Seeing the Unseen: Semantic-in-Gaussian for Sparse-View 3D Generalization

Zeyang Bai[1], Yunpeng Wang[2], Yunbiao Wang[1*], and Jun Xiao[1*]

[1] School of Artificial Intelligence, University of Chinese Academy of Sciences, China
[2] Global Institute of Future Technology, Shanghai Jiao Tong University, China

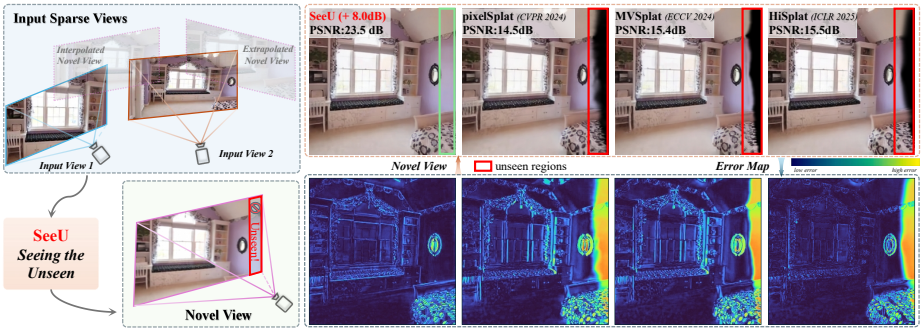

Fig. 1: Comparison of SeeU with sparse-view G-3DGS baselines. Previous methods are pixel-aligned and depth-based, often failing to recover geometry in ambiguous or occluded regions (e.g., unseen surfaces). In contrast, SeeU leverages semantic embeddings to guide the Conditional Gaussian Transformer, which refines 3D Gaussians to enable high-fidelity reconstruction in uncertain regions. Residual maps against the ground-truth novel view are shown in the bottom row.

Abstract. Generalizable 3D Gaussian Splatting (G-3DGS) has emerged as a promising approach for novel view synthesis under sparse-view settings. However, existing frameworks remain restricted by pixel-aligned Gaussian estimation, which struggles in partially observed or occluded regions and often leads to incomplete surfaces or structural collapse. To address these challenges, we propose **SeeU** (**See**ing the **U**nseen), a novel G-3DGS framework. We frame its core design as *Semantic-in-Gaussian*: semantic-conditioned refinement in Gaussian space. Specifically, we introduce a Cross-view Entropy-Aware (CEA) module that aggregates multi-view semantic and geometric cues into compact embeddings. These embeddings guide the Conditional Gaussian Transformer, which applies residual updates to coarse Gaussians, helping recover under-constrained regions of partially observed structures while preserving surface consistency. Comprehensive experiments on multiple benchmarks demonstrate that SeeU consistently improves rendering quality and structural completeness while retaining efficient feed-forward inference. Especially under challenging extrapolation settings, SeeU achieves an average improvement of **2.44 dB** in PSNR compared to recent SOTA G-3DGS methods.

Keywords: Generalizable 3D Gaussian Splatting, Novel View Synthesis, Semantic Embedding, Sparse-view, Conditional Transformer

* Corresponding authors.

1 Introduction

3D reconstruction is a cornerstone of computer vision, powering applications such as autonomous driving, virtual reality, and augmented reality. Recently, 3D Gaussian Splatting (3DGS) [14] has emerged as a powerful paradigm for scene representation and novel view synthesis (NVS) [3]. By modeling a scene as a mixture of 3D Gaussians and leveraging differentiable rasterization, 3DGS enables high-fidelity and real-time rendering from dense multi-view images. However, conventional 3DGS methods typically rely on per-scene optimization with dense input views, which limits scalability and requires costly data acquisition. To alleviate these constraints, generalizable 3DGS (G-3DGS) has been developed to reconstruct scenes from only a few input views. These approaches employ pretrained feed-forward models [5, 6, 22, 32, 34, 35, 42] that encode scene priors from large-scale datasets, enabling rapid inference without scene-specific optimization.

Despite this progress, existing G-3DGS frameworks remain fundamentally restricted by their strong reliance on pixel-aligned unprojection [5, 6]. In these frameworks, estimated depth anchors Gaussian centers to input pixels, causing depth errors to propagate directly into the reconstructed geometry. However, under sparse-view conditions, depth estimation often suffers from occlusions, weak textures, and limited viewpoint overlap. Consequently, pixel-aligned estimation provides insufficient support for occluded or weakly observed parts of partially visible structures, often producing 'black holes' or collapsed geometry in the rendered novel views (Fig. 1).

While previous G-3DGS works attempted to address uncertainty through depth regularization or feature fusion [35,42], they remain inherently constrained by pixel-level priors. Motivated by semantic conditioning in text-to-3D generation [4, 11], we investigate whether semantic priors can compensate for pixel-level uncertainty. Building on this intuition, we introduce **SeeU**, a feed-forward G-3DGS framework that **shifts reconstruction from direct pixel-space estimation to semantic-guided refinement in Gaussian space**. Specifically, SeeU first constructs coarse pixel-aligned latent Gaussians from sparse input views. However, a central challenge is obtaining an effective semantic condition because NVS provides no explicit text prompts. Pseudo-captioning methods such as BLIP [16] offer global semantic cues but often overlook fine-grained structures [24]. We therefore design a **Cross-view Entropy-Aware (CEA)** module that aggregates multi-view semantic cues and uses depth-distribution entropy to emphasize weakly constrained regions. Conditioned on the resulting embeddings, a **Conditional Gaussian Transformer** predicts residual updates to the coarse Gaussians. This refinement supports the recovery of missing surfaces in semantically anchored and partially observed structures while preserving consistency with the input views. By applying a residual refinement within a feed-forward pipeline, SeeU retains efficient inference and reduces its dependence on precise depth alignment.

