SRSplat: Feed-Forward Super-Resolution Gaussian Splatting from Sparse Multi-View Images

Xinyuan Hu[1 *], **Changyue Shi**[1 2 *], **Chuxiao Yang**[1], **Minghao Chen**[1], **Jiajun Ding**[1 †], **Tao Wei**[3], **Chen Wei**[3], **Zhou Yu**[1], **Min Tan**[1]

[1]School of Computer Science and Technology, Hangzhou Dianzi University,
[2]School of AI for Science, Peking University,
[3]Li Auto Inc.
{xinyuan, shicy, chuxiao_yang, djj}@hdu.edu.cn

Abstract

Feed-forward 3D reconstruction from sparse, low-resolution (LR) images is a crucial capability for real-world applications, such as autonomous driving and embodied AI. However, existing methods often fail to recover fine texture details. This limitation stems from the inherent lack of high-frequency information in LR inputs. To address this, we propose **SRSplat**, a feed-forward framework that reconstructs high-resolution 3D scenes from only a few LR views. Our main insight is to compensate for the deficiency of texture information by jointly leveraging external high-quality reference images and internal texture cues. We first construct a scene-specific reference gallery, generated for each scene using Multimodal Large Language Models (MLLMs) and diffusion models. To integrate this external information, we introduce the *Reference-Guided Feature Enhancement (RGFE)* module, which aligns and fuses features from the LR input images and their reference twin image. Subsequently, we train a decoder to predict the Gaussian primitives using the multi-view fused feature obtained from *RGFE*. To further refine predicted Gaussian primitives, we introduce *Texture-Aware Density Control (TADC)*, which adaptively adjusts Gaussian density based on the internal texture richness of the LR inputs. Extensive experiments demonstrate that our SRSplat outperforms existing methods on various datasets, including RealEstate10K, ACID, and DTU, and exhibits strong cross-dataset and cross-resolution generalization capabilities.

Project Page — https://xinyuanhu66.github.io/SRSplat/

1 Introduction

Reconstructing 3D scenes from 2D images is a fundamental task in computer vision and computer graphics (Mildenhall et al. 2021; Chen et al. 2022; Kerbl et al. 2023; Shi et al. 2025c,a; Müller et al. 2022). It plays a crucial role in various applications, such as embodied AI (Huang et al. 2023a) and autonomous driving (Tian et al. 2025). However, in these real-world applications, traditional 3D reconstruction methods like NeRF (Mildenhall et al. 2021) or 3DGS (Kerbl et al. 2023) face several key challenges: 1) ***Resolution Constraints:*** Due to hardware and sensor limitations, acquiring sufficiently high-resolution (HR) images for accurate 3D reconstruction is often impractical. 2) ***Few-Shot Challenges:*** In practice, acquiring high-quality dense views is prohibitively costly and impractical. 3) ***Real-Time Reconstruction:*** Fields like robotics require real-time 3D reconstruction, necessitating efficient feed-forward algorithms.

Our goal is to build a feed-forward 3D reconstruction framework that simultaneously tackles the three practical hurdles so that high-quality 3D scenes can be reconstructed from only limited LR images. The recent emergence of feed-forward 3DGS (Charatan et al. 2024; Chen et al. 2024a; Zhang et al. 2025; Tang et al. 2024; Chen et al. 2024b; Fei et al. 2024) methods has revolutionized 3D scene reconstruction through their real-time reconstruction capabilities. These methods leverage feed-forward networks to make direct 3D Gaussian predictions, eliminating the need for per-scene optimization. However, when provided with low-resolution (LR) input images, existing methods struggle to reconstruct high-quality scenes and often exhibit a loss of texture details. This limitation stems from the inherent lack of high-frequency texture information in LR images compared to their HR counterparts (Feng et al. 2024; Yang et al. 2025).

To this end, we propose **SRSplat**, a novel feed-forward framework that reconstructs HR 3D scenes from sparse LR views. Our main insight is to compensate for the deficiency of texture information by jointly leveraging external high-quality reference images and internal texture cues. Inspired by previous reference-based 2D image super-resolution (SR) methods (Lu et al. 2021; Sun et al. 2024; Yang et al. 2020; Bösiger et al. 2024; Jiang et al. 2021; Cao et al. 2022), we first construct a scene-specific *Reference Gallery*. In this gallery, each reference image is generated as a twin image of its corresponding input scene. Specifically, given a scene, we first employ Multimodal Large Language Models (MLLMs) (Achiam et al. 2023) to generate a concise semantic description. The resulting description is then used to prompt a pre-trained diffusion model (Labs et al. 2025) to synthesize high-quality reference twins. Our novel reference twin generation technique synthesizes images that mir-

*These authors contributed equally.
†Corresponding author.
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Figure 1: In this work, we propose SRSplat, a novel feed-forward framework that reconstructs high-quality 3D scenes with only sparse and LR input views. SRSplat demonstrates superior performance and is capable of handling low-resolution, sparse-view inputs and real-time reconstruction, thereby offering greater functionality and practicality in realistic applications.

ror the target scene's characteristics but in a HR way, supplying more high-frequency cues for the subsequent process.

With the *Reference Gallery*, we propose the *Reference-Guided Feature Enhancement (RGFE)* module to integrate this external information. For the input images and their corresponding HR reference twins, we first use a shared CNN for multi-scale feature extraction. Then *RGFE* performs coarse-to-fine correspondence matching for the extracted features. Finally, a fuse network is used to map the distribution of reference features to LR features, thereby transferring high-frequency information from the reference twins. The resulting fused features are then fed to a decoder to predict the Gaussian primitives. However, since the proposed Gaussian prediction decoder predicts a single Gaussian primitive for each pixel, texture-rich regions are difficult to optimize. To further refine predicted Gaussian primitives, we propose *Texture-Aware Density Control (TADC)*. Specifically, *TADC* builds a learnable texture richness perceptron that perceives texture richness from LR inputs. Based on the internal texture richness, Gaussian primitives can adaptively adjust the density in the scene.

