ZOOMSPLAT: RETHINKING MULTI-VIEW FREQUENCY BOUNDS FOR ANTI-ALIASED FEED-FORWARD 3D GAUSSIAN SPLATTING

Anonymous authors
Paper under double-blind review

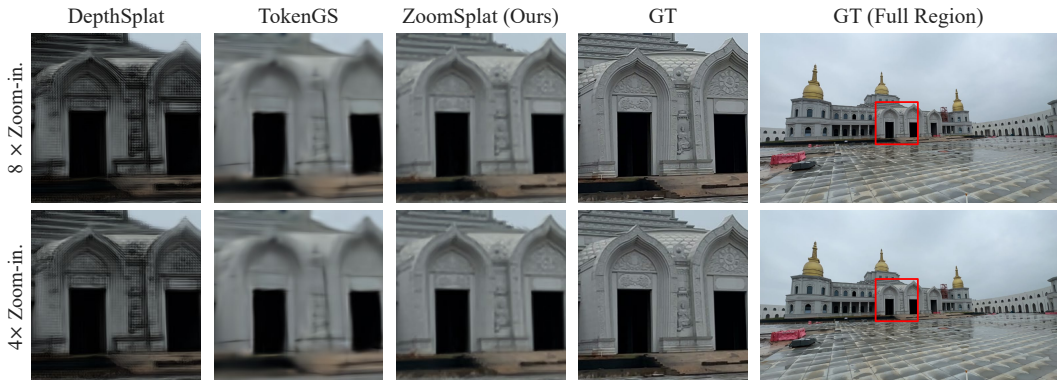


Figure 1: **Detail-preserving anti-aliasing for feed-forward 3D Gaussian splatting.** DepthSplat exhibits severe erosion and dark seams at increased sampling rates (8×, top). TokenGS suffers from blurry reconstructions and highly degenerate (thin) Gaussians. Our ZoomSplat suppresses erosion artifacts while better preserving scene detail.

ABSTRACT

Feed-forward 3D Gaussian Splatting (FF-3DGS) enables efficient sparse-view reconstruction and novel view synthesis, but these reconstructions can exhibit severe erosion and grid-like artifacts when rendered beyond the context-image sampling rates such as zoom-in operation. We propose ZoomSplat for FF-3DGS with detail-preserving reconstruction and smooth higher-resolution rendering from a fixed Gaussian representation. Although FF-3DGS models combine information from multiple input views, existing multi-view 3D frequency constraint governed by the largest sampling rate of individual views does not account for their combined observations and can impose excessive smoothing. To address this, our ZoomSplat integrates *multiview-aware 3D filtering* into Gaussian prediction *for the first time*, through an aggregated multiview frequency bound that permits finer Gaussian scales in regions supported by multiple observations. Explicit filter-size conditioning then enables the FF-3DGS models to account explicitly for the resulting filter sizes. Finally, an Opaque Scene Prior (OSP) combines termination-conditioned color normalization with transmittance regularization to compensate for residual transparency and to encourage opaque coverage. On DL3DV with six context views, ZoomSplat improves average 1.4 dB PSNR across $1\times$, $2\times$, $4\times$, and $8\times$ rendering scales over TokenGS that is the second best FF-3DGS model and lifts average 3.2 dB PSNR over the baseline of DepthSplat.

1 INTRODUCTION

3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) has become a leading representation for novel view synthesis (NVS) from multiview images, combining real-time rendering with photorealistic fidelity. Whereas per-scene 3DGS methods (Guédon & Lepetit, 2024; Lu et al., 2024; Fan et al.,

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.

2024) optimize a separate representation for each scene, feed-forward 3DGS models (Charatan et al., 2024; Szymanowicz et al., 2024; Chen et al., 2024a;b; Xu et al., 2025; Kang et al., 2026; Zhang et al., 2024; Tang et al., 2024) train a single network to directly infer the Gaussians of an unseen scene from sparse context views, eliminating costly per-scene optimization. In practice, the reconstructed scene is meant to be explored freely: moving the camera closer or rendering for a high-resolution display may raise the sampling rates well beyond those of the context images. A reconstructed representation is therefore only as useful as it remains faithful under such closer viewing.

Increasing the rendering sampling rate can expose erosion and high-frequency artifacts that are less apparent at the context sampling rate. Mip-Splatting(Yu et al., 2024) addresses this issue through sampling-rate-based 3D smoothing and 2D Mip filtering. AA-2DGS(Younes & Boukhayma, 2026) adapts this approach to planar primitives using flat smoothing and object-space Mip filtering. However, both use a maximum-rate constraint that does not account for cumulative observation density across views and may therefore oversmooth details supported by complementary context observations in multiview feed-forward reconstruction. SurfSplat(He et al., 2026) instead promotes surface continuity and multiview contribution through a surface continuity prior and forced alpha blending. However, it does not explicitly constrain primitive scales based on observation sampling rates, and our evaluation shows residual artifacts at increased rendering sampling rates.

To address these limitations, we propose ZoomSplat, a feed-forward 3D Gaussian reconstruction framework for faithful novel-view synthesis at sampling rates beyond those of the input context views. We first introduce multiview-adaptive sampling-rate-based 3D filtering (MAS3DF), which aggregates the sampling rates of context views observing each Gaussian to determine a local filter size, suppressing artifacts while preserving detail supported by multiple views. However, controlling Gaussian scales alone does not ensure that their opacity contributions accumulate sufficiently to represent opaque surfaces, potentially leaving dark seams against the background. We therefore complement MAS3DF with an opaque-scene prior (OSP) that compensates for color attenuation and encourages opaque coverage. Moreover, applying the filter only after Gaussian prediction leaves the attribute head without explicit knowledge of the smoothing that will modify its outputs. We therefore encode the local filter size and incorporate it into the Gaussian prediction network as an additional conditioning input, enabling filter-aware attribute prediction. Experimental results show that ZoomSplat outperforms TokenGS, the second-best feed-forward method, by 1.38 dB in average PSNR on DL3DV across sampling rates of $2\times$, $4\times$, and $8\times$. It also demonstrates zero-shot generalization to the Train and Truck scenes of Tanks and Temples, exceeding TokenGS by 0.91 dB on average across $2\times$ and $4\times$ sampling rates without fine-tuning. Our contributions are as follows:

- We propose ZoomSplat, a feed-forward 3D Gaussian reconstruction framework for faithful novel-view synthesis at sampling rates beyond those of the input context views.
- We introduce multiview-adaptive sampling-rate-based 3D filtering with an opaque-scene prior and filter-size conditioning to suppress artifacts while preserving scene detail.
- ZoomSplat surpasses TokenGS, which is a very recent feed-forward method, by 1.38 dB on DL3DV and 0.91 dB in zero-shot Tanks and Temples in average PSNR across the evaluated sampling rates above $1\times$.

2 Related Work

2.1 Per-Scene Optimization-Based Novel View Synthesis

Neural Radiance Fields (NeRFs) (Mildenhall et al., 2021) established photorealistic novel view synthesis (NVS) from posed images and inspired numerous extensions (Bui et al., 2025; Cao & Johnson, 2023; Chen et al., 2022; Fridovich-Keil et al., 2022; 2023; Garbin et al., 2021; Hu et al., 2022; Müller et al., 2022; Yu et al., 2021). 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) instead represents a scene as an explicit set of anisotropic 3D Gaussian primitives, each parameterized by a center, covariance, opacity, and view-dependent color, which are projected to screen space and alpha-composited by tile-based rasterization. As the entire rendering pipeline is differentiable, the primitives are optimized directly under photometric supervision, yielding photorealistic real-time rendering and spawning a large body of follow-up work (Guédon & Lepetit, 2024; Yang et al., 2024b; Wu et al., 2024; Lu et al., 2024; Fan et al., 2024; Keetha et al., 2024; Yan et al., 2024; Bui

et al., 2026; Park et al., 2025). Both approaches, however, rely on per-scene optimization: reconstruction is slow, does not transfer to unseen scenes, and overfits under sparse input views.

2.2 Feed-Forward 3D Gaussian Splatting

Feed-forward 3DGS models learn scene priors from multiple scenes to predict Gaussian representations of unseen scenes in a single forward pass (Charatan et al., 2024; Szymanowicz et al., 2024; Chen et al., 2024b; Park et al., 2026; Ye et al., 2025; Huang & Mikolajczyk, 2025a;b). MVSplat (Chen et al., 2024a) predicts pixel-aligned Gaussians from multiview depth estimates, while DepthSplat (Xu et al., 2025) incorporates monocular depth priors to improve reconstruction robustness. However, these representations can exhibit erosion and grid-like artifacts when rendered above the context-image resolution. SurfSplat (He et al., 2026) reconstructs surfaces using 2D Gaussian primitives with surface continuity priors and forced alpha blending. However, we found that SurfSplat still exhibits rendering artifacts at large magnifications. SubSplat (Lee et al., 2026) addresses rendering at increased sampling rates by subdividing pixel-aligned Gaussians into finer primitives. This increases the primitive count according to the target resolutions. In contrast, our ZoomSplat applies multiview-derived 3D filtering and an opaque-scene prior to suppress rendering artifacts at increased sampling rates while retaining a fixed Gaussian representation whose primitive count is determined by the context images.