The main contributions are summarized as follows:

– We introduce **SeeU**, a framework for **semantic**-conditioned refinement **in Gaussian** space. SeeU employs a Conditional Gaussian Transformer to per-

form residual refinement on coarse Gaussians, reducing its dependence on pixel-aligned depth estimation.
- We propose a Cross-view Entropy-Aware module that combines multi-view semantic cues with depth-distribution entropy, providing structure-aware conditioning for weakly constrained regions.
- We evaluate SeeU across interpolation, extrapolation, and zero-shot cross-dataset settings, consistently improving rendering quality. On extrapolation of RealEstate10K [45], SeeU improves PSNR by **+2.32 dB** over recent G-3DGS SOTA method HiSplat [32].

2 Related Work

2.1 Novel View Synthesis

Novel view synthesis (NVS) aims to render photo-realistic images from novel viewpoints using only a limited set of input images [3]. Neural Radiance Fields (NeRF) [1, 2, 9, 21, 26, 40] models scenes as continuous volumetric radiance fields parameterized by neural networks. While NeRF-based methods have yielded impressive results in dense multi-view settings, they typically suffer from slow training times, high memory usage, and suboptimal performance under sparse viewpoints due to their heavy reliance on per-ray MLP evaluations. In contrast, 3D Gaussian Splatting (3DGS) [14, 38, 41] introduces an explicit scene representation by modeling surfaces with anisotropic 3D Gaussian primitives. Each Gaussian is described by its position, covariance, color, and opacity, which can be differentiably rendered via a forward projection to the image plane. This explicit design significantly accelerates the rendering process compared to vanilla NeRF pipelines, yet many existing 3DGS approaches still assume relatively dense coverage of views for accurate geometry and appearance reconstruction. Consequently, their performance deteriorates for extremely sparse inputs, where geometric ambiguity and insufficient texture cues become major challenges.

2.2 Sparse-View Generalizable 3DGS

Sparse-View generalizable 3DGS methods focus on learning a feed-forward model capable of handling unseen scenes without per-scene re-optimization. PixelSplat [5] predict 3D Gaussian parameters from sparse multi-view inputs, leveraging an epipolar transformer for depth estimation. MVSplat [6] relies on cost-volume construction via plane sweeping to infer depth distributions. TranSplat [42] introduces a transformer-based architecture with depth-aware deformable matching for coarse-to-fine refinement. HiSplat [32] integrates hierarchical Gaussian features, leveraging iterative Gaussian alignment. eFreeSplat [22] eliminates epipolar priors by leveraging cross-view completion. DepthSplat [35] bridges Gaussian splatting and depth estimation by leveraging pre-trained monocular depth features to enhance multi-view depth prediction. Despite these diverse approaches, most still rely heavily on estimated depth maps, which can become noisy or

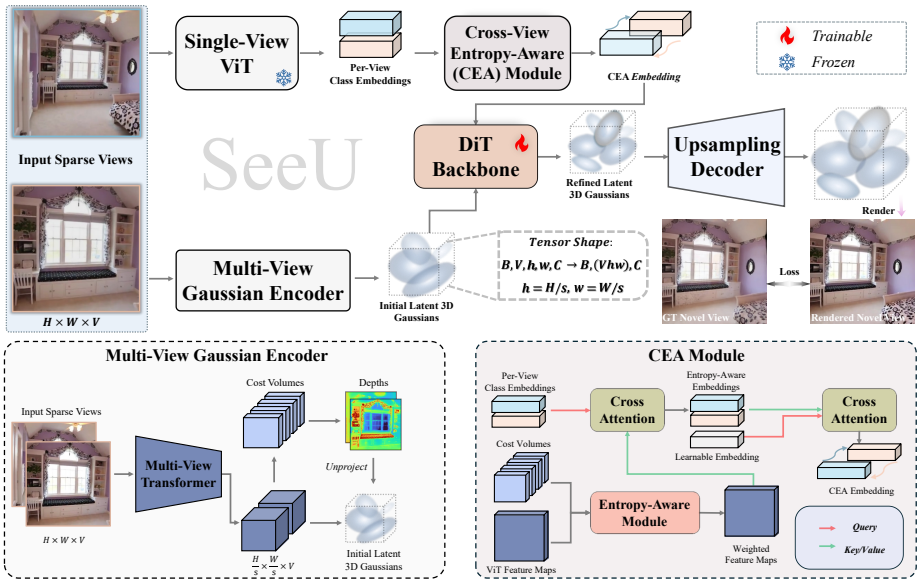

Fig. 2: Overview of SeeU. Given sparse input views, our framework first constructs an initial set of latent 3D Gaussians through the multi-view Gaussian encoder. In parallel, a frozen single-view ViT extracts per-view class embeddings, which are fused by the proposed Cross-view Entropy-Aware (CEA) module to produce a unified multi-view semantic embedding. This embedding guides the DiT-based Conditional Gaussian Transformer to perform residual refinement on the latent Gaussians, followed by an upsampling decoder that restores them to the original spatial resolution. The refined 3D Gaussians are rendered to synthesize novel views, and the entire model is optimized end-to-end using photometric loss.

unreliable under sparse-view conditions. They also often utilize pixel-aligned Gaussian estimation, causing difficulties in recovering fine details or resolving ambiguities in unseen regions.

