VolSplat: Rethinking Feed-Forward 3D Gaussian Splatting with Voxel-Aligned Prediction

Weijie Wang[1,2*] Yeqing Chen[1*] Zeyu Zhang[2] Hengyu Liu[2,3]
Haoxiao Wang[1] Zhiyuan Feng[4] Wenkang Qin[2] Feng Chen[5]
Jia-Wang Bian[6] Zheng Zhu[2†] Donny Y. Chen[7] Bohan Zhuang[1†]

Zhejiang University[1] GigaAI[2] The Chinese University of Hong Kong[3]
Tsinghua University[4] Adelaide University[5] Nanyang Technological University[6]
Monash University[7]
* Equal contribution † Corresponding authors

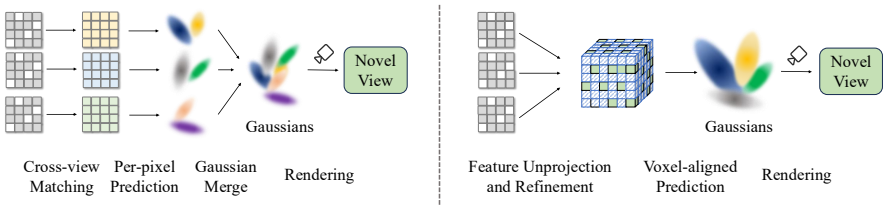
Project Page: lhmd.top/volsplat

Abstract. Feed-forward 3D Gaussian Splatting (3DGS) has emerged as a highly effective solution for novel view synthesis. Existing methods predominantly rely on a *pixel-aligned* Gaussian prediction paradigm, where each 2D pixel is mapped to a 3D Gaussian. We rethink this widely adopted formulation and identify several inherent limitations: it renders the reconstructed 3D models heavily dependent on the number of input views, leads to view-biased density distributions, and introduces alignment errors, particularly when source views contain occlusions or low texture. To address these challenges, we introduce VolSplat, a new multi-view feed-forward paradigm that replaces pixel alignment with voxel-aligned Gaussians. By directly predicting Gaussians from a predicted 3D voxel grid, it overcomes pixel alignment's reliance on error-prone 2D feature matching, ensuring robust multi-view consistency. Furthermore, it enables adaptive control over density based on 3D scene complexity, yielding more faithful Gaussians, improved geometric consistency, and enhanced novel-view rendering quality. Experiments on widely used benchmarks demonstrate that VolSplat achieves state-of-the-art performance, while producing more plausible and view-consistent results.

Keywords: 3D Gaussians · Feed-Forward Reconstruction · View Synthesis

1 Introduction

3D reconstruction is a cornerstone of modern robotics, empowering autonomous systems with the critical ability to perceive, map, and comprehend their physical environment [41], which is fundamental for advanced navigation, object manipulation, and world models. Traditional optimization based approaches, including Neural Radiance Fields (NeRF) [33] and 3D Gaussian Splatting (3DGS) [21], obtain high fidelity results by iteratively enforcing photometric or geometric consistency. These methods achieve excellent accuracy but are computationally intensive and slow to run at inference time. By contrast, feed-forward approaches [3–6, 18, 42, 45, 51, 54, 58, 61] trade per instance optimization for fast



(a) Pixel-aligned Feed-forward 3DGS (b) Voxel-aligned Feed-forward 3DGS (Ours)

Fig. 1: Comparison between the pixel-aligned feed-forward method and our approach. Pixel-aligned feed-forward 3DGS methods suffer from two primary limitations: 1) 2D feature matching struggles to effectively resolve the multi-view alignment problem, and 2) the Gaussian density is constrained and cannot be adaptively controlled according to scene complexity. We propose VolSplat, a framework that directly regresses Gaussians from 3D features based on a voxel-aligned prediction strategy. This approach achieves adaptive control over scene complexity and resolves the multi-view alignment challenge.

learned inference. A single forward pass predicts scene geometry or a 3D representation directly from input images. This speed and simplicity make feed-forward systems attractive for real time applications, large scale datasets, and downstream tasks that require many reconstructions.

Prior feed-forward 3DGS methods [3, 5, 18, 44, 49, 51, 54] commonly rely on pixel alignment as their fundamental mechanism for associating image features with pixel aligned Gaussians. In this design, per-pixel features from precomputed image feature maps are unprojected to define the corresponding Gaussians. The prevailing consensus has been to perform fusion directly within the 2D feature representation. However, pixel aligned designs inherit two primary limitations. 1) Sampling at discrete pixel locations is sensitive to camera calibration and discretization error, produces inconsistent sampling patterns across views. 2) The rigid pixel-to-Gaussian association enforces a uniform density distribution that ignores scene complexity, leading to redundant primitives in simple regions while failing to capture fine-grained 3D structures.

In this work, we shift the alignment paradigm from pixels to voxels, as illustrated in Fig. 1. Instead of sampling features at projected pixel coordinates, we align and aggregate image features directly into a 3D voxel grid. Multi-view image features are aggregated directly into this 3D voxel space, effectively decoupling feature fusion from the camera view frustums. Within this unified voxel space, we employ a 3D U-Net [7] to reason about scene geometry and appearance volumetrically. Finally, rather than predicting Gaussians per pixel, we predict primitives directly from the refined voxel features, allowing the distribution of 3D Gaussians to be determined by the volumetric structure itself.

There are practical and conceptual advantages to voxel alignment. Specifically, volumetric aggregation reduces floaters and view dependent inconsistency because information from multiple views is fused into a shared 3D container before Gaussian prediction. Simultaneously, operating in a 3D grid enables the use of well studied 3D decoder and regularization strategies, which naturally en-

code locality and geometrical context. Instead of the integration of auxiliary 3D signals such as depth maps [54] and point clouds [39], our approach naturally resolves spatial ambiguities within the unified voxel space, thereby eliminating the need for such ad hoc priors or auxiliary supervision signals. Furthermore, voxel representations are amenable to modern acceleration strategies such as sparse data structures, making the approach practical at the resolutions required for high quality reconstruction.

In this paper we present a feed-forward three dimensional reconstruction framework built around voxel alignment. As shown in Fig. 2, we first construct 3D feature grids using the extracted 2D image features, then refine the 3D features and use them to predict voxel-aligned Gaussians. We analyze the alignment errors that arise in pixel aligned pipelines and show how voxel alignment reduces these errors both conceptually and empirically. Through systematic experiments on synthetic and real world benchmarks, we demonstrate that voxel aligned feed-forward models achieve more accurate and robust reconstructions than comparable pixel aligned baselines on large-scale benchmarks such as RealEstate10K [65], ScanNet [8] and ACID [25]. Our contributions are as follows:

- We introduce voxel alignment as a principled alternative to pixel alignment for feed-forward 3DGS and present a practical end-to-end framework.
- We provide an analysis of alignment induced errors in pixel aligned systems and show how volumetric aggregation mitigates these failure modes.
- Experimental results demonstrate that VolSplat achieves state-of-the-art (SOTA) performance on several large-scale benchmarks.

