

InstantSplat++: Registration-Free Gaussian Splatting from Sparse Image Sequences

Gonzalo Lope Carrasco, Björn Kremser, Moritz Pötzsch, Osama Salem
Technical University of Munich
Munich, Germany



Figure 1. **Overview of InstantSplat++.** Starting from an unposed image sequence (left), we employ InstantSplat++ to reconstruct the scene via Gaussian splats (center). A subsequent rasterization process then generates novel view synthesis (NVS), depth maps, and surface normals (right).

Abstract

Novel View Synthesis (NVS) from sparse inputs often relies on Structure-from-Motion (SfM) to estimate camera parameters; however, SfM is computationally expensive and can be unreliable with limited views. While InstantSplat improves efficiency using point-based representations combined with Multi-View Stereo (MVS), it still depends on pairwise frame predictions, global alignment, and grid-based sampling, leading to overparameterization.

*We introduce **InstantSplat++**, a faster and more robust approach that eliminates pairwise alignment by leveraging Spann3r to directly predict a point cloud in the first frame’s coordinate space, accommodating arbitrary input frame counts. **InstantSplat++** replaces grid-based sampling with a more efficient cumulative masking technique and supports 2D Gaussian splatting. Consequently, **InstantSplat++** achieves significantly accelerated training while maintaining high-quality NVS performance across diverse datasets.*

1. Introduction

3D reconstruction from sparse image sets is a fundamental challenge in computer vision. Traditional methods often

rely on modular pipelines, including Structure-from-Motion (SfM) for camera pose estimation [13]. However, SfM is computationally demanding and prone to errors with limited views, creating significant bottlenecks. Recent methods such as *InstantSplat* [3] improve efficiency through Multi-View Stereo (MVS) and point-based representations, yet they still require pairwise frame alignment and grid-based sampling, leading to overparameterization and lengthy optimization.

We introduce **InstantSplat++**, a faster and more scalable approach that eliminates pairwise frame predictions and global alignment. Our method leverages *Spann3r* [14] to directly predict a point cloud in the coordinate space of the first frame, enabling seamless handling of arbitrary input frame counts. Instead of grid-based sampling, we propose **cumulative masking**, a novel technique utilizing frustum overlap and depth consistency to efficiently filter redundant points while preserving structural details. Furthermore, our framework supports both 3D Gaussian Splatting (3D-GS) [8] and 2D Gaussian Splatting (2D-GS) [7], enhancing adaptability across neural rendering techniques.

We validate **InstantSplat++** through comprehensive evaluation on the ScanNet++ dataset [17], demonstrating state-of-the-art efficiency and reconstruction quality.

Our main contributions are:

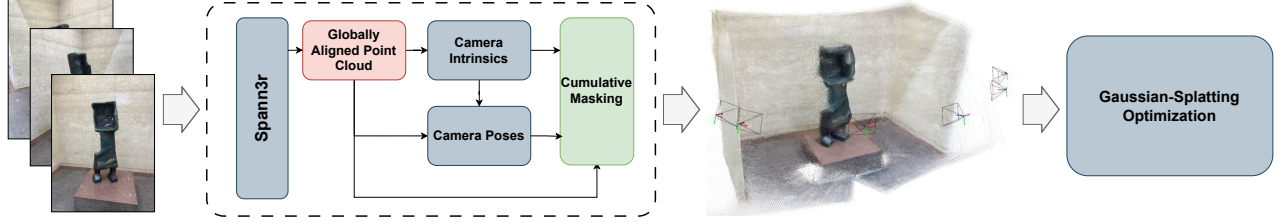


Figure 2. **Pipeline Overview.** Given a sparse set of input images, *Spann3r* predicts a globally aligned point cloud in a single step, along with camera intrinsics and poses. Cumulative masking eliminates redundant points based on frustum overlap and depth consistency. The processed point cloud and camera parameters are then fed into the Gaussian Splatting optimization, producing a high-fidelity 3D reconstruction.

- Direct point cloud prediction using *Spann3r*, eliminating pairwise registration and global alignment.
- A novel **cumulative masking** technique based on frustum overlap and depth consistency, improving efficiency and representation quality.
- A unified framework supporting both 2D-GS and 3D-GS, enhancing adaptability across neural rendering techniques.
- Extensive evaluation on ScanNet++, demonstrating superior performance in speed and reconstruction accuracy.