2.3 Semantic-Conditioned Gaussian Methods

Conditional 3D generation methods use text or image cues to guide content creation. Text-conditioned methods generate point clouds or other 3D representations from language prompts [4, 23], while image-conditioned methods reconstruct 3D assets from single-view or generated multi-view observations [12, 15, 29, 31]. Recent approaches further generate Gaussian representations directly using diffusion-based models [17, 37]. However, these methods primarily target 3D content generation, whereas sparse-view scene reconstruction requires fidelity to observed views and multi-view geometric consistency. SeeU therefore does not generate a scene from noise or employ a diffusion objective. Instead, it uses cross-view semantic embeddings from CEA to condition the Conditional Gaussian Transformer, which performs residual refinement on coarse pixel-aligned

Gaussians. This design provides semantic compensation for under-constrained regions while preserving structural alignment with the input views.

3 Methodology

Following the G-3DGS framework [5, 6, 42], the input consists of V sparse-view images $\mathcal{I} = \{I^1, I^2, \ldots, I^V\}$, where each image $I^i \in \mathbb{R}^{H\times W\times 3}$ is accompanied by its camera projection matrix derived from intrinsic and extrinsic parameters. The goal is to reconstruct the underlying 3D scene as a set of Gaussian primitives $\Theta = \{G_j\}_{j=1}^{N}$, where each primitive G_j is parameterized by its center $\boldsymbol{\mu}_j$, opacity α_j, covariance $\boldsymbol{\Sigma}_j$, and color $\boldsymbol{c}_j$. The number of Gaussians is typically set to $N = H \times W \times V$, corresponding to the input resolution and number of views. These primitives are subsequently rendered into novel views through differentiable Gaussian splatting [14].

As illustrated in Figure 2, SeeU performs semantic-conditioned refinement directly in Gaussian space. The pipeline comprises four stages: Gaussian initialization, Cross-view Entropy-Aware (CEA) conditioning, residual refinement, and Gaussian decoding. First, a multi-view Gaussian encoder extracts cross-view features and constructs coarse pixel-aligned latent Gaussians from the sparse input views. The CEA module then combines semantic features from a frozen ViT with matching uncertainty derived from the cost volumes, producing a cross-view conditioning embedding. Guided by this embedding, the DiT-based Conditional Gaussian Transformer predicts residual updates to refine the coarse latent Gaussians. Finally, an upsampling decoder maps the refined representation to full-resolution Gaussian centers, opacity, covariance, and color for novel-view rendering.

3.1 Gaussian Initialization

A reliable coarse initialization anchors the subsequent residual refinement to the observed views. We use a multi-view Gaussian encoder to construct latent 3D Gaussians directly from sparse input views. By aggregating multi-view features and estimating coarse scene geometry, the encoder produces a structured and pixel-aligned Gaussian representation. These initialized Gaussians serve as the input to the Conditional Gaussian Transformer, which refines their parameters under CEA conditioning.

Latent Feature Extraction To control the computational cost of subsequent Gaussian refinement, we extract multi-view features at a reduced spatial resolution. Each input image I^i is first processed by a shallow ResNet [10] to produce an s-fold downsampled feature map. A multi-view Swin Transformer [19] with cross-view attention then aggregates information across the input views, producing features $\boldsymbol{F}^i \in \mathbb{R}^{\frac{H}{s}\times\frac{W}{s}\times C}$, where C denotes the feature dimension. This latent representation preserves cross-view interactions while reducing the sequence length processed during Gaussian refinement. We use the resulting features to initialize the latent Gaussian parameters.

Coarse Matching To initialize coarse Gaussian geometry and estimate matching uncertainty for CEA conditioning, we construct cost volumes with plane sweeping [36, 39] to model multi-view feature correspondences. Specifically, for each view i, we uniformly sample D depth candidates $\{d_m\}_{m=1}^{D}$ in the inverse depth domain between near and far planes. Features from other views ($\boldsymbol{F}^j, j \neq i$) are warped to view i using camera parameters and depth candidate d_m, producing D warped features $\{\boldsymbol{F}_{d_m}^{j\to i}\}_{m=1}^{D}$. The correlation $\boldsymbol{C}_{d_m}^{i}$ between $\boldsymbol{F}^i$ and $\boldsymbol{F}_{d_m}^{j\to i}$ is computed with the dot-product operation:

$$\boldsymbol{C}_{d_m}^{i} = \frac{F_i \cdot F_{d_m}^{j\to i}}{\sqrt{C}}, \quad m = 1, 2, \ldots, D. \tag{1}$$

We average $\boldsymbol{C}_{d_m}^{i}$ across all other views to form the cost volume $\mathbf{C}^i \in \mathbb{R}^{\frac{H}{s}\times\frac{W}{s}\times D}$. Subsequently, we apply the softmax operation to compute per-view depth map:

$$\boldsymbol{Z}^i = \text{softmax}(\mathbf{C}^i) \cdot \boldsymbol{\Lambda}, \tag{2}$$

where $\boldsymbol{\Lambda} = [d_1, d_2, \ldots, d_D]$ are the depth candidates. The coarse depth map $\boldsymbol{Z}^i \in \mathbb{R}^{\frac{H}{s}\times\frac{W}{s}}$ is then unprojected to form preliminary Gaussian centers $\boldsymbol{\mu}$ using the camera parameters, and other Gaussian parameters are predicted by additional lightweight heads from feature maps. This ensures that the initial positions are geometrically consistent with the input views. The constructed cost volumes $\mathbf{C}^i$ are further retained as conditional inputs to the CEA module, providing pixel-wise structural cues about the scene.

3.2 Conditional Gaussian Transformer

Coarse pixel-aligned Gaussians remain susceptible to depth errors in weakly constrained regions. To correct these errors without discarding geometry anchored by the input views, we introduce a DiT-based Conditional Gaussian Transformer for residual refinement in Gaussian space. Conditioned on CEA embeddings, the module predicts residual updates to the latent Gaussian parameters.

