Omni-Scene: Omni-Gaussian Representation for Ego-Centric Sparse-View Scene Reconstruction

Dongxu Wei[1,3], Zhiqi Li[1,2], and Peidong Liu[1*]

[1]School of Engineering, Westlake University
[2]College of Computer Science and Technology, Zhejiang University
[3]Institute of Advanced Technology, Westlake Institute for Advanced Study
{weidongxu, lizhiqi49, liupeidong}@westlake.edu.cn
https://wswdx.github.io/omniscene

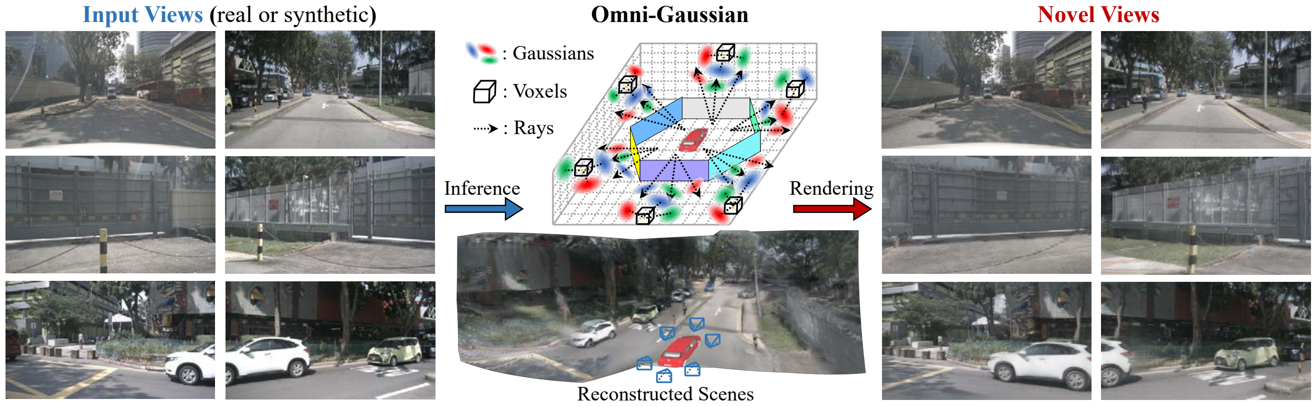

Figure 1. Provided with six surrounding images captured in real world or synthesized by 2D diffusion models, we can generate high-quality 3D Gaussians based on our Omni-Gaussian representation for ego-centric scene reconstruction and novel view synthesis.

Abstract

Prior works employing pixel-based Gaussian representation have demonstrated efficacy in feed-forward sparse-view reconstruction. However, such representation necessitates cross-view overlap for accurate depth estimation, and is challenged by object occlusions and frustum truncations. As a result, these methods require scene-centric data acquisition to maintain cross-view overlap and complete scene visibility to circumvent occlusions and truncations, which limits their applicability to scene-centric reconstruction. In contrast, in autonomous driving scenarios, a more practical paradigm is ego-centric reconstruction, which is characterized by minimal cross-view overlap and frequent occlusions and truncations. The limitations of pixel-based representation thus hinder the utility of prior works in this task. In light of this, this paper conducts an in-depth analysis of different representations, and introduces Omni-Gaussian representation with tailored network design to complement their strengths and mitigate their drawbacks. Experiments show that our method significantly surpasses state-of-the-art methods, pixelSplat and MVSplat, in ego-centric reconstruction, and achieves comparable performance to prior works in scene-centric reconstruction.

1. Introduction

Reconstructing 3D scenes from sparse observations is a crucial task in computer vision and graphics. Recent efforts [1–22] have integrated 3D structural priors as inductive biases into neural networks, enabling the prediction of implicit neural field [23], light field [10], or explicit 3D Gaussians [24] for scene reconstruction in a single forward pass. Notably, due to the efficiency of rasterization-based rendering and the explicit nature of 3D Gaussians [24], Gaussian-based methods [13–22] have shown superiority in both inference speed and visual quality compared to those based on neural field [1–9] or light field [10–12]. Typically, these methods assume existence of large overlaps among the observed input views. Thus they can utilize techniques such as

[*] Corresponding author

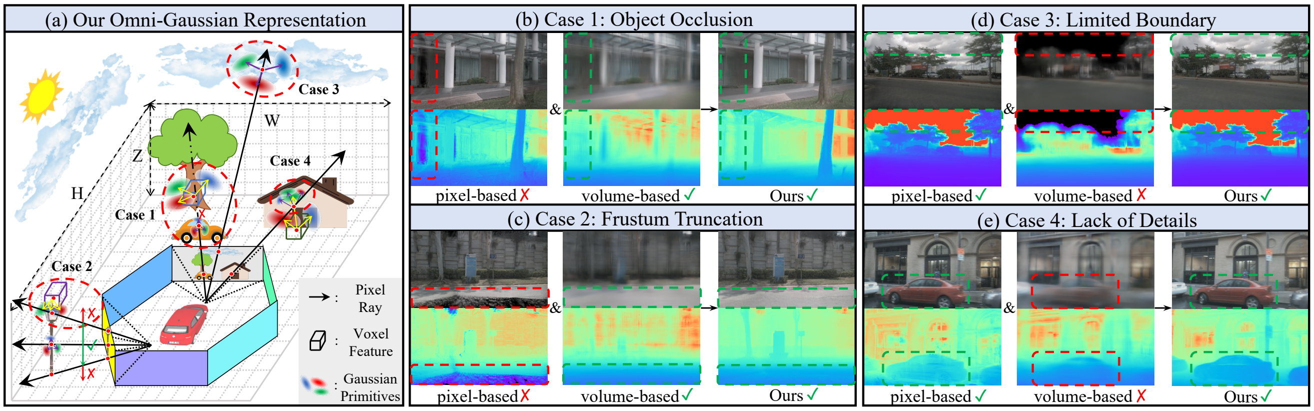

Figure 2. Illustration of our Omni-Gaussian representation. Our Omni-Gaussian incorporates two representations, pixel-based and volume-based Gaussians. In (a), we illustrate bad cases when relying solely on one representation (i.e., Case 1 and 2 for pixel-based, Case 3 and 4 for volume-based), and how we use the other one to compensate for the shortcomings. In (b)-(e), we present examples of these cases under the task of ego-centric driving scene reconstruction. *Green dashed lines* denote areas plausibly rendered in novel views, while *red ones* highlight undesirable artifacts due to weaknesses of pixel-based or volume-based Gaussian. We can observe that Omni-Gaussian leveraging the complementary nature of the two representations can achieve optimal results for all cases.

multi-view cross attention [16, 18, 22], epipolar lines [13] or cost volumes [14, 21] to learn pixel-level cross-view correlation, and then infer per-pixel depths with proper scales. Hence they can further predict per-pixel Gaussians and use depths to unproject them to 3D along pixel rays for scene reconstruction. A common feature for all of these methods is the use of pixel-based Gaussian representation.

Although works utilizing the pixel-based Gaussian representation have achieved great success, they pose strong hypothesis regarding the existence of large overlaps among input views. This implies the necessity of capturing input-view images encircling the scene. Otherwise, they would fail to predict accurate per-pixel depths due to the scale ambiguity [13]. In contrast to such scene-centric reconstruction, a more practical case especially for autonomous driving systems, is ego-centric reconstruction, where we can only acquire input views from cameras rigidly mounted around the car, with minimal overlaps ($<$15%) existing only between adjacent cameras. As evident in Sec.4.2, previous methods with pixel-based representation would fail for ego-centric reconstruction. Despite the difficulty in predicting per-pixel depths, their failure can be attributed to two underlying weaknesses inherent in the pixel-based representation as showcased by Case 1 and 2 in Fig.2. In Case 1, when object in the target novel view is occluded in the input view (e.g., tree behind the car in Fig.2(a)), pixel-based representation can only rely on 2D local features of the non-occluded object for inferring the occluded one, which fails especially when their appearances are far different from each other. In Case 2, when object in the novel view falls outside of the input view frustum (e.g., top of the streetlight in Fig.2(a)), pixel-based representation cannot predict positions of Gaussians through unprojection along pixel rays.

These two cases also pose challenges for 3D perception tasks like 3D object detection [25–27] and occupancy prediction [28–30], which require to perceive partially occluded or truncated objects. Existing 3D perception works resort to volume-based representations like bird's eye view (BEV) grids [25–27] and 3D voxels [28–30] as the solution. Since volume is spatially-continuous in 3D space, contents absent in the 2D inputs can be supplemented by their neighbors at the 3D level. Besides, with camera projection knowledge to enable 3D-to-2D cross attention [25], we can directly lift 2D features to 3D space instead of relying on cross-view overlap for depth-based 2D-to-3D unprojection. Intuitively, we conjecture that we can utilize a volume-based Gaussian representation in the reconstruction task (i.e., represent Gaussians with voxels in the volume) to minimize dependence on cross-view overlap and mitigate bad effects brought by occlusions and truncations. However, as illustrated by Case 3 and 4 in Fig.2, this representation also has drawbacks. Due to the bounded nature of volume (i.e., bounded within the range of $H\times W\times Z$ around the car), volume-based Gaussian cannot reconstruct elements far away from the car (e.g., sky in Case 3 of Fig.2(a)). Besides, encoding features for a volume with cubic complexity limits the volume resolution, potentially resulting in the lack of details (e.g., house in Case 4 of Fig.2(a)).

In this paper, considering limitations of pixel and volume-based Gaussians, we propose Omni-Scene, which employs Omni-Gaussian representation and tailored network designs to reach the best of both worlds. *The core lies in how to optimize volume and pixel-based Gaussians to their full potential, and leverage their unique attributes to enable their collaboration.* Specifically, for volume-based Gaussian, we propose Volume Builder composed of Triplane Transformer and Volume Decoder to reconstruct coarse 3D structures with voxel-anchored Gaussians. In particular, our Triplane Transformer uses triplane as a light-weight alternative of volume, where we employ cross-image and cross-plane deformable attentions to enhance volumetric feature encoding. For pixel-based Gaussian, we propose Pixel Decorator, which complements volume-based Gaussian with distant elements and better details. Our Pixel Decorator comprises Multi-View U-Net and Pixel Decoder, responsible for cross-view attended feature extraction and per-pixel Gaussian prediction, respectively. To enable collaboration between the two representations, we introduce Projection-Based Feature Fusion and Depth-Guided Training Decomposition for their seamless fusion and better complementarity, thereby boosting the performance. In summary, our main contributions are as follows:

- We propose Omni-Scene, an Omni-Gaussian representation with tailored network design for ego-centric reconstruction, taking advantages of both pixel and volume-based representations while eliminating their drawbacks.
- We introduce a novel ego-centric reconstruction task to a popular driving dataset (i.e., nuScenes [31]), with the aim of scene-level 3D reconstruction and novel view synthesis given only single-frame surrounding images. We hope this can facilitate further research in this field.
- Experiments show that our method significantly outperforms state-of-the-art feed-forward reconstruction methods including pixelSplat [13] and MVSplat [14] on the ego-centric task. We also achieve competitive performance with prior works on the scene-centric task performed on RealEstate10K dataset [32].

