HISPLAT: HIERARCHICAL 3D GAUSSIAN SPLATTING FOR GENERALIZABLE SPARSE-VIEW RECONSTRUCTION

Anonymous authors
Paper under double-blind review

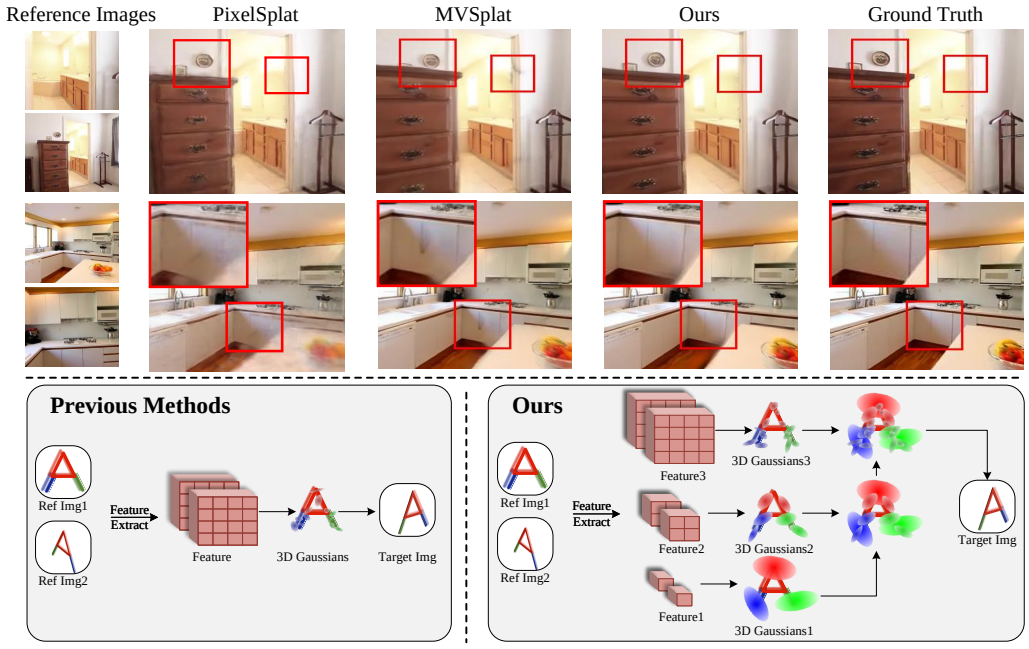

Figure 1: **Comparison between HiSplat and previous methods.** HiSplat constructs hierarchical 3D Gaussians which can better represent large-scale structures (more accurate location and less crack), and texture details (fewer artefacts and less blurriness).

ABSTRACT

Reconstructing 3D scenes from multiple viewpoints is a fundamental task in stereo vision. Recently, advances in generalizable 3D Gaussian Splatting have enabled high-quality novel view synthesis for unseen scenes from sparse input views by feed-forward predicting per-pixel Gaussian parameters without extra optimization. However, existing methods typically generate single-scale 3D Gaussians, which lack representation of both large-scale structure and texture details, resulting in mislocation and artefacts. In this paper, we propose a novel framework, HiSplat, which introduces a hierarchical manner in generalizable 3D Gaussian Splatting to construct hierarchical 3D Gaussians via a coarse-to-fine strategy. Specifically, HiSplat generates large coarse-grained Gaussians to capture large-scale structures, followed by fine-grained Gaussians to enhance delicate texture details. To promote inter-scale interactions, we propose an Error Aware Module for Gaussian compensation and a Modulating Fusion Module for Gaussian repair. Our method achieves joint optimization of hierarchical representations, allowing for novel view synthesis using only two-view reference images. Comprehensive experiments on various datasets demonstrate that HiSplat significantly enhances reconstruction quality and cross-dataset generalization compared to prior single-scale methods. The corresponding ablation study and analysis of different-scale 3D Gaussians reveal the mechanism behind the effectiveness. Codes will be released upon acceptance.

1 INTRODUCTION

As a crucial task in stereo vision, reconstructing 3D scenes from multiple viewpoints has been extensively explored. Recently, 3D Gaussian Splatting (Kerbl et al., 2023; Chen et al., 2024a) has emerged, which utilizes 3D Gaussians as explicit scene representation and adopts gradient descent to optimize the corresponding primitives. Different from previous methods based on ray-marching-based volume rendering, e.g., Neural Radiance Fields (NeRF) (Mildenhall et al., 2020), 3D Gaussian splatting (3D-GS) implements an efficient splatting rendering pipeline, enabling faster and higher-quality scene reconstruction without extra depth information. Nevertheless, the original 3D-GS requires a substantial number of multi-view images for per-scene optimization, which hinders its implementation under limited resources and sparse obversed images. To boost its generalization and transferability, data-driven generalizable 3D-GS (Szymanowicz et al., 2024; Charatan et al., 2024; Chen et al., 2024b; Zhang et al., 2024; Wewer et al., 2024) have been proposed. Generalizable 3D-GS leverages networks to feed-forward predict per-pixel Gaussian splatting parameters for unseen scenes, which can be described as mapping each 2D pixel to a fixed number of 3D Gaussians, namely splatter images. By generating splatter images, generalizable 3D-GS allows for high-quality novel view synthesis using only sparse views, especially the most challenging two views, of the new scene without additional optimization during inference.

However, current generalizable 3D-GS methods generate fixed-resolution splatter images using extracted single-scale features. The uniform 3D Gaussians result in a lack of hierarchical representation, making it challenging to simultaneously capture large-scale structures and delicate texture details. Consequently, it causes issues such as blurriness, cracks, mislocation, and artefacts as illustrated in Fig. 1. In 2D visual perception tasks such as segmentation and detection, hierarchical and multi-scale representations (He et al., 2014; Liu et al., 2015; Lin et al., 2016; Kong et al., 2018; Ghiasi et al., 2019) play an essential role in effectively capturing high-level semantic information for large-scale objects and structures while preserving low-level details. Inspired by the success of 2D multi-scale features, a natural question arises: as an explicit 3D representation akin to 2D features, **can 3D Gaussians benefit from a hierarchical structure to unleash the potential of representational capabilities?** Nonetheless, applying the hierarchical manner to generalizable 3D-GS is not straightforward. A simple preliminary experiment that extracts multi-scale features to generate hierarchical 3D Gaussians and mixes them for rendering has been conducted. As shown in the first line of Table 3, compared with the previous methods, constructing a vanilla hierarchical 3D Gaussians cannot obtain improvement. The core issue lies in that the multi-scale 3D Gaussians are generated independently and suffer from the inability to capture inter-scale information.

Different from predicting multi-scale Gaussians independently, we propose a novel hierarchical generalizable 3D-GS framework, namely **HiSplat**, for producing 3D Gaussians in a coarse-to-fine manner. Specifically, HiSplat generates large coarse-grained Gaussians to establish the large-scale structure as a skeleton, followed by fine-grained Gaussians around the coarse Gaussians to gradually enhance texture details as decoration. To prompt the interaction of different scales, we propose an attention-based **Error Aware Module** that allows finer-grained Gaussians to focus on compensating for errors generated by the coarse-grained Gaussians, referred to as Gaussian compensation. Additionally, to facilitate further correction of erroneous Gaussians, we propose a **Modulating Fusion Module** to reweight the opacities of larger-scale Gaussians based on the rendering quality of input reference images and the current joint features, termed Gaussian repair. By integrating Gaussian Compensation and Repair, the joint optimization of hierarchical 3D Gaussians can be achieved. During training, supervision is applied to the rendering images in every stage to introduce more gradient flow and provide richer depth information, enabling faster convergence and improved localization. During inference, only the final fusion Gaussians are utilized for rendering.

Extensive experiments demonstrate that, compared to previous methods with single-scale 3D Gaussian representations, HiSplat significantly enhances the quality of novel view synthesis, e.g., surpassing the leading open-source method **+0.82 PSNR** on RealEstate10K, while exhibiting stronger generalization capabilities on diverse unseen scenes, e.g., improving **+3.19 PSNR** for zero-shot testing on Replica. In summary, our contributions are as follows:

- We first study and introduce hierarchical 3D Gaussians representation in generalizable 3D-GS. It can simultaneously reconstruct higher-quality large-scale structures and more delicate texture details to significantly alleviate mislocation and artefacts.

- We construct a novel generalizable framework named HiSplat, generating hierarchical 3D Gaussians to reconstruct the scene with only two-view reference images. To exploit the inter-scale information and achieve joint optimization, we propose Error Aware Module for Gaussian compensation and Modulating Fusion Module for Gaussian repair.
- Comprehensive experiments on various mainstream datasets demonstrate that, compared with previous methods, HiSplat obtains superior reconstruction quality and cross-dataset generalization. Besides, ablation study and analysis on different scale Gaussians explain the effectiveness of HiSplat.

