

# Urban4D: Semantic-Guided 4D Gaussian Splatting for Urban Scene Reconstruction

Ziwen Li

Jiaxin Huang

Runnan Chen

Yunlong Che

Yandong Guo

Tongliang Liu

Fakhri Karray

Mingming Gong

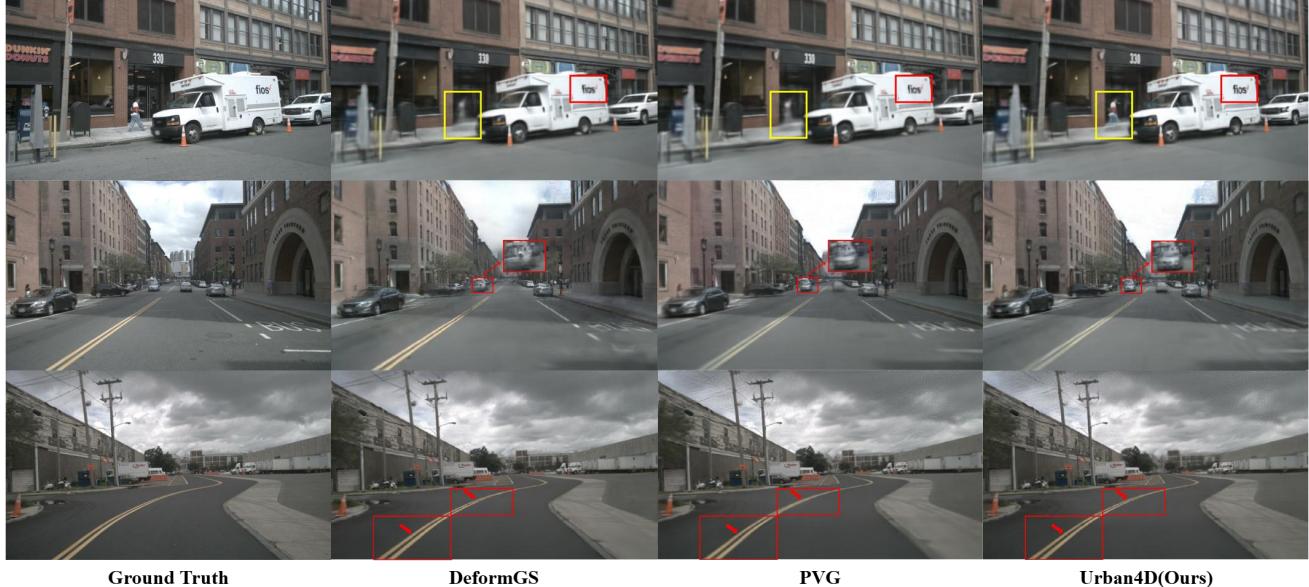


Figure 1. Qualitative comparison on the nuScenes [4] dataset. While DeformGS [40] achieves comparable results on static regions, it fails on dynamic objects, producing severe artifacts and blurred reconstructions. In contrast, our **Urban4D** maintains high fidelity for both dynamic objects and static backgrounds, also surpassing the reconstruction quality of PVG [7].

## Abstract

Reconstructing dynamic urban scenes presents significant challenges due to their intrinsic geometric structures and spatiotemporal dynamics. Existing methods that attempt to model dynamic urban scenes without leveraging priors on potentially moving regions often produce suboptimal results. Meanwhile, approaches based on manual 3D annotations yield improved reconstruction quality but are impractical due to labor-intensive labeling. In this paper, we revisit the potential of 2D semantic maps for classifying dynamic and static Gaussians and integrating spatial and temporal dimensions for urban scene representation. We introduce **Urban4D**, a novel framework that employs a semantic-guided decomposition strategy inspired by advances in deep 2D semantic map generation. Our approach

*distinguishes potentially dynamic objects through reliable semantic Gaussians. To explicitly model dynamic objects, we propose an intuitive and effective 4D Gaussian splatting (4DGS) representation that aggregates temporal information through learnable time embeddings for each Gaussian, predicting their deformations at desired timestamps using a multilayer perceptron (MLP). For more accurate static reconstruction, we also design a k-nearest neighbor (KNN)-based consistency regularization to handle the ground surface due to its low-texture characteristic. Extensive experiments on real-world datasets demonstrate that **Urban4D** not only achieves comparable or better quality than previous state-of-the-art methods but also effectively captures dynamic objects while maintaining high visual fidelity for static elements.*



Figure 2. Different from previous 3DGS-based methods which leverage pre-trained segmentation models to predict 2D semantic maps for downstream tasks, we revisit the potential of these estimated semantic maps to handle potentially dynamic objects for reconstruction. From left to right: input image, semantic map, our reconstruction, and reconstruction without potentially dynamic objects. Our method effectively preserves the geometry and appearance of both static structures and potentially dynamic objects.

## 1. Introduction

Recent advancements in urban scene reconstruction have witnessed remarkable progress, particularly with the introduction of 3D Gaussian Splatting (3DGS) [13], which has enhanced real-time rendering while maintaining high reconstruction quality for static scenes. However, directly extending these capabilities to model dynamic and large-scale urban scenes presents significant challenges due to spatiotemporal dynamics and intricate geometric structures.

Urban scenes can generally be categorized into two types: *dynamic objects*, such as vehicles and pedestrians, which exhibit complex motion patterns, and *static elements*, which maintain consistent spatial relationships. Specifically, low-texture regions (*e.g.*, ground surfaces) pose challenges for accurate static reconstruction due to their lack of salient patterns. These unique characteristics make reconstructing dynamic objects in urban scenes particularly difficult. The coexistence of static and dynamic elements necessitates a model that can distinguish between and process them independently to ensure accurate reconstruction.