2. Related Work

Neural Reconstruction and Rendering. Recent approaches [23, 24, 33–37] leveraging neural rendering and reconstruction techniques can model scenes as learnable 3D representations, and achieve 3D reconstruction and novel view synthesis through iterative back propagation. NeRF [23] has been recognized for its ability to capture high-frequency details in reconstructed scenes. However, it requires dense queries for each ray during rendering, which, despite subsequent efforts for acceleration [34, 35], still results in high computational demand that limits its real-time capability. 3D Gaussian Splatting (3DGS) [24] mitigates this issue by explicitly modeling scenes with 3D Gaussians and employing an efficient rasterization-based rendering pipeline. Although 3DGS and NeRF, along with their variants [33, 36, 38, 39], have demonstrated superior performance in single-scene reconstruction, they usually require per-scene optimization and dense scene captures, making the reconstruction process time-consuming and unscalable. Different from these works, our method can reconstruct 3D scenes from sparse observations in a single forward pass.

Feed-Forward Reconstruction with Implicit 3D Representations. This line of works incorporate implicit 3D priors, such as NeRF [23] or light field [10], into their networks to achieve feed-forward reconstruction. NeRF-based methods [1–9] leverage transformers with multi-view cross attentions [8, 16, 18, 22], or employ projective 3D priors like epipolar lines [1–3, 13] and cost volumes [4–7, 14, 21] to estimate radiance fields for reconstruction, which inherits the expensive ray querying process of NeRF rendering. Consequently, these methods are exceedingly time-consuming during both training and inference phases. In contrast, light field-based approaches [10–12] can bypass NeRF rendering by directly regressing per-ray colors based on ray-to-image cross attentions, which sacrifices interpretability for efficiency. However, due to the lack of interpretable 3D structure, they fail to reconstruct 3D geometries of scenes.

Feed-Forward Reconstruction with 3D Gaussians. Recent methods [13–22] utilizing 3DGS can achieve both interpretability and efficiency. Typically, they adopt 3D priors akin to NeRF-based methods (e.g., epipolar lines [13], cost volumes [14, 21] and multi-view cross attentions [16, 18, 22]) into their networks, and employ pixel-based Gaussian representation to predict per-pixel Gaussians along the rays for reconstruction. However, such pixel-based representation depends on large cross-view overlap to predict depths, and suffers from object occlusion and frustum truncation, thus only suits for scene-centric reconstruction with limited applicability. In contrast, this paper concentrates on ego-centric reconstruction, which is characterized by minimal cross-view overlap and frequent occurrences of object occlusion and frustum truncation. This has motivated our research into a novel 3D representation that is not overly dependent on cross-view overlap, and can address the limitations of pixel-based representation in the meantime.

Neural Representation in 3D Perception. Similar to 3D reconstruction from multi-view images, 3D perception works [25–30, 40] also utilize multi-view images as input and perform 3D perception tasks like 3D detection [25–27], map segmentation [25, 26, 40], and 3D occupancy prediction [28–30]. Early 3D perception attempts [26, 27, 40] like Lift-Splat-Shoot (LSS)[40] employ pixel-wise depths to unproject pixel-wise features to 3D along camera rays, and project them onto BEV plane to enable 3D-level estimation. Similar to pixel-based representation in 3D reconstruction, such pixel-based approach would fail in cases of object occlusion. Recent 3D perception methods [25, 28–30] manage to bypass pixel-wise unprojection that sensitive to occlusion. In particular, they directly encode feature at 3D level by employ volume-based representation (e.g., BEV grids [25] or 3D voxels [28–30]), and achieves better performance especially when some of the objects are occluded by those closer to the camera. Although these methods show potential to accurate 3D perception, the perception task itself is much more coarse-grained compared to 3D reconstruction task, making low-resolution volume sufficient for perception-oriented feature modeling. In contrast, this paper focuses on 3D reconstruction task that re-

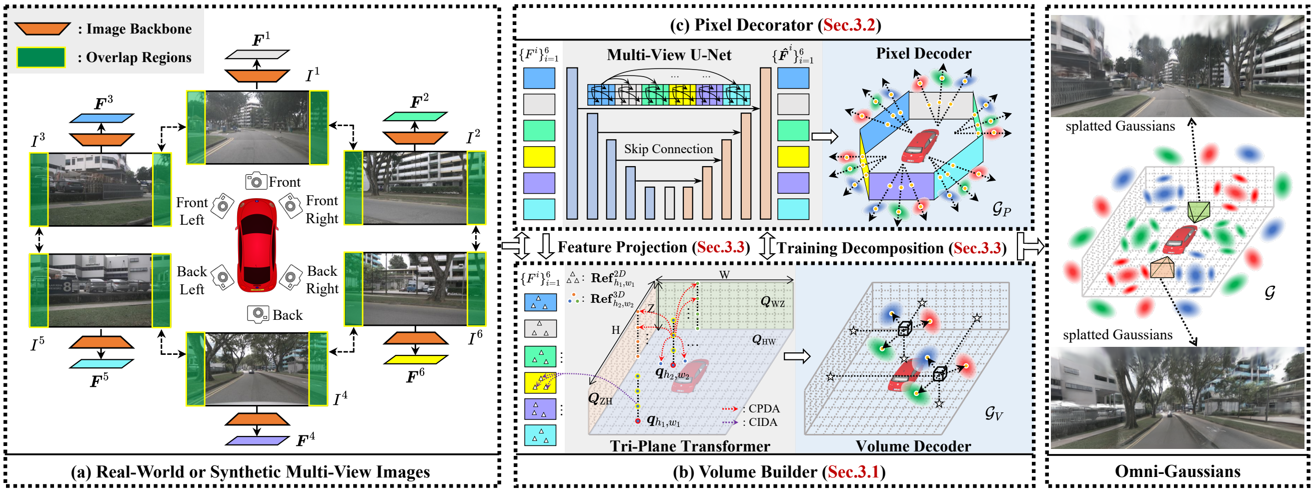

Figure 3. Overview. (a) Obtain images $\{\boldsymbol{I}^i\}_{i=1}^K$ from surrounding cameras with minimal overlap (e.g., adjacent image areas enclosed by *green rectangles*) in a single frame, and extract 2D features using image backbone. (b) For Volume Builder, we first use Triplane Transformer to lift 2D features $\{\boldsymbol{F}^i\}_{i=1}^K$ to 3D volume space compressed by three orthogonal planes, where we employ cross-image and cross-plane deformable attentions to enhance feature encoding. Then, Volume Decoder takes voxels as anchors, and predict nearby Gaussians $\mathcal{G}_V$ for each voxel given features sampled from the three planes through bilinear interpolation. (c) For Pixel Decorator, we use Multi-View U-Net to propagate information across views and extract multiple 2D features for Pixel Decoder to predict pixel-based Gaussians $\mathcal{G}_P$ along rays. Through Volume-Pixel Collaborations including Projection-Based Feature Fusion and Depth-Guided Training Decomposition, we can make $\mathcal{G}_V$ and $\mathcal{G}_P$ complement for each other, and obtain the full Omni-Gaussians $\mathcal{G}$ for novel-view rendering.

quires fine-grained feature modeling, which exceeds the capability of volume representation.

3. Method

The overall pipeline of Omni-Scene, a feed-forward approach to ego-centric sparse-view reconstruction, is shown in Fig.3. As depicted in Fig.3(a), we accept K surrounding images $\mathcal{I} = \{\boldsymbol{I}^i\}_{i=1}^K$ as inputs, which are captured or synthesized within a single frame. We utilize a ResNet-50 [41] backbone pre-trained with DINO objective [42] to extract $4\times$ downsampled features $\mathcal{F} = \{\boldsymbol{F}^i\}_{i=1}^K$ for $\mathcal{I}$. Then, as detailed in Fig.3(b)-(c), the features are shared and fed into our Volume Builder (Sec.3.1) and Pixel Decorater (Sec.3.2) to predict volume-based Gaussians $\mathcal{G}_V$ and pixel-based Gaussians $\mathcal{G}_P$, respectively. Utilizing Volume-Pixel Collaboration designs (Sec.3.3) including Projection-Based Feature Fusion and Depth-Guided Training Decomposition, we enable feature interaction between $\mathcal{G}_V$ and $\mathcal{G}_P$, and distinguish their attributes during training. By fusing $\mathcal{G}_V$ and $\mathcal{G}_P$, we can obtain Omni-Gaussians $\mathcal{G}$ for reconstruction.

3.1. Volume Builder

Our Volume Builder aims to predict coarse 3D structures with volume-based Gaussians. The primary challenge is how to lift 2D multi-view image features to the 3D volume space without explicitly maintaining dense voxels. We address this using Triplane Transformer. Then, Volume Decoder is proposed to predict voxel-anchored Gaussians $\mathcal{G}_V$.

Triplane Transformer. Representing volume as voxels and encoding features for each is expensive due to the cubic complexity of $H\times W\times Z$. Therefore, we resort to triplane to disentangle volume into three axis-aligned orthogonal planes HW, ZH and WZ. Some object-level 3D reconstruction works [8, 43, 44] also adopt triplane representation to compress volume. However, they either rely on dense per-pixel cross attention between triplanes and images [43, 44], or require input images to be also axis-aligned with triplanes [8] for direct 2D-level feature encoding. Neither of them is suitable for real-world scenes with much larger volume scales and unconstrained data collection.

Inspired by recent 3D perception methods [25, 30] that replace global full-image attention with local deformable attention to efficiently lift information from 2D to 3D, our Triplane Transformer also utilize deformable attention to enable sparse but effective spatial correlations between 2D and 3D spaces. Here we take the feature encoding of HW plane as an example for explanation. As shown in Fig.3(b), we define a group of grid-shaped learnable embeddings $\boldsymbol{Q}_{HW} \in \mathbb{R}^{H\times W\times C}$ as the plane queries of transformer, where C denotes the embedding channels. Then, for query $\boldsymbol{q}_{h,w}$ positioned at (h, w), we expand it to multiple 3D pillar points evenly spread along the Z axis, and calculate their reference points $\mathbf{Ref}^{2D}_{h,w}$ in 2D space by projecting them back to the input views. Due to the sparse nature of such perspective projection, only the most relevant 2D features from 1∼2 input views will be attended for $\boldsymbol{q}_{h,w}$, balancing efficiency and feature expressiveness. The above operation, namely Cross-Image Deformable Attention (CIDA), is denoted by *purple dashed arrows* in Fig.3(b). We derive it as follows:

$$\boldsymbol{q}^{CIDA}_{h,w} = \frac{1}{K'}\sum_{i=1}^{K'}\mathrm{DA}(\boldsymbol{q}_{\mathrm{h,w}}, \mathbf{Ref}^{\mathrm{2D}}_{\mathrm{h,w,i}}, \boldsymbol{F}_{\mathrm{i}}), \tag{1}$$

where K', $\mathbf{Ref}^{2D}_{h,w,i}$, DA represent the number of correlated views, 2D reference points in the i-th correlated view and deformable attention function, respectively.