2. Related Work

3D Representations and Gaussian Splatting Novel View Synthesis (NVS) aims to generate unseen views from sparse images. While *NeRF* [10] pioneered MLP-based 3D representations, its training is slow. More efficient point-based methods, such as **3D Gaussian Splatting (3D-GS)**, use differentiable splatting but typically rely on Structure-from-Motion (SfM) for camera parameters and initial sparse point clouds. However, SfM is computationally expensive, error-prone, and struggles in sparse-view scenarios, leading to degraded reconstruction quality.

InstantSplat improves efficiency by leveraging Multi-View Stereo (MVS) to generate dense surface points and jointly optimize scene parameters. However, it still depends on pairwise frame predictions, global alignment, and grid-based sampling, resulting in redundant Gaussians and slower optimization.

Sparse-View and Pose-Free NVS *NeRF* and **3D-GS** often require dense multi-view images and SfM preprocessing, which is computationally expensive and unreliable with sparse inputs. Some methods, including *NeRFmm* [16] and *BARF* [9], jointly optimize camera poses, while others (e.g., *Depth-NeRF* [2] and *RegNeRF* [11]) use depth priors to regularize geometry. More recent approaches, such as *CF-3DGS* [5] and *Nope-NeRF* [1], attempt pose-free optimization but remain slow, often requiring hours per scene.

InstantSplat++ builds upon *InstantSplat* but eliminates

pairwise frame alignment and global registration. It leverages *Spann3r* for single-step, structured point cloud prediction in the first frame’s coordinate system. Additionally, we introduce **cumulative masking**, which refines scene representations using frustum overlap and depth consistency, eliminating the need for grid-based heuristics. Unlike *InstantSplat*, **InstantSplat++** supports both **3D-GS** and **2D Gaussian Splatting (2D-GS)**, enhancing adaptability. These contributions result in a faster, more scalable, and more robust approach, significantly reducing training time while maintaining high-quality reconstructions.

3. Method

Overview **InstantSplat++** accelerates point-based scene representation initialization (both 3D and 2D Gaussians) for Novel View Synthesis (NVS) from sparse views. We bypass the limitations of Structure-from-Motion (SfM) and *InstantSplat*’s pairwise processing by leveraging *Spann3r* for direct, multi-view point cloud prediction. This is followed by a novel **cumulative masking** technique for efficient point selection and a joint optimization phase.

As shown in Figure 2, our pipeline consists of three key stages:

1. **Direct multi-view initialization:** *Spann3r*-based point cloud and initial camera parameter estimation (§3.1).
2. **Cumulative masking:** Point cloud refinement using frustum overlap and depth consistency (§3.2).
3. **Joint optimization:** Final optimization of Gaussian and camera parameters (§3.3).

3.1. Direct Multi-View Initialization

Instead of relying on pairwise stereo matching and global alignment, we use *Spann3r*, a multi-view consistent geometry estimation network. As illustrated in Figure 3, while *DUST3R* requires local reconstructions followed by global alignment, *Spann3r* directly predicts a globally consistent point cloud $\mathbf{P}_1 \in \mathbb{R}^{H \times W \times 3}$ in the coordinate frame of the first input view (I_1). This eliminates the scale-ambiguous pairwise processing and global alignment required by previous methods.

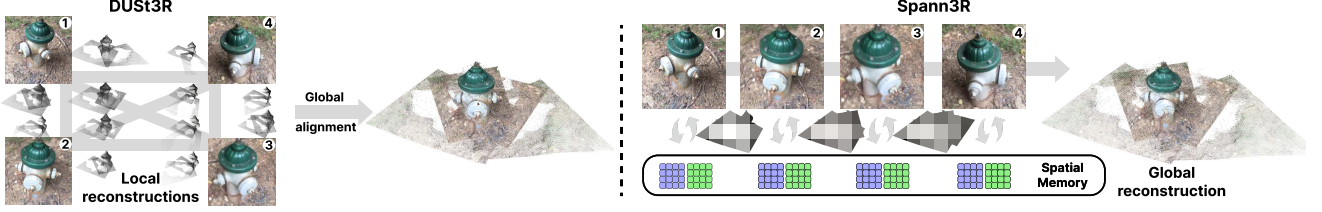


Figure 3. **Point cloud initialization** DUST3R (left) generates point clouds for each image pair, necessitating a computationally expensive global alignment step. InstantSplat++ (right) uses Spann3r for direct multi-view point cloud prediction utilizing an internal memory module.