2 Related Work

2.1 Novel View Synthesis

Novel view synthesis aims to generate photorealistic images from unseen viewpoints given a set of input images. Neural Radiance Fields (NeRF) (Mildenhall et al., 2020; Yu et al., 2021; Pumarola et al., 2020; Barron et al., 2021; 2022) marked a significant breakthrough by modeling scenes as continuous volumetric radiance fields parameterized with neural networks. Despite achieving high-quality results, NeRF suffers from slow training speeds and high memory usage due to extensive MLP evaluations and the storage of numerous point samples. In contrast, 3D Gaussian Splatting (3D-GS) (Kerbl et al., 2023; Yu et al., 2024; Yang et al., 2023) offers an explicit scene representation using anisotropic 3D Gaussians, defined by their positions, covariances, colors, and opacities. A differentiable renderer projects these Gaussians onto the image plane, allowing efficient rendering and gradient computation. However, traditional NeRF and 3DGS methods require dense multi-view images for single-scene optimization and lack generalization, performing poorly in sparse-view reconstruction tasks. Our approach addresses these limitations by achieving high generalization with only two-view reference images.

2.2 Generalizable 3D Gaussian Splatting

Generalizable 3D Gaussian Splatting has become a key approach for efficient 3D scene representation and novel view synthesis. Splatter Image (Szymanowicz et al., 2024) predicts 3D Gaussian parameters from a single image using an image-to-image neural network, enabling ultra-fast single-view 3D reconstruction. PixelSplat (Charatan et al., 2024) extends this to sparse multi-view settings, using image pairs and an epipolar transformer to learn cross-view correspondences and predict depth distributions. MVSplat (Chen et al., 2024b) constructs cost volumes via plane sweeping to capture cross-view similarities, improving geometry reconstruction by predicting depth maps and unprojecting them to obtain Gaussian centers. TranSplat (Zhang et al., 2024) employs a transformer-based model for sparse-view reconstruction, using depth-aware deformable matching and monocular depth priors for refinement. Despite these advancements, existing methods often fail to capture fine-grained details due to single-scale features, leading to artefacts and reduced quality, especially in complex regions. To address these limitations, we propose a hierarchical 3D Gaussian splatting method for coarse-to-fine generalizable multi-view reconstruction.

2.3 Hierarchical Neural Representation

Hierarchical and multi-scale representations are fundamental in computer vision and graphics (Clark, 1976; Burt & Adelson, 1983; Adelson et al., 1984). In 2D visual perception, SPP-net (He et al., 2014) introduced multi-scale pooling layers for robust recognition across scales. SSD (Liu et al., 2015) utilized multi-resolution feature maps for efficient object detection, while FPN (Lin et al., 2016) constructed feature pyramids with top-down pathways and lateral connections, enabling high-level semantic features at all scales. Subsequent works (Kong et al., 2018; Ghiasi et al., 2019) continued to refine multi-scale feature representations. In 3D novel view synthesis, hierarchical approaches have enhanced efficiency and quality. NeRF (Mildenhall et al., 2020) used hierarchical sampling with coarse and fine networks. Subsequent works like KiloNeRF (Reiser et al., 2021), and PyNeRF (Turki et al., 2023) employed strategies such as spatial partitioning and multi-scale NeRF models to improve rendering speed and quality across varying scene complexities. Our work extends hierarchical principles to generalizable 3D-GS, drawing inspiration primarily from 2D vision

techniques. We are the first to apply hierarchical neural representation to explicit 3D Gaussian representations, unleashing the potential of 3D Gaussian representations in capturing rich scene details across scales.

3 METHOD

3.1 FRAMEWORK OVERVIEW

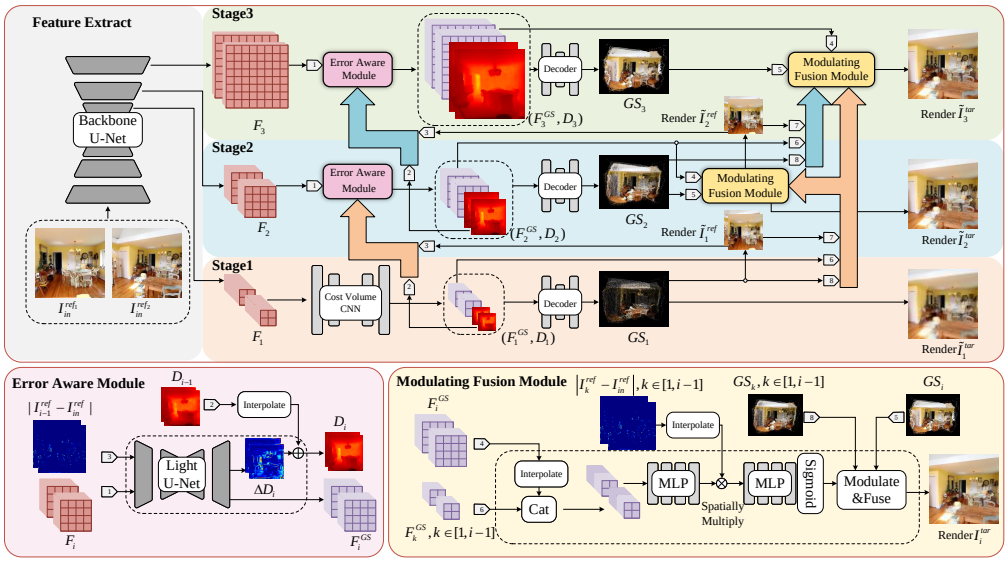

Figure 2: **The overall framework of HiSplat.** For simplicity, the situation with two input images is illustrated. HiSplat utilizes a shared U-Net backbone to extract different-scale features. With these features, three processing stages predict pixel-aligned Gaussian parameters with different scales, respectively. Error aware module and modulating fusion module perceive the errors in the early stages and guide the Gaussians in the later stages for compensation and repair. Finally, the fusing hierarchical Gaussians can reconstruct both the large-scale structure and texture details.

The overall framework of HiSplat is illustrated in Figure 2. Given a set of sparse sequential input images with their corresponding and the target camera pose information, HiSplat aims to feed-forward render the photorealistic target images from unseen views. To achieve this, HiSplat adopts a shared U-Net CNN with a multi-view transformer to extract the hierarchical cross-view features. The subsequent processing can be divided into three interrelated stages. For the lowest stage with the lowest resolution, the cross-view features are fed into a CNN with cost volume formulation to predict the depth and Gassian features, which can be decoded into corresponding Gaussian parameters. For the other two higher stages, they utilize the information (depth, features, and image errors) from the lower stages to predict their corresponding Gaussians via the Error Aware Module. Finally, the Modulating Fusion Module is used to adjust the Gaussian parameters and aggregate them to generate fusing multi-scale Gaussians. During training, fusing multi-scale Gaussians in each stage will be used to render target views parallelly with the supervision of ground truth. For inference, only the fusing Gaussians in the last stage are adopted.

3.2 HIERARCHICAL FEATURE EXTRACTION AND DEPTH ESTIMATION

Hierarchical Cross-view Feature Extraction. To extract hierarchical multi-scale features from input multi-view 2D images, we construct a CNN and Transformer mixed feature extraction backbone network with a U-Net (Ronneberger et al., 2015) architecture. Specifically, the network is divided into an encoder and a decoder, each composed of three 3×3 standard residual blocks (He et al., 2016). Each residual block performs upsampling/downsampling to extract features at different scales, and residual connections are used to integrate features of the same scale from the encoder and

decoder. At the end of the encoder, features are fed to a multi-view Transformer (Xu et al., 2023; 2022) that employs cross-attention and self-attention to extract mutual information from multi-view images and injects it into the subsequent decoding stream. For features extracted from different decoder stages, a simple convolutional head is used to obtain the corresponding scale features. To further enhance the generalization of feature extraction (see Table. 3), we follow MVSFormer++ (Cao et al., 2024) to introduce features of frozen DINOv2 (Oquab et al., 2023) and interpolate DINOv2 features to add with each scale features. Formally, Given N input images $I_{in}^{ref} \in \mathbb{R}^{N\times H\times W\times 3}$ with image size $H \times W$, the cross-view feature for stage i are denoted as

$$F_i = \mathcal{N}(I_{in}^{ref};\theta_{\mathcal{N}})_i + Interp(\mathcal{D}(I_{in}^{ref};\theta_{\mathcal{D}})), \tag{1}$$

where $F_i \in \mathbb{R}^{N\times \frac{H}{2^{3-i}}\times \frac{W}{2^{3-i}}\times \frac{C}{2^{i-1}}}$, C is the channel number of F_1; $\mathcal{N}$ and $\mathcal{D}$ are the U-Net backbone and DINOv2 respectively; θ is the corresponding parameters of network; $\mathcal{N}(\cdot;\cdot)_i$ denotes the output features from i_{th} stage of U-Net backbone decoder; $Interp(\cdot)$ is the interpolating operation.

