SubSplat: High-Resolution Pixel-aligned 3DGS via Sub-pixel Gaussian Reparameterization

Jiun Lee[1], Jaekwang Kim[2], and Sangmin Lee[3]*

[1] AimFuture(c), jiun.lee@aimfuture.ai
[2] Sungkyunkwan University, linux@skku.edu
[3] Korea University, sangmin-lee@korea.ac.kr

Abstract. Pixel-aligned Gaussian splatting enables efficient and generalizable novel-view synthesis. However, high-resolution rendering faces a critical trade-off where increasing input resolution improves detail at the expense of quadratically rising network computational cost. Conversely, maintaining low-resolution inputs stabilizes this cost but results in insufficient Gaussian density and artifacts. To address this, we propose SubSplat, which introduces Sub-pixel Gaussian Reparameterizer(SPGR) to subdivide primary Gaussians into fine-grained primitives, restoring structural density directly from low-resolution features. We further enhance the reparameterization quality through feature aggregation, which effectively captures high-frequency details across multiple views. Experiments on RealEstate10K and ACID demonstrate that SubSplat achieves high-fidelity rendering with superior efficiency. Our results validate that the proposed framework successfully resolves the trade-off between reparameterization fidelity and network computational cost inherent in pixel-aligned Gaussian Splatting.

1 Introduction

Novel-View Synthesis (NVS) aims to generate photorealistic images of a scene from new camera viewpoints given a small set of calibrated input photos [6, 12, 15, 23, 24]. NVS supports important applications in AR/VR telepresence, robotic perception, and content creation. For practical deployment, these applications require sharp, temporally stable views at interactive rates on high-resolution displays, including resource-constrained devices [6, 15, 24].

Recent neural rendering [1, 6, 15, 24] advances have significantly improved NVS quality and practicality. These advances motivate designs that preserve image quality while keeping computation and memory costs predictable across devices and display scales. Among these advances, 3D Gaussian Splatting(3DGS) [15] achieves high-quality rendering with learnable 3D anisotropic Gaussians and a differentiable rasterizer. Prior works improve rendering quality through anti-aliasing [36, 41] and adaptive densification [15, 27, 46]. However, these methods require per-scene optimization, limiting generalization and preventing efficient

* Corresponding author

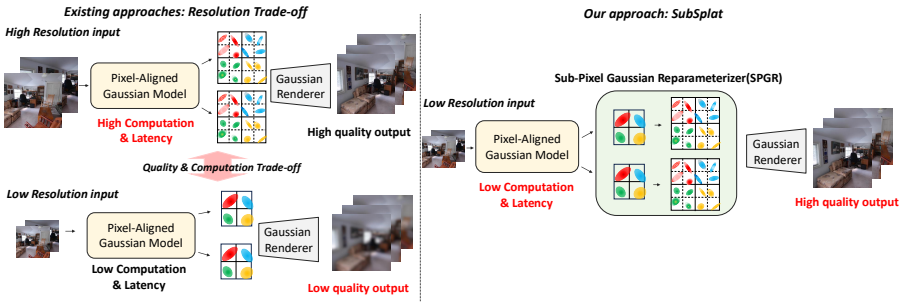

Fig. 1: Comparison of High-Resolution Rendering Strategies. Unlike prior methods limited by fixed grids, SubSplat generates Sub-pixel Reparameterized Gaussians from low-resolution features to mitigate the quadratic cost of high-resolution inputs. This enables high-fidelity reconstruction at extended scales while maintaining stable network latency.

deployment across diverse scenes. To tackle this problem, pixel-aligned Gaussian methods [3, 5, 30, 34, 42, 43] train feed-forward networks across diverse scenes, enabling generalized Gaussian generation without per-scene optimization. These methods predict Gaussian primitives on a fixed image grid from sparse input views and render them with a differentiable splatting backend, achieving faster inference on novel scenes.

Despite recent progress, pixel-aligned methods face significant challenges in scaling to high-resolution rendering. These methods predict Gaussian attributes based on a fixed image grid, where the total number of primitives is intrinsically tied to the input resolution. This grid-based design creates a fundamental quality-efficiency dilemma, as the total system latency is primarily dominated by the backbone network's forward pass.

The straightforward strategy to achieve high-resolution rendering is to increase the input resolution accordingly. However, this causes quadratic computational growth in the backbone network, as the network must process a much denser grid to predict the additional primitives. Since the total latency is primarily dominated by this backbone forward pass, doubling the resolution requires $4\times$ more computation, making it impractical for interactive use. To avoid this burden, one might use low-resolution inputs and render the resulting sparse Gaussians on a high-resolution screen. Yet, this leads to insufficient Gaussian density relative to the target pixels. This under-parameterization inevitably results in noticeable blur and halo artifacts, as the limited number of grid-anchored primitives cannot adequately represent high-frequency details or sharp structural edges. Consequently, existing pixel-aligned methods are trapped in a trade-off: they must choose between (i) quadratic increase in backbone computational cost when the input resolution is scaled up to match the target output resolution or (ii) quality degradation when forcing high-resolution outputs from coarse input grids.

To address the computational inefficiency and quality degradation of existing pixel-aligned models, we propose SubSplat, a novel framework featuring

Sub-pixel Gaussian Reparameterizer(SPGR). As shown in Fig. 1, our approach moves beyond the constraints of fixed grid-sampling by introducing SPGR that redistributes Gaussian geometry and appearance for the target resolution. Instead of increasing the backbone's input resolution, which leads to a quadratic increase in computational cost, our method maintains a stable backbone latency by utilizing low-resolution inputs. The reparameterizer transforms each grid-anchored primary Gaussian into a set of fine-grained sub-pixel primitives through an efficient module. This enables content-aware density scaling to higher target resolutions (e.g. $\times 2$, $\times 4$ the input) with negligible computational overhead, effectively restoring structural details while preserving inference efficiency.