Cross-view Entropy-Aware Embedding Global conditioning signals, such as BLIP-derived text features [16], capture scene-level semantics but may underrepresent fine-grained spatial structures, providing weaker guidance near object boundaries [24]. When extracted independently from each input view, these signals also retain overlapping content without explicitly consolidating cross-view evidence. We therefore introduce a Cross-view Entropy-Aware (CEA) module that uses matching entropy to emphasize weakly constrained regions and aggregates per-view semantics into a compact multi-view embedding.

We first propose an entropy-aware module that leverages the cost volume $\mathbf{C}^i$ to compute the matching entropy H^i, thereby identifying weakly constrained regions (e.g., occluded, weakly observed, or textureless areas). A frozen single-view ViT encoder pretrained with DINOv3 [30] extracts a class embedding $\mathbf{E}_{\text{CLS}}^{i}$

and a spatial feature map $\mathbf{F}^i$ from each view. For each pixel p, we obtain a depth posterior by applying softmax along the depth dimension of the cost volume:

$$P^i(d \mid p) = \mathrm{softmax}_d\big(\mathbf{C}^i(p, d)\big). \tag{3}$$

We then compute the matching entropy as

$$H^i(p) = -\sum_{d \in \Lambda} P^i(d \mid p) \log P^i(d \mid p). \tag{4}$$

Higher entropy indicates greater ambiguity in multi-view matching. We normalize the entropy by its maximum value over D depth candidates:

$$w^i(p) = \frac{H^i(p)}{\log D}, \qquad w^i(p) \in [0, 1]. \tag{5}$$

The resulting weight emphasizes weakly constrained locations in the spatial feature map:

$$\tilde{\mathbf{F}}^i(p) = w^i(p)\mathbf{F}^i(p). \tag{6}$$

We project the class embedding into a query and the entropy-weighted spatial features into keys and values:

$$\mathbf{Q}^i = \mathbf{E}^i_{\mathrm{CLS}}\mathbf{W}_Q, \quad \mathbf{K}^i = \tilde{\mathbf{F}}^i\mathbf{W}_K, \quad \mathbf{V}^i = \tilde{\mathbf{F}}^i\mathbf{W}_V. \tag{7}$$

Cross-attention then incorporates uncertainty-weighted spatial cues into the per-view semantic embedding:

$$\tilde{\mathbf{E}}^i_{\mathrm{CLS}} = \mathrm{CrossAttn}\big(\mathbf{Q}^i, \mathbf{K}^i, \mathbf{V}^i\big)\,. \tag{8}$$

To aggregate complementary evidence across views while compressing overlapping content, we employ cross-view Perceiver-style attention with a set of learnable latent queries $\mathbf{Q}_\ell$. We concatenate the refined per-view embeddings $\tilde{\mathbf{E}}^i_{\mathrm{CLS}}$ and project them into keys $\mathbf{K}_{\mathrm{mv}}$ and values $\mathbf{V}_{\mathrm{mv}}$. The CEA embedding is then computed as

$$\mathbf{E}_{\mathrm{CEA}} = \mathrm{CrossAttn}\big(\mathbf{Q}_\ell,\, \mathbf{K}_{\mathrm{mv}},\, \mathbf{V}_{\mathrm{mv}}\big). \tag{9}$$

$\mathbf{E}_{\mathrm{CEA}}$ summarizes uncertainty-aware semantics across views. It conditions the residual refinement performed by the Conditional Gaussian Transformer.

Gaussian-Structured Representation Although Gaussian primitives conceptually form a set, our initialization preserves a pixel-to-Gaussian correspondence. We therefore organize the latent parameters as $\Theta_l \in \mathbb{R}^{B\times V\times h\times w\times C}$, where B is the batch size, V is the number of views, $h \times w = \frac{H}{s} \times \frac{W}{s}$ is the latent spatial resolution, and C is the Gaussian parameter dimension. Applying a convolutional architecture such as UNet [27] would impose fixed local grid neighborhoods on this tensor, although nearby image locations need not correspond to nearby

Table 1: Comparison of interpolated NVS. We evaluate performance on the RealEstate10K and ACID datasets by rendering three novel interpolation views from two reference viewpoints, averaging across all scenes. The dataset's training and testing split follows the identical protocol established by pixelSplat. Note that 3DGS-based methods render extremely fast ($\sim$ 500FPS).

	RealEstate10K			ACID			Inference Time
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	(s)
pixelNeRF [40]	20.43	0.589	0.550	20.97	0.547	0.533	5.299
MuRF [9]	26.10	0.858	0.143	28.09	0.841	0.155	0.186
pixelSplat [5]	25.89	0.858	0.142	28.14	0.839	0.150	0.104
MVSplat [6]	26.39	0.869	0.128	28.25	0.843	0.144	**0.044**
eFreeSplat [22]	26.45	0.865	0.126	28.30	0.851	0.140	0.061
TranSplat [42]	26.69	0.875	0.125	28.35	0.845	0.143	0.087
HiSplat [32]	27.21	0.881	0.117	28.75	0.853	**0.133**	0.510
SeeU (Ours)	**27.56**	**0.888**	**0.114**	**28.77**	**0.855**	**0.133**	0.089

Gaussians in 3D. We instead use a DiT-based Transformer module to model interactions among Gaussian tokens. Specifically, we flatten the structured tensor into a token sequence:

$$\mathbf{T}_l = \texttt{rearrange}(\Theta_l,\ \texttt{B V h w C} \rightarrow \texttt{B (Vhw) C}), \tag{10}$$

where $\mathbf{T}_l \in \mathbb{R}^{B\times(Vhw)\times C}$ is processed by the DiT architecture [25]. Rather than regressing the refined Gaussians directly, the module predicts a residual update to the coarse parameters:

$$\hat{\Theta}_l = \Theta_l + f_\theta(\Theta_l, \mathbf{E}_{\text{CEA}}), \tag{11}$$

where f_θ is the DiT-based predictor conditioned on the CEA embedding $\mathbf{E}_{\text{CEA}}$. This residual formulation corrects errors in weakly constrained regions while preserving geometry that is already well anchored by the input views.