Depth Estimation. With the camera matrix, depth is used to reproject the image pixel to 3D space, which is important for the correct location of Gaussians. For hierarchical Gaussians, the larger-scale Gaussians form the skeleton, while smaller Gaussians are near larger-scale Gaussians as a supplement. Therefore, the depth prediction principle is estimating the largest-scale Gaussian depth accurately, and using it as a reference to predict the depth of smaller-scale Gaussians. For depth in the first stage, we utilize the cost volume matching in Multi-View Stereo (MVS) (Cao et al., 2022; 2024; Yao et al., 2018) for accurate depth estimation. Since cost volume matching introduces unbearable computational load at high resolutions, we only apply it in the first stage. Specifically, given the feature $F \in \mathbb{R}^{N\times \frac{H}{4}\times \frac{W}{4}\times C}$ and the corresponding camera pose $P \in \mathbb{R}^{N\times 4\times 4}$, a plane-sweep stereo approach (Collins, 1996; Yao et al., 2018; Im et al., 2019) is utilized to sample R different depth candidates $V = \{d_1, d_2, ..., d_R\}$ from $[d_{near}, d_{far}]$. Then, the matching feature $F_{warp,k}^{ij} \in \mathbb{R}^{\frac{H}{4}\times \frac{W}{4}\times C}$ is sampled by warping the j_{th} view feature F^j to i_{th} view according to depth plane d_k, which is denoted as

$$F_{warp,k}^{ij} = Warp(F^j, P^i, P^j, d_k) \tag{2}$$

where $Warp$ is the warping and sampling operation (Yao et al., 2018; Xu et al., 2023; Chen et al., 2024b). The matching features from other views are utilized to generate the cost volume feature $F_{cv}^i = \{F_{cv,1}^i, F_{cv,2}^i, ..., F_{cv,R}^i\}$ by matching it with F^i, computed as

$$F_{cv,k}^i = \frac{1}{N-1}\sum_{j=1,j\neq i}^{N} \frac{F_{warp,k}^{ij}\cdot F^i}{\sqrt{C}}. \tag{3}$$

The cost volume features are fed into a lightweight CNN (Chen et al., 2024b) to obtain refined $\hat{F}_{cv,k}^i$ and the Gaussian feature F^{GS}. The depth of i_{th} view $D^i \in \mathbb{R}_+^{\frac{H}{4}\times \frac{W}{4}}$ is computed as

$$D^i = softmax(\hat{F}_{cv}^i)V. \tag{4}$$

For Gaussians in subsequent stages, we employ an Error Aware Module to predict their relative depth offsets, enabling the placement of Gaussians in appropriate positions where the large-scale Gaussians lack details or are incorrect.

3.3 Error Aware Module

To enable small-scale Gaussians to supplement the lacking details and correct structural errors of the large-scale Gaussians, we render the mixed Gaussians from the previous stage from input views and compute an error map with the input images. A lightweight 2D U-Net (Ronneberger et al., 2015; Zhou et al., 2018b) with two predictors is used to aggregate and generate the depth offsets and Gaussian features. To ensure that the decorative Gaussians are always near the skeletal Gaussians, we have restricted the range of depth offsets. Formally, given the images $\tilde{I}_{i-1}^{ref} \in \mathbb{R}^{N\times H\times W\times 3}$ rendered from Gaussians of stage $i-1$, the depth offsets degree $\alpha_i \in [0,1]^{N\times \frac{H}{2^{3-i}}\times \frac{W}{2^{3-i}}}$ and Gaussian features $F_i^{GS} \in \mathbb{R}^{N\times \frac{H}{2^{3-i}}\times \frac{W}{2^{3-i}}\times C_{GS}}$ can be computed as

$$(\alpha_i, F_i^{GS}) = \mathcal{U}(|\tilde{I}_{i-1}^{ref} - I_{in}^{ref}|, F_i; \theta_{\mathcal{U}}), \tag{5}$$

where $\mathcal{U}$ and $\theta_{\mathcal{U}}$ are the lightweight U-Net and its corresponding weights. Given a maximum depth coefficient $\eta \in [0, 1]$ (typical value 0.1), the depth D_i in stage i can be computed as

$$D_i = \Delta D_i + Interp(D_{i-1}), \Delta D_i = (2\alpha_i - 1) \cdot \eta Interp(D_{i-1}), \tag{6}$$

where $Interp$ is the interpolating operation aiming to upsameple D_{i-1} to the same spacial size as D_i. By introducing η, the ΔD_i can be restricted in $[-\eta Interp(D_{i-1}), \eta Interp(D_{i-1})]$.

3.4 Gaussian Parameter Prediction

After obtaining the Gaussian features F_i^{GS} and corresponding depth D_i in stage i, we follow PixelSplat (Charatan et al., 2024) to predict the pixel-aligned Gaussian parameters, including the Gaussian center, opacity, covariance, and spherical harmonics coefficients. Specifically, for the Gaussian center, we utilize the D_i along the pixel-aligned ray to unproject the 2D pixel to the 3D space location as the Gaussian center. For other Gaussian parameters, stacked convolutional layers with the corresponding predictor are adopted to generate them.

3.5 Modulating Fusion Module

Simply fusing Gaussians of different scales struggles with achieving optimal joint optimization, and merely introducing smaller-scale Gaussians cannot fully correct the potential errors generated by large-scale Gaussians. Therefore, inspired by spatial attention (Jaderberg et al., 2015; Almahairi et al., 2016), we propose the Modulating Fusion Module. It focuses on modulating and reweighting the Gaussians' opacity in the areas with significant errors. Formally, given the Gaussian features from current and previous stages F_i^{GS} and $\{F_k^{GS} | k \in [1, i-1]\}$, we obtain the concatenate Gaussians' features $F^{cat} = \{F_k^{cat} | k \in [1, i-1]\}$, denoted as

$$F_k^{cat} = cat(interp(F_i^{GS}), F_k^{GS}), \tag{7}$$

where $interp(\cdot)$ is interploating F_i^{GS} to the same corresponding spatial size as each F_k^{GS}; $cat(\cdot, \cdot)$ is concatenating in the channel dimension. An MLP_1 is used to squeeze the dimension of F^{cat} to multiply the corresponding error maps spatially, following another MLP_2 with sigmoid outputs the modulating coefficient $\xi_k \in [0, 1]^{N \times \frac{H}{2^{3-k}} \times \frac{W}{2^{3-k}}}$ of previous Gaussians' opacity $O_k \in [0, 1]^{N \times \frac{H}{2^{3-k}} \times \frac{W}{2^{3-k}}}$, denoted as

$$\xi_k = Sigmoid(MLP_2(MLP_1(F_k^{cat}; \theta_{MLP_1}) \cdot interp(|\tilde{I}_k^{ref} - I_{in}^{ref}|); \theta_{MLP_2})), \tag{8}$$

where $Sigmoid(\cdot)$ is the non-linear sigmoid operation; θ_{MLP} is the corresponding weights. The previous Gaussians' opacity $\{O_1, O_2, ..., O_{i-1}\}$ is updated following $O_k := O_k \cdot \xi_k$.

3.6 Training objective

During training, we render the fusing hierarchical Gaussians in all stages to obtain images from target novel views and utilize the photometric losses (Charatan et al., 2024; Chen et al., 2024b), including Mean Squared Error (MSE) loss and Learned Perceptual Image Patch Similarity (LPIPS) losses (Zhang et al., 2018), to supervise each image. The all training objective is

$$\mathcal{L}_{all} = \sum_{i=1}^{3} \lambda_{mse} \mathcal{L}_{mse}(\tilde{I}_i^{tar}, I^{tar}) + \lambda_{lpips} \mathcal{L}_{lpips}(\tilde{I}_i^{tar}, I^{tar}), \tag{9}$$

where the loss coefficient $\lambda_{mse} = 1$, $\lambda_{lpips} = 0.05$. Except for the frozen feature extractor of DINOv2, other parameters are trainable and supervised by the training objective.

4 Experiment

4.1 Experimental Setting

Datasets. To comprehensively evaluate the reconstruction ability, we train and test models in two large-scale datasets, RealEstate10K (Zhou et al., 2018a) and ACID (Liu et al., 2021). The

RealEstate10K dataset comprises videos sourced from YouTube, divided into 67,477 training scenes and 7,289 testing scenes. The ACID dataset consists of nature scenes captured via aerial drones, with 11,075 scenes for training and 1,972 scenes for testing. Both datasets are calibrated with Structure-from-Motion (SfM) (Schonberger & Frahm, 2016) algorithm to estimate camera intrinsic and extrinsic parameters for each frame. Following the novel view synthesis settings of previous works (Charatan et al., 2024; Chen et al., 2024b; Zhang et al., 2024), two context images are as input, and three novel target views are rendered for each test scene. Besides, to compare the cross-dataset generalization ability, we select other two multi-view datasets, including real object-centric dataset DTU (Jensen et al., 2014) and synthetic indoor dataset Replica (Straub et al., 2019), for zero-shot test (without fine-tuning or training). Following (Chen et al., 2024b) and (Zhi et al., 2021), we select sixteen scenes in DTU and eight scenes in Replica for testing. To quantitatively measure the rendering quality of novel views from different aspects, Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM) (Wang et al., 2004) and Learned Perceptual Image Patch Similarity (LPIPS) losses (Zhang et al., 2018) are adopted as testing metrics.