Previous dynamic NeRF-based methods, such as [32, 38], attempt to address these challenges by introducing time-conditioned neural networks to model motion, but they suffer from slow rendering speeds and limited capacity to handle complex and large-scale urban scenes. Some methods explore suitable representations, such as Periodic Vibration Gaussian (PVG) [7], which proposes a unified framework by incorporating periodic vibration-based temporal dynamics in urban scenes. However, such a unified representation may degrade performance when treating static elements with temporal processes. Other 3DGS-based methods resort to manually labeled 3D bounding boxes to process dynamic objects [37, 41], which is labor-intensive and impractical for large-scale or continuously changing environments.

Furthermore, we have witnessed that some of these methods adopt pre-trained segmentation models, such as SegFormer [35], for semantic novel view rendering. Semantic segmentation based on deep learning provides more reliable classifications of scene components in 2D images.

Leveraging these results, we may effectively classify dynamic and static Gaussians without relying on manually labeled 3D annotations. The observations in Figure 2 lead us to an intriguing question: *Can we revisit the potential of 2D semantic maps for classifying potentially dynamic objects and leverage spatiotemporal information to adequately represent urban scenes?* The answer to this question uncovers an unexplored area in reconstructing dynamic objects based on 4D Gaussian splatting (4DGS), enabling a more holistic and precise representation.

Inspired by this perspective, we propose **Urban4D**, a novel framework to overcome the limitations of traditional 3DGS-based methods in handling temporal dynamics and reconstructing urban scenes. Specifically, we design a semantic-guided decomposition strategy, which builds on recent progress in deep 2D semantic map generation to distinguish potentially dynamic objects from static elements by assigning reliable semantic classes to Gaussians. To explicitly model dynamic objects, we propose an intuitive and effective 4DGS representation of dynamic objects by incorporating a learnable time embedding with each Gaussian and predicting Gaussian deformations at desired timestamps through a multilayer perception (MLP). Moreover, we employ a k-nearest neighbor (KNN)-based consistency regularization mechanism to enforce consistency across neighboring Gaussians to alleviate the inaccurate static reconstruction caused by the low-texture characteristic of the ground surface.

Our extensive experiments on multiple real-world datasets demonstrate that **Urban4D** achieves comparable or better quality than previous state-of-the-art methods. The proposed framework effectively captures dynamic objects while maintaining high visual fidelity for static elements. By addressing the limitations of existing methods and introducing innovative techniques for semantic-guided decomposition and temporal modeling, Urban4D appears to be a compelling solution for representing the complexity and detail of dynamic urban environments. Our contributions can be summarized as follows:

- We propose **Urban4D**, a novel framework that extends

the capabilities of 3DGS to effectively handle dynamic urban scenes by leveraging semantic and spatiotemporal information.

- An intuitive semantic-guided decomposition strategy based on 2D semantic maps, which distinguishes between dynamic and static elements through reliable semantic Gaussians.
- An intuitive and efficient 4D Gaussian splatting representation that incorporates a learnable time embedding for each Gaussian and predicts deformations over time with an MLP, effectively modeling dynamic objects in urban scenes.
- A KNN-based consistency regularization mechanism to handle the low-texture ground surface in urban scenes, improving the accuracy of static reconstruction.

## 2. Related Work

### 2.1. Neural Scene Representations

Neural representations have revolutionized novel view synthesis, with NeRF [20] leading significant advances in this field. NeRF utilizes multi-layer perceptrons (MLPs) and differentiable volume rendering to reconstruct 3D scenes from 2D images and camera poses. While demonstrating impressive results for bounded scenes, its application to large-scale unbounded scenes remains challenging due to computational constraints and the requirement for consistent camera-object distances.

Various improvements have been proposed to address NeRF’s limitations. Training speed has been enhanced through techniques like voxel grids [9, 10, 25], hash encoding [21], and tensor factorization [6], while rendering quality has been improved through better anti-aliasing [3, 14, 17] and reflection modeling [11, 31].

More recently, 3D Gaussian Splatting [13] has emerged as a promising alternative, offering faster training and rendering while maintaining high quality results. This explicit representation combines the advantages of volumetric rendering with efficient rasterization-based techniques. Compared to previous explicit representations (*e.g.*, mesh, voxels), 3D-GS can model complex shapes while allowing fast, differentiable rendering through splat-based rasterization.

### 2.2. Dynamic Scene Modeling

Dynamic scene reconstruction methods generally fall into two categories: deformation-based and modulation-based approaches. Deformation-based methods [5, 22–24, 28] model scene dynamics through canonical space mapping and deformation networks, while modulation-based approaches [15, 16, 18, 33] incorporate temporal information directly. These methods have shown promising results in controlled environments but face significant challenges when applied to complex real-world scenarios with multi-

ple dynamic objects.

For urban environments, several pioneering works have tackled static scene reconstruction by introducing multi-scale NeRF variants [19, 26, 29] and incorporating advanced rendering techniques like mipmapping [1, 2]. Building upon these foundations, recent methods [30, 36, 38] have explored the integration of multi-modal data, combining RGB images with LiDAR point clouds to enhance geometric accuracy. However, the challenge of jointly modeling static and dynamic elements remains complex, particularly due to high-speed movements and sparse viewpoints typical in driving scenarios.

To address these challenges, recent works have proposed various scene decomposition strategies. Scene graph representations [27, 32, 37, 39, 41] enable explicit object-level modeling and control. However, most current approaches either treat all dynamic elements uniformly [7] or rely heavily on manual annotations [37, 41].

