Learning Compact 3D Gaussians via Feed-Forward Point Fusion

Brandon Smart[1] Chuanxia Zheng[1,2] Iro Laina[1] Victor Adrian Prisacariu[1]
[1]University of Oxford [2]Nanyang Technological University
{brandon, cxzheng, iro, victor}@robots.ox.ac.uk

Abstract

We present SPLATT3RFUSION*, a feed-forward neural network that, given a set of unposed and uncalibrated images, directly reconstructs a compact and high-quality 3D Gaussian Splat representation of a scene. Unlike prior pixel-aligned feed-forward methods that typically predict one 3D Gaussian primitive per pixel in each image – producing severe redundancy, duplication, and ghosting on one physical surface – our approach efficiently fuses points in 3D space through a multi-scale octree structure, yielding a compact and coherent representation. Built upon VGGT, a foundation model for pose-free 3D geometry prediction,* SPLATT3RFUSION *introduces a Gaussian prediction branch that infers primitive parameters using only photometric supervision. We also introduce the ability to control the number of 3D Gaussians generated at test-time, allowing for a controllable tradeoff between PSNR and the number of 3D Gaussian primitives used. The model is efficient, reducing both memory usage and rendering cost, while achieving state-of-the-art results on RealEstate10k and ScanNet++.*

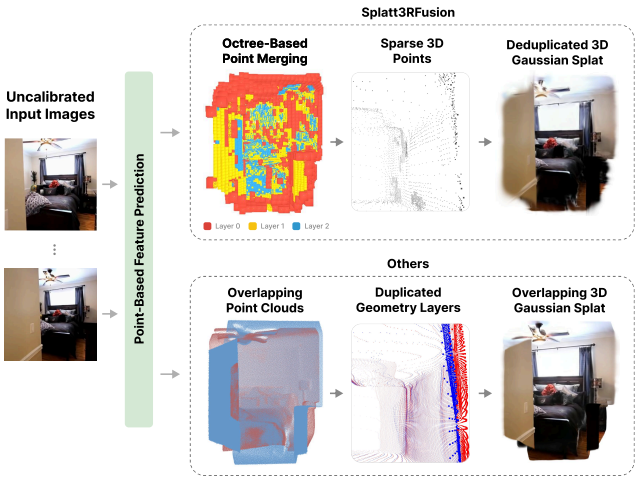


Figure 1. **SPLATT3RFUSION** is a feed-forward model that removes redundant 3D points produced by direct pixel-aligned prediction. It constructs a multi-scale octree to spatially merge points using cosine similarity of learned features, and then outputs a compact, deduplicated set of 3D Gaussians from the fused representations. We highlight the reduction of 'ghosting' on a wall, where regular feed-forward models predict multiple layers of points representing one physical surface, our method merges these points together into a physically coherent representation of the scene.

1. Introduction

We consider the problem of reconstructing photorealistic 3D scenes from a set of uncalibrated images using a feed forward neural network. Recent advancements in 3D reconstruction and NVS have been propelled by encoding 3D scenes using differentiable representations [24, 38, 49, 50]. While these methods have demonstrated impressive geometry and visual fidelity, they are far from being accessible to casual users, due to reliance on computationally intensive, per-scene optimization. Typically, they require dense input images and the corresponding camera parameters to reconstruct a scene, and take minutes or even hours to converge, posing significant barriers to real-world deployment.

Recent feed-forward 3D reconstructors directly predict 3D reconstructions from sparse images [5–10, 12, 22, 54, 55, 60, 65, 66, 70, 78]. Among these approaches, a growing number adopt 3D Gaussians [24] as their representation [5, 8, 9, 54, 55, 66, 78]. These methods typically regress pixel-aligned Gaussian parameters for every pixel in each input image. However, this *one-Gaussian-primitive-per-pixel* strategy leads to significant redundancy, particularly when reconstruction is performed from images with large overlapping regions. When a 3D point is observed across multiple views, current methods redundantly predict multiple duplicate and inconsistent Gaussians for the same physical point, violating the principle that each Gaussian primitive should uniquely represent its own spatial extent. This not only increases memory and rendering costs, but also degrades the geometry for the heavily covisible regions, especially as the number of input views increases.

Several methods have emerged to tackle the *non-pixel-aligned* 3D reconstruction from sparse views. DIG3D [64] and Gamba [48] use transformer- and Mamba-based architectures, respectively, where tokens are used as queries to

reason about 3D Gaussians. However, they are limited to *object*-level examples. For *scenes*, Gaussian Graph Networks (GGN) [75] reduce redundancy by establishing pixel correspondences across views, followed by feature fusion via weighted averaging. Very recently, concurrent work such as EVolSplat [37] and SplatVoxel [62] use voxel grids to realize non-pixel-aligned 3D reconstruction. However, the former does not explicitly fuse duplicated points, while the latter uses predicted opacity to merge the splat features, which does not consider the *semantic* attributes, and also requires a voxel grid aligned with the position and resolution of the target camera. Most importantly, (1) they still require accurate camera poses as input, which are difficult to obtain in sparse, in-the-wild settings; (2) they use uniform voxel grids, which prohibit different levels of detail in different regions of the scene; and (3) they do not allow for control over how many 3D Gaussian primitives are generated at inference time.

In this paper, we take a further step towards removing the need for camera poses, and enabling spatially and semantically coherent fusion of physical 3D. We do so by introducing SPLATT3RFUSION, a pose-free, feed-forward method for in-the-wild 3D reconstruction and novel view synthesis that avoids redundant 3D Gaussian primitives by merging points in 3D space using a predicted, multi-scale octree representation of the 3D scene. SPLATT3RFUSION predicts non-pixel-aligned 3D Gaussian Splats in a single forward pass from a sparse set of uncalibrated images.

At the core of SPLATT3RFUSION is a multi-scale 3D octree structure, designed to spatially group and merge 3D points that represent the same physical point in space. To this end, we build upon VGGT [59], a recent pose-free feed-forward 3D reconstructor that predicts 3D pointmaps from multi-view images. Specifically, we first lift 2D pixels to 3D points using VGGT's DPT module. Then, instead of directly using these points to form 3D Gaussian Splats, we introduce a novel 3D point fusion module that merges 3D points in an octree structure. At each octree cell, we merge points whose feature vectors are sufficiently similar, producing a single 3D Gaussian primitive that represents the entire cell. This process is performed hierarchically from coarse to fine levels of the octree, allowing points to be merged at different scales depending on their spatial distribution and feature similarity.

