AVSplat: Dense-View Feed-Forward 3D Gaussian Splatting with Assist-View Preconditioning

Muyu Xu[1], Fangneng Zhan[2], Yu Wei[1], Hanspeter Pfister[2], and Shijian Lu[1]

[1] Nanyang Technological University, Singapore 639798, Singapore
{muyu001, ywei012}@e.ntu.edu.sg
Shijian.Lu@ntu.edu.sg
[2] Harvard University, MA 02138, USA
{fnzhan, pfister}@seas.harvard.edu

Abstract. Pose-free feed-forward 3D Gaussian Splatting enables novel view synthesis from uncalibrated multi-view images. Although more views should improve performance, existing methods often degrade with dense-view inputs because global aggregation spreads attention over many tokens, and naive voxel fusion averages many Gaussians into overly smooth representations. We present AVSplat, a framework that turns additional views into reliable signals for both aggregation and representation. Before global attention, each view performs a single lightweight interaction with a small set of Assist Views chosen for relevance and diversity, and the cached features provide a focused scene context that stabilizes correspondence. For representation, we use adaptive temperature-aware voxel fusion that sharpens attribution under high occupancy, guided by occupancy and point confidence. Crucially, AVSplat restores positive view scaling where performance remains stable or improves as more input views are added, instead of degrading in the dense-view regime. Ablations show that Assist View Preconditioning is primarily responsible for preventing dense-view degradation, while Occupancy-guided Voxel Fusion contributes most of the single-point image-quality gains.

Keywords: 3D Gaussian Splatting · Novel View Synthesis · Pose-Free Reconstruction · Dense-View Reconstruction

1 Introduction

Recent feed-forward 3D Gaussian Splatting (3DGS) [5,6,12,22,26] enables pose-free novel view synthesis from uncalibrated multi-view images. A single forward pass predicts camera intrinsics and extrinsics together with a compact set of Gaussian primitives for differentiable rasterization. These advances are based on geometric foundation models that infer structure directly from images. DUSt3R [15] and MASt3R [8] estimate dense point maps and pixel correspondences so downstream reconstruction can use point-level supervision without precomputed calibration. VGGT [14] estimates camera parameters, point maps, depth maps, and point tracks from one or many views and simplifies multi-view

reasoning. Feed-forward 3DGS systems adopt such backbones and encode images with a geometry transformer before decoding features into Gaussian parameters and camera poses. This unifies geometric inference and radiance representation within a single pass.

Despite this progress, scaling from a few input views toward dense input views often yields a counterintuitive outcome in which reconstruction fidelity decreases as the number of input views increases [5, 16]. Two common design choices explain this behavior. The first lies in the global aggregation of the backbone [14] when the sequence length grows without known poses. In the absence of known poses, global aggregation relies on softmax attention to discover correspondences across many tokens. As the sequence length grows, the attention mass diffuses over many candidates, reducing the weight assigned to correct matches and weakening cross-view consistency [13]. The second lies in voxel fusion [5]. Per-pixel Gaussians are aggregated into a voxel grid to support efficient differentiable rasterization. When many Gaussians fall into the same voxel, a normalized average behaves as a low-pass filter. Strong contributors are diluted by many weak ones and high-frequency detail is lost. The issue becomes worse when the confidence of the contributor varies, but the fusion weights are not adaptive.

We propose AVSplat to improve global correspondence reasoning and voxel representation in the dense-view setting. We first propose an Assist View Preconditioning module. This module is designed to address a dense-view failure mode of prior pose-free pipelines, where the global attention becomes increasingly diffuse and cross-view matching becomes unstable as the number of input views grows. In this module, each current view performs a single lightweight interaction with a small Assist View set before any global attention. The Assist Views are both relevant to the current view and complementary to each other. Selection first finds strong neighbors by embedding similarity, then promotes diversity with a coverage-aware criterion, so that a small budget captures complementary observations and improves scene coverage. The features of Assist Views are cached for efficiency. The current view attends to these features and obtains conditioned tokens that carry the scene context before global aggregation. This focuses the correspondence search on plausible matches and reduces attention dispersion. A gated residual controls conditioning strength early in training and stabilizes optimization.

AVSplat also introduces Occupancy-guided Voxel Fusion, which adapts the fusion temperature to voxel occupancy and local consistency. High occupancy produces a sharper weight distribution, allowing reliable contributors to dominate the fused attributes. Under low occupancy, a smoother distribution retains information from multiple candidates and avoids brittle winner-take-all fusion. The temperature is constrained to a fixed range to stabilize optimization, while multi-view agreement and photometric residuals reduce the influence of outliers. This design helps preserve high-frequency structure in crowded voxels.

By converting additional views into a useful signal during both aggregation and representation, AVSplat preserves detail under high voxel occupancy and

keeps correspondence search focused at long sequence length. In pose-free settings, it consistently improves overall performance with more input views under comparable compute and memory. Ablations confirm that Assist View Preconditioning and Occupancy-guided Voxel Fusion are both necessary for consistent gains. The analysis explains why prior feed-forward pipelines degrade when inputs become dense and shows that modeling attention sparsity and voxel attribution together leads to scalable and high-fidelity 3D Gaussian Splatting.
Our main contributions are:

- We present AVSplat, a pose-free feed-forward framework that maintains positive view scaling in dense multi-view settings through improvements in global aggregation and voxel representation.
- We introduce Assist View Preconditioning, a lightweight per-view conditioning step that restores positive view scaling by preventing attention dilution and stabilizing cross-view matching at long sequence lengths.
- We propose Occupancy-guided Voxel Fusion, where the sampling distribution varies with the voxel occupancy and local consistency so that high-frequency detail is preserved.
- We conduct extensive experiments in pose-free settings, showing that AVSplat maintains or improves performance as the number of input views increases.

