GASPLAT: GEOMETRY-AWARE DISTRACTOR-FREE 3D GAUSSIANS SPLATTING UNDER SPARSE-VIEW SETTINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

3D Gaussian Splatting (3DGS) achieves real-time photorealistic 3D static scene reconstruction, yet its performance degrades significantly when sparse input images also contain dynamic transient objects. Existing distractor-free reconstruction methods misclassify static hard-to-reconstruct regions as distractors, sparse-view reconstruction methods don't block gradients of distractors and their vanilla combination enhances both static and transient regions. To tackle this new problem, we propose GASplat, a novel framework reconstructs static scene with sparse input containing distractors. Specifically, with aim of better distinguishing static and transient regions to identify distractors with memorization effect, we generate coarse masks through pretrained image matching model and geometry-aware filtering. Then, we enhance initial point clouds of high confident static regions with coarse masks. Next, we recover point clouds and masks of far background regions mitigating limitations of pretrained model. Finally, in order to precisely block gradients from transient objects and refine distractors masks, we design a progressive training schedule to reconstruct static scenes. Extensive experiments on distractors datasets under sparse view settings demonstrate that our approach can detect transient objects given sparse input images, outperforming current state-of-the-art methods.

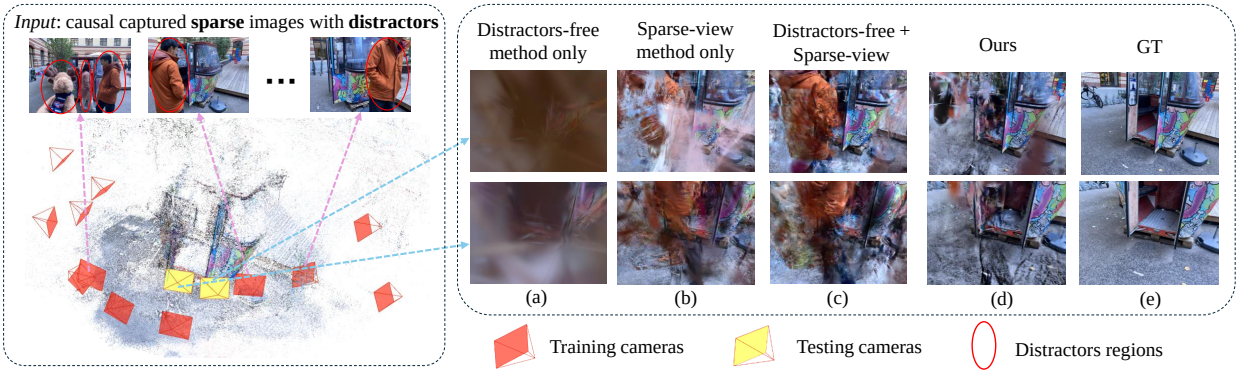

Figure 1. **The overall problem and comparison results.** Given sparse in-the-wild images with distractors, our methods successfully reconstruct static regions whereas existing distractor-free reconstruction method such as RobustSplat (Fu et al., 2025), sparse-view reconstruction method such as NexusGS (Zheng et al., 2025) and their vanilla combination all fail to achieve satisfactory results.

1 INTRODUCTION

The 3D scene reconstruction and novel view synthesis from 2D images is a long standing problem in computer vision which has a wide applications in robotics (Deng et al., 2026), autonomous driving (Wei et al., 2026), AR/VR (Ding et al., 2026), etc. The radiance field methods such as Neural Radiance Field (NeRF) (Mildenhall et al., 2021) and 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) can render photorealistic novel view. In particular, 3DGS has emerged as one of the most popular approaches due to its explicit representation and fast rendering speed.

However, both of 3D representations assume input images are dense and satisfy the multi-view consistency which limits the applicability in practical settings. As shown in Fig. 1, in real world scenarios, in-the-wild images captured from a static scene such as a colorful street kiosk are usually sparse and may contain transient objects like moving people. These transient objects are also commonly referred to as distractors. Reconstruction from these collections will cause the artifacts in the novel view synthesis and thus degrade the quality of 3D scene.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Existing work mainly focus only on either sparse views reconstruction or transient-free reconstruction, but overlook the scenarios that are captured by sparse images and containing transient objects.

For distractor-free reconstruction, the transient objects in training images provide incorrect gradients which harm the optimization of Gaussians. Therefore, most of work (Wang et al., 2026; Fu et al., 2025; Tang et al., 2025a) focus on identifying masks of distractors to block gradient propagation. Memorization effect (Wang et al., 2024b) of radiance field can be effectively used to find transient regions by comparing rendered image and ground truth image. The default initialized point clouds from SfM (Schonberger & Frahm, 2016) can represent simple-to-reconstruct regions with enough input, but they are inadequate under sparse-view settings and thus need many optimization iterations to reconstruct static regions through densification process. Hence, the model misclassify static hard-to-reconstruct regions as transient objects and considers most of regions are distractors, thus fails to reconstruct static scene. As shown in the Fig. 1 (a), the rendered image are burring and cannot reconstruct static scene.

For sparse-view reconstruction, the limited training images usually cause the overfitting problems and wrong 3D geometry. Some work incorporate additional regularization constrains (Li et al., 2024; Zhu et al., 2024). Another series of work augment the initial point clouds of 3DGS by using pretrained mono-depth estimation model (Zheng et al., 2025; Zhang et al., 2025) or pretrained large reconstruction model (Leroy et al., 2024). However, these methods fail when the scene has transient objects. Because without explicitly removing distractors, they will provide wrong gradients for updating parameters of Gaussians and finally generate many floating useless Gaussians to fit distractors in each training image. The Fig. 1 (b) show the floating distractors in rendered image.

A straightforward way is to directly apply techniques from sparse-view reconstruction methods into the pipeline of distractor-free reconstruction. However, this will not work from our experiments. Distractor-free reconstruction needs the memorization effect to detect distractors during the early stage of reconstruction, which cannot be achieved by existing sparse-view reconstruction methods. The main reason is that current sparse-view reconstruction methods simultaneously strengthen static and transient regions. The model cannot distinguish two kinds of regions by comparing rendered image and ground truth image. Consequently, the model considers all regions as static and nearly none of them are identified as transient objects. As shown in Fig. 1 (c), both of transient and static regions are enhanced.

In order to address the problems of transient-free reconstruction with sparse view input, we propose GASplat. We incorporate MGCMT module to enhance initial point clouds of static regions with epipolar geometry and cycle consistency. The pretrained image matching model is used to triangulate point clouds of high confident static regions and create a coarse masks of distractors. In order to fix the problem that pretrained image matching model cannot handle far background well, we cluster the depth of image and initialize some points at these regions to facilitate subsequent optimization process. Finally, with aim of precisely blocking gradient of transient objects, we design a progressive training schedule to refine distractors masks and reconstruct purely static regions. It firstly utilizes the coarse masks to reconstruct static regions, then gradually trains MLP and finally use it to predict accurate masks of distractors.

Our contributions can be summarized as follows:

- We propose a novel geometry-aware point clouds enhancement method called MGCMT to strengthen high confident static regions and create coarse transient masks, and cluster depth values to supplement the background points.
- We propose progressive training schedule to refine the masks and reconstruct static regions with the global modeling of uncertainty and pixel-wise prediction of transient regions.
- We comprehensively analyze the sparse-view and transient-free reconstruction and conduct extensive experiments on the common distractors datasets to show the state-of-the-art performance with sparse view settings.