While existing pose-free, feed-forward methods construct 3D Gaussian Splats by taking the union of several per-image 3D Gaussian Splats, by fusing points in 3D space we are able to have each 3D Gaussian primitive be more geometrically meaningful as the unique representation of its own spatial extent in a scene-aware manner. In addition, by controlling the threshold used to merge points at test-time, we can control the number of 3D Gaussian primitives generated, while maintaining dynamic allocation of 3D Gaussians throughout the scene. We observe that we can match the performance of NoPoSplat [68] using only 25% of the 3D Gaussian primitives on RealEstate10k [79], and match the performance of Splatt3R [51] on ScanNet++ [69] using only 15.5% of the primitives.

2. Related Work

2.1. Novel View Synthesis

3D Novel View Synthesis (NVS) has been widely studied in computer vision and graphics [1, 15, 28]. Recent advances in neural rendering have been driven by Neural Radiance Fields (NeRFs) [38], which represent scenes as continuous volumetric radiance fields parameterized by neural networks trained on densely collected image sets [3, 38, 39]. More recently, 3D Gaussian Splatting (3DGS) [24] has greatly increased the training and rendering speed of radiance fields by training a set of 3D Gaussian primitives to represent the radiance of each point in space, and rendering them through an efficient rasterization process. However, these methods typically require slow, per-scene optimization, making them less practical for real-world applications.

To address this, generalizable NVS pipelines haven been developed, which infer 3D representations directly from multi-view images [6, 10, 21, 22, 29, 32, 33, 47, 52, 53, 60, 63, 65, 70, 78]. By training on large scale datasets, these methods have evolved to work with sparse image sets [8, 9, 31, 34, 41] and even stereo image pairs [5, 12, 26, 79], significantly reducing the number of reference images required to obtain a radiance field for novel view synthesis.

Recent methods, such as pixelSplat [5], MVSplat [8], MVSplat360 [9], Splatter Image [54], Flash3D [54] and DepthSplat [66] use a set of 3D Gaussian primitives placed along camera rays explicitly calculated from known camera parameters. However, they assume the availability of camera intrinsics and extrinsics for each image at testing time, which limits their applicability to in-the-wild photo pairs. To address this, several methods aim to perform generalizable, feed-forward reconstruction without camera poses, including Splatt3R [51], NoPoSplat [68], Large Spatial Model [13] and GGRt [29]. However these methods predict redundant, overlapping 3D Gaussian primitives due to predicting one primitive for each pixel in each image. In contrast, we seek to reduce the number of 3D Gaussian primitives in the scene by fusing point predictions together.

2.2. Non-pixel-Aligned 3D Gaussian Splatting

A dynamic number of 3D Gaussian primitives are used to represent a scene in original 3DGS representations [24, 25], which split or merge 3D Gaussians using non-differentiable heuristics during per-scene optimization to better fit the scene geometry and appearance. However, they are not applicable to feed-forward methods [5, 8, 54], which directly

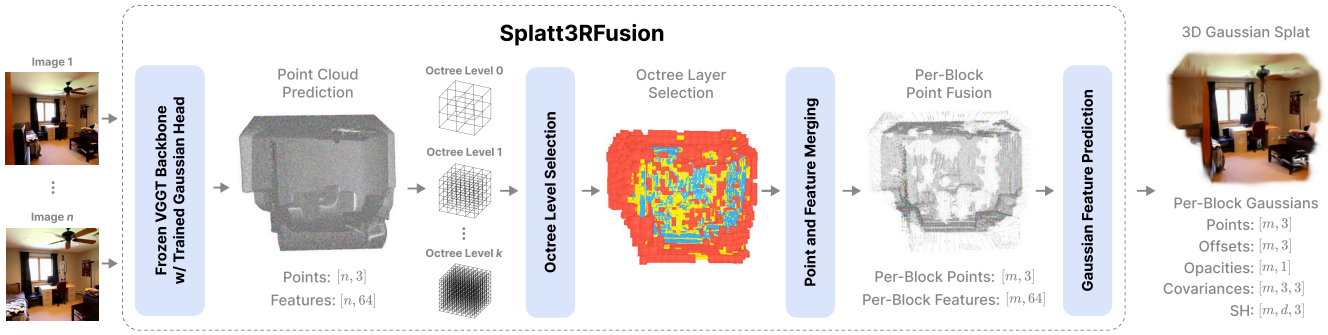


Figure 2. **Method overview.** We encode the uncalibrated images using VGGT's pretrained encoder, which we freeze during training. In addition to VGGT's prediction head for point clouds/depths and camera poses, we introduce a 'Gaussian Feature Head' that predicts feature vectors for downstream 3D Gaussian primitive prediction. These features are merged using an octree structure, and the resultant merged points and features are rendered as a 3D Gaussian Splat.

predict a fixed set of 3D gaussian primitives for per-pixel in each image.

To address this, Gaussian Graph Networks [75] builds a graph structure over per-pixel primitives to merge Gaussians by applying weighted averaging of features based on pixel correspondences. More closely related to our work, SplatVoxel [62] and EVolSplat [37] use voxel grids to merge 3D points before decoding 3D Gaussian primitives. The former focuses on dynamic scenes, using predicted opacities to merge splat features within each voxel, while the latter uses sparse CNNs to process voxel features. However, these methods still require known camera poses as input, which are difficult to obtain in sparse, in-the-wild settings. In addition, they use uniform voxel grids, which prohibit different levels of detail in different regions of the scene. Instead, we introduce a multi-scale octree structure based on feature similarity to dynamically merge 3D points in a scene-aware manner, without requiring camera poses as input.

We also note that voxel-based fusion of point primitives has been explored in the context of online 3D reconstruction and SLAM, such as being used to help encode truncated signed distance fields (TSDFs) [40, 42, 44], or implicit neural features which can be decoded into a TSDF using a learned neural network [30]. Unlike these methods, which focus on online reconstruction, we focus on using a static octree with dynamic spatial resolution to reduce the number of point primitives, and therefore 3D Gaussian primitives, in the final scene.

2.3. Pose-Free Feed-Forward 3D Reconstruction

Traditionally, the stereo reconstruction task involves a sequence of steps. Starting with keypoint detection and feature matching [11, 16, 35, 57], camera parameters estimation [36, 45, 77], establishing dense correspondences [2, 4, 20, 23, 71, 72], and triangulation of 3D points [17–19]. However, these methods rely on explicit correspondences, making them prone to failure when the overlap between images is limited, or when the input images are sparse.

