HDGS: Hierarchical Dynamic Gaussian Splatting for Urban Driving Scenes

Fudong Ge[1,2,5*], **Jin Gao**[1,2,5†], **Hanshi Wang**[1,2,5], **Yiwei Zhang**[1,2,5], **Ke Wang**[7], **Weiming Hu**[1,2,5,6], **Zhipeng Zhang**[3,4†]

[1]State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS), CASIA
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]AutoLab, School of Artificial Intelligence, Shanghai Jiao Tong University
[4]Anyverse Robotics
[5]Beijing Key Laboratory of Super Intelligent Security of Multi-Modal Information
[6]School of Information Science and Technology, ShanghaiTech University
[7]KargoBot
{fudong.ge.cv, zhipeng.zhang.cv}@outlook.com, jin.gao@nlpr.ia.ac.cn

Abstract

This paper tackles the challenging task of achieving storage-efficient yet high-fidelity motion representation in large-scale dynamic 3D Gaussian Splatting. Our motivation stems from the truth that existing urban-scale methods, which rely on massive and unstructured individual Gaussians for scene modeling, face a critical scalability bottleneck. Inspired by recent advances in the 3DGS-based compression beyond autonomous driving, we address this challenge by leveraging the compression capability of anchor-driven methods. However, this is non-trivial as our exploratory experiments reveal that the direct application of this paradigm to dynamic, large-scale urban scenes results in performance degradation. We attribute this phenomenon to the hierarchical anchor design that severely loses dynamic information. To this end, we propose Hierarchical Dynamic Gaussian Splatting (HDGS), a novel framework designed to adapt the anchor-based Gaussian paradigm to 4D urban environments. We first establish a local support network to reinforce inter-anchor consistency, mitigating geometric and appearance fractures caused by supervision attenuation in deep hierarchies. Then, we handle heterogeneous object motion via coarse-to-fine decomposition, where high-level anchors model coarse dynamics and low-level anchors refine them with residual deformations. Third, we introduce a hybrid supervision scheme that fuses global geometric constraints and local pixel-level cues to alleviate geometrically inconsistent reconstruction under sparse LiDAR. Extensive experiments show that HDGS reduces storage by 69.0% while maintaining or even improving rendering fidelity compared to state-of-the-art methods.

Code — https://github.com/AutoLab-SAI-SJTU/HDGS

1 Introduction

Effective validation of end-to-end autonomous driving systems (*e.g.*, perception (Li et al. 2024; Zhang et al. 2025b; Ge et al. 2024)), hinges on realistic closed-loop simulation, where the ego vehicle dynamically diverges from the original data-logging trajectory, creating a critical demand for generating physically accurate sensor observations from novel viewpoints. This requirement for high-fidelity novel view synthesis has been a primary bottleneck, as inaccuracies introduce a detrimental "sim-to-real" gap. The advent of 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) addresses this challenge. While celebrated for reconstruction, its paramount contribution in this context lies in enabling high-quality, real-time novel view synthesis. By faithfully preserving real-world appearance and geometry from any viewpoint, 3DGS enables the creation of scalable simulation environments where end-to-end systems can be tested in a truly interactive and realistic manner (Chen et al. 2025).

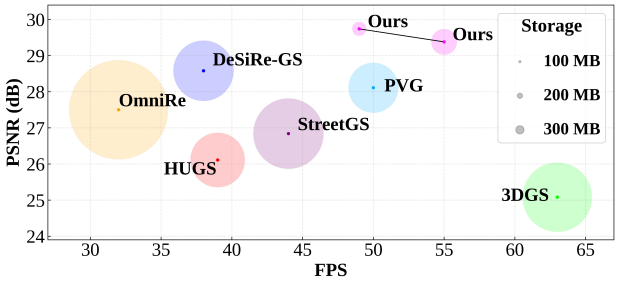

Figure 1: Quantitative comparison of 4D reconstruction methods on novel view synthesis over the Waymo dataset.

To evolve 3DGS into a practical simulator of autonomous driving, two requirements must be satisfied: **(I)** the scalability to represent vast, diverse driving scenes, and **(II)** the fidelity to model the complex motions. While recent efforts have focused on the latter by adapting 3DGS for dynamic objects (Zhou et al. 2024b; Yan et al. 2024), they inherit a core limitation from the original framework, where they model scenes with a massive, unstructured collection of individual Gaussians. This design severely restricts scalability. For instance, storing 100 hours of 10Hz, high-resolution driving data with a method like DeSiRe-GS (Peng et al. 2025) could demand an impractical ∼100TB of storage.

*Work done co-mentored by Prof. Zhipeng Zhang.
†Corresponding author.

Model	Scene	PSNR	SSIM	LPIPS	Stor. (MB)
3DGS	static &	24.89	-	-	1606
Scaffold-GS	outdoor	**27.01**	-	-	**203**
3DGS	dynamic	27.99	0.866	0.293	1023
Scaffold-GS	&	27.34	0.842	0.320	**394**
DesiRe-GS	urban	**32.88**	**0.901**	**0.203**	917

Table 1: Comparison of different methods. The anchor-based Scaffold-GS targets static scenes. DeSiRe-GS is designed for dynamic urban scenes. Stor. means Storage.

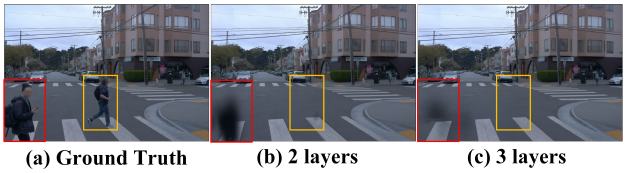

Figure 2: Visualization of spatio-temporal information loss induced by direct hierarchical compression. As the hierarchy deepens, dynamic objects gradually become less visible.

Upon revisiting the research beyond autonomous driving, we observe promising advancements in 3DGS-based compression techniques (Wang et al. 2024b; Fan et al. 2025; Morgenstern et al. 2024). A representative paradigm is the anchor-based approach (Lu et al. 2024). However, the direct application of this paradigm to dynamic, large-scale urban scenes results in performance degradation (see Tab. 1 anchor-based Scaffold-GS $v.s.$ 3DGS/DeSiRe-GS). Our analysis, supported by visualizations in Fig. 2, reveals that this degradation stems from a critical loss of information about dynamics. More specifically, the core issue lies in the hierarchical anchor design. As the hierarchy deepens, high-level anchors become increasingly sparse and cover large receptive fields, yet they lack any explicit motion model. Optimized solely through a reconstruction loss on static views, these anchors struggle with the spatio-temporal inconsistencies of moving objects, which yield unstable supervision and noisy gradients. To minimize training loss, the model learns to suppress these difficult-to-reconstruct dynamic regions, causing the high-level anchors to primarily represent the static background. Consequently, dynamic objects are poorly reconstructed, appearing blurred or disappearing entirely from the final rendering. This leads to a question: *Is it possible to preserve the realism of complex object motions in urban-scale driving scenes while maintaining the storage efficiency required for practical deployment?*