Implementation Details. For a fair comparison, we follow the commonly used training settings (Chen et al., 2024b; Charatan et al., 2024; Zhang et al., 2024). Specifically, the input images are resized as 256×256, and the model is optimized by Adam (Kingma, 2014) for 300,000 iterations. The training experiments are implemented in 8 RTX4090 with batch size 2 for two days. There are more implementation details in A.1.

4.2 Main Results

4.2.1 Novel views synthesis

To verify the effectiveness of the proposed HiSplat on novel view synthesis, we train and test the model on large-scale multi-view datasets RealEstate10K and ACID, respectively. We select four typical feed-forward NeRF-based methods, including pixelNeRF (Yu et al., 2021), GPNR (Suhail et al., 2022), AttnRend (Du et al., 2023), MuRF (Xu et al., 2024), and three recent Gaussian-Splatting-based methods, including PixelSplat (Charatan et al., 2024), MVSplat (Chen et al., 2024b), TranSplat (Zhang et al., 2024) as comparisons. As shown in Table 1, compared with previous methods, HiSplat can consistently achieve state-of-the-art (SOTA) performance on different datasets and metrics with significant improvement, which demonstrates the superiority of the proposed hierarchical structure of Gaussian primitives. Specifically, on RealEstate10K dataset, HiSplat surpasses the latest state-of-the-art (SOTA) open-source method MVSplat by +0.82 PSNR, and exceeds the leading method Transplat by +0.52 PSNR, achieving a new milestone by obtaining higher than 27 PSNR in the challenging two-view reconstruction task. On ACID dataset, HiSplat outperforms other methods by +0.5 PSNR than MVSpalt and +0.4 PSNR than TranSplat. For the patch-level SSIM and feature-level LPIPS metrics, HiSplat also gains significant improvement, suggesting that HiSplat can reconstruct details and large-scale structures with higher quality. Besides the performance, we also report the inference time and peak GPU memory in A.2.

Table 1: **Evaluation on RealEstate10K and ACID.** Compared with the previous NeRF-based and generalizable Gaussian-Splatting-based method, HiSplat can consistently obtain higher rendering quality of unseen views.

Method	RealEstate10K			ACID		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
pixelNeRF (Yu et al., 2021)	20.43	0.589	0.550	20.97	0.547	0.533
GPNR (Suhail et al., 2022)	24.11	0.793	0.255	25.28	0.764	0.332
AttnRend (Du et al., 2023)	24.78	0.820	0.213	26.88	0.799	0.218
MuRF (Xu et al., 2024)	26.10	0.858	0.143	28.09	0.841	0.155
PixelSplat (Charatan et al., 2024)	25.89	0.858	0.142	28.14	0.839	0.150
MVSplat (Chen et al., 2024b)	26.39	0.869	0.128	28.25	0.843	0.144
TranSplat (Zhang et al., 2024)	26.69	0.875	0.125	28.35	0.845	0.143
HiSplat(Ours)	27.21	0.881	0.117	28.75	0.853	0.133

4.2.2 Cross-dataset Generalization

To verify the generalization ability of HiSplat, we train the model on RealEstate10K and directly test it on DTU, ACID, and Replica in a zero-shot setting. As depicted in Figure 3, the images generated by HiSplat are more aligned with human perception, featuring less edge blurriness, less artefacts

and more accurate location. The quantitive results are shown in Table 2. Compared with previous single-scale Gaussian-Splatting-based methods, HiSplat performs a significant generalization improvement, e.g., obtaining +1.05 PNSR on object-centric dataset DTU, +0.55 PSNR on outdoor dataset ACID and +3.19 PSNR on indoor dataset Replica over the suboptimal methods, suggesting that HiSplat is more effectively deployed in the practical open-world scenario with various data distribution. It is worth noting that on ACID, HiSplat's zero-shot performance even outperforms others trained specially on ACID significantly. The outstanding generalization ability can be explained from two aspects: 1) the hierarchical Gaussian representation can reconstruct scenes of vastly different scales simultaneously 2) the error-aware mechanism is scale-invariant, which can benefit consistently across a wide range of scenes.

Table 2: **Evaluation on cross-dataset generalization.** We train models on RealEstate10K, and test them on object-centric dataset DTU, outdoor dataset ACID, and indoor dataset Replica in a zero-shot setting. Compared with previous methods, HiSplat can better handle various scenes with different distributions and scales.

Method	RealEstate10K → DTU			RealEstate10K→ACID			RealEstate10K→Replica		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
PixelSplat (Charatan et al., 2024)	12.89	0.382	0.560	27.64	0.830	0.160	23.98	0.821	0.202
MVSplat (Chen et al., 2024b)	13.94	0.473	0.385	28.15	0.841	0.147	23.79	0.817	0.165
TranSplat (Zhang et al., 2024)	14.93	0.531	0.326	28.17	0.842	0.146	-	-	-
HiSplat(Ours)	16.05	0.671	0.277	28.66	0.850	0.137	27.17	0.899	0.113

4.3 Ablation Experiment

To verify the effectiveness of each proposed technique of HiSplat, we conduct ablation experiments to eliminate each component step by step, and compare them with previous methods on RealEstate10K. As illustrated in Table 3, different degrees of performance degradation are observed of any removal, validating the necessity of each component. It merits attention that the vanilla hierarchical 3D Gaussian representation cannot perform competitively compared with previous methods (-0.21 PSNR compared with MVSpalt). Whereas, when combined with the proposed EAM and MFM, the potential of hierarchical manner is unleashed with prominent improvement (+0.82 PSNR). The reason is that the EAM and MFM promote the transfer of error-aware information and features across different scales, driving Gaussians in the later stages to compensate for lacking details and repair the error of earlier stages for joint optimization, as stated in Sec 1.

Table 3: **The ablation study on RealEstate10K.** Hier: Hierarchical Gaussian, EAM: Error Aware Module, MFM: Modulating Fusion Module, DINO: using DINOv2 feature. Only using vanilla hierarchical Gaussians cannot perform better than previous methods. Each proposed component contributes to the final superior results.

	Hier	EAM	MFM	DINO	PSNR↑	SSIM↑	LPIPS↓
Variants of HiSplat(Ours)	✓	×	×	×	26.18	0.869	0.135
	✓	✓	×	×	26.76	0.874	0.123
	✓	✓	✓	×	27.02	0.879	0.120
	✓	✓	✓	✓	27.21	0.881	0.117
Other methods	PixelSplat (Charatan et al., 2024)				25.89	0.858	0.142
	MVSplat (Chen et al., 2024b)				26.39	0.869	0.128

4.4 Analysis of Hierarchical Gaussians

In this section, we visually analyze the different stages during the inference process to reveal the mechanism behind the effectiveness of HiSplat.

Analysis of 3D Gassian Primitives. To visually display the fusing Gaussians in different stages, we take the center of Gaussian primitives as the 3D position of point clouds and utilize the color of corresponding context images to draw the Gaussian primitives. As shown in Figure 4, for the later stage, as the number of Gaussian primitives increases, the rendering quality also gradually improves, demonstrating the effectiveness of adding Gaussian primitives. To further analyze the function of adding Gaussians of each stage, we report the statistical probability of opacity and mean scale of Gaussians. It is noteworthy that because the fusing Gaussians in the later stage contain the Gaussians from the early stages, for a clear analysis, we individually report the statistical value of fusing Gaussians from each stage. It can be observed that the Gaussian primitives in stage 1 are sparse, solid,

