This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

VAD-GS: Visibility-Aware Densification for 3D Gaussian Splatting in Dynamic Urban Scenes

Yikang Zhang[1] yikangzhang@tongji.edu.cn

Rui Fan[1,2,3]✉ rui.fan@ieee.org

[1]Shanghai Research Institute for Intelligent Autonomous Systems, Tongji University
[2]College of Electronic and Information Engineering, Tongji University
[3]National Key Laboratory of Human-Machine Hybrid Augmented Intelligence, Xi'an Jiaotong University

Abstract

3D Gaussian splatting (3DGS) has demonstrated impressive performance in synthesizing high-fidelity novel views. Nonetheless, its effectiveness critically depends on the quality of the initialized point cloud. Specifically, achieving uniform and complete point coverage over the underlying scene structure requires overlapping observation frustums, an assumption that is often violated in unbounded, dynamic urban environments. Training Gaussian models with partially initialized point clouds often leads to distortions and artifacts, as camera rays may fail to intersect valid surfaces, resulting in incorrect gradient propagation to Gaussian primitives associated with occluded or invisible geometry. Additionally, existing densification strategies simply clone and split Gaussian primitives from existing ones, incapable of reconstructing geometry from missing structures. To address these limitations, we propose VAD-GS, a 3DGS framework tailored for geometry recovery in challenging urban scenes. Our method identifies unreliable geometry structures via voxel-based visibility reasoning, selects informative supporting views through diversity-aware view selection, and recovers missing structures via multi-view stereo reconstruction. This design enables the generation of new Gaussian primitives guided by reliable geometric priors, even in regions lacking initial points. Extensive experiments on the Waymo and nuScenes datasets demonstrate that VAD-GS outperforms state-of-the-art 3DGS approaches and significantly improves the quality of reconstructed geometry for both static and dynamic objects. Our project webpage is at mias.group/VAD-GS/.

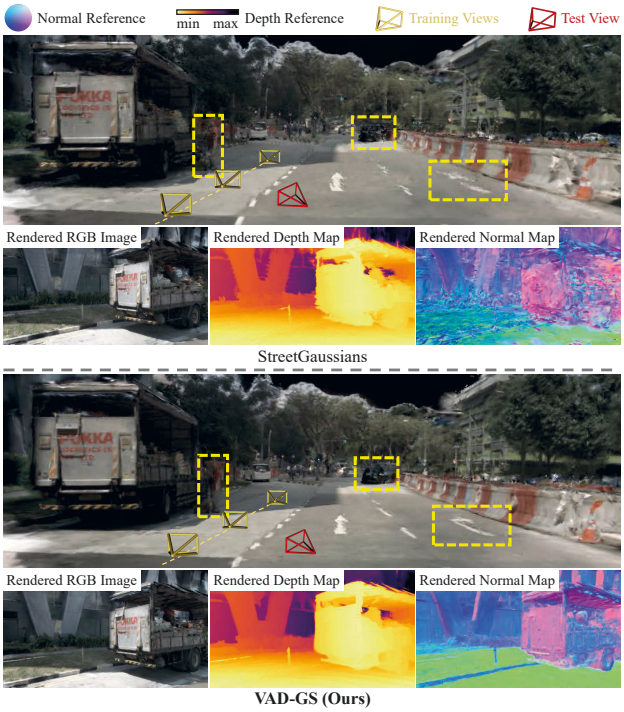

Figure 1. **A comparison between VAD-GS and StreetGaussians.** While both methods achieve comparable rendering quality, VAD-GS demonstrates superior recovery of incomplete or unreliable scene geometry, as evidenced by notable improvements in the rendered depth and normal maps. The supplement provides additional results, including a video demonstration of VAD-GS.

1. Introduction

Realistic simulation is critical for the development and validation of autonomous driving systems [1]. Traditional simulators rely on handcrafted assets, inherently limiting scene scalability and diversity [2]. Recent advances in neural scene representations have enabled data-driven, photorealistic novel view synthesis (NVS), which provides a more efficient and scalable alternative. Specifically, neural radiance field (NeRF)-based approaches [3] represent scenes using neural networks and achieve high-fidelity 3D recon-

✉Corresponding author.

struction. Nevertheless, volume rendering is typically computationally intensive, thereby limiting the practical applicability of NeRF and its variants. To enable real-time rendering, 3D Gaussian splatting (3DGS) explicitly represents scenes as anisotropic 3D Gaussian primitives with learnable geometry and appearance attributes. These primitives are jointly optimized to align with the underlying scene structure by enforcing photometric consistency across images captured from multiple views. Building on its impressive performance in object-level reconstruction, recent extensions [4, 5] have demonstrated strong potential for reconstructing complex urban scenes.