Figure 4. **Cumulative masking.** Top: Input images. Bottom: Corresponding cumulative masks generated by our method. Black pixels indicate masked regions (points not selected).

Intrinsic Parameter Estimation Given the *Spann3r*-predicted point cloud $\mathbf{P}_1$, where the z -coordinate represents depth relative to the first camera, we estimate the initial intrinsic camera parameters. Assuming a shared focal length f across all views (similar to COLMAP’s [12] single-camera assumption), we use the Weiszfeld algorithm

$$f^* = \arg \min_f \sum_{i=1}^W \sum_{j=1}^H \left\| (i', j') - f \frac{(\mathbf{P}_1^{i,j,1}, \mathbf{P}_1^{i,j,2})}{\mathbf{P}_1^{i,j,3}} \right\|_2^2 \quad (1)$$

where $i' = i - \frac{W}{2}$ and $j' = j - \frac{H}{2}$ are centered pixel indices.

Initial Pose Estimation For each additional view I_n ($n = 2, \dots, N$), we estimate its initial pose $\mathbf{T}_n = [\mathbf{R}_n | \mathbf{t}_n]$ relative to the first view. We use the PnP algorithm [6] within a RANSAC framework [4]. 2D-3D correspondences are established between 2D pixel locations (i, j) in I_n and the corresponding 3D points in $\mathbf{P}_1$.

3.2. Cumulative Masking for Point Cloud Refinement

To address redundancy in dense point clouds, we introduce **cumulative masking**, which efficiently selects a representative subset of points while ensuring coverage across all

views. Unlike grid-based sampling, cumulative masking considers visibility and depth consistency, reducing redundant points while preserving geometric structure.

Let $\mathbf{M}_n \in \{0, 1\}^{H \times W}$ be a binary mask for view I_n , where 1 indicates a selected point and 0 a discarded point.

1. **Initialization:** For the first view ($n = 1$), all points are initially selected:

$$\mathbf{M}_1^{i,j} = 1, \quad \forall i, j. \quad (2)$$

2. **Cumulative Masking (for $n > 1$):** For each subsequent view I_n :

- **Projected Points:** Project $\mathbf{P}_1$ into the image plane of I_n using $\mathbf{T}_n$ and f^* :

$$\mathbf{p}_n^{i,j} = \text{project}(\mathbf{P}_1^{i,j}; \mathbf{T}_n, f^*), \quad (3)$$

where $\mathbf{p}_n^{i,j} = (u, v, d)$ are the projected pixel coordinates and depth.

- **Frustum Overlap and Depth Check:** Each point (i, j) in I_n is selected if its projected coordinates (u, v) are within bounds and its depth d is within a threshold τ_d of the depth predicted by *Spann3r*.
- **Cumulative Mask Update:**

$$\mathbf{M}_n^{i,j} = \begin{cases} 1, & \text{if } (u, v) \text{ is within bounds and} \\ & |d - \mathbf{p}_n^{u,v,3}| < \tau_d \text{ and} \\ & \exists k < n : \mathbf{M}_k^{u,v} = 0. \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

3. **Final Point Cloud:** The refined point cloud $\mathbf{P}_{\text{refined}}$ consists of points in $\mathbf{P}_1$ for which $\exists n : \mathbf{M}_n^{i,j} = 1$.

3.3. Joint Optimization

After initialization and refinement, we jointly optimize the Gaussian parameters (position $\mathbf{x}$, SH coefficients $\mathbf{c}$, opacity α , rotation $\mathbf{q}$, and scaling $\mathbf{s}$ for 3D Gaussians, or position, color, and opacity for 2D Gaussians) and camera poses $\mathbf{T}_n$. The optimization minimizes the photometric loss:

$$\mathbf{G}^*, \mathbf{T}^* = \arg \min_{\mathbf{G}, \mathbf{T}} \sum_{v \in N} \sum_{i=1}^{HW} \left\| \tilde{\mathbf{C}}_v^i(\mathbf{G}, \mathbf{T}) - \mathbf{C}_v^i \right\|_2^2, \quad (5)$$