To further enhance these sub-pixel primitives, we incorporate deformable attention to align Gaussian primitives with multi-view features. This mechanism aggregates contextual cues across views, ensuring that the increased Gaussian density translates into sharp details and geometric consistency. Together, these designs allow our method to produce high-fidelity, high-resolution outputs while maintaining a highly efficient inference pipeline. By separating the output primitive density from the backbone's computational resolution, we achieve high-resolution rendering without a quadratic increase in network costs. This approach effectively solves the structural bottleneck where high-quality results previously depended on using heavy and expensive backbone grids.

The contributions of this work are as follows:

- We propose SubSplat, a novel framework that enables high-resolution rendering while keeping the input resolution low, preventing the quadratic scaling of network computation costs which exist in the previous pixel-aligned approaches.
- We design a Sub-pixel Gaussian Reparameterizer(SPGR) that subdivides each grid-anchored Gaussian into fine-grained primitives through dedicated geometry and appearance features. Additionally, a footprint-aware opacity redistribution mechanism ensures that total opacity is conserved across subdivisions.
- SubSplat achieves state-of-the-art performance across multiple target scales, delivering over $3\times$ latency reduction compared to full-resolution baselines while providing superior reconstruction quality that surpasses both pixel-aligned methods and even external upsampling approaches.

2 Related Work

3D Gaussian splatting. 3DGS [15] represents scenes with anisotropic Gaussians and a differentiable rasterizer, enabling real-time novel-view synthesis and fast optimization. Early point-based renderers [2, 28] pioneered point-based representations with multi-resolution hierarchies. EWA splatting for points and surfaces [49, 50] established footprint-based anti-aliasing. Building on this foundation, Mip-Splatting [41], Multi-Scale 3DGS [36], Analytic-Splatting [19], and Anti-Aliased 2DGS [39] based on [12] further mitigate aliasing across scales.

Adaptive densification adds Gaussians in regions with large gradients [15], with recent work refining schedules and density control [8, 22, 27, 46]. However, these approaches still rely on per-scene optimization and fixed resolutions, limiting their generalization and scalability. To address this, SubSplat employs a feed-forward architecture with a subdivision mechanism to mitigate the resolution constraints of the feature grid. This allows for fine-grained reconstruction while reducing the dependency on the initial feature scale.

High-resolution novel-view synthesis(NVS). High-resolution NVS typically relies on per-scene optimization with various supervision strategies. NeRF [23]-based methods [13,20,32] enhance sub-pixel details but require high-resolution ground truth or suffer from slow rendering. Recent approaches leverage external 2D priors: single-image SR models [38] or diffusion models with 3D synchronization [17]. In the 3DGS line, GaussianSR [40] applies diffusion-based Score Distillation Sampling, SuperGS [33] introduces a coarse-to-fine framework with latent feature fields, and S2Gaussian [31] densifies low-resolution Gaussians through Shuffle Split and refines with super-resolved multi-view supervision. SRGS [10] uses pre-trained SR backbones [18] to provide pseudo-HR supervision, though multi-view inconsistencies can introduce ambiguities. While effective at enhancing detail, these methods require per-scene optimization, limiting cross-scene generalization. Pixel-aligned feed-forward pipelines address this by enabling immediate inference without scene-specific tuning.

Pixel-aligned Gaussian splatting. Pixel-aligned methods [3,5,30,34,42] train feed-forward networks to predict Gaussian parameters on a fixed image grid, enabling generalized novel-view synthesis without per-scene optimization. Multi-view variants [5, 30] aggregate information across views via cross-attention and construct plane-sweep cost volumes to guide correspondence and geometry hypotheses [7, 35, 37]. By relying on a fixed number of Gaussians tied to the input grid, these designs face an inherent quality-cost constraint. Improving reconstruction detail necessitates higher input resolutions, which triggers a quadratic surge in computational overhead. Depth-guided variants [34, 42] leverage pre-trained depth priors to improve reconstruction quality, yet remain grid-anchored at a fixed feature scale. Recent refinement approaches [9,25] add detail by modulating per-pixel attributes or synthesizing additional Gaussians. However, these methods remain constrained by the feature grid resolution, making output fidelity dependent on the backbone's computational scale. To address this, SubSplat introduces a subdivision mechanism that generates sub-pixel primitives from coarse Gaussians. This alleviates the resolution dependency, enabling high-fidelity rendering with improved computational efficiency.

3 Proposed Method

3.1 Overview

SubSplat is an end-to-end pipeline that synthesizes high-resolution scenes from coarse feature representations. The process begins with a pixel-aligned backbone [5] initializing primary Gaussians. To enrich these primitives, a deformable-

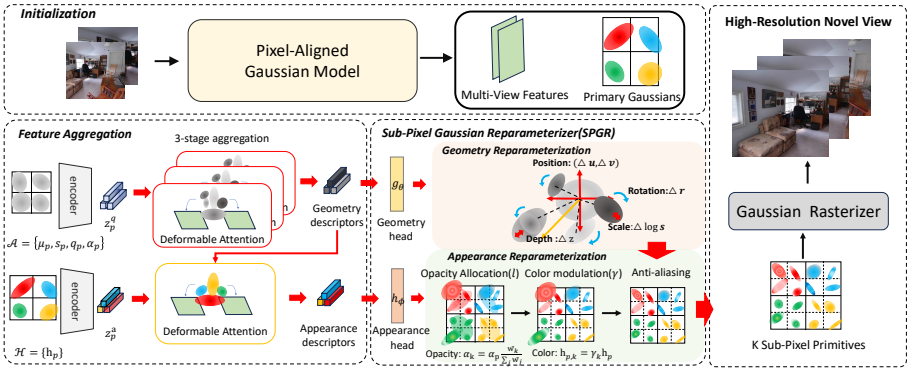

Fig. 2: Overview of the SubSplat Architecture. SubSplat aggregates multi-view features via Deformable Attention to extract rich geometry and appearance descriptors. These descriptors guide the Sub-pixel Gaussian Reparameterizer(SPGR) to subdivide Gaussians into fine-grained primitives, restoring high-resolution details while incurring only a marginal increase in total inference time.

attention module [9, 14] performs feature aggregation to integrate auxiliary geometry and appearance information, providing the necessary contextual cues for the reparameterization stage. At the core of our method, a sub-pixel Gaussian reparameterizer(SPGR) subdivides these primary Gaussians into multiple sub-pixel primitives. This subdivision alleviates the resolution dependency on the backbone's feature scale, while an attribute redistribution strategy modulates color and preserves integrated opacity for high-fidelity reconstruction. By shifting detail enhancement to this subdivision stage, SubSplat recovers sharp edges and fine textures with only a marginal increase in network computational overhead.