2 Related Works

2.1 Novel View Synthesis with sparse input

The emergence of radiance field has revolutionized the 3D reconstruction tasks. The types of radiance field includes implicit and explicit representation. The Neural Radiance Field (NeRF) (Mildenhall et al., 2021) use neural networks to implicitly represent 3D scenes but suffer from high computational demand and low rendering speed. The prevailing 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) used in this paper utilizes explicit 3D Gaussian primitives to represent and render 3D scenes which is much faster and can still have photorealistic results.

However, both 3D representations require dense images of the scene for training which is not practical in real life scenario. The sparse input usually causes the overfitting problem and wrong geometry of the scene which severely degrade the scene quality.

In order to solve above problems, some work use different regularization strategies or pretrained model to avoid overfitting phenomenon. DNGaussion (Li et al., 2024), FSGS (Zhu et al., 2024) and CoherentGS (Paliwal et al., 2024) use pretrained mono-depth prediction model to regularize the training process. DropGaussian (Park et al., 2025) randomly dropout some of Gaussians to facilitate the gradient backpropagation and regularize the training. Cor-gs (Zhang et al., 2024b) simultaneously train multiple sets of Gaussians and use co-regularization to prune the Gaussians.

Another line of work choose to enhance the initialization point cloud of 3DGS. It can divided into three main methods. (1) Depth map back-projection. Zheng et al. (2025); Li et al. (2025) uses pretrained model like optical flow model or mono-depth prediction model to refine the depth prediction and back-project depth map to create dense point clouds. (2) Pretrained large reconstruction model. These work (Tang et al., 2025a;b) use pretrained large reconstruction model such as DUSt3R (Wang et al., 2024a), MASt3R (Leroy et al., 2024) or VGGT (Wang et al., 2025) to get dense point clouds. (3) Triangulation based initialization. These work (Dai & Xing, 2025; Kotovenko et al., 2026; Kim et al., 2026) find corresponding points and triangulate to get dense point clouds.

All of these work can work well in sparse view settings, but fail in scenes with transient objects due to without blocking gradient back propagation of distractors.

2.2 Robust 3D reconstruction with transient objects

The vanilla radiance field such as 3DGS assumes the scene is static which fails to reconstruct in-the-wild scenario with transient objects like moving cars or people. The mainstream method is to detect the transient regions and use weighted loss to train the radiance field model.

Existing work can be divided into two paradigms. (1) Uncertainty theory to model transient objects. NeRF-On-the-go (Ren et al., 2024) and WildGaussians (Kulhanek et al., 2024) use MLP to predict uncertainty to suppress influence of high error pixels. (2) Memorization effect (Wang et al., 2024b) of radiance field. It utilizes the fact that radiance field reconstructs static regions first at early stage, so transient regions can be detected by difference between rendered image and ground truth image during the optimization process. One line of work improve the performance by incorporating different priors such as features of UNet (Zhang et al., 2024a), DINO (Kulhanek et al., 2024), diffusion model (Sabour et al., 2025), FiT3D (Wang et al., 2026) or incorporating segmentation model (Tang et al., 2025a) like SAM (Kirillov et al., 2023) to provide accurate object boundary. Another line of work focus on the heuristics strategy. SpotLessSplats (Sabour et al., 2025) trains robust estimator with semantic features. RobustSplat (Fu et al., 2025) uses the delayed densification strategy to suppress distractors fitting. DualSplat (Wang et al., 2026) proposes failure-to-prior paradigm with two-stage reconstruction.

However, these distractor-free methods cannot distinguish static hard-to-reconstructed regions and transient parts which misclassify former one as distractors.

3 Method

3.1 Preliminaries

3D scenes representation The 3D scenes in this paper are presented by 3DGS (Kerbl et al., 2023) which consists of substantial gaussian primitives. The total learnable parameters for i-th gaussian are $\Theta_i = \{\mu_i, s_i, q_i, \alpha_i, c_i\}$, where $\mu_i \in \mathbb{R}^3$ is the center of the gaussian, $s_i \in \mathbb{R}^3$ and $q_i \in \mathbb{R}^4$ are used to establish the scaling matrix S and rotation matrix R, $\alpha_i \in \mathbb{R}$ is opacity value and color value c_i is represented by spherical harmonics coefficients.

For rendering color $C(x_p)$ of one pixel x_p in novel view, 3DGS utilizes splatting based rasterization by ordering N Gaussians that contributes to that pixel and applyingl α-blending technique:

$$C(x_p) = \sum_{i \in N} c_i \alpha_i \mathcal{G}_i^{2D}(x_p) \prod_{j=1}^{i-1} (1 - \alpha_j \mathcal{G}_j^{2D}(x_p)) \tag{1}$$

where projected 2D Gaussian is $\mathcal{G}_i^{2D}(x_p) = e^{-\frac{1}{2}(x_p - \mu')^T \Sigma'^{-1} (x_p - \mu')}$, with 2D center position μ', covariance $\Sigma' = JW\Sigma W^T J^T$, viewing transform matrix W and affine approximation of projective transformation matrix J.

The training objective of 3DGS is the photometric loss combining L_1 loss and SSIM loss:

$$\mathcal{L}_{\text{3DGS}} = \lambda L_1(\hat{I}, I) + (1 - \lambda) L_{\text{SSIM}}(\hat{I}, I) \tag{2}$$

where the I is ground truth image, $\hat{I}$ is rendered image and λ is the controlling weight.

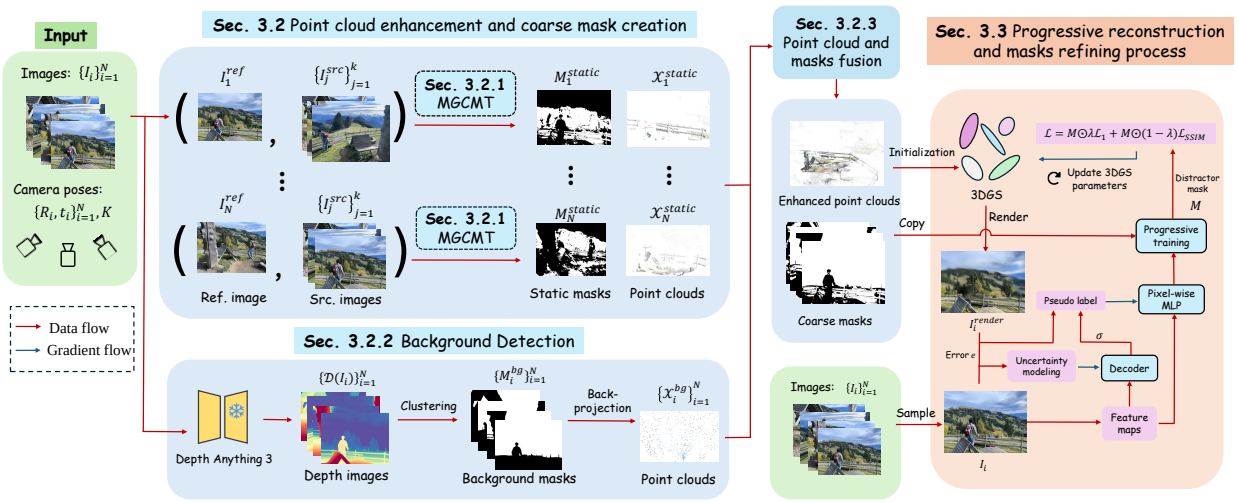

Figure 2. **Overview of proposed method.** Input includes N images and corresponding camera poses. Then, they will be fed into Sec. 3.2.1 and Sec. 3.2.2 in parallel. Next, results will be merged in Sec. 3.2.3. Finally, Sec. 3.3 will optimize parameters of 3DGS and reconstruct static scene.