2 Related Work

Novel view synthesis. Traditional approaches to Novel View Synthesis (NVS) primarily rely on geometry-based rendering methods that reconstruct explicit 3D scene geometry from images [9], image-based rendering techniques that interpolate between captured views without full 3D reconstruction [16], and light field rendering that samples and reprojects densely captured rays in space [24]. These methods required either accurate geometric proxies, densely sampled viewpoints, or both to produce convincing visual results, limiting their applicability in real-world scenarios. The emergence of NeRF [33] marked a paradigm shift, significantly improving both rendering quality and robustness over prior methods, which learns a continuous, implicit scene representation by utilizing a MLP to map position and viewing direction to a corresponding color and volume density. While NeRF-based methods [1, 2] require a long training time due to the per-ray rendering. 3DGS [21] and its variants [10, 26, 67] have been introduced to represent the 3D scene using a set of anisotropic 3D Gaussians.

3D voxelization. Voxelization, which discretizes 3D space into regular voxel grids, has been a foundational representation in 3D reconstruction and modeling [31]. Prior methods used dense grids for their simplicity, but suffered from

high memory costs and poor scalability [20]. To address this, sparse structures like octrees were introduced for more efficient storage and computation [22]. In modern applications, voxels are widely used as input to 3D Convolutional Neural Network (CNN) for tasks such as object detection [66] and semantic segmentation [38, 40]. More recently, voxels are often used as sparse scaffolding rather than as the final representation, supporting more advanced rendering techniques. Representative methods include Plenoxels [12] and K-Planes [11], which optimize voxel-based radiance fields for fast, high-quality rendering, as well as structured strategies such as Scaffold-GS [30] and Octree-GS [37], which leverage voxel grids to organize and accelerate 3DGS.

Feed-forward 3D Gaussian Splatting. Recent developments in feed-forward 3DGS [3, 5, 15, 17, 34, 47, 51, 54, 63] offer a compelling alternative that directly predicts 3D Gaussians from input images in a single forward pass: pixelSplat [3] proposes a two-view feed-forward pipeline that combines epipolar transformers and depth prediction to generate Gaussians. MVSplat [5] introduces a cost-volume-based fusion strategy to enhance multi-view consistency. DepthSplat [54] leverages monocular depth features to improve fine 3D structure reconstruction from sparse views. Follow-up work extends feed-forward 3DGS to more complex scenarios, including pose-free inputs [19, 57], online stream inputs [51], and more dense inputs [45], which can provide priors for the world models [46, 48, 59]. ECCV While these works adopt a pixel-aligned strategy to predict Gaussian primitives, the pixel-wise formulation struggles to handle multiple input views due to redundancy and inconsistency across pixels. Existing methods attempt to improve the per-pixel strategy by pruning the number of Gaussians [63], token merging [45, 68] and voxel-based fusion [17, 27, 50]. However, these approaches do not fundamentally address the limitations inherent in per-pixel processing. EVolSplat [32] has explored voxel features in autonomous driving scenarios, but it has not been generalized to general scenarios and requires explicit 3D point clouds as intermediate representations. In contrast, our method introduces a voxel-aligned method, which eliminate the need for per-query 2D prediction patterns. This alignment enables more stable multi-view fusion, cleaner occlusion handling, and more coherent joint inference of geometry and appearance.

3 Method

3.1 Preliminary and Observation

Feed-forward 3D reconstruction aims to learn a mapping from N input images $\mathcal{I} = \{\mathbf{I}_i\}_{i=1}^N$ where $\mathbf{I}_i \in \mathbb{R}^{H\times W\times 3}$ and their corresponding camera poses $\mathcal{P} = \{\mathbf{P}_i\}_{i=1}^N$, to a 3D scene representation. In the context of pixel-aligned 3DGS, features are extracted from images and refined by cross-view interaction:

$$\mathcal{F} = \{\mathbf{F}_i\}_{i=1}^N = h(\Phi_{\text{image}}(\mathcal{I}, \mathcal{P})), \quad \mathbf{F}_i \in \mathbb{R}^{\frac{H}{p}\times\frac{W}{p}\times C} \tag{1}$$

where Φ_{image} is a pretrained image encoder. The function h is responsible for processing these features from different viewpoints, with its core purpose being

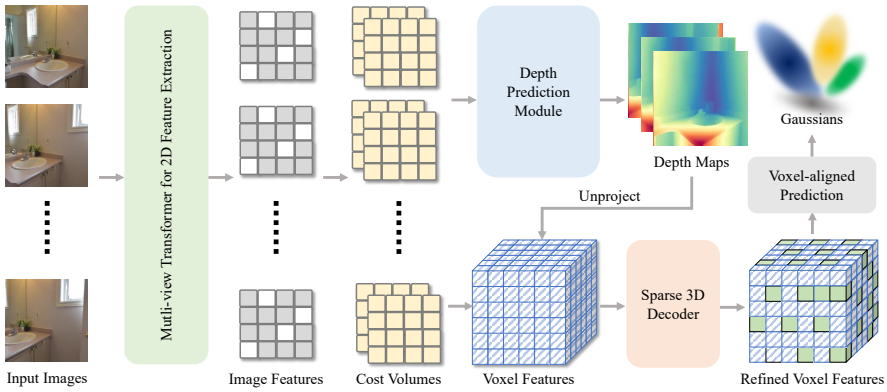


Fig. 2: Overview of VolSplat. Given multi-view images as input, we first extract 2D features for each image using a Transformer-based network and construct per-view cost volumes with plane sweeping. Depth Prediction Module then estimates a depth map for each view, which is used to unproject the 2D features into 3D space to form a voxel feature grid. Subsequently, we employ a sparse 3D decoder (details in Sec. 3.3) to refine these features in 3D space and predict the parameters of a 3D Gaussian for each occupied voxel. Finally, novel views are rendered from the predicted 3D Gaussians.

to perform cross-view feature matching and fusion. For pixel-aligned Gaussian prediction, the features must be upsampled to the same resolution as the input image:

$$\mathcal{F}_{\text{full}} = U(\mathcal{F}) \in \mathbb{R}^{N\times H\times W\times C}, \tag{2}$$

where $\mathcal{F}_{\text{full}}$ denotes the full-resolution feature maps, U is a feature upsampler such as CNN-based network (in MVSplat [5]) and deconvolution-based network [35] (in DepthSplat [54]). Per-pixel Gaussian predictions are then performed using the upsampled features:

$$\mathcal{G} = \{(\mu_i, \mathbf{\Sigma}_i, \alpha_i, \mathbf{c}_i)\}_{i=1}^{H\times W\times N} = \Psi_{\text{pred}}(\mathcal{F}_{\text{full}}, \mathcal{P}), \tag{3}$$

where the position of the Gaussians are determined by the predicted depth and pixel location.