## 3. Methodology

**Method overview.** Our proposed method aims to reconstruct dynamic urban scenes by leveraging semantic information to effectively distinguish between static and dynamic elements. Given a sequence of images  $\{I_t\}_{t=1}^T$  and corresponding LiDAR point clouds  $\{P_t\}_{t=1}^T$  captured by a moving vehicle, we aim to reconstruct dynamic urban scenes. For each frame, semantic maps  $\{S_t\}_{t=1}^T$  are predicted using an off-the-shelf pre-trained segmentation model. Based on our aforementioned insight, we develop a novel framework to intuitively represent dynamic elements in 4D and enhance the reconstruction of static elements.

Our method consists of three main elements (Figure 4): (1) semantic-guided decomposition that separates the scene into static and potentially dynamic Gaussians based on semantic information (Sec. 3.2), (2) 4D Gaussians Splatting representation for dynamic scenes (Sec. 3.3), and (3) a regularization method that enforces consistency in low-texture regions, particularly ground surfaces, through KNN-based optimization (Sec. 3.4).

### 3.1. Preliminaries

Deformable Gaussian Splatting represents a dynamic scene using a set of deformable 3D Gaussians  $\{\mathcal{G}_i\}_{i=1}^N$ . Each Gaussian  $\mathcal{G}_i$  is characterized by its canonical position  $\mu_i \in \mathbb{R}^3$ , rotation  $\mathbf{r}_i \in \mathbb{R}^4$ , and scale  $\mathbf{s}_i \in \mathbb{R}^3$ . At time  $t$ , both position and shape undergo deformation:

$$\begin{aligned}\mu_i(t) &= \mu_i + \mathcal{F}_{pos}(\mu_i, t), \\ \mathbf{r}_i(t) &= \mathbf{r}_i + \mathcal{F}_{rot}(\mu_i, t), \\ \mathbf{s}_i(t) &= \mathbf{s}_i + \mathcal{F}_{scale}(\mu_i, t),\end{aligned}\quad (1)$$

where  $\mathcal{F}_{pos}$ ,  $\mathcal{F}_{rot}$ , and  $\mathcal{F}_{scale}$  are deformation MLPs predicting position offset, rotation change, and scale adjust-

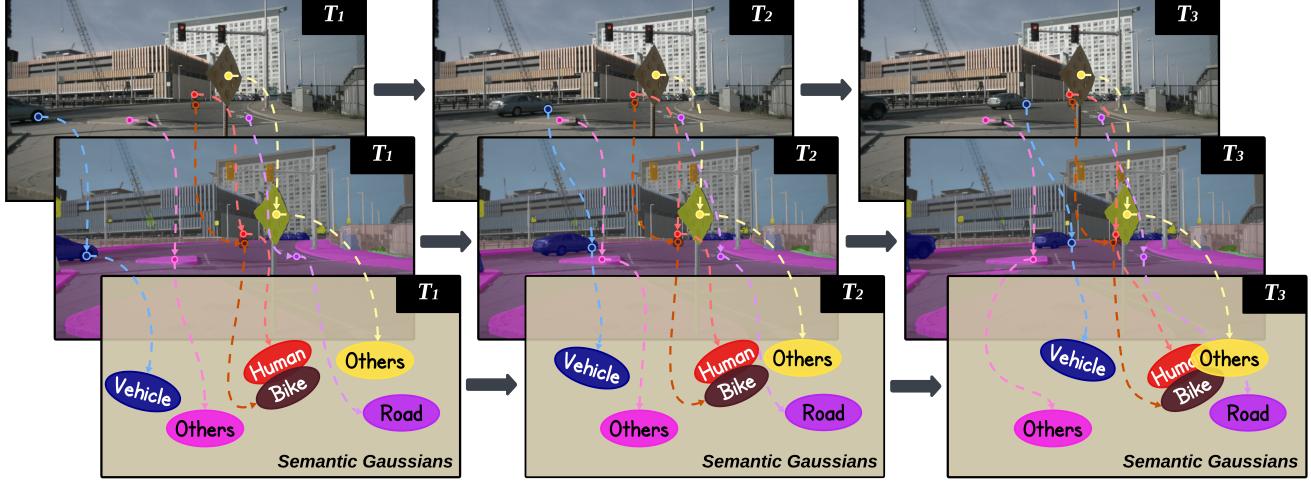


Figure 3. Semantic-guided decomposition over time. For each timestamp ( $T_1, T_2, T_3$ ), semantic Gaussians of the current frame are obtained through rendering and supervision of corresponding semantic maps. Dynamic classes include vehicles, pedestrians, and cyclists, while the static set comprises buildings, vegetation, and roads. For simplicity, we use the "Road" to represent ground surfaces.

ment respectively based on the Gaussian's position and time. The covariance matrix at time  $t$  is then computed as:

$$\Sigma_i(t) = \mathbf{r}_i(t)\mathbf{s}_i(t)\mathbf{s}_i(t)^T\mathbf{r}_i(t)^T. \quad (2)$$

Each Gaussian also maintains its opacity  $\alpha_i \in [0, 1]$  and color represented as spherical harmonics coefficients  $\mathbf{c}_i$  for appearance modeling. The rendering process follows standard Gaussian splatting with the deformed parameters.

### 3.2. Semantic-guided Decomposition

Effectively separating dynamic and static elements is essential for handling urban scenes. Leveraging 2D semantic maps from a pre-trained segmentation model, we introduce a systematic approach to identify and separate potentially dynamic elements, allowing for specialized treatment of movable objects and static elements. Specifically, as shown in Figure 3, for every timestamp, each Gaussian  $\mathcal{G}_i$  of the current frame is assigned a semantic label  $s_i \in \{1, \dots, K\}$ , obtained through rendering and supervision of semantic maps, where  $K$  represents the total number of semantic classes.