Recently, DUSt3R [61] and MASt3R [27] addressed this challenge by learning to predict point maps for a pair of uncalibrated stereo images in one coordinate system. Several follow-up works aim to increase the number of images which can be used in a single feed-forward pass of the network [56, 67], or use memory mechanisms to avoid the expensive global alignment used by the original DUSt3R paper [58]. VGGT [59] achieves state-of-the-art results by increasing the size of the model and length of training, while alternating frame-wise and global attention. They also observe performance benefits by directly predicting 3D camera poses and depth maps, and creating 3D point clouds through unprojection.

These methods have achieved promising 3D reconstruction results even when there is little or no overlap between the images. While the raw point maps are sufficiently accurate for several downstream applications like pose estimation, they are not designed to be directly rendered. In contrast, our method augments VGGT to predict 3D Gaussian primitives, enabling fast and photo-realistic novel view synthesis, while avoiding the redundant, per-pixel points predicted by these feed-forward methods.

3. Method

Given a set of n uncalibrated images $\mathcal{I} = \{\boldsymbol{I}^i \in \mathbb{R}^{H\times W\times 3}\}_{i=\{1,\dots,n\}}$, our goal is to learn a mapping $f_{\boldsymbol{\theta}}$ from the images $\mathcal{I}$ to 3D Gaussian parameters:

$$f_{\boldsymbol{\theta}} : \{\boldsymbol{I}^i\}_{i=\{1,\dots,n\}} \rightarrow \{(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, \alpha_j, \boldsymbol{S}_j)\}_{j=\{1,\dots,m\}}, \quad (1)$$

where $\boldsymbol{\mu}_j \in \mathbb{R}^3$ is the mean position of the j-th Gaussian primitive, $\boldsymbol{\Sigma}_j \in \mathbb{R}^{3\times 3}$ is its covariance matrix, $\alpha_j \in \mathbb{R}$ is the opacity, and $\boldsymbol{S}_j \in \mathbb{R}^{3\times d}$ are the parameters of its view-dependent color model (here parameterized using d-degree spherical harmonics), and $f_{\boldsymbol{\theta}}$ is parameterized by learnable weights $\boldsymbol{\theta}$. Like other works, we reparameterize the covari-

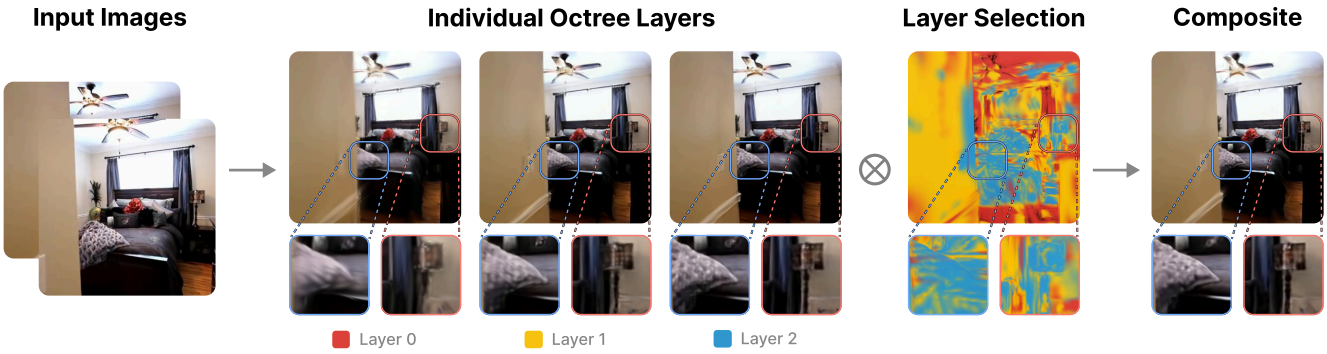


Figure 3. Our method predicts and supervises features at multiple levels of an octree structure. Here, we show the renderings obtained by converting the features at each level of the octree into 3D Gaussians, as well as the layer selection performed by our matching process, and the resulting composite 3D Gaussian Splat.

ance matrix with a rotation quaternion $\boldsymbol{q} \in \mathbb{R}^4$ and scale $\boldsymbol{s} \in \mathbb{R}^3$ to ensure positive semi-definite covariance matrices. Unlike most existing methods [5, 8, 54] that predict one Gaussian primitive per pixel in each image, resulting in significant redundancy, we aim to produce a compact set of m non-pixel-aligned Gaussian primitives by merging points in 3D space using a multi-scale octree structure, where $m \ll n \times H \times W$.

We achieve this by adding a new DPT head [46], that we refer to as the 'Gaussian Feature Head', to a pretrained VGGT model [59]. These 3D points from VGGT are first merged together in an octree, with the level of subdivision determined by the cosine similarity between 'matching features' that we additionally predict for octree level selection. The points in each cell are merged by taking the mean of their 3D positions, and their Gaussian feature vectors. Finally, after merging points together, we use a two-layer MLP to predict the attributes required to form a 3D Gaussian primitive at each merged point. We provide an overview in Fig. 8.

3.1. Background: Feed-Forward 3D Gaussians

Given a set of n images, generalizable 3D-GS methods [5, 8, 13, 51, 54, 55, 68] predict a set of pixel-aligned 3D Gaussian primitives. In particular, for each pixel $\boldsymbol{u} = (u_x, u_y, 1)$, the parameterized Gaussian primitive is predicted with its opacity α, offsets Δ, covariance $\boldsymbol{\Sigma}$ (expressed as rotation and scale), the parameters of the colour model $\boldsymbol{S}$, and either a position x or depth d. The location of each Gaussian is then either calculated from the depth as $\boldsymbol{\mu} = \mathbf{K}^{-1}\boldsymbol{u}d + \boldsymbol{\Delta}$, where $\mathbf{K}$ is the camera intrinsics, or calculated from the predicted position as $\boldsymbol{\mu} = x + \boldsymbol{\Delta}$. For multiple input images, the final set of Gaussian primitives is obtained by taking the union of all per-image Gaussian primitives. However, these methods typically predict one 3D Gaussian primitive for each pixel, causing redundancy and overlap. Our method performs an octree-based post-processing step to merge point predictions based on feature similarity.