2 Related Work

2.1 Geometry foundation models

Recent geometry foundation models predict dense point maps and correspondences directly from few views. DUSt3R [15] recovers accurate point maps without calibration, providing strong priors for reconstruction and tracking. MASt3R [8] grounds the matching in 3D, producing robust, coarse-to-fine correspondences under wide baselines. MUSt3R [2] extends to multi-view sequences and enables high-frame-rate inference for offline or online reconstruction. VGGT [14] advances the foundation model line by inferring camera parameters, depth, point maps, and point tracks in a single pass from one view to many views. The model alternates frame-wise and global aggregation and therefore reasons jointly over all frames. The approach improves the accuracy of feed-forward reconstruction and reduces reliance on multi-stage optimization. Analyses of global attention in VGGT and in related transformers reveal a dominant runtime cost that grows quadratically with sequence length and a sparse pattern in which a small set of patch interactions carries most probability mass. Replacing dense global attention with block sparse kernels preserves accuracy while accelerating inference on long sequences.

2.2 Feed-forward 3D Gaussian Splatting

Feed-forward 3DGS predicts scene-specific Gaussians and renders novel views without per-scene optimization. PixelSplat [3] learns from image pairs with a

3D probability field for stable training and real-time rendering. MVSplat [4] uses plane-sweep cost volumes to extract clean Gaussians from sparse multi-view inputs. DepthSplat [19] couples splatting with monocular depth, yielding mutual gains. MuSASplat [21] improves pose-free sparse-view 3DGS through lightweight multi-scale adaptation and efficient cross-view feature fusion, while ZPressor [16] compresses multi-view inputs into a compact latent state, enabling feed-forward 3DGS models to scale to over 100 input views.

Pose-free and COLMAP-free pipelines operate on uncalibrated inputs. Splatt-3R [12] transfers MASt3R priors to stereo; NoPoSplat [22] anchors Gaussians in a canonical camera space for real-time inference; SelfSplat [6] removes external 3D priors via self-supervised depth and pose. FLARE [26] features a cascaded learning paradigm with camera pose serving as the critical bridge, achieving accurate camera pose estimation and high-quality 3D rendering. PCR-GS [18] tackles COLMAP-free 3DGS from a per-scene optimization perspective through pose co-regularization. In contrast, AVSplat follows the one-pass feed-forward setting and focuses on dense-view scaling.

Recent methods adopt geometry transformers: AnySplat [5] encodes uncalibrated collections with a VGGT-style aggregator and predicts Gaussians and poses with voxel-level grouping. Scaling to dense inputs exposes two issues—diluted global attention and high-occupancy voxel fusion that smooths high-frequency details—which we address by strengthening correspondence before global aggregation and refining fusion in crowded voxels.

Complementary 3DGS advances improve Gaussian representation, detail preservation, and controllable appearance. SOGS [23] introduces second-order anchors to improve anchor-based 3DGS with compact anchor features and reduced model size. FreGS [24] uses progressive frequency regularization to guide Gaussian densification from low to high frequencies and alleviate over-reconstruction. StyleGaussian [11] extends Gaussian splatting to instant 3D style transfer through efficient feature rendering and a KNN-based 3D CNN decoder. These works are orthogonal to AVSplat, which targets dense-view aggregation and voxel attribution in pose-free feed-forward reconstruction.

3 Method

This section first recalls the principles of 3D Gaussian Splatting [7] and the transformer-style multi-view geometry encoder (VGGT [14]) that we build upon. We then present the overall pipeline of our method. The pipeline extends the previous feed-forward Gaussian predictor [5] with two new components. The first component performs assist view preconditioning so that every input view receives compact scene context before global aggregation, with the primary goal of preventing dense-view degradation caused by attention dispersion. The second component performs occupancy-guided temperature fusion so that voxel-level attributes remain sharp under dense inputs.

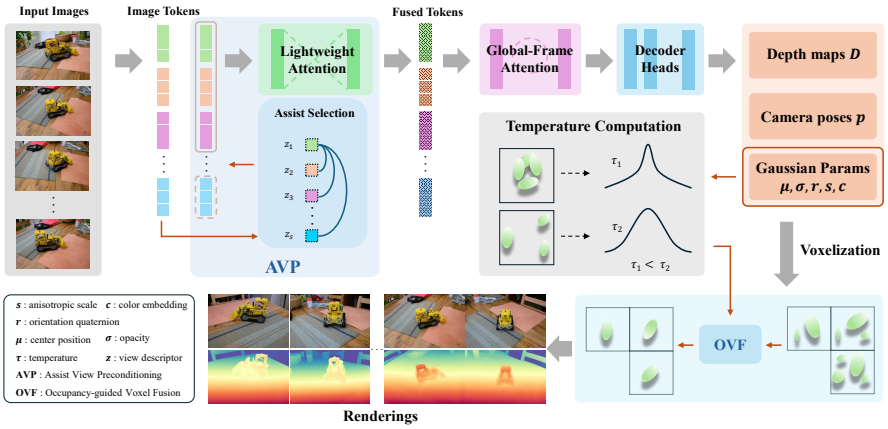

Fig. 1: Overview of the AVSplat pipeline. The encoder converts each input image into patch embeddings (image tokens) and aggregates them into per-view tokens. Assist-View Preconditioning (AVP) computes a descriptor for each view and injects scene context before global aggregation. Decoder heads then predict camera parameters, depth, and Gaussian primitives. Occupancy-guided Voxel Fusion (OVF) forms voxel-level attributes and modulates them by voxel occupancy, followed by differentiable Gaussian rasterization to produce the final rendering.

