Hierarchy UGP: Hierarchy Unified Gaussian Primitive for Large-Scale Dynamic Scene Reconstruction

Hongyang Sun[1,2*], Qinglin Yang[1*], Jiawei Wang[3*], Zhen Xu[1], Chen Liu[2]
Yida Wang[2], Kun Zhan[2], Hujun Bao[1], Xiaowei Zhou[1], Sida Peng[1†]
[1]Zhejiang University, [2]Li Auto, [3]UESTC

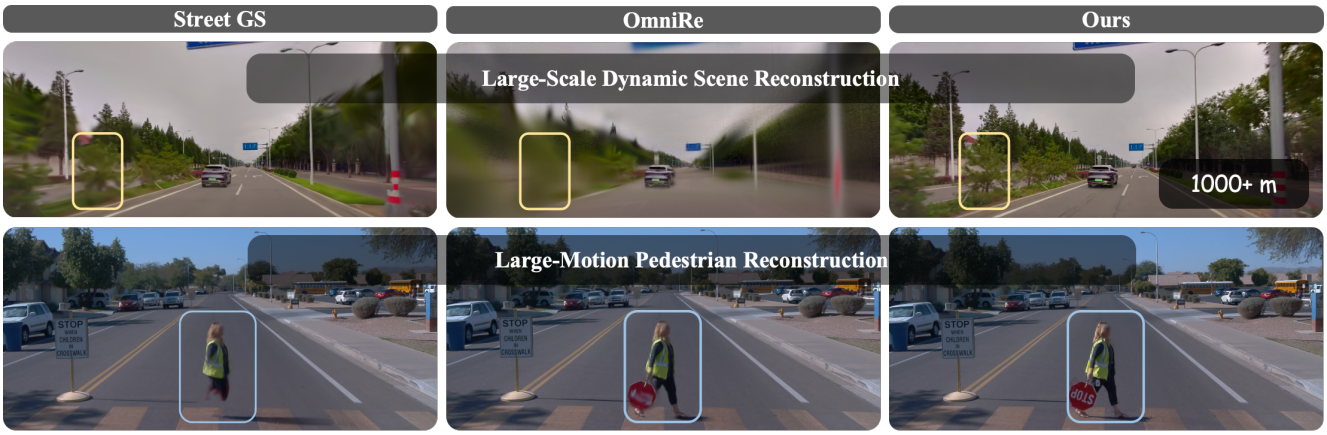


Figure 1. **Overview of the Reconstruction Improvement of Our Hierarchy UGP on Large-Scale and Large-Motion Scenarios.** On the large-scale dynamic scene, the reconstruction quality of Hierarchy UGP significantly outperforms previous methods [4, 40]. For reconstruction of pedestrian with large motions, Hierarchy UGP also demonstrates large improvements.

Abstract

Recent advances in differentiable rendering have significantly improved dynamic street scene reconstruction. However, the complexity of large-scale scenarios and dynamic elements, such as vehicles and pedestrians, remains a substantial challenge. Existing methods often struggle to scale to large scenes or accurately model arbitrary dynamics. To address these limitations, we propose Hierarchy UGP, which constructs a hierarchical structure consisting of a root level, sub-scenes level, and primitive level, using Unified Gaussian Primitive (UGP) defined in 4D space as the representation. The root level serves as the entry point to the hierarchy. At the sub-scenes level, the scene is spatially divided into multiple sub-scenes, with various elements extracted. At the primitive level, each element is modeled with UGPs, and its global pose is controlled by a motion prior related to time. This hierarchical design greatly enhances the model's capacity, enabling it to model large-scale scenes. Additionally, our UGP allows for the reconstruction of both rigid and non-rigid dynamics. We conducted experiments on Dynamic City, our proprietary large-scale dynamic street scene dataset, as well as the public Waymo dataset. Experimental results demonstrate that our method achieves state-of-the-art performance. We plan to release the accompanying code and the Dynamic City dataset as open resources to further research within the community.

1. Introduction

The reconstruction of large-scale dynamic urban scenes plays a crucial role in various applications, such as autonomous driving [12, 16, 42], virtual reality (VR), and smart cities. The core objective of this task is to reconstruct large-scale scenes from long sequences of image frames while accurately modeling dynamic elements, such as vehicles and pedestrians. Traditional methods [1, 27, 28, 45] face significant challenges in rendering quality and handling dynamic elements.

Recently, several methods [4, 40, 49, 50] have introduced compositional 3D Gaussian representations to model

*Equal contribution †Corresponding author

dynamic street scenes, yielding impressive results. However, these approaches primarily focus on small-scale or fragmented scenes and exhibit limited performance when applied to large-scale environments. Furthermore, they encounter difficulties in accurately modeling the dynamic changes of complex non-rigid objects, such as pedestrians. Consequently, these methods still face significant challenges in addressing large-scale dynamic street scenes in real-world scenarios. Modeling such scenes requires representing static objects, dynamic rigid objects, and dynamic non-rigid objects with different Gaussian primitives. Large-scale scenes often involve tens of millions of Gaussian primitives, making it a challenging task to efficiently organize them within a unified framework while ensuring real-time rendering.

In this paper, we present Hierarchy UGP, a novel tree-structured Unified Gaussian Primitive, designed to address the challenges of reconstructing large-scale dynamic scenes from long image sequences. Specifically, our approach consists of three hierarchical levels: the root level, sub-scenes level, and primitive level, with the root level serving as the entry point for managing the entire hierarchy. We spatially partition the large-scale dynamic scene into multiple sub-scenes, forming the sub-scenes level. Within each sub-scene, we extract four types of elements—sky, background, rigid objects, and non-rigid objects. These elements are represented by UGPs at the primitive level, where each element is meticulously designed with distinct attributes specific to its characteristics, ensuring accurate modeling. Additionally, we introduce Level of Detail (LOD) techniques to enable real-time rendering of large-scale dynamic scenes. To the best of our knowledge, this is the first approach capable of reconstructing large-scale dynamic scenes with real-time rendering, while significantly enhancing accuracy of modeling non-rigid entities.