3.2. Adapting VGGT for Novel View Synthesis

Given a set of uncalibrated images $\mathcal{I}$, VGGT [59] encodes these images using a DINOv2 model [43], then uses a transformer decoder alternating between frame-wise and global attention between these images. VGGT has two prediction heads, a Dense Prediction Transformer (DPT) head [46] for predicting depth maps $\boldsymbol{D}$, point maps $\boldsymbol{X}$, and another head for predicting camera poses. We introduce a third head, which we refer to as the *'Gaussian Feature Head'*, that runs in parallel to the existing two heads. This head is also a DPT network, but trained to predict a $K \times D_G$-dimensional feature vector for each pixel in each input image, where K is the number of octree levels, and D_G is the dimension of the Gaussian feature vector at each level. However, unlike existing works that predict 3D Gaussian parameters directly from these features, we use these features for octree-based point fusion first, which is discussed in detail in Sec. 3.3. Later, after fusion in the octree, these Gaussian feature vectors are fed into an MLP that predicts covariances (parameterized by rotation quaternions $q \in \mathbb{R}^4$ and scales $s \in \mathbb{R}^3$), spherical harmonics ($\boldsymbol{S} \in \mathbb{R}^{3\times d}$) and opacities ($\alpha \in \mathbb{R}$) for each fused point. Additionally, we predict an offset ($\boldsymbol{\Delta} \in \mathbb{R}^3$) for each fused point, and parameterize the mean of the Gaussian primitive as $\boldsymbol{\mu} = x + \boldsymbol{\Delta}$. This allows us to construct a complete Gaussian primitive for each fused point in the final scene.

To reduce the computational cost, we freeze all the parameters of the pretrained VGGT model, and only train the Gaussian Feature Head and the final MLP. Following [5, 54, 55], we use different activation functions for each Gaussian parameter type: normalization for quaternions, exponential activations for scales and offsets, and sigmoid activations for opacities. Additionally, to aid in the learning of high-frequency color, we follow other works [13]

Algorithm 1 Adaptive Octree Algorithm

Require: Points $P \in \mathbb{R}^{N\times 3}$, per-layer features $F_K \in \mathbb{R}^{N\times K\times D}$, normalized matching features $F_M \in \mathbb{R}^{N\times D_M}$, base voxel size v_0, resolution factor $r \geq 2$, similarity threshold $\tau \in [-1, 1]$, maximum depth K

bestLevel $\leftarrow K \cdot \mathbf{1}_N$ ▷ Initialize all points to finest level
for $k \in \{K, K-1, \ldots, 0\}$ **do** ▷ Iterate from the finest to the coarsest level of the octree
  $v_k \leftarrow v_0/r^k$
  $I_k \leftarrow \lfloor P/v_k \rfloor$ ▷ Integer voxel indices
  $C_k \leftarrow \text{CellHash}(I_k, k)$ ▷ Calculate the level-dependent hash of the voxel
  if $k = K$ **then**
    $C_{final} \leftarrow C_k$ ▷ Initialize the selected cells at the finest level
    continue
  end if
  $(U, I, N_c) \leftarrow \text{UniqueWithCounts}(C_k)$ ▷ List of cells, point to cell mapping and points-per-cell
  $\bar{F}_M \leftarrow \text{Normalize}(\text{AggregateMean}(F_M, I, N_c))$ ▷ Calculate normalized feature per cell
  $s \leftarrow \sum_{d=1}^{D_M} F_M \odot \bar{F}_{M,norm}[I]$ ▷ Per-point cosine similarity to average cell feature
  $S \leftarrow \text{AggregateSum}(s, I)$; $\bar{s} \leftarrow S./N_c$ ▷ Mean similarity per cell
  $M_{cells} \leftarrow (\bar{s} \geq \tau)$; $M_{points} \leftarrow M_{cells}[I]$ ▷ Points in cells with similarity above the selected threshold
  bestLevel$[M_{points}] \leftarrow k$ ▷ Update the best-level mapping for the selected points
  $C_{final}[M_{points}] \leftarrow C_k[M_{points}]$
end for
$(U_f, I_f, N_f) \leftarrow \text{UniqueWithCounts}(C_{final})$
$F_{\text{sel}}[n] \leftarrow F_L[n, \text{bestLevel}[n], :] \; \forall n \in \{1, \ldots, N\}$
$\bar{F} \leftarrow \text{AggregateMean}(F_{\text{sel}}, I_f, N_f)$ ▷ Mean Gaussian features for selected octree cells
$\bar{P} \leftarrow \text{AggregateMean}(P, I_f, N_f)$ ▷ Mean positions for selected octree cells
$L \leftarrow \text{AggregateReduce}(\text{bestLevel}, I_f)$ ▷ Octree level for selected octree cells
return $\bar{F}, \bar{P}, L$

and predict RGB feature vectors using a one-layer MLP directly from the input images, whose features are concatenated with the Gaussian feature vectors before predicting the 3D Gaussian primitive attributes. Following the convention of predicting the 3D locations of all points in the first image's local coordinate system, predicted covariances and spherical harmonics are also considered as being in the first image's local coordinate system. Note that, we construct point clouds from VGGT using the predicted camera poses and depth maps, rather than using the point cloud prediction head.

3.3. Multi-Scale Octree-based 3D Point Fusion

The key component of SPLATT3RFUSION is a multi-scale octree structure, designed to spatially fuse 3D points that represent the same physical point in space. This allows us to reduce redundancy in the predicted 3D Gaussian primitives, especially with respect to regions with high view overlap.

In particular, given the set of $N = W \times H \times n$ 3D points predicted by VGGT, and their corresponding Gaussian feature vectors ($N \times K \times D_G$) from the Gaussian Feature Head, along with additional "matching features" ($N \times D_M$), we build a multi-scale octree with K levels. In practice, the "matching features" are either learned from a new MLP head, or derived from a frozen, pretrained DINOv2 [43] model, upscaled to the original image resolution using FeatUp [14]. These features are used to determine whether points in a given octree cell should be merged together. In particular, during processing, we consider K layers of the octree, where each layer of the octree is a voxel grid with voxel size $v_k = v_0/2^k$, where v_0 is a hyperparameter for the base voxel size. At each voxel i in each layer k, we compute a "matching score" s_k^i using the average cosine similarity between each point's matching feature and the mean matching feature inside that voxel.

$$s_k^i = \frac{1}{|\mathcal{P}_k^i|} \sum_{j \in \mathcal{P}_k^i} \frac{\boldsymbol{f}_j \cdot \bar{\boldsymbol{f}}_k^i}{\|\boldsymbol{f}_j\| \|\bar{\boldsymbol{f}}_k^i\|}, \tag{2}$$

where $\mathcal{P}_k^i$ is the set of point indices inside voxel i at layer k, $\boldsymbol{f}_j$ is the matching feature of point j, and $\bar{\boldsymbol{f}}_k^i$ is the mean matching feature of all points inside voxel i at layer k. For any voxel where the average cosine similarity is below a selected "merging threshold" τ, we instead consider the finer layer of the voxel grid.