We validate this conjecture by introducing **HDGS** (**H**ierarchical **D**ynamic **G**aussian **S**platting), a novel framework designed to adapt the anchor-based Gaussian paradigm to complex 4D urban scenes. As previously discussed, the primary obstacle in extending prior anchor-based methods from static to dynamic scenes is their inherent inability to model motion, which stems from three fundamental challenges. **First**, **dynamics heterogeneity**: These models fail to capture the heterogeneous motion of diverse objects found in dynamic urban driving scenes. Our framework tackles this with a coarse-to-fine motion decomposition, where high-level anchors model coarse, rigid dynamics, and lower-level anchors refine these with residual, non-rigid deformations (see Fig. 8). Each level is conditioned on the motion prior from the level above, and features are propagated consistently to ensure a coherent, motion-aware representation. **Second**, **inter-anchor inconsistency**: The attenuation of supervisory signals in deep hierarchies prevents uppermost anchors from learning structural consistency, leading to fractures in geometry and appearance (see Fig. 6). To address this, we integrate sub-manifold sparse convolutions at the top levels, creating a local support network that reinforces inter-anchor consistency, with contributions adaptively weighted by positional uncertainty derived from covariance. **Third**, **geometric inconsistency**: The ambiguity from sparse geometric supervision in LiDAR data leads to reconstruction errors caused by geometric inconsistencies (see Fig. 7). To overcome this, we introduce a hybrid supervision scheme that synergistically fuses global geometric constraints and local pixel-level cues. This multi-faceted approach regularizes the optimization, yielding reconstructions that are both sharper and more geometrically faithful.

With the proposed modules integrated, HDGS achieves strong performance on Waymo (Sun et al. 2020) and KITTI (Geiger, Lenz, and Urtasun 2012). Compared to the recent state-of-the-art method DeSiRe-GS (Peng et al. 2025), HDGS reduces storage by 69.0% while improving all rendering fidelity metrics (*e.g.*, PSNR 1.16dB↑).

Our main contributions are as follows: ♠ We propose HDGS, a novel framework that adapts the anchor-based Gaussian paradigm to urban scenes, enabling efficient storage and dynamic modeling. ◆ We address three key challenges: reinforcing inter-anchor consistency via a local support network, modeling heterogeneous dynamics through coarse-to-fine motion decomposition, and mitigating reconstruction errors caused by geometric inconsistencies under sparse LiDAR via hybrid supervision. ♣ Extensive experiments conducted on two large-scale benchmark datasets (Waymo and KITTI) demonstrate that our method achieves efficient compression while maintaining rendering fidelity.

2 Related Work

2.1 Urban Scene Reconstruction

Urban scene reconstruction has seen substantial progress with the emergence of neural representations (Mildenhall et al. 2021; Barron et al. 2022, 2021), notably Neural Radiance Fields (NeRF) (Mildenhall et al. 2021) and 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023). Extensive work (Nguyen et al. 2024; Rematas et al. 2022; Tancik et al. 2022; Turki, Ramanan, and Satyanarayanan 2022) has integrated NeRF into autonomous driving for high-fidelity scene representation. In comparison, 3DGS-based methods (Peng et al. 2025; Zhou et al. 2024a; Huang et al. 2024; Yan et al. 2024) achieve a better trade-off between rendering quality and speed. DrivingGaussian (Zhou et al. 2024b) introduces incremental static 3D Gaussians for static objects and a dynamic Gaussian graph to model relationships among dynamic objects. OmniRe (Chen et al. 2025) enhances scene

graphs with non-rigid nodes for better modeling of pedestrians and cyclists. Recently, DeSiRe-GS (Peng et al. 2025) employs a two-stage training pipeline to enable effective static-dynamic decomposition and high-fidelity surface reconstruction. However, these methods neglect storage efficiency, hindering large-scale deployment. Therefore, incorporating strategies such as structural compression, redundancy reduction is essential for scalable reconstruction.

2.2 Compression of 3D Gaussian Splatting

Vanilla 3DGS face high storage demands. However, lightweight solutions for autonomous driving remains underexplored. Existing 3DGS compression research can be classified into four categories (Zhu et al. 2024; Ali et al. 2025): (a) Gaussian pruning (Fan et al. 2025), (b) Vector quantization (Lee et al. 2024), (c) Entropy encoding (Morgenstern et al. 2024), (d) Anchor-based compression (Liu et al. 2024). Unstructured (a-c) are limited in structural modeling, while structured (d) exploit spatial organization for superior compression and fidelity, aligning with our large-scale reconstruction. However, extending them to 4DGS is non-trivial. Recent works have proposed their solutions. 3dgstream (Sun et al. 2024) adaptively adjusts the number of 3D Gaussian distributions to effectively control model complexity and support streaming. MoDec-GS (Kwak et al. 2025) introduces a global-to-local motion decomposition framework to model dynamic motions. In addition, V3 (Wang et al. 2024a) adopts the H.264 codec, 4DGC (Hu et al. 2025) introduces a motion-aware representation, GIFStream (Li et al. 2025) adopts a sparse feature stream, MEGA (Zhang et al. 2025a) introduces entropy-constrained Gaussian deformation. However, they are predominantly designed for static or small-scale scenes. When benchmarked against the demands of large-scale urban scenes, their performance is even inferior to vanilla 3DGS (see Tab. 1). More analysis can be found in Sec. 3.1.

3 Methodology

Our goal is to learn an urban-scene representation capable of NVS. This poses two challenges: **(I)** scalable modeling of large-scale and diverse environments, and **(II)** high-fidelity representation of complex dynamic motions. We tackle these through a motion-aware anchor-based hierarchical framework (Fig. 3). In Sec. 3.1, we initiate with an analysis of why prior compression methods fail in urban scenes. Sec. 3.2 describes our motion decomposition mechanism, where high-level anchors model coarse dynamics and low-level anchors refine them with residual deformations. Sec. 3.3 introduces a covariance-aware local support network that strengthens inter-anchor consistency. To address geometric inconsistencies under sparse LiDAR, we propose a hybrid supervision strategy integrating global geometric constraints and local pixel-level cues in Sec. 3.4. Moreover, Sec. 3.5 details an adaptive strategy for anchor growth and pruning, and Sec. 3.6 describes some optimization details.