2.3 Anti-Aliased Novel View Synthesis

Anti-aliasing in novel view synthesis addresses artifacts caused by changes in image sampling rates due to rendering resolution or camera configuration. Mip-NeRF and its extensions (Barron et al., 2021; 2022) integrate positional encodings over pixel footprints to reduce aliasing. Mip-Splatting (Yu et al., 2024) combines 3D Gaussian smoothing with 2D screen-space Mip filtering, setting the 3D filter size according to the maximum sampling rate among observing views. AA-2DGS (Younes & Boukhayma, 2026) adapts these filters to planar 2D Gaussian primitives (Huang et al., 2024). While these methods focus on per-scene optimization, our ZoomSplat integrates 3D filtering into feed-forward reconstruction and determines its Gaussian filter sizes by aggregating sampling rates across context views rather than selecting their maximum.

3 Proposed Method: ZoomSplat

3.1 Overview

ZoomSplat aims to preserve scene details and suppress artifacts when rendering at sampling rates higher than those of the context observations. Given N posed context images $\{\mathbf{I}^i\}_{i=1}^N$ of size $H \times W$ and their camera projection matrices $\{\mathbf{P}^i\}_{i=1}^N$, it predicts a 3D Gaussian representation in a single forward pass. The representation contains one Gaussian per context pixel, yielding a fixed set of primitives $\{\mathcal{G}_j\}_{j=1}^{NHW}$ that is reused across target sampling rates.

Figure 2 illustrates our ZoomSplat architecture. We build our ZoomSplat on the DepthSplat (Xu et al., 2025) backbone for multiview feature extraction and depth prediction. Our multiview-adaptive sampling-rate-based 3D filtering (MAS3DF) determines local filter sizes $\{s'_j\}$ for Gaussian primitives $\{\mathcal{G}_j\}$ from context observations to preserve detail while suppressing artifacts, complemented by an opaque-scene prior (OSP) that reduces color darkening and encourages opaque coverage (Section 3.2). We also use the resulting filter sizes $\{s'_j\}$ to condition the Gaussian attribute prediction, allowing the model to account for the filtering applied to its outputs (Section 3.3).

3.2 Multiview-Adaptive 3D Filtering for Rendering at Higher Sampling Rates

Increasing the rendering sampling rate in 3D Gaussian Splatting may entail erosion and high-frequency artifacts. To address these issues, Mip-Splatting (Yu et al., 2024) introduces a 3D smoothing filter that constrains Gaussian scales according to the maximum sampling rate among observing views. While this constraint suppresses artifacts caused by excessively small Gaussians, excessive smoothing may blur fine details. Moreover, in the feed-forward setting, the model jointly reconstructs a Gaussian representation from multiple context views, motivating a scale constraint that

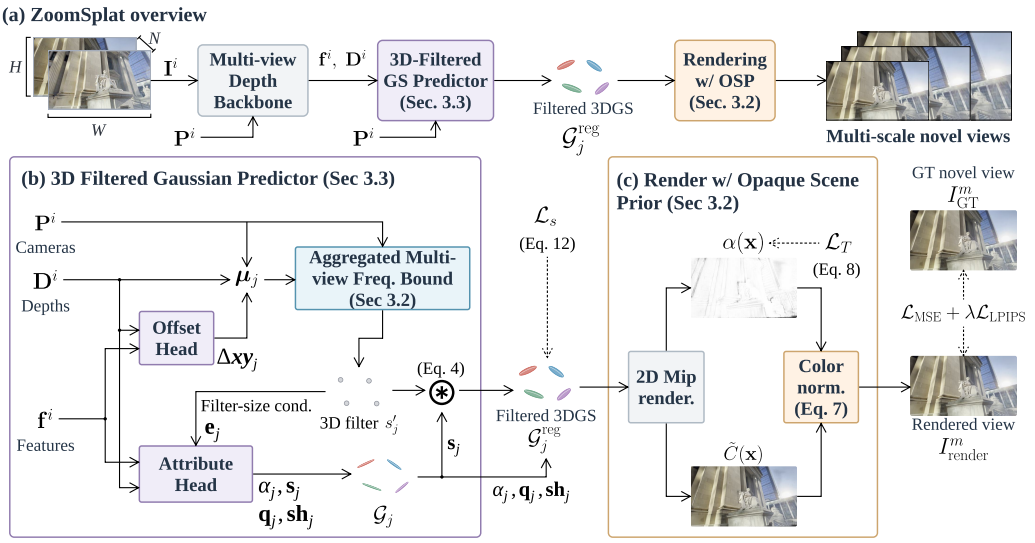


Figure 2: **Architecture of ZoomSplat.** (a) Given posed context images, ZoomSplat predicts a fixed Gaussian representation for novel-view rendering at multiple scales. (c) Decoupled heads first predict Gaussian centers, then condition attribute prediction on the filter size determined by our aggregated multiview frequency bound. Applying the resulting 3D filters produces band-limited Gaussians. (b) Rendering combines 2D Mip filtering with accumulated-alpha color normalization and transmittance regularization under the Opaque Scene Prior.

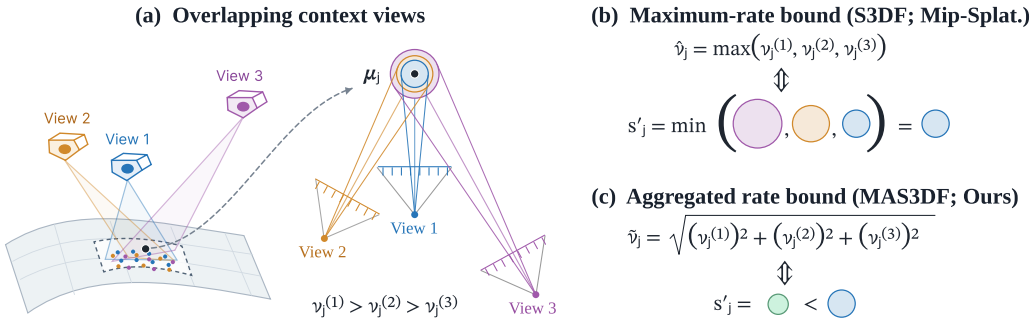


Figure 3: **Comparison of S3DF and MAS3DF.** (a) Three context views observe the same Gaussian center $\boldsymbol{\mu}_j$, with View 1 providing the highest sampling rate. Colored circles illustrate the per-view filter sizes, corresponding to the per-view sampling rates. (b) S3DF selects the smallest per-view filter, determined by the maximum sampling rate. (c) MAS3DF aggregates all three rates through an L_2 norm, yielding a smaller filter. Circles depict filter scales, not final Gaussian extents.

reflects their context observations. We therefore determine a *spatially-varying scale constraint from these multiview observations* to balance artifact suppression and detail preservation when rendering at higher sampling rates.

Conventional sampling-rate-based 3D filtering (S3DF). To motivate our multiview-adaptive filtering, we first revisit the sampling-rate-based 3D filtering used in Mip-Splatting (Yu et al., 2024). For a Gaussian center $\boldsymbol{\mu}_j$, let $d_j^{(i)}$ denote its depth in context camera i and $f^{(i)}$ be the horizontal focal length in pixels. The corresponding linear sampling rate is approximated by $\nu_j^{(i)} = f^{(i)}/d_j^{(i)}$. Its reciprocal represents the world-space pixel spacing at that depth. The maximum-rate rule in Mip-Splatting (Yu et al., 2024) determines the filter scale from

$$\hat{\nu}_j = \max_{i=1,\dots,N} \left\{ \mathbf{1}^{(i)}(\boldsymbol{\mu}_j) \nu_j^{(i)} \right\}, \tag{1}$$

with filter standard deviation $\sigma_s/\hat{\nu}_j$. Here, $\mathbf{1}^{(i)}(\boldsymbol{\mu}_j)$ checks positive depth and projection within an expanded image boundary without accounting for occlusion. It should be noted in Eq. (1) that the

maximum sampling rate is determined by selecting a single context view, not by considering the combined observation that multiple context views contribute.

This process is illustrated in Figure 3(a) and (b). Out of the three context views observing the same Gaussian center $\boldsymbol{\mu}_j$, View 1 has the highest sampling rate and therefore implies the smallest filter size, shown in blue. S3DF selects this smallest per-view filter (Figure 3(b)); the lower sampling rates of Views 2 and 3 do not affect the selected size.

Proposed multiview-adaptive sampling-rate-based 3D filtering (MAS3DF). To address the limitation of S3DF which only selects a single per-view rate, our MAS3DF aggregates contributions from all observing context views to compute the effective sampling rate, and uses its inverse to determine a multiview-adaptive 3D filter size for each Gaussian. Compared with the maximum-rate rule in Eq. (1), our MAS3DF can yield smaller-sized filters in regions where multiple views contribute. Applying these filters to the predicted Gaussians relaxes the minimum-scale constraint, allowing finer details to be reconstructed. We motivate this aggregation through a local observation-density approximation.