3.1 Preliminaries

3D Gaussian Splatting In 3D Gaussian Splatting (3DGS) [7], a scene is represented by a finite set of anisotropic Gaussians

$$\mathcal{G} = \{G_i\}_{i=1}^{N}, \qquad G_i = \big(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \mathbf{c}_i, \alpha_i\big). \tag{1}$$

Here $\boldsymbol{\mu}_i \in \mathbb{R}^3$ is the mean, $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times3}$ is a positive definite covariance, $\mathbf{c}_i \in \mathbb{R}^3$ is the radiance and $\alpha_i \in [0,1]$ is the opacity. For a calibrated camera with projection Π and view direction $\mathbf{v}$, each Gaussian projects to the image plane as

$$\mathbf{u}_i = \Pi(\boldsymbol{\mu}_i) \qquad \mathbf{S}_i = \mathbf{J}_i\, \boldsymbol{\Sigma}_i\, \mathbf{J}_i^{\top}, \tag{2}$$

where $\mathbf{J}_i$ is the Jacobian of Π at $\boldsymbol{\mu}_i$. Let $\mathcal{S}(G_i, \mathbf{u})$ denote the induced 2D Gaussian kernel at pixel $\mathbf{u}$. Rendering proceeds in front-to-back order with color

$$\mathbf{C}(\mathbf{u}) = \sum_{i=1}^{N} T_i(\mathbf{u})\, \alpha_i\, \mathcal{S}(G_i, \mathbf{u})\, \mathbf{c}_i, \tag{3}$$

and transmittance

$$T_i(\mathbf{u}) = \prod_{j<i} \big(1 - \alpha_j\, \mathcal{S}(G_j, \mathbf{u})\big). \tag{4}$$

Classical methods optimize $\mathcal{G}$ per scene from multi-view supervision. Feed-forward 3DGS removes per-scene optimization. A network takes images $\{I_s\}_{s=1}^{S}$ and directly predicts $\mathcal{G}$ together with camera intrinsics and extrinsics so that the rendering above supplies differentiable supervision in a single forward pass. The

main challenges become multi-view feature aggregation and redundancy control in the predicted Gaussians.

Transformer style multi-view geometry encoder In VGGT [14], a geometry-aware encoder receives a sequence of images and produces tokens that carry appearance and 3D structure. For each view s the encoder yields patch tokens

$$\mathbf{X}_s \in \mathbb{R}^{P\times d}, \quad s = 1, \ldots, S. \tag{5}$$

Tokens are processed by alternating view-specific aggregation and global aggregation. The global stage operates on the concatenated set

$$\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_S] \in \mathbb{R}^{(SP)\times d}. \tag{6}$$

This allows information to flow across views when poses are unknown. Special tokens can be attached to support the prediction of camera parameters, depth, and point tracks. Global aggregation has quadratic complexity in the number of tokens, and its attention maps tend to become increasingly diffuse as sequence length grows. Our method addresses this weakness.

3.2 AVSplat pipeline

Given uncalibrated images, the encoder produces view tokens. A scene-level aggregation consumes all tokens and produces latents for camera estimation and Gaussian prediction. From these latents we decode intrinsics, extrinsics, Gaussian centers, scales, orientations, colors, and opacities. The resulting Gaussians are voxelized and rendered. An overview of our pipeline is shown in Figure 1.

Our pipeline introduces two new designs. Before tokens enter global aggregation, we insert an assist view preconditioning module so that each view collects context from a compact set of assisting views. After Gaussian prediction, we replace uniform voxel fusion [5] with an occupancy-guided temperature fusion module that modulates fusion weights according to voxel occupancy and reliability. We refer to these modules as *Assist View Preconditioning* and *Occupancy-guided Voxel Fusion*.

3.3 Assist View Preconditioning

Global aggregation is expected to resolve correspondences across many views. When the number of views increases, tokens from unrelated views inflate the attention pool and the distribution becomes diffuse [13]. Tokens from geometrically compatible views receive less probability mass and long-range constraints fail to strengthen the global match. Each view should first collect context from a compact and diverse set of assisting views and should enter global aggregation with this context.

Coverage driven assist selection For view s with tokens $\mathbf{X}_s \in \mathbb{R}^{P\times d}$ we form a view descriptor by average pooling

$$\mathbf{z}_s = \frac{1}{P}\sum_{p=1}^{P} \mathbf{X}_s^{(p)} \in \mathbb{R}^d. \tag{7}$$

We compute cosine similarity between views

$$m_{st} = \frac{\mathbf{z}_s^\top \mathbf{z}_t}{\|\mathbf{z}_s\|_2 \, \|\mathbf{z}_t\|_2}. \tag{8}$$

A candidate pool is formed by the M nearest neighbors of s in descriptor space

$$\mathcal{N}(s) = \mathrm{TopM}\big\{\, t \neq s \,:\, m_{st} \,\big\}. \tag{9}$$

To measure how well a candidate view supports the tokens of s, we define a token-level support score. Let $\kappa(\mathbf{a},\mathbf{b}) = \mathbf{a}^\top\mathbf{b}/(\|\mathbf{a}\|_2\,\|\mathbf{b}\|_2)$. Using a lightweight projection $\mathbf{W}_c \in \mathbb{R}^{d\times d}$, the support from view t to token p of s is

$$s_{t,p} = \max_{q\in\{1,\ldots,P\}} \kappa\big(\mathbf{X}_s^{(p)}\mathbf{W}_c,\ \mathbf{X}_t^{(q)}\mathbf{W}_c\big). \tag{10}$$