3.1 Analysis of Different Compression Methods

In urban driving scenes, the asynchronous and irregular motions caused by multiple independently moving agents pose challenges to existing compression methods. Some approaches (*e.g.*, C-3DGS) reduce storage through vector quantization and pruning, but rely on encoded dynamic representations, limiting their ability to explicitly capture complex motion. Other methods (*e.g.*, MoDec-GS) assume a globally dominant motion and reconstruct attributes via deformation, making them less effective in multi-agent environments. In contrast, uncompressed methods (*e.g.*, DeSiRe-GS) explicitly parameterize trajectories, enabling more accurate dynamic reconstruction. As shown in Table 2, both C-3DGS and MoDec-GS exhibit notable degradation in urban scenes, underscoring their limited adaptability to complex traffic dynamics, whereas DeSiRe-GS remains robust. These observations motivate our hierarchical motion decomposition, aiming to better balance rendering fidelity and storage efficiency. The following sections detail our method.

Model	Scene	PSNR	SSIM	LPIPS	Stor. (MB)
4DGS[1]	dynamic & outdoor	31.15	0.968	0.049	90
C-3DGS		31.73	0.969	0.041	21.8
4DGS[2]		27.44	0.797	0.302	72.65
MoDec-GS		27.78	0.827	0.219	40.82
4DGS	dynamic & urban	28.91	0.874	0.269	1269
C-3DGS		26.19	0.793	0.350	**251**
MoDec-GS		27.43	0.838	0.297	742
DeSiRe-GS		<u>32.88</u>	<u>0.901</u>	<u>0.203</u>	917
HDGS (ours)		**34.52**	**0.932**	**0.174**	<u>284</u>

Table 2: Comparison of dynamic methods. 4DGS (Wu et al. 2024) targets small-scale scenes. C-3DGS and MoDec-GS are compression methods (gray background). DeSiRe-GS targets urban scenes. 4DGS[1] and 4DGS[2] denote baseline 4DGS trained on data from C-3DGS and MoDec-GS.

3.2 Hierarchical Motion Decomposition

To capture heterogeneous motion of diverse objects, we introduce a coarse-to-fine motion decomposition, where high-level anchors model coarse, rigid dynamics, and lower-level anchors refine these with residual, non-rigid deformations.

Inspired by the core design of Scaffold-GS (Lu et al. 2024), We construct an N-layer hierarchy, where layer $(N-1)$ holds K Gaussians for each initial anchor, while layers 0 to $(N-2)$ contain anchors. Each anchor in layer $i \in [1, N-2]$ is derived from its parent in layer $(i-1)$, with each parent spawning k children, whose attributes are determined by combining the parent's attributes with residual offsets predicted at layer i. First, we define the parameters associated with anchors or Gaussians at different levels:

For an anchor (a&b for layer 0, a for layers 1 - $(N-2)$):

- **a.** 3D center $\tilde{\boldsymbol{\mu}}(t) \in \mathbb{R}^3$, local context feature $\boldsymbol{f} \in \mathbb{R}^{32}$, instance feature $\boldsymbol{h} \in \mathbb{R}^8$ (children inherit parents'), quaternion $\boldsymbol{q} \in \mathbb{R}^4$, scale $\boldsymbol{s} \in \mathbb{R}^3$, offsets $\boldsymbol{o} \in \mathbb{R}^{k\times 3}$;
- **b.** peak time $\tau \in \mathbb{R}$, velocity $\boldsymbol{v} \in \mathbb{R}^3$.

For an Gaussian primitive in layer $N-1$:

- **a.** 3D center $\tilde{\boldsymbol{\mu}}_g(t) \in \mathbb{R}^3$, quaternion $\boldsymbol{q}_g \in \mathbb{R}^4$, scale $\boldsymbol{s}_g \in \mathbb{R}^3$, color $\boldsymbol{c} \in \mathbb{R}^3$, opacity $\alpha(t) \in \mathbb{R}$.

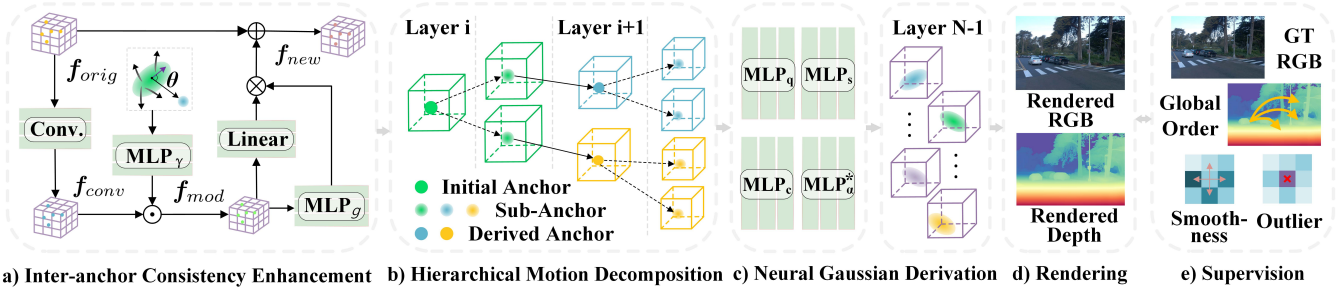

Figure 3: Overview of the HDGS framework. Sub-anchors are generated from the initial anchors via learned offsets $\boldsymbol{o}$ while preserving their attributes, whereas derived anchors are further obtained from the sub-anchors through residual refinements.

Then, the positions and attributes are given by:

- the anchor in layer 0.

Inspired by (Chen et al. 2023; Peng et al. 2025), we employ periodic vibration functions to represent coarse motion:

$$\tilde{\boldsymbol{\mu}}_0(t) = \boldsymbol{\mu}_0 + \frac{p}{2\pi} \cdot \sin\left(\frac{2\pi(t-\tau)}{p}\right) \cdot \boldsymbol{v}, \tag{1}$$

where, $\boldsymbol{\mu}_0 \in \mathbb{R}^3$ is the vibration center at the life peak τ, $p \in \mathbb{R}$ denotes the period, $t \in \mathbb{R}$ is the camera timestamp.

- the anchor in layers 1 to $(N-2)$.

The position is derived through MLP-predicted residuals conditioned on the parent anchor's properties, *i.e.*,

$$\begin{aligned}\{\tilde{\boldsymbol{\mu}}_{i,0}(t), ..., \tilde{\boldsymbol{\mu}}_{i,k-1}(t)\} =& \tilde{\boldsymbol{\mu}}_{i-1}(t) + \boldsymbol{o}_{i-1} + \\ & \text{MLP}_i^p(\boldsymbol{h}, \boldsymbol{e}_{i-1}^{\mu}, \boldsymbol{e}^t),\end{aligned} \tag{2}$$

where $\boldsymbol{e}_{i-1}^{\mu}$ and $\boldsymbol{e}^t$ are position and time embeddings, h encodes anchor identity. Other attributes follow the same residual form. For example, feature values are given by:

$$\{\boldsymbol{f}_{i,0}, ..., \boldsymbol{f}_{i,k-1}\} = \boldsymbol{f}_{i-1} + \text{MLP}_i^f(\boldsymbol{h}, \boldsymbol{e}_{i-1}^{\mu}, \boldsymbol{e}^t). \tag{3}$$

- the Gaussian in layer $(N-1)$.