While straightforward, this pixel-aligned formulation introduces two critical limitations. First, the geometric accuracy of the reconstruction is critically dependent on the quality of the predicted depth map. After depth unprojecting features into 3D space, the lack of interaction with neighboring points within the 3D space significantly contributes to the generation of floaters. Second, the structure of the 3D representation is rigidly tied to the 2D image grid. The total number of Gaussians is fixed at $|S| = H \times W \times N$, which is often suboptimal and cause an over-densification of Gaussians on simple, texture-less surfaces and an insufficient number for representing complex geometry not captured at the pixel level. These observations reveal a fundamental bottleneck and motivate our

proposed voxel-aligned framework, designed to decouple the 3D representation from the 2D pixel grid.

3.2 3D Feature Construction

Feature extraction and matching. For N input images, we first apply a weight-sharing ResNet [14] backbone to each RGB image to obtain $p\times$ downsampled feature maps. These features are then refined with cross-view attention that exchanges information with the two nearest neighboring views. For efficiency, this cross-attention is implemented with the local window attention [28]. After this stage we obtain cross-view–aware Transformer features $\{F_i\}_{i=1}^{N}$ ($F^i \in \mathbb{R}^{\frac{H}{p}\times\frac{W}{p}\times C}$) , where C denotes the feature dimension.

Next, we build per-view cost volumes $\{C_i\}_{i=1}^{N}$ using a plane-sweep strategy [55]. For each view i, we sample D candidate depths $\{d_m\}_{m=1}^{D}$, warp the feature from neighboring views to the reference view at each hypothesized depth, and compute pairwise feature similarities [5].These similarities are aggregated by dot-product matching and stacked along the depth axis to form $\{C_i\}_{i=1}^{N}$, where $C_i \in \mathbb{R}^{\frac{H}{p}\times\frac{W}{p}\times D}$.

To produce robust, multi-view consistent depth estimates, a depth module fuses the monocular features $\{F^i_{mono}\}_{i=1}^{N}$ ($F^i_{mono} \in \mathbb{R}^{\frac{H}{p}\times\frac{W}{p}\times C}$) with the cost volume C^i and regresses a dense per-pixel depth map $D_i \in \mathbb{R}^{H\times W}$, which serves as a geometric prior for lifting image features into 3D space. These per-view features F^i and depths D_i are used in the next stage to construct 3D point clouds and voxel-based features for volumetric reasoning.

Lifting to 3D feature. Given the predicted depth maps D_i and camera parameters, we conveniently aggregate different depth map views by transforming the point clouds into a global coordinate system. First each pixel (u, v) in image space is unprojected to a 3D point in the camera coordinate frame using the camera intrinsics. Then the 3D point is transformed into the world coordinate system via the corresponding extrinsic parameters, including the rotation matrix R_i and translation T_i vector.

$$P_{\text{world}} = R_i\, P_{\text{cam}} + T_i = R_i \left(D_i(u,v)\, K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \right) + T_i. \tag{4}$$

By repeating this process across all views, we obtain a dense $|S| = H \times W \times N$ point cloud in world space, where each 3D point is associated with its corresponding image feature.

To convert the unstructured dense point cloud P into a structured volumetric representation, we voxelize the points [43]. For each 3D point $p = (x_p, y_p, z_p)$ we compute integer voxel index (i, j, k) by dividing by the voxel size v_s and rounding.

$$i = \text{rnd}\left(\frac{x_p}{v_s}\right),\ j = \text{rnd}\left(\frac{y_p}{v_s}\right),\ k = \text{rnd}\left(\frac{z_p}{v_s}\right), \tag{5}$$

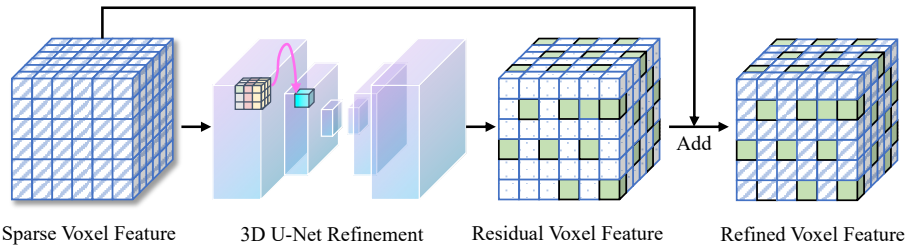


Fig. 3: Architecture of Sparse 3D decoder. Sparse 3D features are fed into a 3D U-Net for processing, which predicts residual features for each voxel. These residual features are then added to the original 3D voxel features to obtain the refined features.

where rnd$(\cdot)$ denotes rounding to the nearest integer.

Let $S_{i,j,k}$ be the set of all points falling into voxel (i, j, k) and $f_p$be the image feature corresponding to each point $p \in S_{i,j,k}$ The features within this voxel are aggregated via average pooling along the channel dimension, resulting in the voxel feature $V_{i,j,k}$:

$$V_{i,j,k} = \frac{1}{|S_{i,j,k}|} \sum_{p \in S_{i,j,k}} f_p. \tag{6}$$

3.3 Feature Refinement and 3D Gaussians Prediction

Feature refinement. To improve the spatial consistency and structural fidelity of the voxel representation, we apply an explicit voxel feature refinement stage as shown in Fig. 3. Given an input voxel grid V (with per-voxel feature vectors), a sparse convolutional 3D U-Net [7] $\mathcal{R}$ predicts a residual voxel field R:

$$R = \mathcal{R}(V), \quad R_i \in \mathbb{R}^{\mathcal{V} \times C}, \tag{7}$$

where $\mathcal{V}$ denotes the set of occupied voxels and the refined voxel features are obtained by a residual update:

$$V' = V + R, \quad V'_i \in \mathbb{R}^{\mathcal{V} \times C}, \tag{8}$$

The refinement network is implemented with hierarchical sparse 3D convolutional blocks, symmetric encoder–decoder stages, and upsampling layers connected by skip connections. This architecture enables multi-scale fusion of local and global geometric context while keeping computation efficient through sparsity. The residual formulation encourages the network to learn correction terms (fine geometric detail and consistency cues) rather than relearning the entire feature content, which empirically stabilizes training and preserves the coarse voxel information supplied by the lifting stage.