At each Gaussian center, we estimate the observation density contributed by each context view by assuming a locally image-parallel surface with pixel spacing $d_j^{(i)}/f^{(i)}$ in horizontal and vertical pixel directions. The corresponding pixel footprint has an area proportional to $(d_j^{(i)}/f^{(i)})^2$, whose reciprocal gives the density contribution $\rho_j^{(i)} = \mathbf{1}^{(i)}(\boldsymbol{\mu}_j)(\nu_j^{(i)})^2$. We sum these contributions and take the square root on the aggregated density contributions across all N context views to define an effective sampling rate at the location of the Gaussian primitive $\mathcal{G}_j$:

$$\tilde{\nu}_j = \sqrt{\textstyle\sum_{i=1}^{N} \rho_j^{(i)}}. \tag{2}$$

We set the isotropic 3D filter's standard deviation as the 3D filter size to

$$s'_j = \sigma_s / \tilde{\nu}_j, \tag{3}$$

where σ_s controls filtering strength and $\sigma_s^2 = 0.2$. For fixed σ_s, $\tilde{\nu}_j \geq \hat{\nu}_j$ yields a 3D filter size equal to or smaller than that of the maximum-rate filter determined by Eq. (1), permitting finer Gaussian size (Figure 3). If no view passes the visibility check, we use the Gaussian's source-view depth and the largest context focal length to compute the filter size. For a predicted covariance $\boldsymbol{\Sigma}_j$ with principal scales $s_{j,k}$, we apply the filter through covariance addition to obtain the covariance $\boldsymbol{\Sigma}_j^{\text{reg}}$ and scale $s_{j,k}^{\text{reg}}$ of the final filtered Gaussian primitive $\mathcal{G}_j^{\text{reg}}$ as:

$$\boldsymbol{\Sigma}_j^{\text{reg}} = \boldsymbol{\Sigma}_j + (s'_j)^2 \mathbf{I}, \qquad s_{j,k}^{\text{reg}} = \sqrt{s_{j,k}^2 + (s'_j)^2}, \tag{4}$$

where $k \in \{x, y, z\}$ denotes the principal axes of the Gaussian primitive.

In the example of Figure 3(c), combining the three squared sampling rates and taking their square root gives an aggregate rate greater than that of View 1 alone. Its inverse determines the smaller filter shown in green, relaxing the minimum Gaussian scale imposed by S3DF. observation-density approximation.

The filtered scales $s_{j,k}^{\text{reg}}$ are bounded below by the filter size s'_j, with additional variance learned by the model. The predicted opacity remains unchanged. The aggregated density in Eq. (2) motivates the scale constraint rather than guaranteeing recoverable scene frequencies, since observations may be redundant and surface orientation and sample arrangement are not explicitly modeled.

Opaque-scene prior (OSP). Even with appropriately constrained scales, Gaussians may accumulate insufficient opacity to represent opaque surfaces, allowing the black background to darken rendered colors. We therefore introduce an OSP that combines accumulated-alpha color normalization to compensate for this attenuation with a transmittance regularization to encourage opaque coverage. For depth-ordered Gaussians $\mathcal{N}(\mathbf{x})$ at pixel $\mathbf{x}$, let $a_j(\mathbf{x})$ be the effective opacity for $\mathcal{G}_j$ after 2D Mip filtering (Yu et al., 2024) and $\mathbf{c}_j$ be its view-dependent color. The compositing weight is

$$w_j(\mathbf{x}) = a_j(\mathbf{x}) \prod_{\substack{k \in \mathcal{N}(\mathbf{x}) \\ k \prec j}} \left(1 - a_k(\mathbf{x})\right), \tag{5}$$

where $k \prec j$ denotes depth precedence. Following SurfSplat (He et al., 2026), we compute accumulated alpha $\alpha(\mathbf{x})$ and color $\tilde{C}(\mathbf{x})$ as:

$$\alpha(\mathbf{x}) = \sum\nolimits_{j \in \mathcal{N}(\mathbf{x})} w_j(\mathbf{x}), \qquad \tilde{C}(\mathbf{x}) = \sum\nolimits_{j \in \mathcal{N}(\mathbf{x})} w_j(\mathbf{x}) \mathbf{c}_j, \tag{6}$$

Then, we obtain normalized color $C_{\mathrm{OSP}}(\mathbf{x})$ from $\tilde{C}(\mathbf{x})$ to compensate for attenuation as:

$$C_{\mathrm{OSP}}(\mathbf{x}) = \begin{cases} \tilde{C}(\mathbf{x}), & \alpha(\mathbf{x}) < \tau_\alpha, \\ \tilde{C}(\mathbf{x})/\alpha(\mathbf{x}), & \alpha(\mathbf{x}) \geq \tau_\alpha. \end{cases} \tag{7}$$

where the color normalization is performed only for $\alpha(\mathbf{x}) \geq \tau_\alpha$ to avoid amplifying negligible contributions in regions where there is insufficient support from reconstructed Gaussians. τ_α is empirically set to 0.1 for training and to 0.001 for evaluation in all experiments. We clamp RGB to $[0, 1]$ and retain sigmoid-bounded opacities without SurfSplat's additional cap. Since normalization alone is not sufficient to enforce opaque coverage, we penalize residual transmittance using accumulated alpha α_m so that the predicted Gaussians can produce more opaque scenes. For M target views, our transmittance regularization $\mathcal{L}_T$ is defined as:

$$\mathcal{L}_T = \frac{1}{M} \sum\nolimits_{m=1}^{M} \frac{1}{|\Omega_m|} \sum\nolimits_{\mathbf{x} \in \Omega_m} \left| 1 - \alpha_m(\mathbf{x}) \right|. \tag{8}$$

where Ω_m is the set of pixels in m-th target view. This encourages the accumulated alpha $\alpha_m(\mathbf{x})$ to be opaque across the scene without requiring each individual Gaussian to be opaque.

3.3 Filter-Size-Conditioned Gaussian Prediction

The scale composition in Eq. (4) enforces the multiview constraint after predicting Gaussian attributes $\{\mathcal{G}_j\}$. However, an unconditioned attribute head does not explicitly receive the filter sizes $\{s'_j\}$ that modify its outputs. Since $\{s'_j\}$ depend on the predicted centers $\boldsymbol{\mu}_j$ and sampling rates across context cameras, we compute $\{s'_j\}$ before predicting the remaining attributes and supply them as conditioning.

Decoupled center prediction. The filters must be evaluated at the Gaussian centers, including their learned image-plane offsets $\{\Delta xy_j\}$. We therefore separate offset prediction from the prediction of opacity α_j, scales $\mathbf{s}_j$, rotation $\mathbf{q}_j$, and appearance $\mathbf{sh}_j$. An offset head consumes the context images, predicted depths, matching confidences, and backbone features to predict two-dimensional offsets $\{\Delta xy_j\}$ bounded to half a pixel per axis. Unprojecting the offset pixels $\{(\mathbf{x}_j + \Delta xy_j)\}$ with their depths and source cameras gives the centers $\{\boldsymbol{\mu}_j\}$ for $\{\mathcal{G}_j\}$, where we compute filter sizes $\{s'_j\}$ using Eq. (3). This ordering resolves the dependency between center prediction, filter computation, and attribute prediction.

Filter-size encoding and attribute prediction. We encode the logarithm of s'_j using six sinusoidal frequencies:

$$\mathbf{e}_j = \left[\ell_j,\ \left\{ \sin(2\pi 2^k \ell_j), \cos(2\pi 2^k \ell_j) \right\}_{k=0}^{5} \right], \quad \ell_j = \log\left(\max(s'_j, \epsilon) \right), \quad \epsilon = 10^{-10}. \tag{9}$$

The logarithm converts multiplicative scale changes into additive changes, while the sinusoidal channels provide multiple frequencies for conditioning. Retaining the unencoded logarithm yields a 13-dimensional encoding. We arrange these encodings on the source-image grids and concatenate them with the inputs to the Gaussian attribute predictor head. These modules predict opacity α_j, pre-filter scales $\mathbf{s}_j$, rotation $\mathbf{q}_j$, and spherical harmonic coefficients $\mathbf{sh}_j$. Applying Eq. (4) to the Attribute head's outputs yields the filtered Gaussians as:

$$\mathcal{G}_j^{\mathrm{reg}} = \left(\boldsymbol{\mu}_j, \alpha_j, \mathbf{s}_j^{\mathrm{reg}}, \mathbf{q}_j, \mathbf{sh}_j \right), \qquad j = 1, \ldots, NHW. \tag{10}$$

3.4 Optimization

Scale regularization loss. To discourage excessive scale growth, we penalize activated scales before clamping. For the scale logit $z_{j,k}$ of Gaussian $\mathcal{G}_j$ along axis $k \in \{x, y, z\}$, the pre-filter scale is

$$u_{j,k} = \mathrm{softplus}(z_{j,k} - 4), \qquad s_{j,k} = \mathrm{clamp}\left(u_{j,k}, s_{\min}, s_{\max} \right), \tag{11}$$

where $s_{\max} = 3$ in reconstruction coordinates following DepthSplat (Xu et al., 2025). With $J = H \cdot W \cdot N$, we define a scale regularization loss as:

$$\mathcal{L}_s = \frac{1}{3J} \sum\nolimits_{j=1}^{J} \sum\nolimits_{k \in \{x,y,z\}} \left[\max\left(u_{j,k} - s_{\max}, 0 \right) \right]^2. \tag{12}$$

Penalizing $u_{j,k}$ preserves a restoring gradient above the clamp. The penalty is averaged over all Gaussian axes before 3D filtering, without opacity weighting.