We then define a coverage objective over an assist set $\mathcal{A} \subset \mathcal{N}(s)$ of size K

$$\begin{aligned} F(\mathcal{A}) = \frac{1}{P}\sum_{p=1}^{P} \max_{t\in\mathcal{A}} s_{t,p} \;+\; \beta\,\frac{1}{K}\sum_{t\in\mathcal{A}} q_t \\ -\;\lambda\,\frac{1}{K(K-1)}\sum_{\substack{t,u\in\mathcal{A}\\ t\neq u}} m_{tu}, \end{aligned} \tag{11}$$

where q_t is a view reliability term derived from token variance, the first term rewards token coverage, the second term prefers reliable helpers, and the third term discourages redundant views. Because the first term is monotone and submodular, greedy selection provides a near-optimal solution at low computational cost. Starting from $\mathcal{A}_0 = \varnothing$, we update

$$\mathcal{A}_{k+1} = \mathcal{A}_k \cup \big\{\arg\max_{t\in\mathcal{N}(s)\setminus\mathcal{A}_k} \Delta F(t \mid \mathcal{A}_k)\big\}, \tag{12}$$

until $|\mathcal{A}| = K$, where $\Delta F(t \mid \mathcal{A}) = F(\mathcal{A}\cup\{t\}) - F(\mathcal{A})$. In practice we evaluate $s_{t,p}$ on a stratified subset of tokens, cache per-view projections, and use random tie-breaking to improve generalization.

Lightweight conditioning Given the selected assist set $\mathcal{A}(s)$, we apply a single cross-view interaction to inject context. With linear maps absorbed into keys, queries, and values, the conditioned tokens are

$$\widetilde{\mathbf{X}}_s = \mathrm{softmax}\Big(\tfrac{1}{\sqrt{d}}\mathbf{Q}_s\mathbf{K}_s^\top\Big)\mathbf{V}_s, \tag{13}$$

where $\mathbf{K}_s$ and $\mathbf{V}_s$ are formed by concatenating tokens from the assist set and $\mathbf{Q}_s$ is obtained from $\mathbf{X}_s$. We combine original and conditioned tokens through a gated residual

$$\mathbf{X}_s^{\text{out}} = \mathbf{X}_s + \gamma(g)\,\widetilde{\mathbf{X}}_s, \tag{14}$$

where g denotes the current training step and $\gamma(g)$ increases from near zero to one. The global aggregation then consumes $\{\mathbf{X}_s^{\text{out}}\}_{s=1}^{S}$.

This selection strategy favors views that cover different regions of the reference view and that agree with its content, while avoiding near duplicates. The single conditioning step moves the burden of discovery from the global module to the selection stage, so the subsequent aggregation focuses on long-range alignment rather than search. The resulting view-conditioned tokens are then fed into the same global aggregation and decoding heads as in the baseline.

3.4 Occupancy-guided Voxel Fusion

After obtaining all Gaussians, we project them onto a voxel grid so that Gaussians falling into the same or nearby voxels can be fused, reducing redundancy. Simple averaging within a voxel attenuates high-frequency appearance and smooths fine-scale geometry, because numerous weak contributors dilute a few strong ones. The effect increases with occupancy and leads to over-smoothed reconstructions.

For voxel v we define the candidate set $\mathcal{G}_v = \{G_i\}_{i\in\mathcal{I}_v}$, and the occupancy $n_v = |\mathcal{I}_v|$, where $\mathcal{I}_v$ is the index set of candidates that are assigned to voxel v by the voxelization step. Each candidate provides a feature vector $\mathbf{f}_i \in \mathbb{R}^c$, a 3D point $\mathbf{p}_i \in \mathbb{R}^3$ and a confidence $c_i \in \mathbb{R}_{\geq 0}$. We define a temperature

$$\tau_v = \text{clip}\Big(\tau_{\min},\, \tau_0\Big(\frac{n_v}{n_{\text{ref}}}\Big)^{-\alpha}\psi_v,\, \tau_{\max}\Big). \tag{15}$$

with base temperature τ_0, exponent α and bounds $\tau_{\min}, \tau_{\max}$. The reliability factor ψ_v depends on geometric consistency. We compute a variance

$$\sigma_v^2 = \frac{1}{n_v}\sum_{i\in\mathcal{I}_v}\|\mathbf{p}_i - \bar{\mathbf{p}}_v\|_2^2, \tag{16}$$

with mean $\bar{\mathbf{p}}_v = \frac{1}{n_v}\sum_{i\in\mathcal{I}_v}\mathbf{p}_i$, and set

$$\psi_v = \frac{1}{1+\lambda\,\sigma_v^2}. \tag{17}$$

Here σ_v^2 measures the dispersion of 3D points within the voxel. Voxels with small variance are more geometrically consistent and therefore assigned a larger reliability factor ψ_v. We then form logits $\ell_i = \frac{c_i}{\tau_v}$ for $i \in \mathcal{I}_v$ and compute softmax weights

$$w_i = \frac{\exp(\ell_i)}{\sum_{j\in\mathcal{I}_v}\exp(\ell_j)}. \tag{18}$$

To reduce the influence of outliers, we sort candidates in descending order of w_i and retain the shortest prefix whose cumulative weight reaches a target mass. The fused voxel feature and position are

$$\mathbf{f}_v = \sum_{i \in \mathcal{I}_v} w_i \, \mathbf{f}_i, \qquad \mathbf{p}_v = \sum_{i \in \mathcal{I}_v} w_i \, \mathbf{p}_i. \tag{19}$$

High occupancy reduces the temperature and sharpens the distribution so that candidates with high confidence or strong geometric agreement dominate. Low occupancy yields a higher temperature so that the fusion remains inclusive. The procedure preserves detail in well-observed regions and remains robust where support is limited. All steps are differentiable and trained jointly with the rest of the network.