3.2 Feature Aggregation

3.2.1 Initialization We use the same input structure as MVSplat [5], keeping camera metadata separate from learnable states.

Anchors and color. From the backbone output, let N primary Gaussians be

$$\mathcal{A} = \{\mathbf{a}_p\}_{p=1}^{N}, \qquad \mathbf{a}_p = [\,\boldsymbol{\mu}_p, \mathbf{s}_p, q_p, \alpha_p\,] \in \mathbb{R}^{11}, \tag{1}$$

together with their corresponding spherical-harmonic(SH) color

$$\mathcal{H} = \{\mathbf{h}_p \in \mathbb{R}^{C\times D_{\text{sh}}}\}_{p=1}^{N}. \tag{2}$$

Here μ_p denotes the world-space position, and $\mathbf{s}_p$ are the per-axis scales that define the anisotropic footprint of the Gaussian in its local frame. The unit quaternion $\mathbf{q}_p$ represents its orientation. We use degree-4 SH by default (C=3,

D_{sh}=25). These sets provide the anchor states and colors that downstream modules will refine.

Backbone features. We also take multi-view feature maps

$$\mathcal{F} = \{\boldsymbol{\Phi}_{v,\ell}\}_{v=1..V,\ \ell=1..L}, \qquad \boldsymbol{\Phi}_{v,\ell} \in \mathbb{R}^{H_\ell \times W_\ell \times D_\ell}, \tag{3}$$

which will be sampled by deformable attention in the feature aggregation stage.

Camera metadata. Camera intrinsics and extrinsics are fixed and provide the projection information used in both the deformable-attention feature aggregation and the SPGR.

3.2.2 Geometry feature aggregation Given primary Gaussian states $\mathcal{A}$ and multi-view features $\mathcal{F}$, we form a per-anchor geometry query as a sum of attribute-wise embeddings followed by a projection using an MLP:

$$\mathbf{z}_p^g = W_g\big(\phi_x(\boldsymbol{\mu}_p) + \phi_s(\mathbf{s}_p) + \phi_q(q_p) + \phi_\alpha(\alpha_p)\big). \tag{4}$$

Each Gaussian then aggregates multi-view evidence with a deformable-attention operator over three aggregation stages:

$$\begin{aligned} \mathbf{f}_p^{g,0} &= \mathbf{0}, \\ \mathbf{f}_p^{g,t+1} &= \mathrm{DA}\big(\mathcal{F};\, \mathbf{z}_p^g, \mathbf{f}_p^{g,t}, \mathcal{P}(\boldsymbol{\mu}_p), \mathbf{a}_p\big), \quad t = 0, 1, 2, \\ \mathbf{f}_p^g &= \mathbf{f}_p^{g,3}. \end{aligned} \tag{5}$$

Here, ϕ_x, ϕ_s, ϕ_q, and ϕ_α are small per-attribute MLPs, and W_g is a linear projection. We align the geometry embedding dimension with the multi-view feature space and compute view-dependent projection matrices from the camera metadata. Each primary Gaussian is projected onto the reference views through these matrices, and the resulting 2D projections are used by the deformable attention module [14] to aggregate view-specific geometric context. Inspired by Deformable DETR's deformable attention [48], we adopt a multi-stage design and stack three deformable-attention layers to progressively aggregate cross-view features. This multi-stage design enhances geometric consistency by integrating local-to-global context across multiple feature scales. This process yields the geometry descriptor $\mathbf{f}_p^g$, a compact, view-consistent representation of cross-view geometry for subsequent SPGR.

3.2.3 Appearance feature aggregation We map the primary SH coefficients to an appearance query:

$$\mathbf{z}_p^a = \phi_c\big(\mathrm{vec}(\mathbf{h}_p)\big), \tag{6}$$

where ϕ_c is an MLP that produces an embedding aligned with the multi-view feature dimension, and $\mathrm{vec}(\mathbf{h}_p) \in \mathbb{R}^{C \cdot D_{\text{sh}}}$ flattens the spherical-harmonic coefficients. We then aggregate appearance features via deformable attention, conditioned on the geometry descriptor

$$\mathbf{f}_p^a = \mathrm{DA}\big(\mathcal{F};\, \mathbf{z}_p^a, \mathbf{f}_p^g, \mathcal{P}(\boldsymbol{\mu}_p), \mathbf{a}_p\big). \tag{7}$$

The geometry descriptor $\mathbf{f}_p^g$ serves as the conditioning input, guiding the aggregation based on the established geometric context. By reusing the same projection context $\mathcal{P}(\mu_p)$ and primary anchor $\mathbf{a}_p$ as the geometry path, this design allows geometry to determine *where to look* and appearance to determine *what to amplify*, maintaining structural integrity and improving color consistency across multiple views.

3.3 Sub-Pixel Gaussian Reparameterizer

Each primary Gaussian p is associated with geometry and appearance descriptors $\mathbf{f}_p^g, \mathbf{f}_p^a \in \mathbb{R}^E$, where E denotes the shared embedding dimension of the feature space used by the prediction heads. For each primary Gaussian p, the Sub-pixel Gaussian Reparameterizer(SPGR) divides it into K sub-pixel primitives at the target resolution. Here, K is defined as the squared ratio between the target and input resolutions (e.g., $K = 4$ for a $2\times$ resolution scale and $K = 16$ for a $4\times$ resolution scale). Using the aggregated descriptors $\mathbf{f}_p^g$ and $\mathbf{f}_p^a$, two dedicated prediction heads generate primitive-level refinements. Specifically, a geometry head g_θ predicts spatial properties including position, depth, scale, and rotation, while an appearance head h_ϕ predicts color and opacity adjustments. These primitive-level attributes are composed with the primary Gaussian's properties to construct the final primitives. Each process is described in detail below.