Our contributions can be summarized as follows:

- We propose SRSplat, a feed-forward framework that generates HR 3D scenes from sparse LR 2D images. To the best of our knowledge, this is the first work to solve this task in a feed-forward manner.
- We construct a scene-specific reference twin gallery using MLLMs and diffusion priors. We propose the *RGFE* and *TADC* module to compensate for the deficiency of texture information by leveraging external reference image and internal texture richness map.
- Experimental results on various public datasets demonstrate that SRSplat outperforms existing methods and exhibits strong cross-dataset and cross-resolution generalization capabilities.

2 Related Works

Feed-Forward 3D Reconstruction. Recently, 3DGS (Kerbl et al. 2023)-based feed-forward reconstruction has emerged as a key approach for efficient 3D scene representation and novel view synthesis. PixelSplat (Charatan et al. 2024) introduces a polar line-based Transformer architecture to model cross-view correspondence and predict depth distribution. MVSplat (Chen et al. 2024a) constructs cost volumes via plane sweeping to achieve enhanced geometric reconstruction accuracy. TranSplat (Zhang et al. 2025) adopts a Transformer-based architecture, combining monocular depth priors to refine the sparse-view reconstruction results. DepthSplat (Xu et al. 2025) presents a robust multi-view depth model by leveraging pre-trained monocular depth features (Yang et al. 2024), thereby enabling high-quality feed-forward reconstructions. Despite these advancements, existing methods often fail to capture high-frequency details from low-resolution inputs, leading to artifacts and reduced quality. To address these limitations, we propose SRSplat, a novel framework that enables super-resolution reconstruction with low-resolution inputs.

Super-Resolution Novel View Synthesis. Super-resolution novel view synthesis aims to reconstruct high-resolution 3D scenes from only low-resolution multi-view inputs. As a pioneer in the field, NeRF-SR (Wang et al. 2021) leverages the sub-pixel constraint to optimize a HR scenes. SRGS (Feng et al. 2024) is the first framework to synthesize HR novel views based on 3DGS using the SwinIR model (Liang et al. 2021), but suffers from multi-view inconsistencies. Recently, S2Gaussian (Wan et al. 2025) employs a multi-stage training paradigm to reconstruct high-resolution 3D scenes

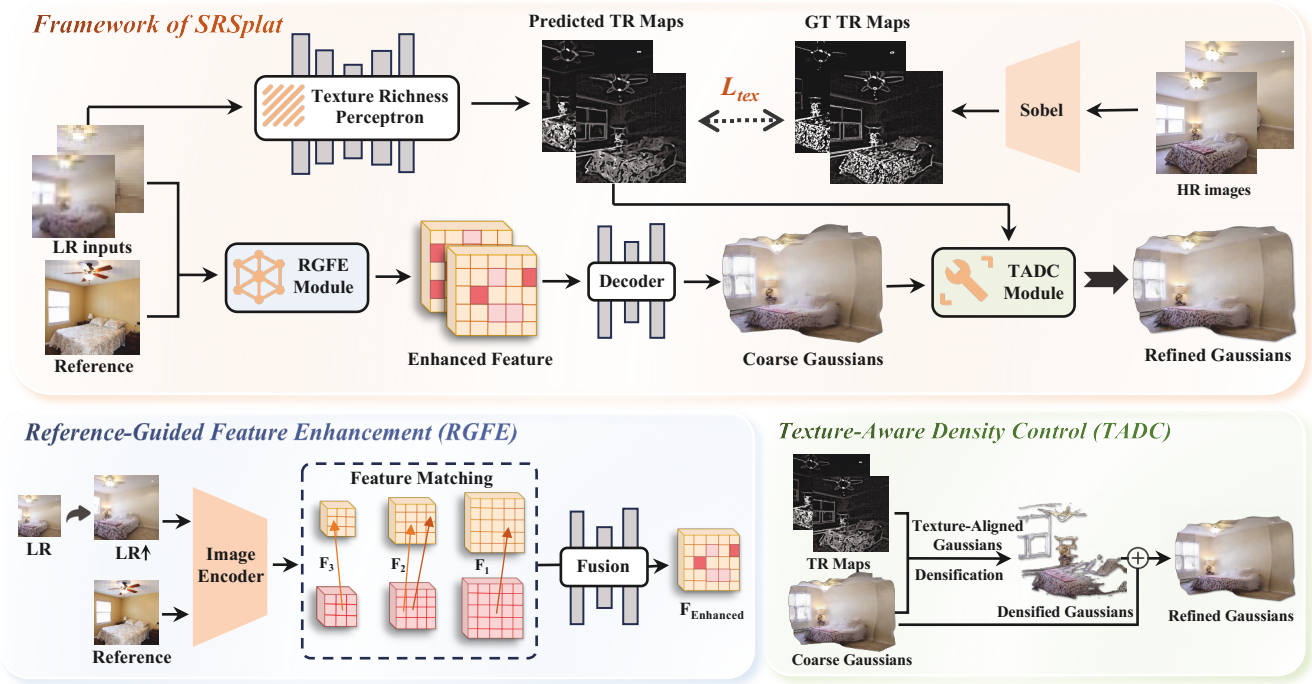

Figure 2: Framework of SRSplat. Our method takes LR images and their corresponding reference as inputs. The *RGFE* module first extracts multi-scale features and effectively fuses these features. Upon decoding the Gaussian primitives, *TADC* adjusts Gaussian density adaptively according to the richness of texture generated by a texture richness perceptron.

solely from only sparse and low-resolution input views. However, per-scene optimization methods rely on iterative optimization to achieve the final 3D representation and exhibit poor generalization ability. In contrast, feed-forward inference methods reconstruct the entire scene in a single feed-forward pass, demonstrating strong potential in super-resolution novel-view synthesis.