Considering query pillar points might be occluded or located beyond the frustum range for any of the input views, we further utilize Cross-Plane Deformable Attention (CPDA) to enrich these points with cross-plane context. In particular, for query $\boldsymbol{q}_{h,w}$, we project its coordinate (h,w) onto the HW, ZH and WZ planes to obtain three sets of reference points $\mathbf{Ref}^{3D}_{h,w} = \mathbf{Ref}^{HW}_{h,w} \cup \mathbf{Ref}^{ZH}_{h,w} \cup \mathbf{Ref}^{WZ}_{h,w}$. Here, $\mathbf{Ref}^{HW}_{h,w}$ denotes neighbors of $\boldsymbol{q}_{h,w}$ within the HW plane. $\mathbf{Ref}^{ZH}_{h,w}$ and $\mathbf{Ref}^{WZ}_{h,w}$ are orthogonal projections onto the ZH and WZ planes, derived from pillar points of (h,w) evenly sampled along the Z axis. Utilizing $\mathbf{Ref}^{3D}_{h,w}$, we extract contextual information from different planes, thereby enhancing the features as denoted by *red dashed arrows* in Fig.3(b). We derive it as follows:

$$\boldsymbol{q}^{CPDA}_{h,w} = \mathrm{DA}(\boldsymbol{q}_{\mathrm{h,w}}, \mathbf{Ref}^{\mathrm{3D}}_{\mathrm{h,w}}, \boldsymbol{Q}_{\mathrm{HW}}, \boldsymbol{Q}_{\mathrm{ZH}}, \boldsymbol{Q}_{\mathrm{WZ}}), \tag{2}$$

where $\boldsymbol{Q}_{ZH}, \boldsymbol{Q}_{WZ}$ denote queries of the other two planes.

Repeating these two cross attentions for queries of all the planes, we can obtain triplane feature with rich semantic and spatial context without dependency on cross-view overlap, which is necessary for previous approaches [13, 14] that solely relied on pixel-based Gaussian representation.

Volume Decoder. Our Volume Decoder is then proposed to estimate voxel-anchored Gaussians. Specifically, given a voxel located at (h,w,z), we first project its coordinate onto the three planes to obtain plane features through bilinear interpolation, which is followed by plane-wise summation to derive the aggregated voxel feature $\boldsymbol{f}_{h,w,z}$. Then, we append three linear layers to $\boldsymbol{f}_{h,w,z}$ to predict parameters $(\boldsymbol{\delta}_v, \boldsymbol{\alpha}_v, \boldsymbol{s}_v, \boldsymbol{q}_v, \boldsymbol{c}_v)\}^{V}_{v=1}$ for V Gaussians $\{\boldsymbol{G}^v\}^{V}_{v=1}$. Each gaussian $\boldsymbol{G}^v$ is anchored near (h,w,z) and shifted to a new position $\boldsymbol{\mu}_v$ according to the offset $\boldsymbol{\delta}_v \in \mathbb{R}^3$. The remaining parameters $\boldsymbol{\alpha}_v$, $\boldsymbol{s}_v$, $\boldsymbol{q}_v$, $\boldsymbol{c}_v$ denote opacity, scale, rotation quaternion and RGB color, respectively. The same operation is repeated for all the voxels to obtain our volume-based Gaussians $\mathcal{G}_V \in \mathbb{R}^{H\times W\times Z\times V\times D}$, where D is the dimension of Gaussian paramters.

3.2. Pixel Decorator

Our pixel decorator consists of Multi-View U-Net and Pixel Decoder, responsible for extracting cross-view correlated features and predicting pixel-based Gaussians $\mathcal{G}_P$, respectively. Since $\mathcal{G}_P$ is obtained in alignment with fine-grained image space, it can add details to coarse voxel-anchored Gaussians $\mathcal{G}_V$. Besides, since $\mathcal{G}_P$ can be unprojected to positions at infinite distance, it can supplement volume-bounded $\mathcal{G}_V$ with distant Gaussians.

Multi-View U-Net. The Multi-View U-Net concatenates image features $\{\boldsymbol{F}^i\}^{K}_{i=1}$ and Plücker ray embeddings $\{\boldsymbol{S}^i\}^{K}_{i=1}$ as inputs, where $\{\boldsymbol{S}^i\}^{K}_{i=1}$ can provide additional camera pose information [16]. Inspired by the patchified token compression introduced by a recent 2D diffusion transformer method [45], we apply patchified cross attentions to our Multi-View U-Net for efficient cross-view correlation as shown in Fig.3(c). Then, we can obtain 3D-aware features $\{\hat{\boldsymbol{F}}^i\}^{K}_{i=1}$ for each input view to decode Gaussians.

Pixel Decoder. Our Pixel Decoder first upsamples the U-Net features $\{\hat{\boldsymbol{F}}^i\}^{K}_{i=1}$ to the original image resolution through bilinear interpolation, followed by several convolution layers to decode per-pixel depth $\boldsymbol{d}_p$ and Gaussian parameters $(\boldsymbol{\delta}_p, \boldsymbol{\alpha}_p, \boldsymbol{s}_p, \boldsymbol{q}_p, \boldsymbol{c}_p)$ for each Gaussian $\boldsymbol{G}^p$. To obtain the center position $\boldsymbol{\mu}_p$, we first use $\boldsymbol{d}_p$ to unproject the pixel from the ray origin $\boldsymbol{o}_p$ to a coarse position along the ray direction $\boldsymbol{r}_p$, and then refine it with the learned offset $\boldsymbol{\delta}_p \in \mathbb{R}^3$. The unprojection process is derived as follows:

$$\boldsymbol{\mu}_p = \boldsymbol{o}_p + \boldsymbol{d}_p \cdot \boldsymbol{r}_p + \boldsymbol{\delta}_p. \tag{3}$$

Moreover, compared to predicting $\boldsymbol{d}_p$ from scratch, we find replacing it with the noisy estimation of a 2D foundation model [46] is beneficial for the performance, demonstrating the importance of Gaussian initialization [24]. By doing the same to pixels of all the input views, we can obtain the pixel-based Gaussians $\mathcal{G}_P \in \mathbb{R}^{K\times R\times D}$, where R is the total number of rays in an input view.

3.3. Volume-Pixel Collaboration

The core of Omni-Gaussian representation lies in the collaboration of volume and pixel-based Gaussian representations. For this purpose, we propose a dual approach that enables the collaboration from two aspects: Projection-Based Feature Fusion and Depth-Guided Training Decomposition.

Projection-Based Feature Fusion. Our Volume Builder is expected to predict Gaussians at positions occluded or truncated in input views, which exceeds the design purpose of Pixel Decorator. Therefore, to make Volume Builder aware of where the occlusion or truncation occurs, we propose to fuse the triplane queries $\boldsymbol{Q}_{HW}, \boldsymbol{Q}_{ZH}, \boldsymbol{Q}_{WZ}$ with projected features of pixel-based Gaussians $\mathcal{G}_P$. Taking the plane of HW as an example, we first filter out Gaussians fallen beyond the volume range of $H\times W\times Z$ for $\mathcal{G}_P$. Then, we collect U-Net features for the remaining Gaussians of $\mathcal{G}_P$ and project them onto the HW plane. Features projected to the same query positions are averaged pooled and added to the corresponding query of $\boldsymbol{Q}_{HW}$ after a linear layer transformation. The same process is applied to the ZH and WZ planes. We demonstrate in our experiments (Sec.4.3) that such feature fusion facilitates a complementary interaction between $\mathcal{G}_V$ and $\mathcal{G}_P$, thereby enhancing performance.

Depth-Guided Training Decomposition. To further strengthen the collaboration, we propose a Depth-Guided Training Decomposition method to decompose our training objective based on the distinct spatial attributes of pixel and volume-based Gaussians. Specifically, due to the lim-

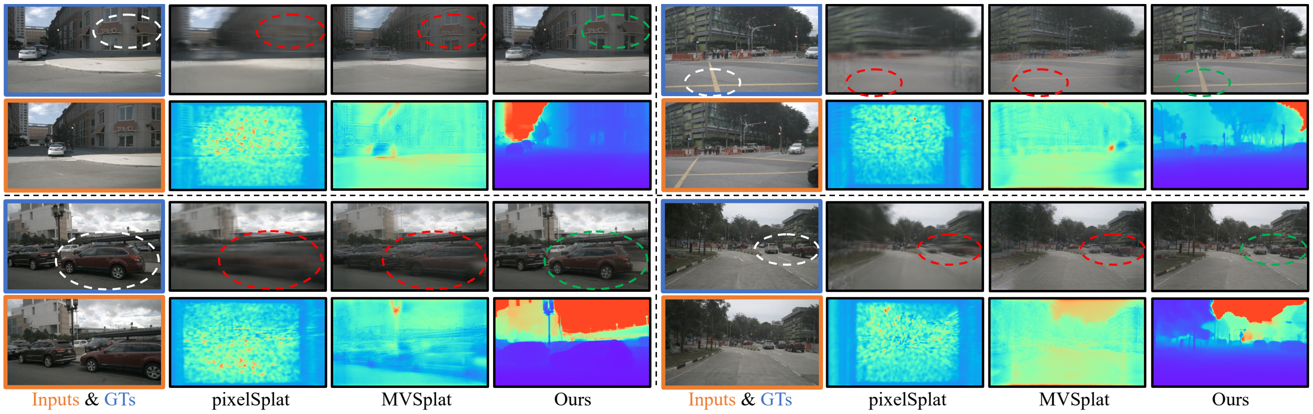

Figure 4. Comparisons on nuScenes [31]. Images of input views (Inputs) and ground-truth novel views (GTs) are outlined by orange and blue rectangles, respectively. The remaining are generated novel views and depth maps (warmer colors denote greater distance while the opposite for cooler colors). The *red dashed circles* denote undesirable artifacts, while the *green ones* denote plausibly-rendered areas.

ited volume range, $\mathcal{G}_V$ is not supposed to reconstruct distant elements, which should be supplemented by $\mathcal{G}_P$ that has no distance limitation. To achieve that, we first use $\mathcal{G}_P$ to render depth maps $\hat{\boldsymbol{D}} = \{\hat{\boldsymbol{D}}^i\}_{i=1}^K$ for all of the K input views. Then, we obtain 3D positions for all the pixels by unprojecting them along ray directions according to the estimated depths. By assigning pixels located within the range of $H \times W \times Z$ to 1, and assigning the remaining pixels to 0, we can obtain masks $\hat{\boldsymbol{M}} = \{\hat{\boldsymbol{M}}^i\}_{i=1}^K$. To enable appropriate supervision for volume-based Gaussian, we use $\hat{\boldsymbol{M}}$ to calculate masked photometric losses (i.e., mean squared error L_V^{mse} and LPIPS loss L_V^{lpips} [47]) as well as masked L1 depth loss L_V^{dpt} for input-view images and depths rendered from $\mathcal{G}_V$, where only pixels with mask values equal to 1 will be used for loss calculation. Note that L_V^{dpt} denotes L1 errors between depths independently rendered from $\mathcal{G}_P$ and $\mathcal{G}_V$, which aims to align $\mathcal{G}_P$ and $\mathcal{G}_V$ to the same scale, and requires no external depth signals for supervision. Combining with the photometric losses L_{full}^{mse} and L_{full}^{lpips} for novel-view images rendered from our full Gaussians $\mathcal{G} = \mathcal{G}_V \cup \mathcal{G}_P$, the overall training objective L can be derived as follows:

$$\begin{aligned} L &= L_{full}^{mse} + \lambda_1 L_{full}^{lpips} + \lambda_2 L_V, \\ L_V &= L_V^{mse} + \lambda_{V_1} L_V^{lpips} + \lambda_{V_2} L_V^{dpt}, \end{aligned} \tag{4}$$

where λ_1 and λ_2 are weights for LPIPS loss of $\mathcal{G}$ and composite loss of $\mathcal{G}_V$, respectively. λ_{V_1} and λ_{V_2} are weights for LPIPS and depth losses of $\mathcal{G}_V$, respectively.

4. Experiments

4.1. Experimental Setup

We conduct experiments for both ego-centric and scene-centric sparse-view reconstruction tasks. The ego-centric task is performed on nuScenes dataset [31], where the large motions and dense traffics in driving scenes pose more challenges for reconstruction. The scene-centric task is performed on RealEstate10K dataset [32] following protocols presented in previous works [13, 14].