Our method changes the order of information integration and the way voxel attribution is formed. Previous methods bring all tokens into a single pool and rely on a global module to discover relations. Assist view preconditioning makes each view context-aware before that stage. Standard fusion treats all candidates inside a voxel as equal competitors. Occupancy-guided Voxel Fusion makes the voxel sensitive to both how many candidates it receives and how consistent they are. These changes address the main causes of saturation in dense-view feed-forward 3DGS and restore the behavior that more views improve reconstruction quality.

4 Experiments

4.1 Experimental setup

Datasets Our model is trained on the DL3DV dataset [9], which provides over 10k training scenes. Novel view synthesis quality is evaluated on Mip-NeRF 360 [1], VR-NeRF [20], and the DL3DV [9] test split. The DL3DV test split contains 140 scenes. For Mip-NeRF 360 and VR-NeRF we follow the settings used in prior work [5]. The sparse view setting uses 3, 6, and 16 input views per scene, and 2 target views per scene. The dense-view setting uses 32, 48, and 64 input views per scene, and 4, 6, and 8 target views, respectively. For VR-NeRF we use four indoor scenes that cover both compact and spacious layouts: apartment, kitchen, raf-furnishedroom, and workshop. For Mip-NeRF 360 we use four scenes with diverse viewpoints and lighting: bonsai, counter, kitchen, and room. These eight scenes span different room types, camera densities, and appearance patterns and are used for both sparse and dense-view evaluation. For each of the 140 DL3DV test scenes we first sample 75 candidate views, fix three random views as target views, and then randomly choose context views from the remaining views according to the desired input count.

Metrics and baselines. We report PSNR, SSIM [17], and LPIPS [25]. SSIM is computed with a Gaussian window in linear color space. LPIPS uses the VGG backbone with official weights. Scores are first averaged over target views within

Table 1: Quantitative results on Mip-NeRF 360 [1], VR-NeRF [20], and DL3DV [9] under sparse and dense inputs with metrics PSNR↑, SSIM↑ [17], LPIPS↓ [25]. The best result in each group is highlighted in red and the second best in orange. AVSplat attains rendering quality comparable to the strongest baselines and avoids the dense-view degradation observed in prior pose-free pipelines. Our performance remains stable or improves as more input views are added, instead of oscillating or degrading at high view counts.

	Sparse inputs								
	3			6			16		
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Mip-NeRF 360 [1]									
NoPoSplat [22]	16.36	0.430	0.453	15.92	0.416	0.541	15.47	0.361	0.606
Flare [26]	13.52	0.350	0.601	15.35	0.407	0.573	13.21	0.348	0.695
AnySplat [5]	17.29	0.429	0.433	17.51	0.446	0.412	17.81	0.448	0.382
AVSplat	17.33	0.451	0.354	17.99	0.467	0.352	18.13	0.428	0.390
VR-NeRF [20]									
NoPoSplat [22]	18.37	0.707	0.437	17.57	0.704	0.466	17.66	0.720	0.472
Flare [26]	18.58	0.717	0.470	18.26	0.717	0.477	17.02	0.709	0.510
AnySplat [5]	20.41	0.733	0.389	21.07	0.729	0.436	21.12	0.714	0.354
AVSplat	20.24	0.764	0.403	20.91	0.736	0.418	21.17	0.724	0.391
DL3DV [9]									
AnySplat [5]	15.04	0.387	0.612	15.74	0.420	0.492	17.43	0.542	0.419
AVSplat	14.41	0.341	0.598	15.62	0.424	0.492	17.80	0.565	0.344
	Dense inputs								
	32			48			64		
Mip-NeRF 360 [1]									
NoPoSplat [22]	OOM			OOM			OOM		
Flare [26]	OOM			OOM			OOM		
AnySplat [5]	18.22	0.445	0.369	19.40	0.476	0.366	19.23	0.500	0.365
AVSplat	18.71	0.483	0.365	19.30	0.482	0.368	20.11	0.528	0.366
VR-NeRF [20]									
NoPoSplat [22]	OOM			OOM			OOM		
Flare [26]	OOM			OOM			OOM		
AnySplat [5]	21.11	0.711	0.380	21.60	0.745	0.338	21.26	0.719	0.350
AVSplat	21.42	0.720	0.365	21.62	0.734	0.317	21.7	0.723	0.363
DL3DV [9]									
AnySplat [5]	21.05	0.680	0.284	21.04	0.678	0.281	21.14	0.684	0.282
AVSplat	21.54	0.710	0.266	21.60	0.713	0.265	21.67	0.717	0.265

each scene and then averaged over scenes. We compare with AnySplat [5] in the dense-view setting. The sparse view comparison includes NoPoSplat [22], FLARE [26], and AnySplat. Since evaluation on Mip-NeRF 360 and VR-NeRF is performed in a zero-shot setting, we retrain AnySplat on DL3DV using the public implementation and recommended configuration so that all methods share the same training data when comparing generalization performance.