Reference-Based Super-Resolution. Reference-based SR methods aim to enhance low-resolution inputs by leveraging high-frequency details from high-quality reference images. MASA-SR (Lu et al. 2021) introduces attention mechanisms to adaptively fuse reference textures into LR feature representations. More recently, RefSR-NeRF (Huang et al. 2023b) further extends this idea to 3D by integrating reference images as auxiliary information. Yet, in real-world scenarios, it is often impractical to assume that every scene can be paired with a corresponding reference image. CoSeR (Sun et al. 2024) leverages prior knowledge from large-scale text-to-image diffusion models to synthesize high-quality reference images for super-resolution. This motivates the development of an automated reference-generation approach.

3 Framework of SRSplat

The overall framework of SRSplat is illustrated in Fig. 2. We first establish a reference gallery in Sec. 3.1. Then we employ *RGFE* to fuse features derived from input and reference images in Sec. 3.2. Based on the fused features, we train a decoder to predict Gaussian attributes in Sec. 3.3. Finally, we use *TADC* to adaptively adjust the density of Gaussian primitives according to texture intensity in Sec. 3.4.

3.1 Reference Gallery Preparation

Drawing inspiration from 2D reference-based SR methods (Lu et al. 2021; Bösiger et al. 2024), we aim to mitigate the high-frequency loss of LR input multi-view images with the information from external reference images. To achieve this, we first establish a reference gallery. Each image in this gallery serves as a twin of each 3D scene, mirroring its characteristic layouts. The process is illustrated in Fig. 3.

We first leverage GPT-4o to generate semantic descriptions of input LR multi-view images. For a set of LR input images (downsampled by a factor of P from training set) of a specific scene $\{I_i^{\mathrm{LR}}\}_{i=1}^{N}, I_i \in \mathbb{R}^{\frac{H}{P}\times\frac{W}{P}\times 3}$, we employ MLLM to directly process the LR image set and generate semantic descriptions $\mathcal{P} = \mathrm{MLLM}(\{I_i^{\mathrm{LR}}\}_{i=1}^{N})$. The generated descriptions $\mathcal{P}$ capture the overall layout of the scene, key objects and their relationships, providing clear semantic guidance for the subsequent process. With the semantic descriptions, we utilize a 2D diffusion model (Labs et al. 2025) $\mathcal{D}$ to generate a HR reference twin $r = \mathcal{D}(\mathcal{P}), r \in \mathbb{R}^{H\times W\times 3}$. By leveraging strong priors of MLLM and diffusion model, the reference twin faithfully preserves scene's layout and remains semantically aligned with the original inputs (as demonstrated in Fig. 4).

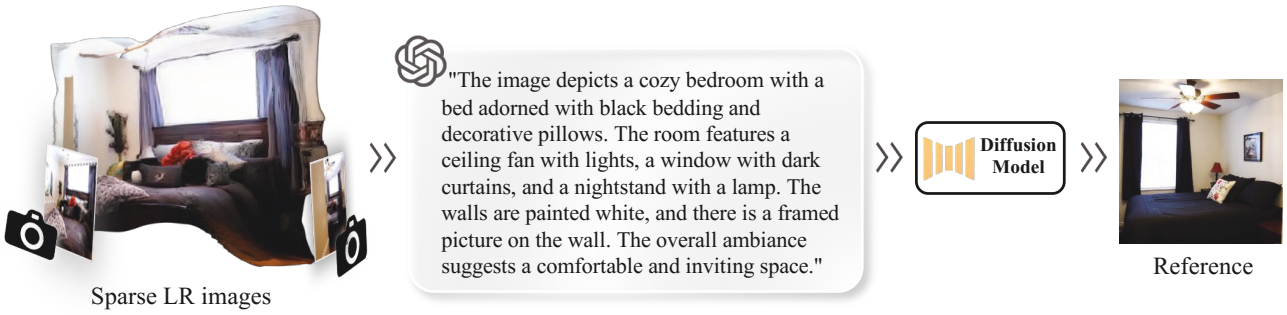

Figure 3: Pipeline of reference gallery generation. Given LR input images for each scene, the MLLM produces semantic descriptions. Subsequently, the diffusion model uses these descriptions to generate reference images tailored to the scene.

Figure 4: Reference gallery examples. Reference twin images share details similar to the LR images.

3.2 Reference-Guided Feature Enhancement

Multi-Scale Feature Extraction. For an LR image $I^{\text{LR}} \in \mathbb{R}^{\frac{H}{P} \times \frac{W}{P} \times 3}$, we employ a shared CNN encoder (Xu et al. 2022, 2023) $\mathcal{E}$ to extract multiscale features from both the upsampled input images $I\uparrow \in \mathbb{R}^{H \times W \times 3}$ and its corresponding reference $r \in \mathbb{R}^{H \times W \times 3}$. In practice, we employ three levels, each halving the resolution of the previous one. The encoded outputs are

$$\begin{cases} \{F_l^{\text{I}}\} = \mathcal{E}(I\uparrow), \\ \{F_l^{\text{ref}}\} = \mathcal{E}(r), \end{cases} \quad l = 1, 2, 3, \tag{1}$$

where F_l^{I}, $F_l^{\text{ref}} \in \mathbb{R}^{H_l \times W_l \times C^{feature}}$ denote the features at scale l with resolution $H_l \times W_l$ and channel $C^{feature}$.