Despite these advances, recovering complete and reliable geometry for unbounded environments from sparse observations remains a major challenge. In 3DGS, scene completeness and texture details are typically enhanced by splitting or cloning existing Gaussian primitives, which are initialized from point clouds obtained through structure from motion (SfM) or LiDAR scan accumulation [1]. However, autonomous driving datasets present inherent limitations: (1) Unlike scene-centric reconstruction settings, multiple cameras mounted on a vehicle capture outward-facing views in urban scenes with limited overlap (typically less than 15%) [6], which makes stereo matching between adjacent, synchronous images unreliable; (2) Although multi-view stereo (MVS) methods can recover static scene structure from asynchronous video frames, they are ineffective in reconstructing dynamic objects; (3) Despite the incorporation of LiDAR points to enhance geometric consistency in pioneering studies [7, 8], substantial blind spots in scene structure persist due to the limited field of view. Consider a typical scenario where a traffic sign is positioned beyond the LiDAR's vertical range (as shown in the supplement and video). Its low-texture surface also provides insufficient visual cues for reliable SfM reconstruction across images. As a result, the initialized point cloud fails to recover accurate surface geometry. Furthermore, during Gaussian training, photometric errors caused by missing geometry are erroneously attributed to background structures, such as trees or buildings behind the sign. Consequently, gradient-based splitting and cloning operations may inadvertently be applied to invisible Gaussian primitives. Although this training process may improve rendering quality for specified views, it distorts the underlying scene geometry and ultimately degrades generalization to unseen perspectives.

Several recent studies have focused on enhancing scene completeness within the original 3DGS framework. For instance, GeoTexDensifier [9] incorporates additional depth and normal priors to guide the splitting process for improved surface alignment, whereas DNGaussian [10] identifies missing geometry by performing global-local normalization between the rendered depth and that estimated using DPT [11]. Nonetheless, these methods are confined to regions with existing Gaussian primitives and are incapable of handling uninitialized areas. To overcome this drawback, GaussianPro [12] introduces a patch matching-based geometry completion strategy, which leverages stereo constraints from a set of images with precomputed camera poses to generate additional point clouds independent of the rendering process. While GaussianPro greatly enhances geometry recovery, it remains limited to static scenes and is incapable of handling dynamic object densification. Moreover, it relies solely on adjacent frames captured using a single camera, thereby missing long-range temporal dependencies and cross-camera visual cues.

To address the aforementioned challenges, this paper introduces a **v**isibility-**a**ware **d**ensification framework for 3D **G**aussian **s**platting (**VAD-GS**) tailored for dynamic, unbounded urban environments. Unlike previous approaches that passively react to photometric errors, VAD-GS actively evaluates structural completeness and selectively reconstructs incomplete regions by leveraging views that provide the most reliable stereo geometry. Specifically, we introduce a voxel-based object surface visibility reasoning approach that provides geometric priors for both static backgrounds and dynamic objects. Each voxel aggregates view-dependent visibility information of the corresponding 3D points, thereby enabling occlusion-aware modeling through depth rasterization with z-buffering. Furthermore, we propose a diversity-aware view sampling strategy that selects informative supporting views for each reference view, aiming to balance view frustum overlap and triangulation quality. The selected views are processed using a patch matching-based MVS algorithm to extract depth and normal information, which serves as reliable geometric priors for new Gaussian primitive initialization and scene consistency enforcement. VAD-GS is evaluated on the Waymo Open dataset [13] and the nuScenes dataset [14], both of which contain complex urban dynamics and sparse multi-view observations. Extensive experiments demonstrate that VAD-GS achieves state-of-the-art (SoTA) rendering quality while producing more consistent geometry with fewer artifacts compared to previous SoTA methods (see Fig. 1). The main contributions of this study are summarized as follows:

- A novel Gaussian splatting framework tailored for dynamic urban scenes, which actively completes missing geometry using multi-camera, cross-frame observations.
- A voxel-based surface visibility reasoning approach that identifies unreliable static and dynamic object geometry.
- A diversity-aware sampling strategy that improves MVS reconstruction quality by optimizing supporting views.
- An extension of MVS reconstruction to dynamic, multi-camera driving scenarios, enabling both Gaussian densification and scene consistency enforcement.

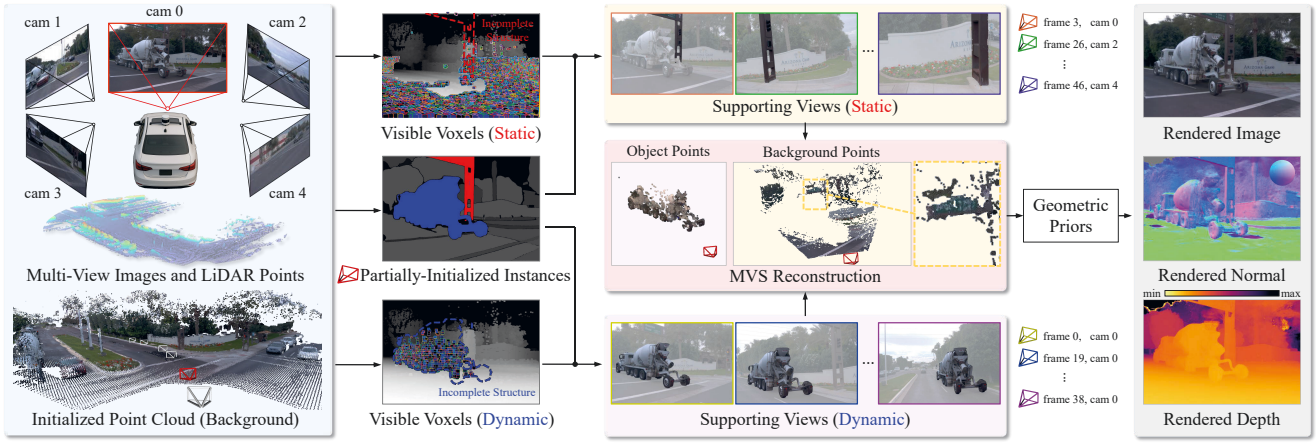

Figure 2. **VAD-GS pipeline.** For each static or dynamic instance with incomplete geometry, VAD-GS first performs voxel-based visibility reasoning to identify a set of potential observation views. It then incrementally selects diverse supporting views to perform MVS reconstruction. The resulting geometric priors are subsequently used for Gaussian densification and optimization.