Our main contributions are summarized as follows:

- We introduce Hierarchy UGP, enabling high-fidelity and efficient reconstruction of large-scale dynamic scenes, along with real-time rendering.
- We propose a series of training strategies that address various challenges in reconstructing large-scale dynamic scenes.
- We collected the Dynamic City dataset, a large-scale dynamic scene dataset, and conducted experiments on both this dataset and the publicly available Waymo dataset, demonstrating the superiority of our approach over baseline methods. We plan to open-source the Dynamic City dataset and the accompanying code to support community development.

2. Related Works

2.1. Large Scale Scene Reconstruction

For decades, researchers have focused on 3D reconstruction of large-scale scenes. Early works [1, 9, 17, 23, 30, 51] applied the Structure-from-Motion (SfM) pipeline [28] to reconstruct such environments. However, these methods incurred substantial computational costs, requiring extensive time and resources, and suffered from error accumulation and scene drift, which compromised reconstruction quality.

In recent years, the rapid development of neural radiance fields [2, 8, 10, 21, 25, 35, 39, 48] has led to a paradigm shift in reconstruction techniques, enabling notable progress in large-scale scene reconstruction. [37] enhanced model representation with multi-level residual blocks, enabling city-scale reconstruction. However, this approach remains computationally demanding and is unsuitable for ground-level data. [34] and [32] addressed large-scale scenes by partitioning them into blocks, each represented by a multi-layer perceptron (MLP), but these methods also demand extensive training time and computational resources, with low rendering efficiency.

Recently, 3DGS advanced the field by using point-based differentiable rendering to achieve high-quality reconstruction and efficient rendering. [18] and [19] further enabled efficient, high-quality city-scale reconstructions using block Gaussians, while [26] and [15] achieved high-quality reconstruction and real-time rendering of large-scale scenes by leveraging hierarchical tree-structured Gaussians.

However, these methods are unable to effectively model complex dynamic objects in large-scale dynamic scenes, which limits the full representation of the scene.

2.2. Dynamic Scene Reconstruction

Dynamic scene reconstruction has been a long-term research task, and with the development of differentiable rendering techniques, especially the emergence of NeRF [21] and 3DGS [14], many excellent algorithms [3, 4, 6, 7, 13, 33, 40, 43, 44] have been developed to achieve impressive results.

Recently, some methods [6, 44] have introduced 4DGS to represent dynamic scenes, while others [3, 4, 40] have decoupled the scene into a static background and a dynamic foreground, using time-related spatial information priors to control the dynamic foreground. However, these approaches also have their limitations. For example, 4DGS [44] and 4D-Rotor GS [6] are limited to small-scale scenes, making them less suitable for complex urban dynamic environments. On the other hand, Street Gaussians [40] achieved high-fidelity reconstruction and real-time rendering by leveraging the 3DGS scene representation, but it struggles to model the subtle deformations of dynamic objects and cannot handle non-rigid objects. Similarly, PVG

[3] introduces Periodic Vibration Gaussians to represent dynamic scenes, achieving static-dynamic decomposition by categorizing Gaussians based on their lifespans. Furthermore, OmniRe [4] models pedestrians using the SMPL model [20], specifically designed for human modeling, and applies deformable-GS [36] to parts that SMPL cannot fit. While this approach achieves good results by controlling these parts using spatial information, it is limited in its general applicability, and the reconstruction quality of parts that SMPL cannot fit remains suboptimal.

3. Method

Given a sequence of images over a long temporal span captured from dynamic large-scale scenes, our goal is to achieve high-fidelity reconstruction and real-time rendering of large-scale scenes containing arbitrary dynamic elements. To this end, we propose a novel scene representation method, Hierarchy Unified Gaussian Primitive (Hierarchy UGP), which aims to efficiently represent large-scale dynamic scenes. The core of this method is the construction of a hierarchical UGP tree structure, providing an efficient representation for this complex mix of static and dynamic environments.

In this section, we first describe the structure of Hierarchy UGP in Sec.3.1. Then, Sec.3.2 discusses the rendering process of Hierarchy UGP. Finally, we introduce how to build Hierarchy UGP in Sec.3.3.

3.1. Hierarchy UGP

Previous methods face limitations in modeling large-scale dynamic scenes: those capable of handling large-scale environments struggle with dynamic elements, while methods focused on dynamic elements are generally restricted to small-scale or fragmented scenes, making it difficult to capture comprehensive street scenes. Inspired by prior work [15, 44], we propose Hierarchy UGP, which models large-scale scenes with arbitrary dynamic elements through hierarchy-structured Unified Gaussian Primitives (UGPs).

Our hierarchy structure consists of three levels: the **Root Level**, the **Sub-scenes Level**, and the **Primitive Level**. The root level abstracts the entire scene and serves as the entry point for managing the hierarchy. At the sub-scenes level, the large-scale scene is divided into multiple sub-scenes based on spatial information and various scene elements are extracted. Each sub-scene is reconstructed individually and merged at the root level. At the primitive level, we model various scene elements by UGPs with distinct properties.

Sub-scenes Level Large-scale scene reconstruction faces challenges in memory, speed, and model capacity, while typical scenes are easier to handle. To address this, we partition the large-scale scene into sub-scenes for parallel reconstruction.

