spatiotemporal modeling, effectively handles temporal evolution in motion prediction.

3D Gaussian Splatting. 3D Gaussian Splatting (3DGS) [13] is a recent technique that uses multiple 3D Gaussians for radiance field rendering, achieving superior performance in both rendering quality and speed [6]. Unlike traditional explicit scene representations such as meshes [20, 21, 31] and voxels [8, 18], 3DGS models complex shapes with fewer parameters. Additionally, it enables fast rendering through splat-based rasterization, projecting 3D Gaussians onto 2D views and rendering image patches with local 2D Gaussians. When comparing with neural radiance fields (NeRF) [1, 17], 3DGS has lower computational complexity, enabling faster rendering while maintaining accuracy. This is particularly important in complex environments with multi-sensor fusion. Recent advancements adapt 3DGS to dynamic scenes [16, 33], with techniques like deformation networks to model Gaussian motion [33] and HexPlane-based rendering for adjacent Gaussians [29]. These methods primarily target monocular or multi-camera scenes, but challenges remain in handling real-world autonomous driving scenarios due to complex backgrounds, where high-speed movement, sparse views, and dynamic objects with occlusion are present [5, 35]. While 3DGS is optimized for static scene rendering, dynamic applications in autonomous driving require further adaptations, particularly to handle fast-changing environments and sparse view inputs. GWM incorporates LiDAR points as 3D Gaussian priors combined with spatiotemporal dynamic modeling, effectively addressing spatial changes and occlusions in dynamic environments. This approach enhances the accuracy of 3D semantic occupancy prediction.

*Corresponding author.

B. Additional Details

B.1. Implementation Details

As shown in Fig. 2 of the main paper, several key modules contribute to our GWM, each playing a distinct role in processing input data and generating future trajectories. Below, we present the implementation details of these modules: 2D Encoder, 3D Encoder, *Gaussian Representation Learner* ϕ, *Uncertainty-Aware Simulator* ψ, and *Uncertainty-Aware Planner* θ.

2D Encoder. 2D Encoder is responsible for extracting visual features from input images of the environment. The implementation uses a ResNet101 network as the backbone.

3D Encoder. 3D Encoder is used for processing LiDAR points and handling interactions between sparse 3D Gaussians. Each Gaussian is treated as a point cloud located at its mean value. These points are projected onto target voxel grids. 3D sparse convolutions are then applied within the voxel grid using spconv[1]. This effectively handles interactions between sparse 3D Gaussians.

Gaussian Representation Learner ϕ. ϕ is responsible for learning Gaussian-based semantic occupancy representations from 2D and 3D data. The *learner* operates as follows: Image Cross-Attention [12]: a set of reference points is generated based on Gaussian covariance and mean, which is then projected onto image features, with deformable attention used for aggregating image features; Gaussian Refinement [12, 13] decodes intermediate attributes from the Gaussian queries and uses them to refine the existing Gaussian attributes. The reconstruction head and semantic head are based on the Differentiable Tile Rasterizer [13].

Uncertainty-Aware Simulator ψ. Similar to [34], ψ predicts future scene representations based on temporal context using a transformer-like architecture: 1) Scene Tokenizer: to obtain discrete tokens, we employ a vector-quantized autoencoder (VQ-VAE) [25]; 2) Spatial Aggregation: tokens from different times are aggregated spatially using a multi-scale representation to represent the scene at different granularities; 3) Masked Temporal Attention: temporal attention is applied to the tokens, with causal masking ensuring that only past and current tokens influence predictions for the future; 4) Decoder: the decoder generates future occupancy maps from the tokens, reconstructing future 3D scenes.

Uncertainty-Aware Planner θ. θ refines and generates trajectories for autonomous navigation: 1) Trajectory Sampling [9]: a sampler generates a set of candidate trajectories τ_*^{T+1} based on high-level commands, such as turning instructions; 2) Trajectory Selection: the cost volume is used to evaluate each candidate, and the trajectory with the lowest cost (τ^{T+1}) is selected, ensuring safety and efficiency; 3) Trajectory Refinement [32]: the selected trajectory is encoded as an ego query and performs cross-attention with the future occupancy probability ($\boldsymbol{h}_{\text{bev}}^{T+1}$), drawing detailed information from the environment. Finally, the enhanced ego query is used to predict the final refined trajectory. The *planner* integrates high-level environmental information and dynamic context to output a safe and reliable trajectory for the ego vehicle.