3.3.1 Geometry Reparameterization The geometry head $g_\theta : \mathbb{R}^E \to \mathbb{R}^{K\times 9}$ predicts spatial properties for each sub-pixel primitive:

$$[\Delta u_k, \Delta v_k, \Delta z_k, \Delta \log \mathbf{s}_k, \Delta \mathbf{r}_k]_{k=1}^K = g_\theta(\mathbf{f}_p^g). \tag{8}$$

Here, $(\Delta u_k, \Delta v_k) \in \mathbb{R}^2$ are the sub-pixel offsets within the primary Gaussian's footprint, $\Delta z_k \in \mathbb{R}$ is the depth residual, $\Delta \log \mathbf{s}_k \in \mathbb{R}^3$ denotes the log-scale residuals for anisotropic scaling, and $\Delta \mathbf{r}_k \in \mathbb{R}^3$ represents the axis-angle rotation.
Position and depth. We select, for each anchor, as the primary view the camera in which the anchor is visible and has the smallest positive depth [4, 37]. SPGR then predicts sub-pixel Gaussian position and depth in that view. Specifically, K sub-Gaussians are anchored to a uniform grid centered on the primary Gaussian. Let z_{sel} denote this depth. Each sub-pixel offset $(\Delta u_k, \Delta v_k)$ is back-projected through the intrinsics and adjusted along the viewing ray [7, 11]:

$$\mu_{p,k} = \mu_p + R_{c2w} \begin{bmatrix} z_{\text{sel}} \Delta u_k / f_x \\ z_{\text{sel}} \Delta v_k / f_y \\ 0 \end{bmatrix} + \Delta \hat{z}_k \, R_{c2w} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{9}$$

where (f_x, f_y) are the focal lengths and R_{c2w} is the camera-to-world rotation. To keep the reparameterization stable across resolutions and viewpoints, the depth residual is bounded via a tanh mapping,

$$\Delta \hat{z}_k = \tanh(\Delta z_k), \tag{10}$$

which constrains updates along the ray within a small, scene-independent range. Please refer to Supplementary A.3 for more details.

Anisotropic scale and rotation. Each sub-pixel Gaussian inherits its primary Gaussian's orientation and scale, refined locally in the original coordinate system [15]. The updated scale is computed as

$$\mathbf{s}_{p,k} = \mathbf{s}_p \odot \exp(\Delta \log \mathbf{s}_k). \tag{11}$$

where $\odot$ denotes element-wise multiplication, allowing each sub-pixel Gaussian to stretch or compress independently along the three axes while remaining numerically stable.

For rotation, the residual $\Delta\mathbf{r}_k \in \mathbb{R}^3$ is an axis–angle vector (in radians) that represents a small local rotation [29] with respect to the sub-pixel Gaussian's orientation. This residual is converted into a quaternion, composed with the sub-pixel Gaussian's quaternion, and then normalized to form the final orientation:

$$q_{p,k} = \mathrm{norm}\big(q_p \otimes q(\Delta\mathbf{r}_k)\big). \tag{12}$$

Following 3D Gaussian Splatting [15], the rotation matrix $R(q_{p,k})$ and scale matrix $\mathrm{diag}(\mathbf{s}_{p,k}^2)$ are combined to form the covariance:

$$\Sigma_{p,k} = R(q_{p,k})\,\mathrm{diag}(\mathbf{s}_{p,k}^2)\,R(q_{p,k})^\top. \tag{13}$$

3.3.2 Appearance Reparameterization The appearance head $h_\phi : \mathbb{R}^E \to \mathbb{R}^{K\times 2}$ predicts two key components for each sub-pixel primitive:

$$[\ell_k, \gamma_k]_{k=1}^K = h_\phi(\mathbf{f}_p^a). \tag{14}$$

The first term ℓ_k is a distribution logit that partitions the primary Gaussian's opacity among its primitives via softmax normalization, ensuring the integrated opacity is preserved. The second term γ_k is a bounded scalar for color modulation, which refines the primary Gaussian's color to capture sub-pixel appearance details. Together, they achieve high-fidelity reconstruction by shifting detail enhancement to the subdivision stage.

Opacity allocation. To maintain integrated opacity, we distribute the primary Gaussian's opacity α_p among its assigned sub-pixel primitives. This process is guided by a footprint-aware formulation [19,39], concentrating opacity on smaller primitives to sharpen structural boundaries. By giving higher weights to primitives with smaller screen-space footprints, the model effectively sharpens structural edges. A final renormalization ensures that the total opacity of the primitives remains equal to that of the primary source, satisfying $\sum_k \alpha_k = \alpha_p$. Let $J_p \in \mathbb{R}^{2\times 3}$ be the projection Jacobian at the primary position. The projected area of each primitive is estimated as:

$$A_k \approx \sqrt{\det\big(J_p, \Sigma_k, J_p^\top\big)}. \tag{15}$$

We derive the inverse-area weights by combining the distribution logits ℓ_k with the estimated footprints:

$$w_k = \mathrm{softmax}(\ell_k), \qquad \tilde{w}_k = \frac{w_k}{A_k + \varepsilon}. \tag{16}$$

The final opacity for each primitive is then assigned as:

$$\alpha_k = \alpha_p \cdot \frac{\tilde{w}_k}{\sum_j \tilde{w}_j}. \tag{17}$$

This mechanism enforces distributional integrity from the primary Gaussian while stabilizing high-resolution synthesis by prioritizing density on intricate scene details.