Training objective. At a single target resolution, we supervise each depth stage r with mean target-view MSE and LPIPS (Zhang et al., 2018) reconstruction losses and the OSP transmittance loss $\mathcal{L}_T^{(r)}$. The total objective is

$$\mathcal{L} = \sum\nolimits_{r=1}^{R} \omega_r \left(\mathcal{L}_{\text{MSE}}^{(r)} + \lambda \, \mathcal{L}_{\text{LPIPS}}^{(r)} + \lambda_T \, \mathcal{L}_T^{(r)} \right) + \lambda_s \, \mathcal{L}_s, \tag{13}$$

where stages are ordered from intermediate to final and $\omega_r = 0.9^{R-r}$. We set $\lambda = 0.05$ and $\lambda_s = 0.03$ throughout training, and use $\lambda_T = 0.001$ for RE10K pretraining and $\lambda_T = 0.005$ for DL3DV training.

4 Experiments

Datasets and metrics. We train on RE10K(Zhou et al., 2018) with approximately 66K scenes, and DL3DV(Ling et al., 2024) with total 9,568 scenes. We evaluate on 139 held-out DL3DV scenes (no overlap with training scenes) with high-resolution (4K) ground-truth images. Each scene representation is reconstructed once from 256×448 context images and reused for rendering at $1\times$, $2\times$, $4\times$, and $8\times$ scales, following the single-scale reconstruction and multi-scale testing protocol established in per-scene works (Yu et al., 2024; Younes & Boukhayma, 2026). In addition, we evaluate cross-dataset generalization from DL3DV on Tanks&Temples (Knapitsch et al., 2017) that is a real-world indoor and outdoor 3D reconstruction benchmark. Here, we use two scenes, 'Train' and 'Truck', reconstructing each from 256×448 context images and rendering at $1\times$, $2\times$, and $4\times$ scales without fine-tuning, following a common protocol in 3DGS literature (Kerbl et al., 2023; Ziwen et al., 2025). We report PSNR, SSIM, and LPIPS against ground-truth images at each evaluated sampling rate.

Implementation details. We build ZoomSplat on the DepthSplat backbone. For comparisons with existing methods, we first pre-train ZoomSplat on RE10K for 300k steps with an effective batch size of 16, sampling two context views and four target views. We then fine-tune the full model on DL3DV for 200k steps with an effective batch size of two, sampling two to six context views and eight target views. All training images have 256×448 resolution. For controlled ablations, we instead initialize the depth predictor from the released RE10K-pretrained DepthSplat Base checkpoint and train the Gaussian prediction heads from scratch, using the same DL3DV training schedule.

4.1 Main Results

Results on DL3DV. We compare our ZoomSplat with (i) per-scene methods 3DGS, Mip-Splatting, and AA-2DGS; (ii) feed-forward methods MVSplat (Chen et al., 2024a), SurfSplat (He et al., 2026), DepthSplat (Xu et al., 2025), IDESplat (Long et al., 2026), and TokenGS (Ren et al., 2026). We additionally compare with (iii) two render-and-upsample cascade methods, where novel views are rendered at the native ($1\times$) sampling rate from the baseline DepthSplat, and are compared for bilinear and bicubic interpolations by $2\times$, $4\times$ and $8\times$. For fair comparisons, we fine-tune MVSplat (Chen et al., 2024a) and SurfSplat (He et al., 2026) on DL3DV (Ling et al., 2024), and use released DL3DV checkpoints for the other feed-forward methods such as original DepthSplat (Xu et al., 2025) and IDESplat (Long et al., 2026). Table 1 and Figure 4 show the quantitative and qualitative comparison results on DL3DV dataset, respectively. As shown in Table 1, MVSplat, DepthSplat, and IDESplat exhibit substantial drops in rendering quality as the sampling rate increases. DepthSplat and IDESplat also show degraded subjective qualities as shown in Figures 4 (See more in Appendix D). It is noted that in Figure 4 SurfSplat exhibits severe hole artifacts at $8\times$ rendering. Our ZoomSplat can produce visually coherent novel views even under the extreme $8\times$ zoom-in.

Cross-dataset generalization results on Tanks&Temples. Table 2 compares cross-dataset generalization results on Tanks&Temples benchmark for all the feed-forward methods under comparison without fine-tuning or per-scene optimization. As shown, our ZoomSplat achieves superior adaptability across all sampling rates (see more results in Appendix D). Averaged over $2\times$ and $4\times$, its PSNR exceeds TokenGS, the recent feed-forward method, by 0.91 dB. At $4\times$, it also exceeds the second-best method, DepthSplat with bicubic upsampling ('DS+Bic.'), by 0.66 dB.

Table 1: **Results on in-distribution DL3DV dataset.** We report PSNR↑ / SSIM↑ / LPIPS↓. "PS" refers to per-scene optimization-based methods, "FF" refers to feed-forward methods, and "CC" refers to render-then-upsample cascade methods. * refers to models re-trained on DL3DV for fair comparison. DepthSplat 1× rendering followed by Bilinear and Bicubic upsampling are denoted as DS+Bil. and DS+Bic., respectively. Best and second-best are shown in **red** and blue, respectively.

Types	Methods	1×	2×	4×	8×	Avg.
PS	3DGS	22.56 / 0.632 / 0.318	18.36 / 0.574 / 0.445	16.80 / 0.546 / 0.536	15.94 / 0.586 / 0.568	18.41 / 0.585 / 0.467
	Mip-Splat.	22.18 / 0.611 / 0.341	19.44 / 0.600 / 0.430	18.77 / 0.617 / 0.502	18.33 / 0.667 / 0.553	19.68 / 0.624 / 0.457
	AA-2DGS	21.91 / 0.603 / 0.337	19.14 / 0.585 / 0.436	18.46 / 0.592 / 0.511	18.01 / 0.634 / 0.550	19.38 / 0.604 / 0.458
CC	DS+Bil.	24.16 / 0.819 / 0.146	22.70 / 0.734 / 0.319	21.88 / 0.701 / 0.427	21.29 / 0.725 / 0.508	22.51 / 0.745 / 0.350
	DS+Bic.	24.16 / 0.819 / 0.146	22.83 / 0.746 / 0.296	21.99 / 0.708 / 0.416	21.39 / 0.726 / 0.505	22.59 / 0.750 / 0.340
FF	MVSplat*	23.03 / 0.776 / 0.187	20.86 / 0.704 / 0.311	19.07 / 0.652 / 0.435	18.25 / 0.672 / 0.500	20.30 / 0.701 / 0.358
	DepthSplat	24.16 / 0.819 / 0.146	20.97 / 0.727 / 0.283	18.53 / 0.641 / 0.436	17.42 / 0.647 / 0.503	20.27 / 0.709 / 0.342
	SurfSplat*	24.13 / 0.817 / 0.151	22.38 / 0.731 / 0.322	20.04 / 0.601 / 0.524	17.42 / 0.536 / 0.588	20.99 / 0.671 / 0.397
	IDESplat	24.67 / 0.830 / 0.145	21.01 / 0.711 / 0.331	17.73 / 0.586 / 0.491	16.33 / 0.587 / 0.542	19.94 / 0.679 / 0.377
	TokenGS	23.63 / 0.755 / 0.315	22.30 / 0.679 / 0.445	21.47 / 0.668 / 0.513	20.91 / 0.711 / 0.546	22.07 / 0.703 / 0.455
	Ours	**25.22** / **0.844** / **0.129**	**23.81** / **0.779** / **0.245**	**22.85** / **0.735** / **0.368**	**22.16** / **0.739** / **0.458**	**23.51** / **0.775** / **0.300**

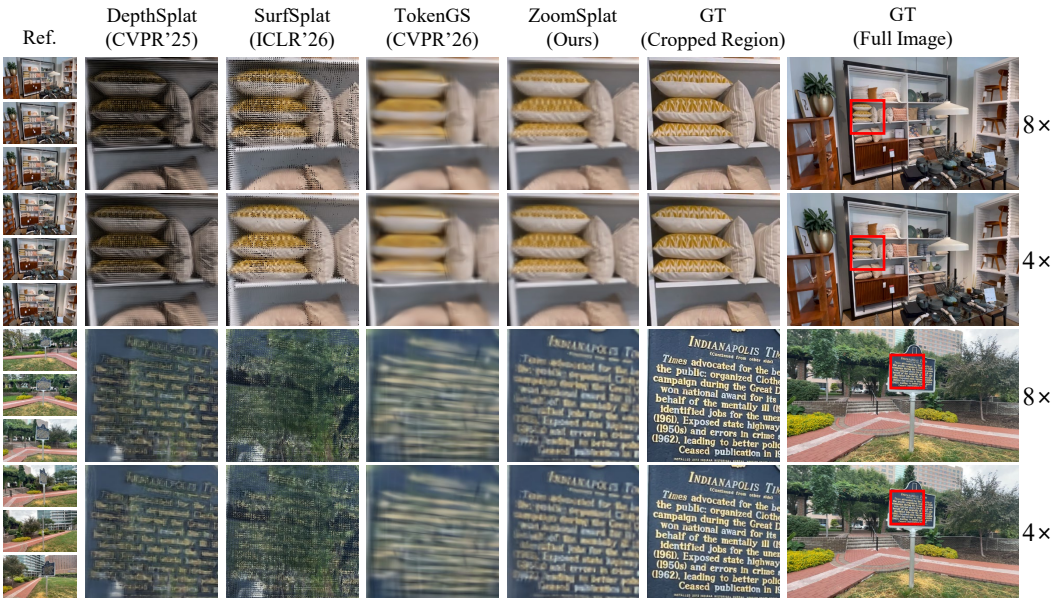


Figure 4: **Qualitative comparisons on in-distribution DL3DV dataset.** Our ZoomSplat preserves scene details while suppressing unnatural high-frequency artifacts even under extreme zoom-in. Best viewed zoomed-in.