This process continues from the coarsest layer of the octree to the finest layer, until all voxels have either been selected for merging, or the finest layer has been reached. The final set of selected voxels across all layers of the octree determines the final set of merged points. The point positions

Method	Close ($\phi \geq 0.9, \psi \geq 0.9$)			Medium ($\phi \geq 0.7, \psi \geq 0.7$)			Wide ($\phi \geq 0.5, \psi \geq 0.5$)			Very Wide ($\phi \geq 0.3, \psi \geq 0.3$)		
	PSNR ↑	LPIPS ↓	# Gauss. ↓	PSNR ↑	LPIPS ↓	# Gauss. ↓	PSNR ↑	LPIPS ↓	# Gauss. ↓	PSNR ↑	LPIPS ↓	# Gauss. ↓
Ours - $\tau = 0.995$	28.84	0.085	**74.4K**	25.59	0.115	**76.7K**	24.59	0.126	**82.5K**	23.96	0.128	**92.2K**
Ours - $\tau = 0.999$	**28.11**	**0.080**	91.8K	**25.75**	0.111	94.7K	**24.71**	0.122	101.3K	**24.05**	0.125	109.0K
Splatt3R [51]	26.41	0.081	524.3K	24.80	**0.102**	524.3K	24.43	**0.109**	524.3K	**24.05**	**0.111**	524.3K

Table 1. **Results on ScanNet++**. We report similar or superior PSNR across most scenes, despite using only 15.5% of the 3D Gaussian primitives on average. We report our results using the octree similarity threshold of $\tau = 0.995$ and $\tau = 0.999$. Best results are **bolded**.

and Gaussian features for each point inside the final voxels are averaged together to create a single new point and feature vector to represent each voxel. In this way, clusters of similar points predicted from nearby viewpoints can be combined, while maintaining a dynamic spatial resolution that assigns more points to densely detailed regions of the scene. This is as opposed to a uniform voxel grid [62], where a constant spatial resolution is used throughout the scene. For each point, we predict $K \times D_G$ feature vectors – one for each level of the octree – and during merging we select the gaussian feature corresponding to the selected level of the octree.

We provide a visualization of the 3D Gaussians associated with the features at each layer of the octree, as well as an example of the layer selection in Fig. 3, and provide a pseudo-code description of the algorithm in Algorithm 1.

3.4. Training Procedure and Loss Functions

Following existing works [5, 8, 54], we train our model using only rendering losses. The training loss $\mathcal{L} = \lambda_{\text{mse}}\ell_2 + \lambda_{\text{lpips}}\mathcal{L}_{\text{lpips}}$ is caculated as a linear combination of mean squared error ℓ_2 and LPIPS losses ($\mathcal{L}_{\text{lpips}}$) [73], between renderings from the predicted 3D Gaussian Splats and ground truth images, where λ_{mse} and λ_{lpips} are hyperparameters.

However, due to the misalignments between predicted scene geometry and the true scene geometry, we utilize a two-stage training strategy. First, we exclusively supervise the reconstructed 3D scene from the perspective of the context cameras. During training, each sample consists of k_c 'context' images which we use to reconstruct the scene. Then rendering losses are calculated using renders of the scene from the camera poses predicted for the context images by VGGT. During the second stage of training, we additionally use k_t target images, like existing works, however we optimize the camera pose use photometric mean squared error loss before calculating the target rendering losses. Because our entire octree pipeline is differentiable with respect to the features predicted by our network, we can perform end-to-end training of the network. In addition to supervising the 3D Gaussians obtained after the layer selection process, we additionally convert the features from each layer of the octree into 3D Gaussian primitives, and supervise these renderings with the same loss weights. The final loss is given as the sum of the losses for each rendering.

When training on ScanNet++, we follow Splatt3R [51] and apply covisibility-based loss masks to focus the loss on regions of the scene which are visible to the context views. For RealEstate10k, since we do not have covisibility information, we following existing work and apply rendering losses across the entire image. Note that, rather than using a single octree merging threshold τ, we randomly sample a different threshold for each batch, so that different merging thresholds can be selected at test-time. Thresholds are sampled from the range $[0.8, 0.999]$, sampling logarithmically from the space $1 - \tau$.

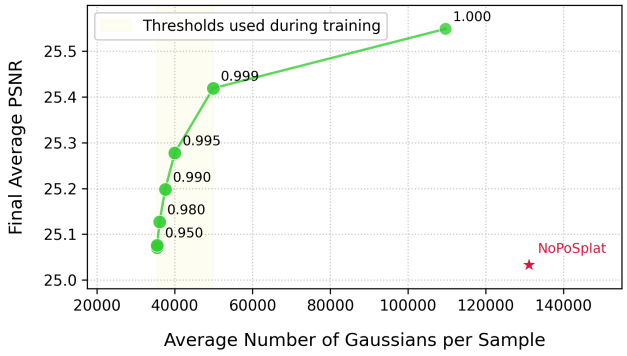
4. Experiments

4.1. Settings

Implementation details. SPLATT3RFUSION is built upon VGGT [59]. Unless otherwise specified, we randomly sample $k_c = 2$ context images and $k_t = 3$ target images during training. We train our model at a resolution of 518×518 at a batch size of 8, using $\lambda_{\text{mse}} = 1.0$ and $\lambda_{\text{lpips}} = 0.05$. We optimize using the AdamW optimizer at a learning rate of 3.0×10^{-5}, with a weight decay of 0.05, and a gradient clip value of 0.01. The primary experiments are performed on 4x 48GB RTX A6000 GPUs, and secondary experimental results and ablations are reported on training runs completed with 2x 24GB RTX A5000 GPUs. Our models are trained for 70,000 iterations during the first stage of training, for 5,000 iterations during the second stage. For our octree processing, unless otherwise specified, we use a two-level octree, with a coarse layer of size 0.01 (in the normalized prediction space used by VGGT), and a finer layer of size 0.005.