Feature Matching and Fusion. Following previous researches (Lu et al. 2021; Bösiger et al. 2024), SRSplat matchs the features between input and reference features using cosine similarity. Specifically, it first performs coarse-to-fine matching, starting with a coarse grid using a stride, followed by dense matching within a fixed-size window around the initial correspondences. This procedure yields a mapping m from input feature indices to corresponding reference feature indices:

$$m_{I \to ref}^{l} : (x, y) \in F_l^I \longmapsto \big\{(u, v) \in F_l^{ref},\ s \in \mathbb{R}\big\}, \tag{2}$$

where (u, v) is the best match coordinate in the reference for the input coordinate (x, y) and s is the corresponding match score. Next, we warp the reference features according to $m_{I \to ref}^{l}$ and weight them by their match scores, yielding the warped reference features $\{F_1^{\text{ref}\to\text{I}}, F_2^{\text{ref}\to\text{I}}, F_3^{\text{ref}\to\text{I}}\}$. Weighting the warped features by their matching scores reduces low-confidence correspondences. This enables the model to use the reference features only when they have a confident match. Finally, we fuse the warped reference features $F_l^{\text{ref}\to\text{I}}$ with the image features F_l^{I} using a learnable fusion network $\mathcal{H}$:

$$F_{enhanced} = \mathcal{H}\big(\{F_l^I\}, \{F_l^{ref \to I}\}\big), \tag{3}$$

where $F_{enhanced} \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times C^{enhanced}}$ represents the final enhanced feature. To further enhance information exchange across views, we feed the features into the Swin Transformer (Liu et al. 2021b), which employs both cross-attention and self-attention mechanisms.

3.3 Depth Estimation and Gaussian Prediction

We utilize the cost volume matching in Multi-View Stereo (MVS) (Yao et al. 2018; Cao, Ren, and Fu 2022) for depth estimation. Specifically, we first construct a depth candidate $d_{\text{cand}}^{i} \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times D}$ using the plane-sweep stereo approach (Yao et al. 2018). Then, we warp the feature of the j-th view F^j into the i-th view via the two camera projection matrices P^i and $P^j \in \mathbb{R}^{4 \times 4}$, producing

$$F_{\text{Warp}}^{ij} = \text{Warp}\big(F^j,\ P^i,\ P^j,\ d_{\text{cand}}^{i}\big), \tag{4}$$

where Warp denotes the warping operation (Xu et al. 2023) and D denotes the depth dimension.

We then obtain the cost volume by computing the dot product between F^i and F_{Warp}^{ij}:

$$C^i = \frac{F^i \otimes F_{\text{Warp}}^{ij}}{\sqrt{C}}, \tag{5}$$

Figure 5: Error maps show the intensity of inconsistency between the rendered image and the ground truth. We observe that Regions with high texture richness are often under-optimized. Therefore, we propose *TADC* dynamically control the density of Gaussians according to texture richness.

where $\otimes$ denotes element-wise multiplication followed by summation over the channel dimension and a CNN-based upsampler.

Finally, denoting the vector of depth candidates by $G = [d_1, d_2, \ldots, d_D] \in \mathbb{R}^D$, we obtain the depth map:

$$D^i = \text{softmax}\big(C^i\big)\, G, \quad D^i \in \mathbb{R}^{H \times W}. \tag{6}$$

We follow previous research (Chen et al. 2024a; Charatan et al. 2024) to predict Gaussian parameters G_{coarse}, including the Gaussian center μ_i, opacity α_i, covariance Σ_i, and color c_i. Specifically, we utilize the depth map D^i to unproject the 2D pixel to the 3D space location as the Gaussian center and predict opacity using a simple MLP layer. We calculate color from the predicted spherical harmonic coefficients and use a scaling matrix s and a rotation matrix $R(\theta)$ to represent the covariance matrix.

$$\Sigma_i = R(\theta)^\top \,\text{diag}(s)\, R(\theta). \tag{7}$$

3.4 Texture-Aware Density Control

How to Represent Texture Richness? Regions with significant color variation (i.e., large color gradients) contain rich texture details (Hu et al. 2025; Shi et al. 2025b). To provide an estimate of texture richness, we apply a high-pass filter (Sobel in this paper) to compute the first-order finite differences of pixel intensities. Given an RGB image $I \in \mathbb{R}^{H \times W \times 3}$, we compute two gradient maps by convolving I with the standard horizontal and vertical Sobel operators, respectively:

$$T_x = I * \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, T_y \quad = I * \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}. \tag{8}$$

We then compute the gradient magnitude, thereby deriving the texture richness map $TR = \sqrt{(T_x)^2 + (T_y)^2}$.

Observations. The proposed Gaussian prediction module predicts one Gaussian primitive for each pixel. However, we observed that regions with high texture richness are often under-optimized and show large inconsistencies with the ground truth in error maps (see Fig. 5). More Gaussians primitives are required to fit these texture-rich areas (Hu et al. 2025). To this end, we propose *Texture-Aware Density Control (TADC)*, a module that dynamically controls the density of Gaussians according to texture richness.