4.2 Ablation Studies

Table 3 reports ablation results on DL3DV with six context views, showing the effects of MAS3DF, OSP, and filter-size conditioning across sampling rates. The baseline of variant (a) suffers from severe erosion artifacts under zoom-ins. The variant (b) with Mip-Splatting's 2D Mip and 3D post-filters (S3DF) mitigates most severe artifacts, but grid artifacts remain visible. The variant (c) with our MAS3DF achieves about 0.2 dB PSNR improvement over variant (b) at 1× sampling rate. The variant (d) with our opaque scene prior (OSP) further improves reconstruction metrics at every evaluated sampling rate. The variant (e) adds 3D filter-size conditioning on top of variant (d) and also achieves a modest further improvement. We present additional ablations in Appendix B.2.

Table 2: **Zero-shot generalization from DL3DV to Tanks&Temples.** We evaluate methods trained on DL3DV on Tanks&Temples without fine-tuning. Gaussians are reconstructed from 6 context views at 256×448 resolution and evaluated at $1\times$, $2\times$, and $4\times$ scales to simulate zoom-in effects.

Types	Methods	1×	2×	4×	Avg.
CC	DS+Bil.	22.10 / 0.804 / 0.127	20.79 / 0.691 / 0.307	20.13 / 0.640 / 0.426	21.01 / 0.711 / 0.286
	DS+Bic.	22.10 / 0.804 / 0.127	20.87 / 0.705 / 0.281	20.20 / 0.648 / 0.413	21.06 / 0.719 / 0.274
FF	MVSplat*	19.81 / 0.679 / 0.205	17.62 / 0.566 / 0.355	15.69 / 0.498 / 0.483	17.71 / 0.581 / 0.348
	DepthSplat	22.10 / 0.804 / 0.127	18.50 / 0.668 / 0.279	15.38 / 0.525 / 0.453	18.66 / 0.666 / 0.287
	SurfSplat*	21.63 / 0.793 / 0.146	20.43 / 0.704 / 0.273	19.30 / 0.611 / 0.429	20.45 / 0.702 / 0.282
	IDESplat	22.25 / 0.819 / 0.131	19.17 / 0.669 / 0.320	15.10 / 0.490 / 0.498	18.84 / 0.660 / 0.316
	TokenGS	21.98 / 0.747 / 0.294	20.70 / 0.643 / 0.436	19.99 / 0.609 / 0.517	20.89 / 0.666 / 0.416
	Ours	**22.97 / 0.830 / 0.118**	**21.64 / 0.742 / 0.235**	**20.86 / 0.676 / 0.365**	**21.82 / 0.749 / 0.239**

Table 3: **Ablation studies on DL3DV.** Gaussians are reconstructed with 6 context views at 256×448 resolution and evaluated at $1\times$, $2\times$, $4\times$, and $8\times$ scales to simulate zoom-in effects.

Variant	1×	2×	4×	8×	Avg.
(a) DepthSplat*	23.99 / 0.815 / 0.151	20.91 / 0.725 / 0.284	18.51 / 0.642 / 0.434	17.43 / 0.649 / 0.503	20.21 / 0.708 / 0.343
(b) (a) + S3DF	24.05 / 0.817 / 0.150	22.87 / 0.757 / 0.272	22.06 / 0.721 / 0.390	21.46 / 0.733 / 0.476	22.61 / 0.757 / 0.322
(c) (b) + MAS3DF	24.24 / 0.822 / 0.147	23.02 / 0.760 / 0.265	22.18 / 0.722 / 0.385	21.57 / 0.731 / 0.470	22.75 / 0.759 / 0.317
(d) (c) + OSP	24.84 / 0.833 / 0.136	23.48 / 0.768 / 0.253	22.57 / 0.728 / 0.375	21.91 / 0.735 / **0.462**	23.20 / 0.766 / **0.306**
(e) (d) + cond. (Ours)	**24.93 / 0.835 / 0.134**	**23.56 / 0.770 / 0.252**	**22.63 / 0.729 / 0.374**	**21.97 / 0.736 / 0.462**	**23.27 / 0.767 / 0.306**

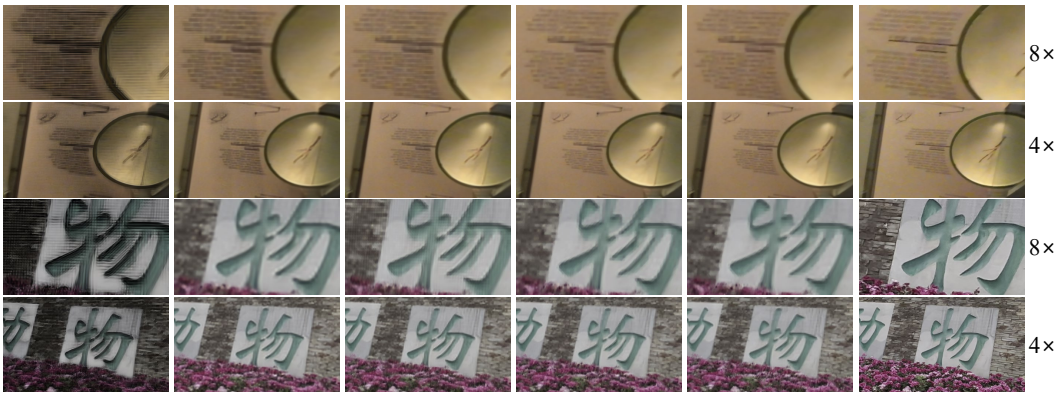


Figure 5: **Visual results for ablation studies.** Best viewed zoomed in.

5 Conclusion

We presented ZoomSplat, a feed-forward 3D Gaussian reconstruction framework for rendering beyond the context-image sampling rates. Our ZoomSplat incorporates a novel multiview-adaptive sampling-rate-based 3D filtering that aggregates observation densities across context views to determine local Gaussian scale constraints, permitting finer scales than the conventional maximum-rate rule. Filter-size conditioning makes these constraints explicitly available during Gaussian attribute prediction, while an opaque-scene prior combines accumulated-alpha color normalization with transmittance regularization to address residual transparency. Together, these components integrate sampling-aware scale control and opaque coverage into the reconstruction of a fixed Gaussian representation, without resolution-dependent primitive subdivision or higher-resolution training supervision.

AI use statement

In this work, we used generative AI tools for aiding or polishing writing. Specifically, we used ChatGPT to help improve the clarity and grammar of the text, as well as to assist with LaTeX formatting

of the manuscript. We have not used generative AI tools for retrieval and discovery, research ideation or execution, drafting sections of the paper, generation of synthetic datasets, or proving mathematical claims. We have not used generative AI tools for developing theoretical models or conceptual frameworks, or propose or refine hypotheses, core experiment designs, research ideation or execution, and the remaining required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. AI-assisted language suggestions were reviewed, revised, and incorporated by the authors only where appropriate. We take responsibility for the final contents of this work, including text, figures, and tables, and any other artifacts produced with the aid of generative AI.

REFERENCES

Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5835–5844. IEEE, 2021.

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5460–5469. IEEE, 2022.

Minh-Quan Viet Bui, Jongmin Park, Jihyong Oh, and Munchurl Kim. Moblurf: Motion deblurring neural radiance fields for blurry monocular video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(9):7752–7770, 2025.

Minh-Quan Viet Bui, Jongmin Park, Juan Luis Gonzalez, Jaeho Moon, Jihyong Oh, and Munchurl Kim. Mobgs: Motion deblurring dynamic 3d gaussian splatting for blurry monocular video. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2480–2489, 2026.

Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 130–141. IEEE, 2023.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19457–19467. IEEE, 2024.

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European conference on computer vision*, pp. 333–350. Springer, 2022.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European conference on computer vision*, pp. 370–386. Springer, 2024a.

Yuedong Chen, Chuanxia Zheng, Haofei Xu, Bohan Zhuang, Andrea Vedaldi, Tat-Jen Cham, and Jianfei Cai. Mvsplat360: Feed-forward 360 scene synthesis from sparse views. *Advances in Neural Information Processing Systems*, 37:107064–107086, 2024b.

Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *Advances in neural information processing systems*, 37:140138–140158, 2024.

Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5491–5500. IEEE, 2022.

Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12479–12488. IEEE, 2023.

Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14326–14335. IEEE, 2021.

Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5354–5363. IEEE, 2024.

Bing He, Jingnan Gao, Yunuo Chen, Ning Cao, Gang Chen, Zhengxue Cheng, Li Song, and Wenjun Zhang. Surfsplat: Conquering feedforward 2d gaussian splatting with surface continuity priors. *arXiv preprint arXiv:2602.02000*, 2026.

Tao Hu, Shu Liu, Yilun Chen, Tiancheng Shen, and Jiaya Jia. Efficientnerf-efficient neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12892–12901. IEEE, 2022.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pp. 1–11, 2024.

Ranran Huang and Krystian Mikolajczyk. No pose at all: Self-supervised pose-free 3d gaussian splatting from sparse views. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27947–27957. IEEE, 2025a.

Ranran Huang and Krystian Mikolajczyk. Spfsplatv2: Efficient self-supervised pose-free 3d gaussian splatting from sparse views. *arXiv preprint arXiv:2509.17246*, 2025b.