Given thousands of images captured by calibrated cameras covering a large-scale scene, we begin by spatially defining the size of each sub-scene, which enables us to partition the scene into multiple sub-scenes. Subsequently, based on the spatial relationships between the camera poses and each sub-scene, we assign the corresponding image sequences to their respective sub-scenes.

We categorize the elements in each sub-scene into four types: background, sky, rigid objects, and non-rigid objects. During reconstruction, each element is modeled in its respective local coordinate system. When rendering, the elements are transformed into the global coordinate system, followed by joint optimization. The transformation of each element from the local to the global coordinate system is governed by a time-dependent motion prior:

$$M(t) = (R_t \mid T_t), \tag{1}$$

where R_t and T_t represent the rotation matrix and translation vector of the element in the global coordinate system at time t, respectively. For any time t, the motion for the background and sky is defined as $M(t) = (I \mid O)$, while for rigid and non-rigid objects, R_t and T_t can be readily obtained using an off-the-shelf tracker [24, 41]. By decoupling various elements within sub-scenes, we enable more refined modeling at the primitive level.

Primitive Level Inspired by [6, 44], we model the elements within a sub-scene as Unified Gaussian Primitives defined in 4D space:

$$\mathcal{G} = (\mu, \Sigma, o, \mathbf{SH}) \in \mathbb{R}^4 \times \mathbb{S}^4_{++} \times [0,1] \times \mathbb{R}^{16}, \tag{2}$$

where $\mu = (\mu_x, \mu_y, \mu_z, \mu_t)$ is the 4D mean vector, representing the spatial-temporal position of the primitive. Σ is the 4D covariance matrix, capturing the spatial-temporal extent. o is the opacity of the primitive, controlling its visibility during rendering. $\mathbf{SH}$ denotes the spherical harmonic coefficients, encoding the appearance properties of the primitive.

The covariance matrix Σ is parameterized to represent a 4D ellipsoid as follows:

$$\Sigma = RSS^TR^T \tag{3}$$

Here, $S = diag(s_x, s_y, s_z, s_t)$ is a diagonal matrix representing the scale of the UGP in 4D space, and the 4D rotation is parameterized by two isotropic quaternions, $q_l = (a, b, c, d)$ and $q_r = (p, q, r, s)$ [44] and constructed as follows:

$$\begin{aligned} R &= L(q_l)\, R(q_r) \\ &= \begin{pmatrix} a & -b & -c & -d \\ b & a & -d & c \\ c & d & a & -b \\ d & -c & b & a \end{pmatrix} \begin{pmatrix} p & -q & -r & -s \\ q & p & s & -r \\ r & -s & p & q \\ s & r & -q & p \end{pmatrix} \end{aligned} \tag{4}$$

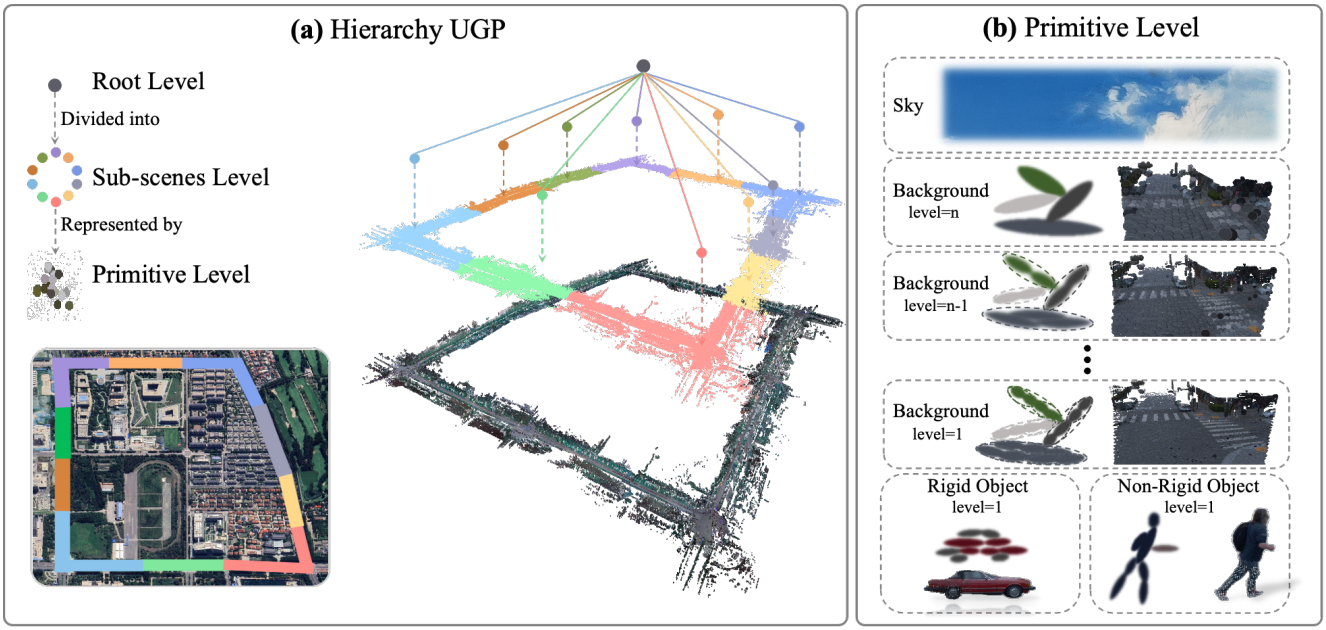


Figure 2. **Method Overview.** The large-scale dynamic scene is constructed as a hierarchical tree structure, where both static and dynamic elements are represented using Unified Gaussian Primitives, enabling efficient large-scale dynamic scene reconstruction. As shown in (a), the scene hierarchy consists of the Root Level, Sub-scenes Level, and Primitive Level. The Root Level serves as the entry point for managing the entire structure. At the Sub-scenes Level, the scene is spatially divided into multiple sub-scenes, which are further categorized into Sky, Background, Rigid Objects, and Non-Rigid Objects. As depicted in (b), at the Primitive Level, each element is modeled using Unified Gaussian Primitives with distinct properties.