3.3 Rendering and Training Loss

The upsampling decoder produces the full-resolution Gaussian set Θ, which is rendered from target viewpoints using differentiable Gaussian rasterization [14]. We optimize SeeU end-to-end through image-space supervision. The training objective combines a pixel-wise ℓ_2 reconstruction term with an LPIPS perceptual term [43]:

$$\mathcal{L}_{\text{photo}} = \|\mathcal{R}(\Theta) - \mathcal{I}_{\text{gt}}\|_2^2 + 0.05 \cdot \text{LPIPS}(\mathcal{R}(\Theta), \mathcal{I}_{\text{gt}}), \tag{12}$$

where $\mathcal{R}(\Theta)$ denotes the target-view image rendered from Θ. Following prior G-3DGS methods [5, 6], we set the LPIPS weight to 0.05.

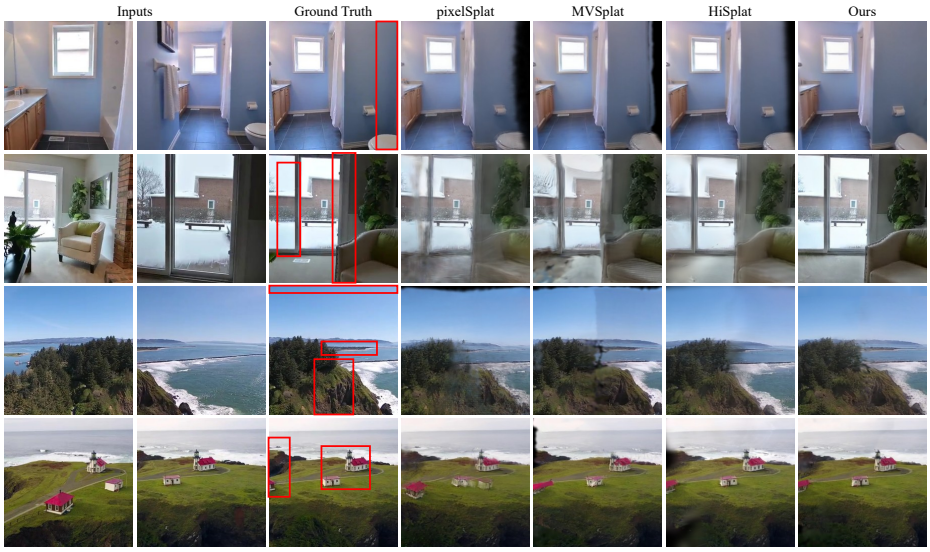

Fig. 3: Qualitative comparison of interpolated NVS. The first two columns show sparse input views, while the third column presents the ground truth for the target interpolated view between them. SeeU better reconstructs fine details and occluded regions (highlighted in red) in both indoor (RealEstate10K, top two rows) and outdoor (ACID, bottom two rows) settings.

4 Experiments and Discussions

4.1 Experimental Settings

For within-dataset experiments, we train and evaluate SeeU on two large-scale datasets, RealEstate10K [45] and ACID [18]. RealEstate10K contains 67,477 training scenes and 7,289 test scenes, while ACID contains 11,075 training scenes and 1,972 test scenes. Both datasets provide per-frame camera intrinsics and extrinsics estimated through Structure-from-Motion (SfM) [28]. Following prior works [5,6], we use two context images and render three target views for each test scene. We evaluate both interpolated and extrapolated NVS, with extrapolated target views lying outside the reference-view range. For extrapolation training, we follow the pixelSplat curriculum [5] and increase the target-view sampling interval to 45 frames before and after the reference views. We report PSNR, SSIM [33], and LPIPS [44] for rendering quality. For zero-shot cross-dataset evaluation, the model trained on RealEstate10K is evaluated on 16 DTU validation scenes [13], with four novel target views per scene and no fine-tuning, following the MVSplat protocol [6]. On DTU, we additionally report depth RMSE and Overall Chamfer to assess geometric accuracy. Implementation details, ACID cross-dataset results, and evaluations with more input views are provided in the Appendix.

Table 2: Comparison of extrapolated NVS on RealEstate10K. We evaluate model performance on RealEstate10K by rendering three novel extrapolated views from two reference views, averaging across all scenes under identical training settings.

Method	PSNR↑	SSIM↑	LPIPS↓
pixelSplat [5]	21.76	0.779	0.217
MVSplat [6]	21.92	0.787	0.199
TranSplat [42]	21.89	0.791	0.201
HiSplat [32]	22.01	0.794	0.191
SeeU (Ours)	**24.33** (+2.32)	**0.846** (+0.052)	**0.152** (−0.039)

Fig. 4: Qualitative results for extrapolated NVS on RealEstate10K. Extrapolated target views lie outside the input trajectory where novel pixels lack reference correspondences from input views. Baseline methods exhibit voids and distorted geometry within these unseen regions while SeeU reduces missing-geometry artifacts and preserves boundary structures.

4.2 Main Results

Interpolated Novel View Synthesis Table 1 shows that SeeU achieves better performance on both RealEstate10K and ACID. Figure 3 shows that, compared with pixel-aligned G-3DGS baselines, SeeU better preserves fine structures and reduces artifacts in occluded regions through CEA-conditioned residual refinement. The full framework runs at 0.089 s per frame, remaining competitive with lightweight baselines and substantially faster than HiSplat (0.510 s). Additional qualitative results on RealEstate10K are provided in the Appendix.