Gyeongjin Kang, Seungtae Nam, Seungkwon Yang, Xiangyu Sun, Sameh Khamis, Abdelrahman Mohamed, and Eunbyung Park. ilrm: An iterative large 3d reconstruction model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37332–37342, 2026.

Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatam: Splat, track & map 3d gaussians for dense rgb-d slam. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21357–21366. IEEE, 2024.

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.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

Jiun Lee, Jaekwang Kim, and Sangmin Lee. Subsplat: High-resolution pixel-aligned 3dgs via subpixel gaussian reparameterization. *arXiv preprint arXiv:2607.20813*, 2026.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22160–22169. IEEE, 2024.

Wei Long, Haifeng Wu, Shiyin Jiang, Jinhua Zhang, Xinchun Ji, and Shuhang Gu. Idesplat: Iterative depth probability estimation for generalizable 3d gaussian splatting. *arXiv preprint arXiv:2601.03824*, 2026.

Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20654–20664. IEEE, 2024.

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.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

Jongmin Park, Minh-Quan Viet Bui, Juan Luis Gonzalez Bello, Jaeho Moon, Jihyong Oh, and Munchurl Kim. Splinegs: Robust motion-adaptive spline for real-time dynamic 3d gaussians from monocular video. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26866–26875. IEEE, 2025.

Jongmin Park, Minh-Quan Viet Bui, Juan Luis Gonzalez, Jaeho Moon, Jihyong Oh, and Munchurl Kim. Ecosplat: Efficiency-controllable feed-forward 3d gaussian splatting from multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26010–26020, 2026.

Jiawei Ren, Michal Jan Tyszkiewicz, Jiahui Huang, and Zan Gojcic. Tokengs: Decoupling 3d gaussian prediction from pixels with learnable tokens. *arXiv preprint arXiv:2604.15239*, 2026.

Stanislaw Szymanowicz, Chrisitian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10208–10217, 2024.

Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pp. 1–18. Springer, 2024.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20310–20320. IEEE, 2024.

Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13941–13958, 2023.

Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16453–16463. IEEE, 2025.

Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19595–19604. IEEE, 2024.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in neural information processing systems*, 37:21875–21911, 2024a.

Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20331–20341. IEEE, 2024b.

Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. In *International Conference on Learning Representations*, volume 2025, pp. 54009–54033, 2025.

Mae Younes and Adnane Boukhayma. Anti-aliased 2d gaussian splatting. *Advances in Neural Information Processing Systems*, 38:9767–9792, 2026.

Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5732–5741. IEEE, 2021.

Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19447–19456. IEEE, 2024.

Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision*, pp. 1–19. Springer, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

Chen Ziwen, Hao Tan, Kai Zhang, Sai Bi, Fujun Luan, Yicong Hong, Li Fuxin, and Zexiang Xu. Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4349–4359. IEEE, 2025.

A More Implementation Details

A.1 Architecture and Training

Backbone and initialization. We use the DepthSplat Base architecture with a ViT-B monocular branch, two feature-matching scales, and a coarsest feature resolution of one eighth of the input resolution. Our full model initializes the multi-view feature-matching branch of the backbone weights from UniMatch(Xu et al., 2023), and the monocular branch from Depth Anything V2(Yang et al., 2024a). For ablation studies, we initialize the depth predictor from the released DepthSplat checkpoint trained on RE10K with two 256×448 context views, while the remaining parameters in the Gaussian attribute predictor are randomly initialized.

RE10K pretraining. For comparisons with existing methods, we pretrain ZoomSplat on RE10K(Zhou et al., 2018) for 300k steps using AdamW with weight decay 0.01. The peak learning rates are 2×10^{-6} for the pretrained monocular backbone and 2×10^{-4} for the remaining parameters. We use a OneCycle learning-rate schedule with cosine annealing and an initial increasing phase occupying 1% of the schedule, approximately 3,000 steps. The effective batch size is 16. Each training example contains two context views and four sampled target views, all at 256×448 resolution. The LPIPS, transmittance, and scale penalty weights are 0.05, 0.001, and 0.03, respectively. The resulting full-model checkpoint initializes the subsequent DL3DV fine-tuning stage.

DL3DV training. We train on DL3DV for 200k steps using AdamW with weight decay 0.01. The peak learning rates are 2×10^{-6} for the pretrained monocular backbone and 2×10^{-4} for the remaining parameters. We use a OneCycle learning-rate schedule with cosine annealing and an initial increasing phase occupying 1% of the schedule, approximately 2,000 steps. The effective batch size is two. Each training example contains two to six context views and eight target views, all at 256×448 resolution. For the full model, the LPIPS, transmittance, and scale penalty weights are 0.05, 0.005, and 0.03, respectively. The ablation models retain the same base training recipe except for the explicitly varied components.

Filtering and rendering settings. We use L_2 norm to aggregate the visible per-view sampling rates. The coefficient under the square root in the 3D filter standard deviation is 0.2, following Mip-Splatting (Yu et al., 2024). Filter-size conditioning uses six sinusoidal frequencies applied to the logarithm of the filter size, together with the unencoded logarithm. We use gsplat's antialiased rasterization mode with $\epsilon_{2\mathrm{D}} = 0.1$ for 2D Mip filtering. The accumulated-alpha threshold for OSP color normalization is 0.1 during training and 0.001 during inference. Our full model does not apply an additional per-primitive opacity cap. The filter-size conditioning input is detached from gradient computation. Unlike Mip-Splatting, our 3D filtering modifies Gaussian covariances without rescaling their opacities.

A.2 Dataset Preparation and Evaluation

DL3DV. We evaluate on 139 test scenes with high-resolution ground-truth imagery. For each context-view setting, we reconstruct a single Gaussian representation from two, four, or six 256×448 images and reuse it across the four rendering scales. Six-view results are reported in the main paper, with two- and four-view results in Table 4.

Tanks and Temples. We use the *Train* and *Truck* scenes. Starting from 1080×1920 images, we estimate cameras using COLMAP and undistort the images. We then construct 1024×1792 crops

using subpixel shifts to center the camera principal points, updating the intrinsics accordingly. We use BasicSR's antialiased bicubic resizing to obtain 512×896 and 256×448 images. The models receive six context views at 256×448 and render at 256×448, 512×896, and 1024×1792. The DL3DV-trained checkpoints are evaluated without target-dataset fine-tuning. For the results in Table 2, we exclude all context frames from the target-view set.

Metrics. We report PSNR, SSIM, and LPIPS against the corresponding ground-truth images. For each metric, the reported average across scales assigns equal weight to each evaluated scale.

A.3 Baseline Configurations

Per-scene methods. We optimize 3DGS(Kerbl et al., 2023), Mip-Splatting(Yu et al., 2024), and AA-2DGS(Younes & Boukhayma, 2026) for 30k iterations from random initialization, retaining the remaining default settings of their respective implementations.

Feed-forward methods. We use the authors' released checkpoints for the 256×448 DL3DV dataset in DepthSplat and IDESplat. For TokenGS, we use the released DL3DV checkpoints corresponding to two, four, and six context views. We evaluate TokenGS in its feed-forward configuration, without its optional test-time token-tuning (TT). We fine-tune MVSplat from its released two-view 256×256 RE10K checkpoint for 200k steps on DL3DV, using a learning rate of 4×10^{-5} and an effective batch size of two. We fine-tune SurfSplat from its released two-view 256×448 RE10K Base checkpoint for 200k steps, using a learning rate of 2×10^{-4} and an effective batch size of two. Both fine-tuning runs use the same view-sampling protocol as our model. The retrained DepthSplat baseline in the ablation study is distinguished from the released checkpoint used in the main comparison.

Render-then-upsample cascaded baselines. We reconstruct the scene using DepthSplat and render target views at 256×448. We then resize these renderings by each evaluated target scale factor using bilinear or bicubic interpolation. The Gaussian representation is shared across output scales.

B Additional Experiments and Analysis

B.1 Additional DL3DV Comparisons

In Table 4, we present comparisons on DL3DV with 2 and 4 context views. Our ZoomSplat achieves state-of-the-art in these settings as well. Notably, ZoomSplat's average PSNR gains over DepthSplat and TokenGS increase with the number of context views.

B.2 Additional Ablations

OSP settings. We ablate the per-primitive opacity upper bound and transmittance regularization. All three variants in Table 5 use exactly the same full architecture: the decoupled 3D Gaussian prediction head with filter-size conditioning and our proposed visible-view L_2 frequency bound. Accumulated-alpha color normalization is enabled in all three variants, and the scale penalty weight is fixed at $\lambda_s = 0.03$. We vary only the primitive-opacity setting τ_{opacity} and the transmittance regularization weight λ_T. We additionally report the validation curves in Figure 6.

The configuration with $\tau_{\text{opacity}} = 0.6$ and no transmittance regularization follows SurfSplat's FAB in its primitive-opacity upper bound and accumulated-alpha color normalization. Figure 6 shows a transient drop in validation PSNR near 105k training steps, followed by recovery and subsequent fluctuations. Adding transmittance regularization avoids a comparable drop in the reported run and improves all three final evaluation metrics (Table 5). With regularization enabled, changing τ_{opacity} from 0.6 to 1.0 yields similar final metrics. We therefore use $\tau_{\text{opacity}} = 1.0$, omitting additional primitive-opacity rescaling.