Color modulation. The color of each sub-pixel Gaussian is scaled by a bounded gain factor γ_k:

$$\mathbf{h}_{p,k} = \gamma_k \, \mathbf{h}_p. \tag{18}$$

Here, $\mathbf{h}_p \in \mathbb{R}^{C \times D_{\mathrm{sh}}}$ denotes the primary Gaussian's spherical harmonic coefficients. This operation modulates color intensity within a stable range while preserving the primary color direction, consistent with multiplicative adjustments used for color calibration in Gaussian-based rendering [16, 26]. By scaling the existing coefficients, the model introduces fine-grained appearance variations without deviating from the color context inherited from the primary Gaussian.

Anti-aliasing. From an anti-aliasing perspective, A_k represents the projected screen-space footprint of each primitive. To stabilize sub-pixel Gaussian rendering, we enforce a minimum area $A_{\min}$; when $A_k < A_{\min}$, the in-plane scales are minimally adjusted to ensure consistent pixel coverage. Consistent with scale-aware formulations for 3DGS [19, 39], this clamping strategy preserves the effective footprint, mitigating flickering and aliasing artifacts during high-resolution synthesis. By maintaining a lower bound on the primitive size, the model ensures that the reparameterization stage remains robust even when generating extremely fine-grained structures.

3.4 Training Objective

Following MVSplat [5], our model predicts sub-pixel 3D Gaussian parameters and renders novel views from them. The network is trained with ground-truth RGB images using a weighted combination of ℓ_2 and LPIPS [44] losses:

$$\mathcal{L} = \mathcal{L}_{\ell_2} + \lambda \, \mathcal{L}_{\mathrm{LPIPS}}, \tag{19}$$

where λ is the weighting hyperparameter that balances the two losses. We set λ to 0.05 following the common protocol [3, 5, 42].

4 Experiments

To evaluate SubSplat, we conduct comparative experiments focusing on the trade-off between rendering resolution and computational cost. First, we reconstruct scenes from 256×256 inputs and render them at 512×512 ($\times 2$)

Table 1: Quantitative evaluation of resolution scalability on RealEstate10K [47] and ACID [21]. All methods reconstruct scenes from a fixed 256×256 input resolution, and performance is evaluated at $\times 2$ (512×512) and $\times 4$ (1024×1024) output scales to assess high-fidelity reconstruction capabilities. Metrics: PSNR↑ / SSIM↑ / LPIPS↓. Best and second-best results are bold and underlined, respectively.

Method	**RealEstate10K** [47]						**ACID** [21]					
	Output($\times 2$) 512×512			Output($\times 4$) 1024×1024			Output($\times 2$) 512×512			Output($\times 4$) 1024×1024		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
PixelSplat [3]	<u>23.44</u>	0.789	0.227	21.44	0.767	0.323	<u>24.41</u>	0.709	0.292	<u>22.95</u>	0.661	0.409
MVSplat [5]	19.46	0.751	0.269	16.98	0.671	0.424	18.38	0.645	0.349	16.57	0.616	0.489
TranSplat [42]	19.61	0.756	0.265	17.16	0.679	0.424	18.64	0.657	0.333	16.93	0.642	0.499
DepthSplat [34]	19.54	0.720	0.309	16.29	0.587	0.461	17.19	0.535	0.469	16.32	0.630	0.476
HiSplat [30]	23.26	<u>0.803</u>	<u>0.207</u>	<u>22.05</u>	<u>0.779</u>	<u>0.309</u>	23.59	<u>0.724</u>	<u>0.273</u>	22.03	<u>0.674</u>	<u>0.388</u>
Ours (SubSplat)	**25.52**	**0.850**	**0.167**	**22.65**	**0.781**	**0.268**	**26.03**	**0.775**	**0.216**	**23.35**	**0.674**	**0.330**

and 1024×1024 ($\times 4$) resolutions to observe structural details. We compare our model against baselines where the input and output resolutions are coupled at 512×512. This analysis validates that SubSplat achieves comparable fidelity while significantly reducing network overhead, specifically in inference latency and peak memory. We also contrast our 3D primitive-level refinement with image-space upsamplers, such as bilinear interpolation and HiT-SR [45], to assess the impact of densifying geometry in 3D space. Finally, we perform an ablation study to verify the effectiveness of each component in our model for scene reconstruction.

4.1 Experimental Setup

Datasets. All data follows the PixelSplat mining procedure [3] to ensure consistency across evaluations. Following the MVSplat setup, SubSplat and all baselines evaluate sparse-view performance using exactly two input views with frame gaps of 45–192. We train all models on the 720p RealEstate10K [47] training split, which consists of 64,002 scenes. During inference, the 720p Real-Estate10K [47] test set (7,064 scenes) is used to evaluate $\times 2$ resolution scalability, backbone efficiency, and comparisons with image-space upsamplers, as well as for the ablation study. We utilize the 1080p RealEstate10K split for $\times 4$ inference evaluations and ablation studies. Due to the scarcity of high-resolution training data(1080p), we instead train a base model on 720p sequences using 128×128 inputs to target 512×512 outputs. This configuration is then directly applied to 1024×1024 inference from 256×256 inputs, validating SubSplat's resolution scalability across 3,703 high-resolution scenes. To evaluate cross-dataset generalization, we test the models on the 720p ACID [21] dataset (2,026 scenes) for $\times 2$ scene reconstruction. Furthermore, we assess $\times 4$ generalization using the 1080p ACID [21] dataset, which contains 1,622 test scenes.

Baselines. We compare SubSplat with pixel-aligned models, PixelSplat [3], MVSplat [5] and HiSplat [30], and depth-guided variants, TranSplat [42] and

Fig. 3: Qualitative results at 512×512 screen space. While pixel-aligned baselines suffer from geometric instability and blurring when rendering beyond their 256×256 input grid, SubSplat(ours) recovers sharp, structurally consistent details on RealEstate10K [47] and ACID [21] through sub-pixel Gaussian reparameterization.

DepthSplat [34]. To evaluate the benefits of 3D-level densification, we also compare against image-space upsamplers, including bilinear interpolation and HiT-SR [45]. These upsamplers are applied to MVSplat [5] by reconstructing scenes from 256×256 inputs and rendering at 512×512 resolution.