Table 3: Zero-shot evaluation on DTU. All methods are trained on RealEstate10K and evaluated on DTU without fine-tuning.

Method	PSNR↑	SSIM↑	LPIPS↓	Depth RMSE↓	Overall Chamfer↓
MVSplat [6]	13.94	0.473	0.385	0.847	7.37
HiSplat [32]	16.05	0.671	0.277	0.958	8.41
SeeU (Ours)	**16.31**	**0.704**	**0.269**	**0.841**	**6.75**

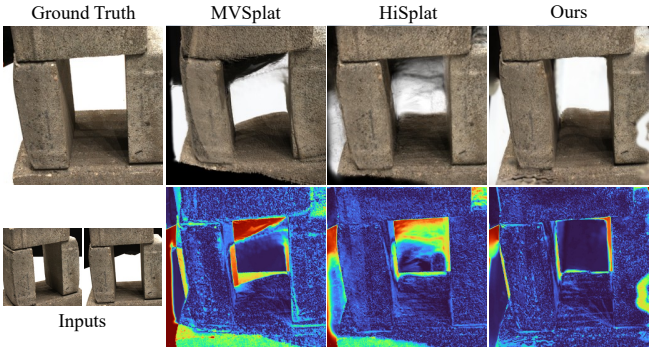

Fig. 5: Zero-shot qualitative comparison on DTU. All models are trained on RealEstate10K and evaluated on DTU without fine-tuning. The lower-right three panels visualize residuals between the rendered novel views and ground truth.

Extrapolated Novel View Synthesis Extrapolated NVS evaluates target viewpoints outside the reference-view range, where limited input-view support increases reconstruction ambiguity. For a fair comparison, we retrain all baselines using the same extrapolation-aware training protocol as SeeU, with target views sampled up to 45 frames before and after the reference-view interval. As summarized in Table 2, SeeU outperforms all baselines, exceeding HiSplat by 2.32 dB in PSNR while reducing LPIPS by 20%. Figure 4 shows that representative pixel-aligned baselines, MVSplat and HiSplat, exhibit voids and geometric artifacts in weakly constrained regions, whereas SeeU better preserves boundary structures. Its CEA-conditioned residual refinement incorporates semantic cues beyond pixel-aligned correspondences directly in Gaussian space.

Zero-Shot Cross-Dataset Evaluation on DTU Table 3 shows that SeeU improves rendering fidelity and geometric accuracy simultaneously under cross-dataset transfer. It achieves better PSNR, SSIM, and LPIPS while also obtaining the lower Depth RMSE (0.841) and Overall Chamfer (6.75). Notablely, although HiSplat is the strongest rendering baseline, its geometry errors indicate that image-space quality alone does not guarantee accurate 3D reconstruction. In Figure 5, SeeU yields smaller residuals and better preserves the boundaries of the partially observed structure. We attribute this robustness to CEA-conditioned Gaussian refinement, which emphasizes weakly constrained regions using cross-view semantics and matching uncertainty, thereby reducing reliance on precise pixel-aligned depth estimates under domain shift.

Table 4: Ablations of Gaussian refinement. Models trained on RealEstate10K with two input views. *w/o Refinement* renders directly from the initial Gaussians, while *w/ UNet Backbone* replaces the Conditional Gaussian Transformer with a 2D UNet. Green indicates drops relative to the full model.

Setup	PSNR↑	SSIM↑	LPIPS↓
w/o Refinement	25.47 (-2.09)	0.849 (-0.039)	0.149 (+0.035)
w/ UNet Backbone	26.71 (-0.85)	0.873 (-0.015)	0.123 (+0.009)
Full Model	**27.56**	**0.888**	**0.114**

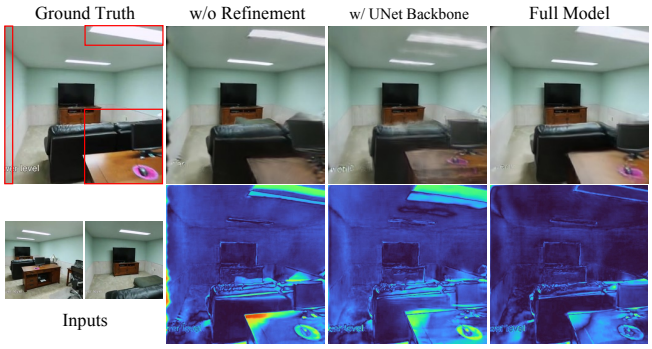

Fig. 6: Qualitative ablations of Gaussian refinement. The second row displays residual maps between rendered images and the ground truth. Removing Gaussian refinement (*w/o Refinement*) leads to structural distortions and incomplete geometry, while replacing the Conditional Gaussian Transformer with a 2D UNet produces blurry edges. The full model produces sharper, more complete reconstructions with improved geometric consistency.

4.3 Ablation Studies

Ablations on Gaussian Refinement We ablate Gaussian refinement from two aspects: (i) removing the refiner (*w/o Refinement*), where the initial Gaussians are directly upsampled for rendering, and (ii) replacing the Conditional Gaussian Transformer with a 2D UNet (*w/ UNet Backbone*). Quantitative comparisons are summarized in Table 4, and qualitative examples under challenging viewpoint shifts are given in Figure 6. Removing refinement leads to pronounced geometric degradation, including collapsed structures and missing surfaces around occlusion boundaries. This confirms that the initial pixel-aligned Gaussians alone are insufficient for reliable reconstruction in sparse-view settings. Replacing the Conditional Gaussian Transformer with a UNet produces more complete geometry than the no-refinement baseline but introduces blurred contours and texture artifacts (e.g., distorted lighting). We attribute this to convolutional inductive biases that impose rigid 2D grid priors misaligned with the unordered nature of Gaussian primitives.