Based on the characteristics of different types of elements in a sub-scene, we design UGPs with different properties. Specifically, since the temporal scale s_t of the UGP controls its temporal influence range, for non-rigid objects, we initialize s_t based on the frame rate of the data collection:

$$s_t = -\frac{(\lambda v \mathrm{d}t)^2}{2log(o_{th})}, \tag{5}$$

where, λ is a hyperparameter used to fine-tune the initial s_t, v is the image acquisition frequency, $\mathrm{d}t$ is the time duration corresponding to each frame, and o_{th} is the opacity threshold for pruning.

For background and rigid objects, whose positions μ in the local coordinate system do not change over time, we set μ_t in the 4D mean vector to 0. We also set the 4D rotation quaternion q_r to (1, 0, 0, 0) and set s_t to ∞, indicating that their shape remains unchanged over time. For the sky element [15], we build upon these settings and optimize only their opacity o and spherical harmonic coefficients $\mathbf{SH}$.

3.2. Rendering Process

In this subsection, we describe the rendering process of Hierarchy UGP, which comprises three primary stages: spatial-temporal projection, primitives selection, and image synthesis.

Spatial-Temporal Projection To render Hierarchy UGP, we need to perform spatial-temporal projection on the UGPs modeling each element. Specifically, given a time t, we first obtain the conditional 3D mean and covariance of the UGPs at time t through a slicing process [6, 44]:

$$\begin{aligned} \mu_{xyz|t} &= \mu_{1:3} + \Sigma_{1:3,4}\Sigma_{4,4}^{-1}(t - \mu_t) \\ \Sigma_{xyz|t} &= \Sigma_{1:3,1:3} - \Sigma_{1:3,4}\Sigma_{4,4}^{-1}\Sigma_{4,1:3} \end{aligned} \tag{6}$$

After slicing the UGP into 3D space, we use the motion prior of each element at that time step, $M(t) = (R_t \mid T_t)$, to transform the UGPs from the local coordinate system to the global coordinate system:

$$\begin{aligned} \Sigma_{\text{global}} &= R_t \Sigma_{xyz|t} R_t^T \\ \mu_{\text{global}} &= R_t \mu_{xyz|t} + T_t \end{aligned} \tag{7}$$

Primitives Selection After performing spatial-temporal projection, we select the appropriate UGPs for rendering based on the given viewpoint, rather than directly rendering all UGPs. This process accelerates the rendering speed, enabling real-time rendering of large-scale scenes. Specifically, starting from the root node of the Hierarchy UGP, we

iteratively select UGPs at each level until sufficiently accurate UGPs are chosen to render the image.

To achieve this, we set a threshold value τ and begin the selection process from the root node of the hierarchy [15, 19, 29]. For each UGP, we compute its diameter on the image plane. We then check if the diameter is smaller than τ or if the primitive has no child nodes. If neither condition is met, we proceed to evaluate the child nodes of the current UGP, and repeat the assessment process. This iterative selection continues until either all UGPs meet the condition of having a diameter smaller than the threshold τ, or no further child nodes are available.

To calculate the diameter of a UGP on the image plane, we first project its covariance matrix onto the 2D image plane:

$$\Sigma' = JW\Sigma_{\text{global}}W^T J^T, d = 2\sqrt{\lambda_{\max}(\Sigma')}, \tag{8}$$

where W is the view transformation matrix and J is the Jacobian matrix representing the affine approximation of the projection transformation. The projected covariance matrix Σ' defines an ellipse on the image plane, with d, the diameter of the UGP, determined by the length of the major axis of this ellipse. Additionally, $\lambda_{\max}(\Sigma')$ represents the largest eigenvalue of the covariance matrix Σ'.

Image Synthesis After computing the color c of each UGP using spherical harmonics (**SH**), we render the image $\mathcal{I}$ at time t using differentiable alpha blending:

$$\mathcal{I}(t) = \sum_{i=1}^{N} \alpha_{t|i} c_i \prod_{j=1}^{i-1} \left(1 - \alpha_{t|j}\right), \tag{9}$$

where α_t is the opacity weighted at time t, based on the probability of each UGP's presence at that moment:

$$\alpha_t = exp(\frac{1}{2}\Sigma_{4,4}^{-1}(t-\mu_t)^2)\alpha \tag{10}$$

3.3. Building the Hierarchy UGP

In this subsection, we outline the process of building the Hierarchy UGP, which is divided into three stages: initialization of the global UGP scaffold, optimization of sub-scenes level, and the final merging process.

Initialization We start by initializing a UGP model using LiDAR point clouds from large-scale dynamic scenes. Through coarse training, we construct a global UGP scaffold. During sub-scene initialization, the global UGP scaffold distributes global Gaussians to the sub-scenes based on spatial partitions, ensuring consistent training of all sub-scenes.

Optimization Subsequently, we perform training on each sub-scene individually. Leveraging the motion prior of each element, we model the elements within each sub-scene in local space and then transform them into global space for joint optimization.