3D Gaussians prediction. The output of our network for each voxel v is a set of learnable Gaussian parameters $\{[\bar{\mu}_j, \bar{\alpha}_j, \Sigma_j, c_j] \in \mathbb{R}^{38}\}$. These include the offset of the Gaussian center $\bar{\mu}_j$, opacity $\bar{\alpha}_j$, covariance Σ_j, and spherical

harmonic color representation c_j. To obtain the final rendering parameters, we apply the following transformations:

$$\begin{aligned} \mu_j &= r \cdot (\sigma(\bar{\mu}_j) - 0.5) + \text{Center}_j, \\ \alpha_j &= \sigma(\bar{\alpha}_j), \end{aligned} \tag{9}$$

where μ_j is the predicted 3D Gaussian center, and Center_j is the centroid of voxel v. We utilize the sigmoid activation $\sigma(\cdot)$ to restrict the learnable offset within a localized neighborhood. Specifically, the -0.5 shift facilitates symmetrical, bi-directional movement from the voxel center, while r (set to $3\times$ voxel size) acts as a scaling factor to control the effective spatial extent of these refinements.

3.4 Training Objectives

Our network predicts a collection of 3D Gaussians $\{(\mu_v, \alpha_v, \Sigma_v, c_v)\}_{v\in\mathcal{V}}$. These per-voxel Gaussians are subsequently used to synthesize images at novel camera poses. To ensure a fair comparison and maintain benchmarking consistency with SOTA feed-forward methods, we follow the training protocol established by DepthSplat [54]. Specifically, the network is trained end-to-end using ground-truth RGB images as supervision. For a forward pass that renders M novel views, we optimize a combined photometric and perceptual loss:

$$\mathcal{L} = \sum_{m=1}^{M} \left(\mathcal{L}_{\text{MSE}}(I^{(m)}_{\text{render}}, I^{(m)}_{\text{gt}}) + \lambda \mathcal{L}_{\text{LPIPS}}(I^{(m)}_{\text{render}}, I^{(m)}_{\text{gt}}) \right), \tag{10}$$

where M is the number of novel views rendered in a single pass. Following DepthSplat [54], the weight λ for the perceptual loss $\mathcal{L}_{\text{LPIPS}}$ is set to 0.05.

4 Experiments

4.1 Experimental Setup

Datasets. We train our method using two expansive datasets, RealEstate10K [65] and ScanNet [8], and evaluate its performance on the held-out test splits of both. For RealEstate10K, we adopt the conventional partition of 67,477 training scenes and 7,289 test scenes. As for ScanNet, which consists of 1,513 videos of indoor scenes, we follow past work [13, 51, 64] in using roughly 100 scenes for training and 8 scenes for evaluation. These datasets span a wide variety of environments, including indoor and outdoor real-estate walkthroughs (RealEstate10K), and real-world videos of numerous scenes suitable for indoor robot applications (ScanNet). We resize training and test images to 256×256.

Baselines. We benchmark VolSplat against several recent feed-forward methods for sparse-view novel view synthesis, including both pixel-aligned and enhanced pixel-aligned Gaussian splatting approaches. Pixel-aligned methods predict Gaussian parameters on a per-pixel basis in image space before unprojecting

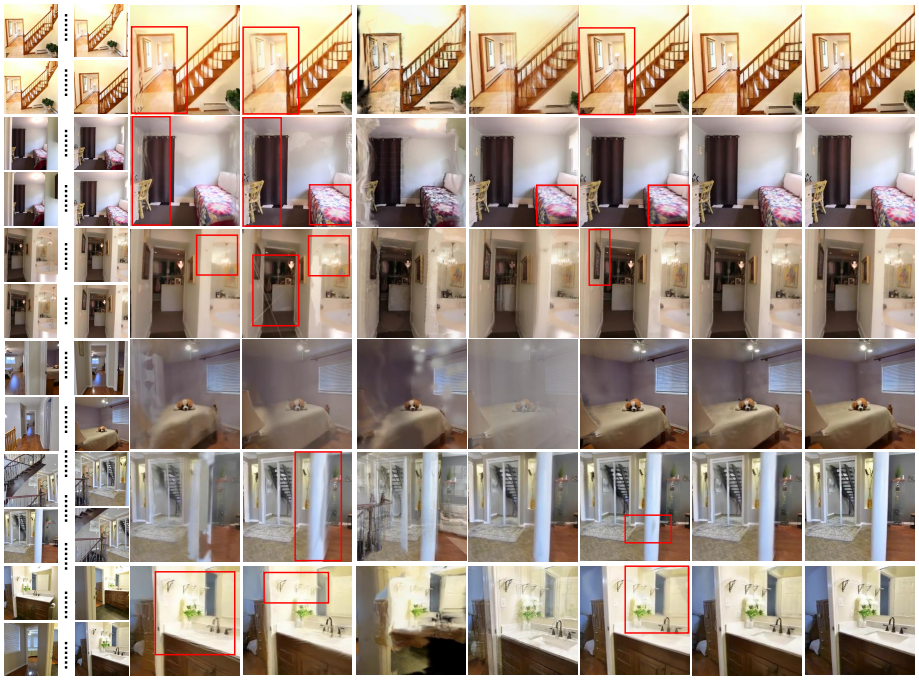


Fig. 4: Qualitative comparison on RealEstate10K [65]. We compare VolSplat against SOTA pixel-aligned baselines under sparse-view inputs. While competing methods often suffer from blurring and geometric distortions in complex environments, VolSplat leverages its voxel-aligned prediction to maintain superior visual fidelity and structural consistency.

to 3D. These include pixelSplat [3], MVSplat [5], FreeSplat [51], TranSplat [60] and DepthSplat [54]. Gaussian Graph Network (GGN) [63], AnySplat [17] and WorldMirror [27] refines the pixel-aligned approach by modeling the relationships between groups of predicted Gaussians across different views while building upon it. In contrast to both pixel-aligned and enhanced pixel-aligned methods, our VolSplat employs a voxel-aligned approach, predicting Gaussian primitives within a 3D voxel grid. This method aggregates multi-view evidence in 3D space, aligning Gaussian predictions to a voxel structure, which facilitates better geometric consistency and efficient redundancy reduction.

Metrics. For quantitative evaluation, we adopt standard image quality metrics commonly used in NVS, including pixel-level PSNR, patch-level SSIM [53],and feature-level LPIPS [62].

Implementation details. We implement VolSplat using PyTorch [36] and optimize the model with the AdamW [29] optimizer and a cosine learning rate schedule. The monocular Vision Transformer backbone is implemented using the xFormers [23] library. For the pre-trained Depth Anything V2 [56] backbone, we use a lower learning rate of 2×10^{-6}, while other layers are trained with a learning rate of 2×10^{-4} following DepthSplat [54]. For experiments on

Table 1: Quantitative comparisons on RealEstate10K [65]. The top section are pixel-aligned methods, and the middle section are methods that performs post-processing on pixel-aligned Gaussians. All baselines are retrained for fair comparison. Ground truth camera poses are provided for both AnySplat [17] and WorldMirror [27]. "OOM" represents that model cannot infer on a 96G GPU. We compare VolSplat against pixel-aligned and post-processing baselines under 6, 12, and 24 input views. VolSplat consistently achieves the best performance across all metrics and view settings.