Dataset	Method	PSNR↑	SSIM↑	LPIPS↓	PCC↑
nusc [31]	pixelSplat [13]	21.51	0.616	0.372	0.001
	MVSplat [14]	21.61	0.658	0.295	0.181
	Ours	**24.27**	**0.736**	**0.237**	**0.800**
re10k [32]	AttnRend [12]	24.78	0.820	0.213	N/A
	MuRF [9]	26.10	0.858	0.143	0.344
	pixelSplat [13]	25.89	0.858	0.142	0.285
	MVSplat [14]	**26.39**	**0.869**	**0.128**	0.363
	Ours	26.19	0.865	0.131	**0.368**

Table 1. Quantitative results on nuScenes [31] and RealEstate10K [32]. We **bold 1st-place** results and underline 2nd-place results. PCC is not available (N/A) for light field-based method AttnRend which has no interpretable 3D structure for depth rendering.

Ego-Centric Task. The nuScenes dataset comprises 700 scenes for training and 150 scenes for validation, with each containing a video of approximately 20 seconds captured at 12 Hz. We partition each scene into equally spaced bins along the vehicle trajectories, with a 3.2m interval between the first and the last captured frames. The central frame of each bin, featuring 6 surround-view images, serves as input views, while the first and the last frames, comprising 12 images, constitute the target novel views. Thus, we obtain 135,941 bins used for training and 30,080 bins for validation in total. We adopt the image resolution of 224×400 in our experiments for compatibility with the 2D diffusion model [48]. For evaluation, we compare our method against pixelSplat [13] and MVSplat [14], both are state-of-the-art methods for feed-forward sparse-view reconstruction.

Scene-Centric Task. To further evaluate our method against prior works, we also conduct experiments on RealEstate10K, a large-scale scene-centric dataset containing both indoor and outdoor scenes. Following the protocols adopted by previous works [13, 14], we use 67,477 scenes for training and 7,289 scenes for testing. For evaluation, we conduct comprehensive comparisons with previous methods including 3DGS-based pixelSplat [13] and MVSplat [14], light field-based AttnRend [12], and NeRF-based

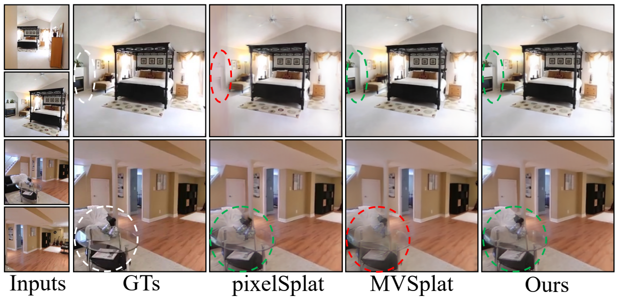

Figure 5. Comparisons on RealEstate10K [32]. The *red dashed circles* denote undesirable artifacts, while the *green ones* denote plausibly-rendered areas.

MuRF [9]. The results of these methods are adopted from their papers directly.

Metrics. To measure visual quality, we use three metrics: peak signal-to-noise-ratio (PSNR), structural similarity (SSIM) [49], and perceptual distance (LPIPS) [47]. For PSNR and SSIM, larger values are preferable, whereas the opposite for LPIPS. To further assess geometric quality of 3D scenes, we compare rendered depth maps of novel views with those predicted by [50], which has been demonstrated to produce highly accurate and robust depth predictions in real-world scenarios. Since [50] can only obtain relative depths without scales, we use Pearson Correlation Coefficient (PCC) [51] as the scale-invariant metric, which quantifies the statistical relationship between any two variables. The PCC ranges from -1 to 1, where -1 and 1 indicate perfect negative and positive relationships, respectively.

Implementation Details. By default, the volume size $H \times W \times Z$ is set to 192×192×16, corresponding to the real world range of [-50m, -50m, -3m, 50m, 50m, 12m] around the vehicle. The Triplane Transformer consists of three layers, with the first two incorporating both cross-image and cross-plane deformable attentions, while the last layer featuring only cross-plane deformable attention. The Volume Decoder adopts three linear layers to decode Gaussian parameters for voxel features. For nuScenes dataset [31], our model is trained on two A100 GPUs for 100,000 iterations with the batch size of 4. For RealEstate10K dataset [32], our model is trained on a single A100 GPU for 300,000 iterations with the batch size of 8. The AdamW [52] optimizer is adopted with the learning rate of 1×10^{-4} following cosine learning rate decay strategy. More details can be found in our supplementary material.

4.2. Main Results

Ego-Centric Reconstruction. For evaluation, we make comparisons with state-of-the-art sparse-view reconstruction methods pixelSplat [13] and MVSplat [14] re-implemented following their official code. They both adopt pixel-based Gaussian as the representation. The quantitative results are shown in Table 1. We can see that our method significantly surpasses others in terms of all metrics, especially for PCC that measures the geometric quality. The main reason is that large cross-view overlap is unavailable for ego-centric reconstruction. Other methods cannot predict accurate depths using pixel-level 3D priors (e.g., epipolar lines [13] or cost volumes [14]) that are dependent on cross-view correlation. Besides, the drawbacks of pixel-based Gaussian also pose challenges for reconstruction. Instead, we utilize volume-based Gaussian to lift 2D features to 3D space and predict Gaussians at the 3D level without relying on cross-view overlap. Thanks to our dual-path deformable attentions, we can further mitigate the spatial limitations of pixel-based Gaussian during feature encoding. The qualitative results are shown in Fig.4. We can see that both pixelSplat and MVSplat fail to render plausible depths, causing blurriness or inconsistency with ground truths in their results. In contrast, our method can generate high-quality images and depths even if significant viewpoint changes exist between the input and the novel views.

Method	PSNR↑	SSIM↑	LPIPS↓	PCC↑
Volume-based	22.21	0.640	0.357	0.701
Pixel-based w/o depth init.	22.92	0.692	0.287	0.572
Pixel-based	22.89	0.698	0.290	0.780
Full w/o train decomp.	23.75	0.717	0.258	0.795
Full w/o feat fuse.	23.35	0.708	0.262	0.786
Full	**24.27**	**0.736**	**0.237**	**0.800**

Table 2. Ablations on nuScenes [31]. The 2nd to the 4th rows show results of models with only singular representations (volume or pixel-based Gaussian). The 5th to 7th rows show results of models with full Omni-Gaussian representation. Besides, "depth init.", "train decomp." and "feat fuse." denote components of Depth Initialization for pixel-based Gaussian, Depth-Guided Training Decomposition and Projection-based Feature Fusion, respectively.

Scene-Centric Reconstruction. For evaluation, we compare our approach with more baseline methods on a widely-used scene-centric dataset (RealEstate10K [32]). As evident by the quantitative results in Table 1, our method achieves comparable visual quality (measured by PSNR, SSIM and LPIPS) to state-of-the-art methods, and outperforms all prior works in terms of geometric quality (measured by PCC). We also conduct qualitative comparisons in Fig.5, where we can obtain novel views with better details than those produced by others, especially for cases of large motions. Both of the quantitative and qualitative results show that our method not only exhibits superior performance in ego-centric reconstruction but also possesses competence in scene-centric reconstruction.

Multi-Modal Generation. Our Omni-Scene can not only serve as a standalone reconstruction model, but also be seamlessly integrated with a 2D diffusion model [48] to achieve feed-forward text-to-3D or layout-to-3D scene generation. Specifically, given multi-modal conditions of textual descriptions or 3D layouts (e.g., 3D boxes, BEV map), we utilize [48] to produce six single-frame surrounding images. Then, we can feed the images into our model to generate the corresponding explorable 3D scene with explicit 3D Gaussians. The most relevant work to us is Mag-

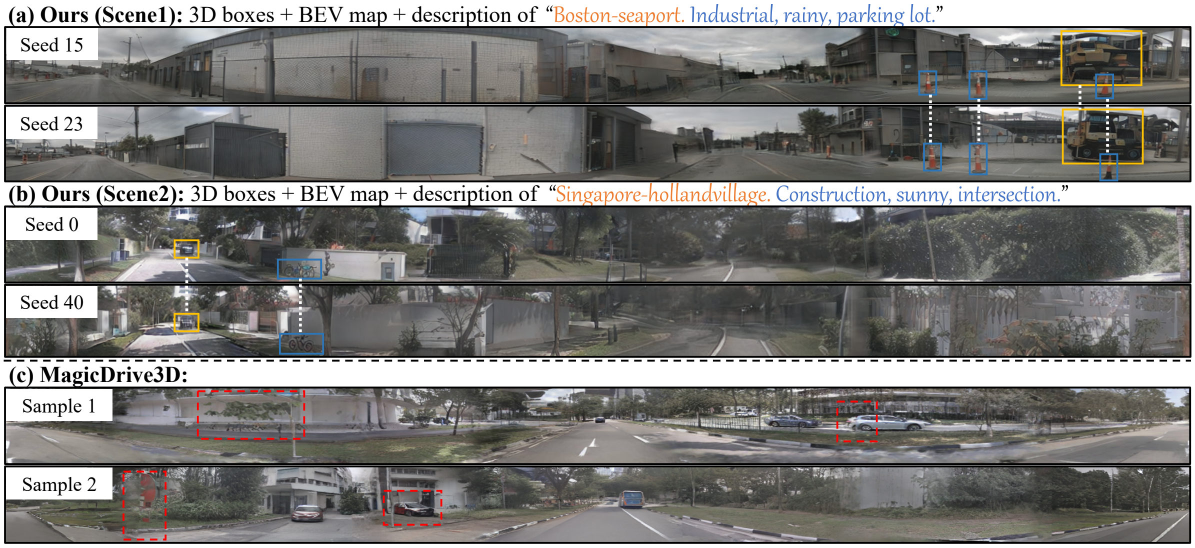
(a) Ours (Scene1): 3D boxes + BEV map + description of "Boston-seaport. Industrial, rainy, parking lot."
Seed 15
Seed 23
(b) Ours (Scene2): 3D boxes + BEV map + description of "Singapore-hollandvillage. Construction, sunny, intersection."
Seed 0
Seed 40
(c) MagicDrive3D:
Sample 1
Sample 2

Figure 6. Multi-modal 3D scene generation. We accept multi-modal conditions (i.e., 3D boxes, BEV map, textual descriptions) as inputs, and generate the corresponding 3D driving scenes in a feed-forward manner. For better visualization, we render 360-degree rotation videos for the generated 3D scenes, and stitch frames into panoramic images as shown in (a) and (b). We can see that the styles of the generated scenes closely match the textual conditions. Besides, when the appearances vary with random seeds, the spatial consistency with conditional 3D boxes (denoted by colored rectangles in (a) and (b)) is well preserved. Compared to per-scene optimization-based method MagicDrive3D [53] that leads to artifacts highlighted by red dashed lines in (c), we achieve higher quality with better visual details. *Please consult our supplementary material for comparisons in video format*, where we can better observe the differences in visual quality.

icDrive3D [53], which first uses video diffusion model to generate multi-view video with approximately 100 images, and then reconstruct the scene based on deformable Gaussians [54]. Such scene-by-scene reconstruction is inefficient and demands high spatial and temporal consistency from the generated videos, often failing and introducing noise or jitter artifacts in the synthetic 3D scenes. As shown in Fig.6(a)-(b), our generated results exhibit high fidelity and good diversity, while also ensure consistency with both the textual and layout conditions. Compared to results from MagicDrive3D [53] as shown in Fig.6(c), we achieve better quality in a much more efficient feed-forward manner. This demonstrates the potential of Omni-Scene for generation, pioneering a new approach to multi-modal generation of 3D driving scenes in a feed-forward manner.