During optimization, we use a Block-wise Objects training strategy for dynamic objects that span multiple sub-scenes, avoiding interference between sub-scenes. To improve the reconstruction quality of non-rigid objects, we perform the temporal scale s_t initialization, as shown in Equation 5, to enhance fitting in areas with large motion amplitudes.

We use the following objective function to optimize the sub-scenes:

$$\mathcal{L} = \lambda_{\text{rgb}}\mathcal{L}_{\text{rgb}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \lambda_{\text{warp}}\mathcal{L}_{\text{warp}}, \tag{11}$$

where $\mathcal{L}_{\text{rgb}}$ combines the L1 and SSIM losses, $\mathcal{L}_{\text{depth}}$ is the L1 loss between rendered depth and the depth generated by projecting sparse LiDAR points onto the camera plane, and $\mathcal{L}_{\text{warp}}$ is the L1 loss between rendered image and the virtual warping view [5]. For more details, please refer to the supplementary materials.

Hierarchy Construction Drawing inspiration from [15, 19, 26], we organize the UGPs into a BVH (Bounding Volume Hierarchy) tree and compute UGP attributes for the intermediate nodes. Specifically, we recursively perform spatial binary median splits on the UGPs until each UGP is assigned to a leaf node. Once the BVH tree is constructed, we begin with the leaf nodes, which are at level 1, and recursively merge the UGP attributes of the child nodes in a bottom-up manner.

Higher-level UGPs exhibit larger three-dimensional volumes, resulting in a coarser representation of the scene. In contrast, lower-level UGPs have smaller three-dimensional volumes, enabling a more refined representation. We obtain the attributes of the l level UGPs by interpolating the attributes of the $l-1$ level UGPs based on elaborately designed weights w [15]:

$$\mathcal{G}^l(\mu, \Sigma, o, \mathbf{SH}) = \sum_{i}^{N} w_i \mathcal{G}^{l-1}(\mu, \Sigma, o, \mathbf{SH}) \tag{12}$$

Since dynamic objects generally occupy limited space within the scene, we empirically set them as leaf nodes to achieve a better trade-off between rendering quality and speed.

Merging at the Sub-scenes Level After constructing the hierarchy for all sub-scenes, we merge them into the root level. For each sub-scene, we load the UGPs and filter out unnecessary ones based on their distance from the center

Method	Large 001				Large 002				Sub-scene 001				Sub-scene 002			
	PSNR↑	SSIM↑	LPIPS↓	FPS↑	PSNR↑	SSIM↑	LPIPS↓	FPS↑	PSNR↑	SSIM↑	LPIPS↓	FPS↑	PSNR↑	SSIM↑	LPIPS↓	FPS↑
4D-GS [44]	26.30	0.844	0.277	36	24.59	0.826	0.285	30	27.11	0.892	0.156	50	25.51	0.859	0.194	45
Hierarchical GS [15]	26.74	0.861	0.188	28	25.14	0.843	0.202	15	26.80	0.892	0.179	90	25.94	0.851	0.187	30
Street GS [40]	25.59	0.834	0.285	19	23.94	0.814	0.301	21	25.94	0.864	0.210	33	26.18	0.833	0.220	22
OmniRe [4]	24.48	0.776	0.336	19	22.17	0.734	0.386	20	26.24	0.839	0.146	15	24.08	0.763	0.171	13
Ours	27.01	0.862	0.180	32	25.25	0.844	0.198	21	27.81	0.894	0.150	102	27.61	0.870	0.168	44

Table 1. **Quantitative Comparison on Dynamic City.** We selected every 10th frame as a test frame and computed visual quality metrics and frames per second (FPS) for our method compared to previous work on two large scenes and two sub-scenes. For each large scene, we loaded the entire hierarchical structure and used an LOD level of $\tau = 9$. For each sub-scene, we used an LOD level of $\tau = 1$. Each cell is colored to indicate the best and second best.

(a) GT (b) Ours (c) OmniRe (d) Street GS (e) 4D-GS (f) Hierarchical GS

Figure 3. **Interpolation Comparison on the Dynamic City Dataset.** We conducted qualitative experiments on the Dynamic City Dataset by selecting every 10th frame as a test frame. The results show that our method significantly outperforms others in terms of reconstruction quality, achieving high-quality reconstructions of large-scale dynamic scenes.

of the sub-scene. The UGPs are then sequentially merged at the root level. Using a partitioning strategy with 50% spatial overlap between adjacent sub-scenes, we seamlessly integrate the sub-scenes [15, 18, 32]. The global UGP scaffold ensures consistent training of the sub-scenes, which allows the resulting merged scene to maintain visual continuity without noticeable boundary artifacts.

4. Experiment

In this section, we first present our implementation details in Sec.4.1. Next, we provide metric evaluations comparing several state-of-the-art methods on the Dynamic City Dataset (Sec.4.2) and the Waymo Open Dataset [31] (Sec.4.3). We use PSNR, SSIM, and LPIPS [47] to evaluate the visual quality of interpolation experiments, FID [11] for extrapolation experiments, and measure the rendering FPS. Ablation studies on specific strategies from the paper are discussed in Sec.4.4.

4.1. Implementation Details

We implemented our method using PyTorch [22] and custom CUDA kernels. Experiments on the Dynamic City Dataset were run on H20 GPUs, completing parallel training of multiple sub-scenes within 3 hours. For the Waymo Dataset, experiments on a single 4090 GPU completed training in 2 hours. See supplementary material for details.