[1] https://github.com/traveller59/spconv

B.2. Training Losses

Our GWM is trained in two stages to effectively learn the Gaussian representations and accurately model the joint evolution of the ego vehicle and its environment.

First Stage: Learning Gaussian Representations. In the first stage, we focus on training the 2D/3D encoders and the *Gaussian representation learner* ϕ to construct the scene representations. The total loss function for this stage is:

$$\mathcal{J}_\phi = \lambda_1 \mathcal{L}_{\text{sem}} + \lambda_2 \mathcal{L}_{\text{recon}}. \tag{1}$$

Semantic loss $\mathcal{L}_{\text{sem}}$ calculates the discrepancy between projected semantic images and raw images:

$$\mathcal{L}_{\text{sem}} = \frac{\beta}{2}\big(1 - \text{SSIM}(\boldsymbol{S}, \hat{\boldsymbol{S}})\big) + (1-\beta)\|\boldsymbol{S} - \hat{\boldsymbol{S}}\|_1, \tag{2}$$

where $\boldsymbol{S}$ and $\hat{\boldsymbol{S}}$ represent the projected and ground-truth semantic segmentations, respectively; β balances the Structural Similarity Index Measure (SSIM) loss and the L1 reconstruction loss. Reconstruction loss ($\mathcal{L}_{\text{recon}}$) enforces consistency between projected images $\hat{\boldsymbol{I}}$ from the Gaussian representation and raw input images $\boldsymbol{I}$:

$$\mathcal{L}_{\text{recon}} = \frac{\beta}{2}\big(1 - \text{SSIM}(\boldsymbol{I}, \hat{\boldsymbol{I}})\big) + (1-\beta)\|\boldsymbol{I} - \hat{\boldsymbol{I}}\|_1. \tag{3}$$

In the main text, we use $\boldsymbol{I}$ as a shorthand to denote both the semantic maps and the raw input images for brevity; however, in the present context these two uses should be distinguished.

Second Stage: Learning, Forecasting, and Planning. In the second stage, we train the *simulator* ψ and the *planner* θ to predict future occupancy states and generate safe and efficient trajectories. The loss function is:

$$\mathcal{J}_{\psi,\theta} = \mathcal{L}_{\text{fcst}} + \mathcal{L}_{\text{plan}}, \tag{4}$$

where forecasting loss ($\mathcal{L}_{\text{fcst}}$) supervises the prediction of future occupancy states. It averages the occupancy loss (*i.e.*, a combination of the cross-entropy loss $\mathcal{L}_{\text{ce}}$ and the Lovász-Softmax loss $\mathcal{L}_{\text{lov}}$ [2]) across N future frames:

$$\begin{aligned}\mathcal{L}_{\text{fcst}} = \frac{1}{N}\sum\nolimits_{f=1}^{N} \big(&\mathcal{L}_{\text{ce}}(\boldsymbol{o}^{T+f}, \hat{\boldsymbol{o}}^{T+f}) + \mathcal{L}_{\text{lov}}(\boldsymbol{o}^{T+f}, \hat{\boldsymbol{o}}^{T+f}) \\ &+\mathcal{L}_{\text{bce}}(\boldsymbol{o}_b^{T+f}, \hat{\boldsymbol{o}}_b^{T+f})\big),\end{aligned} \tag{5}$$

where $\boldsymbol{o}^{T+f}$, $\boldsymbol{o}_b^{T+f}$, $\hat{\boldsymbol{o}}^{T+f}$, and $\hat{\boldsymbol{o}}_b^{T+f}$ represent predicted and ground-truth semantic and binary occupancies for future frame f, respectively. Planning loss ($\mathcal{L}_{\text{plan}}$) ensures the *planner* generates safe, efficient, and expert-like trajectories. It

Table S1. **Ablation study** on *simulator* (§C.1).

Method	Forecasting		Planning	
	mIoU(%)↑	IoU(%)↑	L2 (m)↓	Collision (%)↓
GWM (*w/o* TA)	8.03	18.99	1.27	0.71
GWM (*w/* LSTM)	9.14	22.95	1.22	0.69
GWM (Full Model)	**10.12**	**24.60**	**1.13**	**0.59**

Table S2. **Ablation study** on *planner* (§C.2). We report L2 error (m) and collision rate (%) at 1s, 2s, 3s, and their average (Avg.). TR denotes Trajectory Refinement, UL denotes Uncertainty Loss.