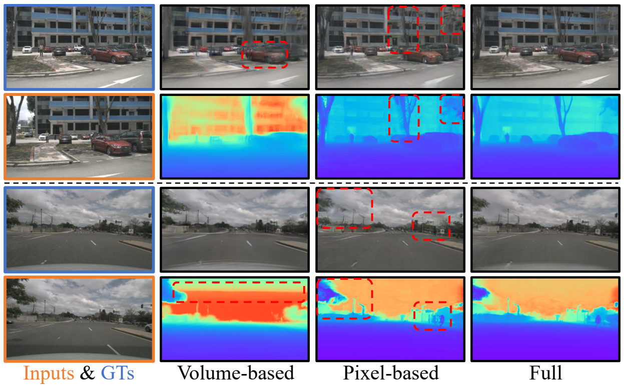

Figure 7. Ablations on Omni-Gaussian representation. Images of input views (Inputs) and ground-truth novel views (GTs) are outlined by orange and blue rectangles, respectively.

Additional Results. *Please refer to our supplementary material* for additional results including **scene-exploring videos**, runtime analysis, more comparisons, etc.

4.3. Ablation Study

Effectiveness of Omni-Gaussian Representation. The core of our method lies in the Omni-Gaussian representation. Therefore, we train two variant models with only volume or pixel-based Gaussian representation for comparisons. By comparing our "Full" method with "Volume-based" and "Pixel-based" in Table 2, we can see that the full method with Omni-Gaussian representation surpasses the two singular representation variants by a large margin. Without volume-based Gaussian to address object occlusions and frustum truncations, we can observe a performance drop of 1.38dB PSNR, 0.038 of SSIM, and 0.02 of PCC. Removing pixel-based Gaussian which refines details and reconstructs distant elements, PSNR, SSIM and PCC will be decreased by 2.06dB, 0.096 and 0.099, respectively. Such deterioration is more evident in Fig.7. With only volume-based Gaussian, we observe clear depth boundaries (i.e., last row, 2nd column of Fig.7) and lack of visual details (i.e., 1st row, 2nd column of Fig.7), corresponding to Case 3 and 4 in Fig.2(a). With only pixel-based Gaussian, we observe noise artifacts in areas occluded or truncated in 2D (3rd column of Fig.7), corresponding to Case 1 and 2 in Fig.2(a). With the collaboration of the two representations, our full method can eliminate these artifacts and achieve best consistency with GTs (4th column of Fig.7). Moreover, we show examples in Fig.2(b)-(e) to further illustrate benefits from such collaboration.

Effectiveness of Volume-Pixel Collaboration. As shown in Table 2, our "Full" method outperforms the variants "Full w/o feat fuse." and "Full w/o train decomp." in terms of all metrics. This indicates that the collaboration between volume and pixel-based representations is important in both the feature encoding stage and the training stage. The Projection-Based Feature Fusion strategy enables our Volume Builder to be aware of which areas are already covered by pixel-based Gaussian, allowing it to better complement the uncovered areas. The Depth-Guided Training Decomposition mechanism allows our Volume Builder to focus on reconstruction within the volume range, while also ensuring the spatial alignment between the volume and the pixel-based Gaussians, which avoids scale ambiguity. Visual results can be found in our supplementary material.

Effectiveness of Depth Initialization. We also train a pixel-based variant model without initializing per-pixel depths using [46] to investigate the impact of depth initialization on performance. As shown in Table 2, although the variant "Pixel-based w/o depth init." can achieve comparable visual quality to the full-version "Pixel-based", it leads to a 0.208 drop of PCC. The main reason for this gap is that the depth initialization can ease the prediction of complex 3D geometries under the ego-centric setting. Visual results can be found in our supplementary material.

5. Conclusion

We have introduced Omni-Scene, a method with Omni-Gaussian representation that can reach the best of both pixel and volume-based Gaussian representations for ego-centric sparse-view scene reconstruction. Employing designs that encourage Volume-Pixel collaboration, we achieve high-fidelity scene reconstruction from only single-frame surrounding observations. Extensive experiments demonstrate our superiority in ego-centric reconstruction compared to previous methods. Furthermore, we integrate a 2D diffusion model into our framework, which enables multi-modal 3D scene generation with versatile applications.

References

[1] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4578–4587, 2021. 1, 3

[2] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2021.

[3] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7824–7833, 2022. 3

[4] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14124–14133, 2021. 3

[5] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18365–18375, 2022.

[6] Haotong Lin, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Efficient neural radiance fields for interactive free-viewpoint video. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022.

[7] Xiaoshuai Zhang, Sai Bi, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. Nerfusion: Fusing radiance fields for large-scale scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5449–5458, 2022. 3

[8] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023. 3, 4

[9] Haofei Xu, Anpei Chen, Yuedong Chen, Christos Sakaridis, Yulun Zhang, Marc Pollefeys, Andreas Geiger, and Fisher Yu. Murf: Multi-baseline radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20041–20050, 2024. 1, 3, 6, 7, 2

[10] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light field neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8269–8279, 2022. 1, 3, 2

[11] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *European Conference on Computer Vision*, pages 156–174. Springer, 2022.

[12] Yilun Du, Cameron Smith, Ayush Tewari, and Vincent Sitzmann. Learning to render novel views from wide-baseline stereo pairs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4970–4980, 2023. 1, 3, 6, 2

[13] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19457–19467, 2024. 1, 2, 3, 5, 6, 7, 4

[14] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. *arXiv preprint arXiv:2403.14627*, 2024. 2, 3, 5, 6, 7, 4

[15] Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, João F Henriques, Christian Rup-

precht, and Andrea Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. *arXiv preprint arXiv:2406.04343*, 2024.

[16] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2025. 2, 3, 5

[17] Stanislaw Szymanowicz, Chrisitian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10208–10217, 2024.

[18] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. *arXiv preprint arXiv:2403.14621*, 2024. 2, 3

[19] Chubin Zhang, Hongliang Song, Yi Wei, Yu Chen, Jiwen Lu, and Yansong Tang. Geolrm: Geometry-aware large reconstruction model for high-quality 3d gaussian generation. *arXiv preprint arXiv:2406.15333*, 2024.

[20] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10324–10335, 2024.

[21] Tianqi Liu, Guangcong Wang, Shoukang Hu, Liao Shen, Xinyi Ye, Yuhang Zang, Zhiguo Cao, Wei Li, and Ziwei Liu. Fast generalizable gaussian splatting reconstruction from multi-view stereo. *arXiv preprint arXiv:2405.12218*, 2024. 2, 3

[22] Hao Li, Yuanyuan Gao, Dingwen Zhang, Chenming Wu, Yalun Dai, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, and Junwei Han. Ggrt: Towards generalizable 3d gaussians without pose priors in real-time. *arXiv preprint arXiv:2403.10147*, 2024. 1, 2, 3

[23] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 3, 2

[24] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1, 3, 5, 2

[25] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision*, pages 1–18. Springer, 2022. 2, 3, 4

[26] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE international conference on robotics and automation (ICRA)*, pages 2774–2781. IEEE, 2023. 3

[27] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *European Conference on Computer Vision*, pages 531–548. Springer, 2022. 2, 3

[28] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21729–21740, 2023. 2, 3

[29] Yunpeng Zhang, Zheng Zhu, and Dalong Du. Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9433–9443, 2023.

[30] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9223–9232, 2023. 2, 3, 4

[31] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 3, 6, 7, 1, 2, 5

[32] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 3, 6, 7

[33] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5855–5864, 2021. 3

[34] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19697–19705, 2023. 3

[35] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 3

[36] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19447–19456, 2024. 3

[37] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 3

[38] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, et al. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023. 3

[39] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21634–21643, 2024. 3

[40] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 194–210. Springer, 2020. 3

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[42] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 4, 1

[43] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, et al. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. *arXiv preprint arXiv:2311.09217*, 2023. 4

[44] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. Crm: Single image to 3d textured mesh with convolutional reconstruction model. *arXiv preprint arXiv:2403.05034*, 2024. 4

[45] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-\sigma: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. *arXiv preprint arXiv:2403.04692*, 2024. 5, 1

[46] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *arXiv preprint arXiv:2404.15506*, 2024. 5, 9

[47] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6, 7

[48] Ruiyuan Gao, Kai Chen, Enze Xie, HONG Lanqing, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. Magicdrive: Street view generation with diverse 3d geometry control. In *The Twelfth International Conference on Learning Representations*, 2023. 6, 7

[49] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 7

[50] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. 7

[51] Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. *Noise reduction in speech processing*, pages 1–4, 2009. 7

[52] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7, 1

[53] Ruiyuan Gao, Kai Chen, Zhihao Li, Lanqing Hong, Zhenguo Li, and Qiang Xu. Magicdrive3d: Controllable 3d generation for any-view rendering in street scenes. *arXiv preprint arXiv:2405.14475*, 2024. 8, 2

[54] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20331–20341, 2024. 8

[55] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1

Omni-Scene: Omni-Gaussian Representation for Ego-Centric Sparse-View Scene Reconstruction

Supplementary Material

In this document, we first provide implementation details including data preprocessing of nuScenes [31] (Sec.6.1), network architecture and hyperparameters (Sec.6.2). We follow with additional experiment discussions including introduction to our supplementary videos (Sec.7.1), quantitative comparisons with more baseline methods (Sec.7.2), runtime analysis (Sec.7.3), more ablations (Sec.7.4), further discussions on generalizability to larger bins (Sec. 7.5) and effectiveness of our Volume-Pixel Collaboration (Sec. 7.6), more qualitative results on scene-centric reconstruction (Sec.7.7). ***We strongly recommend to view the accompanying video ("video.mp4")***, which contains 360-degree exploring videos of both reconstructed and synthetic scenes, as well as comparisons with other methods.

6. Additional Implementation Details

6.1. Data Preprocessing

As described in Sec.4.1 of our main manuscript, we partition each scene of nuScenes dataset [31] into equally spaced bins, with each bin serving as one data sample. For nuScenes dataset, each video is captured in a single scene along with the car trajectory. The length of the trajectory ranges drastically from several meters to hundreds of meters. If we segment the trajectory into bins according to frame indexes, the spatial ranges of bins would exhibit significant variation, which leads to non-IID data distribution for training and evaluation. To circumvent this issue, we segment the bins based on the distance traveled by the car as detailed in Fig.8. Specifically, for videos with a trajectory length exceeding 3.2 meters, we uniformly segment them into N bins, each 3.2 meters in length. For each bin, we use the central frame with 6 surrounding images to derive the observed input views, and the first and last frames with 12 surrounding images as the novel views. For videos with a trajectory length less than 3.2 meters, we directly use the first and last frames of the video as the novel views.