Transient-free reconstruction In order to alleviate the adverse influence of transient objects in static scenes reconstruction, recent work usually use the masked objective loss function of Eq. 2 to block the gradients of transient regions:

$$\mathcal{L}_{\text{robust}} = M \odot \lambda L_1(\hat{I}, I) + M \odot (1-\lambda) L_{\text{SSIM}}(\hat{I}, I) \tag{3}$$

where M denotes the masks of transient regions and $\odot$ means the element-wise multiplication. The main focus of this paper is how to find transient object masks in the sparse view setting.

3.2 Point Cloud Enhancement and Coarse Mask Creation

3.2.1 Multi-view Geometric Constraints Based Mask Filter and Triangulation

In this section, we propose Multi-view Geometric Constraints based Mask filter and Triangulation (MGCMT) module. Given i-th reference image I_i^{ref} and corresponding nearest k source images $\{I_j^{\text{src}}\}_{j=1}^k$, the goal of this module is to find potential coarse masks of distractor in I_i^{ref} and create dense point clouds of high confident static regions.

Mask creation In order to obtain coarse masks of transient objects for blocking gradients in progressive training module (Sec. 3.3) and regions for point clouds creation, we use epipolar geometry and cycle consistency with pretrained image matching model to filter them out.

Specifically, for each pixel (u, v) in I_i^{ref}, we use pretrained image matching network $\mathcal{W}$ (Edstedt et al., 2025) to predict the matched points $(u', v') = \mathcal{W}_{i\to j}(u, v)$ and corresponding confidence $\mathcal{C}_{i\to j}(u, v)$ in each source image I_j^{src}.

A key characteristic of pixel in static regions is that it corresponds to projection of one static 3D point. Since projections of the same 3D point across different views should satisfy the underlying multi-view geometric constraints, we can determine whether a point is static by examining its matching points. Based on this observation, we select the highest-confidence matched point from nearest k source views candidates in order to reduce the influence of occlusions or low-texture-induced errors, and verify whether it satisfies multi-view geometric constraints. Hence, for each pixel (u, v) in I_i^{ref}, we select the most confident matched pixel (u'_{j^*}, v'_{j^*}) of source image $j^* = \arg\max_j \mathcal{C}_{i\to j}(u, v)$. Then, we use two evaluation criteria to judge whether each pixel in reference image is valid.

From the epipolar geometry, the matched pixel in the reference view should laying in the epipolar line computed from the fundamental matrix $\mathbf{F}_{ij}$ as following:

$$\mathbf{E}_{ij} = [\mathbf{t}_j - \mathbf{R}_j \mathbf{R}_i^\top \mathbf{t}_i]_\times \mathbf{R}_j \mathbf{R}_i^\top \tag{4}$$

$$\mathbf{F}_{ij} = (\mathbf{K}^{-1})^\top \mathbf{E}_{ij} \mathbf{K}^{-1} \tag{5}$$

where $\mathbf{E}_{ij}$ is the essential matrix, $\mathbf{R}_i$, $\mathbf{t}_i$, $\mathbf{R}_j$, $\mathbf{t}_j$ and $\mathbf{K}$ are the camera extrinsic and intrinsic parameters of reference image and source image. $[\cdot]_\times$ denotes the skew-symmetric matrix operator of the cross product.

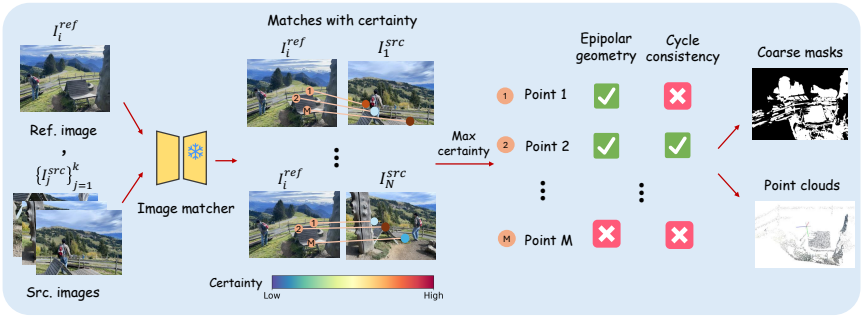

Figure 3. **Multi-view geometric constraints based mask filter and triangulation (MGCMT).**

Intuitively, the distance between the predicted matching point and epipolar line can judge the reliability of pixel of reference view. In order to obtain more precise results, we use Sampson distance (Luong & Faugeras, 1996) to present minimum displacement required to reach correct matched positions:

$$s_i(u,v) = \frac{(\mathbf{p}_i^\top \mathbf{F}_{ij}\mathbf{p}_j)^2}{\|(\mathbf{F}_{ij}\mathbf{p}_i)_{0:1}\|_2^2 + \|(\mathbf{F}_{ij}^\top \mathbf{p}_j)_{0:1}\|_2^2} \tag{6}$$

where $\mathbf{p}_i = (u_i, v_i, 1)$, $\mathbf{p}_j = (u_j, v_j, 1)$ and $(\cdot)_{0:1}$ means taking the first two components of the vector. Then the pixel that above the threshold is marked as the transient object.

Another criteria is the cycle consistency error which use the image matching model to check whether the matched point in the source view can go back to the pixel in the reference view.

$$e_i(u,v) = \|\mathcal{M}_{j\to i}(\mathcal{M}_{i\to j}(u,v)) - (u,v)\|_2 \tag{7}$$

Then the pixel (u, v) in i-th reference image that lower both thresholds is marked as the static object correspond to static masks, otherwise marked as the transient object correspond to transient masks:

$$M_i^{\text{static}} = \{(u,v) | \, (s_i(u,v) \le \tau_s) \wedge (e_i(u,v) \le \tau_e)\} \tag{8}$$

The characteristic of coarse masks is that the regions with value of 0 contain most of distractors although they may inevitably include some false-negative regions which will be corrected in background detection module (Sec. 3.2.2) and progressive training schedule (Sec. 3.3).

Point clouds creation of static regions With aim of enhancing static regions to use memorization effect to distinguish distractor regions and static regions, we triangulate the corresponding points of static regions obtained in the mask creation module.