Method	6v			12v			24v		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
pixelSplat [3]	28.95	0.900	0.163	OOM	OOM	OOM	OOM	OOM	OOM
MVSplat [5]	29.13	0.924	0.091	26.97	0.912	0.101	26.23	0.903	0.108
TranSplat [60]	29.62	0.928	0.084	28.00	0.920	0.089	26.65	0.884	0.115
DepthSplat [54]	30.52	0.931	0.079	28.54	0.919	0.088	26.26	0.880	0.115
GGN [63]	26.68	0.879	0.133	25.83	0.870	0.142	23.86	0.840	0.171
AnySplat [17]	19.05	0.576	0.305	19.86	0.627	0.288	20.21	0.658	0.279
WorldMirror [27]	24.86	0.819	0.079	26.03	0.859	0.072	26.47	0.875	0.071
Ours	**31.30**	**0.941**	**0.075**	**29.40**	**0.928**	**0.085**	**27.21**	**0.896**	**0.111**

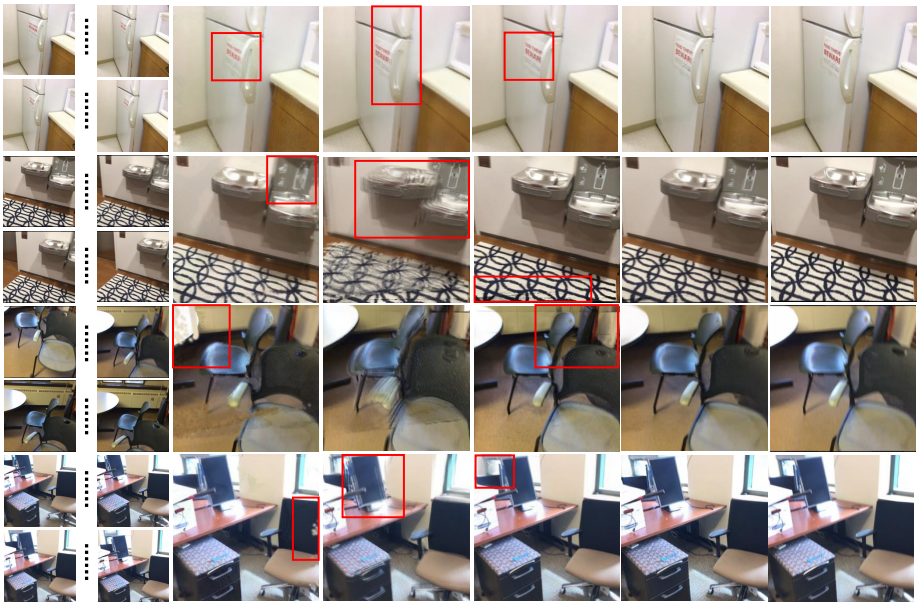


Fig. 5: Qualitative comparison on ScanNet [8]. Compared to recent baseline methods (FreeSplat [51], AnySplat [17], and WorldMirror [27]), VolSplat significantly reduces common floaters and visual artifacts. Our method produces cleaner object boundaries and a more coherent 3D scene reconstruction.

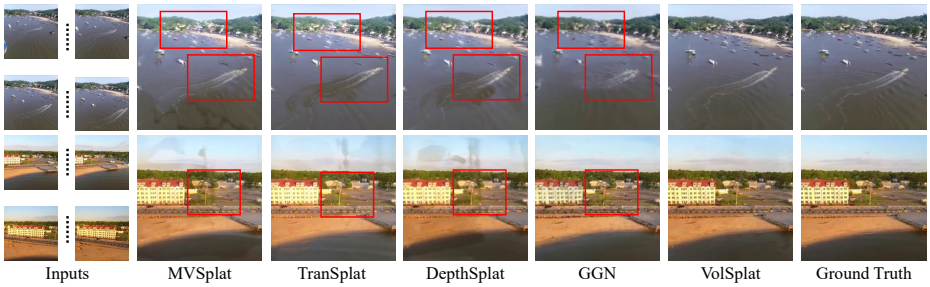


Fig. 6: Qualitative results on ACID [25]. Despite being trained solely on RealEstate10K [65], VolSplat generalizes exceptionally well to outdoor scenes. Our method reconstructs fine-grained details in natural landscapes and structures, producing photorealistic novel views where baseline methods often fail to maintain visual coherence.

Table 2: Quantitative comparisons on ScanNet [8] (6 input views). We compare VolSplat against SOTA methods that rely on post-hoc Gaussian fusion strategies. VolSplat outperforms all baselines.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
FreeSplat [51]	27.45	0.829	0.222
FreeSplat++ [52]	27.45	0.829	0.223
AnySplat [17]	19.45	0.626	0.344
WorldMirror [27]	25.83	0.819	0.136
VolSplat	**28.41**	**0.906**	**0.127**

Table 3: Cross-dataset generalization on ACID [25] (6 input views). Models trained on RealEstate10K [65] (indoor scenes) are directly evaluated on ACID [25] (outdoor scenes) without fine-tuning.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
MVSplat [5]	28.15	0.841	0.147
TranSplat [60]	28.17	0.842	0.146
DepthSplat [54]	28.37	0.847	0.141
GGN [63]	26.97	0.814	0.196
VolSplat	**32.65**	**0.932**	**0.092**

the RealEstate10K [65] and ScanNet [8] dataset, we train the model for 150,000 iterations using 4× NVIDIA H20 GPUs with a total batch size of 4. Following the setting of the baseline, we use 256×256 as input resolution. In the training stage, the number of input views is set to 6, and we evaluate the model's performance with same numbers of input views. We will make our codes and pre-trained models publicly available.

4.2 Experimental Results and Analysis

Comparisons with SOTA models. As shown in Tab. 1 and Tab. 2, we report VolSplat's performance compared to current mainstream pixel-aligned models [3,5,51,54,60] and their variants [17,27,63]. On both the RealEstate10K [65] and ScanNet [8] datasets, VolSplat achieves SOTA results. Notably, since AnySplat [17] and WorldMirror [27] are primarily designed for joint pose-geometry optimization in pose-free settings, they fail to effectively utilize the provided ground truth camera poses, resulting in lower performance compared to other baselines. Our experiments reveal a critical distinction between pixel-aligned and voxel-aligned paradigms. A key observation is that under sparse multi-view set-

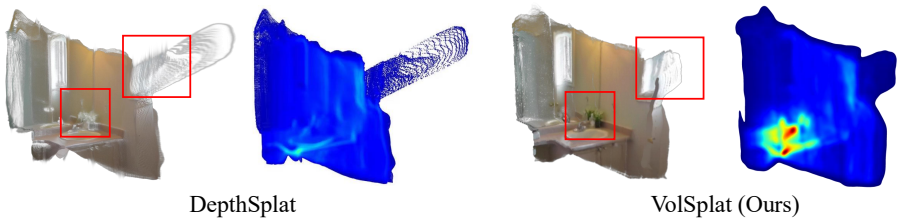


Fig. 7: Visualization of Gaussians and density maps. We compare the rendered results and the spatial distribution of Gaussian centers between DepthSplat [54] and VolSplat. DepthSplat is constrained by the pixel grid, resulting in a uniform but redundant distribution regardless of scene content. In contrast, VolSplat adaptively concentrates Gaussians on complex geometric structures (e.g., the washbasin boundaries) while remaining sparse in flat or empty regions, demonstrating a much more efficient and geometry-aware representation.

tings, all pixel-aligned models exhibit a significant degradation in performance. In contrast, VolSplat demonstrates promising performance to these challenging conditions. As illustrated in Fig. 4 and Fig. 5, images rendered by our method are largely free of the common floaters and artifacts that plague competing methods at object boundaries. This visual improvement stems directly from the ability of our model to resolve multi-view alignment issues within its 3D feature representation, resulting in cleaner edges and a more coherent 3D scene reconstruction.