2. Related Work

2.1. Novel View Synthesis

NVS aims to generate photorealistic images of objects or scenes from previously unseen viewpoints, without explicitly modeling 3D simulation assets or lighting maps. NeRF [3], a pioneering work in this field, represents 3D scenes as learnable volumetric density fields, parameterized by a large multi-layer perceptron (MLP). Subsequent studies have primarily focused on improving both rendering quality and computational efficiency. For example, Instant-NGP [15] introduces a multi-resolution hash encoding scheme that adaptively allocates higher representational capacity to geometrically complex regions, thereby significantly improving rendering efficiency. Mip-NeRF [16] improves point sampling in ray marching to mitigate aliasing artifacts caused by resolution mismatches, while its extension Mip-NeRF 360 [17] further adapts the approach to handle unbounded scenes. However, the computational cost of volume rendering remains a major barrier to the practical deployment of NeRF models in real-world applications.

Recent 3DGS approaches [18] have introduced an alternative NVS paradigm, enabling real-time rendering of large-scale scenes. By projecting anisotropic 3D Gaussian ellipsoids onto the 2D image plane using splatting-based rasterization and computing pixel colors through depth sorting and α-blending, these methods effectively circumvent the computational overhead of ray marching. Since then, several studies have extended 3DGS to dynamic urban scenes. Notably, StreetGaussians [7] models dynamic vehicles in the foreground as rigid groups of Gaussian primitives and employs a 4D spherical harmonics model to capture appearance variation over time. Similarly, DrivingGaussian [8] uses a composite dynamic Gaussian graph to model multiple dynamic objects, OmniRe [19] incorporates skinned multi-person linear Gaussians to represent non-rigid entities such as pedestrians and cyclists, while HUGSIM [20] introduces a multi-plane ground model to alleviate lane distortion in extrapolated views. Other works [21–24] have also explored related extensions in different directions. Despite the appealing results achieved by this paradigm, two major limitations persist: (1) new Gaussian primitives are typically generated by splitting or cloning existing ones, making reconstruction quality highly dependent on the accuracy and completeness of the initialized point clouds; (2) recovering missing geometry based solely on photometric errors is inherently challenging, as gradient updates may be incorrectly propagated to view-proximal yet geometrically unrelated Gaussians, leading to the distortion of neighboring Gaussians and ultimately degrading the overall scene geometry. To address these issues, we explore the incorporation of additional geometric cues, particularly MVS constraints, to obtain more reliable structural information beyond gradient propagation.

2.2. Multi-View Stereo

MVS is a fundamental computer vision technique that reconstructs dense 3D scene geometry from a set of images with known intrinsic parameters [25]. Online MVS methods typically select keyframes from low-resolution video streams to perform real-time camera pose estimation and point cloud generation [26]. In contrast, offline MVS approaches, typically following the SfM pipeline [27–29], aim for high-resolution, large-scale scene reconstruction at the expense of greater computational complexity. Despite their differing objectives, both pipelines face critical challenges that directly impact reconstruction quality, particularly in view selection and depth estimation. For view selection, GP-MVS [30] employs a heuristic pose-distance measure function to select informative keyframes, while MVS-

Net [31] introduces a score function that ranks neighboring views based on frustum overlap. In terms of depth estimation, plane sweeping-based methods discretize depth candidates to construct cost volumes and measure feature similarity across warped views, thereby favoring reconstructions with higher resolution [32–34]. In contrast, patch matching-based approaches achieve high efficiency by randomly sampling depth guesses for individual pixels and iteratively propagating plausible estimates from neighboring pixels [35, 36]. In this work, we exploit MVS consistency not only to guide the densification of Gaussian primitives for each object, but also to provide geometric supervision that complements photometric gradient-based optimization. More importantly, we extend MVS consistency to dynamic settings where both objects and cameras are in motion, which remains challenging for existing methods.

3. Preliminaries

3DGS-based approaches represent a scene using a set of anisotropic 3D Gaussian primitives [18]. Each primitive $\boldsymbol{x}$ is modeled by a Gaussian distribution, defined as follows:

$$G(\boldsymbol{x}) = \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right), \quad (1)$$

where $\boldsymbol{\mu} \in \mathbb{R}^3$ denotes the primitive center, and the covariance matrix $\boldsymbol{\Sigma}$ is parameterized using a scaling factor and a rotation quaternion. Unlike volumetric representations such as NeRF, 3DGS avoids the computational overhead of volumetric ray marching by adopting a tile-based rasterization pipeline. The color C at each pixel is obtained by compositing the overlapping Gaussians along the pixel ray via front-to-back α-blending, as expressed as follows:

$$C = \sum_{i\in\mathcal{N}} \boldsymbol{c}_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j), \quad (2)$$

where $\mathcal{N}$ denotes the set of Gaussian primitives intersected by the pixel ray, while $\boldsymbol{c}_i$ and α_i represent the color and opacity of the i-th primitive, respectively. The photometric error of C propagates gradients to Gaussians that exhibit inconsistencies in shape or appearance. To improve scene geometry representation, 3DGS adopts a densification strategy by splitting existing Gaussians with large covariance into smaller primitives. Theoretically, these newly generated Gaussians are iteratively refined to converge toward consistent geometry and appearance, thereby improving both coverage and structural fidelity.