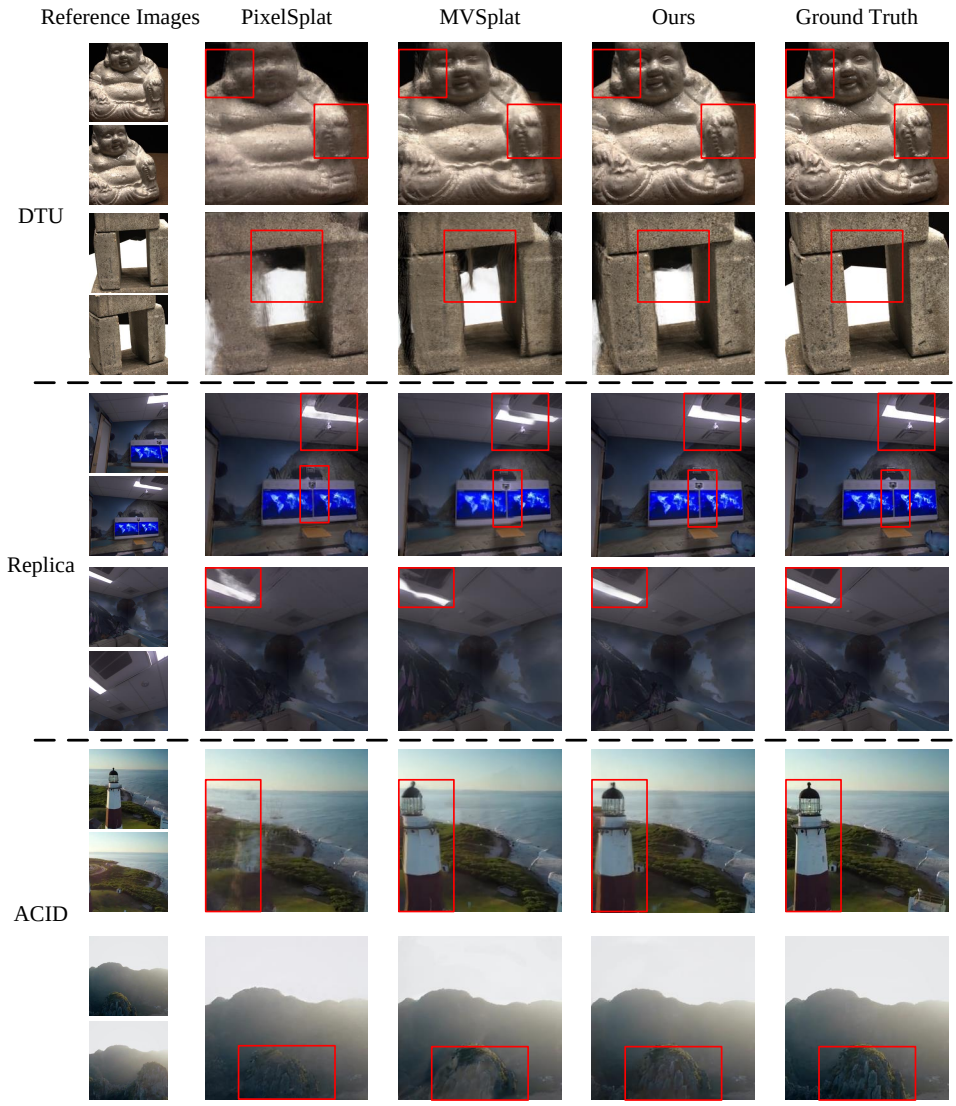

Figure 3: **Qualitative comparison of generalization ability.** For the scenes out of training distribution, HiSplat can generate higher-quality novel-view images. More comparison is provided in A.3.

and large while the adding Gaussian primitives from later stages gradually become dense, transparent and small. This pattern of Gaussian attributes aligns with our hypothesis: **larger and more solid Gaussian primitives form the skeleton of the scene, aiming to establish the large-scale basic structure, while smaller and more transparent Gaussian primitives serve as decoration, further refining the texture details, which can be described as "from bone to flesh".** Following this hypothesis, we explain the function of the displayed hierarchical Gaussians that Gaussian primitives from stage 1 and stage 2 are relatively solid, adhering closely to the surface of the scene, which gradually completes the outline and basic structure; as for the Gaussian primitives from stage 3, they are very tiny and transparent, introducing richer texture details and illumination. Besides, compared with other methods, HiSplat can render a high-quality novel-view image with more accurate and consistent geometry, especially the toy bird's crest in Figure 4.

Analysis of 2D Error Map. To analyze how hierarchical Gaussians reconstruct the scene step by step in detail, we show the colored error map of different stages, generated by comparing the rendering images with the ground truth in Figure 5. The brighter and redder pixels exhibit more significant errors and a rectangle highlights the regions with complex and rich textures. It can be observed that as the stage index increases, the overall error continuously decreases, especially in areas with complex textures. Besides, by directly examining the rendered images, we can observe

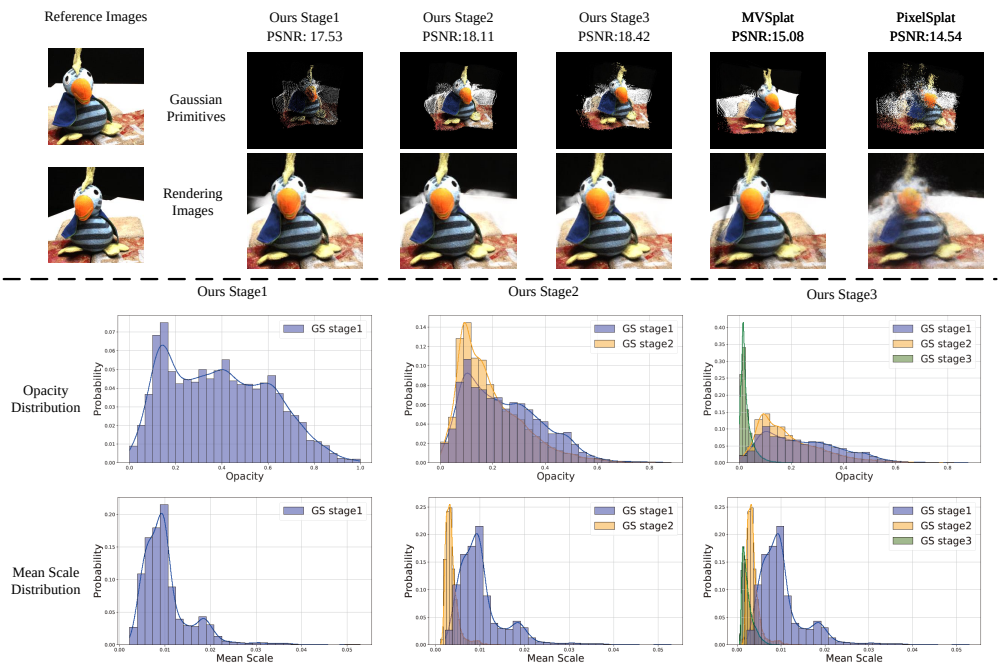

Figure 4: **Comparison of Gaussian primitives in different stages on DTU.** HisPlat can gradually generate large-scale solid Gaussians as "bone" and small-scale transparent Gaussians as "flesh", confirming better rendering quality and geometry.

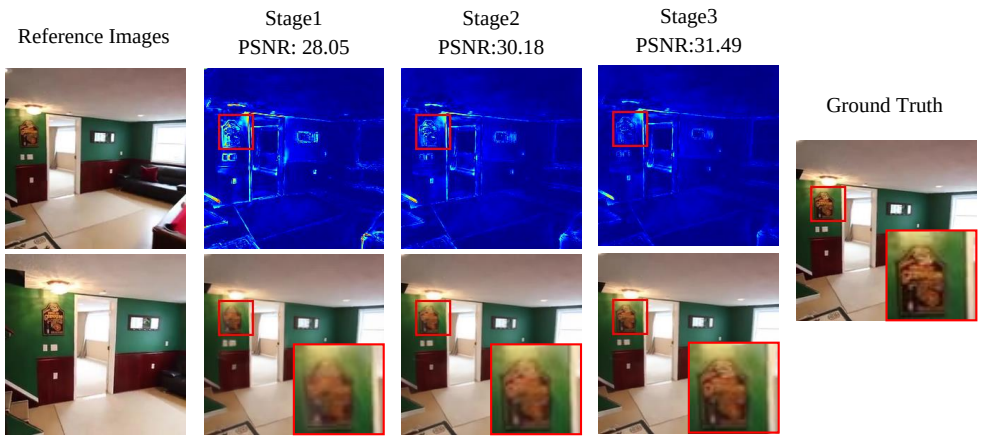

Figure 5: **Comparison of rendering images from different stages on RealEstate10K.** HiSplat can perceive the error, and utilize Gaussians in the later stages to add details and correct errors gradually.

that there is less blurriness and more details. It suggests that HiSplat can perceive the error in the early stage, and utilize Gaussians in the later stage for repair and compensation.

5 Conclusion

In this paper, we propose a novel generalizable 3D Gaussian Splatting framework, HiSplat, aiming to render novel-view images from sparse (only two) context images. Different from previous methods predicting single-scale Gaussians, HiSplat gradually generates multi-scale hierarchical Gaussians to reconstruct the large-scale structure and texture details with higher quality. To facilitate the information interaction of different stages, we propose Error Aware Module and Modulating Fusion Module for Gaussian compensation and repair. Extensive experiments across multiple datasets demonstrate that HiSplat significantly enhances the quality of reconstructions and cross-dataset generalization, surpassing previous single-scale methods.

REFERENCES

Edward H. Adelson, Peter J. Burt, Charles H. Anderson, Joan M. Ogden, and James R. Bergen. Pyramid methods in image processing. 1984. URL https://api.semanticscholar.org/CorpusID:1330580.

Amjad Almahairi, Nicolas Ballas, Tim Cooijmans, Yin Zheng, Hugo Larochelle, and Aaron Courville. Dynamic capacity networks. In *International conference on machine learning*, pp. 2549–2558. PMLR, 2016.

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. *ICCV*, 2021.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. *CVPR*, 2022.