Texture Richness Perceptron for LR Images. To begin with, we need to recognize where the texture-rich regions are from the 2D LR images. To achieve this, we carefully designed the *Texture Richness Perceptron* $\mathcal{E}_{tex}$, which is implemented using a convolutional neural network. Given an LR input image I^{LR}, we predict its texture richness map $\hat{TR} = \mathcal{E}_{tex}(I^{\text{LR}})$. To optimize $\mathcal{E}_{tex}$, we employ the texture richness map TR obtained as described above as supervisory signal:

$$\mathcal{L}_{\text{tex}} = \mathcal{L}_1\big(\hat{TR}, TR\big)\,. \tag{9}$$

Density Control. After obtaining the predicted Gaussian set G_{coarse} and the texture richness map $\hat{TR}$, we select the Gaussian primitives set G_{den} exhibiting the high texture richness. Each selected Gaussian primitive is further decomposed into multiple finer primitives.

Specifically, their positions and features, denoted as $\mathcal{X}_{\text{den}}$ and $\mathcal{F}_{\text{den}}$, are input to a dedicated densification network $\mathcal{N}_{\text{dens}}$ that predicts the upsampled positions and features (Nam et al. 2024). These upsampled positions and features are subsequently passed to a *HEAD* module, where they are converted into fine Gaussian parameters. The set of densified Gaussian primitive is defined as:

$$G_{\text{dense}} = \mathit{HEAD}(\mathcal{N}_{\text{dens}}(\mathcal{X}_{\text{den}}, \mathcal{F}_{\text{den}}), G_{den}). \tag{10}$$

Finally, we obtain the refined Gaussian primitives G_{refine}:

$$G_{\text{refine}} = G_{\text{dense}} \cup G_{\text{coarse}} \tag{11}$$

3.5 Training Loss

During training, we directly supervise the quality of the novel RGB images using Mean Squared Error (MSE) and Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al. 2018) losses. We also introduce $\mathcal{L}_{\text{tex}}$ to supervise the texture richness perceptron. The total training loss is:

$$\mathcal{L} = \lambda_{\text{mse}}\,\mathcal{L}_{\text{mse}} + \lambda_{\text{lpips}}\,\mathcal{L}_{\text{lpips}} + \lambda_{\text{tex}}\,\mathcal{L}_{\text{tex}}, \tag{12}$$

where λ_{mse}, λ_{lpips} and λ_{tex} are balancing weights.

4 Experiments

4.1 Experimental Settings

Datasets. Our model is trained on the large-scale RealEstate10K (Zhou et al. 2018) and ACID (Liu et al. 2021a) datasets. We evaluate our method on the RealEstate10K, ACID, and DTU datasets. We downsample the original training sets by factors of 2, 4, and 8 for training and evaluation. Following MVSplat (Chen et al. 2024a), the model is trained with two context views, and all methods are evaluated on three novel target views. For the DTU dataset, results are reported on 16 validation scenes, each with four novel views.

Baselines and metrics. We compare SRSplat with typical methods in scene-level novel view synthesis, including pixelSplat (CVPR24) (Charatan et al. 2024),

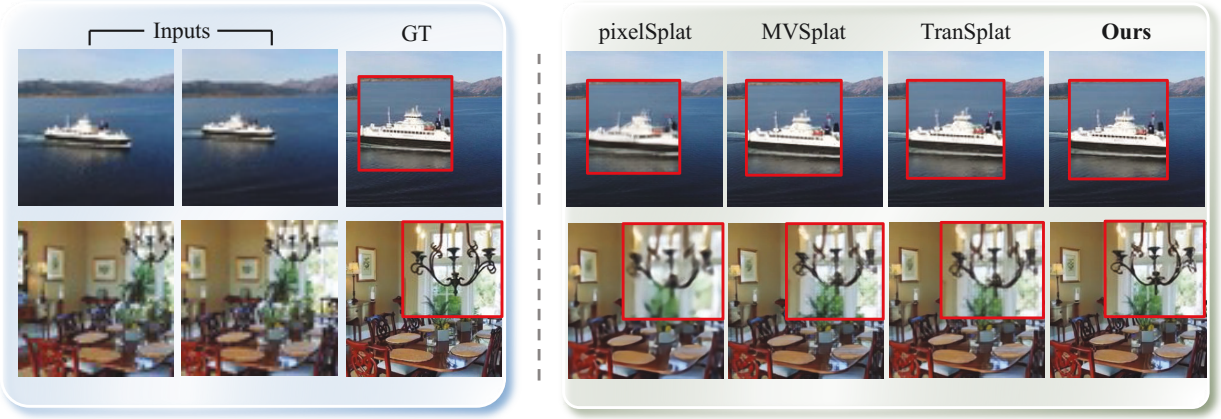

Figure 6: Qualitative comparison of novel views on RealEstate10k and ACID datasets. Compared with baseline methods, our approach yields sharper high-resolution novel views with cleaner fine-grained details (e.g., the chandelier and wall painting).