The position of $n = K/k^i$ Gaussians is calculated the same as anchors in layer $N-2$, and the final formula is:

$$\begin{aligned}\{\tilde{\boldsymbol{\mu}}_{g,0}(t), ..., \tilde{\boldsymbol{\mu}}_{g,n-1}(t)\} =& \tilde{\boldsymbol{\mu}}_0(t) + (\sum_{i=0}^{N-1} \boldsymbol{o}_i + \\ & \text{MLP}_i^p(\boldsymbol{h}, \boldsymbol{e}_i^{\mu}, \boldsymbol{e}^t)).\end{aligned} \tag{4}$$

Other attributes are obtained through individual MLPs. For example, color values are calculated by:

$$\{\boldsymbol{c}_0, ..., \boldsymbol{c}_{n-1}\} = \text{MLP}_c(\boldsymbol{f}_{N-2}, \delta, \boldsymbol{d}), \tag{5}$$

where δ and $\boldsymbol{d}$ are relative distance and direction from camera viewpoint to the anchor. In addition, we correlate opacity with a local decay term $F(\cdot)$ (Chen et al. 2023):

$$\{\alpha_0(t), ..., \alpha_{n-1}(t)\} = \text{MLP}_\alpha(\boldsymbol{f}_{N-2}, \delta, \boldsymbol{d}) \cdot F(t). \tag{6}$$

3.3 Inter-anchor Consistency Enhancement

Our hierarchical framework starts with coarse anchors and recursively refines them into denser ones. However, bottom-up supervision weakens with depth. As a result, high-level anchors struggle to learn structure-aware features, leading to fractures in geometry and appearance (see Fig. 6).

We propose a local support network to reinforce inter-anchor consistency as shown in Fig. 3(a). We first voxelize anchor positions into a sparse 3D grid, and apply submanifold sparse convolution (Graham, Engelcke, and Van Der Maaten 2018) to facilitate local feature interaction, yielding $\boldsymbol{f}_{conv}$ from the original features $\boldsymbol{f}_{orig} \in \mathbb{R}^{32}$. To incorporate directional awareness, each anchor is associated with a base covariance matrix characterizing the spatial distribution (Zwicker et al. 2001; Kerbl et al. 2023) of meaningful regions, where the principal eigenvector corresponding to the largest eigenvalue, indicates the dominant spatial orientation of this region. For each anchor pair, we compute the angle $\theta \in \mathbb{R}$ between the anchor's principal direction and the direction toward its neighbor, which, together with the eigenvalues $\boldsymbol{\lambda} \in \mathbb{R}^3$, is fed into an MLP$_\gamma(\cdot)$ to produce per-channel modulation weights for feature interaction, *i.e.*,

$$\boldsymbol{f}_{mod} = \text{MLP}_\gamma(\cos\theta, \boldsymbol{\lambda}) \odot \boldsymbol{f}_{conv}. \tag{7}$$

To avoid oversmoothing and retain semantic diversity, we incorporate a gated residual fusion mechanism. Concretely, $\boldsymbol{f}_{mod}$ is fed into a linear layer to predict a residual $\Delta\boldsymbol{f}$, modulated by a gating coefficient g derived from $\boldsymbol{f}_{mod}$:

$$\boldsymbol{f}_{new} = \boldsymbol{f}_{orig} + g \cdot \Delta\boldsymbol{f}, \tag{8}$$

where, $g = \text{MLP}_g(\boldsymbol{f}_{mod})$, $\Delta\boldsymbol{f} = \text{Linear}(\boldsymbol{f}_{mod})$.

3.4 Geometric Consistency Enhancement

In urban scenes, the ambiguity from sparse LiDAR supervision causes geometric inconsistency (see Fig. 7). We propose a hybrid supervision scheme that synergistically fuses global geometric constraints and local pixel-level cues for geometrically consistent reconstructions.

Global Depth Supervision (GDS). Benefiting from recent depth estimation (Yang et al. 2024b), we attempt to utilize it as supervision. However, the pre-trained model still exhibits deviations in absolute depth for unseen scenes. We find that the pseudo-depth preserves reliable depth order (Sun et al. 2023), motivating the use of Spearman correlation as global

supervisory, which captures relative rankings rather than absolute values. Since discrete ranks are non-differentiable, we adopt a soft ranking approach for end-to-end trainability:

$$\tilde{\boldsymbol{r}}_i(\boldsymbol{z}) = \sum_{j=1}^{N_d} \sigma\left(\frac{\boldsymbol{z}_j - \boldsymbol{z}_i}{\tau}\right), \tag{9}$$

where $\tau > 0$ is a temperature constant, $\sigma(\cdot)$ is the sigmoid function, $\boldsymbol{z} \in \mathbb{R}^{h\times w}$ (h/w: image height/width, $N_d = h \times w$ is the pixel number) represents rendered depth maps $\boldsymbol{z}_r$ or estimated depth maps $\boldsymbol{z}_e$. Let $\boldsymbol{r}_r$, $\boldsymbol{r}_e \in \mathbb{R}^{h\times w}$ denote soft-rank matrices and $\tilde{\boldsymbol{r}}_r$, $\tilde{\boldsymbol{r}}_e \in \mathbb{R}^{N_d}$ their vectorized forms, the Spearman correlation is then approximated by:

$$\rho_g \approx \frac{\mathrm{Cov}(\tilde{\boldsymbol{r}}_r, \tilde{\boldsymbol{r}}_e)}{\sqrt{\mathrm{Var}(\tilde{\boldsymbol{r}}_r)\,\mathrm{Var}(\tilde{\boldsymbol{r}}_e)}}. \tag{10}$$

The loss encourages rank consistency via:

$$\mathcal{L}_g = 1 - \rho_g. \tag{11}$$

Local Depth Supervision (LDS). However, only a global ordering metric is insufficient to suppress local anomalies such as outliers or oscillations. Thus, we employ local supervision, comprising a smoothness term $\mathcal{L}_{sm}$ promoting depth continuity among neighboring pixels, and an outlier penalty $\mathcal{L}_{out}$ penalizing large local errors. $\mathcal{L}_{sm}$ is defined as:

$$\mathcal{L}_{sm} = \frac{1}{N_d}\sum_{x,y} \boldsymbol{w}_1(x,y)\cdot\left(\boldsymbol{\nabla}_x(x,y) + \boldsymbol{\nabla}_y(x,y)\right), \tag{12}$$

$$\text{where} \quad \begin{cases} \boldsymbol{w}_1(x,y) = \exp(-\beta\cdot\|\boldsymbol{\nabla}\boldsymbol{I}(x,y)\|), \\ \boldsymbol{\nabla}_x(x,y) = |\boldsymbol{z}_r(x+1,y) - \boldsymbol{z}_r(x,y)|, \\ \boldsymbol{\nabla}_y(x,y) = |\boldsymbol{z}_r(x,y+1) - \boldsymbol{z}_r(x,y)|, \end{cases}$$