Our method focuses on potentially dynamic objects in urban scenes, typically comprising well-defined semantic categories. We identify dynamic classes  $\mathcal{C}_d$ , including vehicles, pedestrians, and cyclists, which frequently exhibit motion and require specialized handling. The remaining classes form the static set  $\mathcal{C}_s$ , including buildings, vegetation, and road surfaces. This semantic understanding enables the decomposition of the scene into two disjoint sets:

$$\begin{aligned} \mathcal{G}_i^d &= \{\mathcal{G}_i | s_i \in \mathcal{C}_d\}, \\ \mathcal{G}_i^s &= \{\mathcal{G}_i | s_i \in \mathcal{C}_s\}, \end{aligned} \quad (3)$$

where  $\mathcal{G}_i^d$  represents dynamic Gaussians (i.e., Gaussians of potentially dynamic objects) requiring 4D modeling, and  $\mathcal{G}_i^s$  represents static Gaussians. This semantic-guided decomposition offers several key advantages: (1) it enables specialized motion modeling for dynamic objects while preserving static scene consistency, (2) it eliminates the need for labor-intensive manual annotations, and (3) it ensures consistent motion treatment for all dynamic objects, regardless of their movement state.

The decomposition lays the foundation for our two-stream optimization strategy. Dynamic Gaussians in  $\mathcal{G}_i^d$  undergo dedicated motion refinement (Sec. 3.3) to accurately capture movement, while static Gaussians receive geometric regularization (Sec. 3.4) to enhance scene stability. Additionally, we employ an optimizable environment texture map for sky representation, which is rendered separately and composited with the Gaussian-rendered image through alpha blending, as described in [7]. This separation allows targeted optimization without interference between dynamic and static components. Optimizing stationary dynamic Gaussians (e.g., parked vehicles) within a distinct subset is still more efficient than methods like DeformGS [40], which jointly model entire scene dynamics. This targeted approach results in a more accurate reconstruction of both dynamic and static elements.

### 3.3. 4D Gaussian Splatting Representation

To represent our dynamic Gaussians  $\{\mathcal{G}_i^d\}$  in 4D, our method extends the deformation mechanism of DeformGS [40] by introducing a learnable time embedding for each Gaussian. While DeformGS [40] directly maps spatial positions and time to deformations, our approach leverages temporal context through Gaussian-specific embeddings.

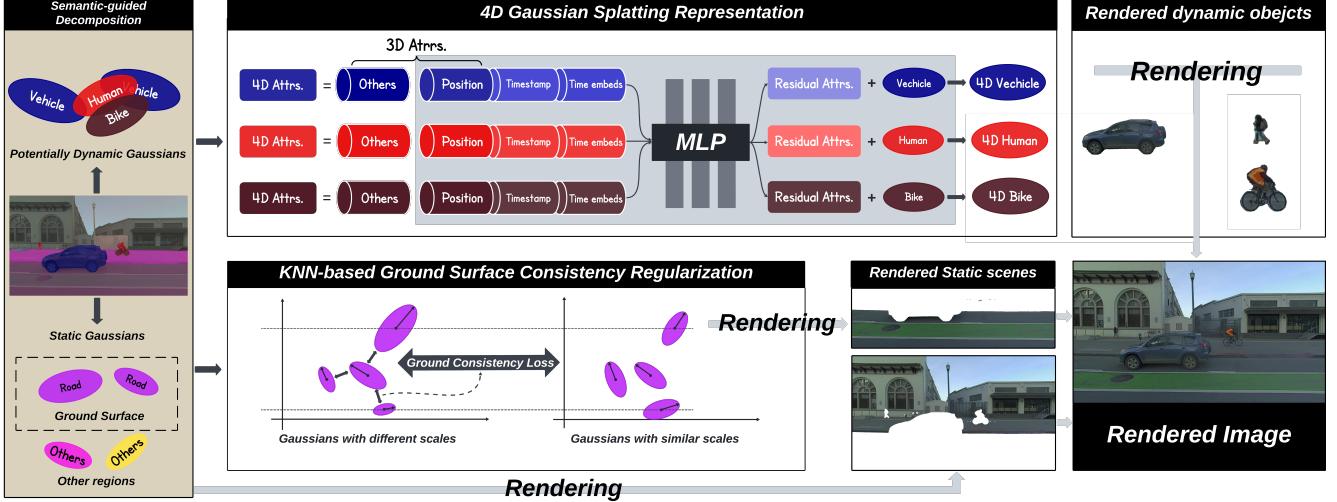


Figure 4. Overview of **Urban4D** framework. Given input images with semantic information, Gaussians are classified into static and potentially dynamic elements through semantic-guided decomposition. Dynamic objects are represented in 4D Gaussian Splatting that captures motion patterns by incorporating a learnable time embedding, with deformations predicted at desired timestamps using an MLP. For static Gaussians, particularly low-texture areas like ground surfaces, consistency is enforced using a KNN-based regularization mechanism.

Specifically, for each dynamic Gaussian  $\mathcal{G}_i^d$ , we maintain a learnable time embedding vector  $\mathbf{e}_i \in \mathbb{R}^{D_e}$ . At time step  $t$ , we form the input feature by concatenating this temporal embedding with position and time information:

$$\mathbf{h}_i(t) = [\mu_i; t; \mathbf{e}_i], \quad (4)$$

where  $[;]$  denotes concatenation,  $t$  is the normalized time stamp, and  $\mu_i \in \mathbb{R}^3$  represents the original 3D position of the Gaussian. This temporal-aware design enables more accurate modeling of complex motions compared to the direct mapping used in DeformGS [40].