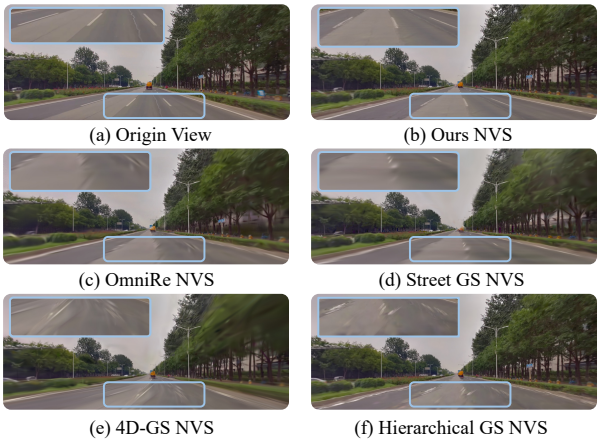
4.2. Dynamic City Dataset

Currently, large-scale dynamic street scene datasets are not publicly available. To address this gap, we introduce the Dynamic City Dataset, which comprises sequences of image and radar data captured at a frequency of 10 Hz, covering street scenes ranging from 600 meters to over one kilometer. Compared to publicly available datasets like Waymo [31] and PandaSet [38], the Dynamic City Dataset includes a broader range of street scenes. We intend to release this dataset as an open resource to advance research in large-scale dynamic street scene reconstruction.

To demonstrate the efficacy of our algorithm and ensure

Methods	Scene Reconstruction						Novel View Synthesis					
	014			023			014			023		
	PSNR*↑	SSIM*↑	PSNR↑	PSNR*↑	SSIM*↑	PSNR↑	PSNR*↑	SSIM*↑	PSNR↑	PSNR*↑	SSIM*↑	PSNR↑
4D-GS [44]	19.97	0.575	30.86	19.40	0.622	30.00	18.95	0.486	30.24	18.32	0.590	28.19
Hierarchical GS [15]	16.02	0.400	30.74	13.58	0.432	24.65	15.93	0.369	30.58	13.42	0.430	24.34
Street GS [40]	23.33	0.719	34.01	24.75	0.815	32.14	21.90	0.637	33.34	21.79	0.688	30.75
OmniRe [4]	22.16	0.678	32.93	29.23	0.862	36.40	19.61	0.493	30.67	23.07	0.724	32.56
Ours	30.20	0.889	36.10	34.10	0.911	37.16	24.23	0.665	31.69	21.64	0.597	31.27

Table 2. **Quantitative Comparison on Waymo.** We selected every 10th frame as a test frame and computed visual quality metrics, where * denotes the metrics for the pedestrian regions. Each cell is colored to indicate the best and second best.



(a) Origin View (b) Ours NVS (c) OmniRe NVS (d) Street GS NVS (e) 4D-GS NVS (f) Hierarchical GS NVS

Figure 4. **Extrapolation Comparison on the Dynamic City Dataset.** We shifted the original viewpoint by 2 meters to the left to evaluate extrapolation performance. The results show that our method significantly outperforms others in extrapolation.

Methods	FID↓ Lane Shift@2m			
	Large 001	Large 002	Sub-scene 001	Sub-scene 002
4D-GS [44]	77.83	84.60	108.39	68.14
Hierarchical GS [15]	78.73	70.64	76.63	65.81
Street GS [40]	90.99	79.67	65.56	63.57
OmniRe [4]	90.35	119.96	68.83	58.11
Ours	**68.27**	**63.36**	**51.59**	**56.30**

Table 3. **Extrapolation Comparison on Dynamic City.** We conducted an extrapolation comparison on the Dynamic City dataset by calculating the FID, and the results show that our method consistently outperforms others.

a fair comparison, we conducted two experiments: one with large-scale dynamic scenes and the other with sub-scenes extracted from large-scale scenes.

The results presented in Table 1 and Figure 3 demonstrate that, for interpolation tasks, our method consistently outperforms existing approaches in both large-scale dynamic scenes and smaller sub-scenes. Moreover, by integrating LOD techniques and engineering enhancements based on gsplat [46], our method enables real-time rendering for large-scale dynamic scenes.

Furthermore, the results in Table 3 and Figure 4 show that, for extrapolation tasks, our method significantly outperforms all other baseline methods. This improvement is primarily attributed to the supervision from virtual warping views. Additional experimental results and a more detailed discussion of virtual warping views can be found in the supplementary material.

4.3. Waymo Dataset

Waymo [31] is a real-world dataset comprising thousands of driving segments collected on actual roads. Each segment contains 20 seconds of sensor data sampled at 10 Hz.

The quantitative results in Table 2 and the qualitative results in Figure 5 demonstrate that our method outperforms others in terms of reconstruction. Specifically, our approach achieves significantly higher visual metrics for human objects, highlighting its strong capability to model dynamic elements, even in areas with large motion.

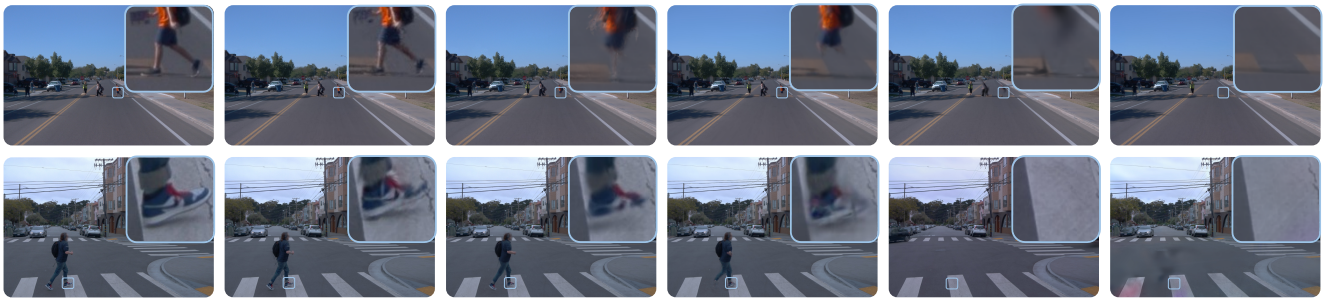
Additionally, the results in Table 2 show that both our method and OmniRe [4] experience more pronounced degradation in visual quality on the test frames. This degradation occurs because, in these frames, pedestrian movement is completely unknown, and neither UGP nor SMPL can fully capture the pedestrian's motion, leading to a noticeable performance drop. Nevertheless, our method remains competitive with state-of-the-art approaches.