4. Methodology

As illustrated in Fig. 2, the proposed VAD-GS framework identifies visible surfaces of unreliable geometry, selects informative supporting views, and complements missing structures via MVS-guided densification.

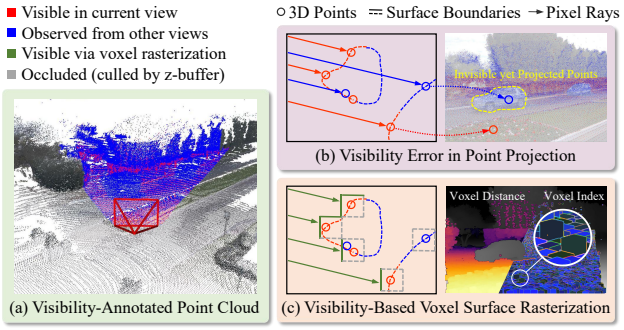

Figure 3. **Voxel-based visibility reasoning.** (a) Red points are visible, whereas blue points, captured from other views, are invisible in the reference view. (b) The invisibility of blue points may result from occlusions or insufficient sampling rays in the reference view. (c) Rasterizing the distances and indices of visible voxels (in green) yields dense depth maps and accurate pixel-voxel mapping.

4.1. Voxel-Based Visibility Reasoning

The effectiveness of 3DGS depends critically on well-initialized point clouds. In the training phase, photometric-guided optimization and densification aggregate all Gaussian primitives along each viewing ray via alpha compositing. However, when the initialized geometry is incomplete, photometric error gradients may erroneously accumulate on Gaussians associated with occluded or invisible surfaces, leading to mismatched appearance, geometric distortions, and floater artifacts [37]. To address this issue, explicit view-dependent visibility and occlusion reasoning for scene objects must be incorporated into the reconstruction process. As illustrated in Fig. 3, a 3D point sampled along a pixel ray should correspond to the first intersected surface, visible from the given viewpoint. Existing methods either extract visible points independently from each single view or aggregate points from all available views without considering occlusion. While sampling points from a specific view guarantees valid surface intersections, it provides limited scene coverage due to observation constraints. In contrast, aggregating multiple point clouds from different timestamps improves spatial coverage but lacks occlusion awareness, allowing rays to traverse occluded structures and resulting in erroneous updates on non-visible geometry.

To enable efficient reasoning about scene visibility and occlusion, in the first step, VAD-GS applies voxelization to the initialized point cloud to enforce uniform spatial density. The visibility of each voxel is defined as the union of the observation views associated with its constituent points. Specifically, each LiDAR point is sourced from an individual frame, whereas SfM points are triangulated from at least two views. Reasoning about voxel visibility provides two key advantages: (1) Rasterizing the distances of visible voxel surfaces via classical z-buffering [38] produces a denser and more reliable depth map, compared

to conventional geometric supervision methods that rely on sparse point clouds and nearest-neighbor search. This rasterized depth map reduces missed surface intersections, constrains depth errors within the voxel resolution, and naturally excludes occluded voxels located behind incomplete foreground geometry, thereby preventing erroneous updates during model optimization. (2) Rasterizing a 2D index map establishes a mapping between image pixels and their underlying 3D structures. By storing only the index of the first intersected and visible voxel along each pixel ray, this mapping ensures both validity and efficiency, enabling fast retrieval of geometric attributes such as 3D position, surface normals, and neighborhood connectivity.

Voxel visibility is further utilized to identify incomplete scene structures. Specifically, scene elements such as vehicles, trees, and buildings are individually extracted using an offline instance segmentation network [39]. Pixels belonging to each segmented instance are then mapped to their corresponding voxels according to the rasterized index map. For each instance, two depth values are compared: one rasterized from visible voxels via z-buffering, while the other rendered from existing Gaussians through the splatting pipeline. If the depth rendered from Gaussians is consistently smaller than the voxel-derived depth, it indicates either successful completion of previously missing geometry or acceptable redundancy, both of which can be effectively handled via opacity adjustment. In contrast, if the Gaussian-rendered depth is absent or significantly larger than the voxel depth, it implies that the geometry is either incompletely initialized or distorted in earlier optimization stages. In such cases, the instance is flagged for re-initialization to improve reconstruction completeness.

4.2. Diversity-Aware View Selection

After identifying instances with unreliable structures, the corresponding voxels along with available views can be retrieved using the rasterized index map. To ensure reliable geometry reconstruction via MVS, it is essential to select a representative subset of views that provides strong geometric constraints. Although consecutive video frames captured by a single camera can provide sufficient overlap for 3D reconstruction in static scenes [12], their performance deteriorates in driving scenarios due to the existence of both dynamic objects and continuous ego-motion. These factors necessitate sufficient frustum overlap and strong stereo constraints, which can only be achieved by selecting views from different cameras and timestamps. To this end, we define the following score:

$$s = \frac{N}{\boldsymbol{d}_R^\top \boldsymbol{d}_S} \frac{\sqrt{{t_x}^2 + {t_y}^2}}{|t_z| + \epsilon} \sin\theta, \tag{3}$$

to quantify the geometric diversity between a pair of views, where $\boldsymbol{d}_R$ and $\boldsymbol{d}_S$ are column vectors that store the distances from N voxels visible in both reference and supporting views to their respective viewpoints, $\boldsymbol{t} = (t_x, t_y, t_z)^\top$ denotes the relative translation between the two views, ϵ ensures numerical stability, and θ represents the angular difference between their orientations. Higher scores are achieved when voxels are denser and closer to both views, lateral variations are greater, longitudinal displacements are minimal, and orientation differences are larger. In contrast, lower scores result under opposite conditions, implying fewer stereo cues. Diverse supporting views selected based on this score are subsequently utilized for MVS reconstruction. To balance information diversity and computational cost, a top-k subset is selected for each reference view via (3). Pairwise diversity among the supporting views is also considered to avoid overly similar viewpoints.