Given two matched points in homogeneous coordinates $\mathbf{x}_i = (u, v, 1)^\top$, $\mathbf{x}_j = (u', v', 1)^\top$ in the pixel space, we need to find the corresponding 3D point $\mathbf{X}^{\text{static}} = (x, y, z, 1)^\top$ in the world coordinates. Assume the projection matrix of source view and reference view are $\mathbf{P}$, $\mathbf{P}' \in \mathbb{R}^{3\times4}$, so the projected points are $\mathbf{P}\mathbf{X}^{\text{static}}$, $\mathbf{P}'\mathbf{X}^{\text{static}} \in \mathbb{R}^3$ respectively which are proportional to the $\mathbf{x}_i$, $\mathbf{x}_j$. It indicates that the cross product is zero, hence we have:

$$\mathbf{x}_i \times \mathbf{P}\mathbf{X}^{\text{static}} = 0 \tag{9}$$

$$\mathbf{x}_j \times \mathbf{P}'\mathbf{X}^{\text{static}} = 0 \tag{10}$$

Then 3D point $\mathbf{X}^{\text{static}}$ can be obtained by solving optimization problem (Hartley et al., 2003; Kotovenko et al., 2026). For i-th reference image, we concat all the point clouds in static regions $\mathcal{X}_i^{\text{static}} = \{\mathbf{X}_{uv}^{\text{static}} | \, (u,v) \in M_i^{\text{static}}\}$.

3.2.2 Background Detection

The pretrained image matching model usually fail to get correct matching pairs of infinite background such as sky, which will be marked as transient part by the MGCMT module in Sec. 3.2.1. To address this issue, we utilize the pretrained depth predictor (Lin et al., 2026) to detect background regions and create point clouds around it.

The pretrained depth predictor $\mathcal{D}$ generates depth maps of the input training images. Then, we divide depth value of i-th image $\mathcal{D}(I_i)$ into S groups by agglomerative clustering method Paliwal et al. (2024). The pixels of farthest level

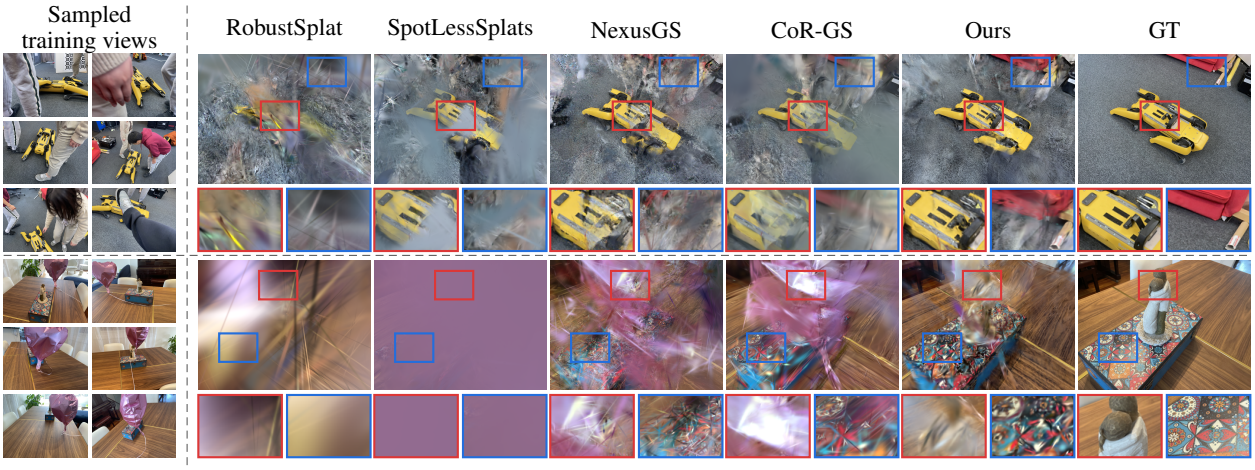

Figure 4. **Qualitative comparisons of different methods.** Left: Sampled training views. Right: Novel view synthesis from testing camera. Top: *Spot* with 12 input views, distractors are moving persons. Bottom: *Statue* with 6 input views, distractor is the dynamic purple balloon.

are marked as the background regions M_i^{bg}. Then the background part will be removed from transient masks to allow the supervision in Sec. 3.3.

For generating point clouds corresponds to M_i^{bg}, we firstly align depth value predicted by pretrained model with existing triangulated 3D points, then randomly back project depth value of these regions to create point clouds. Finally, assuming that pixel (x, y) belongs to background regions, we can create corresponding 3D point as follows:

$$\mathbf{X}^{\text{bg}} = \mathbf{W}_i^{-1}\mathbf{K}_i^{-1}d_i^{\text{bg}}(x, y) \tag{11}$$

where $\mathbf{W}_i$ is the world-to-camera matrix and $\mathbf{K}_i$ is the camera intrinsic parameters of i-th image.

We concat all the point clouds in background regions $\mathcal{X}_i^{\text{bg}} = \{\mathbf{X}_{uv}^{\text{bg}} | \, (u, v) \in M_i^{\text{bg}}\}$. After supplementing point clouds and adjusting masks, we have more precise coarse masks and point clouds in static regions.

3.2.3 Point clouds and masks fusion

The results obtained in Secs. 3.2.1 and 3.2.2 are fused into the enhanced point clouds $\mathcal{X}$ and coarse masks M:

$$\mathcal{X} = \bigcup_{i=1}^{N} \left(\mathcal{X}_i^{\text{static}} \vee \mathcal{X}_i^{\text{bg}}\right), \quad M = \bigcup_{i=1}^{N} \left(M_i^{\text{static}} \vee M_i^{\text{bg}}\right) \tag{12}$$

where N is total numbers of training images, and $\vee$ indicates the or operation.

3.3 Progressive reconstruction and mask refining

In order to reconstruct static regions, we need to precisely prevent gradient propagation of distractors. The mask predicted from the Sec. 3.2 may come from transient objects or some hard-matched static regions. The goal of this section is to refine the mask and reconstruct the static scenes by using neural network and progressive training schedule.

Feature extraction and Uncertainty modeling In order to provide more information for model to judge transient regions, we combine DINOv3 (Siméoni et al., 2025), SAM3 (Carion et al., 2026), per image embedding and 2D position Fourier feature encodings into f_c to enhance semantics, boundary and spatial information of objects.

To enrich training signals, we leverage uncertainty modeling (Ling et al., 2025). Specifically, we train a convolutional neural network as decoder to predict pixel-wise uncertainty value σ. The training objective is to minimize negative log-likelihood: $\mathcal{L}_{\text{uncertainty}} = (c - \hat{c})^2/2\sigma^2 + \log \sigma$, where ground truth pixel color is c, rendered pixel color is $\hat{c}$.