Datasets	Methods	1/2 Resolution			1/4 Resolution			1/8 Resolution		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
RealEstate10K	pixelSplat (Charatan et al. 2024)	24.56	0.805	0.254	22.80	0.720	0.358	20.64	0.618	0.491
	MVSplat (Chen et al. 2024a)	24.06	0.812	0.177	22.20	0.719	0.262	20.32	0.614	0.375
	TranSplat (Zhang et al. 2025)	24.60	0.827	0.168	22.53	0.727	0.256	20.51	0.623	0.369
	SRSplat (Ours)	**25.20**	**0.844**	**0.152**	**23.99**	**0.802**	**0.189**	**21.98**	**0.712**	**0.267**
ACID	pixelSplat (Charatan et al. 2024)	26.56	0.780	0.284	24.81	0.687	0.405	22.97	0.603	0.527
	MVSplat (Chen et al. 2024a)	26.18	0.784	0.192	24.25	0.675	0.282	22.61	0.583	0.377
	TranSplat (Zhang et al. 2025)	26.25	0.786	0.193	24.34	0.677	0.278	22.57	0.582	0.378
	SRSplat (Ours)	**27.39**	**0.823**	**0.163**	**25.62**	**0.757**	**0.218**	**23.79**	**0.660**	**0.288**

Table 1: Quantitative comparison under different input resolutions. SRSplat surpass all baseline methods in terms of PSNR, SSIM, and LPIPS. (**Bold** figures indicate the best and underlined figures indicate the second best)

Methods	Re10k→DTU			Re10k→ACID		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
pixelSplat	12.66	0.367	0.577	24.70	0.679	0.416
MVSplat	13.75	0.410	0.511	24.11	0.677	0.297
TranSplat	13.56	0.391	0.532	24.22	0.675	0.296
Ours	**13.80**	**0.417**	**0.445**	**25.55**	**0.754**	**0.226**

Table 2: Quantative comparisons of cross-dataset generalization. We conduct zero-shot evaluations on the ACID and DTU datasets with models trained on RealEstate10K.

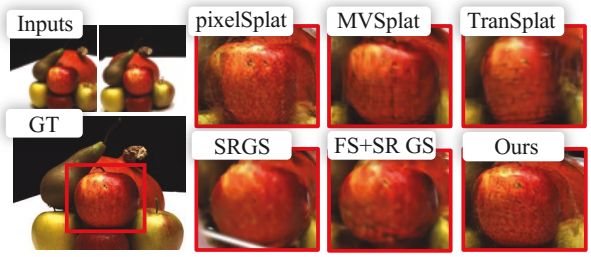

Figure 7: The qualitative comparisons on DTU datasets.

MVSplat (ECCV24) (Chen et al. 2024a), and TranSplat (AAAI25) (Zhang et al. 2025). We also compare the per-scene optimization method SRGS (Feng et al. 2024) and FSGS (Zhu et al. 2024) on DTU dataset. In the following sections, we report the average PSNR, SSIM (Wang et al. 2004), and LPIPS (Zhang et al. 2018) for all baselines.

Implementation. We implement SRSplat using the PyTorch framework. For each scene, we downsample images by factors of 2, 4, and 8 to create LR inputs. Training is conducted with a batch size of 10 across 5 NVIDIA RTX 4090 GPUs using the Adam (Kingma 2014) optimizer for 10k iterations, requiring approximately 2 days.

4.2 Main Results

Quantitative Results. The quantitative results are presented in Tab. 1. SRSplat achieves state-of-the-art performance across all visual quality metrics on both RealEstate10K and ACID benchmarks. Notably, as the input image resolution decreases, the reconstruction performance of previous methods degrade significantly, while SRSplat maintains comparatively strong metrics. This is because our proposed method effectively preserves and enhances high-frequency texture representations under LR inputs.

Qualitative Results. Fig. 6 presents the visualization results of SRSplat and other methods. SRSplat achieves superior quality on novel view images across various challeng-

Methods	PSNR↑	SSIM↑	LPIPS↓	Time↓
SRGS (Feng et al. 2024)	12.42	0.327	0.598	300s
SRGS + FSGS (Zhu et al. 2024)	13.72	**0.444**	0.481	420s
Ours	**13.80**	0.417	**0.445**	**0.2s**

Table 3: Quantitative results comparing with the per-scene optimization method.

Methods	Setting(a)			Setting(b)		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
pixelSplat	21.38	0.642	0.415	21.63	0.704	0.365
MVSplat	21.04	0.643	0.289	20.52	0.686	0.289
TranSplat	21.24	0.652	0.279	20.88	0.698	0.286
Ours	**22.07**	**0.727**	**0.219**	**21.74**	**0.711**	**0.280**

Table 4: Quantative comparisons of cross-resolution generalization. Setting (a): train on 1/8 resolution inputs with original-resolution supervision, test with 4× upsampling. Setting (b): train on 1/4 resolution inputs with 1/2 resolution supervision, test with 4× upsampling.

ing scenes. In the first row (large smooth regions like the open water band), our approach reconstructs sharper object boundaries (ship hull and mast) despite scarce intrinsic texture. In texture-rich regions (second row), our *TADC* module adaptively densifies Gaussians, yielding cleaner texture details while other methods blur them.

4.3 Other Results

Cross-dataset generalization. To verify the cross-dataset generalization ability of SRSplat, we train the model on the RealEstate10K dataset downsampled by a factor of 4 and directly evaluate it on the ACID and DTU datasets. The quantitative results are presented in Tab. 2. Compared with previous methods, SRSplat achieves significant improvements in generalization, obtaining +0.85 dB PSNR, +0.075 SSIM, and –0.07 LPIPS on the ACID dataset compared to suboptimal baselines. As depicted in Fig. 7, the images generated by SRSplat exhibit finer texture details and less blurriness.

Cross-resolution generalization. We assess cross-resolution generalization in two experimental configurations. First, we train on RealEstate10K images downsampled by a factor of 8, using original-resolution (1×) ground truth as supervision, and then perform 4× super-resolution during testing. Second, we train on inputs downsampled by a factor of 4, supervised by the 2×-downsampled ground truth, and then apply 4× super-resolution during evaluation. As shown in Tab. 4, SRSplat exhibits superior cross-resolution generalization compared to baselines.