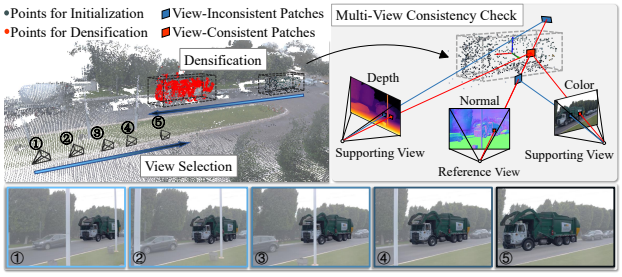

Figure 4. **View Selection and MVS Reconstruction**. Image patches are warped across views to check the consistency of depth, normal, and color. Only consistently matched patches (in red) are considered valid for MVS reconstruction, while inconsistent ones (in blue) are discarded. The reconstructed geometry is then used to guide Gaussian densification.

4.3. MVS Reconstruction from Selected Views

By incorporating the selected supporting views that provide sufficient geometric constraints, reliable 3D points can be generated to complement incomplete scene structures. To achieve this goal, we adopt a multi-view patch matching approach, which has been widely used for dense 3D reconstruction in static scenes [35]. The method estimates local surface planes by matching small image patches across multiple views, under the assumption that the scene geometry is locally piecewise planar. Specifically, an image pixel located at $\boldsymbol{p} = (u, v)^\top$ is associated with a local 3D plane, expressed as: $z\,\boldsymbol{n}^\top \boldsymbol{K}^{-1}\widetilde{\boldsymbol{p}} + d = 0$, where z denotes the depth value at $\boldsymbol{p}$, $\boldsymbol{n}$ represents the corresponding surface normal, $\boldsymbol{K}$ denotes the camera intrinsic matrix, $\widetilde{\boldsymbol{p}}$ represents the homogeneous coordinates of $\boldsymbol{p}$, and d represents the distance between the surface and the camera origin. In the patch matching process, $\boldsymbol{p}$ in the reference view, associated with a plane hypothesis $(d, \boldsymbol{n})$, is projected to $\boldsymbol{p}'$ in a supporting view using the following expression:

$$\widetilde{\boldsymbol{p}}' \simeq \boldsymbol{K}\left(\boldsymbol{R} - \frac{\boldsymbol{t}\boldsymbol{n}^\top}{d}\right)\boldsymbol{K}^{-1}\widetilde{\boldsymbol{p}}, \tag{4}$$

where $[\boldsymbol{R}, \boldsymbol{t}]$ denotes the relative pose from the reference view to the supporting view. Patches are warped across views to assess photometric consistency using RGB images and geometric consistency using depth and normal maps. As shown in Fig. 4, a pair of patches is deemed consistent if the reference patch and warped supporting patch exhibit similar image features, and their plane hypotheses $(d, \boldsymbol{n})$ are well aligned. These hypotheses are iteratively refined by propagating candidates from neighboring pixels under the assumption of local structural similarity. Gaussian-rendered results serve as reasonable initial guesses, effectively reducing the number of random re-sampling iterations required when consistent matches are absent. Through repeated updates and consistency checks, patches consistent across the majority of views are obtained for robust reconstruction.

Although reliable static structures can be recovered in this way [12, 35], handling dynamic objects remains a significant challenge. In theory, a rigid vehicle in motion can be treated as static by transforming all observation views into its local coordinate system. However, in practice, object masks derived from 3D bounding boxes cannot accurately delineate the boundaries between foreground and background. Including pixels from the static background may introduce misleading patch matches and disrupt the local geometric consistency assumptions for the moving object. Although instance segmentation methods provide more precise, contour-aligned masks, their performance is highly sensitive to the quality of input prompts. A prompt derived from a single LiDAR frame may fail to capture the entire object due to limited point coverage, whereas one generated from a temporally aggregated point cloud often introduces occluding scene structures unrelated to the object. This challenge, nevertheless, can be effectively addressed by leveraging our rasterized visible voxels, which inherently incorporate occlusion cues and provide more accurate instance-level prompts for segmentation. Consequently, the method restricts the patch matching process within either static or dynamic regions across all relevant views, effectively minimizing cross-region interference and greatly enhancing reconstruction accuracy.

4.4. Loss Function

We optimize the model by minimizing a weighted sum of four loss terms, as expressed as follows:

$$\mathcal{L} = \mathcal{L}_{\text{color}} + \lambda_{\text{normal}}\mathcal{L}_{\text{normal}} + \lambda_{\text{hard}}\mathcal{L}_{\text{hard}} + \lambda_{\text{soft}}\mathcal{L}_{\text{soft}}, \quad (5)$$

where $\mathcal{L}_{\text{color}}$ quantifies the discrepancy between rendered and observed images, $\mathcal{L}_{\text{normal}}$ quantifies the angular deviations between the rendered surface normals and those obtained via patch matching, and $\mathcal{L}_{\text{hard}}$ and $\mathcal{L}_{\text{soft}}$ measure depth errors using fixed and learned Gaussian opacities, respectively. By incorporating our visibility-aware Gaussian densification strategy, the optimization of (5) achieves superior geometric reconstruction performance in complex dynamic urban settings.