Peter J. Burt and Edward H. Adelson. The laplacian pyramid as a compact image code. *IEEE Trans. Commun.*, 31:532–540, 1983. URL https://api.semanticscholar.org/CorpusID:8018433.

Chenjie Cao, Xinlin Ren, and Yanwei Fu. Mvsformer: Multi-view stereo by learning robust image features and temperature-based depth. *arXiv preprint arXiv:2208.02541*, 2022.

Chenjie Cao, Xinlin Ren, and Yanwei Fu. Mvsformer++: Revealing the devil in transformer's details for multi-view stereo. *arXiv preprint arXiv:2401.11673*, 2024.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. 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.

Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *arXiv preprint arXiv:2406.06521*, 2024a.

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. *arXiv preprint arXiv:2403.14627*, 2024b.

James H. Clark. Hierarchical geometric models for visible surface algorithms. *Seminal graphics: pioneering efforts that shaped the field*, 1976. URL https://api.semanticscholar.org/CorpusID:8231831.

Robert T Collins. 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.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

Yilun Du, Cameron Smith, Ayush Tewari, and Vincent Sitzmann. Learning to render novel views from wide-baseline stereo pairs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4970–4980, 2023.

Golnaz Ghiasi, Tsung-Yi Lin, Ruoming Pang, and Quoc V. Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7029–7038, 2019. URL https://api.semanticscholar.org/CorpusID:118634258.

Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2495–2504, 2020.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37:1904–1916, 2014. URL https://api.semanticscholar.org/CorpusID:436933.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. Dpsnet: End-to-end deep plane sweep stereo. *arXiv preprint arXiv:1905.00538*, 2019.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.

Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 406–413, 2014.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

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

Tao Kong, Fuchun Sun, Chuanqi Tan, Huaping Liu, and Wenbing Huang. Deep feature pyramid reconfiguration for object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, 2016. URL https://api.semanticscholar.org/CorpusID:10716717.

Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. 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.

W. Liu, Dragomir Anguelov, D. Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, 2015. URL https://api.semanticscholar.org/CorpusID:2141740.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *International Conference on Computer Vision (ICCV)*, 2021.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113, 2016.

Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs. 2024. URL https://arxiv.org/abs/2408.13912.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *European Conference on Computer Vision*, pp. 156–174. Springer, 2022.

Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Haithem Turki, Michael Zollhöfer, Christian Richardt, and Deva Ramanan. Pynerf: Pyramidal neural radiance fields. In *Thirty-Seventh Conference on Neural Information Processing Systems*, 2023.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024a.

Yunsong Wang, Tianxin Huang, Hanlin Chen, and Gim Hee Lee. Freesplat: Generalizable 3d gaussian splatting towards free-view synthesis of indoor scenes. 2024b. URL https://arxiv.org/abs/2405.17958.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Christopher Wewer, Kevin Raj, Eddy Ilg, Bernt Schiele, and Jan Eric Lenssen. latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. *arXiv preprint arXiv:2403.16292*, 2024.

Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8121–8130, 2022.

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*, 2023.

Haofei Xu, Anpei Chen, Yuedong Chen, Christos Sakaridis, Yulun Zhang, Marc Pollefeys, Andreas Geiger, and Fisher Yu. Murf: Multi-baseline radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20041–20050, 2024.

Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023.

Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 767–783, 2018.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4578–4587, 2021.

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

Chuanrui Zhang, Yingshuang Zou, Zhuoling Li, Minmin Yi, and Haoqian Wang. Transplat: Generalizable 3d gaussian splatting from sparse multi-view images with transformers. *arXiv preprint arXiv:2408.13770*, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 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.

Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15838–15847, 2021.

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

Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*, pp. 3–11. Springer, 2018b.

A APPENDIX

A.1 DETAILED ARCHITECTURE AND EXPERIMENT SETTING

In this section, we provide more details of the architecture and experiments.

More Architecture details. For the U-Net backbone, the encoder and decoder are constructed with a similar architecture consisting of stacked residual blocks. The kernel size of each block is 3×3, with increasing channels $[32, 64, 128]$ and half resolution for each stage. For combining the U-Net features with DINOv2 features, we follow the Side View Attention of MVSformer++ (Cao et al., 2024) to further improve the information aggregation from different views. There is a convolution layer to align the channels of DINOv2 features with the corresponding U-Net features for addition. Then, another convolutional layer is used to generate the features of each stage.

More Experiment details. For the training schedule on ACID and RealEstate10K, we follow (Chen et al., 2024b) to set a learning rate of 0.0002 with warm-up cosine decay. The warm-up steps are 2000. Random horizontal flipping is used as data augmentation for each image set. All experiments are based on the deep learning framework PyTorch and utilize Pytorch-Lightning for better organization. For results on RealEstate10K and ACID, because our basic experiment setting is as same as (Chen et al., 2024b; Charatan et al., 2024; Zhang et al., 2024), we directly cite the results from the original literature. For the zero-shot results on Replica, we only conduct experiments to compare with open-source methods MVSplat and PixelSplat, using their latest official checkpoints.

A.2 EFFICIENCY AND EFFECTIVENESS OF DIFFERENT STAGES

To report the effectiveness and efficiency of the proposed HiSplat, we test the inference cost and performance of HiSplat and previous methods. All experiments are conducted on an RTX4090 with batch size 8. The input image size is 256×256, and the reported inference time and peak memory are the average value of 10,000 recorded data points with the same inference settings. Because the early-stage Gaussians are predicted without relying on the later-stage Gaussians, HiSplat can terminate the inference in the early stages and reduce the inference cost, we add two variants of HiSplat, HiSplat-Stage 1 and HiSplat-Stage 2, which only predict the Gaussians in the stage 1 and 2, respectively. Thanks to the hierarchical character of HiSplat, it is very simple to change the mode of HiSplat for a flexible implementation. The results are shown in Table 4. It can be observed that compared with the previous methods, the complete version of HiSplat (HiSplat-Stage 3) can obtain the best performance (+0.82 PSNR compared with MVSplat) with relatively high but bearable inference cost, due to the three-stage processing and the introduction of DINOv2 feature. Flash Attention (Dao, 2023; Dao et al., 2022) and stage-wise parallel techniques are the optimal methods for future cost reduction. Besides, other modes of HiSplat can properly sacrifice the performance for lightweight implementation. Thus, compared with the previous fixed methods, HiSplat has the potential to form a dynamic framework for various practical requirements of resource or quality, suggesting HiSplat can achieve a better balance between performance and inference cost.

Table 4: **The efficiency and effectiveness of HiSplat and other methods.** HiSplat-Stage 1,2 represents only forwarding the 1,2 stages of HiSplat to generate part of hierarchical Gaussians for rendering, which can sacrifice performance for higher efficiency. Under different resource constraints, HiSplat can switch inference mode flexibly, achieving a good balance between performance and inference cost.

Method	Peak Memory/GB↓	Inference Time/s↓	PSNR↑	SSIM↑	lPIPS↓
PixelSplat (Charatan et al., 2024)	24.17	1.47	25.89	0.858	0.142
MVSplat (Chen et al., 2024b)	14.08	0.27	26.39	0.869	0.128
HiSplat-Stage 1	12.91	0.24	26.13	0.858	0.141
HiSplat-Stage 2	14.74	0.36	26.99	0.879	0.120
HiSplat-Stage 3	23.34	0.51	27.21	0.881	0.117

A.3 MORE QUALITATIVE COMPARISON

For a more detailed visual comparison, there are extended qualitative comparisons including novel-view-synthesis results on RealEstate10K in Figure 6; zero-shot results on DTU in Figure 7, ACID in Figure 8, Replica in Figure 9.

Figure 6: **Extended qualitative comparison on RealEstate10K.** Compared with the previous methods, HiSplat can handle large-scale structures (the location or shape of objects) and texture details better.

A.4 POSE-FREE SPARSE-VIEW RECONSTRUCTION

By integrating DUSt3R (Wang et al., 2024a), HiSplat can easily perform sparse-view reconstruction without relying on pre-calibrated camera parameters. To evaluate the HiSplat's ability to handle pose-free scenarios, we conduct a simple experimental setup to test it on the DTU and Replica datasets, using sparse views without camera parameters. Specifically, we use pre-trained DUSt3R to extract camera parameters, including intrinsics and extrinsics, from input images and then directly apply HiSplat (pre-trained on RealEstate10K) for novel view synthesis. The results are shown in Table 5. Compared with using relatively accurate poses from the dataset, using poses generated by DUSt3R leads to reduced performance but still outperforms MVSplat. This is because the simple experimental setup lacks sufficient coupling between DUSt3R's pose extraction and HiSplat's reconstruction process. The resulting less accurate poses affect HiSplat's ability to precisely determine the position and shape of Gaussian primitives, thus leading to reduced performance.