Cross-dataset generalization. We assess the generalization capabilities of our model on unseen outdoor datasets to verify its broad reliability. To this end, we conducted a cross-dataset generalization experiment by taking our model pre-trained on the RealEstate10K [65] dataset and evaluating it directly on the ACID [25] dataset without any fine-tuning. As demonstrated in Tab. 3 and Fig. 6, VolSplat maintains significantly higher performance in this zero-shot transfer setting. We attribute this superior generalization to the inherent robustness of our voxel-aligned framework. Pixel-aligned models exhibit a much higher sensitivity to the variations in data complexity and distribution between different datasets. In contrast, VolSplat is less susceptible to these domain shifts.

Analysis of Gaussian density. A fundamental principle of 3D reconstruction is that the complexity of the representation should adapt to the complexity of the scene. Real-world environments contain a mix of simple, planar surfaces and intricate, high-frequency geometric details. An ideal model should allocate its descriptive capacity accordingly. However, pixel-aligned methods are inherently limited in this regard. Their paradigm of predicting one Gaussian per pixel results in a fixed number of primitives, predetermined by the input image resolution (e.g., $H \times W$ Gaussians from a reference view), regardless of whether the scene is a simple room or a complex outdoor environment.

In stark contrast, our voxel-aligned framework enables adaptive control over the density of the 3D Gaussians. By predicting primitives based on the occu-

Table 4: Analysis of voxel size. "PGS" stands for "average number of per-view Gaussians". We investigate the impact of voxel resolution on reconstruction quality and efficiency. A voxel size of 0.1 yields the optimal trade-off, achieving the best performance. Notably, further reducing the voxel size to 0.05 degrades performance due to the loss of coherent spatial context in overly sparse grids, while larger voxels fail to capture fine geometric details.

Voxel Size (cm)	PSNR ↑	SSIM ↑	LPIPS ↓	PGS	Memory(GB)	Inference Time(s)
0.05	29.34	0.919	0.092	65415	9.19	0.802
0.1 (default)	**29.40**	**0.928**	**0.085**	60523	9.04	0.768
0.5	27.33	0.899	0.108	59788	8.98	0.744
1	20.78	0.602	0.323	51806	8.74	0.739

Table 5: Ablation of sparse 3D decoder. "w/ 3D CNN" means replacing the 3D U-Net with a sparse 3D CNN, "w/o residual" means predicting refined voxel feature without residual design, and "w/o decoder" means removing the refinement stage. We validate the necessity of our specific refinement module. Replacing our proposed sparse 3D decoder for feature refinement results in performance drop. This demonstrates that our specific design is essential for refining the voxel features and producing high-quality 3D Gaussians.

Components	PSNR ↑	SSIM ↑	LPIPS ↓	Memory(GB)	Inference Time(s)
default	**29.40**	**0.928**	**0.085**	9.04	0.768
w/ 3D CNN	28.01	0.919	0.098	9.03	0.705
w/o residual	27.92	0.908	0.101	9.04	0.765
w/o decoder	27.47	0.901	0.102	8.99	0.687

pancy of 3D voxel features, VolSplat naturally allocates a higher concentration of Gaussians to regions of high geometric detail while using a sparser representation for simple or empty spaces.

This adaptive capability is quantitatively validated by the results we reported in Tab. 1, Tab. 2 and Tab. 3. Here, we analyze these findings in greater detail. The data shows that pixel-aligned methods consistently generate constant density of Gaussians, irrespective of the scene content. This leads to significant redundancy, as well as an insufficient representational capacity in areas with intricate details. Conversely, Gaussians of VolSplat demonstrate significant variance across different regions, confirming its ability to tailor complexity of the scene, as shown in Fig. 7. Notably, VolSplat often achieves superior rendering quality with a non-uniform set of Gaussians compared to the brute-force density of pixel-aligned approaches.

4.3 Ablation Study

In this section, we study the properties of our key components with 12 input views on the RealEstate10K [65] dataset.

Ablation of Voxel Size. The voxel size is a critical hyperparameter in our framework, as it dictates the resolution of the 3D feature grid. This choice involves a fundamental trade-off between the fidelity of the geometric representation and computational resource consumption. In Tab. 4, we analyze this trade-off by comparing our default setting against configurations with different voxels.

Using a small voxel size increases the granularity of the 3D grid, allowing the model to capture finer geometric details. It comes at a significant cost, substantially increasing memory usage and processing time due to the cubic growth of the voxel volume. Conversely, employing a large voxel size reduces the computational footprint but results in a coarser quantization of the 3D space. Our default configuration strikes an effective balance, achieving SOTA performance while maintaining manageable computational requirements.

Ablation of Model Architecture. Directly predicting Gaussians from the initial unprojected 3D features is less effective, particularly for challenging scenes with complex geometry or sparse viewpoints. To address this, we incorporate a 3D U-Net architecture to refine and enhance this raw feature volume, predicting the residual features. To validate the necessity and efficacy of this design, we conduct an ablation study with three variants: 1) removing the refinement module entirely, 2)directly predicting the refined feature, and 3) replacing the 3D U-Net with a standard 3D CNN.

The results, presented in Tab. 5, confirm our architectural choices. Removing the refinement stage altogether leads to a significant drop in performance, demonstrating that processing the initial voxel features is critical for producing a coherent 3D representation. While substituting our module with a sparse 3D CNN or removing the residual design yields better results than no refinement, it still falls short of the performance of our full model. The multi-scale feature fusion inherent in the U-Net structure and residual design are crucial for capturing both fine-grained local details and broader spatial context.

5 Conclusion

We address the fundamental limitations inherent in the prevailing pixel-aligned paradigm for feed-forward 3D Gaussian Splatting. We identify that existing methods suffer from a rigid coupling of Gaussian density to input image resolution and a high sensitivity to multi-view alignment errors. To overcome these challenges, we introduce VolSplat, a novel framework that fundamentally shifts the reconstruction process from 2D pixels to a 3D voxel-aligned space. By constructing 3D voxel feature and predicting Gaussians directly from this unified representation, our method effectively decouples the 3D scene from the constraints of the input views. This voxel-centric design enables adaptive control over Gaussian density according to scene complexity and inherently resolves alignment ambiguities, leading to more geometrically consistent and faithful reconstructions for downstream tasks.