Ablations on filter-size conditioning. Table 6 compares three variants trained with the same controlled ablation schedule, all using MAS3DF filtering and OSP. Variant (a) is the full ablation model, with both the decoupled head architecture and filter-size conditioning embedding. Variant (b) retains the decoupled head architecture but removes the filter-size conditioning embedding. Variant (c) instead uses an unconditioned joint head followed by MAS3DF filtering; we refer to this configuration as *post-filtering*, since the filter is applied after Gaussian attribute prediction without providing its

Table 4: **Additional comparisons on 2-/4-view DL3DV dataset.** Gaussians are reconstructed with 2/4 context views at 256×448 resolution and evaluated at $1\times$, $2\times$, $4\times$, and $8\times$ scales to simulate zoom-in effects. We report PSNR↑ / SSIM↑ / LPIPS↓. V refers to the number of context views. "PS" refers to per-scene optimization-based methods, "FF" refers to feed-forward methods, and "CC" refers to render-then-upsample cascade methods. * refers to models re-trained on DL3DV for fair comparison.

V	Type	Method	1×	2×	4×	8×	Avg.
2	PS	3DGS	15.42 / 0.324 / 0.540	13.81 / 0.364 / 0.600	13.27 / 0.431 / 0.637	12.94 / 0.519 / 0.647	13.86 / 0.410 / 0.606
		Mip-Splat.	15.72 / 0.334 / 0.542	14.45 / 0.388 / 0.594	14.15 / 0.476 / 0.632	13.95 / 0.577 / 0.665	14.57 / 0.444 / 0.608
		AA-2DGS	15.27 / 0.313 / 0.549	14.10 / 0.366 / 0.599	13.80 / 0.447 / 0.633	13.60 / 0.544 / 0.652	14.19 / 0.417 / 0.608
	CC	DS+Bil.	19.62 / 0.626 / 0.299	18.90 / 0.606 / 0.435	18.45 / 0.629 / 0.512	18.12 / 0.685 / 0.567	18.77 / 0.637 / 0.453
		DS+Bic.	19.62 / 0.626 / 0.299	18.92 / 0.608 / 0.418	18.47 / 0.629 / 0.504	18.14 / 0.683 / 0.566	18.79 / 0.637 / 0.447
	FF	MVSplat*	17.57 / 0.525 / 0.393	16.73 / 0.528 / 0.481	16.09 / 0.563 / 0.547	15.74 / 0.631 / 0.579	16.53 / 0.562 / 0.500
		DepthSplat	19.62 / 0.626 / 0.299	18.17 / 0.593 / 0.410	17.06 / 0.591 / 0.507	16.50 / 0.638 / 0.550	17.84 / 0.612 / 0.441
		SurfSplat*	19.63 / 0.620 / 0.308	18.55 / 0.574 / 0.457	16.84 / 0.498 / 0.594	15.35 / 0.506 / 0.623	17.59 / 0.549 / 0.496
		IDESplat	19.96 / 0.638 / 0.298	18.33 / 0.586 / 0.440	16.92 / 0.567 / 0.535	16.24 / 0.612 / 0.567	17.86 / 0.601 / 0.460
		TokenGS	19.92 / 0.618 / 0.432	19.14 / 0.598 / 0.528	18.65 / 0.627 / 0.571	18.30 / 0.687 / 0.583	19.00 / 0.632 / 0.528
		Ours	**20.42 / 0.648 / 0.281**	**19.63 / 0.626 / 0.384**	**19.12 / 0.641 / 0.473**	**18.74 / 0.689 / 0.534**	**19.48 / 0.651 / 0.418**
4	PS	3DGS	19.62 / 0.518 / 0.402	16.79 / 0.499 / 0.501	15.74 / 0.511 / 0.568	15.13 / 0.571 / 0.590	16.82 / 0.525 / 0.515
		Mip-Splat.	19.58 / 0.503 / 0.419	17.50 / 0.515 / 0.493	17.01 / 0.560 / 0.550	16.69 / 0.634 / 0.550	17.69 / 0.553 / 0.503
		AA-2DGS	19.24 / 0.492 / 0.419	17.21 / 0.502 / 0.498	16.71 / 0.540 / 0.555	16.38 / 0.606 / 0.584	17.39 / 0.535 / 0.514
	CC	DS+Bil.	23.16 / 0.779 / 0.175	21.89 / 0.706 / 0.342	21.16 / 0.685 / 0.444	20.64 / 0.716 / 0.520	21.71 / 0.722 / 0.370
		DS+Bic.	23.16 / 0.779 / 0.175	21.99 / 0.716 / 0.320	21.25 / 0.690 / 0.434	20.71 / 0.717 / 0.517	21.78 / 0.726 / 0.361
	FF	MVSplat*	21.71 / 0.721 / 0.227	19.89 / 0.664 / 0.346	18.49 / 0.634 / 0.456	17.82 / 0.667 / 0.514	19.48 / 0.671 / 0.386
		DepthSplat	23.16 / 0.779 / 0.175	20.55 / 0.700 / 0.305	18.61 / 0.639 / 0.442	17.69 / 0.657 / 0.507	20.00 / 0.694 / 0.357
		SurfSplat*	23.11 / 0.775 / 0.181	21.52 / 0.696 / 0.350	19.17 / 0.573 / 0.542	16.75 / 0.524 / 0.598	20.14 / 0.642 / 0.418
		IDESplat	23.59 / 0.790 / 0.174	20.44 / 0.683 / 0.350	17.79 / 0.588 / 0.493	16.63 / 0.601 / 0.542	19.61 / 0.666 / 0.390
		TokenGS	22.76 / 0.727 / 0.336	21.58 / 0.662 / 0.460	20.84 / 0.659 / 0.523	20.33 / 0.706 / 0.552	21.38 / 0.689 / 0.468
		Ours	**24.10 / 0.804 / 0.157**	**22.86 / 0.745 / 0.273**	**22.02 / 0.714 / 0.389**	**21.42 / 0.729 / 0.474**	**22.60 / 0.748 / 0.323**

Table 5: **Ablations on OSP hyperparameters.** Six-view DL3DV results at $1\times$ scale. All variants use the full conditioned decoupled head and proposed visible-view L_2 bound, and are trained using the controlled ablation protocol. Observed instability indicates whether a pronounced transient degradation occurred during the reported training run. Metrics are ranked independently using the unrounded source values; best and second-best values are marked in **bold red** and underlined blue, respectively.

τ_{opacity}	λ_T	λ_s	PSNR↑ / SSIM↑ / LPIPS↓	Observed Instability
0.6	0	0.03	24.55 / 0.811 / 0.174	Yes
0.6	0.005	0.03	24.92 / **0.835** / **0.134**	No
1.0	0.005	0.03	**24.93** / 0.835 / 0.134	No

size to the prediction head. Variants (a) and (c) correspond to variants (e) and (d), respectively, in Table 3. Comparing (a) and (b) shows that removing filter-size conditioning directly hurts reconstruction performance in this controlled comparison.

Table 6: **Ablations on filter-size conditioning and the decoupled head.** Six-view DL3DV results under the same controlled ablation training schedule, with MAS3DF and OSP enabled in all variants. Cond. denotes the filter-size conditioning embedding and DH denotes the decoupled head architecture; O and X indicate enabled and disabled, respectively. Variant (c) uses an unconditioned joint head with post-filtering. Entries report PSNR↑ / SSIM↑ / LPIPS↓; Avg. averages each metric over the four scales. Best and second-best distinct values are marked in **bold red** and underlined blue, respectively, with ties determined at the supplied source precision.

Method	Cond.	DH	1×	2×	4×	8×	Avg.
(a)	O	O	**24.93 / 0.835 / 0.134**	**23.56 / 0.770 / 0.252**	**22.63 / 0.729 / 0.374**	**21.97 / 0.736 / 0.462**	**23.27 / 0.767 / 0.306**
(b)	X	O	24.76 / 0.827 / 0.143	23.43 / 0.765 / 0.260	22.53 / 0.726 / 0.379	21.88 / 0.735 / 0.465	23.15 / 0.763 / 0.312
(c)	X	X	24.84 / 0.833 / 0.136	23.48 / 0.768 / 0.253	22.57 / 0.728 / 0.375	21.91 / 0.735 / **0.462**	23.20 / 0.766 / **0.306**

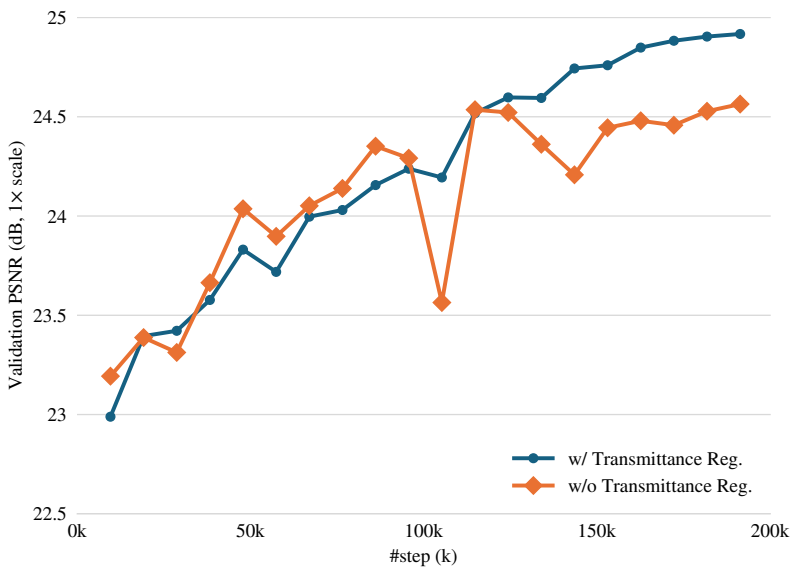


Figure 6: Effect of transmittance regularization on training stability. We report validation PSNR at $1\times$ scale versus training steps. Both runs use MAS3DF, the filter-conditioned decoupled head, $\tau_{\text{opacity}} = 0.6$, and $\lambda_s = 0.03$. We compare $\lambda_T = 0$ (w/o regularization) with $\lambda_T = 0.005$ (w/ regularization). Each curve represents one training run.