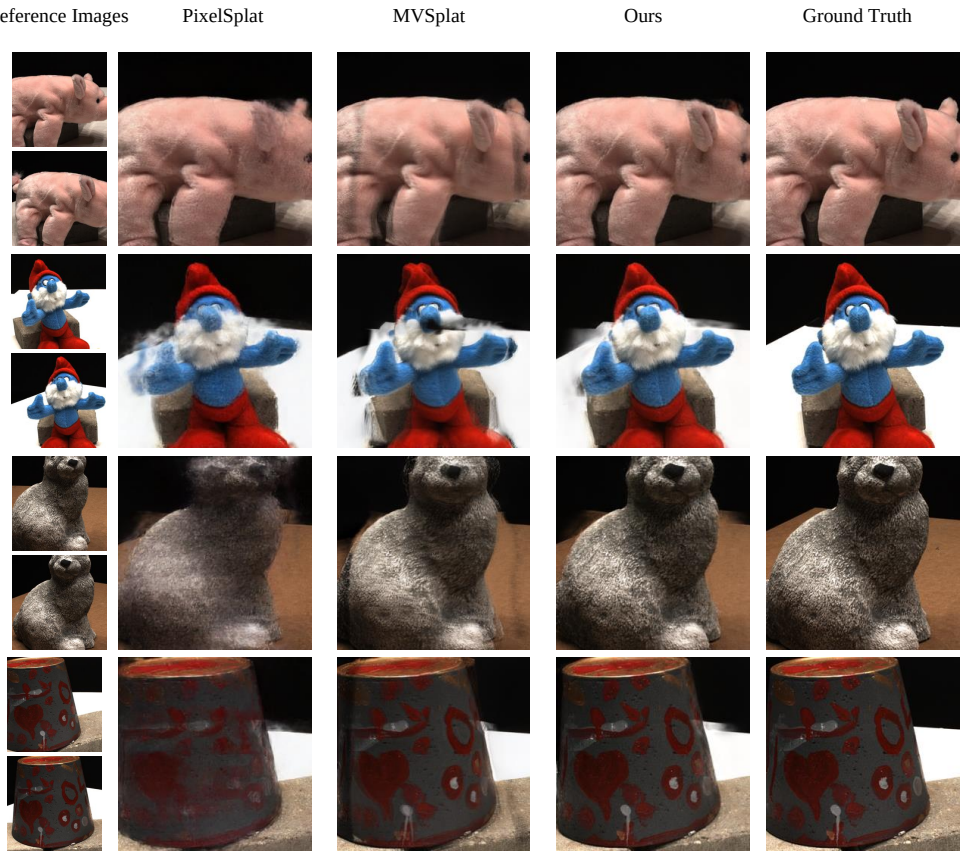

Figure 7: **Extended qualitative comparison on DTU.**For unseen objects, HiSplat can reconstruct the texture details more clearly with fewer artefacts and less blurriness.

Additionally, we test in real-life scenarios using three casually taken photos with a smartphone: two as reference images and one as the ground truth. Using the same simplified pipeline described above, the results are shown in Fig. 10. It demonstrates that HiSplat+DUSt3R's simplified pipeline has the ability to handle sparse-view reconstructions in real-life settings where only the images are inputted, producing a slightly satisfactory novel view synthesis. We believe designing a more robust and systematic pose-free reconstruction method based on HiSplat is a promising future direction.

Table 5: **Cross-dataset evaluation with pose-free settings.** HiSplat+GT pose: using the camera parameters from datasets; HiSplat+DUSt3R pose: using the camera parameters generated by pre-trained DUSt3R, which is used for pose-free settings. By simply combining with DUSt3R, HiSplat can handle pose-free sparse-view reconstruction.

Method	RealEstate10K→DTU	RealEstate10K→Replica
MVSplat+GT pose	13.94	23.79
MVSplat+DUSt3R pose	13.18	21.04
HiSplat+GT pose	16.05	27.21
HiSplat+DUSt3R pose	14.58	22.20

A.5 Implementation for more reference views

Although HiSplat focuses on two-view reconstruction, the main components of HiSplat, such as the hierarchical Gaussian representation, shared feature extractors, Error Aware Module, and Modulat-

Figure 8: **Extended qualitative comparison on ACID.** For unseen outdoor coastal scenes, HiSplat can generate more accurate illumination and location with fewer artefacts.

ing Fusion Module, are view-invariant, which means that HiSplat can be easily extended to handle reconstruction from an arbitrary number of views. To perform a preliminary verification, we directly use the weights trained by inputting two views on RealEstate10K to conduct cross-dataset testing with three views on DTU and Replica. The results are shown in Table 6. Compared with the two-view case, the performance slightly improves with inputting three views, and HiSplat still outperforms MVSplat, demonstrating the feasibility of using HiSplat for multi-view reconstruction directly. However, since HiSplat is not specifically trained for three-view tasks and lacks specific structural adjustments for more views in Table 6, the gain is marginal.

Table 6: **Cross-dataset evaluation with more reference views.** We train the models on RealEstate10K with two reference views and test them on DTU and Replica with three reference views.

Method	RealEstate10K→DTU	RealEstate10K→Replica
MVSplat+2 views	13.94	23.79
MVSplat+3 views	14.29	24.75
HiSplat+2 views	16.05	27.17
HiSplat+3 views	16.35	27.47

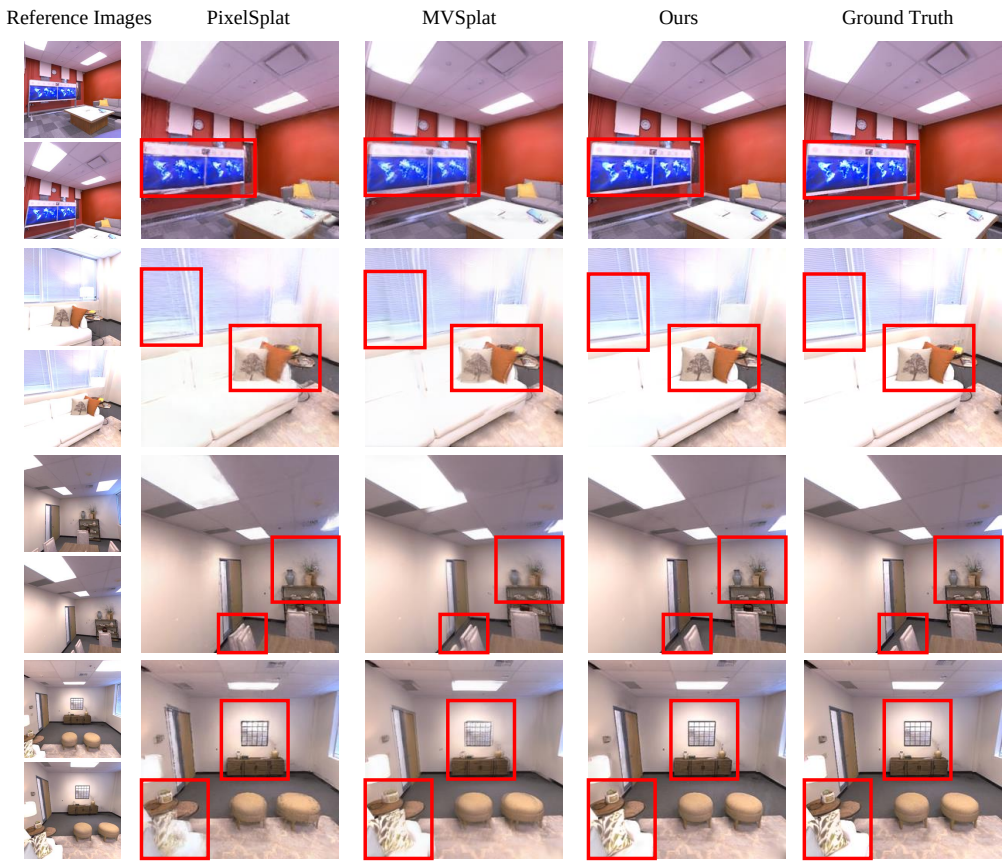

Figure 9: **Extended qualitative comparison on Replica.** For unseen indoor scenes, HiSplat can reconstruct the details better with less blurriness and fewer artefacts.

A.6 Comparison of different Gaussian Primitive number

In this section, we discuss the impact of the number of Gaussian primitives. In the third stage, the number of Gaussian primitives is the same as in MVSplat. However, as the reduction of stage index is reduced, the number of Gaussian primitives decreases exponentially by a factor of 4, resulting in significantly fewer Gaussian primitives in the early stages. Therefore, the total number of Gaussian primitives is only 1.3125 (0.0625 + 0.25 + 1) times compared with MVSplat. The comparison of the number of Gaussian primitives and performance is shown in Table 7. Compared with other methods, HiSPlat achieves significant performance improvement with only a slight increase in Gaussian primitives. It is worth noting that even if MVSplat triples the number of Gaussian primitives, it does not yield noticeable improvement, indicating that the increase in Gaussian primitives is not the main reason for HiSplat's performance gain.