This combined feature vector is processed by a lightweight MLP to predict residual corrections:

$$[\Delta\mu_i(t), \Delta\alpha_i(t), \Delta\mathbf{r}_i(t), \Delta\mathbf{s}_i(t)] = \text{MLP}(\mathbf{h}_i(t)), \quad (5)$$

where  $\Delta\mu_i(t) \in \mathbb{R}^3$ ,  $\Delta\alpha_i(t) \in \mathbb{R}$ ,  $\Delta\mathbf{r}_i(t) \in \mathbb{R}^4$ ,  $\Delta\mathbf{s}_i(t) \in \mathbb{R}^3$  are the predicted position, opacity, rotation and scale residuals, respectively. The final parameters of 4D Gaussians at time  $t$  are obtained by:

$$\begin{aligned} \mu'_i(t) &= \mu_i + \Delta\mu_i(t), \\ \alpha'_i(t) &= \alpha_i + \Delta\alpha_i(t), \\ \mathbf{r}'_i(t) &= \mathbf{r}_i + \Delta\mathbf{r}_i(t), \\ \mathbf{s}'_i(t) &= \mathbf{s}_i + \Delta\mathbf{s}_i(t), \end{aligned} \quad (6)$$

where  $\mu_i \in \mathbb{R}^3$ ,  $\alpha_i \in [0, 1]$ ,  $\mathbf{r}_i \in \mathbb{R}^3$ ,  $\mathbf{s}_i \in \mathbb{R}^3$  denotes the initial 3D position, opacity, rotation and scaling of the  $i$ -th Gaussian. These refined parameters are then used in the standard 3D Gaussian Splatting rendering process to generate the final images. This refinement mechanism allows

each dynamic Gaussians to adapt its attributions based on its temporal context, enabling more accurate representation of moving objects in the scene.

The MLP architecture is intentionally kept lightweight to maintain computational efficiency while providing sufficient capacity for modeling temporal dynamics. The detailed architecture of MLPs used in our method is provided in the supplementary material for reproducibility. The entire refinement process is end-to-end trainable along with the main 3DGS optimization objectives.

### 3.4. Ground Surface Consistency Regularization

In urban driving scenes, ground surfaces constitute a significant portion of the environment and typically exhibit low-texture characteristics. While ground-level Gaussians should theoretically share similar properties due to their homogeneous nature, enforcing strict uniformity across all ground Gaussians would be oversimplified and impractical, as real-world surfaces often contain variations and irregularities. The scale parameter of a Gaussian, derived from its covariance matrix, inherently encodes local geometric information analogous to surface normals [8, 12]. A well-behaved ground surface should exhibit smooth transitions in its local geometry, making scale a particularly suitable target for regularization. This motivates us to regularize the scale parameters rather than other Gaussian properties.

For each ground Gaussian  $\mathcal{G}_i \in \mathcal{G}_g$  (where  $\mathcal{G}_g \subset \mathcal{G}_i^s$  denotes the set of ground surface Gaussians), we identify its  $N$  nearest neighbors:

$$\mathcal{N}_i = \text{KNN}(\mathcal{G}_i, \mathcal{G}_g, N), \quad (7)$$

where KNN retrieves the  $N$  spatially closest Gaussians to  $\mathcal{G}_i$  from the global set  $\mathcal{G}_g$ . The neighbors are determined based on the Euclidean distance between Gaussian centers  $\mu_i$ , forming a local neighborhood for geometric consistency. We then introduce a local consistency loss that encourages similar scale properties within each local neighborhood:

$$\mathcal{L}_{\text{ground}} = \sum_{\mathcal{G}_i \in \mathcal{G}_g} \|s_i - \frac{1}{N} \sum_{\mathcal{G}_j \in \mathcal{N}_i} s_j\|_2^2, \quad (8)$$

where  $s_i$  and  $s_j$  represents the scale parameter of the Gaussian  $\mathcal{G}_i$  and  $\mathcal{G}_j$ . By regularizing the scale parameters, we effectively enforce consistency in the local surface geometry while preserving the ability to model natural surface variations. This approach leads to more coherent ground surface reconstruction, as similar scale parameters in a local neighborhood implicitly enforce consistent surface normal orientations, resulting in improved geometric fidelity of the ground surface representation.

### 3.5. Optimization Strategy

Our optimization objective comprises multiple loss terms that jointly ensure high-quality visual rendering, geometric accuracy, and semantic consistency. The overall loss function is formulated as follows:

$$\begin{aligned} \mathcal{L}_{\text{total}} = & \lambda_1 \mathcal{L}_{\text{L1}} + \lambda_2 \mathcal{L}_{\text{SSIM}} + \lambda_3 \mathcal{L}_{\text{sem}} \\ & + \lambda_4 \mathcal{L}_{\text{ground}} + \lambda_5 \mathcal{L}_{\text{depth}} + \lambda_6 \mathcal{L}_{\text{sky}} \end{aligned} \quad (9)$$

where  $\{\lambda_i\}_{i=1}^6$  are weighting coefficients balancing different loss terms. Each loss term serves a specific purpose in our optimization.

**Appearance Supervision.** The L1 loss and SSIM loss work together to ensure accurate color reproduction and structural similarity:

$$\begin{aligned} \mathcal{L}_{\text{L1}} &= \|I_{\text{rendered}} - I_{\text{gt}}\|_1, \\ \mathcal{L}_{\text{SSIM}} &= 1 - \text{SSIM}(I_{\text{rendered}}, I_{\text{gt}}), \end{aligned} \quad (10)$$

where  $I_{\text{rendered}}$  and  $I_{\text{gt}}$  denote the rendered image and ground-truth image respectively.

**Semantic Consistency.** The semantic loss ensures correct class predictions for each Gaussian:

$$\mathcal{L}_{\text{sem}} = \text{CE}(S_{\text{rendered}}, S_{\text{gt}}), \quad (11)$$

where CE denotes cross-entropy loss between rendered semantic maps  $S_{\text{rendered}}$  and ground-truth semantic maps  $S_{\text{gt}}$ .