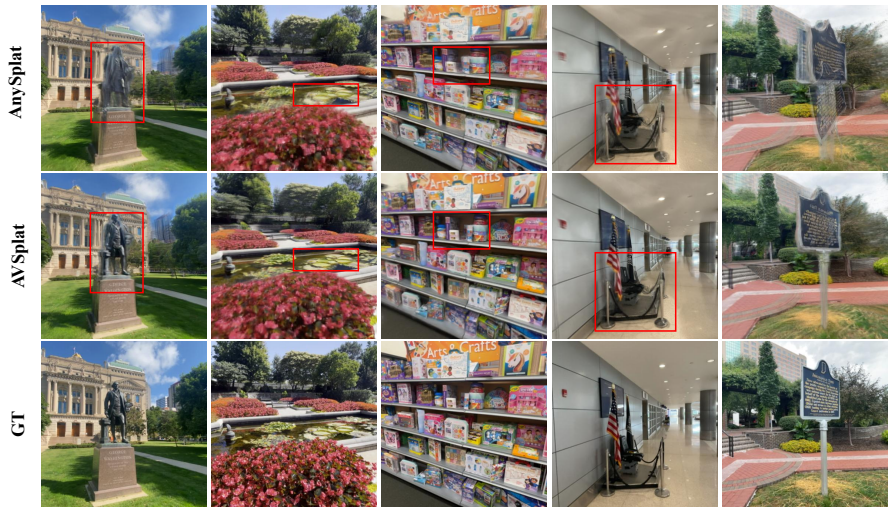

Fig. 2: Qualitative comparison between AVSplat and AnySplat [5] under dense-view inputs. We focus on the dense-view setting, where the artifacts we target primarily occur, and therefore omit sparse-view results. Red boxes highlight that our method preserves more fine-grained surface details with fewer artifacts and less oversmoothing-induced blur, while the rightmost column shows more accurate global matching.

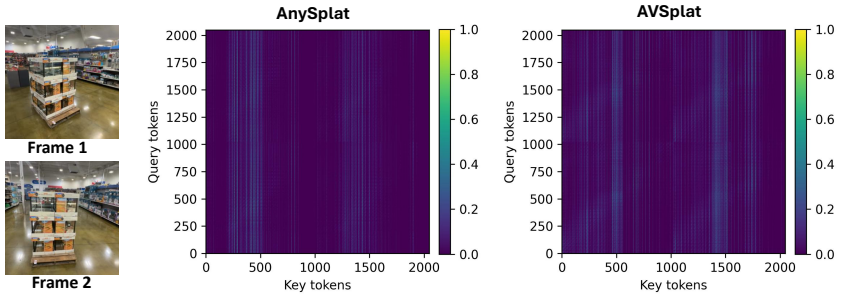

Fig. 3: Cross-view attention at layer 15 for a fixed frame pair under 16-view inputs. The maps are extracted from the full attention matrix after removing self-frame blocks and special tokens, and are displayed with identical normalization. AVSplat produces more concentrated cross-view responses and the geometric consistency is quantified in Tab. 3.

4.2 Implementation details

Backbone and tokens The backbone is a transformer-style multi-view geometry encoder derived from VGGT [14]. Images are resized to a shorter side of 518 with preserved aspect ratio. The encoder uses a patch size of 14, an embedding

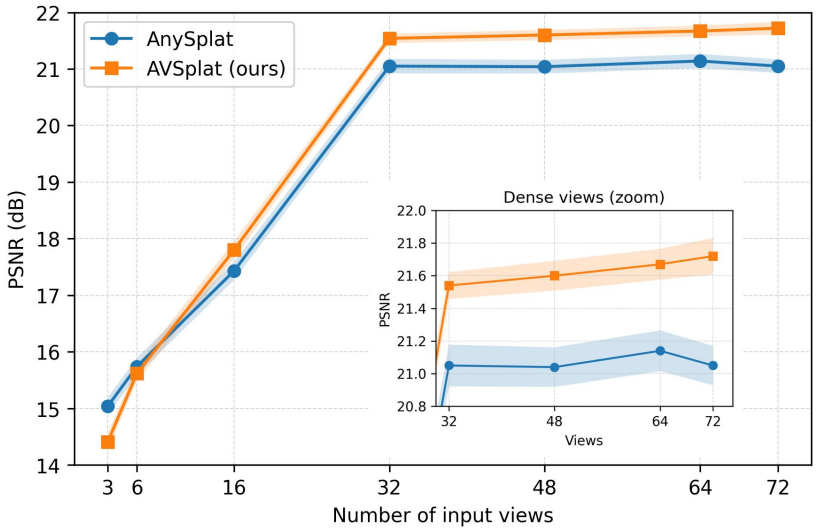

Fig. 4: Performance versus input count with 3, 6, 16, 32, 48, 64, and 72 input views. The vertical axis reports PSNR and the horizontal axis shows the number of input views. Shaded regions indicate the standard deviation across multiple trials. In the dense-view setting, AVSplat consistently outperforms AnySplat [5]. The inset in the lower right zooms into the dense-view region and shows that AnySplat begins to oscillate as the number of input views increases, whereas AVSplat maintains stable or improving performance without degradation.

Table 2: Ablations on the DL3DV [9] in the 64 views setting. Occupancy-guided Voxel Fusion contributes most of the single-point image-quality improvement at 64 views, while Assist-View Preconditioning mainly targets the dense-view scaling failure mode by stabilizing global matching as the input count grows.

Variant	PSNR ↑	SSIM↑	LPIPS ↓
Baseline	21.14	0.684	0.282
+ Fusion only	21.57	0.692	0.272
+ Assist + Fusion	21.67	0.717	0.265

Table 3: Additional dense-view diagnostics. Left: cross-view attention analysis on DL3DV with 16 input views. GeoMass@1patch measures the attention assigned to a one-patch-width epipolar band. Right: PSNR (dB) on the overlap-heavy ACID evaluation. Ground-truth poses are used only to compute the diagnostic metric and are never provided to either model.

Attention diagnostics			ACID overlap stress test		
Method	Entropy ↓	GeoMass ↑	Method	48 views	72 views
AnySplat	0.693	0.286	AnySplat	23.12	22.45
AVSplat	**0.642**	**0.374**	AVSplat	**23.15**	**23.24**

dimension of 1024, and 24 layers with 16 attention heads. Each view yields a grid of patch tokens and a small set of special tokens. A global aggregation stage consumes tokens from all views.