Table 7: **Comparison of different Gaussian primitive number.** Compared with other pixel-wise-Gaussian-based methods, HiSplat utilizes hierarchical Gaussian representation, which increases marginally the number of Gaussian primitives and boosts the performance to achieve a better trade-off.

Method	Gaussian Primitive Number	PSNR↑	SSIM↑	LPIPS↓
PixelSplat	65,632(×1.0)	25.89	0.858	0.142
MVSplat(1 Gaussian per pixel)	65,632(×1.0)	26.39	0.869	0.128
MVSplat(3 Gaussian per pixel)	196,896(×3.0)	26.54	0.872	0.127
HiSplat	86,142(×1.3125)	27.21	0.881	0.117

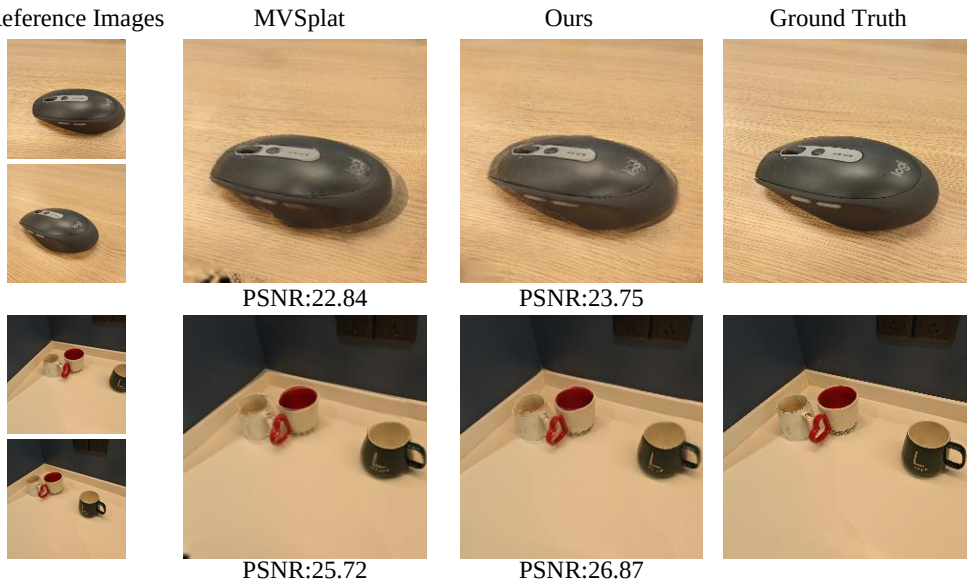

Figure 10: **Pose-free sparse-view reconstruction in real-life scenarios.** We take three photos using a smartphone: two as reference images and one as the ground truth. Then, a pre-trained DUSt3R is used to extract the camera parameters from sparse views. The generated camera parameters and two reference images are inputted in MVSplat or HiSplat for novel view synthesis.

A.7 More generalization cases

To better display the generalization capability of HiSplat, we provide additional visualization cases generated by HiSplat trained on Re10K and tested on DTU. Refer to Fig. 11 for details.

A.8 More related works

In this section, we supplement more related works. FreeSplat (Wang et al., 2024b)proposed a feed-forward generalizable model tailored for indoor scenes, capable of reconstructing global 3D Gaussians from arbitrary numbers of input views. It introduces efficient cross-view feature aggregation and a pixel-wise triplet fusion module to reduce Gaussian redundancy and aggregate multi-view features. Although FreeSplat also employs a multi-scale U-Net as an efficient feature aggregator, the features it ultimately uses to predict Gaussian attributes are from the last layer and still single-scale, resulting in single-scale Gaussian primitives without hierarchical structure. On the contrary, HiSplat leverages multi-scale features from different layers of the U-Net to generate different-scale Gaussians, which represent the scene from bone to flesh.

Splatt3R (Smart et al., 2024) introduced a pose-free, feed-forward method for 3D reconstruction and novel view synthesis from uncalibrated image pairs. Building upon DUSt3R's architecture (Wang et al., 2024a), it predicts 3D Gaussian primitives without requiring camera parameters or depth information. As a promising direction for improvement, HiSplat has the potential to be combined with Splatt3R to handle pose-free scenarios.

CasMVS (Gu et al., 2020) aims to improve memory and time efficiency in multi-view stereo and stereo matching by introducing a hierarchical cost volume formulation. It constructs a feature pyramid to encode geometry and context, narrowing the depth or disparity range at each stage and recovering high-resolution outputs in a coarse-to-fine manner. Although CasMVS uses a similar hierarchical architecture to extract and process features, there are some core technical differences. CasMVS utilizes the depth from the previous stage as the foundation for the next stage's prediction, without incorporating mechanisms for error-aware evaluation of the coarse depth. On the other hand, HiSplat employs the Error Aware Module and Modulating Fusion Module to refine Gaussian primitives across stages, enhancing inter-stage information exchange and improving the robustness of the results. Furthermore, HiSplat's final Gaussian primitives fuse multi-scale output Gaussian

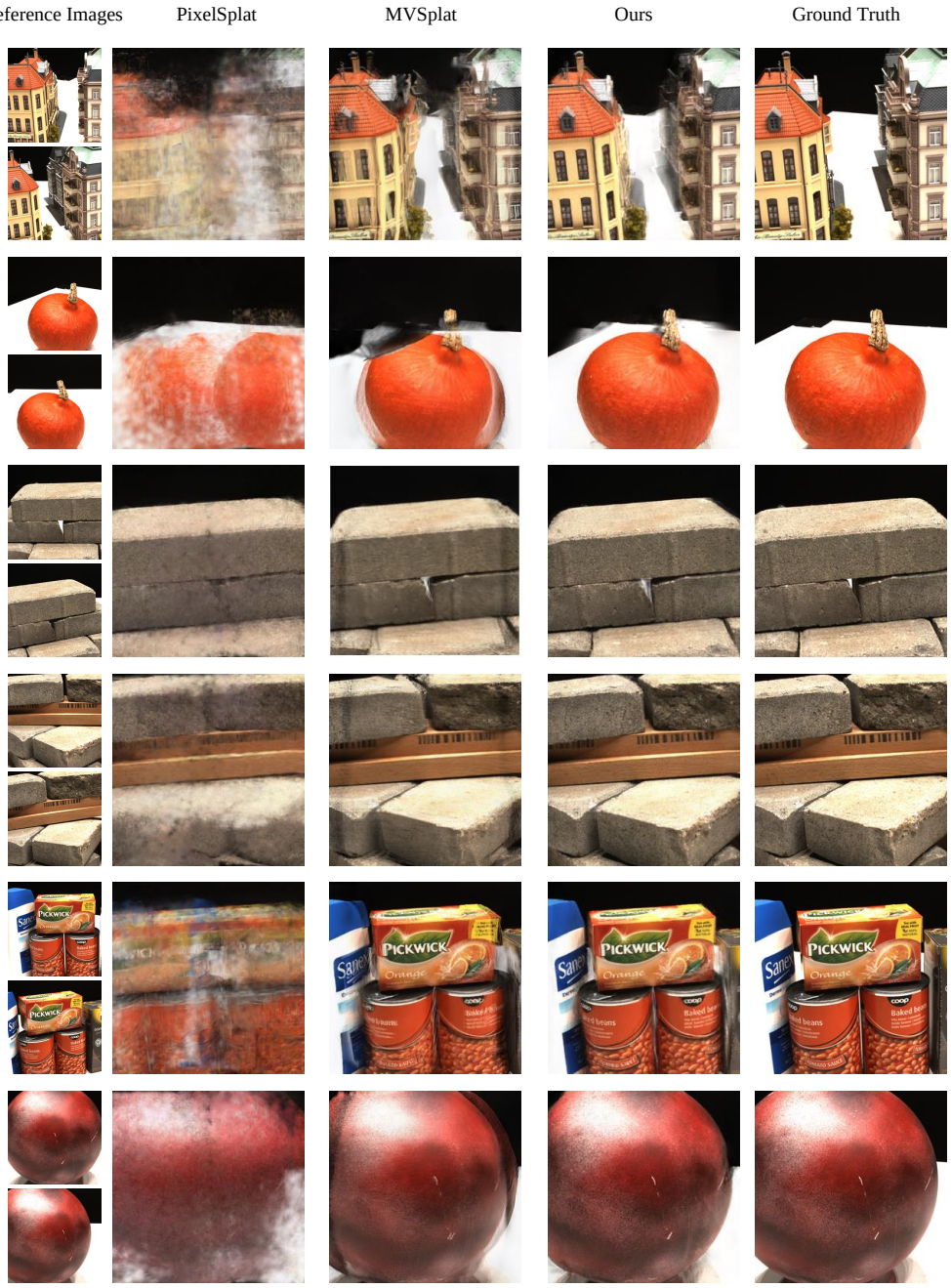

Figure 11: **More cases on DTU to demonstrate generalization ability.**.

primitives in a hierarchical manner, whereas CasMVS outputs the finest level of depth estimation directly, lacking a hierarchical structure.