**Ground Consistency.** As detailed in Sec. 3.4, the ground consistency loss enforces local geometric similarity as defined in Eq. 8.

**Geometric Supervision.** The inverse depth loss aligns the scene geometry with LiDAR measurements:

$$\mathcal{L}_{\text{depth}} = \left\| \frac{1}{D_{\text{rendered}}} - \frac{1}{D_{\text{lidar}}} \right\|_1, \quad (12)$$

where  $D_{\text{rendered}}$  and  $D_{\text{lidar}}$  represent the rendered depth and LiDAR depth respectively.

**Sky Region Handling.** For sky regions, we encourage low opacity to prevent incorrect geometry:

$$\mathcal{L}_{\text{sky}} = \sum_{\mathcal{G}_i \in \mathcal{G}_{\text{sky}}} \|\alpha_i\|_1, \quad (13)$$

where  $\mathcal{G}_{\text{sky}}$  represents the set of sky Gaussians.

These joint loss terms collectively constrain the scene reconstruction process, ensuring high-quality results.

## 4. Experiments

### 4.1. Implementation Details

We initialize Gaussian points from both LiDAR points (with projected RGB and semantic values) and 200K random points sampled within a sphere. Following previous 3DGS-based methods to predict reliable semantic Gaussians, we use SegFormer [35] as our pre-trained segmentation model. The learning rate of MLP starts from  $1.6 \times 10^{-4}$  and decreases to  $1.6 \times 10^{-6}$ . For each loss term, the weighting coefficients are empirically set to  $\lambda_1 = 0.8$ ,  $\lambda_2 = 0.2$ ,  $\lambda_3 = 0.01$ ,  $\lambda_4 = 0.0001$ ,  $\lambda_5 = 0.1$  and  $\lambda_6 = 0.01$ . All experiments are conducted on a single NVIDIA V100s with 32GB memory.

### 4.2. Datasets

Our experiments are conducted on two popular autonomous driving datasets: PandaSet [7] and nuScenes [4]. We primarily evaluate our method on nuScenes [4], which consists of 1000 scenes captured in Boston and Singapore under diverse urban scenarios with multimodal sensor data, featuring challenging weather conditions and complex urban scenes. We perform ablation studies on PandaSet [7], which is a comprehensive autonomous driving dataset captured in San Francisco, containing 103 sequences of synchronized LiDAR-camera data and images. For both datasets, we use the officially provided camera poses.

### 4.3. Results and Comparisons

**Results on nuScenes [4].** We evaluate our method against previous approaches on the nuScenes [4] dataset, with a comprehensive comparison across different scene components and object categories. As shown in Table 1, our method demonstrates consistent improvements over NeRF-based methods [30, 38], traditional 3DGS [13], PVG [7] and DeformGS [40] across all evaluation metrics, with significant improvements in dynamic object reconstruction.

For full image reconstruction, our method achieves 26.47 PSNR and 0.839 SSIM, surpassing PVG [7] by 0.24 PSNR. The improvement is more pronounced in non-sky regions, where our approach shows a substantial gain of 2.21 PSNR

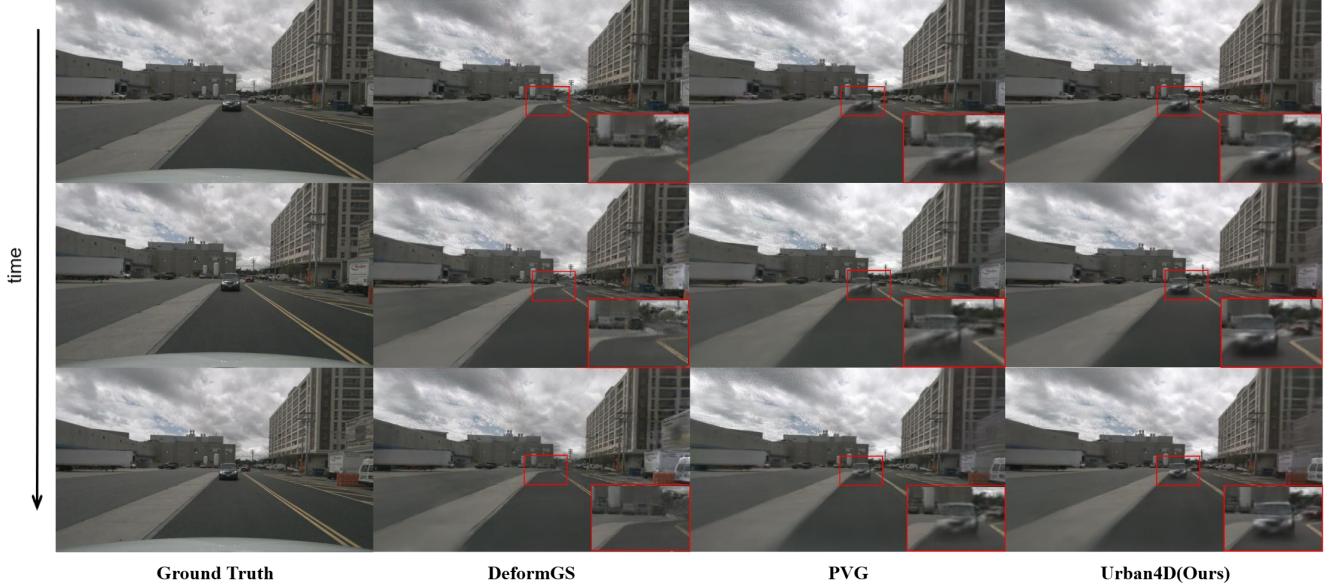


Figure 5. comparison of reconstruction quality across consecutive frames. DeformGS [40] struggles significantly with reconstructing dynamic objects, resulting in severe artifacts and a failure to accurately represent motion. PVG [7] captures dynamic vehicles to some extent but suffers from noticeable blurring, particularly in the lower parts of the objects. In contrast, **Urban4D** delivers superior reconstruction quality, maintaining high fidelity and preserving clear details throughout the dynamic objects.