	PSNR↑	PSNR*↑	SSIM↑	LPIPS↓
3D GS [18]	29.64	21.25	0.918	0.117
NSG [40]	28.31	24.32	0.862	0.346
MARS [41]	29.75	26.54	0.886	0.264
EmerNeRF [42]	30.87	21.67	0.905	0.133
PVG [43]	31.82	24.68	0.910	0.122
OmniRe [19]	31.12	25.20	0.902	0.123
StreetGS [7]	34.61	30.23	0.938	0.079
Ours	**35.59**	**31.31**	**0.950**	**0.047**

Table 1. Quantitative results on Waymo Open Dataset. PSNR*: evaluated on dynamic objects only.

5. Experiments

5.1. Datasets

We conduct extensive experiments on two datasets: Waymo Open [13] and nuScenes [14]. The Waymo Open dataset comprises 1,150 driving scenes recorded in suburban and urban environments. Each frame contains images captured using five cameras and fused point clouds collected using five LiDARs, with an average of 177k points per frame. Following the study [7], we select eight sequences, each containing around 100 frames under dynamic traffic conditions. For the nuScenes dataset, we follow the study [19], which also provides baseline implementations of PVG, OmniRe, and StreetGaussians. The data are collected using a 32-beam LiDAR, resulting in significantly sparser point clouds (with an average of 34k points per frame) and more uneven spatial coverage. The increased sparsity poses greater challenges for Gaussian initialization and densification.

5.2. Implementation Details

Incomplete structures are identified and re-initialized when either the ratio between the rendered depth and the voxel rasterization result exceeds 1.1 or more than 25% of the pixels belonging to an instance have accumulated opacity below 0.7. The normal and depth loss weights are set to 0.02 and 0.1, respectively, while λ_{hard} is disabled after 80% of the training iterations. Supporting views are selected via maximum-weight k-clique optimization, which maximizes the total pairwise similarity among candidate views, where k is set to 4 in practice. To alleviate photometric overfitting caused by imbalanced view sampling during optimization, training views are sampled without replacement. Voxel visibility-based densification is performed every five complete sampling cycles. Enabled by seamless CUDA integration with the Gaussian training framework, each operation takes approximately 48 ms and is invoked only when missing structures are detected, adding negligible computational overhead. All experiments are conducted on a single

	PVG [43]				OmniRe [19]				StreetGS [7]				**VAD-GS (Ours)**			
	PSNR↑	SSIM↑	LPIPS↓	#G	PSNR↑	SSIM↑	LPIPS↓	#G	PSNR↑	SSIM↑	LPIPS↓	#G	PSNR↑	SSIM↑	LPIPS↓	#G
Scene 00	22.77	0.63	0.22	195k	22.71	0.63	0.16	153k	22.87	0.65	**0.13**	146k	**25.54**	**0.81**	0.16	234k
Scene 01	22.91	0.66	0.24	196k	22.27	0.64	**0.22**	108k	22.12	0.66	0.23	109k	**23.69**	**0.76**	0.24	200k
Scene 03	24.30	0.77	0.14	172k	24.56	0.76	**0.12**	133k	24.18	0.77	0.13	128k	**26.57**	**0.88**	0.13	148k
Scene 04	24.37	0.65	0.15	184k	24.32	0.64	**0.12**	162k	24.37	0.67	0.13	140k	**25.57**	**0.76**	0.21	245k
Scene 05	**20.76**	0.50	0.37	249k	19.80	0.46	0.29	155k	20.20	0.50	**0.28**	146k	20.06	**0.56**	0.30	329k
Scene 06	22.68	0.64	0.17	173k	23.08	0.67	**0.12**	177k	23.58	0.70	0.12	169k	**25.64**	**0.83**	0.16	179k

Table 2. Quantitative comparisons between VAD-GS and other SoTA approaches on the nuScenes dataset. "#G" denotes the number of Gaussian primitives. Scene 02 is excluded due to the stationary ego vehicle, which provides no additional supporting views. Scenes 07–09 are also omitted, as their extreme nighttime illumination conditions fall beyond the scope of this study. More details are in the supplement.



Figure 5. Qualitative comparisons between VAD-GS and other SoTA approaches on the nuScenes dataset.

NVIDIA GeForce RTX 4090 GPU. Please refer to the supplement and video for additional results.

5.3. Quantitative and Qualitative Results

We compare VAD-GS with several baseline approaches [7, 18, 19, 40–43]. Evaluation metrics, including the peak signal-to-noise ratio (PSNR), the structural similarity index measure (SSIM), and the learned perceptual image patch similarity (LPIPS), are used to quantify models' performance. As shown in Table 1, VAD-GS consistently outperforms all baseline methods across all evaluation metrics on the Waymo Open dataset. In particular, VAD-GS improves PSNR by ∼2.8% and PSNR* (evaluated on dynamic objects only) by ∼3.6%. These improvements can be primarily attributed to the complemented geometry, which effectively suppresses photometric distortions in erroneously exposed Gaussians. VAD-GS also outperforms the second-best method in terms of SSIM and LPIPS by 0.012 and 0.032, respectively, owing to its more complete and high-fidelity reconstruction of geometry and appearance, which in turn enhances photometric consistency. However, these improvements reflect the potential of our strategy only to a limited extent, as existing baselines report reconstruction results achieved using only the forward-facing camera from the Waymo Open dataset. To ensure fair comparison, we have to adopt the same settings, which restrict the exploitation of cross-camera cues, a key advantage of VAD-GS.