Method	L2(m) ↓				Collision(%) ↓			
	1s	2s	3s	Avg.	1s	2s	3s	Avg.
GWM (*w/* ego decoder [34])	0.50	1.36	2.49	1.45	0.13	0.48	1.64	0.75
GWM (*w/o* TR)	0.45	1.27	2.42	1.38	0.10	0.41	1.59	0.70
GWM (*w/o* UL)	0.42	1.20	2.13	1.25	0.08	0.38	1.55	0.67
GWM (Full Model)	**0.34**	**1.01**	**2.05**	**1.13**	**0.07**	**0.26**	**1.45**	**0.59**

includes: 1) Max-Margin loss: penalizes low-cost trajectories τ^{T+1} that deviate from expert trajectory; 2) Imitation learning loss: a naive $\mathcal{L}_2$ loss for imitation learning; 3) Collision loss: penalizes trajectories that intersect with obstacles; 4) Uncertainty Loss: it is discussed in the main paper.

$$\mathcal{L}_{\text{plan}} = \mathcal{L}_{\text{mm}} + \mathcal{L}_2 + \mathcal{L}_{\text{coll}} + \mathcal{L}_{\text{unct}}, \tag{6}$$

where $\mathcal{L}_{\text{mm}}$, $\mathcal{L}_2$, $\mathcal{L}_{\text{coll}}$, and $\mathcal{L}_{\text{unct}}$ denote the max-margin, imitation, collision losses, and uncertainty loss, respectively. By combining these losses, GWM effectively learns to represent the environment using Gaussian-based semantic and geometric features, anticipate future dynamics for safe and precise decision-making, and plan trajectories that avoid obstacles while following expert-like behavior. This comprehensive training regime ensures the robustness and efficiency of GWM in dynamic and complex driving scenarios.

C. Additional Experiments

C.1. Simulator

We assess the impact of the Spatiotemporal Transformer [34] in *uncertainty-aware simulator*. In Tab. S1, we compare our full GWM model with two variants: one where the Transformer is replaced with an LSTM network (*w/* LSTM), and another where temporal modeling is removed entirely (*w/o* TA). The model with LSTM achieves 9.14% Avg. mIoU and 22.95% Avg. IoU in forecasting, and an average L2 error of 1.22m in motion planning. Without TA, the performance further drops to 8.03% mIoU and 18.99% IoU, and an average L2 error of 1.27m. Our full GWM model with the Spatiotemporal Transformer achieves the best results, confirming that the Transformer is essential for capturing temporal dependencies and accurately predicting future scene evolutions.

C.2. Planner

In our ablation study on the *planner*, we evaluate the individual contributions of Trajectory Refinement [32] and Uncertainty Loss. The Trajectory Refinement module leverages the future occupancy probability by encoding the selected trajectory as an ego query, which then performs cross-attention with the BEV representation of the future occupancy. This mechanism enables the *planner* to extract detailed environmental context, facilitating fine-grained trajectory adjustments that enhance both navigation safety and efficiency. As demonstrated in Tab. S2, the baseline OccWorld planner (with an ego decoder) achieves an average L2 error of 1.45m and a collision rate of 0.75%. GWM (*w/o* TR) achieves performance to an average L2 error of 1.38m and a collision rate of 0.70%, highlighting its role in refining trajectory quality.

On the other hand, the Uncertainty Loss is designed to reduce the discrepancy between predicted and expert trajectories by computing a reconstruction nonconformity score and integrating a conformal prediction-based quantile term into the training objective. This approach enforces the model to generate more calibrated and reliable uncertainty estimates. The impact of GWM (*w/o* UL) is evident from our experiments: without Uncertainty Loss, the *planner* registers an average L2 error of 1.25m and a collision rate of 0.67%, while the full GWM planner that incorporates both modules achieves the best performance – with an average L2 error of 1.13m and a collision rate of 0.59%. These results confirm that both trajectory refinement and uncertainty loss are essential for generating precise and safe trajectories under varying environmental conditions.

C.3. Failure Case Study: Truck and Bus Confusion

In this subsection, we present an analysis of a specific failure case observed in our GWM. This issue arises primarily in dense urban environments with high occlusion, limited sensor views, and similar object characteristics. As shown in Fig. 3 of the main paper, the bottom row (GWM at t = 1.5 and 2.0s) illustrates a common failure where the model incorrectly classifies a bus as a truck. This misclassification occurs because trucks and buses share similar physical dimensions (*i.e.*, similar shapes and sizes), and occlusions often obscure critical features, such as the vehicle's front or rear design, making differentiation challenging. The Gaussian-based occupancy representation tends to generalize these two categories under low visibility conditions, especially when color and texture information are absent. To address this failure case, we suggest the following potential solutions: 1) Enhanced Feature Extraction: improving the feature extraction capabilities of the 2D and 3D encoders could provide more distinctive information, aiding in better classification; 2) Temporal Information Integration: using temporal data from past frames could help distinguish trucks and buses, leveraging motion and shape changes over time. We will explore these approaches in future work to address this failure case.