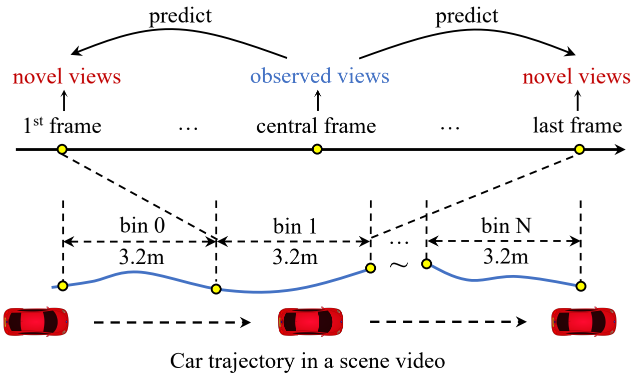

Figure 8. Data preprocessing of nuScenes [31].

6.2. Network Architecture and Hyperparameters

In Table 3(a), the order from top to bottom are the parameters of Triplane Transformer (i.e., number of transformer layers, embedding dimensions, number of 2D and 3D reference points used in our cross-image and cross-plane deformable attentions, and number of attention heads), Voxel Decoder (i.e., number of Gaussians decoded for each voxel, number of linear layers used for decoding Gaussian parameters), Multi-View U-Net (i.e., feature dimensions and patch sizes of patchified cross attentions [45] used in U-Net downsample and upsample blocks), Pixel Decoder (i.e., number of convolution layers used for decoding Gaussian parameters), respectively. In Table 3(b), we specify loss weights for Eq.(4) in our main manuscript, which is followed by parameters used in our training phase.

(a) Network Architecture		
2D Image Encoder	backbone	R50-DINO [42]
	neck	FPN (P2 only) [55]
Triplane Transformer	# layers	3
	# embed dims	128
	# 2D ref points	8, 16, 16
	# 3D ref points	16, 16, 16
	# attn heads	8
Voxel Decoder	# Gaussians per voxel	3
	# linear layers	3
Multi-View U-Net	# downsample feats	128, 256, 512, 512
	# upsample feats	512, 512, 256, 128
	# downsample patches	8, 8, 4, 2
	# upsample patches	2, 4, 8, 8
Pixel Decoder	# conv layers	3
(b) Hyperparameters		
Loss Weights	# $\lambda_1, \lambda_2, \lambda_{V_1}, \lambda_{V_2}$	0.05, 1.0, 0.05, 0.01
Training Details	learning rate scheduler	Cosine
	# iterations	100,000
	# learning rate	1e-4
	optimizer	Adam [52]
	# beta1, beta2	0.9, 0.999
	# weight decay	0.01
	# warm-up	1000
	# gradient clip	1.0

Table 3. Details of network architecture and hyperparameters. In the table, "#" denotes numerical parameters. We present parameters that specify our network architecture, and parameters used in our loss functions and training phase, in (a) and (b), respectively.

Method	Time(s)	Param(M)	PSNR↑	SSIM↑	LPIPS↓	PCC↑
AttnRend [12]	9.98	125.1	20.96	0.533	0.467	N/A
MuRF [9]	0.672	**5.3**	20.34	0.504	0.433	-0.332
pixelSplat [13]	0.508	125.4	21.51	0.616	0.372	0.001
MVSplat [14]	0.174	12.0	21.61	0.658	0.295	0.181
Ours	**0.088**	81.7	**24.27**	**0.736**	**0.237**	**0.800**

Table 4. Additional quantitative results on ego-centric reconstruction task performed on nuScenes [31]. We **bold 1st-place** results and underline 2nd-place results. PCC is not available (N/A) for AttnRend which has no interpretable 3D structure for depth rendering.

7. Additional Experiments

7.1. Video Results

To better demonstrate the quality of 3D reconstruction, we provide **exploring video demos in "video.mp4"** along with our supplementary material. Specifically, given six surrounding images of a scene, we conduct inference and obtain 3D Gaussians for reconstructing the scene. Then, we utilize these Gaussians to render a 360-degree rotation video at 30fps with the camera FOV set to 70 degree following [31]. In the video, each frame that falls between the input viewpoints can be considered as a novel view unseen in the inputs. To further demonstrate the model's performance in the presence of object occlusions and frustum truncations, we move the camera forward and backward by 3 meters in the front and rear view perspectives, respectively, ensuring that there are contents invisible from the input views. It's also noted that the camera's movement range has reached 6 meters, exceeding the 3.2-meter range of bin samples seen during training, thereby showcasing the model's capability to reconstruct scenes at greater distances.

Comparisons with other methods. We first present comparisons with state-of-the-art methods pixelSplat [13] and MVSplat [14] from *00:00 to 01:40 in "video.mp4"*. Our approach significantly outperforms other methods in both visual and geometric quality. Notably, due to the minimal cross-view overlap among input views, pixelSplat and MVSplat fail to predict accurate depths based on pixel-level 3D priors (e.g., epipolar lines, cost volumes), which results in artifacts in the rendered videos especially when the camera is substantially moved forward and backward.

Exploring videos of reconstructed scenes. Then, we present more examples to illustrate our functionality on scene reconstruction. Examples with normal conditions are shown from *01:41 to 02:49 in "video.mp4"*. Examples with extreme conditions (e.g., low-light, bad weather) are shown from *02:50 to 03:26 in "video.mp4"*. We can see that our method achieves high-quality reconstruction and maintains robustness in both normal and hard cases.

Exploring videos of generated scenes. We also present examples to illustrate our functionality on scene generation from *03:27 to 05:07 in "video.mp4"*. The left side of the video shows the our generated results given different random seeds. The right side of the video shows examples of MagicDrive3D [53], which are directly adopted from their official website[1]. We can see that our method achieves better visual details than per-scene optimization-based MagicDrive3D in a much more efficient feed-forward manner.

7.2. Comparisons with More Baselines

We also make comparisons with more baseline methods (i.e., MuRF [9] and AttnRend [12]) for ego-centric sparse-view reconstruction task. Specifically, MuRF and AttnRend are feed-forward reconstruction methods based on NeRF [23] and light field [10], respectively. They are both leading and representative methods within their respective lines of works, which constitute the mainstream feed-forward methods together with 3DGS-based approaches such as pixelSplat [13] and MVSplat [14]. As shown in Table 4, our method surpasses MuRF and AttnRend significantly in terms of all metrics. We can also observe that methods with explicit Gaussians as representations (i.e., ours, pixelSplat, MVSplat) outperform methods with implicit NeRF or light field as representations (i.e., AttnRend, MuRF), showing the effectiveness of explicit 3D representation.

7.3. Runtime Analysis

As shown in Table 4, we conduct runtime analysis on the ego-centric reconstruction task to demonstrate the efficiency of our method. It's noted that the inference speed is reported based on the time cost of six-view reconstruction averaged by 2,048 times. From the table, we can see that our method achieves the shortest inference time (i.e., "Time" in Table 4), which is nearly 2× faster than that of the 2nd place method MVSplat [14]. We attribute this advantage to our triplane-based volume feature encoding in Triplane Transformer, and efficient patchified cross-attention module in Multi-View U-Net. Besides, our method is also lightweight with model size (i.e., "Param" in Table 4) comparable to other methods. Furthermore, we observe that, thanks to the efficient rendering of 3DGS [24], 3DGS-based methods (i.e., our method, pixelSplat [13], MVSplat [14]) show significant superiority in speed compared to methods based on implicit representations (i.e., MuRF [9], AttnRend [12]).

7.4. Additional Ablations

We present more ablation results to demonstrate the effectiveness of our components.

Qualitative Ablations on Volume-Pixel Collaboration. In

[1] https://gaoruiyuan.com/magicdrive3d/

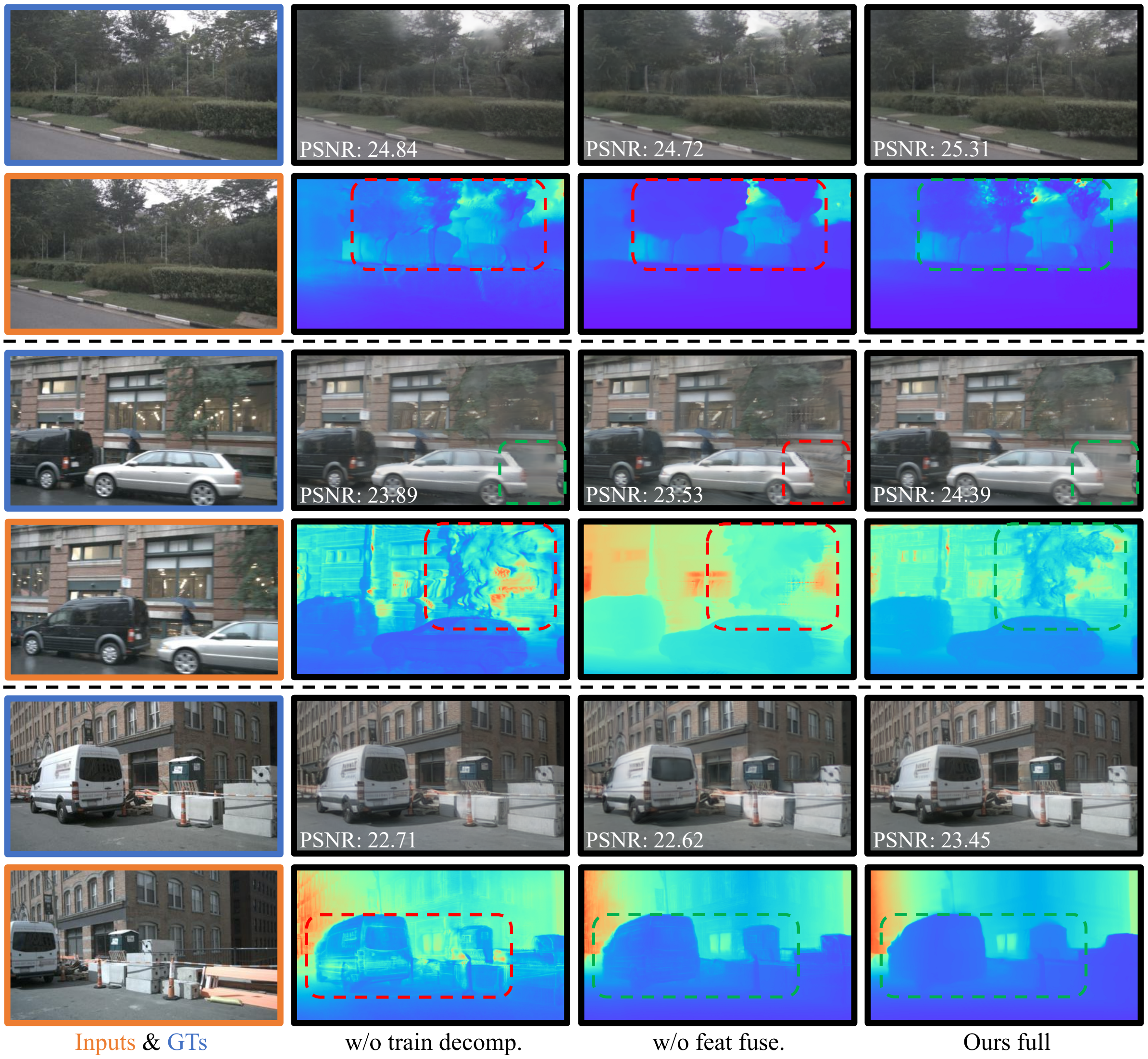

Figure 9. Qualitative ablations on Volume-Pixel Collaboration. Images of input views (Inputs) and ground-truth novel views (GTs) are outlined by orange and blue rectangles, respectively. The remaining are novel-view images and depths generated by our variant models and full model. From left to right, the order is the variant without Depth-Guided Training Decomposition, the variant without Projection-Based Feature Fusion, and our full method. The *red dashed lines* highlight undesirable artifacts (e.g., noise, over-smooth), while the *green ones* denote plausibly-rendered areas (e.g., better and sharper details). We also show PSNR values of the generated images for better comparison.

our main manuscript, we have quantitatively compared our full method with the two variants without the Volume-Pixel Collaboration designs (i.e., Projection-based Feature Fusion and Depth-Guided Training Decomposition). Here, we show additional qualitative results in Figure 9 for visual comparisons. It can be observed that our full method can generate images with higher quality and depths with sharper details, which demonstrate that our collaboration designs can effectively encourage the complementarity between pixel-based and volume-based Gaussian representations, and further improve the performance.