C Limitations

ZoomSplat aims to reconstruct a fixed set of pixel-aligned Gaussian primitives for detail-preserving anti-aliased rendering, without explicit super-resolution architectures or higher-resolution training supervision. Its design therefore does not guarantee recovery of fine detail absent from context view observations. Although ZoomSplat suppresses rendering artifacts under magnification, rendered views may remain soft at large zoom factors. We leave the recovery of high-resolution detail through complementary super-resolution designs for future work.

D Additional Visual Results

We present additional visual comparisons with existing feed-forward 3DGS methods. Figure 7 shows visual comparisons on DL3DV against DepthSplat, IDESplat, SurfSplat, and TokenGS, across $1\times$, $2\times$, $4\times$, and $8\times$ scales. Figure 8 shows visual comparisons on Tanks&Temples against DepthSplat, SurfSplat, and TokenGS, across $1\times$, $2\times$, $4\times$, and $8\times$ scales.

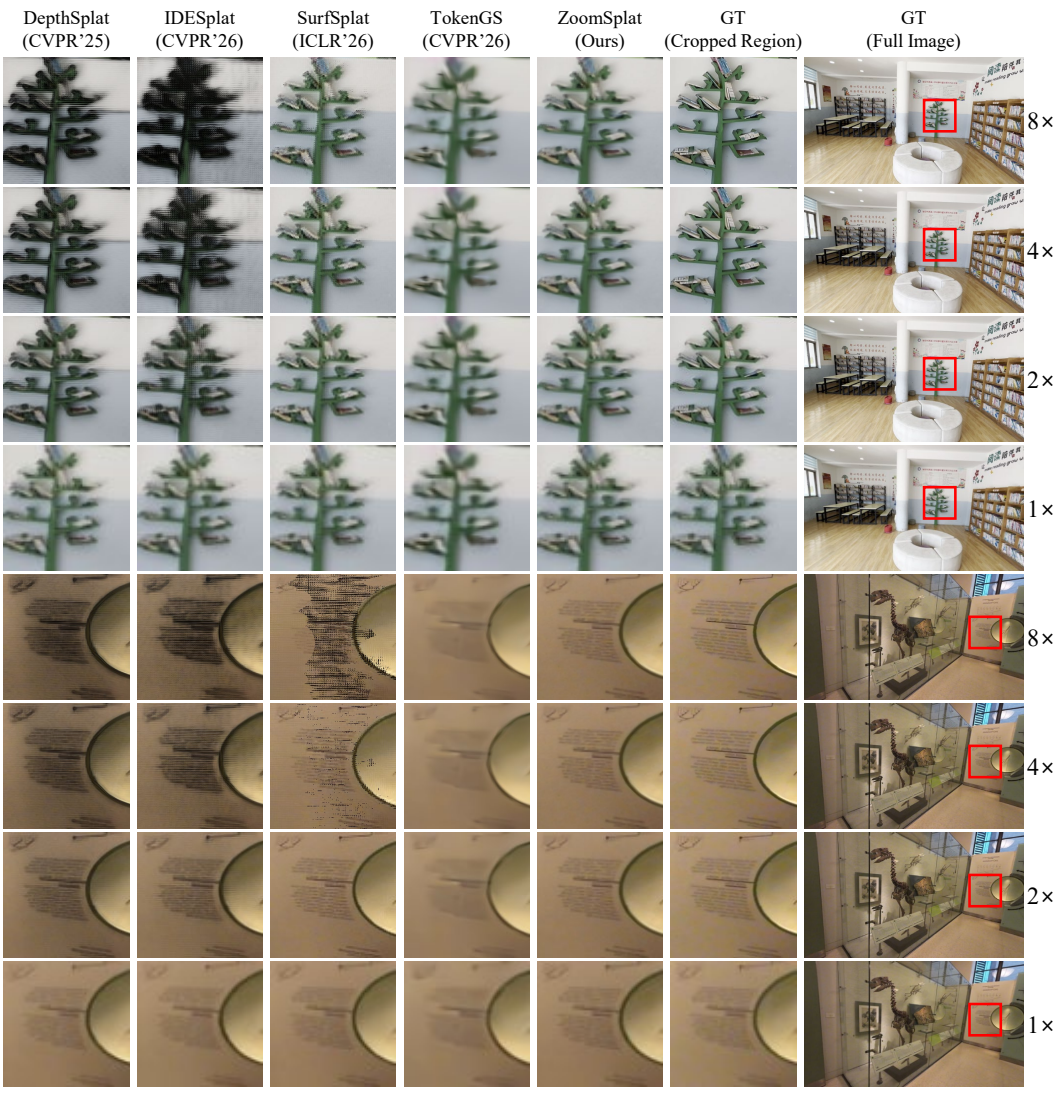


Figure 7: **Additional visual results on DL3DV.** Best viewed zoomed-in.

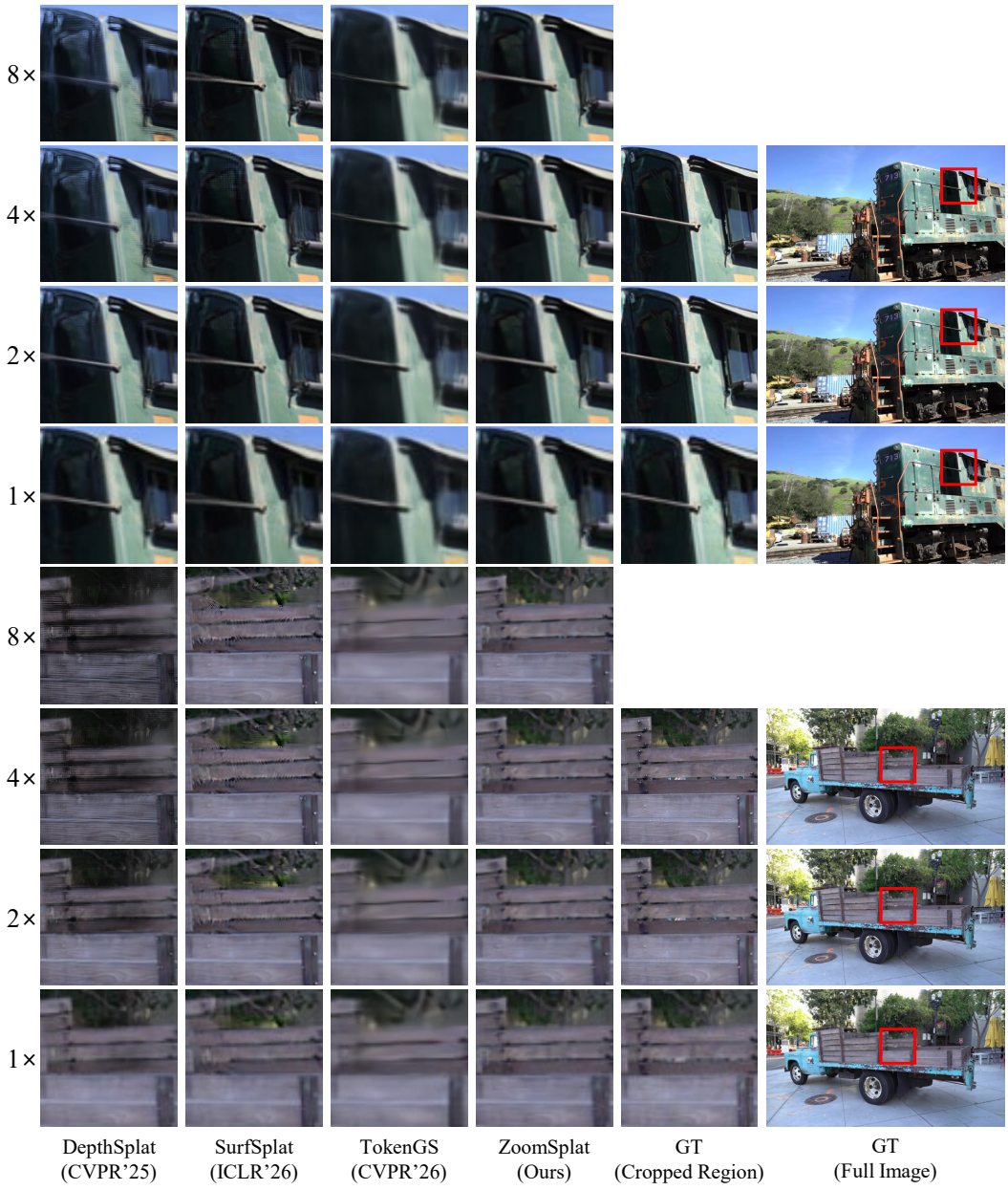


Figure 8: **Visual results on zero-shot Tanks&Temples.** Best viewed zoomed-in.