Implementation Details. SubSplat is trained for 300K iterations on a single NVIDIA A100 (80 GB) GPU with a batch size of 12. We use the Adam optimizer with a learning rate of 2×10^{-4} and a OneCycleLR scheduler featuring a 2K-step warm-up. Our primary model is trained to reconstruct scenes from 256×256 inputs and render them at 512×512 resolution. To analyze resolution scalability for extensive output resolutions, we train a model on a 128×128 to 512×512 task ($\times 4$ scaling). This trained model is then directly applied to evaluate 1024×1024 inference from 256×256 inputs, demonstrating its ability to maintain high-fidelity performance across larger absolute scales while preserving the same $\times 4$ scaling ratio. Efficiency is evaluated by comparing our 256×256 input model against baselines using 512×512 inputs. All efficiency metrics are measured on the same NVIDIA A100 GPU to ensure a fair comparison. Regarding the inference setup, baselines are not evaluated under the 1024×1024 input setting because their memory-intensive architectures result in Out-of-Memory (OOM) issues. All image-space upsamplers, including HiT-SR [45], are applied to MVSplat [5] outputs using the 256×256 to 512×512 configuration. All variants in the ablation study, including the baseline, were trained using the exact same objective with low-resolution inputs and high-resolution target.

4.2 Results and Analysis

Resolution Scalability. Figures 3 and 4 demonstrate that SubSplat recovers sharper geometry and more stable textures than existing models across various output scales. While MVSplat [5] and DepthSplat [34] suffer from edge flickering and aliasing at 512×512 and 1024×1024, SubSplat maintains structural

Fig. 4: Qualitative results at 1024×1024 **screen space.** Even with a 256×256 input, SubSplat(ours) maintains structural consistency and sharp details on RealEstate10K [47] and ACID [21]. In contrast, pixel-aligned baselines exhibit blurring or severe halos when rendering at $4\times$ resolution.

Table 2: Cost–performance comparison on RealEstate10K [47] at a fixed target resolution. SubSplat achieves the best balance of quality and efficiency, delivering competitive reconstruction fidelity with significantly lower latency and peak memory by utilizing lower-resolution inputs.

Method	Input Res.	Output Res.	Latency (ms) ↓	Peak Memory (GB) ↓	PSNR ↑	SSIM ↑	LPIPS ↓
PixelSplat [3]	512×512	512×512	561	9.56	24.80	0.839	0.190
MVSplat [5]	512×512	512×512	131	4.27	24.98	0.842	0.172
TranSplat [42]	512×512	512×512	192	5.14	23.50	0.804	0.197
DepthSplat [34]	512×512	512×512	<u>123</u>	<u>4.58</u>	24.70	**0.851**	0.170
HiSplat [30]	512×512	512×512	1960	4.87	<u>25.34</u>	0.850	<u>0.168</u>
Ours (SubSplat)	256×256	512×512	**42**	**1.57**	**25.52**	<u>0.850</u>	**0.167**

clarity and fine details through its sub-pixel Gaussian reparameterization. Interestingly, PixelSplat [3] shows relatively less fidelity loss than other baselines by assigning a fixed number of three Gaussians per pixel, providing better coverage for high-resolution grids. Nevertheless, SubSplat consistently outperforms PixelSplat [3] by exhibiting superior robustness to increasing output scales, achieved by directly optimizing high-density Gaussian primitives for multi-scale rendering. Quantitative results in Table 1 further validate this robustness. SubSplat achieves state-of-the-art performance across all metrics on RealEstate10K [47] and ACID [21], maintaining high fidelity with minimal degradation as the target resolution scales from 512×512 to 1024×1024 using the same 256×256 inputs. These results confirm that our approach remains robust against variations in rendering scale, consistently delivering high-quality reconstructions regardless of the target resolution.

Efficiency and Cost-performance. We assess efficiency at a fixed 512×512 target by comparing pixel-aligned baselines against our configuration. While PixelSplat [3], MVSplat [5], and DepthSplat [34] deliver strong image quality, they incur significant latency and memory overhead due to their full-resolution input requirements. In particular, HiSplat [30] achieves competitive reconstruction fi-

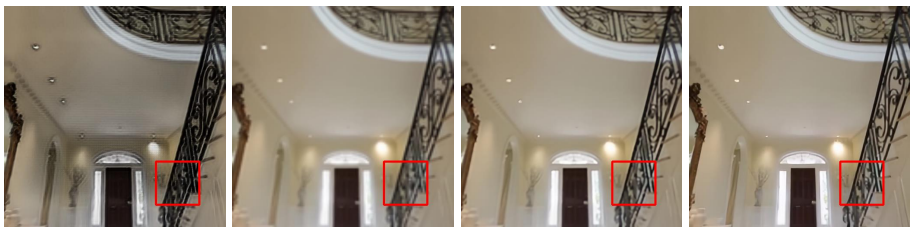

Fig. 5: Qualitative Comparison with Image-Space Upsamplers. All methods render at 512×512 resolution starting from the same 256×256 inputs. While image-space upsamplers fail to recover structural details, SubSplat restores sharp textures and geometric consistency by generating high-density primitives directly in 3D space.

Table 3: Comparison with image-space upsamplers on RealEstate10K [47]. All methods reconstruct from 256×256 inputs and render at 512×512 resolution. SubSplat significantly outperforms 2D post-processing methods by directly densifying geometry in 3D space, maintaining high fidelity with negligible latency overhead.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	Latency ↓ (s)
MVSplat + Bilinear (post)	24.01	0.801	0.237	**0.042**
MVSplat + HiT-SR [45] (post)	23.88	0.807	0.203	0.215
Ours (SubSplat)	**25.52**	**0.850**	**0.167**	**0.042**

delity by extracting Gaussians across three distinct feature stages. However, this multi-stage feed-forward process leads to substantial latency degradation (1960ms), as shown in Table 2. In contrast, SubSplat achieves 25.52 PSNR and 0.850 SSIM at only 42ms per frame (24 FPS) by effectively restoring density from lower-resolution features. This makes it the only method among the evaluated baselines to achieve real-time performance. Note that baselines were not evaluated with 1024×1024 inputs due to Out-of-Memory (OOM) issues. These results demonstrate a superior cost-performance trade-off for our SPGR design, sustaining high-fidelity rendering with significantly lower computational and memory budgets.