Table S3. **Component-wise performance and efficiency** of GWM (§C.5). Measurements are conducted on an NVIDIA V100 GPU.

Component	Stage	Training	Inference	Metric	Value
2D Encoder	*1st stage*	127.34 ms	29.87 ms	Feature Extraction	-
3D Encoder	*1st stage*	133.21 ms	36.12 ms	Spatial Representation	-
ϕ	*1st stage*	299.12 ms	89.56 ms	mIoU	27.53%
ϕ & ψ	*2nd stage*	1380.96 ms	501.72 ms	Avg. mIoU	10.12%
θ	*2nd stage*	395.04 ms	204.92 ms	Avg. L2 Error	1.13 m

C.4. Qualitative Results

Fig. S1 provides a visual comparison of the occupancy predictions. GWM demonstrates its capability in accurately predicting the occupancy of moving objects.

C.5. Component-wise Performance and Efficiency

In Tab. S3, we measure the performance, training speed, and inference speed of each component in both stages.

1st Stage: 2D/3D Encoders and Gaussian Representation Learner ϕ. The integration of 2D and 3D data enables comprehensive scene understanding. The Gaussian representation enhances the model's ability to handle uncertainties and improves occupancy prediction accuracy. Here, we evaluate the training and inference times of the 2D and 3D encoders and *Gaussian learner* ϕ, in terms of the milliseconds spent per frame. On an NVIDIA V100 GPU, the training time per frame is approximately 127.34ms for the 2D encoder, 133.21ms for the 3D encoder, and 299.12ms for the *Gaussian representation learner*. During inference, the 2D encoder runs at 29.87ms per frame, the 3D encoder at 36.12ms per frame, and the *Gaussian representation learner* at 89.56ms per frame. The *Gaussian learner* ϕ contributes significantly to the initial occupancy prediction accuracy (mIoU of 27.53%).

2nd Stage: Simulator ψ and Planner θ. The *simulator* ψ is a spatiotemporal Transformer that predicts future scene evolutions based on past observations. The *planner* θ generates safe and efficient trajectories by evaluating potential paths against predicted occupancy maps. The spatiotemporal Transformer effectively captures temporal dependencies, leading to accurate future occupancy predictions. The *planner* minimizes collision risks, enhancing navigation safety. Training ψ and θ involves learning complex temporal dynamics and planning strategies. The *simulator* ψ requires more computational resources due to the complexity of modeling temporal dynamics with the spatiotemporal Transformer. The *learner* and *simulator* ψ have a training time of approximately 1380.96ms per frame, while the *planner* θ requires about 395.04ms per frame on an NVIDIA V100 GPU. During inference, ϕ and ψ operates at 501.72ms per frame, and θ at 204.92ms per frame. The *planner* θ is efficient and enhances trajectory planning with a low average L2 error of 1.13m.

D. Limitation and Societal Impact

Limitation. 1) Although the effectiveness of GWM has been validated on the nuScenes [3] dataset, further investigation is needed to evaluate its generalization across a broader range of driving conditions, such as varying weather conditions, diverse traffic densities, and geographically distinct locations. To address this, we plan to conduct additional experiments on the Occ3D-Waymo dataset [23], which will provide deeper insights into the model's robustness and adaptability. 2) A failure case has been identified in our study: the truck and bus confusion. To address this failure, we propose potential solutions such as enhanced feature extraction and temporal information integration. We will explore these approaches in future work to mitigate this failure case.

Societal Impact. Our work aims to improve the safety and efficiency of autonomous driving systems by enhancing the accuracy of environmental perception and motion prediction. By providing more robust and accurate modeling of dynamic environments, GWM has the potential to reduce traffic accidents and improve transportation efficiency. However, the deployment of such advanced autonomous driving technologies raises societal concerns, including the potential displacement of jobs in the driving sector, ethical considerations in decision-making algorithms, and privacy issues related to the collection and processing of sensor data. It is important for stakeholders to address these concerns by developing appropriate regulations and ensuring transparency in the deployment of autonomous systems.