$\boldsymbol{\nabla}\boldsymbol{I}(x,y)$ is image Sobel gradient, $\|\cdot\|$ and $|\cdot|$ denote L2 and L1 norms. The weight $\boldsymbol{w}_1(x,y)$ weakens regularization near edges to preserve depth discontinuities. The hyperparameter β controls edge sensitivity. The outlier penalty is given by:

$$\mathcal{L}_{out} = \frac{1}{N_d}\sum_{x,y} \boldsymbol{w}_2(x,y)\cdot\boldsymbol{m}(x,y)\cdot\boldsymbol{e}(x,y), \tag{13}$$

$$\text{where} \quad \begin{cases} \boldsymbol{e}(x,y) = |\boldsymbol{r}_r(x,y) - \boldsymbol{\mu}_r(x,y)|, \\ \boldsymbol{m}(x,y) = \mathbf{1}\left(\boldsymbol{e}(x,y) > \delta\right), \end{cases}$$

and $\boldsymbol{\mu}_r(x,y)$ denotes the local neighborhood mean, δ is a rank deviation threshold, $\mathbf{1}(\cdot)$ is the indicator function. For simplicity, $\boldsymbol{w}_2(x,y)$ is formulated identically to $\boldsymbol{w}_1(x,y)$.

3.5 Adaptive Control of Anchors

Growing. Urban scenes exhibit highly diverse spatial characteristics, with both texture-rich (*e.g.*, signs) and texture-sparse regions (*e.g.*, roads). In this context, traditional uniform anchor growing strategies (Lu et al. 2024; Chen et al. 2024) are suboptimal, as equal voxel resolution across all regions hinders detail capture in texture-rich areas. We propose an adaptive method (**GDG**) guided by dual thresholds on gradient and density. Neural Gaussians are voxelized to estimate local density and average gradient over $\mathcal{N}_g$ iterations. Regions with both high gradient and density trigger finer anchor subdivision to capture detailed structures, while those with high gradient but low density grow fewer anchors, as shown in Fig. 4. For temporal changes, we restrict anchor growth to randomly sampled regions with high temporal gradient to prevent excessive anchor proliferation.

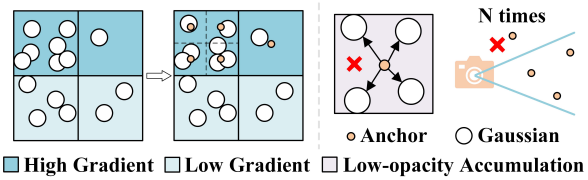

Figure 4: Adaptive control of anchors.

Pruning is based on two criteria: **(I)** anchors with low-opacity Gaussians over $\mathcal{N}_p$ iterations are removed. **(II)** anchors outside the view are tracked by unvisited record (**UR**) and pruned when exceeding a set threshold, ensuring efficient memory use in unbounded urban environments.

3.6 Optimization

All the learnable parameters are simultaneously optimized in an end-to-end manner, with the following loss function:

$$\begin{aligned}\mathcal{L} =& (1-\lambda_r)\mathcal{L}_1 + \lambda_r\mathcal{L}_{ssim} + \lambda_g\mathcal{L}_g + \lambda_{sm}\mathcal{L}_{sm} + \\ & \lambda_{out}\mathcal{L}_{out} + \lambda_l\mathcal{L}_l + \lambda_s\mathcal{L}_s,\end{aligned} \tag{14}$$

where $\mathcal{L}_1$ and $\mathcal{L}_{ssim}$ are L1 and SSIM losses for supervising RGB rendering, respectively. $\mathcal{L}_g$, $\mathcal{L}_{sm}$, $\mathcal{L}_{out}$ are global Spearman loss, local depth loss, outlier depth loss. $\mathcal{L}_l$ compares the rendered depth of Gaussians with sparse depth measurements obtained from LiDAR. $\mathcal{L}_s$ is a binary cross entropy loss for sky supervision, where the sky mask is estimated by a pre-trained segmentaion model (Xie et al. 2021).

4 Experiments

4.1 Experimental Setup

Implementation Details. We adopt a three-layer hierarchy and a doubling expansion strategy, with each parent anchor spawning $k = 2$ children. The number of Gaussian primitives is set to $K = 12$. The voxel size is set to 0.001. The local feature dimension is set to 32, and the instance feature dimension is set to 8. The loss weights λ_r, λ_g, λ_{sm}, λ_{out}, λ_l, λ_s are set to 0.2, 0.005, 0.001, 0.001, 0.01, 0.05, respectively. Our method runs on a single NVIDIA V100 GPU.

Dataset. We execute experiments on Waymo (Sun et al. 2020) and KITTI (Geiger, Lenz, and Urtasun 2012). For Waymo, we select 4 and 8 sequences chosen by PVG (Chen et al. 2023) and OmniRe (Chen et al. 2025), named SA and SB. We utilize the three front-facing cameras for evaluation. For KITTI, we use the left and right cameras, following (Chen et al. 2023). Storage efficiency is quantified in megabytes (MB) by averaging the model size per scene.

4.2 Comparison with State-of-the-Art Methods

Rendering Comparison. Tab. 3 and Tab. 4 report the quantitative results of image reconstruction and novel view synthesis. Our method surpasses state-of-the-art approaches in

Method	Venue	Box	Resolution	Image reconstruction PSNR ↑	SSIM ↑	LPIPS ↓	Novel view synthesis PSNR ↑	SSIM ↑	LPIPS ↓	FPS ↑	Stor. ↓
Mars (Wu et al. 2023)	CAAI	✓✓	1280×1920	21.81	0.681	0.430	20.69	0.636	0.453	0.030	479
EmerNeRF (Yang et al. 2024a)	ICLR	-	1280×1920	28.11	0.786	0.373	25.92	0.763	0.384	0.053	-
PVG (Chen et al. 2023)	ArXiv	-	1280×1920	32.46	0.910	0.229	28.11	0.849	0.279	**50**	788
DeSiRe-GS (Peng et al. 2025)	CVPR	-	1280×1920	32.88	0.901	0.203	28.58	0.839	0.281	38	917
HDGS (Ours)	-	-	1280×1920	**34.52**	**0.932**	**0.174**	**29.74**	**0.858**	**0.254**	49	**284**
OmniRe (Chen et al. 2025)	ICLR	✓✓	640×960	33.34	0.918	0.197	29.66	0.859	0.226	48	357
HDGS (Ours)	-	-	640×960	**34.98**	**0.934**	**0.165**	**30.11**	**0.876**	**0.210**	**59**	**135**
HUGS (Zhou et al. 2024a)	CVPR	✓	640×960	28.26	0.923	0.092	27.65	0.914	0.097	44	309
StreetGS (Yan et al. 2024)	ECCV	✓	640×960	29.08	0.936	0.087	28.54	0.928	0.105	47	420
OmniRe (Chen et al. 2025)	ICLR	✓	640×960	32.53	0.945	0.066	30.88	0.931	0.075	35	376
OmniRe (Chen et al. 2025)	ICLR	✓✓	640×960	**34.25**	0.954	0.058	**32.57**	0.942	0.067	40	383
DeSiRe-GS (Peng et al. 2025)	CVPR	-	640×960	32.71	0.949	0.103	30.67	0.933	0.118	42	336
HDGS (Ours)	-	-	640×960	33.26	**0.960**	**0.056**	31.26	**0.947**	**0.063**	**51**	**128**