Comparison with Image-space Upsamplers. We compare our approach against image-space upsamplers, including Bilinear interpolation and a $\times 2$ Super Resolution (SR) head [45]. While Bilinear interpolation remains blurry and the SR head introduces ringing and cross-view inconsistencies, SubSplat maintains structural clarity at 512×512 resolution. As 2D upsamplers operate purely in image space, they cannot resolve the underlying grid-anchored primitive limitations. In contrast, SubSplat optimizes Gaussian distributions via sub-pixel reparameterization, capturing geometry-aware details that image-space methods fail to recover. Quantitatively (Table 3), post-processing upsamplers are inferior to our configuration. SubSplat yields higher PSNR/SSIM and lower LPIPS with latency on par with Bilinear interpolation, while being significantly faster than the SR head. This demonstrates that image-space upsampling cannot compen-

Table 4: Ablation Study on RealEstate10K. We evaluate the effectiveness of each module at $\times 2$ ($K = 4$) and $\times 4$ ($K = 16$) scales. SPGR provides the primary performance gain, while Feature Aggregation ensures structural consistency as the scale increases.

Modules	$K = 4$ (512×512)			$K = 16$ (1024×1024)		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Baseline (MVSplat [5])	19.46	0.751	0.269	16.98	0.671	0.424
+ Sub-pixel Reparam.	25.15	0.843	0.173	19.49	0.736	0.415
+ Feature Aggregation (Full)	**25.52**	**0.850**	**0.167**	**22.65**	**0.781**	**0.268**

Fig. 6: Ablation study of SubSplat modules across subdivision scales. We evaluate the effectiveness of each module. SPGR provides the primary performance gain, while Feature Aggregation ensures structural consistency as the scale increases.

sate for under-resolved primitives, whereas our method delivers cleaner and more temporally stable results without additional post-processing.

Table 5: Ablation of Deformable Attention (DA) stages on RealEstate10K [47]. We evaluate the impact of incremental DA stages on scene reconstruction quality under the 256×256 to 512×512 configuration. Increasing the attention stages consistently improves structural fidelity and perceptual quality.

Configuration	PSNR ↑	SSIM ↑	LPIPS ↓
One-stage DA	23.94	0.833	0.194
Two-stage DA	24.52	0.839	0.183
Three-stage DA (Full)	**25.52**	**0.850**	**0.167**

4.3 Ablation Study

We evaluate the contribution of SubSplat modules across subdivision scales $\times 2$ ($K = 4$) and $\times 4$ ($K = 16$). While SPGR acts as the primary recovery mechanism by providing a +5.69 dB PSNR boost at $K = 4$, the significance of Feature Aggregation grows with task difficulty, increasing its contribution to +3.16 dB

at the $K = 16$. As illustrated in Fig. 6, this synergy is effective for restoring high-frequency textures and preserving structural clarity at larger scales. Additionally, Table 5 shows the impact of Deformable Attention (DA) stages. Performance improves with more stages as iterative feature sampling better captures spatial details, where a three-stage configuration yields a 1.58 dB PSNR boost over the one-stage baseline.

5 Conclusion

We have proposed SubSplat, a novel framework that achieves high-fidelity rendering by generating fine-grained sub-pixel primitives directly from low-resolution features. This mechanism mitigates the quadratic cost of backbones with high-resolution inputs while maintaining stable network latency across extended output scales. Although rendering overhead increases with primitive count, SubSplat delivers a superior quality-efficiency trade-off and real-time performance. Future work will explore content-aware primitive density control to further optimize the balance between rendering throughput and structural fidelity.

Acknowledgements

This research was supported by the Basic Science Research Program of the National Research Foundation (NRF) funded by the Korean government (MSIT) (No. IITP-2026-RS-2024-00346737), by the NRF grant funded by the Korea government(MSIT) (No. RS-2025-00563942), by the IITP grant funded by the Korea government(MSIT)(No. IITP-2026-RS-2020-II201819, 10%), and by the AI Computing Infrastructure Enhancement (GPU Rental Support) User Support Program funded by the Ministry of Science and ICT (MSIT), Republic of Korea.

References

1. Barron, J.T., Mildenhall, B., Srinivasan, P.P., Tancik, M., Hedman, P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
2. Botsch, M., Kobbelt, L.: High-quality point-based rendering on modern gpus. In: 11th Pacific Conference onComputer Graphics and Applications, 2003. Proceedings. pp. 335–343. IEEE (2003)
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, A., Xu, H., Esposito, S., Tang, S., Geiger, A.: Lara: Efficient large-baseline radiance fields. In: European conference on computer vision. pp. 338–355. Springer (2024)
5. 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)