Method	PSNR ↑	LPIPS ↓	# Gauss. ↓
NoPoSplat [68]	25.03	0.160	131.1k
FLARE [76]	23.77	0.191	131.1k
Ours - $\tau = 0.995$	**25.27**	**0.103**	**40.0k**

Table 2. **Results on RealEstate10k**. We report state-of-the-art results, despite using fewer 3D Gaussians. Best results are **bolded**.



Threshold	PSNR ↑	LPIPS ↓	Num Gaussians ↓
$\tau = 0.8$	25.07	0.110	35.4k
$\tau = 0.9$	25.07	0.110	35.4k
$\tau = 0.95$	25.08	0.110	35.5k
$\tau = 0.98$	25.12	0.108	36.1k
$\tau = 0.99$	25.19	0.106	37.5k
$\tau = 0.995$	25.27	0.103	40.0k
$\tau = 0.999$	25.41	0.097	49.9k
$\tau = 1.0$	25.55	0.091	109.6k

Figure 4. Results on RealEstate10k obtained when modifying the octree merging threshold at test-time. We observe a controllable trade-off between PSNR and the number of 3D Gaussian primitives in the scene.

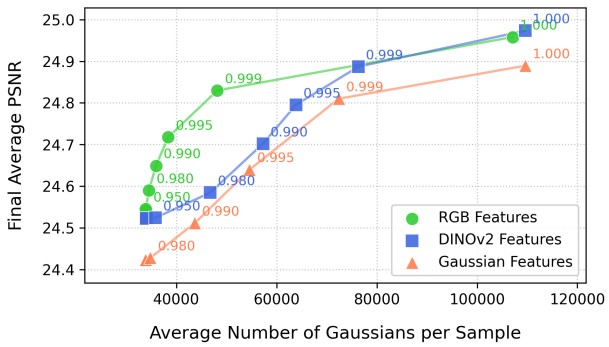


Figure 5. Results on RealEstate10k using different features for merging: RGB, Gaussian and Truncated DINOv2 features

Datasets and evaluation protocol. For evaluation, we report PSNR and LPIPS (Learned Perceptual Image Patch Similarity) [74]. We also report the number of 3D Gaussians used in the scene to demonstrate the compactness of the representation. All models are evaluated on two standard datasets for novel view synthesis: ScanNet++ [69] and RealEstate10k [79]. For ScanNet++, we follow the same experimental setup as Splatt3R [51], using the same training and testing splits, and evaluating on four different testing splits with varying levels of difficulty and loss masking. For RealEstate10k, we follow existing works [5, 8, 68], and report metrics across the entire image. Following NoPoSplat [68] and FLARE [76], we report all results after performing 100 steps of test-time camera pose optimization. We note that our backbone model VGGT is designed to work with 518×518 images, whereas existing RealEstate10k are reported on 256×256 images. Therefore, we upsample the 256×256 images to 518×518 to perform inference, and downsample our renderings to 256×256 to perform metric calculation for fair comparison.

4.2. Results

We begin by reporting our quantitative results for ScanNet++ in Tab. 1 and our results for RealEstate10k in Tab. 2.

On ScanNet++, we see that we are able to match or surpass the performance of Splatt3R across different testing splits at $\tau = 0.995$ despite using on average 15.5% of the 3D Gaussian primitives that Splatt3R uses. This demonstrates the redundancy of the dense 3D Gaussian primitives used by Splatt3R, and the effectiveness of our merging procedure. On RealEstate10k, we are able to surpass the performance of NoPoSplat [68] and FLARE [76] while using fewer primitives. We also note that NoPoSplat and FLARE use the ground truth intrinsics for the input images, whereas our method does not require any known camera intrinsics. Because SPLATT3RFUSION natively operates at a resolution of 518×518, we directly predict 536.6k point primitives, but our octree merging procedure reduces this to 61.9k 3D Gaussian primitives, less than half of the 3D Gaussians used by NoPoSplat and FLARE.

Next, we explore our adjustable octree merging threshold τ. By using randomly selected values for τ for each batch at training time, at test-time we can use this threshold to control the number of 3D Gaussian primitives in the final scene. In Fig. 4, we show the results of our method on RealEstate10k for different values of this threshold. We see that the threshold allows for a controllable tradeoff between PSNR and the number of 3D Gaussians in the scene. For low threshold values, where points are combined in the octree more aggressively, we are able to maintain the performance of NoPoSplat with roughtly $\frac{1}{3}$ of the 3D Gaussian primitives, whereas if we use a threshold of $\tau = 1.0$ – indicating that no merging should be performed except for on the final layer of the octree – then we are able to achieve a PSNR 0.52dB higher than NoPoSplat.

	2 Image Inputs		4 Input Images		8 Input Images		12 Input Images		16 Input Images	
Threshold	PSNR	Gaussians	PSNR	Gaussians	PSNR	Gaussians	PSNR	Gaussians	PSNR	Gaussians
$\tau = 0.8$	28.16	67.4k	27.51	94.1k	27.09	132.9k	26.67	166.8k	26.31	196.6k
$\tau = 0.9$	28.16	67.4k	27.51	94.1k	27.09	132.9k	26.67	166.8k	26.31	196.6k
$\tau = 0.95$	28.17	67.4k	27.52	94.2k	27.09	133.0k	26.67	167.0k	26.30	196.9k
$\tau = 0.98$	28.28	68.3k	27.58	95.7k	27.12	135.9k	26.69	171.2k	26.31	202.5k
$\tau = 0.99$	28.36	69.8k	27.63	98.5k	27.13	141.1k	26.69	178.7k	26.29	212.3k
$\tau = 0.995$	28.45	72.0k	27.67	102.6k	27.15	148.6k	26.69	189.3k	26.28	226.1k
$\tau = 0.999$	28.66	80.3k	27.77	117.8k	27.17	176.1k	26.69	227.5k	26.25	276.9k
$\tau = 1$	29.08	216.5k	28.02	332.5k	27.31	507.3k	26.80	659.7k	26.35	801.7k

Figure 6. Reconstruction of 3D Scenes from different numbers of input images on a model trained with pairs of images from ScanNet++. We observe non-linear increases in the number of 3D gaussians used by our octree.