Table 5: Ablations on conditional embeddings. Models trained on RealEstate10K with two input views. CEA outperforms DINOv3, CLIP, and BLIP-2 under identical conditioning interfaces and training settings. Green indicates drops relative to the best model (CEA).

Setup	PSNR↑	SSIM↑	LPIPS↓
Class Embedding	26.13 (-1.43)	0.859 (-0.029)	0.139 (+0.025)
CLIP Embedding	26.41 (-1.15)	0.867 (-0.021)	0.126 (+0.012)
BLIP Embedding	26.54 (-1.02)	0.870 (-0.018)	0.125 (+0.011)
CEA Embedding	**27.56**	**0.888**	**0.114**

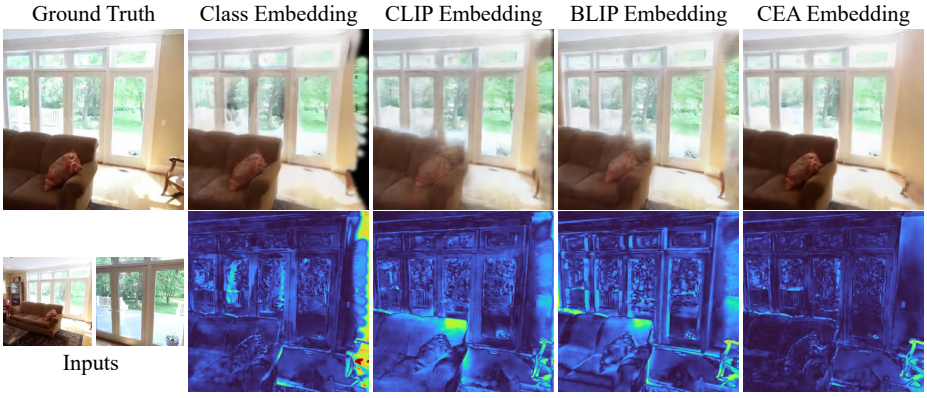

Fig. 7: Qualitative ablations of various embeddings. The second row displays error maps that indicate differences from the ground truth. Models using class embeddings exhibit artifacts near boundaries. Models conditioned on CLIP and BLIP embeddings exhibit errors along fine structures such as sofa contours. The model utilizing the CEA embedding produces sharper edges and exhibits fewer geometric errors.

Ablations on conditional embeddings To assess the role of conditioning, we compare our Cross-view Entropy-Aware (CEA) embedding with three alternatives under identical training and injection configurations: (i) a DINOv3 class embedding, (ii) a CLIP image–text embedding, and (iii) a BLIP-2 pseudo-caption embedding. Notably, all pre-trained encoders are kept frozen. Their extracted embeddings are linearly projected into a shared 768-dimensional token space and injected via the exact same cross-attention interface, ensuring that the embedding source is the sole variable. As summarized in Table 5, CEA achieves consistent gains across all metrics, improving PSNR by +1.02 dB and reducing LPIPS by $\sim 8.8\%$ relative to the next best variant (BLIP). These gains stem from CEA's ability to fuse semantics across views while weighting features according to geometric uncertainty. By emphasizing ambiguous or weakly constrained regions, the CEA embedding provides more reliable guidance to the Conditional Gaussian Transformer than single-view or text-derived embeddings, which often capture only global or redundant semantics. Qualitative comparisons in Figure 7 corroborate these trends. Embeddings derived from individual views (DINOv3)

Table 6: Ablations on extrapolation training range. We vary the sampling range of extrapolated target views during training on RealEstate10K with two input views. *w/ no ext.* disables extrapolated-view sampling, whereas *ext. N frames* samples targets within N frames before and after the input-view range. Green indicates drops relative to the 45-frame setting.

Variant	PSNR↑	SSIM↑	LPIPS↓
SeeU w/ no ext.	22.78 (-1.55)	0.804 (-0.042)	0.178 (+0.026)
SeeU ext. 90 frames	23.41 (-0.92)	0.820 (-0.026)	0.169 (+0.017)
SeeU ext. 45 frames	**24.33**	**0.846**	**0.152**

tend to overlook boundary information and may introduce spurious artifacts (e.g., tree-like structures outside the window). CLIP and BLIP embeddings better capture salient global content but still struggle with fine structures such as sofa contours. This could be attributed to their sensitivity to multi-view overlap and redundancy, which diminishes the attention to unique objects appearing only once. In contrast, CEA recovers sharper edges and more complete geometry, demonstrating its capacity to highlight uncertain regions while suppressing redundancy.

Extrapolation Training Range We vary the range of extrapolated target views sampled during RealEstate10K training. Table 6 shows that removing extrapolated targets reduces rendering performance, demonstrating the importance of extrapolation-aware supervision. The 90-frame setting still improves over no extrapolation, but remains consistently below the 45-frame setting. At very large offsets, target views may expose substantially more content with limited support from the inputs, increasing reconstruction ambiguity and weakening the supervision signal. A moderate range instead presents challenging out-of-interval views while retaining sufficient visual evidence to associate semantic cues with scene geometry. This balance improves rendering fidelity and structural preservation, indicating that overly aggressive extrapolation does not necessarily yield better generalization.

5 Conclusion

In this work, we propose SeeU, a novel G-3DGS framework for novel view synthesis from sparse-view inputs. Unlike prior methods that rely on pixel-aligned Gaussian estimation, our approach uses a CEA-conditioned Conditional Gaussian Transformer to apply residual updates directly in Gaussian space, improving geometry in partially observed or uncertain scenes. To enhance both global semantic coherence and fine-grained detail awareness, we introduce a Cross-view Entropy-Aware embedding that provides semantically rich guidance for Gaussian refinement. Extensive experiments on multiple datasets demonstrate that SeeU consistently enhances reconstruction fidelity, particularly in occluded and unseen regions. These results highlight the potential of semantic-conditioned refinement in Gaussian space for generalizable 3D reconstruction.