Table 1. Quantitative comparison with existing methods on the nuScenes [4] dataset. The best results in each group are highlighted in **bold**. Our method outperforms previous approaches across all evaluation metrics.

Method	Full Image		Non-Sky		Human		Vehicle		Dynamic	
	PSNR↑	SSIM↑								
SUDS [30]	20.02	0.605	20.45	0.621	21.20	0.633	21.98	0.678	21.49	0.646
DeformGS [40]	25.32	0.825	25.27	0.822	21.44	0.6456	21.72	0.700	21.62	0.684
3DGS [13]	26.02	0.825	26.45	0.836	23.20	0.721	23.98	0.794	23.59	0.756
EmerNeRF [38]	26.12	0.830	26.50	0.840	23.45	0.733	24.69	0.808	24.12	0.767
PVG [7]	26.23	0.834	26.72	0.841	23.98	0.743	24.73	0.815	24.33	0.774
<b>Urban4D (Ours)</b>	<b>26.47</b>	<b>0.839</b>	<b>28.93</b>	<b>0.878</b>	<b>24.88</b>	<b>0.783</b>	<b>26.05</b>	<b>0.855</b>	<b>25.33</b>	<b>0.810</b>

(28.93 vs 26.72) and 0.037 SSIM (0.878 vs 0.841) compared to PVG [7], indicating better handling of complex geometric structures in urban environments.

The advantages of our method are most evident in dynamic object reconstruction. For human instances, we achieve 24.88 PSNR and 0.783 SSIM, outperforming PVG [7] by 0.90 PSNR and 0.040 SSIM. In vehicle reconstruction, our method demonstrates even larger improvements, reaching 26.05 PSNR and 0.855 SSIM, surpassing PVG [7] by 1.32 PSNR and 0.040 SSIM. These significant gains can be attributed to our dynamic-focused semantic decomposition strategy, which enables specialized treatment of moving objects.

We present qualitative comparisons in Fig. 5. For dynamic objects like vehicles and pedestrians, our method

shows significant improvements over PVG [7] and DeformGS [40], with notably reduced motion blur and sharper object boundaries. This is particularly evident in regions with fast-moving objects, where previous methods tend to produce blurry reconstructions. Meanwhile, our approach also demonstrates better reconstruction quality for static scene elements such as roads, preserving more detailed textures and geometric structures. These visual improvements align well with our quantitative results, where we achieve consistently higher scores across both dynamic and static regions. The qualitative results in Fig. 5 further support these findings. In consecutive frame reconstruction, DeformGS [40] fails to handle dynamic objects, producing severe artifacts. While PVG [7] captures the overall shape of moving vehicles, it suffers from noticeable blurring ar-

Table 2. Quantitative comparison with state-of-the-art methods on the PandaSet [34] dataset. We report image reconstruction and novel view synthesis metrics. The best results are shown in **bold**.

Method	Image Reconstruction		Novel View Synthesis	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
3DGS [13]	23.67	0.743	22.14	0.713
EmerNeRF [38]	26.45	0.812	24.89	0.765
PVG [7]	27.15	0.836	25.92	0.798
<b>Urban4D (Ours)</b>	<b>27.74</b>	<b>0.850</b>	<b>26.56</b>	<b>0.814</b>

tifacts, particularly in the lower parts of the vehicles. In contrast, our method achieves clearer and more consistent reconstruction across all dynamic objects.

**Results on PandaSet [34].** To further validate the generalization capability of our method, we conduct additional experiments on the PandaSet [34] dataset. As shown in Table 2, our approach demonstrates consistently superior performance across both image reconstruction and novel view synthesis tasks. For image reconstruction, our method achieves 27.74 PSNR and 0.850 SSIM, outperforming PVG [7] by 0.59 PSNR and 0.014 SSIM. The performance advantage is even more pronounced when compared to vanilla 3D Gaussian Splatting (3DGS [13]), showing improvements of 4.07 PSNR and 0.107 SSIM. This significant gap highlights the effectiveness of our dynamic-focused scene decomposition strategy.

In the more challenging novel view synthesis task, our method maintains its lead with 26.56 PSNR and 0.814 SSIM, surpassing PVG [7] by 0.64 PSNR and 0.016 SSIM. Notably, while all methods experience some performance degradation in novel view synthesis compared to image reconstruction, our method shows the smallest drop (1.18 PSNR vs 1.23 PSNR for PVG [7]), indicating better scene understanding and geometry modeling. The consistent performance improvements across different datasets and evaluation metrics demonstrate the robustness and generalizability of our approach across varied urban scenes and capture conditions.

#### 4.4. Ablation Study

**Effectiveness of consistency regularization.** To validate the effectiveness of our ground surface consistency module, we conduct extensive experiments on the PandaSet [34] dataset. Table 3 shows the quantitative results of different variants of our method. The baseline model without road consistency achieves 26.31 PSNR and 0.810 SSIM. After incorporating our consistency constraint, the performance significantly improves by 0.25 dB in PSNR (26.56 vs. 26.31) and 0.004 in SSIM (0.814 vs. 0.810). This demonstrates that enforcing local geometric and appearance consistency on ground surfaces effectively enhances the re-

Table 3. Ablation study on road consistency module. We evaluate different design choices and parameter settings of our road consistency constraint on PandaSet [7].