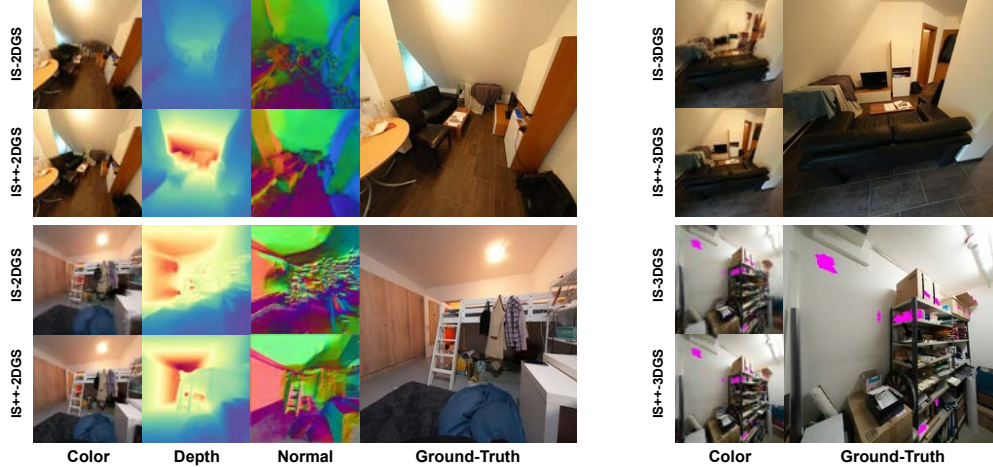


Figure 5. **Visual Comparisons between InstantSplat and InstantSplat++ under 9-view setting.** We present the rendering results of RGB images, depth maps, and normal maps from the Scannet++ dataset.

where G represents the Gaussians (2D or 3D) and T the camera poses.

4. Evaluation

We evaluate InstantSplat++ on the ScanNet++ dataset [17], demonstrating its speed and rendering quality compared to InstantSplat [3]. We also analyze the impact of our cumulative masking technique. Experiments are conducted with both 2D and 3D Gaussian representations.

4.1. Experimental Setup

Dataset We use five scenes from ScanNet++, each with two randomly selected sequences. For each sequence, training sets are sampled with 3, 6, 9, 12, and 15 views. Test views are sampled at an interval of 3, ensuring approximately two test views between each pair of training views. This setup allows us to evaluate performance across varying levels of sparsity.

Implementation Details For both InstantSplat++ and InstantSplat, we initialize the Gaussian parameters (2D or 3D) as described in their respective methods. All Gaussian optimization runs are performed for a fixed 1000 iterations, using identical hyperparameters and training schedules for both methods and both Gaussian representations. This ensures a fair comparison focused on the impact of initialization and masking strategies. The *Spann3r* model used is the pretrained version provided by the authors.

Metrics We evaluate rendering quality using standard metrics:

- **Mean Squared Error (MSE)**

- **Learned Perceptual Image Patch Similarity (LPIPS)** [18]

- **Structural Similarity Index Measure (SSIM)** [15]

For 2D Gaussian experiments, we additionally report depth error metrics:

- τ (**Threshold Accuracy**)
- **Relative Error (REL ERR)**

These depth metrics provide insights into the geometric accuracy of the 2D representations.

4.2. Quantitative Results

Comparison with InstantSplat Tables 1 and 2 present quantitative results for 2D and 3D Gaussian splatting, comparing InstantSplat++ and InstantSplat across different numbers of training views.

InstantSplat++ consistently outperforms InstantSplat in MSE, LPIPS, and SSIM, demonstrating improved rendering quality. Furthermore, for 2D Gaussians, InstantSplat++ achieves lower τ and REL ERR values, indicating better depth accuracy. These results highlight the effectiveness of our *Spann3r*-based initialization and cumulative masking in providing a better starting point for optimization.

4.3. Ablation Study: Cumulative Masking

To validate our cumulative masking technique, we conduct an ablation study comparing it against an alternative strategy: *no masking* (using all points from *Spann3r*). Table 2 presents the results for both 2D and 3D Gaussians.

While using *no masking* slightly improves certain metrics, it does so at the cost of significantly increased parameter count (up to 43% more). Cumulative masking achieves a better balance between rendering quality and efficiency, providing comparable or slightly better results while using

Table 1. Quantitative comparison of InstantSplat++ and InstantSplat on ScanNet++ using both 2D and 3D Gaussian Splatting. All models are trained for 1000 iterations.