Qualitative Ablations on Depth Initialization. In our main manuscript, we have quantitatively demonstrate the effectiveness of depth initialization for our pixel-based Gaussian representation. Here, we show additional qualitative results in Figure 10 for visual comparisons. From the figure, we can see that, although the depth initialization has no significant impact on visual quality, it is beneficial for improving geometric quality. The main reason is that the depth initialization can ease the learning of complex scene geometries for our Pixel Decorator that built upon pixel-based representation. Besides, with the collaboration of volume-based representation, our full method significantly surpasses the two variants with only pixel-based representations both visually and geometrically, further demonstrating the advantage of our Omni-Gaussian representation.

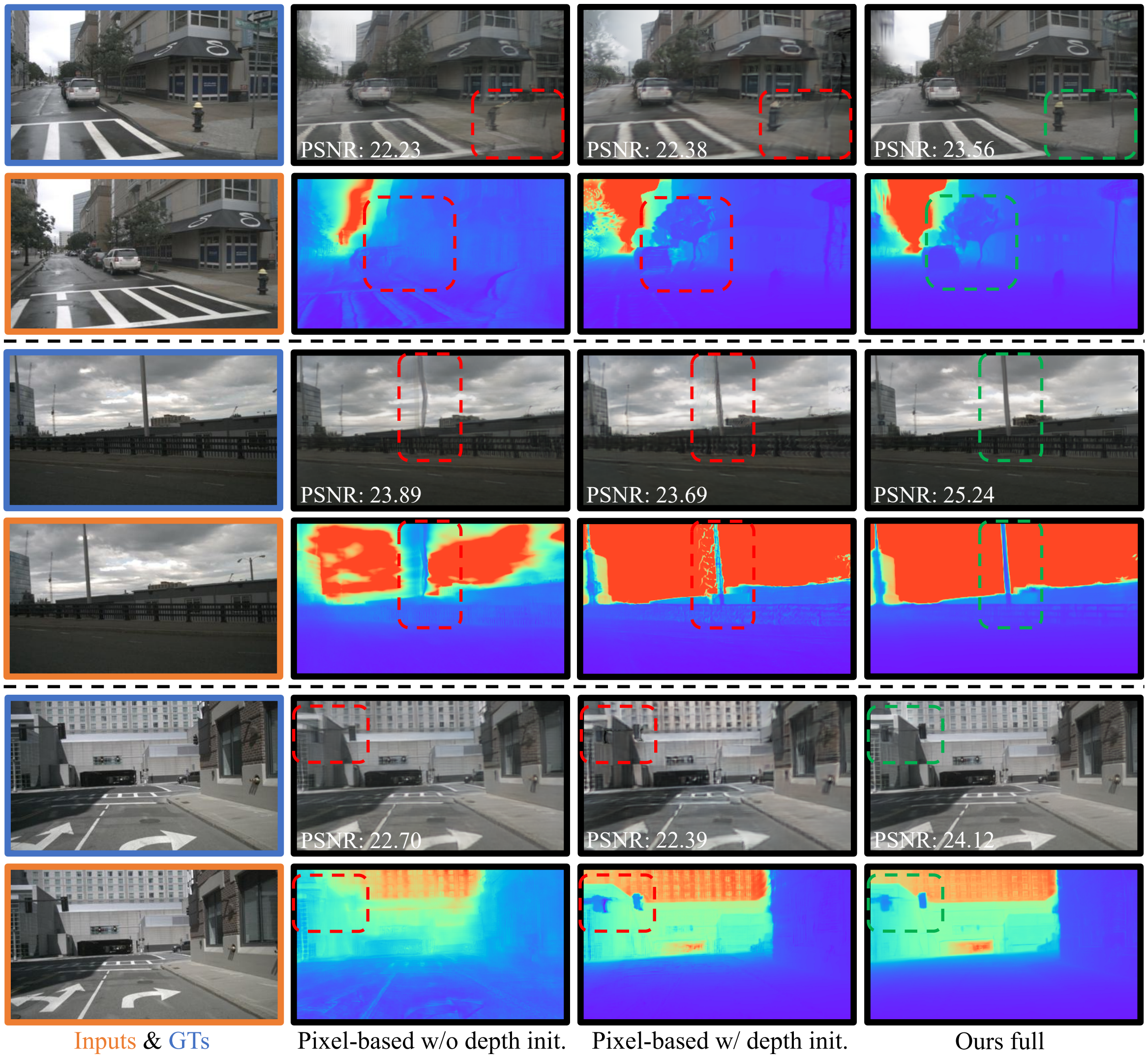

Figure 10. Qualitative ablations on Depth Initialization. The 1st column present images of input views (Inputs) and ground-truth novel views (GTs). The 2nd and the 3rd columns are results generated by two variant models with only pixel-based representation (i.e., Pixel Decorator), one with the depth initialization and one without. The last column denote results generated by our full method. The *red dashed lines* highlight undesirable artifacts, while the *green ones* denote plausibly-rendered areas. PSNR values are shown for better comparison.

Ablations on Deformable Attentions. As described in Sec.3.1 of our main manuscript, we employ cross-image and cross-plane deformable attentions in our Volume Builder to enhance volumetric feature encoding. Given camera parameters (i.e., intrinsics and extrinsics) that enable 3D-to-2D projection, our cross-image deformable attention module can lift 2D features to the 3D volume space, which enables the prediction of 3D Gaussians directly at the 3D level. This differs from previous methods [13, 14] that require cross-view overlap to estimate per-pixel depths and predict 3D Gaussians at the 2D level. To further address the issue that some elements in 3D might be occluded or truncated for any of the 2D input views, we utilize our cross-plane deformable attention to enhance each triplane query with cross-plane context, which means information absent in one plane can be complemented by those from other planes at the 3D level. To validate the effectiveness of such dual-path design, we train three Volume Builder models, where one contains both of the cross-image and cross-plane attentions, while the other two contain only one of the attentions. As demonstrated in Table 5 and Fig. 11, the model with both attentions significantly outperforms the other two variants, showing the importance of such dual-path feature encoding to our Volume Builder. We further compare these volume-only variants with our full method

cross-image	cross-plane	PSNR↑	SSIM↑	LPIPS↓	PCC↑
✗	✓	14.29	0.428	0.578	0.539
✓	✗	21.29	0.595	0.412	0.686
✓	✓	**22.21**	**0.640**	**0.357**	**0.701**

Table 5. Ablations on cross-image & plane deformable attentions.

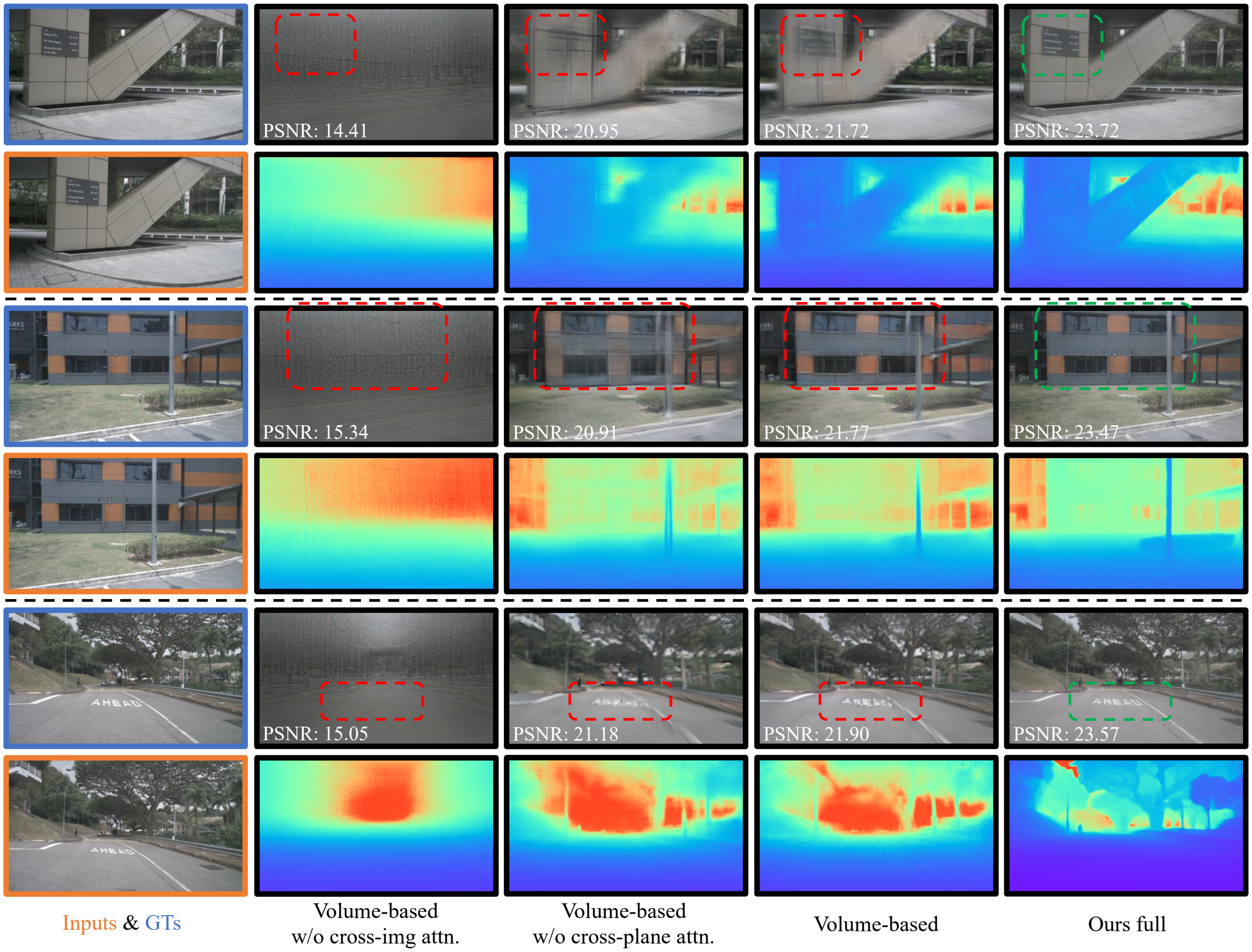

Figure 11. Qualitative ablations on deformable attentions. The 1st column present images of input views (Inputs) and ground-truth novel views (GTs). The 2nd to 4th columns are results generated by three variant models with only volume-based representation (i.e., Volume Builder), one without the cross-image deformable attention ("cross-img attn."), one without the cross-plane deformable attention ("cross-plane attn."), and one with both of the attentions. The last column denote results generated by our full method. The *red dashed lines* highlight undesirable artifacts, while the *green ones* denote plausibly-rendered areas. PSNR values are shown for better comparison.

in Fig. 11. It's observed that results generated by our full method are with better details, showing the effectiveness of our Omni-Gaussian representation.