Pseudo-labels based transient masks prediction With aim of refining masks of transient regions and exclude false negative static parts, we use coarse masks and point clouds of high confident static regions as prior knowledge, utilizing MLP to predict distractors based on memorization effect of radiance field. Because there is no ground truth of transient masks, we generate the self-supervised pseudo labels Sabour et al. (2025); Ling et al. (2025) of high likelihood static

Table 1. **Quantitative comparisons on NeRF On-the-go dataset with distractor-free reconstruction methods.**

Methods	PSNR↑					SSIM↑					LPIPS↓				
	6-view	9-view	12-view	24-view	36-view	6-view	9-view	12-view	24-view	36-view	6-view	9-view	12-view	24-view	36-view
3DGS [TOG'23]	12.18	13.20	13.74	14.89	16.13	0.207	0.283	0.331	0.451	0.542	0.593	0.548	0.517	0.445	0.383
WildGS [NeurIPS'24]	11.25	12.37	13.06	15.14	17.35	0.169	0.232	0.284	0.446	0.570	0.658	0.603	0.570	0.476	0.387
RobustSplat [ICCV'25]	6.96	9.47	10.85	16.97	19.75	0.074	0.188	0.247	0.536	0.688	0.754	0.680	0.627	0.394	0.250
SLS [TOG'25]	12.03	13.21	14.04	16.55	18.31	0.181	0.264	0.312	0.481	0.582	0.657	0.596	0.556	0.439	0.347
OCSplats [ICCV'25]	13.35	12.59	13.78	17.59	19.83	0.296	0.260	0.323	0.555	0.672	0.723	0.643	0.575	0.370	0.267
DualSplat [CVPR'26]	11.72	13.10	14.13	16.82	18.75	0.197	0.287	0.350	0.531	0.641	0.565	0.479	0.433	0.310	0.224
Ours	**13.74**	**15.71**	**16.92**	**19.72**	**20.59**	**0.377**	**0.482**	**0.544**	**0.679**	**0.725**	**0.434**	**0.343**	**0.301**	**0.212**	**0.178**

Table 2. **Quantitative comparisons on RobustNeRF dataset with distractor-free reconstruction methods.**

Methods	PSNR↑					SSIM↑					LPIPS↓				
	6-view	9-view	12-view	24-view	36-view	6-view	9-view	12-view	24-view	36-view	6-view	9-view	12-view	24-view	36-view
3DGS [TOG'23]	15.28	17.63	17.46	20.43	21.76	0.489	0.619	0.650	0.765	0.793	0.515	0.404	0.383	0.273	0.247
WildGS [NeurIPS'24]	12.77	16.09	17.17	19.87	22.27	0.372	0.514	0.590	0.740	0.792	0.660	0.561	0.501	0.391	0.334
RobustSplat [ICCV'25]	9.30	15.10	17.12	24.33	26.09	0.215	0.563	0.603	0.832	0.855	0.647	0.496	0.394	0.206	0.181
SLS [TOG'25]	15.65	18.20	20.01	24.29	25.66	0.497	0.623	0.689	0.805	0.832	0.515	0.370	0.296	0.181	0.151
OCSplats [ICCV'25]	15.31	15.93	17.26	23.33	25.63	0.547	0.549	0.615	0.812	0.840	0.591	0.525	0.446	0.237	0.201
DualSplat [CVPR'26]	14.40	18.28	20.39	24.69	26.02	0.460	0.623	0.702	0.825	0.849	0.419	0.255	0.189	**0.096**	**0.076**
Ours	**18.81**	**21.10**	**22.64**	**25.90**	**26.98**	**0.663**	**0.747**	**0.781**	**0.843**	**0.859**	**0.243**	**0.167**	**0.139**	**0.096**	0.086

and transient regions to train the MLP. The main clue we use for i-th image is $R_i = \lambda_e e_i + \lambda_\sigma \sigma_i$ which combines RGB residual error e_i and uncertainty value σ_i histogram, updated by exponential moving average.

Instead of using hand-crafted threshold (Sabour et al., 2025) to get pseudo labels, we utilize Gaussian Mixture Model (GMM) to fit static and transient parts and dynamically adjust the threshold. Specifically, we use $p(x) = \sum_{k=1}^{2} \beta_k \cdot \mathcal{N}(x|\mu_k, \sigma_k^2)$ to fit probability distributions of static and distractors respectively. Then, $M_\text{s} = \mathbb{1}\{\mathbb{1}\{R_i < \mu_1\} \vee (\mathbb{1}\{R_i < \mu_1\} \circledast \mathbf{B} \geq 0.5)\}$ and $M_\text{t} = \mathbb{1}\{\mathbb{1}\{R_i > \mu_2\} \vee (\mathbb{1}\{R_i > \mu_2\} \circledast \mathbf{B} \geq 0.5)\}$ are pseudo labels of high confident static and transient parts with μ_1, σ_1 and μ_2, σ_2 denoting mean and variance of high confident static and transient parts, $\mathbf{B}$ denoting 3×3 box filter. Finally, the pixel-wise MLP $\Phi_\theta(f_c)$ predicts whether each pixel belongs to the transient region. The loss function is:

$$\mathcal{L}_\text{mask} = M_\text{s} \cdot (1 - \Phi_\theta(f_c)) + M_\text{t} \cdot \Phi_\theta(f_c) \tag{13}$$

which indicates that for pixels belonging to high confident static or transient regions, loss function force $\Phi_\theta(f_c)$ to predict 1 or 0 and for low confident pixels the MLP does not learn from them.

Progressive training schedule Unlike previous work (Fu et al., 2025; Wang et al., 2026) use the predefined exponential decay or delaying densification strategy to improve the reconstruction of static regions at early stage, we use the masks obtained by last module and progressive training schedule. Specifically, in the first t_early steps, we use coarse masks from Sec. 3.2.3 to reconstruct high confident static regions without updating weights of masks prediction model. Then from t_early to t_middle steps, we still use precomputed masks but start to train weights of masks prediction model. Lastly, from t_middle to t_final steps, we completely use masks from masks prediction model and update their weights as well.

Compared with existing methods, our progressive training strategy better exploits the prior geometric knowledge in Sec. 3.2.3, ultimately producing more accurate distractor masks and enabling reconstruction of static regions.

4 Experiments

4.1 Setups

Datasets We conduct comprehensive evaluation on two representative datasets for causal capture scenarios including transient objects: NeRF-On-the-go dataset (Ren et al., 2024) and RobustNeRF dataset (Sabour et al., 2023). The NeRF-On-the-go dataset contains five scenes including *corner*, *fountain*, *mountain*, *patio*, *patio-high* and *spot* with varying occlusion levels ($5\% \sim 30\%$). It includes indoor and outdoor scenes with moving transient objects. The RobustNeRF dataset has four scenes including *android*, *crab2*, *statue* and *yoda* with different types of distractors. Both datasets assume that appearance of static objects and environmental illumination have little change.

The original NeRF-On-the-go and RobustNeRF datasets are designed for dense reconstruction, with approximately 147 training images per scene on average. In order to get the sparse sequence of distractors datasets, we follow the methods used in sparse-view reconstruction tasks to equally sample training images according to camera poses. Specifically, we randomly select one training camera view and then use farthest point sampling methods to select remaining views to ensure that it can cover the whole scenes. We sample 6, 9, 12, 24 and 36 views for exhaustive evaluation.