Methods	PSNR↑	$PSNR^{+}$ ↑
w/o Block-wise Objects	28.58	21.51
w/ Block-wise Objects	**28.64**	**23.15**

Table 4. **Ablation on Block-wise Object training strategy.** Here, $^{+}$ denotes the metrics for the vehicle regions. The results show that our Block-wise Object training strategy significantly improves the visual metrics for moving objects.

4.4. Ablation Study and Analysis

Block-wise Objects training strategy Table 4 and Figure 6 show the qualitative and quantitative results of our ablation study on the Block-wise Objects training strategy.

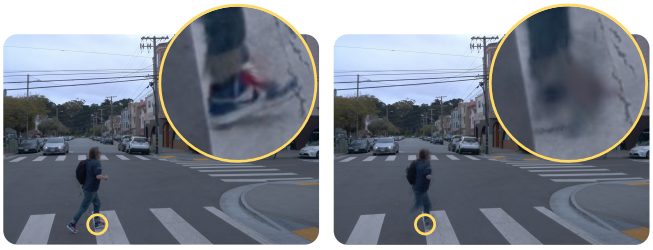


(a) GT (b) Ours (c) OmniRe (d) Street GS (e) Hierarchical GS (f) 4D-GS

Figure 5. **Qualitative Comparison on Waymo Dataset.** Our method significantly improves performance in large-motion dynamic areas, such as the feet and legs of pedestrians, compared to others.

w/ Block-wise Objects w/o Block-wise Objects

Figure 6. **Ablation on Block-wise Objects training strategy.** We emphasize the impact of our Block-wise Objects training strategy on vehicles to showcase its effectiveness.



w/ Temporal Scale Initialization w/o Temporal Scale Initialization

Figure 7. **Ablation on Temporal Scale Initialization.** We emphasize the impact of Temporal Scale Initialization on the large-motion components of pedestrian modeling to visually demonstrate its effectiveness.

More details about this strategy are provided in the supplementary material.

Methods	PSNR↑	SSIM↑
w/o Temporal Scale Initialization	23.22	0.791
w/ Temporal Scale Initialization	**34.93**	**0.943**

Table 5. **Ablation on Temporal Scale Initialization.** We conducted an ablation study on Temporal Scale s_t Initialization in the Waymo dataset and calculated the visual metrics for humans.

Temporal Scale Initialization Table 5 and Figure 7 show the results of our ablation study. The temporal scale s_t initialization, as shown in Equation 5. We found that the best practice is to initialize s_t based on the frame rate of the data collection, which has a significant impact on the results.

NVS w/ Virtual Warping View NVS w/o Virtual Warping View

Figure 8. **Ablation on Virtual Warping View.** The figure demonstrates that supervision by the virtual warping view can effectively improve the visual quality of the novel view (with the original viewpoint shifted by 2 meters).

Virtual Warping View We shifted the original viewpoint 2 meters to the left and conducted an ablation study on the supervision of virtual warping views for novel views. Figure 8 demonstrates its effectiveness. For more details, please refer to the supplementary material.

5. Conclusion

This paper introduced Hierarchy UGP, a hierarchical model that enhances representation capacity for large-scale dynamic scenes and enables real-time rendering. It uses Unified Gaussian Primitives to represent static and dynamic elements. We demonstrate the state-of-the-art performance of our method on our own collected Dynamic City dataset and the Waymo dataset, validating its effectiveness through ablation studies. We plan to open-source code and the Dynamic City dataset to promote community development.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54 (10):105–112, 2011. 1, 2

[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. 2

[3] 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. 2, 3

[4] Ziyu Chen, Jiawei Yang, Jiahui Huang, Riccardo de Lutio, Janick Martinez Esturo, Boris Ivanovic, Or Litany, Zan Gojcic, Sanja Fidler, Marco Pavone, et al. Omnire: Omni urban scene reconstruction. *arXiv preprint arXiv:2408.16760*, 2024. 1, 2, 3, 6, 7

[5] Kai Cheng, Xiaoxiao Long, Wei Yin, Jin Wang, Zhiqiang Wu, Yuexin Ma, Kaixuan Wang, Xiaozhi Chen, and Xuejin Chen. Uc-nerf: Neural radiance field for under-calibrated multi-view cameras in autonomous driving. *arXiv preprint arXiv:2311.16945*, 2023. 5

[6] Yuanxing Duan, Fangyin Wei, Qiyu Dai, Yuhang He, Wenzheng Chen, and Baoquan Chen. 4d-rotor gaussian splatting: Towards efficient novel view synthesis for dynamic scenes. In *Proc. SIGGRAPH*, 2024. 2, 3, 4

[7] Tobias Fischer, Jonas Kulhanek, Samuel Rota Bulò, Lorenzo Porzi, Marc Pollefeys, and Peter Kontschieder. Dynamic 3d gaussian fields for urban areas. *arXiv preprint arXiv:2406.03175*, 2024. 2

[8] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong 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. 2

[9] Christian Früh and Avideh Zakhor. An automated method for large-scale, ground-based city model acquisition. *International Journal of Computer Vision*, 60:5–24, 2004. 2