License. We conduct our experiments using publicly available datasets, such as nuScenes [3], which is provided under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0)[2]. Any use of the dataset must comply with the terms specified by the dataset providers.

Computing Infrastructure. Our experiments are conducted on a computing cluster with NVIDIA V100 GPUs, each with 32GB of VRAM. The operating system used is Ubuntu 18.04, and the models are implemented using the PyTorch framework.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021. S1

[2] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *CVPR*, 2018. S2

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Gi-

[2] https://www.nuscenes.org/terms-of-use

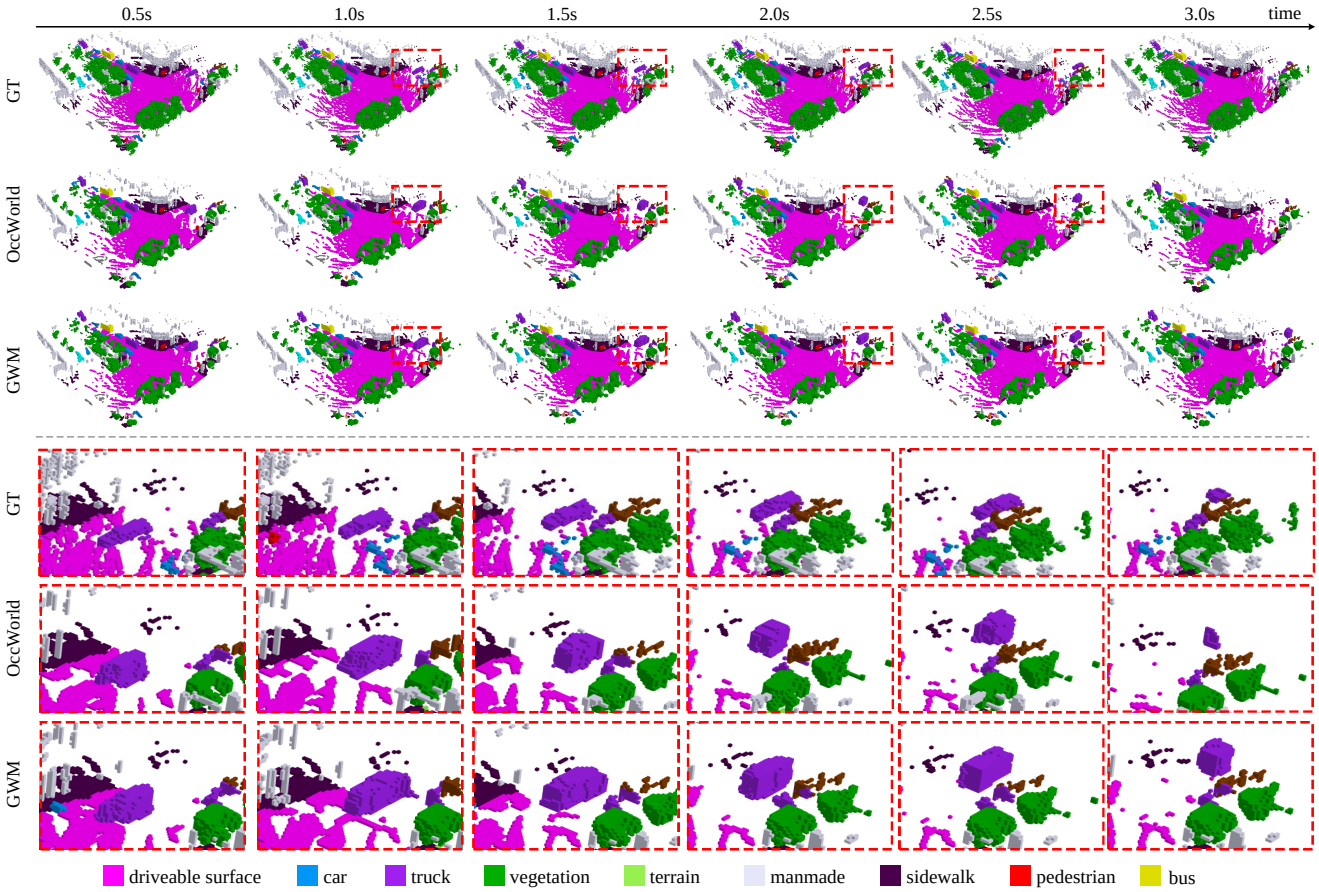

Figure S1. **Qualitative results of 4D occupancy forecasting on the Occ3D dataset [23] (§C.4).**

ancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. S4

[4] Anh-Quan Cao and Raoul De Charette. Monoscene: Monocular 3d semantic scene completion. In *CVPR*, 2022. S1