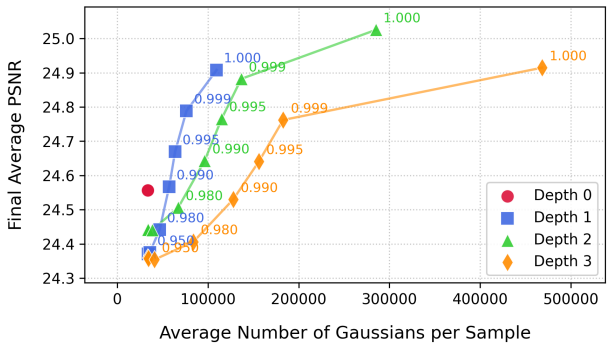


Figure 7. Results on RealEstate10k from training models with octrees of different depths

4.3. Ablation studies

For the ablations in this section, we compare results after performing Stage 1 of training, and report metrics on a subset of 250 samples from RealEstate10k. In Fig. 5, we explore using different features for 'matching' in the octree. We perform these experiments on 2x 24GB RTX A5000 GPUs, and therefore test with DINOv2 features truncated from 384 dimensions to 64. We observe similar performance between truncated DINOv2 features and learned Gaussian features, but observe that for similar thresholds using the RGB features – derived by running the input RGB values through a one layer MLP – are able to achieve similar PSNR scores with fewer 3D Gaussian primitives.

Unlike existing pose-free works which are limited to performing inference from a pair of images, SPLATT3RFUSION can perform inference from much larger collections of images. In Fig. 6, we train a model using two context images, and then report the results of our model when presented with different numbers of input images at test-time. We construct testing samples by taking the $\alpha, \beta = 0.9$ dataset on ScanNet++ specified by Splatt3R, and adding additional frames to the input which are between the two given frames. If a sufficient number of frames are not available, we take the closest frames from outside the given range. We see that our method can effectively perform inference from larger collections of images, and that the number of 3D Gaussian primitives in the final reconstruction grows non-linearly with the number of frames.

In Fig. 7, we show the performance of our method using different depths for the octree on RealEstate10k. The octree of depth 1 is equivalent to a uniform voxel gird, and due to the lack of levels to select between, changing the selected threshold value has no effect on the output. Among the deeper octrees, we see that the best tradeoff between PSNR and the number of 3D Gaussian primitives in the scene is achieved with an octree of depth 2, although the highest PSNR is achieved with an octree of depth 3, at the cost of a higher number of primitives.

5. Conclusion

We present SPLATT3RFUSION, a feed-forward model for predicting 3D Gaussian Splats from uncalibrated stereo images, without relying on camera intrinsics, extrinsics, or depth information. The key of our success lies in the proposed octree-based 3D point fusion module, which effectively merges 3D points that represent the same physical location in space. This differs from existing feed-forward pose-free methods that simply take the union of per-image 3D Gaussian Splats, resulting in many redundant and overlapping 3D Gaussians. With our octree-based merging, our SPLATT3RFUSION is able to significantly reduce the number of 3D Gaussians in the final representation, while maintaining visual rendering quality in two large-scale scene-level datasets.

Acknowledgments. Chuanxia Zheng is supported by NTU SUG-NAP and National Research Foundation, Singapore, under its NRF Fellowship Award NRF-NRFF17-2025-0009.

References

[1] Edward H Adelson and James R Bergen. *The plenoptic function and the elements of early vision*. MIT Press, 1991. 2

[2] Stephen T Barnard and Martin A Fischler. Computational stereo. *ACM Computing Surveys (CSUR)*, 1982. 3

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022. 2

[4] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *CVPR*, 2018. 3

[5] 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 (CVPR)*, pages 19457–19467, 2024. 1, 2, 4, 6, 7

[6] 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 (ICCV)*, pages 14124–14133, 2021. 2

[7] Yuedong Chen, Haofei Xu, Qianyi Wu, Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Explicit correspondence matching for generalizable neural radiance fields. *arXiv preprint arXiv:2304.12294*, 2023.

[8] 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. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 370–386. Springer, 2024. 1, 2, 4, 6, 7

[9] Yuedong Chen, Chuanxia Zheng, Haofei Xu, Bohan Zhuang, Andrea Vedaldi, Tat-Jen Cham, and Jianfei Cai. Mvsplat360: Feed-forward 360 scene synthesis from sparse views. In *Neural Information Processing Systems (NeurIPS)*, 2024. 1, 2

[10] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7911–7920, 2021. 1, 2

[11] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018. 3

[12] Yilun Du, Cameron Smith, Ayush Tewari, and Vincent Sitzmann. Learning to render novel views from wide-baseline stereo pairs. In *CVPR*, 2023. 1, 2

[13] Zhiwen Fan, Jian Zhang, Wenyan Cong, Peihao Wang, Renjie Li, Kairun Wen, Shijie Zhou, Achuta Kadambi, Zhangyang Wang, Danfei Xu, et al. Large spatial model: End-to-end unposed images to semantic 3d. *Advances in neural information processing systems*, 37:40212–40229, 2024. 2, 4

[14] Stephanie Fu, Mark Hamilton, Laura E. Brandt, Axel Feldmann, Zhoutong Zhang, and William T. Freeman. Featup: A model-agnostic framework for features at any resolution. In *The Twelfth International Conference on Learning Representations*, 2024. 5

[15] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Computer Graphics and Interactive Techniques*, 1996. 2

[16] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey Vision Conference*, 1988. 3

[17] Richard Hartley and Frederik Schaffalitzky. L/sub/spl infin//minimization in geometric reconstruction problems. In *CVPR*, 2004. 3

[18] Richard I Hartley and Peter Sturm. Triangulation. *Computer Vision and Image Understanding*, 1997.

[19] Richard I Hartley, Rajiv Gupta, and Tom Chang. Stereo from uncalibrated cameras. In *CVPR*, 1992. 3

[20] Hiroshi Ishikawa and Davi Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *ECCV*, 1998. 3

[21] Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. *arXiv preprint arXiv:2410.17242*, 2024. 2

[22] 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 (CVPR)*, pages 18365–18375, 2022. 1, 2

[23] Takeo Kanade, Atsushi Yoshida, Kazuo Oda, Hiroshi Kano, and Masaya Tanaka. A stereo machine for video-rate dense depth mapping and its new applications. In *CVPR*, 1996. 3

[24] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ToG*, 2023. 1, 2

[25] Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Weiwei Sun, Yang-Che Tseng, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. 3d gaussian splatting as markov chain monte carlo. *Advances in Neural Information Processing Systems*, 37:80965–80986, 2024. 2

[26] Haechan Lee, Wonjoon Jin, Seung-Hwan Baek, and Sunghyun Cho. Generalizable novel-view synthesis using a stereo camera. *CVPR*, 2024. 2

[27] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision (ECCV)*, pages 71–91. Springer, 2024. 3