Table 3. **Quantitative comparisons on NeRF On-the-go dataset with sparse-view reconstruction methods.**

Methods	PSNR↑					SSIM↑					LPIPS↓				
	6-view	9-view	12-view	24-view	36-view	6-view	9-view	12-view	24-view	36-view	6-view	9-view	12-view	24-view	36-view
3DGS [TOG'23]	12.18	13.20	13.74	14.89	16.13	0.207	0.283	0.331	0.451	0.542	0.593	0.548	0.517	0.445	0.383
CoR-GS [ECCV'24]	13.07	13.90	14.57	16.38	17.69	0.292	0.354	0.391	0.525	0.606	0.635	0.571	0.534	0.429	0.364
FSGS [ECCV'24]	12.44	13.66	14.48	16.16	17.54	0.236	0.326	0.374	0.493	0.567	0.620	0.564	0.541	0.468	0.412
DropGS [CVPR'25]	12.56	13.65	14.42	16.34	17.62	0.254	0.337	0.385	0.518	0.585	0.602	0.550	0.518	0.436	0.391
NexusGS [CVPR'25]	12.87	13.82	14.51	15.72	16.36	0.240	0.307	0.359	0.461	0.516	0.575	0.526	0.495	0.432	0.394
EDGS [CVPR'26]	12.28	13.13	13.80	15.38	16.59	0.361	0.404	0.432	0.508	0.567	0.598	0.557	0.534	0.463	0.414
Ours	**13.74**	**15.71**	**16.92**	**19.72**	**20.59**	**0.377**	**0.482**	**0.544**	**0.679**	**0.725**	**0.434**	**0.343**	**0.301**	**0.212**	**0.178**

Table 4. **Quantitative comparisons on RobustNeRF dataset with sparse-view reconstruction methods.**

Methods	PSNR↑					SSIM↑					LPIPS↓				
	6-view	9-view	12-view	24-view	36-view	6-view	9-view	12-view	24-view	36-view	6-view	9-view	12-view	24-view	36-view
3DGS [TOG'23]	15.28	17.63	17.46	20.43	21.76	0.489	0.619	0.650	0.765	0.793	0.515	0.404	0.383	0.273	0.247
CoR-GS [ECCV'24]	15.79	18.90	19.51	21.81	23.37	0.576	0.691	0.727	0.799	0.828	0.486	0.365	0.328	0.259	0.224
FSGS [ECCV'24]	16.80	18.55	20.06	22.95	24.32	0.541	0.661	0.723	0.806	0.826	0.496	0.406	0.344	0.270	0.249
DropGS [CVPR'25]	16.30	18.58	19.29	22.27	23.94	0.566	0.674	0.722	0.808	0.834	0.483	0.388	0.344	0.264	0.235
NexusGS [CVPR'25]	15.23	17.69	18.87	20.56	21.66	0.472	0.591	0.647	0.707	0.730	0.525	0.415	0.349	0.295	0.266
EDGS [CVPR'26]	16.21	18.15	18.73	20.47	22.06	0.621	0.688	0.710	0.763	0.781	0.521	0.445	0.413	0.344	0.307
Ours	**18.81**	**21.10**	**22.64**	**25.90**	**26.98**	**0.663**	**0.747**	**0.781**	**0.843**	**0.859**	**0.243**	**0.167**	**0.139**	**0.096**	**0.086**

Baselines As there is no directly related work, we compare GASplat with current state-of-the-art distractors-free reconstruction model and sparse-view reconstruction model. For distractors-free reconstruction methods, we compare DualSplat (Wang et al., 2026), RobustSplat (Fu et al., 2025), OCSplats (Ling et al., 2025), SLS (Sabour et al., 2025) and WildGS (Kulhanek et al., 2024). For sparse-view reconstruction methods, we compare CoR-GS (Zhang et al., 2024b), DropGS (Park et al., 2025), FSGS (Zhu et al., 2024), NexusGS (Zheng et al., 2025) and EDGS (Kotovenko et al., 2026). To ensure a fair comparison, we rerun all experiments using corresponding official public implementation with the same sparse-view settings.

Implementation Details We implement the GASplat in PyTorch based on official 3DGS codebase. For the mask creation stage of MGCMT, the nearest source images of each reference image are determined by camera transformation matrix. The number of source images k is 4, 6, 8, 14 and 20 for 6, 9, 12, 24, 36 views input. For progressive training stage, we set t_{early} to 400, t_{middle} to 900 and t_{final} to 2000. All of experiments are conducted on a single NVIDIA RTX 4090 GPU. More details are listed in the appendix.

4.2 Comparison

Comparison with distractor-free reconstruction methods Quantitative comparisons with distractor-free reconstruction are reported in Tables 1 and 2. We achieve state-of-the-art performance in PSNR, SSIM and LPIPS metrics across all views settings. We also conduct qualitative comparisons in Fig. 4. Existing distractor-free reconstruction methods misclassify static hard-to-reconstruction regions as transient objects thus cannot reconstruct static scene.

Comparison with sparse-view reconstruction methods As shown in Tables 3 and 4, we achieve state-of-the-art performance in PSNR, SSIM and LPIPS metrics across 6-/9-/12-/24-view settings. The qualitative comparisons with sparse-view reconstruction methods are shown in Fig. 4. Although sparse-view reconstruction methods can enhance some of static regions, it cannot filter out transient objects which lead to undesirable artifacts.

Comparison with vanilla combination of distractor-free and sparse-view methods In order to show that the vanilla combination of distractor-free and sparse-view methods cannot work well, we conduct experiments in Table 6. We combine RobustSplat with different plug-and-play modules in sparse-view reconstruction methods such as dropout regularization or point clouds enhancement methods like EDGS (Kotovenko et al., 2026) or NexusGS (Zheng et al., 2025). Results demonstrate that it improves the performance of existing distractor-free reconstruction methods, while still remaining inferior to our method. Fig. 1 shows that it will strengthen both distractors and static objects.

Comparison with initialization point clouds In order to show that existing point clouds enhancement methods are either failing or including distractors, we generate point clouds from SfM, pretrained large reconstruction model and back-projection from depth map. From the point cloud visualizations, we observe that the point clouds from SfM are overly sparse, those generated by MASt3R are dense but contain distracting regions, and those generated by NexusGS suffer from distortions and including transient objects. Although our point clouds are not the densest, they are the cleanest with nearly no distractors.

Method	Geo.	Bg.	Pro.	PSNR↑	SSIM↑	LPIPS↓
w/o **Geo.**	✗	✓	✓	18.53	0.626	0.276
w/o **Bg.**	✓	✗	✓	19.25	0.655	0.255
w/o **Pro.**	✓	✓	✗	19.87	0.657	0.233
Full model	✓	✓	✓	**20.27**	**0.670**	**0.220**

Table 5. **Ablation study of components in GASplat. Geo.** refers to the ablation on geometry-aware filter, **Bg.** indicates the ablation on background detection, and **Pro.** denotes the ablation on progressive training. Results are average values across all datasets and view settings.



w/o **Geo.** w/o **Bg.** w/o **Pro.** **Full model**

Figure 5. **Component ablation visualizations.** Orange regions represent the mask of transient objects. Top: *Patio* with 9 input views. Bottom: *Mountain* with 6 input views.

Method	PSNR↑	SSIM↑	LPIPS↓
RobustSplat	15.59	0.480	0.463
RobustSplat+dropout	16.43	0.528	0.472
RobustSplat+NexusGS	16.83	0.506	0.420
RobustSplat+EDGS	17.12	0.536	0.402
Ours	**20.27**	**0.670**	**0.220**

Table 6. **Quantitative comparisons with vanilla combination of distractor-free and sparse-view methods.**



SfM MASt3R NexusGS Ours

Figure 6. **Point cloud visualizations.** Top: *Corner* with 6 input views. Bottom: *Patio High* with 24 input views.