Comparison with the per-scene optimization method. We further compare SRSplat with the representative super-resolution method SRGS (Feng et al. 2024) and an enhanced version of SRGS with FSGS (Zhu et al. 2024) on the DTU dataset. Qualitative results are shown in Fig. 7 and the quantitative results are reported in Tab. 2. Compared to per-scene optimization methods, SRSplat not only achieves competitive performance but also enables real-time reconstruction.

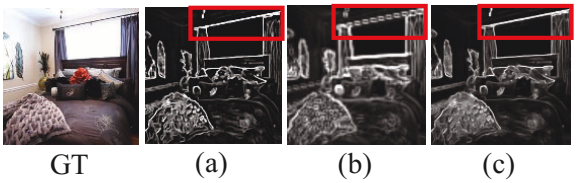

Figure 8: Comparison of texture-richness (TR) maps. (a) TR map obtained by applying the Sobel to the HR GT image; (b) TR map extracted by Sobel on the LR image; (c) TR map perceived by our TR perceptron from the LR image.

FGFE	*TADC*	PSNR↑	SSIM↑	LPIPS↓
		22.20	0.719	0.262
✓		23.56	0.782	0.207
✓	✓	**23.99**	**0.802**	**0.189**
w/o TR perceptron		23.75	0.795	0.192

Table 5: Ablation study on RealEstate10K.

4.4 Ablation Study

We conduct a detailed ablation study of SRSplat on the RealEstate10K datasets, downsampled by a factor of 4.

Importance of each module of SRSplat. In this section, we analyze the effectiveness of the proposed modules in detail. The *RGFE* module compensates for the lack of high-frequency information in LR input images by leveraging external reference images. The *TADC* adaptively densifies Gaussian primitives based on the texture richness of internal images. The experimental results are summarized in Tab. 5 (rows 1–3), demonstrating the pivotal roles of these modules in super-resolution reconstruction.

Importance of texture richness perceptron. The texture richness (TR) perceptron plays a crucial role in distilling texture richness from LR inputs, which is essential for subsequent texture-aware density control. When we remove TR perceptron and use the Sobel operator directly to obtain TR maps from LR images, we observe a decrease in reconstruction quality, as shown in Tab. 5 (row 4). Fig. 8 presents a comparison of TR maps extracted by different methods. The maps produced by the TR perceptron align more accurately with scene texture details (e.g., the curtain rod).

5 Conclusion

In this work, we propose **SRSplat**, the first feed-forward Super-Resolution framework. Our method initially constructs a reference gallery tailored to the unique characteristics of each scene. With this reference gallery, we introduce *RGFE* to align and fuse features from the LR input images and their reference. To further refine scene details, we propose *TADC*, which adaptively adjusts the density of scene's Gaussians according to the texture richness. Extensive experiments on multiple datasets demonstrate that SRSplat outperforms existing methods.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants (No. 62206082, 62422204, 62502135), the Key Research and Development Program of Zhejiang Province (No. 2025C01026), the Zhejiang Provincial Natural Science Foundation of China under Grants (No. LQN25F030014), the Scientific Research Innovation Capability Support Project for Young Faculty. This research was also supported by the National College Student Innovation and Entrepreneurship Training Program of China under Grants (No. 202410336018 and 202510336016).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Bösiger, L.; Dusmanu, M.; Pollefeys, M.; and Bauer, Z. 2024. MaRINeR: Enhancing Novel Views by Matching Rendered Images with Nearby References. In *European Conference on Computer Vision*, 76–94. Springer.

Cao, C.; Ren, X.; and Fu, Y. 2022. MVSFormer: Multi-view stereo by learning robust image features and temperature-based depth. *arXiv preprint arXiv:2208.02541*.

Cao, J.; Liang, J.; Zhang, K.; Li, Y.; Zhang, Y.; Wang, W.; and Gool, L. V. 2022. Reference-based image super-resolution with deformable attention transformer. In *European conference on computer vision*, 325–342. Springer.

Charatan, D.; Li, S. L.; Tagliasacchi, A.; and Sitzmann, V. 2024. 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*, 19457–19467.

Chen, A.; Xu, Z.; Geiger, A.; Yu, J.; and Su, H. 2022. Tensorf: Tensorial radiance fields. In *European conference on computer vision*, 333–350. Springer.

Chen, Y.; Xu, H.; Zheng, C.; Zhuang, B.; Pollefeys, M.; Geiger, A.; Cham, T.-J.; and Cai, J. 2024a. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, 370–386. Springer.

Chen, Y.; Zheng, C.; Xu, H.; Zhuang, B.; Vedaldi, A.; Cham, T.-J.; and Cai, J. 2024b. Mvsplat360: Feed-forward 360 scene synthesis from sparse views. *Advances in Neural Information Processing Systems*, 37: 107064–107086.

Fei, X.; Zheng, W.; Duan, Y.; Zhan, W.; Tomizuka, M.; Keutzer, K.; and Lu, J. 2024. Pixelgaussian: Generalizable 3d gaussian reconstruction from arbitrary views. *arXiv preprint arXiv:2410.18979*.

Feng, X.; He, Y.; Wang, Y.; Yang, Y.; Li, W.; Chen, Y.; Kuang, Z.; Fan, J.; Jun, Y.; et al. 2024. SRGS: Super-Resolution 3D Gaussian Splatting. *arXiv preprint arXiv:2404.10318*.

Hu, X.; Shi, C.; Yang, C.; Chen, M.; Gu, X.; Ding, J.; He, J.; and Fan, J. 2025. Texture-aware 3d Gaussian splatting for sparse view reconstructions. *Applied Soft Computing*, 113530.