Table 3: Comparison of methods on the Waymo Open Dataset. The upper part utilizes data subset SA, while the lower part uses data subset SB. ✓ indicates predicted boxes, ✓✓ means GT boxes. The evaluation unit for Stor. (Storage) is megabytes (MB).

Method	Venue	Box	Resolution	Image reconstruction PSNR ↑	SSIM ↑	LPIPS ↓	Novel view synthesis PSNR ↑	SSIM ↑	LPIPS ↓	FPS ↑	Stor. ↓
EmerNeRF (Yang et al. 2024a)	ICLR	-	375×1242	26.95	0.828	0.218	25.24	0.801	0.237	0.28	-
PVG (Chen et al. 2023)	ArXiv	-	375×1242	32.83	0.937	0.070	27.43	0.896	0.114	51	776
StreetGS (Yan et al. 2024)	ECCV	-	375×1242	31.28	0.909	0.083	26.91	0.874	0.156	**62**	738
OmniRe (Chen et al. 2025)	ICLR	✓✓	375×1242	33.47	0.965	0.038	28.96	0.910	0.101	50	758
SplatAD (Hess et al. 2025)	CVPR	✓✓	375×1242	32.35	0.922	0.079	27.01	0.884	0.126	46	739
DeSiRe-GS (Peng et al. 2025)	CVPR	-	375×1242	33.94	0.949	0.04	28.87	0.901	0.106	41	691
HDGS (Ours)	-	-	375×1242	**34.38**	**0.972**	**0.032**	**29.25**	**0.914**	**0.093**	53	**265**

Table 4: Comparison of methods on the KITTI Dataset.

rendering quality on both the Waymo and KITTI datasets, demonstrating the superiority of the hierarchical dynamic framework. Fig. 5 further illustrates superior visual quality, especially in detail regions and dynamic objects.

Runtime and Storage Comparison. In terms of storage, we evaluate the storage requirements of various methods in Tab. 3 and Tab. 4. Our approach demonstrates a significant reduction in storage requirements, effectively inheriting the characteristic of anchor-based architectures. In comparison to DeSiRe-GS (Peng et al. 2025), our method significantly improves rendering performance while reducing storage requirements by approximately 69.0%. In terms of runtime, our method achieves comparable or superior speed to all other state-of-the-art approaches except for vanilla 3DGS.

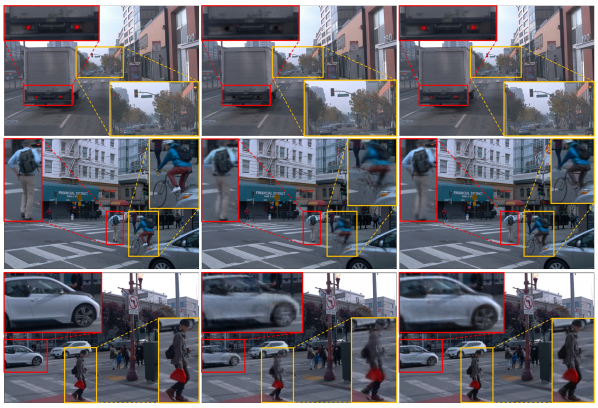

Figure 5: Qualitative comparison of novel view synthesis.

4.3 Ablation Studies

Component-wise Analysis. In Tab. 5, our model is degenerated into five versions to demonstrate the effectiveness of each component. **(I)** enhances inter-anchor consistency in hierarchical structures, improving rendering performance. As shown in Fig. 6, it mitigates geometric and appearance fractures caused by weakened supervision and structural awareness at high levels. Tab. 6(a) provides supporting evidence that performance gains become more pronounced with increasing layers. Tab. 6(b) further confirms that modulation or gating modules yields significant performance gains with minimal computational overhead. **(II)** and **(III)** improves rendering performance, which stems from improved geometric consistency as shown in Fig. 7. However, when integrated into the Gaussian-based DeSiRe-GS (upper part of Tab. 5), the module shows limited impact, likely due to the locally-focused nature of our hierarchical anchor structure. **(IV)** enables to capture finer details, thereby improving the rendering quality of texture-rich edges and further enhancing overall performance. **(V)** illustrates the optimization enabled by the anchor removal module, which reduces the count of meaningless anchors for rendering.

Motion Mode Analysis. We investigate the influence of hierarchical levels on rendering quality and storage efficiency. As observed in Tab. 8, the image quality of novel view synthesis exhibits significant variation with the num-

Model	PSNR	SSIM	LPIPS	Stor.
w GDS	27.05	0.854	0.263	902
w LDS	27.01	0.851	0.265	898
DeSiRe-GS (Peng et al. 2025)	27.03	0.853	0.266	904
(I) w/o ICE	27.23	0.860	0.259	281
(II) w/o GDS	27.38	0.862	0.251	273
(III) w/o LDS	27.50	0.877	0.245	278
(IV) w/o GDG	27.55	0.873	0.248	250
(V) w/o UR	27.98	0.878	0.224	303
HDGS (Ours)	28.04	0.881	0.217	268

Table 5: Ablation study of each module. ICE - inter-anchor consistency enhancement, GDS - global depth supervision, LDS - local depth supervision, GDG - gradient and density-based growing, UR - unvisited record.

Layer	Model	PSNR	Stor.
$N = 2$	w/o ICE	27.38	455
	w ICE	27.99	451
	Δ	0.61	-4
$N = 3$	w/o ICE	27.23	281
	w ICE	28.04	268
	Δ	**0.81**	**-13**

(a) ICE *v.s.* Layer Number.

Model	PSNR	FPS
w/o Mod.	27.39	58
w/o Gate	27.78	52
Full	28.04	47

(b) Ablation of ICE components. Mod. means feature modulation.