Complexity analysis The total trainable parameters of neural network part are about 0.54 million$+16V$, where V indicates the number of input views and 16 is the dimension of image embedding. More details on the running time analysis are provided in the appendix.

4.3 Ablation Study

Ablation on geometry-aware filter To verify the importance of geometry-aware filter in MGCMT module, we remove geometry evaluation criteria for point clouds and coarse masks creation for ablation. As shown in Table 5, this module contributes the largest performance gain with 1.74dB in PSNR indicating that the geometry-aware filter is the key to the success. From the visualization in Fig. 5, the model cannot find any distractors without geometry-aware filter.

Ablation on background detection To clarify the importance of background detection module, we don't add points at far background and corresponding masks for ablation. As shown in Table 5, without this module will cause 1.02dB decrease in PSNR. Fig. 5 shows that without adding points at far background and remove corresponding masks, it will misclassify the background regions as transient regions.

Ablation on progressive training With aim of showing effectiveness of progressive training, we remove coarse masks and only utilize the MLP to predict masks. Table 5 show that the absence of this module causes 0.88 dB drop in PSNR. Results in Fig. 5 demonstrate that progressive training produces more precise boundary for transient regions.

5 Conclusion

This paper introduces GASplat, a novel framework for distractor-free reconstruction under sparse-view settings. In order to strengthen static regions distinguishing from transient objects, we use geometry-aware MGCMT module to enhance initial point clouds and generate coarse masks. Then background detection module is used to mitigate limitations of pretrained model. Finally, with aim of precisely blocking gradients of distractors, we use progressive training with MLP to predict transient objects for robust reconstruction. Extensive experiments and ablation study have shown that our method can successfully reconstruct most of static regions comparing with existing methods.

Limitations. The quality of the pretrained models such as pretrained image matching model and extracted pretrained features may have influence on the performance. Another limitation is that we only remove distractors without generating information behind them. In the future work, we may consider using generative model to fix this problem.

AI USE STATEMENT

In this paper, we used generated AI for assisting with translation and implementing some parts of methods. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, clean and reformat dataset, supporting qualitative and thematic data analysis, interpreting results. We have carefully reviewed all AI-generated code and tested by all authors.

REPRODUCIBILITY STATEMENT

We have made best efforts to ensure reproducibility of our work. All datasets used in the paper are public, and dataset processing procedures are clearly described in the paper. To facilitate replication, the comprehensive implementation details are listed in experiment part and the appendix part. The source code will be available upon publication.

REFERENCES

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Dongrui Dai and Yuxiang Xing. Eap-gs: efficient augmentation of pointcloud for 3d gaussian splatting in few-shot scene reconstruction. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16498–16507. IEEE, 2025.

Yinan Deng, Kejia Hu, Ye Chen, Jianyu Dou, Jiahui Wang, Jingyu Zhao, Haojia Ao, Yi Yang, and Yufeng Yue. Video2robo: 3dgs-based synthetic data from one video enables scalable robot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6695–6705, June 2026.

JunRan Ding, WeiHang Liu, YuKe Li, YiZhou Wang, AnTong Li, QiHan Ding, Xin Lou, and JingYi Yu. Sharp-gs: Scalable high-fidelity accelerated rendering pipeline for ultra-high resolution 3d gaussian splatting. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–11, 2026.

Johan Edstedt, David Nordström, Yushan Zhang, Georg Bökman, Jonathan Astermark, Viktor Larsson, Anders Heyden, Fredrik Kahl, Mårten Wadenbäck, and Michael Felsberg. Roma v2: Harder better faster denser feature matching. *arXiv preprint arXiv:2511.15706*, 2025.

Chuanyu Fu, Yuqi Zhang, Kunbin Yao, Guanying Chen, Yuan Xiong, Chuan Huang, Shuguang Cui, and Xiaochun Cao. Robustsplat: Decoupling densification and dynamics for transient-free 3dgs. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27126–27136. IEEE, 2025.

Richard Hartley, Andrew Zisserman, et al. *Multiple view geometry in computer vision*, volume 2. Cambridge university press Cambridge, 2003.

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

Hyeseong Kim, Geonhui Son, Deukhee Lee, and Dosik Hwang. Twings: Thin plate splines warp-aligned initialization for sparse-view gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26062–26071, June 2026.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

Dmytro Kotovenko, Olga Grebenkova, and Björn Ommer. Edgs: Eliminating densification for efficient convergence of 3dgs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41065–41076, 2026.

Jonas Kulhanek, Songyou Peng, Zuzana Kukelova, Marc Pollefeys, and Torsten Sattler. Wildgaussians: 3d gaussian splatting in the wild. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 21271–21288. Curran Associates, Inc., 2024. doi: 10.52202/079017-0670. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/25c0fe7b157821dd3140727dc07461da-Paper-Conference.pdf.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pp. 71–91. Springer, 2024.

Deming Li, Kaiwen Jiang, Yutao Tang, Ravi Ramamoorthi, Rama Chellappa, and Cheng Peng. Ms-gs: Multi-appearance sparse-view 3d gaussian splatting in the wild. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Jiahe Li, Jiawei Zhang, Xiao Bai, Jin Zheng, Xin Ning, Jun Zhou, and Lin Gu. Dngaussian: Optimizing sparse-view 3d gaussian radiance fields with global-local depth normalization. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20775–20785. IEEE, 2024.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 141261–141285, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/e4cd50120b6d7e8daff1749d6bbaa889-Paper-Conference.pdf.

Han Ling, Xian Xu, Yinghui Sun, and Quansen Sun. Ocsplats: Observation completeness quantification and label noise separation in 3dgs. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 25680–25689. IEEE, 2025.

Quan-Tuan Luong and Olivier D Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *International journal of computer vision*, 17(1):43–75, 1996.

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

Avinash Paliwal, Wei Ye, Jinhui Xiong, Dmytro Kotovenko, Rakesh Ranjan, Vikas Chandra, and Nima Khademi Kalantari. Coherentgs: Sparse novel view synthesis with coherent 3d gaussians. In *European Conference on Computer Vision*, pp. 19–37. Springer, 2024.

Hyunwoo Park, Gun Ryu, and Wonjun Kim. Dropgaussian: Structural regularization for sparse-view gaussian splatting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21600–21609. IEEE, 2025.

Weining Ren, Zihan Zhu, Boyang Sun, Jiaqi Chen, Marc Pollefeys, and Songyou Peng. Nerf on-the-go: Exploiting uncertainty for distractor-free nerfs in the wild. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8931–8940. IEEE, 2024.

Sara Sabour, Suhani Vora, Daniel Duckworth, Ivan Krasin, David J Fleet, and Andrea Tagliasacchi. Robustnerf: Ignoring distractors with robust losses. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20626–20636. IEEE, 2023.

Sara Sabour, Lily Goli, George Kopanas, Mark Matthews, Dmitry Lagun, Leonidas Guibas, Alec Jacobson, David Fleet, and Andrea Tagliasacchi. Spotlesssplats: Ignoring distractors in 3d gaussian splatting. *ACM Transactions on Graphics*, 44(2):1–11, 2025.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113, 2016.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Jiadong Tang, Yu Gao, Dianyi Yang, Liqi Yan, Yufeng Yue, and Yi Yang. Dronesplat: 3d gaussian splatting for robust 3d reconstruction from in-the-wild drone imagery. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 833–843. IEEE, 2025a.