Huang, C.; Mees, O.; Zeng, A.; and Burgard, W. 2023a. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 10608–10615. IEEE.

Huang, X.; Li, W.; Hu, J.; Chen, H.; and Wang, Y. 2023b. Refsr-nerf: Towards high fidelity and super resolution view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8244–8253.

Jiang, Y.; Chan, K. C.; Wang, X.; Loy, C. C.; and Liu, Z. 2021. Robust reference-based super-resolution via c2-matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2103–2112.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4).

Kingma, D. P. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Labs, B. F.; Batifol, S.; Blattmann, A.; Boesel, F.; Consul, S.; Diagne, C.; Dockhorn, T.; English, J.; English, Z.; Esser, P.; Kulal, S.; Lacey, K.; Levi, Y.; Li, C.; Lorenz, D.; Müller, J.; Podell, D.; Rombach, R.; Saini, H.; Sauer, A.; and Smith, L. 2025. FLUX.1 Kontext: Flow Matching for In-Context Image Generation and Editing in Latent Space. arXiv:2506.15742.

Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; and Timofte, R. 2021. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1833–1844.

Liu, A.; Tucker, R.; Jampani, V.; Makadia, A.; Snavely, N.; and Kanazawa, A. 2021a. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14458–14467.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021b. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Lu, L.; Li, W.; Tao, X.; Lu, J.; and Jia, J. 2021. Masa-sr: Matching acceleration and spatial adaptation for reference-based image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6368–6377.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Müller, T.; Evans, A.; Schied, C.; and Keller, A. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4): 1–15.

Nam, S.; Sun, X.; Kang, G.; Lee, Y.; Oh, S.; and Park, E. 2024. Generative Densification: Learning to Densify Gaussians for High-Fidelity Generalizable 3D Reconstruction. *arXiv preprint arXiv:2412.06234*.

Shi, C.; Chen, M.; Mao, Y.; Yang, C.; Hu, X.; Ding, J.; and Yu, Z. 2025a. REALM: An MLLM-Agent Framework for Open World 3D Reasoning Segmentation and Editing on Gaussian Splatting. *arXiv preprint arXiv:2510.16410*.

Shi, C.; Yang, C.; Hu, X.; Chen, M.; Pan, W.; Yang, Y.; Ding, J.; Yu, Z.; and Yu, J. 2025b. Sparse4DGS: 4D Gaussian Splatting for Sparse-Frame Dynamic Scene Reconstruction. *arXiv preprint arXiv:2511.07122*.

Shi, C.; Yang, C.; Hu, X.; Yang, Y.; Ding, J.; and Tan, M. 2025c. MMGS: Multi-Model Synergistic Gaussian Splatting for Sparse View Synthesis. *Image and Vision Computing*, 158: 105512.

Sun, H.; Li, W.; Liu, J.; Chen, H.; Pei, R.; Zou, X.; Yan, Y.; and Yang, Y. 2024. Coser: Bridging image and language for cognitive super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 25868–25878.

Tang, S.; Ye, W.; Ye, P.; Lin, W.; Zhou, Y.; Chen, T.; and Ouyang, W. 2024. Hisplat: Hierarchical 3d gaussian splatting for generalizable sparse-view reconstruction. *arXiv preprint arXiv:2410.06245*.

Tian, Q.; Tan, X.; Xie, Y.; and Ma, L. 2025. Drivingforward: Feed-forward 3d gaussian splatting for driving scene reconstruction from flexible surround-view input. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 7374–7382.

Wan, Y.; Cheng, Y.; Shao, M.; and Zuo, W. 2025. S2Gaussian: Sparse-View Super-Resolution 3D Gaussian Splatting. *CVPR*.

Wang, C.; Wu, X.; Guo, Y.-C.; Zhang, S.-H.; Tai, Y.-W.; and Hu, S.-M. 2021. NeRF-SR: High-Quality Neural Radiance Fields using Super-Sampling. *arXiv*.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612.

Xu, H.; Peng, S.; Wang, F.; Blum, H.; Barath, D.; Geiger, A.; and Pollefeys, M. 2025. Depthsplat: Connecting gaussian splatting and depth. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 16453–16463.

Xu, H.; Zhang, J.; Cai, J.; Rezatofighi, H.; and Tao, D. 2022. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8121–8130.

Xu, H.; Zhang, J.; Cai, J.; Rezatofighi, H.; Yu, F.; Tao, D.; and Geiger, A. 2023. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11): 13941–13958.

Yang, C.; Shi, C.; Hu, X.; Zhu, S.; Ding, J.; Wang, Y.; and Tan, M. 2025. SR4D: Dynamic Scene Super Resolution from Monocular Videos. *Knowledge-Based Systems*, 114869.

Yang, F.; Yang, H.; Fu, J.; Lu, H.; and Guo, B. 2020. Learning texture transformer network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5791–5800.

Yang, L.; Kang, B.; Huang, Z.; Xu, X.; Feng, J.; and Zhao, H. 2024. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10371–10381.

Yao, Y.; Luo, Z.; Li, S.; Fang, T.; and Quan, L. 2018. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, 767–783.

Zhang, C.; Zou, Y.; Li, Z.; Yi, M.; and Wang, H. 2025. Transplat: Generalizable 3d gaussian splatting from sparse multi-view images with transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 9869–9877.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

Zhou, T.; Tucker, R.; Flynn, J.; Fyffe, G.; and Snavely, N. 2018. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*.

Zhu, Z.; Fan, Z.; Jiang, Y.; and Wang, Z. 2024. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *European conference on computer vision*, 145–163. Springer.