References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Int. Conf. Comput. Vis. pp. 5855–5864 (2021) 3
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. In: Int. Conf. Comput. Vis. pp. 19697–19705 (2023) 3
3. Charatan, D., Li, S.L., Tagliasacchi, A., Sitzmann, V.: pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 19457–19467 (2024) 1, 2, 4, 9, 10, 11
4. Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., Su, H.: Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In: Int. Conf. Comput. Vis. pp. 14124–14133 (2021) 1
5. Chen, Y., Xu, H., Zheng, C., Zhuang, B., Pollefeys, M., Geiger, A., Cham, T.J., Cai, J.: Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In: Eur. Conf. Comput. Vis. pp. 370–386. Springer (2024) 1, 2, 4, 5, 6, 9, 10, 11
6. Chen, Y., Zheng, C., Xu, H., Zhuang, B., Vedaldi, A., Cham, T.J., Cai, J.: Mvsplat360: Feed-forward 360 scene synthesis from sparse views. Adv. Neural Inform. Process. Syst. **37**, 107064–107086 (2024) 1
7. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3d u-net: learning dense volumetric segmentation from sparse annotation. In: International conference on medical image computing and computer-assisted intervention. pp. 424–432. Springer (2016) 2, 7
8. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 5828–5839 (2017) 3, 8, 10, 11
9. Debevec, P.E., Taylor, C.J., Malik, J.: Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. p. 11–20. SIGGRAPH '96, Association for Computing Machinery, New York, NY, USA (1996). https://doi.org/10.1145/237170.237191 3
10. Fan, Z., Wang, K., Wen, K., Zhu, Z., Xu, D., Wang, Z., et al.: Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. Adv. Neural Inform. Process. Syst. **37**, 140138–140158 (2024) 3
11. Fridovich-Keil, S., Meanti, G., Warburg, F.R., Recht, B., Kanazawa, A.: K-planes: Explicit radiance fields in space, time, and appearance. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 12479–12488 (2023) 4
12. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 5501–5510 (2022) 4
13. Gao, Y., Cao, Y.P., Shan, Y.: Surfelnerf: Neural surfel radiance fields for online photorealistic reconstruction of indoor scenes. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 108–118 (2023) 8
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 770–778 (2016) 6
15. Huang, R., Mikolajczyk, K.: No pose at all: Self-supervised pose-free 3d gaussian splatting from sparse views. arXiv preprint arXiv: 2508.01171 (2025) 4
16. Ji, D., Kwon, J., McFarland, M., Savarese, S.: Deep view morphing. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 2155–2163 (2017) 3

17. Jiang, L., Mao, Y., Xu, L., Lu, T., Ren, K., Jin, Y., Xu, X., Yu, M., Pang, J., Zhao, F., Lin, D., Dai, B.: Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. arXiv preprint arXiv:2505.23716 (2025) 4, 9, 10, 11
18. Kang, G., Nam, S., Sun, X., Khamis, S., Mohamed, A., Park, E.: ilrm: An iterative large 3d reconstruction model. arXiv preprint arXiv:2507.23277 (2025) 1, 2
19. Kang, G., Yoo, J., Park, J., Nam, S., Im, H., Shin, S., Kim, S., Park, E.: Selfsplat: Pose-free and 3d prior-free generalizable 3d gaussian splatting. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 22012–22022 (2025) 4
20. Kaufman, A.E., Mueller, K.: Overview of volume rendering. The visualization handbook **7**, 127–174 (2005) 4
21. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023) 1, 3
22. Koneputugodage, C.H., Ben-Shabat, Y., Gould, S.: Octree guided unoriented surface reconstruction. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 16717–16726 (2023) 4
23. Lefaudeux, B., Massa, F., Liskovich, D., Xiong, W., Caggiano, V., Naren, S., Xu, M., Hu, J., Tintore, M., Zhang, S., Labatut, P., Haziza, D., Wehrstedt, L., Reizenstein, J., Sizov, G.: xformers: A modular and hackable transformer modelling library. https://github.com/facebookresearch/xformers (2022) 9
24. Levoy, M., Hanrahan, P.: Light field rendering. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. p. 31–42. SIGGRAPH '96, Association for Computing Machinery, New York, NY, USA (1996). https://doi.org/10.1145/237170.237199 3
25. Liu, A., Tucker, R., Jampani, V., Makadia, A., Snavely, N., Kanazawa, A.: Infinite nature: Perpetual view generation of natural scenes from a single image. In: Int. Conf. Comput. Vis. pp. 14458–14467 (2021) 3, 11, 12
26. Liu, H., Wang, Y., Li, C., Cai, R., Wang, K., Li, W., Molchanov, P., Wang, P., Wang, Z.: Flexgs: Train once, deploy everywhere with many-in-one flexible 3d gaussian splatting. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 16336–16345 (2025) 3
27. Liu, Y., Min, Z., Wang, Z., Wu, J., Wang, T., Yuan, Y., Luo, Y., Guo, C.: Worldmirror: Universal 3d world reconstruction with any-prior prompting. arXiv preprint arXiv:2510.10726 (2025) 4, 9, 10, 11
28. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Int. Conf. Comput. Vis. pp. 10012–10022 (2021) 6
29. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017) 9
30. Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B.: Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 20654–20664 (2024) 4
31. Meagher, D.: Geometric modeling using octree encoding. Computer graphics and image processing **19**(2), 129–147 (1982) 3
32. Miao, S., Huang, J., Bai, D., Yan, X., Zhou, H., Wang, Y., Liu, B., Geiger, A., Liao, Y.: Evolsplat: Efficient volume-based gaussian splatting for urban view synthesis. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 11286–11296 (2025) 4
33. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021) 1, 3