Yutao Tang, Yuxiang Guo, Deming Li, and Cheng Peng. Spars3r: Semantic prior alignment and regularization for sparse 3d reconstruction. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26810–26821. IEEE, 2025b.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025.

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

Xu Wang, Zhiru Wang, Shiyun Xie, Chengwei Pan, and Yisong Chen. Dualsplat: Robust 3d gaussian splatting via pseudo-mask bootstrapping from reconstruction failures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4912–4921, June 2026.

Yukun Wang, Kunhong Li, Minglin Chen, Longguang Wang, Shunbo Zhou, Kaiwen Xue, and Yulan Guo. Distractor-free novel view synthesis via exploiting memorization effect in optimization. In *European Conference on Computer Vision*, pp. 477–493. Springer, 2024b.

Xiaobao Wei, Zhangjie Ye, Yuxiang Gu, Zunjie Zhu, Yunfei Guo, Yingying Shen, Shan Zhao, Ming Lu, Haiyang Sun, Bing Wang, Guang Chen, Rongfeng Lu, and Hangjun Ye. Parkgaussian: Surround-view 3d gaussian splatting for autonomous parking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19085–19095, June 2026.

Dongbin Zhang, Chuming Wang, Weitao Wang, Peihao Li, Minghan Qin, and Haoqian Wang. Gaussian in the wild: 3d gaussian splatting for unconstrained image collections. In *European Conference on Computer Vision*, pp. 341–359. Springer, 2024a.

Jiawei Zhang, Jiahe Li, Xiaohan Yu, Lei Huang, Lin Gu, Jin Zheng, and Xiao Bai. Cor-gs: sparse-view 3d gaussian splatting via co-regularization. In *European conference on computer vision*, pp. 335–352. Springer, 2024b.

Wenyuan Zhang, Yixiao Yang, Han Huang, Liang Han, Kanle Shi, Yu-Shen Liu, and Zhizhong Han. Monoinstance: Enhancing monocular priors via multi-view instance alignment for neural rendering and reconstruction. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21642–21653. IEEE, 2025.

Yulong Zheng, Zicheng Jiang, Shengfeng He, Yandu Sun, Junyu Dong, Huaidong Zhang, and Yong Du. Nexusgs: Sparse view synthesis with epipolar depth priors in 3d gaussian splatting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26800–26809. IEEE, 2025.

Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *European conference on computer vision*, pp. 145–163. Springer, 2024.

APPENDIX

A IMPLEMENTATION DETAILS

A.1 MULTI-VIEW GEOMETRIC CONSTRAINTS BASED MASK FILTER AND TRIANGULATION

After simplifying the equations Eqs. 9 and 10 and remove the redundant parts, we can get:

$$\mathbf{A}\mathbf{X}^{\text{static}} = 0, \quad \mathbf{A} = \begin{bmatrix} v\mathbf{P}_{\text{row},3}^{\top} - \mathbf{P}_{\text{row},2}^{\top} \\ \mathbf{P}_{\text{row},1}^{\top} - u\mathbf{P}_{\text{row},3}^{\top} \\ v'\mathbf{P}_{\text{row},3}'^{\top} - \mathbf{P}_{\text{row},2}'^{\top} \\ \mathbf{P}_{\text{row},1}'^{\top} - u'\mathbf{P}_{\text{row},3}'^{\top} \end{bmatrix} \tag{14}$$

$$\mathbf{X}^{\text{static}} := \arg\min_{\mathbf{X}^{\text{static}}} \|\mathbf{A}\mathbf{X}^{\text{static}}\|_2^2, \quad \mathbf{X}_{\text{row},4}^{\text{static}} = 1 \tag{15}$$

Then the 3D point $\mathbf{X}^{\text{static}}$ can be obtained by using optimization theory.

The threshold τ_s and τ_e are determined by the median of error minus the median of the absolute deviation of the error from the median which is less sensitive to the outliers.

For the point cloud creation stage of MGCMT, we use fixed numbers of point clouds for each sparse-view settings. For each reference image we allocate the point clouds according to the proportion of reliable pixels.

A.2 BACKGROUND DETECTION

We align the depth Kerbl et al. (2023) predicted by the pretrained depth estimation model with existing 3D points by regressing scale s^* and offset t^* in reference image i:

$$s_i^{\text{tri}} = \frac{1}{V}\sum \left| d_i^{\text{tri}}(u,v) - m_i^{\text{tri}} \right|, \quad s_i^{\text{bg}} = \frac{1}{V}\sum \left| d_i^{\text{bg}}(u,v) - m_i^{\text{bg}} \right| \tag{16}$$

$$s_i^* = \frac{s_i^{\text{tri}}}{s_i^{\text{bg}}}, \quad t_i^* = m_i^{\text{tri}} - s_i^* \cdot m_i^{\text{bg}} \tag{17}$$

where pixel $(u,v) \in M_i^{\text{static}}$, V is the total number of valid static points, d is the inverse of depth, m is the middle number of inverse of depth sequence.

For background detection module, we randomly sample points in farthest depth regions and use fixed point clouds number for each reference image.

A.3 PROGRESSIVE RECONSTRUCTION AND MASK REFINING

In order to provide more information for model to judge transient regions, we use DINOv3 to get global semantics of objects f_d and SAM3 to get more detailed object boundary f_s. We further use per image embedding f_i and 2D position Fourier feature encodings f_p to let model have more spatial information of image and scenes.

For the uncertainty modeling, we project DINO feature to 32 dimensions, SAM feature to 32 dimensions to align the dimension of learnable image embedding and positional embedding. Then they are concatenated into CNN to get uncertainty of each pixel for one forward process.

For the transient detection model, only difference is that we use hierarchical features of SAM model and replace CNN with MLP to predict the pixel-wise probability. All training images are downsampled by 8.

The main clues to generate pseudo labels for i-th image are RGB residual error e_i and uncertainty value σ_i which combined into $R_i = \lambda_{\text{e}} e_i + \lambda_\sigma \sigma_i$. We firstly flatten R_i and divide it into k buckets to get the normalized histogram H, then use the exponential moving average to get updated histogram $\hat{H}$ every iteration t: $\hat{H}_i^t = \lambda_h H_i^t + (1-\lambda_h)\hat{H}_i^{t-1}$.

$$\hat{H}_i^t = \lambda_h H_i^t + (1-\lambda_h)\hat{H}_i^{t-1} \tag{18}$$

where λ_{e} is 0.4, λ_σ is 0.3, λ_h is 0.95.

Stage	Component	Time (s)
Feature extraction	Depth features	16.45
	DINO features	5.25
	SAM features	10.85
Sec. 3.2.1	Mask creation	11.69
	Point cloud creation	0.63
Sec. 3.2.2	Background Detection	1.68
Sec. 3.3	Progressive reconstruction and mask refining	59.15
Total	-	**77.28**

Table 7. **Runtime analysis.** Tested in *patio high* with 12 views input.

B RUNNING TIME ANALYSIS

As shown in Table 7, the Sec. 3.3 spends most of time which is the core reconstruction process. The feature extraction process and Sec. 3.2.1 can be done simultaneously. Afterward, we proceed with Sec. 3.2.2 and Sec. 3.3 procedures.