Table 6: Ablation of ICE module.

Setting	Error			Accuracy	
	AbsRel↓	SqRel↓	RMSE↓	δ_1↑	δ_2↑
w/o	0.242	12.550	18.398	0.743	0.881
w	0.188	7.954	15.031	0.825	0.927

Table 7: Evaluation on KITTI for depth supervision module.

Layer	PSNR	FPS	Stor.
$N = 1$	26.63	**60**	842
$N = 2$	27.79	54	451
$N = 3$	**28.04**	47	268
$N = 4$	25.72	39	144
$N = 5$	25.40	32	**95**

Table 8: Ablation of hierarchical motion decomposition.

ber of hierarchical levels. In our experiments, a 3-level structure achieves the best rendering quality. In addition, adding one more layer can reduce storage requirements by approximately 50%, alongside a modest cost in inference speed of 10%-20%. This highlights an impressive balance, prioritizing considerable storage savings with a slight cost in speed.

4.4 Visualization and Analysis

This section provides a visual analysis of how our method captures different motion patterns. In the top two rows of Fig. 8, we depict the spatial position and corresponding velocity of rendered pixels. The green ellipses over the pedestrian show the hierarchical interplay between primary and residual motions in a three-layer architecture, which translates to distinct, multi-level velocity variations. In contrast,

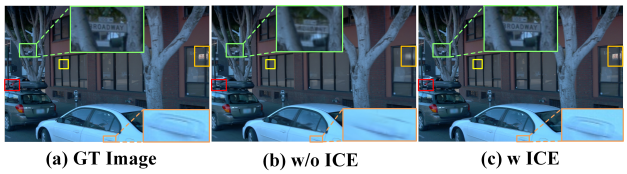

Figure 6: Qualitative comparison for ICE module.

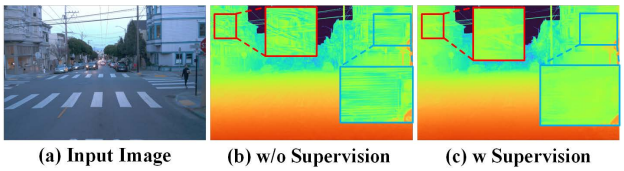

Figure 7: Qualitative comparison of depth.

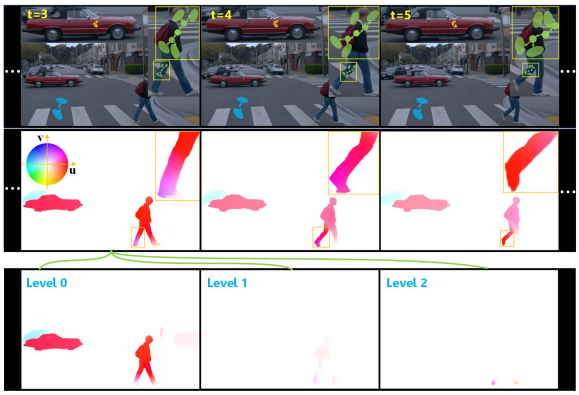

Figure 8: Motion visualization. top: anchor/Gaussian position, middle: whole velocity, bottom: multi-level velocity.

the background's ellipses (blue) remain static, while the vehicle ellipses (orange) move consistently across layers. The bottom row shows the decomposition of total motion into sub-motions across layers for a given frame. High-level anchors primarily model the coarse, rigid motion, while lower layers refine it through residual and non-rigid deformations. These prove our method's ability in capturing both coarse and fine-grained dynamics, supporting the physical interpretability of the hierarchical decomposition.

5 Conclusion

This paper presents HDGS, a novel hierarchical framework designed to adapt the anchor-based Gaussian paradigm to 4D urban environments. By introducing three core components, a local support network for inter-anchor consistency, a motion decomposition module for heterogeneous object motion, and a hybrid depth supervision for geometric consistency, our method achieves efficient compression while maintaining or even improving rendering fidelity. We hope our HDGS will facilitate downstream applications in autonomous driving, such as the closed-loop simulation.

Acknowledgments

This work was supported in part by the Beijing Natural Science Foundation (Grant No. L223003, JQ22014), the Natural Science Foundation of China (Grant No. 62422317, U22B2056, 62036011, 62192782, U2441241, 62503323).

References

Ali, M. S.; Zhang, C.; Cagnazzo, M.; Valenzise, G.; Tartaglione, E.; and Bae, S.-H. 2025. Compression in 3d gaussian splatting: A survey of methods, trends, and future directions. *arXiv preprint arXiv:2502.19457*.

Barron, J. T.; Mildenhall, B.; Tancik, M.; Hedman, P.; Martin-Brualla, R.; and Srinivasan, P. P. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *CVPR*, 5855–5864.

Barron, J. T.; Mildenhall, B.; Verbin, D.; Srinivasan, P. P.; ; and Hedman, P. 2022. Mipnerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 5470–5479.

Chen, Y.; Gu, C.; Jiang, J.; Zhu, X.; and Zhang, L. 2023. Periodic vibration gaussian: Dynamic urban scene reconstruction and real-time rendering. *arXiv preprint arXiv:2311.18561*.

Chen, Y.; Wu, Q.; Lin, W.; Harandi, M.; and Cai, J. 2024. Hac: Hash-grid assisted context for 3d gaussian splatting compression. In *ECCV*, 422–438.

Chen, Z.; Yang, J.; Huang, J.; de Lutio, R.; Esturo, J. M.; Ivanovic, B.; Litany, O.; Gojcic, Z.; Fidler, S.; Pavone, M.; et al. 2025. Omnire: Omni urban scene reconstruction. *ICLR*.

Fan, Z.; Wang, K.; Wen, K.; Zhu, Z.; Xu, D.; and Wang, Z. 2025. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. In *NeurIPS*, 140138–140158.

Ge, F.; Zhang, Y.; Shen, S.; Hu, W.; Wang, Y.; and Gao, J. 2024. BEV2PR: BEV-Enhanced Visual Place Recognition with Structural Cues. In *IROS*, 13274–13281.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 3354–3361.

Graham, B.; Engelcke, M.; and Van Der Maaten, L. 2018. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, 9224–9232.

Hess, G.; Lindström, C.; Fatemi, M.; Petersson, C.; and Svensson, L. 2025. Splatad: Real-time lidar and camera rendering with 3d gaussian splatting for autonomous driving. In *CVPR*, 11982–11992.

Hu, Q.; Zheng, Z.; Zhong, H.; Fu, S.; Song, L.; Zhang, X.; Zhai, G.; and Wang, Y. 2025. 4DGC: Rate-Aware 4D Gaussian Compression for Efficient Streamable Free-Viewpoint Video. In *CVPR*, 875–885.