[10] Jianfei Guo, Nianchen Deng, Xinyang Li, Yeqi Bai, Botian Shi, Chiyu Wang, Chenjing Ding, Dongliang Wang, and Yikang Li. Streetsurf: Extending multi-view implicit surface reconstruction to street views. *arXiv preprint arXiv:2306.04988*, 2023. 2

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 6

[12] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862, 2023. 1

[13] Nan Huang, Xiaobao Wei, Wenzhao Zheng, Pengju An, Ming Lu, Wei Zhan, Masayoshi Tomizuka, Kurt Keutzer, and Shanghang Zhang. S3gaussian: Self-supervised street gaussians for autonomous driving. *arXiv preprint arXiv:2405.20323*, 2024. 2

[14] 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

[15] Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics (TOG)*, 43(4):1–15, 2024. 2, 3, 4, 5, 6, 7

[16] Wei Li, CW Pan, Rong Zhang, JP Ren, YX Ma, Jin Fang, FL Yan, QC Geng, XY Huang, HJ Gong, et al. Aads: Augmented autonomous driving simulation using data-driven algorithms. *Science robotics*, 4(28):eaaw0863, 2019. 1

[17] Xiaowei Li, Changchang Wu, Christopher Zach, Svetlana Lazebnik, and Jan-Michael Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part I 10*, pages 427–440. Springer, 2008. 2

[18] Jiaqi Lin, Zhihao Li, Xiao Tang, Jianzhuang Liu, Shiyong Liu, Jiayue Liu, Yangdi Lu, Xiaofei Wu, Songcen Xu, Youliang Yan, et al. Vastgaussian: Vast 3d gaussians for large scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5166–5175, 2024. 2, 6

[19] Yang Liu, Chuanchen Luo, Lue Fan, Naiyan Wang, Junran Peng, and Zhaoxiang Zhang. Citygaussian: Real-time high-quality large-scale scene rendering with gaussians. In *European Conference on Computer Vision*, pages 265–282. Springer, 2025. 2, 5

[20] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023. 3

[21] 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. 2

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 6

[23] Marc Pollefeys, David Nistér, J-M Frahm, Amir Akbarzadeh, Philippos Mordohai, Brian Clipp, Chris Engels, David Gallup, S-J Kim, Paul Merrell, et al. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*, 78:143–167, 2008. 2

[24] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junt-

ing Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 3

[25] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12932–12942, 2022. 2

[26] Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *arXiv preprint arXiv:2403.17898*, 2024. 2, 5

[27] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1352–1359, 2013. 1

[28] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1, 2

[29] Qing Shuai, Haoyu Guo, Zhen Xu, Haotong Lin, Sida Peng, Hujun Bao, and Xiaowei Zhou. Real-time view synthesis for large scenes with millions of square meters. In *CVPR*, 2024. 5

[30] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 2006. 2

[31] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 6, 7

[32] 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. 2, 6

[33] Fischer Tobias, Porzi Lorenzo, Rota Bulò Samuel, Pollefeys Marc, and Kontschieder Peter. Multi-level neural scene graphs for dynamic urban environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[34] 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. 2

[35] Zian Wang, Tianchang Shen, Jun Gao, Shengyu Huang, Jacob Munkberg, Jon Hasselgren, Zan Gojcic, Wenzheng Chen, and Sanja Fidler. Neural fields meet explicit geometric representations for inverse rendering of urban scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8370–8380, 2023. 2

[36] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024. 3

[37] Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In *European conference on computer vision*, pages 106–122. Springer, 2022. 2

[38] 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. 6

[39] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5438–5448, 2022. 2

[40] Yunzhi Yan, Haotong Lin, Chenxu Zhou, Weijie Wang, Haiyang Sun, Kun Zhan, Xianpeng Lang, Xiaowei Zhou, and Sida Peng. Street gaussians for modeling dynamic urban scenes. In *ECCV*, 2024. 1, 2, 6, 7

[41] Cheng-Yen Yang, Hsiang-Wei Huang, Wenhao Chai, Zhongyu Jiang, and Jenq-Neng Hwang. Samurai: Adapting segment anything model for zero-shot visual tracking with motion-aware memory, 2024. 3

[42] Zhenpei Yang, Yuning Chai, Dragomir Anguelov, Yin Zhou, Pei Sun, Dumitru Erhan, Sean Rafferty, and Henrik Kretzschmar. Surfelgan: Synthesizing realistic sensor data for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11118–11127, 2020. 1

[43] 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. 2

[44] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. In *International Conference on Learning Representations (ICLR)*, 2024. 2, 3, 4, 6, 7

[45] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018. 1

[46] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for Gaussian splatting. *arXiv preprint arXiv:2409.06765*, 2024. 7

[47] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the*

IEEE conference on computer vision and pattern recognition, pages 586–595, 2018. 6

[48] MI Zhenxing and Dan Xu. Switch-nerf: Learning scene decomposition with mixture of experts for large-scale neural radiance fields. In *The Eleventh International Conference on Learning Representations*, 2022. 2

[49] Hongyu Zhou, Jiahao Shao, Lu Xu, Dongfeng Bai, Weichao Qiu, Bingbing Liu, Yue Wang, Andreas Geiger, and Yiyi Liao. Hugs: Holistic urban 3d scene understanding via gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21336–21345, 2024. 1

[50] 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. 1

[51] Siyu Zhu, Runze Zhang, Lei Zhou, Tianwei Shen, Tian Fang, Ping Tan, and Long Quan. Very large-scale global sfm by distributed motion averaging. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4568–4577, 2018. 2