Method	PSNR↑	SSIM↑
Baseline (w/o road consistency)	26.31	0.810
KNN-8	26.45	0.812
KNN-16 (Ours)	<b>26.56</b>	<b>0.814</b>
KNN-32	26.48	0.812

Table 4. Ablation study on time embedding. We report PSNR and SSIM on the PandaSet [34].

Method	PSNR↑	SSIM↑
w/o Time Embeddings	26.40	0.810
w/ Time Embeddings	<b>26.56</b>	<b>0.814</b>

construction quality.

We further investigate the impact of different numbers of nearest neighbors in our module. As shown in Table 3, using 16 neighbors yields the best performance (26.56 PSNR), outperforming both smaller ( $k=8$ , 26.45 PSNR) and larger ( $k=32$ , 26.48 PSNR) neighbor sizes. We hypothesize that 8 neighbors are insufficient to capture the local surface structure, while 32 neighbors may introduce noise from points that are too far away. The optimal choice of 16 neighbors provides a good local structure preservation.

**Effectiveness of Time Embeddings.** We also validate the effectiveness of our temporal modeling strategy through ablation studies. As shown in Table 4, equipping each Gaussian with a learnable time embedding improves the rendering quality (PSNR: 26.40 → 26.56, SSIM: 0.810 → 0.814) on the PandaSet [34] dataset. This demonstrates that point-specific temporal context helps capture more accurate deformation patterns compared to directly mapping spatial positions to deformations.

## 5. Conclusions

In conclusion, Urban4D provides a novel semantic-guided decomposition and 4D representation strategy for reconstructing dynamic urban scenes. By leveraging 2D semantic information, our approach effectively separates static and potentially dynamic components. We introduced specialized processing methods for different elements: a 4D Gaussian Splatting representation for dynamic objects and KNN-based consistency regularization for the ground surface. Both quality and quantitative results demonstrated that Urban4D improves rendering quality across various scene components, particularly for dynamic objects and low-texture regions.

## 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5855–5864, 2021. 3
- [2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022. 3
- [3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19697–19705, 2023. 3
- [4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giacomo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 1, 6, 7
- [5] Hongrui Cai, Wanquan Feng, Xuetao Feng, Yan Wang, and Juyong Zhang. Neural surface reconstruction of dynamic scenes with monocular rgb-d camera. *Advances in Neural Information Processing Systems*, 35:967–981, 2022. 3
- [6] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European conference on computer vision*, pages 333–350. Springer, 2022. 3
- [7] Yurui Chen, Chun Gu, Junzhe Jiang, Xiatian Zhu, and Li Zhang. Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering. *arXiv preprint arXiv:2311.18561*, 2023. 1, 2, 3, 4, 6, 7, 8
- [8] Kai Cheng, Xiaoxiao Long, Kaizhi Yang, Yao Yao, Wei Yin, Yuexin Ma, Wenping Wang, and Xuejin Chen. Gaussianpro: 3d gaussian splatting with progressive propagation. In *Forty-first International Conference on Machine Learning*, 2024. 5
- [9] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinrong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5501–5510, 2022. 3
- [10] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14346–14355, 2021. 3
- [11] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. Nerfren: Neural radiance fields with reflections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18409–18418, 2022. 3
- [12] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussian-shader: 3d gaussian splatting with shading functions for reflective surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5322–5332, 2024. 5
- [13] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2, 3, 6, 7, 8
- [14] Georgios Kopanas, Thomas Leimkühler, Gilles Rainer, Clément Jambon, and George Drettakis. Neural point catacaustics for novel-view synthesis of reflections. *ACM Transactions on Graphics (TOG)*, 41(6):1–15, 2022. 3
- [15] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022. 3
- [16] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021. 3
- [17] Junchen Liu, Wenbo Hu, Zhuo Yang, Jianteng Chen, Guoliang Wang, Xiaoxue Chen, Yantong Cai, Huan-ang Gao, and Hao Zhao. Rip-nerf: Anti-aliasing radiance fields with ripmap-encoded platonic solids. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 3
- [18] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *3DV*, 2024. 3
- [19] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7210–7219, 2021. 3
- [20] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 3
- [21] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 3
- [22] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 3
- [23] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021.
- [24] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields

- for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 3
- [25] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5459–5469, 2022. 3
- [26] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022. 3
- [27] Adam Tonderski, Carl Lindström, Georg Hess, William Ljungbergh, Lennart Svensson, and Christoffer Petersson. Neurad: Neural rendering for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14895–14904, 2024. 3
- [28] Edgar Treitschke, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12959–12970, 2021. 3
- [29] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12922–12931, 2022. 3
- [30] Haithem Turki, Jason Y Zhang, Francesco Ferroni, and Deva Ramanan. Suds: Scalable urban dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12375–12385, 2023. 3, 6, 7
- [31] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. 3
- [32] 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. 2, 3
- [33] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9421–9431, 2021. 3
- [34] Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, et al. Pandaset: Advanced sensor suite dataset for autonomous driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3095–3101. IEEE, 2021. 8
- [35] Enze Xie, Wenhui Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34: 12077–12090, 2021. 2, 6
- [36] Ziyang Xie, Junge Zhang, Wenye Li, Feihu Zhang, and Li Zhang. S-nerf: Neural radiance fields for street views. *arXiv preprint arXiv:2303.00749*, 2023. 3
- [37] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *ECCV*, 2024. 2, 3
- [38] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, et al. Emergenterf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023. 2, 3, 6, 7, 8
- [39] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1389–1399, 2023. 3
- [40] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20331–20341, 2024. 1, 4, 5, 6, 7
- [41] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21634–21643, 2024. 2, 3