Training schedule We train the model for 40,000 steps using AdamW with a learning rate of 1×10^{-4} and a weight decay of 5×10^{-2}. We use a cosine learning-rate schedule with 2,000 warm-up steps. Total steps are 40000. Mixed precision and activation checkpointing are enabled. Gradient norm is clipped at 1. During training, the number of input images per scene is a random integer in $[4, 24]$. Our model is trained with a batch size of 1 on a single NVIDIA A100 GPU with 80 GB memory.

Assist View Preconditioning Each view selects 4 assisting views. We apply lightweight jittering with a probability of 0.05. The gating factor starts near 0 during a 2,000-step warm-up period and then increases linearly for 8000 steps until it reaches 1. We stop gradients through the assist branch for the first 1,500 steps. In the sparse-view setting, the number of available views is too small to form a meaningful assist set. Therefore, Assist-View Preconditioning is disabled in the sparse-view setting.

Occupancy-guided Voxel Fusion Voxel fusion uses a temperature that varies with occupancy and reliability. The base temperature is 0.7. The occupancy exponent is 0.5. The reference count is 8. The temperature is clipped to $[0.15, 2.0]$. The geometric-variance term is assigned a weight of 2.0. We use a top-mass threshold of 0.8 to truncate low-weight candidates.

4.3 Results and analysis

Main results across sparse and dense inputs Table 1 reports results for sparse inputs with 3, 6, and 16 views and dense inputs with 32, 48, and 64 views. AVSplat remains competitive across all three datasets and shows its clearest gains on DL3DV. In the sparse setting, AVP is disabled because too few views are available to form a meaningful assist set. Under dense inputs, AVSplat exhibits more

stable improvements as the view count increases, which is the setting targeted by AVP and occupancy-guided fusion.

Figure 2 provides a qualitative comparison between AVSplat and AnySplat [5] under dense-view inputs. Since the artifacts we target primarily arise in the dense-view setting, we only show qualitative comparisons for dense-view inputs here. Consistent with the quantitative trends, regions highlighted by red boxes show that AVSplat recovers sharper surface details with fewer artifacts and less oversmoothing-induced blur. The rightmost column further indicates that AVSplat produces more accurate global matching, leading to better large-scale structural consistency in the reconstructed scenes.

Dense-view scaling and overlap stress test. Under the fixed-target DL3DV protocol, AVSplat improves over AnySplat by 0.49, 0.56, and 0.53 dB at 32, 48, and 64 input views, respectively. More importantly, its PSNR continues to increase from 21.54 to 21.67 dB over this range, whereas the closest baseline largely saturates.

The difference becomes more pronounced for highly overlapping inputs. As shown in Table 3, on ACID [10], AnySplat decreases from 23.12 dB at 48 views to 22.45 dB at 72 views, whereas AVSplat increases from 23.15 to 23.24 dB, producing a 0.79 dB advantage at 72 views. The practical benefit is therefore not that every additional view produces a large single-step gain, but that redundant dense views no longer become harmful.

Attention analysis We visualize the activations of the global aggregation stage in Figure 3 to better understand how assist-view preconditioning affects cross-view matching. For a fixed pair of input frames, we extract the post-softmax attention matrix from layer 15 of the full global-aggregation sequence. We remove special tokens and self-frame blocks, and then retain the cross-frame block corresponding to the selected pair. Given these tokens, we compute the attention matrix as $\mathrm{softmax}(QK^\top)$, which yields a $2P \times 2P$ map where P denotes the number of patch tokens per frame. The baseline exhibits diffuse attention mass when the number of views is large, indicating that global matching is poorly focused. In contrast, AVSplat produces attention patterns that are more sharply concentrated on geometrically plausible correspondences after preconditioning, demonstrating that our design effectively improves global matching.

4.4 Ablation studies

We ablate the Assist-View Preconditioning and Occupancy-guided Voxel Fusion modules in the 64-view setting on DL3DV [9]. Removing preconditioning reduces attention focus in the global aggregation stage and leads to lower PSNR and SSIM at long sequence lengths. Removing occupancy guidance causes excessive smoothing in high-occupancy regions, producing blur in the rendered images and suppressing fine surface details, which in turn degrades overall performance. As shown in Table 2, both components are effective. Occupancy-guided Voxel Fusion contributes most of the single-point image-quality improvement at 64 views, while Assist-View Preconditioning mainly targets the dense-view scaling failure

mode by stabilizing global matching as the input count grows (see Figures 3 and 4).

Performance vs. Number of input views Figure 4 further analyzes how performance scales with the number of input views. We plot PSNR for AVSplat and AnySplat using 3, 6, 16, 32, 48, 64, and 72 input views, with shaded bands indicating the standard deviation across multiple trials. Across the full range, both methods benefit substantially from moving from sparse to dense inputs. Beyond 32 views, however, their scaling behavior diverges. AnySplat begins to oscillate and slightly degrades at high view counts, whereas AVSplat continues to improve. The absolute gap at a fixed view count can be modest, but the scaling behavior differs consistently, which is the primary effect we target with Assist-View Preconditioning. This shows that AVSplat can consistently exploit additional views in dense settings, and complements the quantitative trends observed in Table 1.