7.5. Generalizability to Different Bin Sizes

Unless otherwise specified, our experiments are conducted with a bin size of 3.2m as stated in Sec.6.1. To validate whether our method can be generalized to synthesize novel views at farther or closer distances, we preprocess nuScenes [31] into three variants with different bin sizes (i.e., 1.6m, 6.4m, 12.8m) from our original dataset. Here we note that, the larger the bin size, the farther distance between the novel and the input views, which is more challenging for novel view synthesis. Practically, for each dataset variant, we employ two approaches to test our model: (1) The model trained under a bin size of 3.2m is directly used for evaluation without additional fine-tuning. (2) The model is further fine-tuned with the new bin size for 50,000 steps before evaluation. As can be seen from the 3rd, 5th and 7th rows of Table 6, despite the lack of supervision, our method exhibits minor degradation in performance for novel view synthesis at farther distances. For instance, we observe only 1.38 dB drop of PSNR, and 0.009 drop of PCC for "bin size = 6.4m", which denotes distances 2× farther than those seen during training. As can be seen from the 4th, 6th, and 8th rows of Table 6, by fine-tuning the model on data renewed with different bin sizes, we can further boost the performance and bring novel view synthesis at farther distances (i.e., "bin size = 6.4m, 12.8m") very close to the results obtained under the original setting of "bin size = 3.2m".

bin size	fine-tuning	PSNR↑	SSIM↑	LPIPS↓	PCC↑
3.2m	–	24.27	0.736	0.237	0.800
1.6m	✗	$25.12^{+0.85}$	$0.771^{+0.035}$	$0.208^{-0.030}$	$0.804^{+0.004}$
	✓	$25.37^{+1.10}$	$0.783^{+0.047}$	$0.201^{-0.037}$	$0.806^{+0.006}$
6.4m	✗	$22.89^{-1.38}$	$0.682^{-0.054}$	$0.287^{+0.050}$	$0.791^{-0.009}$
	✓	$24.15^{-0.12}$	$0.729^{-0.007}$	$0.239^{+0.002}$	$0.797^{-0.003}$
12.8m	✗	$21.57^{-2.70}$	$0.640^{-0.096}$	$0.346^{+0.109}$	$0.771^{-0.029}$
	✓	$23.55^{-0.72}$	$0.711^{-0.025}$	$0.265^{+0.028}$	$0.792^{-0.008}$

Table 6. Results of our method when generalized to different bin sizes with or without additional fine-tuning.

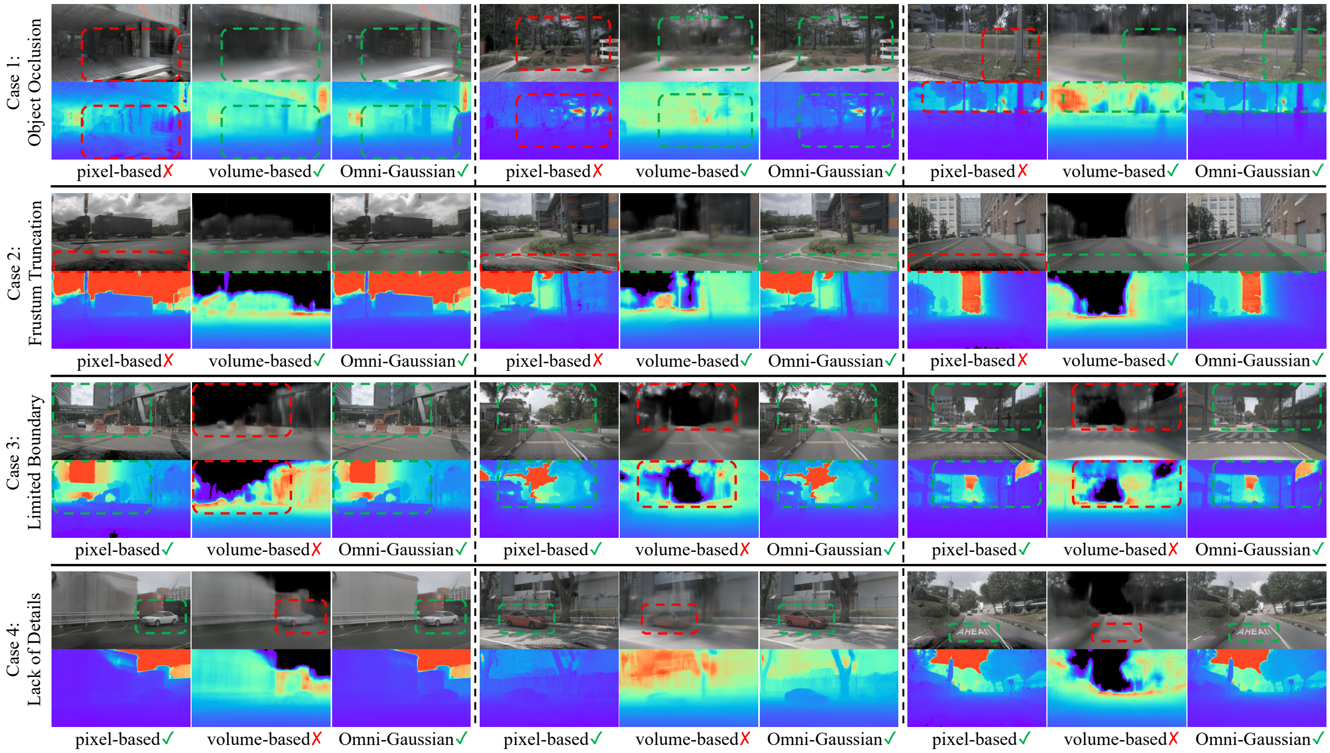

Figure 12. Additional examples of Volume-Pixel Collaboration. The *red dashed lines* highlight artifacts caused by weaknesses of singular representations, while the *green ones* outline how the artifacts are eliminated through Volume-Pixel Collaboration.

7.6. Discussion on Volume-Pixel Collaboration

In Fig.2 of our main manuscript, we have showcased pros and cons of the pixel-based and the volume-based Gaussian representations, and have provided the corresponding examples to illustrate *how the two representations complement for each other in our unified model with the proposed Omni-Gaussian representation*. Here, we present more examples in Fig.12 to demonstrate the effectiveness of their collaboration case by case:

- In "Case 1" of Fig.12, when objects in the novel view are occluded in the input views, pixel-based representation focuses on the non-occluded areas, with the occluded parts supplemented by volume-based representation.
- In "Case 2" of Fig.12, when objects in the the novel view fall beyond the frustum range for any of the input views, pixel-based representation focuses on the non-truncated areas, with the truncated parts supplemented by volume-based representation.
- In "Case 3" of Fig.12, for distant elements out of the volume range, volume-based representation concentrates on reconstruction within the volume, leaving the reconstruction of distant elements to pixel-based representation.
- In "Case 4" of Fig.12, for objects with fine-grained details (e.g., cars, lane markings), volume-based representation aims to predict their coarse 3D structures, leaving the surface details to pixel-based representation.

7.7. Additional Comparisons on RealEstate10K

As shown in Fig.13, we present more qualitative comparisons with state-of-the-art methods pixelSplat [13] and MVSplat [14] on RealEstate10K [32], a large-scale dataset for scene-centric reconstruction task. We can see that our method can render novel view images and depths with comparable and even superior quality to other methods.

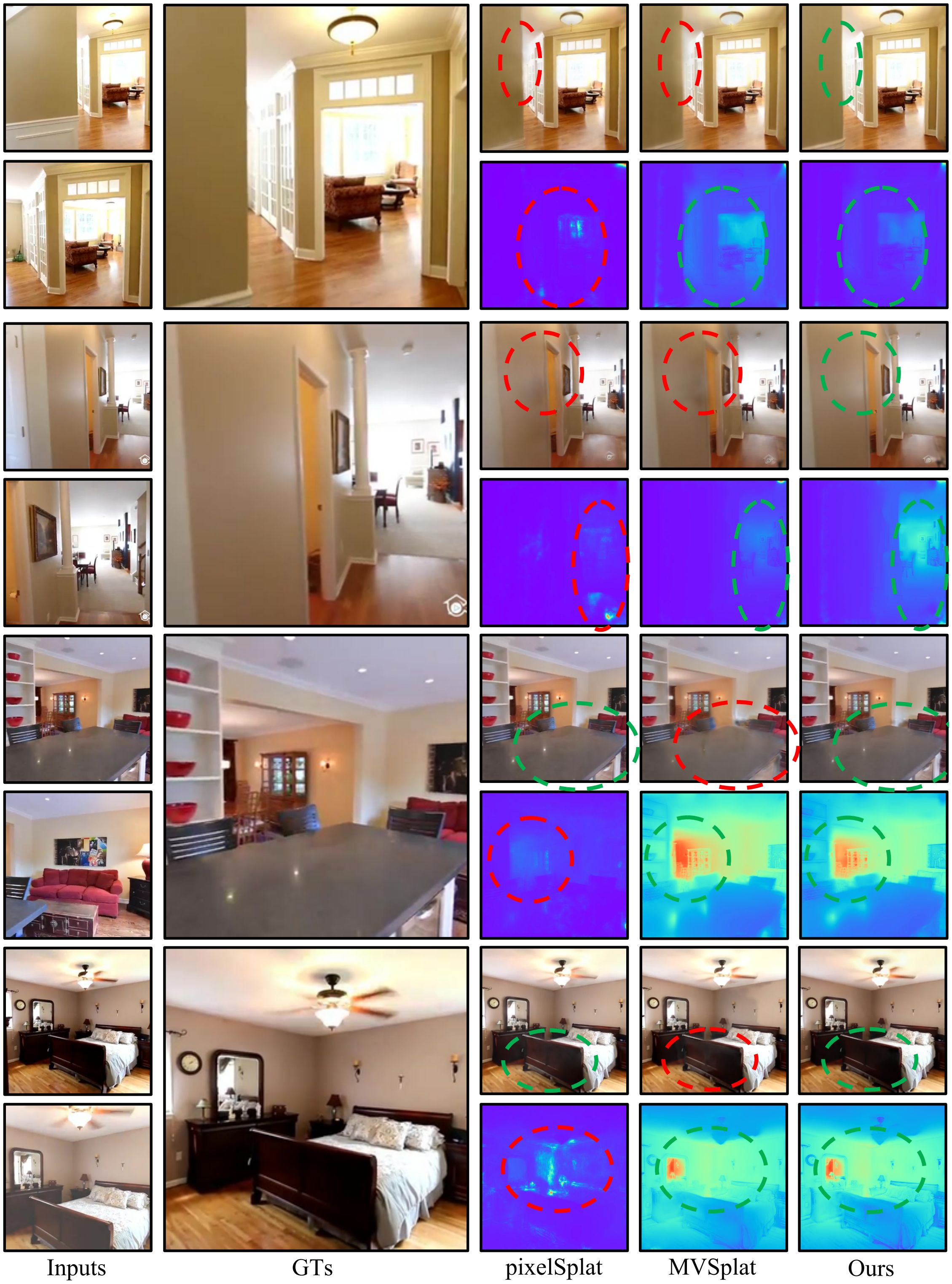

Figure 13. Additional qualitative results on scene-centric reconstruction performed on RealEstate10K [32]. The first two columns are images of input views and ground-truth novel views. The remaining three columns are results generated by pixelSplat [13], MVSplat [14] and our method, respectively. The *red dashed lines* highlight undesirable artifacts, while the *green ones* denote plausibly-rendered areas.