6. Chen, Z., Funkhouser, T., Hedman, P., Tagliasacchi, A.: Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16569–16578 (2023)
7. Collins, R.T.: A space-sweep approach to true multi-image matching. In: Proceedings CVPR IEEE computer society conference on computer vision and pattern recognition. pp. 358–363. Ieee (1996)
8. Fang, G., Wang, B.: Mini-splatting: Representing scenes with a constrained number of gaussians. In: European conference on computer vision. pp. 165–181. Springer (2024)
9. Fei, X., Zheng, W., Duan, Y., Zhan, W., Tomizuka, M., Keutzer, K., Lu, J.: Pixelgaussian: Generalizable 3d gaussian reconstruction from arbitrary views. arXiv preprint arXiv:2410.18979 (2024)
10. Feng, X., He, Y., Wang, Y., Yang, Y., Li, W., Chen, Y., Kuang, Z., Fan, J., Jun, Y., et al.: Srgs: Super-resolution 3d gaussian splatting. arXiv preprint arXiv:2404.10318 (2024)
11. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
12. Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2d gaussian splatting for geometrically accurate radiance fields. In: ACM SIGGRAPH 2024 conference papers. pp. 1–11 (2024)
13. Huang, X., Li, W., Hu, J., Chen, H., Wang, Y.: Refsr-nerf: Towards high fidelity and super resolution view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8244–8253 (2023)
14. Huang, Y., Zheng, W., Zhang, Y., Zhou, J., Lu, J.: Gaussianformer: Scene as gaussians for vision-based 3d semantic occupancy prediction. In: European Conference on Computer Vision. pp. 376–393. Springer (2024)
15. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., et al.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
16. Kulhanek, J., Peng, S., Kukelova, Z., Pollefeys, M., Sattler, T.: Wildgaussians: 3d gaussian splatting in the wild. arXiv preprint arXiv:2407.08447 (2024)
17. Lee, J.L., Li, C., Lee, G.H.: Disr-nerf: Diffusion-guided view-consistent super-resolution nerf. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20561–20570 (2024)
18. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: Swinir: Image restoration using swin transformer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1833–1844 (2021)
19. Liang, Z., Zhang, Q., Hu, W., Zhu, L., Feng, Y., Jia, K.: Analytic-splatting: Anti-aliased 3d gaussian splatting via analytic integration. In: European conference on computer vision. pp. 281–297. Springer (2024)
20. Lin, C.Y., Fu, Q., Merth, T., Yang, K., Ranjan, A.: Fastsr-nerf: Improving nerf efficiency on consumer devices with a simple super-resolution pipeline. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 6036–6045 (2024)
21. 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)
22. Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B.: Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20654–20664 (2024)

23. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
24. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM transactions on graphics (TOG) **41**(4), 1–15 (2022)
25. Nam, S., Sun, X., Kang, G., Lee, Y., Oh, S., Park, E.: Generative densification: Learning to densify gaussians for high-fidelity generalizable 3d reconstruction. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 26683–26693 (2025)
26. Rematas, K., Liu, A., Srinivasan, P.P., Barron, J.T., Tagliasacchi, A., Funkhouser, T., Ferrari, V.: Urban radiance fields. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12932–12942 (2022)
27. Rota Bulò, S., Porzi, L., Kontschieder, P.: Revising densification in gaussian splatting. In: European Conference on Computer Vision. pp. 347–362. Springer (2024)
28. Rusinkiewicz, S., Levoy, M.: Qsplat: A multiresolution point rendering system for large meshes. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques. pp. 343–352 (2000)
29. Sola, J.: Quaternion kinematics for the error-state kalman filter. arXiv preprint arXiv:1711.02508 (2017)
30. Tang, S., Ye, W., Ye, P., Lin, W., Zhou, Y., Chen, T., Ouyang, W.: Hisplat: Hierarchical 3d gaussian splatting for generalizable sparse-view reconstruction. arXiv preprint arXiv:2410.06245 (2024)
31. Wan, Y., Shao, M., Cheng, Y., Zuo, W.: S2gaussian: Sparse-view super-resolution 3d gaussian splatting. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 711–721 (2025)
32. Wang, C., Wu, X., Guo, Y.C., Zhang, S.H., Tai, Y.W., Hu, S.M.: Nerf-sr: High quality neural radiance fields using supersampling. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 6445–6454 (2022)
33. Xie, S., Wang, Z., Wang, X., Zhu, Y., Pan, C., Dong, X.: Supergs: Super-resolution 3d gaussian splatting enhanced by variational residual features and uncertainty-augmented learning. arXiv preprint arXiv:2410.02571 (2024)
34. 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)
35. Xu, H., Zhang, J., Cai, J., Rezatofighi, H., Yu, F., Tao, D., Geiger, A.: Unifying flow, stereo and depth estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(11), 13941–13958 (2023)
36. Yan, Z., Low, W.F., Chen, Y., Lee, G.H.: Multi-scale 3d gaussian splatting for anti-aliased rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20923–20931 (2024)
37. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: Proceedings of the European conference on computer vision (ECCV). pp. 767–783 (2018)
38. Yoon, Y., Yoon, K.J.: Cross-guided optimization of radiance fields with multi-view image super-resolution for high-resolution novel view synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12428–12438 (2023)
39. Younes, M., Boukhayma, A.: Anti-aliased 2d gaussian splatting. Advances in Neural Information Processing Systems **38**, 9767–9792 (2026)

40. Yu, X., Zhu, H., He, T., Chen, Z.: Gaussiansr: 3d gaussian super-resolution with 2d diffusion priors. arXiv preprint arXiv:2406.10111 (2024)
41. Yu, Z., Chen, A., Huang, B., Sattler, T., Geiger, A.: Mip-splatting: Alias-free 3d gaussian splatting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19447–19456 (2024)
42. Zhang, C., Zou, Y., Li, Z., Yi, M., Wang, H.: Transplat: Generalizable 3d gaussian splatting from sparse multi-view images with transformers. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 9869–9877 (2025)
43. Zhang, K., Bi, S., Tan, H., Xiangli, Y., Zhao, N., Sunkavalli, K., Xu, Z.: Gs-lrm: Large reconstruction model for 3d gaussian splatting. In: European Conference on Computer Vision. pp. 1–19. Springer (2024)
44. 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)
45. Zhang, X., Zhang, Y., Yu, F.: Hit-sr: Hierarchical transformer for efficient image super-resolution. In: European conference on computer vision. pp. 483–500. Springer (2024)
46. Zhang, Z., Hu, W., Lao, Y., He, T., Zhao, H.: Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting. In: European Conference on Computer Vision. pp. 326–342. Springer (2024)
47. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. arXiv preprint arXiv:1805.09817 (2018)
48. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)
49. Zwicker, M., Pfister, H., Van Baar, J., Gross, M.: Surface splatting. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. pp. 371–378 (2001)
50. Zwicker, M., Pfister, H., Van Baar, J., Gross, M.: Ewa splatting. IEEE Transactions on Visualization & Computer Graphics **8**(03), 223–238 (2002)