Completing missing geometry is less critical for the Waymo Open dataset, which provides high-quality point clouds for reliable Gaussian initialization, but becomes essential for the nuScenes dataset due to its significantly sparser LiDAR observations. Given that the difficulty of 3D reconstruction is highly scene-dependent and influenced by factors such as sampling trajectories, occlusions, and dynamic traffic behaviors, we additionally provide per-scene evaluation results on the nuScenes dataset, as shown in Table 2 and Fig. 5. Our method consistently outperforms baseline approaches in terms of SSIM, with improvements



(a) Ground Truth (b) w/o Voxel Visibility Reasoning (c) w/o View Selection (d) w/o Geometric Losses (e) Complete Model

Figure 6. Qualitative ablation study results.

Configurations	PSNR↑	PSNR*↑	SSIM↑	LPIPS↓
w/o voxel visibility reasoning	23.79	22.75	0.753	0.215
w/o view selection	23.92	22.83	0.757	0.212
w/o geometric losses	**24.59**	22.78	0.764	**0.194**
Complete model	24.51	**23.16**	**0.765**	0.199

Table 3. Quantitative ablation study results.

exceeding 0.06, and achieves a significant PSNR gain of over 0.78 dB across most scenes, except for Scene 05, where the ego vehicle follows a linear trajectory. This leads to sparse viewpoint sampling and limited overlap across camera views, making it challenging to observe objects from diverse perspectives. This results in limited cross-camera cues, making it effectively equivalent to independent single-camera reconstructions. Our method does not achieve the lowest LPIPS values in several scenes, primarily because the dynamic objects in these scenes are mostly moving pedestrians. Since MVS-based reconstruction methods generally assume object rigidity, their effectiveness degrades when handling deformable or non-rigid objects such as pedestrians. While the number of Gaussian primitives is not a direct indicator of reconstruction quality, it does reflect modeling efficiency to some extent. The slightly higher Gaussian count observed in our method results from targeted densification in underrepresented regions, rather than uncontrolled growth or redundant duplication in well-initialized geometry. While quantitative evaluation of structure completion is infeasible without ground truth for unseen trajectories, qualitative visualization from deviated test viewpoints (Fig. 1) demonstrates the effectiveness of our geometry recovery strategy.

5.4. Ablation Study

To validate the efficacy of each component, we conduct a comprehensive ablation study on the nuScenes dataset. Given the interdependence among components (voxel visibility reasoning, view selection, and geometric losses), we train three variants of VAD-GS, each omitting one of these components, and compare their performance against the complete model. Fig. 6 and Table 3 present the qualitative and quantitative results, respectively. The first variant removes the voxel-based visibility reasoning component, which consequently disables all other components. Although photometric-based densification remains active, it fails to accurately recover unreliable geometry. As shown in Fig. 6(b), this leads to incorrect gradient updates that distort Gaussian primitives, ultimately causing significant performance degradation. The second variant disables diversity-aware view selection and instead relies on fixed consecutive frames for patch matching. Although this improves densification in static regions, the recovery of missing geometry remains incomplete. Moreover, the absence of explicit separation between static and dynamic regions causes misleading matches between background and foreground geometries, leading to severe floater artifacts, as shown in Fig. 6(c). The third variant excludes geometric losses from the optimization objective. While it achieves comparable or even slightly better photometric metrics, attributed to the strong reliance on image similarity as the sole supervision signal, which encourages overfitting to visual appearance, this variant introduces noticeable artifacts under large viewpoint deviations, such as the rough vehicle surfaces observed in Fig. 6(d).

6. Conclusion and Future Work

This study presented VAD-GS, a novel 3DGS framework designed to enhance geometry recovery under sparse observations, particularly in dynamic, unbounded urban environments. Unlike prior Gaussian densification methods that exclusively clone or split existing Gaussians, VAD-GS reconstructs new Gaussians by leveraging stereo cues from multiple cameras across different timestamps, effectively recovering missing geometry for both static and dynamic objects. The framework explicitly models the view-dependent voxel visibility, enabling occlusion reasoning among scene instances and identification of regions requiring reconstruction. It then strategically selects supporting views based on a newly defined diversity score and generates complementary point clouds that satisfy multi-view consistency for objects even in motion. Extensive experiments on public datasets demonstrate the superiority. Despite the competitive performance in geometry completion, certain limitations persist. As a trade-off for relaxing static MVS assumptions in dynamic settings, the proposed method imposes a rigidity constraint. Consequently, in the absence of rigidity, densification for deformable objects degrades to a standard gradient-based strategy. Extending the approach to handle non-rigid motion is non-trivial and is left for future work.

Acknowledgment

This research was supported by the National Natural Science Foundation of China under Grants 62473288, 62233013, 62388101, and 62333017, the National Key Laboratory of Human-Machine Hybrid Augmented Intelligence, Xi'an Jiaotong University (No. HMHAI-202406), NIO University Programme (NIO UP), the Fundamental Research Funds for the Central Universities, and the Xiaomi Young Talents Program.