Quantitative attention diagnostics. To quantify the effect of AVP beyond the qualitative heatmaps in Fig. 3, we extract the post-softmax attention matrix from layer 15 of the full global-aggregation sequence. We exclude self-frame blocks and special tokens and average the results over four fixed view pairs. For cross-view attention a_{qk}, normalized entropy is the per-query entropy divided by the logarithm of the number of valid key patches. GeoMass@1patch is defined as GeoMass $= \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}}\sum_{k\in\mathcal{E}(q)} a_{qk}$, where $\mathcal{E}(q)$ contains key patches within a one-patch-width epipolar band computed from ground-truth evaluation poses. As shown in Table 3, AVSplat reduces normalized entropy from 0.693 to 0.642 and increases GeoMass@1patch from 0.286 to 0.374, corresponding to an absolute gain of 8.8 points and a relative gain of 30.8%. The mass of a random band is approximately 0.10, so the concentration on geometrically compatible regions increases from approximately 2.9× to 3.7× the random baseline.

5 Conclusion

This work addresses dense-view degradation in pose-free feed-forward 3D Gaussian Splatting. Assist-View Preconditioning conditions each view before global aggregation, while Occupancy-guided Voxel Fusion adapts voxel weights to occupancy and reliability. Experiments on Mip-NeRF 360,VR-NeRF, and DL3DV show more stable performance as the number of input views increases, and the attention analysis and ablations support the contributions of the two modules. Together, these changes improve dense-view pose-free reconstruction without per-scene optimization.

Acknowledgements

This work was supported by the Ministry of Education, Singapore, under the Tier-2 project scheme (Project No. MOE-T2EP20123-0003).

References

1. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5470–5479 (2022)
2. Cabon, Y., Stoffl, L., Antsfeld, L., Csurka, G., Chidlovskii, B., Revaud, J., Leroy, V.: Must3r: Multi-view network for stereo 3d reconstruction. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 1050–1060 (2025)
3. Charatan, D., Li, S.L., Tagliasacchi, A., Sitzmann, V.: 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. pp. 19457–19467 (2024)
4. Chen, Y., Xu, H., Zheng, C., Zhuang, B., Pollefeys, M., Geiger, A., Cham, T.J., Cai, J.: Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In: European Conference on Computer Vision. pp. 370–386. Springer (2024)
5. Jiang, L., Mao, Y., Xu, L., Lu, T., Ren, K., Jin, Y., Xu, X., Yu, M., Pang, J., Zhao, F., et al.: Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. arXiv preprint arXiv:2505.23716 (2025)
6. Kang, G., Yoo, J., Park, J., Nam, S., Im, H., Shin, S., Kim, S., Park, E.: Selfsplat: Pose-free and 3d prior-free generalizable 3d gaussian splatting. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 22012–22022 (2025)
7. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
8. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: European Conference on Computer Vision. pp. 71–91. Springer (2024)
9. Ling, L., Sheng, Y., Tu, Z., Zhao, W., Xin, C., Wan, K., Yu, L., Guo, Q., Yu, Z., Lu, Y., et al.: Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22160–22169 (2024)
10. Liu, A., Tucker, R., Jampani, V., Makadia, A., Snavely, N., Kanazawa, A.: Infinite nature: Perpetual view generation of natural scenes from a single image. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14458–14467 (2021)
11. Liu, K., Zhan, F., Xu, M., Theobalt, C., Shao, L., Lu, S.: Stylegaussian: Instant 3d style transfer with gaussian splatting. In: SIGGRAPH Asia 2024 Technical Communications, pp. 1–4 (2024)
12. Smart, B., Zheng, C., Laina, I., Prisacariu, V.A.: Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs. arXiv preprint arXiv:2408.13912 (2024)
13. Wang, C.S.B., Schmidt, C., Piekenbrinck, J., Leibe, B.: Faster vggt with block-sparse global attention. arXiv preprint arXiv:2509.07120 (2025)
14. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5294–5306 (2025)
15. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20697–20709 (2024)
16. Wang, W., Chen, D.Y., Zhang, Z., Shi, D., Liu, A., Zhuang, B.: Zpressor: Bottleneck-aware compression for scalable feed-forward 3dgs. arXiv preprint arXiv:2505.23734 (2025)

17. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
18. Wei, Y., Zhang, J., Zhang, X., Shao, L., Lu, S.: Pcr-gs: Colmap-free 3d gaussian splatting via pose co-regularizations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 26499–26508 (2025)
19. Xu, H., Peng, S., Wang, F., Blum, H., Barath, D., Geiger, A., Pollefeys, M.: Depthsplat: Connecting gaussian splatting and depth. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 16453–16463 (2025)
20. Xu, L., Agrawal, V., Laney, W., Garcia, T., Bansal, A., Kim, C., Rota Bulò, S., Porzi, L., Kontschieder, P., Božič, A., et al.: Vr-nerf: High-fidelity virtualized walkable spaces. In: SIGGRAPH Asia 2023 Conference Papers. pp. 1–12 (2023)
21. Xu, M., Zhan, F., Zhang, X., Shao, L., Lu, S.: Musasplat: Efficient sparse-view 3d gaussian splats via lightweight multi-scale adaptation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 40, pp. 11343–11351 (2026)
22. Ye, B., Liu, S., Xu, H., Li, X., Pollefeys, M., Yang, M.H., Peng, S.: No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. arXiv preprint arXiv:2410.24207 (2024)
23. Zhang, J., Zhan, F., Shao, L., Lu, S.: Sogs: Second-order anchor for advanced 3d gaussian splatting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11167–11176 (2025)
24. Zhang, J., Zhan, F., Xu, M., Lu, S., Xing, E.: Fregs: 3d gaussian splatting with progressive frequency regularization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21424–21433 (2024)
25. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
26. Zhang, S., Wang, J., Xu, Y., Xue, N., Rupprecht, C., Zhou, X., Shen, Y., Wetzstein, G.: Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 21936–21947 (2025)