Method	Views	2D Gaussian Splatting					3D Gaussian Splatting			Time (s)	
		SSIM $\uparrow$	PSNR $\uparrow$	LPIPS $\downarrow$	Rel Err $\downarrow$	τ $\uparrow$	SSIM $\uparrow$	PSNR $\uparrow$	LPIPS $\downarrow$	Init. $\downarrow$	Total $\downarrow$
InstantSplat	3	0.99	39.37	0.02	8.60	30.03	0.99	43.09	0.01	2.87	12.00
	6	0.96	32.59	0.06	12.23	24.62	0.98	35.15	0.04	4.79	20.00
	9	0.91	27.26	0.14	18.89	14.29	0.95	30.70	0.08	8.61	26.00
	12	0.92	29.30	0.12	13.68	27.26	0.94	31.18	0.09	13.50	31.00
	15	0.88	27.19	0.19	19.31	19.24	0.91	29.31	0.13	21.12	52.00
InstantSplat++	3	0.99	43.39	0.01	5.59	53.33	1.00	47.19	0.00	0.78	9.00
	6	0.98	39.26	0.02	8.29	43.03	0.99	42.10	0.01	1.15	13.00
	9	0.95	33.65	0.07	14.57	33.34	0.98	38.07	0.03	1.56	18.00
	12	0.96	34.68	0.07	11.11	48.91	0.98	37.39	0.04	2.09	21.00
	15	0.91	31.67	0.15	13.06	46.26	0.95	34.94	0.09	2.15	24.00

Table 2. Ablation study of cumulative masking on ScanNet++. Results are shown for both 2D and 3D Gaussian Splatting, with different masking strategies. All models are trained for 1000 iterations with 3 views.

Masking Strategy	Ratio $\downarrow$	2D Gaussian Splatting					3D Gaussian Splatting		
		SSIM $\uparrow$	PSNR $\uparrow$	LPIPS $\downarrow$	Rel Err $\downarrow$	τ $\uparrow$	SSIM $\uparrow$	PSNR $\uparrow$	LPIPS $\downarrow$
No Mask	1.0000	0.9941	43.9267	0.0064	5.3459	54.0252	0.9975	48.2694	0.0035
Only Frustum Overlapping	0.5739	0.9912	41.9371	0.0090	5.1278	56.8799	0.9951	44.9604	0.0057
Cumulative Mask (5e-1 Depth Matching)	0.5752	0.9912	42.0461	0.0089	5.3525	55.1854	0.9951	45.1350	0.0054
Cumulative Mask (5e-2 Depth Matching)	0.5935	0.9915	42.3750	0.0088	5.6609	52.8296	0.9954	45.5554	0.0050
Cumulative Mask (5e-3 Depth Matching)	0.7497	0.9929	43.4089	0.0077	5.6639	53.1233	0.9967	47.2167	0.0040

substantially fewer parameters. This leads to faster optimization, demonstrating its ability to select a representative and geometrically consistent set of points.

5. Conclusion

We introduced **InstantSplat++**, a novel framework designed for ultra-fast initialization of point-based scene representations, including both 2D and 3D Gaussians, for novel view synthesis. By integrating *Spann3r* and a new cumulative masking technique, InstantSplat++ overcomes the constraints of *Structure-from-Motion (SfM)* and eliminates the computational inefficiencies of pairwise processing. Our experiments highlight substantial improvements in initialization speed, reducing it to mere fractions of a second while preserving or even enhancing rendering quality compared to *InstantSplat*, across diverse view configurations and Gaussian representations. Future directions include investigating adaptive sampling strategies and extending the framework to dynamic scenes.

References

- [1] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior, 2023. 2
- [2] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ra-

manan. Depth-supervised nerf: Fewer views and faster training for free, 2024. 2

- [3] Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, Zhangyang Wang, and Yue Wang. Instantsplat: Sparse-view sfm-free gaussian splatting in seconds, 2024. 1, 4
- [4] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 3
- [5] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros, and Xiaolong Wang. Colmap-free 3d gaussian splatting, 2024. 2
- [6] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 3
- [7] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH*, pages 1–11, 2024. 1
- [8] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ToG*, 2023. 1
- [9] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021. 2

- [10] 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. In *ECCV*, 2020. [2](#)
- [11] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs, 2021. [2](#)
- [12] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [3](#)
- [13] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016. [1](#)
- [14] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory, 2024. [1](#)
- [15] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. [4](#)
- [16] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf-: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021. [2](#)
- [17] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes, 2023. [1](#), [4](#)
- [18] 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. [4](#)