[5] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, 2024. S1

[6] Guikun Chen and Wenguan Wang. A survey on 3d gaussian splatting. *arXiv preprint arXiv:2401.03890*, 2024. S1

[7] Xiaokang Chen, Kwan-Yee Lin, Chen Qian, Gang Zeng, and Hongsheng Li. 3d sketch-aware semantic scene completion via semi-supervised structure prior. In *CVPR*, 2020. S1

[8] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. S1

[9] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *ECCV*, 2022. S2

[10] Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. S1

[11] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *CVPR*, 2023. S1

[12] 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. S1, S2

[13] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 42(4):1–14, 2023. S1, S2

[14] Tarasha Khurana, Peiyun Hu, Achal Dave, Jason Ziglar, David Held, and Deva Ramanan. Differentiable raycasting for self-supervised occupancy forecasting. In *ECCV*, 2022. S1

[15] Jie Li, Kai Han, Peng Wang, Yu Liu, and Xia Yuan. Anisotropic convolutional networks for 3d semantic scene completion. In *CVPR*, 2020. S1

[16] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023. S1

[17] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view syn-

thesis. *Communications of the ACM*, 65(1):99–106, 2021. S1

[18] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 41(4):1–15, 2022. S1

[19] Mingjie Pan, Jiaming Liu, Renrui Zhang, Peixiang Huang, Xiaoqi Li, Li Liu, and Shanghang Zhang. Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision. In *ICRA*, 2024. S1

[20] Yi-Ling Qiao, Alexander Gao, Yiran Xu, Yue Feng, Jia-Bin Huang, and Ming C Lin. Dynamic mesh-aware radiance fields. In *ICCV*, 2023. S1

[21] Marie-Julie Rakotosaona, Fabian Manhardt, Diego Martin Arroyo, Michael Niemeyer, Abhijit Kundu, and Federico Tombari. Nerfmeshing: Distilling neural radiance fields into geometrically-accurate 3d meshes. In *3DV*, 2024. S1

[22] Luis Roldao, Raoul de Charette, and Anne Verroust-Blondet. Lmscnet: Lightweight multiscale 3d semantic completion. In *3DV*, 2020. S1

[23] 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. In *NeurIPS*, 2024. S1, S4, S5

[24] Wenwen Tong, Chonghao Sima, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, et al. Scene as occupancy. In *ICCV*, 2023. S1

[25] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *NeurIPS*, 2017. S2

[26] Xiaofeng Wang, Zheng Zhu, Wenbo Xu, Yunpeng Zhang, Yi Wei, Xu Chi, Yun Ye, Dalong Du, Jiwen Lu, and Xingang Wang. Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. In *ICCV*, 2023. S1

[27] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *ICCV*, 2023. S1

[28] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *ICCV*, 2023. S1

[29] 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 *CVPR*, 2024. S1

[30] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In *AAAI*, 2021. S1

[31] Bangbang Yang, Chong Bao, Junyi Zeng, Hujun Bao, Yinda Zhang, Zhaopeng Cui, and Guofeng Zhang. Neumesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. In *ECCV*, 2022. S1

[32] Yu Yang, Jianbiao Mei, Yukai Ma, Siliang Du, Wenqing Chen, Yijie Qian, Yuxiang Feng, and Yong Liu. Driving in the occupancy world: Vision-centric 4d occupancy forecasting and planning via world models for autonomous driving. In *AAAI*, pages 9327–9335, 2025. S2, S3

[33] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *CVPR*, 2024. S1

[34] Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. *arXiv preprint arXiv:2311.16038*, 2023. S1, S2, S3

[35] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *CVPR*, 2024. S1

[36] Sicheng Zuo, Wenzhao Zheng, Yuanhui Huang, Jie Zhou, and Jiwen Lu. Pointocc: Cylindrical tri-perspective view for point-based 3d semantic occupancy prediction. *arXiv preprint arXiv:2308.16896*, 2023. S1

[37] Sicheng Zuo, Wenzhao Zheng, Yuanhui Huang, Jie Zhou, and Jiwen Lu. Pointocc: Cylindrical tri-perspective view for point-based 3d semantic occupancy prediction. *arXiv preprint arXiv:2308.16896*, 2023. S1