[28] Marc Levoy and Pat Hanrahan. Light field rendering. In *SIGGRAPH*, 1996. 2

[29] 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. *ECCV*, 2024. 2

[30] Kejie Li, Yansong Tang, Victor Adrian Prisacariu, and Philip HS Torr. Bnv-fusion: Dense 3d reconstruction using bi-level neural volume fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6166–6175, 2022. 3

[31] Yaokun Li, Chao Gou, and Guang Tan. Taming uncertainty in sparse-view generalizable nerf via indirect diffusion guidance. *arXiv preprint arXiv:2402.01217*, 2024. 2

[32] 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. *ECCV*, 2024. 2

[33] 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 (CVPR)*, pages 7824–7833, 2022. 2

[34] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In *ECCV*, 2022. 2

[35] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 3

[36] Quan-Tuan Luong and Olivier D Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *IJCV*, 1996. 3

[37] Sheng Miao, Jiaxin Huang, Dongfeng Bai, Xu Yan, Hongyu Zhou, Yue Wang, Bingbing Liu, Andreas Geiger, and Yiyi Liao. Evolsplat: Efficient volume-based gaussian splatting for urban view synthesis. *arXiv preprint arXiv:2503.20168*, 2025. 2, 3

[38] B Mildenhall, PP Srinivasan, M Tancik, JT Barron, R Ramamoorthi, and R Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision (ECCV)*, 2020. 1, 2

[39] 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. 2

[40] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011. 3

[41] Zhangkai Ni, Peiqi Yang, Wenhan Yang, Hanli Wang, Lin Ma, and Sam Kwong. Colnerf: Collaboration for generalizable sparse input neural radiance field. In *AAAI*, 2024. 2

[42] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)*, 32 (6):1–11, 2013. 3

[43] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 4, 5

[44] Victor Adrian Prisacariu, Olaf Kähler, Ming Ming Cheng, Carl Yuheng Ren, Julien Valentin, Philip HS Torr, Ian D Reid, and David W Murray. A framework for the volumetric integration of depth images. *arXiv preprint arXiv:1410.0925*, 2014. 3

[45] René Ranftl and Vladlen Koltun. Deep fundamental matrix estimation. In *ECCV*, 2018. 3

[46] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 4

[47] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pages 10901–10911, 2021. 2

[48] Qiuhong Shen, Zike Wu, Xuanyu Yi, Pan Zhou, Hanwang Zhang, Shuicheng Yan, and Xinchao Wang. Gamba: Marry gaussian splatting with mamba for single view 3d reconstruction. *arXiv preprint arXiv:2403.18795*, 2024. 1

[49] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019. 1

[50] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:19313–19325, 2021. 1

[51] Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3r: Zero-shot gaussian splatting from uncalibarated image pairs. *arXiv preprint arXiv:2408.13912*, 2024. 2, 4, 6, 7

[52] Cameron Smith, Yilun Du, Ayush Tewari, and Vincent Sitzmann. Flowcam: Training generalizable 3d radiance fields without camera poses via pixel-aligned scene flow. In *NeurIPS*, 2023. 2

[53] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *European Conference on Computer Vision (ECCV)*, pages 156–174. Springer, 2022. 2

[54] 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 (CVPR)*, pages 10208–10217, 2024. 1, 2, 4, 6

[55] Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, João F Henriques, Christian Rupprecht, and Andrea Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. In *International Conference on 3D Vision (3DV)*, 2025. 1, 4

[56] Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. *arXiv preprint arXiv:2412.06974*, 2024. 3

[57] Miroslav Trajković and Mark Hedley. Fast corner detection. *Image and Vision Computing*, 1998. 3

[58] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *International Conference on 3D Vision (3DV)*, 2025. 3

[59] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt:

Visual geometry grounded transformer. *arXiv preprint arXiv:2503.11651*, 2025. 2, 3, 4, 6

[60] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021. 1, 2

[61] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20697–20709, 2024. 3

[62] Yiming Wang, Lucy Chai, Xuan Luo, Michael Niemeyer, Manuel Lagunas, Stephen Lombardi, Siyu Tang, and Tiancheng Sun. Splatvoxel: History-aware novel view streaming without temporal training. *arXiv preprint arXiv:2503.14698*, 2025. 2, 3, 6

[63] Christopher Wewer, Kevin Raj, Eddy Ilg, Bernt Schiele, and Jan Eric Lenssen. latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. In *European Conference on Computer Vision (ECCV)*, pages 456–473. Springer, 2024. 2

[64] Jiamin Wu, Kenkun Liu, Han Gao, Xiaoke Jiang, and Lei Zhang. Leangaussian: Breaking pixel or point cloud correspondence in modeling 3d gaussians. *arXiv preprint arXiv:2404.16323*, 2024. 1

[65] 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 (CVPR)*, pages 20041–20050, 2024. 1, 2

[66] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1, 2

[67] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. *arXiv preprint arXiv:2501.13928*, 2025. 3

[68] Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. In *International Conference on Learning Representations (ICLR)*, 2025. 2, 4, 6, 7

[69] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023. 2, 7

[70] 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 (CVPR)*, pages 4578–4587, 2021. 1, 2

[71] Jure Zbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *CVPR*, 2015. 3

[72] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *CVPR*, 2019. 3

[73] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

[74] 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 (CVPR)*, pages 586–595, 2018. 7

[75] Shengjun Zhang, Xin Fei, Fangfu Liu, Haixu Song, and Yueqi Duan. Gaussian graph network: Learning efficient and generalizable gaussian representations from multi-view images. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:50361–50380, 2024. 2, 3

[76] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 6, 7

[77] Zhengyou Zhang, Rachid Deriche, Olivier Faugeras, and Quang-Tuan Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 1995. 3

[78] Shunyuan Zheng, Boyao Zhou, Ruizhi Shao, Boning Liu, Shengping Zhang, Liqiang Nie, and Yebin Liu. Gps-gaussian: Generalizable pixel-wise 3d gaussian splatting for real-time human novel view synthesis. In *CVPR*, 2024. 1, 2

[79] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *SIGGRAPH*, 2018. 2, 7

Learning Compact 3D Gaussians via Feed-Forward Point Fusion

Supplementary Material

Figure 8. A qualitative comparison between our method (with threshold values of $\tau = 0.995$ and $\tau = 0.999$), alongside the results from Splatt3R. Underneath each image, we show the number of 3D Gaussian primitives predicted by the models. SPLATT3RFUSION runs at a resolution of 518×518, whereas Splatt3R runs at a resolution of 512×512.