34. Ni, C., Wang, X., Zhu, Z., Wang, W., Li, H., Zhao, G., Li, J., Qin, W., Huang, G., Mei, W.: Wonderturbo: Generating interactive 3d world in 0.72 seconds. In: Int. Conf. Comput. Vis. pp. 27423–27434 (2025) 4
35. Odena, A., Dumoulin, V., Olah, C.: Deconvolution and checkerboard artifacts. Distill (2016). https://doi.org/10.23915/distill.00003, http://distill.pub/2016/deconv-checkerboard 5
36. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Adv. Neural Inform. Process. Syst. **32** (2019) 9
37. Ren, K., Jiang, L., Lu, T., Yu, M., Xu, L., Ni, Z., Dai, B.: Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. arXiv preprint arXiv:2403.17898 (2024) 4
38. Riegler, G., Osman Ulusoy, A., Geiger, A.: Octnet: Learning deep 3d representations at high resolutions. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 3577–3586 (2017) 4
39. Shi, D., Wang, W., Chen, D.Y., Zhang, Z., Bian, J., Zhuang, B., Shen, C.: Revisiting depth representations for feed-forward 3d gaussian splatting. arXiv preprint arXiv:2506.05327 (2025) 3
40. Wang, H., Xiang, A., Sun, H., Sun, P., Pan, C., Chen, Y., Hong, M., Wang, W., Chen, S., Chen, Y., et al.: Diffusion model as a generalist segmentation learner. arXiv preprint arXiv:2604.24575 (2026) 4
41. Wang, H., Zhou, K., Gu, B., Feng, Z., Wang, W., Sun, P., Xiao, Y., Zhang, J., Dong, H.: Transdiff: Diffusion-based method for manipulating transparent objects using a single rgb-d image. In: IEEE Int. Conf. Robot. Autom. pp. 7277–7283. IEEE (2025) 1
42. Wang, Q., Wang, Z., Genova, K., Srinivasan, P.P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.: Ibrnet: Learning multi-view image-based rendering. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 4690–4699 (2021) 1
43. Wang, T., Mao, X., Zhu, C., Xu, R., Lyu, R., Li, P., Chen, X., Zhang, W., Chen, K., Xue, T., et al.: Embodiedscan: A holistic multi-modal 3d perception suite towards embodied ai. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 19757–19767 (2024) 6
44. Wang, W., Cao, Q., Gao, S., Chen, D.Y., Xu, H., Bian, W., Peng, S., Cham, T.J., Zheng, C., Geiger, A., et al.: Feed-forward 3d scene modeling: A problem-driven perspective. arXiv preprint arXiv:2604.14025 (2026) 2
45. Wang, W., Chen, D.Y., Zhang, Z., Shi, D., Liu, A., Zhuang, B.: Zpressor: Bottleneck-aware compression for scalable feed-forward 3dgs. arXiv preprint arXiv:2505.23734 (2025) 1, 4
46. Wang, W., He, X., Gu, Y., Yang, Y., Zhang, Z., He, Y., Ding, Y., Hu, X., Chen, D.Y., He, Z., et al.: World-r1: Reinforcing 3d constraints for text-to-video generation. arXiv preprint arXiv:2604.24764 (2026) 4
47. Wang, W., Li, Z., Shi, J., Zhang, Z., Ye, B., Pollefeys, M., Chen, D.Y., Zhuang, B.: Trisplat: Simulation-ready feed-forward 3d scene reconstruction. arXiv preprint arXiv:2605.26115 (2026) 4
48. Wang, W., Zhao, H., Yang, Y., Chen, F., Zhang, Z., He, Y., Duan, Z., Chen, D.Y., Yang, Y., Zhuang, B.: Latent spatial memory for video world models. arXiv preprint arXiv:2606.09828 (2026) 4
49. Wang, W., Zhu, J., Zhang, Z., Wang, X., Zhu, Z., Zhao, G., Ni, C., Wang, H., Huang, G., Chen, X., et al.: Drivegen3d: Boosting feed-forward driving scene generation with efficient video diffusion. arXiv preprint arXiv:2510.15264 (2025) 2

50. Wang, Y., Chai, L., Luo, X., Niemeyer, M., Lagunas, M., Lombardi, S., Tang, S., Sun, T.: Learning efficient fuse-and-refine for feed-forward 3d gaussian splatting. arXiv preprint arXiv:2503.14698 (2025) 4
51. Wang, Y., Huang, T., Chen, H., Lee, G.H.: Freesplat: Generalizable 3d gaussian splatting towards free view synthesis of indoor scenes. Adv. Neural Inform. Process. Syst. **37**, 107326–107349 (2024) 1, 2, 4, 8, 9, 10, 11
52. Wang, Y., Huang, T., Chen, H., Lee, G.H.: Freesplat++: Generalizable 3d gaussian splatting for efficient indoor scene reconstruction. arXiv preprint arXiv:2503.22986 (2025) 11
53. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004) 9
54. Xu, H., Peng, S., Wang, F., Blum, H., Barath, D., Geiger, A., Pollefeys, M.: Depthsplat: Connecting gaussian splatting and depth. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 16453–16463 (2025) 1, 2, 3, 4, 5, 8, 9, 10, 11, 12
55. Xu, H., Zhang, J., Cai, J., Rezatofighi, H., Yu, F., Tao, D., Geiger, A.: Unifying flow, stereo and depth estimation. IEEE Trans. Pattern Anal. Mach. Intell. **45**(11), 13941–13958 (2023) 6
56. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth anything v2. arXiv:2406.09414 (2024) 9
57. Ye, B., Liu, S., Xu, H., Li, X., Pollefeys, M., Yang, M.H., Peng, S.: No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. arXiv preprint arXiv:2410.24207 (2024) 4
58. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 4578–4587 (2021) 1
59. Zhang, C., Liang, H., Chen, D.Y., Wu, Q., Plataniotis, K.N., Gambardella, C.C., Cai, J.: Panflow: Decoupled motion control for panoramic video generation. In: AAAI. vol. 40, pp. 12385–12393 (2026) 4
60. Zhang, C., Zou, Y., Li, Z., Yi, M., Wang, H.: Transplat: Generalizable 3d gaussian splatting from sparse multi-view images with transformers. In: AAAI. pp. 9869–9877 (2025) 9, 10, 11
61. Zhang, K., Bi, S., Tan, H., Xiangli, Y., Zhao, N., Sunkavalli, K., Xu, Z.: Gs-lrm: Large reconstruction model for 3d gaussian splatting. In: Eur. Conf. Comput. Vis. pp. 1–19. Springer (2024) 1
62. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 586–595 (2018) 9
63. Zhang, S., Fei, X., Liu, F., Song, H., Duan, Y.: Gaussian graph network: Learning efficient and generalizable gaussian representations from multi-view images. Adv. Neural Inform. Process. Syst. **37**, 50361–50380 (2024) 4, 9, 10, 11
64. Zhang, X., Bi, S., Sunkavalli, K., Su, H., Xu, Z.: Nerfusion: Fusing radiance fields for large-scale scene reconstruction. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 5449–5458 (2022) 8
65. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. arXiv preprint arXiv:1805.09817 (2018) 3, 8, 9, 10, 11, 12, 13
66. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 4490–4499 (2018) 4
67. Zhu, Z., Fan, Z., Jiang, Y., Wang, Z.: Fsgs: Real-time few-shot view synthesis using gaussian splatting. In: Eur. Conf. Comput. Vis. pp. 145–163. Springer (2024) 3

68. Ziwen, C., Tan, H., Zhang, K., Bi, S., Luan, F., Hong, Y., Fuxin, L., Xu, Z.: Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. arXiv preprint arXiv:2410.12781 (2024) 4