References

[1] Yanqi Bao et al. 3D Gaussian splatting: Survey, technologies, challenges, and opportunities. *IEEE Transactions on Circuits and Systems for Video Technology*, 35:6832–6852, 2025. 1, 2

[2] Alexey Dosovitskiy et al. CARLA: An open urban driving simulator. In *Conference on Robot Learning (CoRL)*, pages 1–16, 2017. 1

[3] Ben Mildenhall et al. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 3

[4] Bernhard Kerbl et al. A hierarchical 3D Gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics*, 43(4):1–15, 2024. 2

[5] Guanjun Wu et al. 4D Gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20310–20320, 2024. 2

[6] Dongxu Wei et al. Omni-Scene: Omni-Gaussian representation for ego-centric sparse-view scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22317–22327, 2025. 2

[7] Yunzhi Yan et al. Street Gaussians: Modeling dynamic urban scenes with Gaussian splatting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 156–173. Springer, 2024. 2, 3, 6, 7

[8] Xiaoyu Zhou et al. DrivingGaussian: Composite Gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21634–21643, 2024. 2, 3

[9] Hanqing Jiang et al. GeoTexDensifier: Geometry-texture-aware densification for high-quality photorealistic 3d gaussian splatting. *IEEE Transactions on Visualization and Computer Graphics*, 32(03):2671–2683, 2026. 2

[10] Jiahe Li et al. DNGaussian: Optimizing sparse-view 3D Gaussian radiance fields with global-local depth normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20775–20785, 2024. 2

[11] René Ranftl et al. Vision Transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12179–12188, 2021. 2

[12] Kai Cheng et al. GaussianPro: 3D Gaussian splatting with progressive propagation. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 8123–8140, 2024. 2, 5, 6

[13] Pei Sun et al. Scalability in perception for autonomous driving: Waymo Open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454, 2020. 2, 6

[14] Holger Caesar et al. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11621–11631, 2020. 2, 6

[15] Thomas Müller et al. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):1–15, 2022. 3

[16] Jonathan T Barron et al. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5855–5864, 2021. 3

[17] Jonathan T Barron et al. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5470–5479, 2022. 3

[18] Bernhard Kerbl et al. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023. 3, 4, 6, 7

[19] Ziyu Chen et al. OmniRe: Omni urban scene reconstruction. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 85508–85527, 2025. 3, 6, 7

[20] Hongyu Zhou et al. HUGSIM: A real-time, photo-realistic and closed-loop simulator for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(04):4673–4691, 2026. 3

[21] Rui Song et al. CoDa-4DGS: Dynamic Gaussian splatting with context and deformation awareness for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 28031–28041, 2025. 3

[22] Siddharth Tourani et al. Leveraging 2D priors and SDF guidance for dynamic urban scene rendering. *arXiv preprint arXiv:2510.13381*, 2025.

[23] Shengyu Huang et al. Neural LiDAR fields for novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 18236–18246, 2023.

[24] Yilong Li et al. Anchored 4D Gaussian splatting for dynamic novel view synthesis. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pages 1–11, 2025. 3

[25] Fangjinhua Wang et al. Learning-based multi-view stereo: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2026. doi: 10.1109/TPAMI.2026.3654665. 3

[26] Angela Dai et al. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5828–5839, 2017. 3

[27] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. 3

[28] Henrik Aanæs et al. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120:153–168, 2016.

[29] Thomas Schops et al. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3260–3269, 2017. 3

[30] Yuxin Hou et al. Multi-view stereo by temporal nonparametric fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2651–2660, 2019. 3

[31] Yao Yao et al. MVSNet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018. 4

[32] Shuo Cheng et al. Deep stereo using adaptive thin volume representation with uncertainty awareness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2524–2534, 2020. 4

[33] Jiayu Yang et al. Non-parametric depth distribution modelling based depth inference for multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8626–8634, 2022.

[34] Haoyu Guo et al. PlaneStereo: Plane-aware multi-view stereo. *Machine Intelligence Research*, 21(6):1092–1102, 2024. 4

[35] Qingshan Xu and Wenbing Tao. Planar prior assisted patchmatch multi-view stereo. In *Proceedings of the AAAI conference on artificial intelligence (AAAI)*, pages 12516–12523, 2020. 4, 5, 6

[36] Fangjinhua Wang et al. PatchmatchNet: Learned multi-view patchmatch stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14194–14203, 2021. 4

[37] Zheng Zhang et al. Pixel-GS: Density control with pixel-aware gradient for 3D Gaussian splatting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 326–342. Springer, 2024. 4

[38] Tomas Akenine-Möller et al. *Real-Time Rendering*. CRC Press, Boca Raton, FL, fourth edition, 2018. 4

[39] Alexander Kirillov et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 5

[40] Julian Ost et al. Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2856–2865, 2021. 6, 7

[41] Zirui Wu et al. MARS: An instance-aware, modular and realistic simulator for autonomous driving. In *Proceedings of the CAAI International Conference on Artificial Intelligence (CICAI)*, pages 3–15, 2023. 6

[42] Jiawei Yang et al. EmerNeRF: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023. 6

[43] Yurui Chen et al. Periodic Vibration Gaussian: Dynamic urban scene reconstruction and real-time rendering. *International Journal of Computer Vision*, 134(3):83, 2026. 6, 7