Huang, N.; Wei, X.; Zheng, W.; An, P.; Zhan, M. L. W.; Tomizukaand, M.; Keutzer, K.; and Zhang, S. 2024. S3gaussian: Self-supervised street gaussians for autonomous driving. *arXiv preprint arXiv:2311.18561*.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian splatting for real-time radiance field rendering. *ACM TOG*, 139–1.

Kwak, S.; Kim, J.; Jeong, J. Y.; Cheong, W.-S.; Oh, J.; and Kim, M. 2025. Modec-gs: Global-to-local motion decomposition and temporal interval adjustment for compact dynamic 3d gaussian splatting. In *CVPR*, 11338–11348.

Lee, J. C.; Rho, D.; Sun, X.; Ko, J. H.; and Park, E. 2024. Compact 3d gaussian representation for radiance field. In *CVPR*, 21719–21728.

Li, H.; Li, S.; Gao, X.; Batuer, A.; Yu, L.; and Liao, Y. 2025. GIFStream: 4D Gaussian-based Immersive Video with Feature Stream. In *CVPR*, 21761–21770.

Li, Z.; Wang, W.; Li, H.; Xie, E.; Sima, C.; Lu, T.; Yu, Q.; and Dai, J. 2024. Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *TPAMI*.

Liu, X.; Wu, X.; Zhang, P.; Wang, S.; Li, Z.; and Kwong, S. 2024. Compgs: Efficient 3d scene representation via compressed gaussian splatting. In *ACM MM*, 2936–2944.

Lu, T.; Yu, M.; Xu, L.; Xiangli, Y.; Wang, L.; Lin, D.; and Dai, B. 2024. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *CVPR*, 20654–20664.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 99–106.

Morgenstern, W.; Barthel, F.; Hilsmann, A.; and Eisert, P. 2024. Compact 3d scene representation via self-organizing gaussian grids. In *ECCV*, 18–34.

Nguyen, T.-A.-Q.; Roldão, L.; Piasco, N.; Bennehar, M.; and Tsishkou, D. 2024. Rodus: Robust decomposition of static and dynamic elements in urban scenes. *arXiv preprint arXiv:2403.09419*.

Peng, C.; Zhang, C.; Wang, Y.; Xu, C.; Xie, Y.; Zheng, W.; Keutzer, K.; Tomizuka, M.; and Zhan, W. 2025. Desire-gs: 4d street gaussians for static-dynamic decomposition and surface reconstruction for urban driving scenes. In *CVPR*, 6782–6791.

Rematas, K.; Liu, A.; Srinivasan, P. P.; Barron, J. T.; Tagliasacchi, A.; Funkhouser, T.; and Ferrari, V. 2022. Urban radiance fields. In *CVPR*, 12932–12942.

Sun, J.; Jiao, H.; Li, G.; Zhang, Z.; Zhao, L.; and Xing, W. 2024. 3dgstream: On-the-fly training of 3d gaussians for efficient streaming of photo-realistic free-viewpoint videos. In *CVPR*, 20675–20685.

Sun, L.; Bian, J.-W.; Zhan, H.; Yin, W.; Reid, I.; and Shen, C. 2023. Sc-depthv3: Robust self-supervised monocular depth estimation for dynamic scenes. *TPAMI*, 497–508.

Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; and et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2446–2454.

Tancik, M.; Casser, V.; Yan, X.; Pradhan, S.; Mildenhall, B.; Srinivasan, P. P.; Barron, J. T.; and Kretzschmar, H. 2022.

Block-nerf: Scalable large scene neural view synthesis. In *CVPR*, 8248–8258.

Turki, H.; Ramanan, D.; and Satyanarayanan, M. 2022. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *CVPR*, 12922–12931.

Wang, P.; Zhang, Z.; Wang, L.; Yao, K.; Xie, S.; Yu, J.; Wu, M.; and Xu, L. 2024a. Vˆ 3: Viewing Volumetric Videos on Mobiles via Streamable 2D Dynamic Gaussians. *TOG*, 1–13.

Wang, Y.; Li, Z.; Guo, L.; Yang, W.; Kot, A. C.; and Wen, B. 2024b. Contextgs: Compact 3d gaussian splatting with anchor level context model. *arXiv preprint arXiv:2405.20721*.

Wu, G.; Yi, T.; Fang, J.; Xie, L.; Zhang, X.; Wei, W.; Liu, W.; Tian, Q.; and Wang, X. 2024. 4d gaussian splatting for real-time dynamic scene rendering. In *CVPR*, 20310–20320.

Wu, Z.; Liu, T.; Luo, L.; Zhong, Z.; Chen, J.; Xiao, H.; Hou, C.; Lou, H.; Chen, Y.; Yang, R.; et al. 2023. Mars: An instance-aware, modular and realistic simulator for autonomous driving. In *CAAI*, 3–15.

Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J. M.; and Luo, P. 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 12077–12090.

Yan, Y.; Lin, H.; Zhou, C.; Wang, W.; Sun, H.; Zhan, K.; Lang, X.; Zhou, X.; and Peng, S. 2024. Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In *ECCV*, 156–173.

Yang, J.; Ivanovic, B.; Litany, O.; Weng, X.; Kim, S. W.; Li, B.; Che, T.; Xu, D.; Fidler, S.; Pavone, M.; et al. 2024a. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *ICLR*.

Yang, L.; Kang, B.; Huang, Z.; Xu, X.; Feng, J.; and Zhao, H. 2024b. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 10371–10381.

Zhang, X.; Liu, Z.; Zhang, Y.; Ge, X.; He, D.; Xu, T.; Wang, Y.; Lin, Z.; Yan, S.; and Zhang, J. 2025a. Mega: Memory-efficient 4d gaussian splatting for dynamic scenes. In *ICCV*, 27828–27838.

Zhang, Y.; Gao, J.; Ge, F.; Luo, G.; Li, B.; Zhang, Z.; Ling, H.; and Hu, W. 2025b. VQ-Map: Bird's-Eye-View Map Layout Estimation in Tokenized Discrete Space via Vector Quantization. *NIPS*, 70453–70475.

Zhou, H.; Shao, J.; Xu, L.; Bai, D.; Qiu, W.; Liu, B.; Wang, Y.; Geiger, A.; and Liao, Y. 2024a. Hugs: Holistic urban 3d scene understanding via gaussian splatting. In *CVPR*, 21336–21345.

Zhou, X.; Lin, Z.; Shan, X.; Wang, Y.; Sun, D.; and Yang, M.-H. 2024b. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *CVPR*, 21634–21643.

Zhu, S.; Wang, G.; Kong, X.; Kong, D.; and Wang, H. 2024. 3d gaussian splatting in robotics: A survey. *arXiv preprint arXiv:2410.12262*.

Zwicker, M.; Pfister, H.; Van Baar, J.; and Gross, M. 2001. Ewa volume splatting. In *VIS*, 29–538.