References

1. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: ICCV. pp. 5835–5844 (2021). https://doi.org/10.1109/ICCV48922.2021.00580
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: CVPR. pp. 5460–5469 (2022). https://doi.org/10.1109/CVPR52688.2022.00539
3. Buehler, C., Bosse, M., McMillan, L., Gortler, S., Cohen, M.: Unstructured lumigraph rendering. In: SIGGRAPH. p. 425–432 (2001). https://doi.org/10.1145/383259.383309
4. Cao, Z., Hong, F., Wu, T., Pan, L., Liu, Z.: Large-vocabulary 3d diffusion model with transformer. In: ICLR (2024), https://openreview.net/forum?id=q57JLSE2j5
5. Charatan, D., Li, S.L., Tagliasacchi, A., Sitzmann, V.: Pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In: CVPR. pp. 19457–19467 (2024). https://doi.org/10.1109/CVPR52733.2024.01840
6. Chen, Y., Xu, H., Zheng, C., Zhuang, B., Pollefeys, M., Geiger, A., Cham, T.J., Cai, J.: Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In: ECCV. pp. 370–386 (2025)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
8. Fang, G., Li, K., Ma, X., Wang, X.: Tinyfusion: Diffusion transformers learned shallow. In: CVPR. pp. 18144–18154 (2025). https://doi.org/10.1109/CVPR52734.2025.01691
9. Haofei, X., Chen, A., Chen, Y., Sakaridis, C., Zhang, Y., Pollefeys, M., Geiger, A., Yu, F.: Murf: Multi-baseline radiance fields. In: CVPR. pp. 20041–20050 (2024)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
11. He, X., Chen, J., Peng, S., Huang, D., Li, Y., Huang, X., Yuan, C., Ouyang, W., He, T.: Gvgen: Text-to-3d generation with volumetric representation. In: ECCV. pp. 463–479 (2025)
12. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: Lrm: Large reconstruction model for single image to 3d. In: ICLR (2024), https://openreview.net/forum?id=sllU8vvsFF
13. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: CVPR. pp. 406–413 (2014). https://doi.org/10.1109/CVPR.2014.59
14. Kerbl, B., Kopanas, G., Leimkuehler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (2023). https://doi.org/10.1145/3592433
15. Li, J., Tan, H., Zhang, K., Xu, Z., Luan, F., Xu, Y., Hong, Y., Sunkavalli, K., Shakhnarovich, G., Bi, S.: Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. In: ICLR (2024), https://openreview.net/forum?id=2lDQLiH1W4
16. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In: ICML. pp. 19730–19742 (2023)

17. Lin, C., Pan, P., Yang, B., Li, Z., Mu, Y.: Diffsplat: Repurposing image diffusion models for scalable gaussian splat generation. In: ICLR (2025), https://openreview.net/forum?id=eajZpoQkGK
18. Liu, A., Makadia, A., Tucker, R., Snavely, N., Jampani, V., Kanazawa, A.: Infinite nature: Perpetual view generation of natural scenes from a single image. In: ICCV. pp. 14438–14447 (2021). https://doi.org/10.1109/ICCV48922.2021.01419
19. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV. pp. 10012–10022 (2021)
20. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
21. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV. pp. 405–421 (2020)
22. Min, Z., Luo, Y., Sun, J., Yang, Y.: Epipolar-free 3d gaussian splatting for generalizable novel view synthesis. In: NeurIPS (2024), https://openreview.net/forum?id=iO6tcLJEwA
23. Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., Chen, M.: Point-e: A system for generating 3d point clouds from complex prompts (2022), https://arxiv.org/abs/2212.08751
24. Patni, S., Agarwal, A., Arora, C.: Ecodepth: Effective conditioning of diffusion models for monocular depth estimation. In: CVPR. pp. 28285–28295 (2024)
25. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: CVPR. pp. 4195–4205 (2023)
26. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: Neural radiance fields for dynamic scenes. In: CVPR. pp. 10313–10322 (2021). https://doi.org/10.1109/CVPR46437.2021.01018
27. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. pp. 234–241 (2015)
28. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR. pp. 4104–4113 (2016). https://doi.org/10.1109/CVPR.2016.445
29. Shi, Y., Wang, P., Ye, J., Mai, L., Li, K., Yang, X.: MVDream: Multi-view diffusion for 3d generation. In: ICLR (2024), https://openreview.net/forum?id=FUgrjq2pbB
30. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J., Brandt, J., Couprie, C., Mairal, J., Jégou, H., Labatut, P., Bojanowski, P.: DINOv3 (2025), https://arxiv.org/abs/2508.10104
31. Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., Liu, Z.: Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In: ECCV. pp. 1–18 (2025)
32. Tang, S., Ye, W., Ye, P., Lin, W., Zhou, Y., Chen, T., Ouyang, W.: Hisplat: Hierarchical 3d gaussian splatting for generalizable sparse-view reconstruction. In: ICLR (2025), https://openreview.net/forum?id=SBzIbJojs8
33. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing **13**(4), 600–612 (2004). https://doi.org/10.1109/TIP.2003.819861
34. Wewer, C., Raj, K., Ilg, E., Schiele, B., Lenssen, J.E.: Latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. In: ECCV. pp. 456–473 (2025)

35. Xu, H., Peng, S., Wang, F., Blum, H., Barath, D., Geiger, A., Pollefeys, M.: Depthsplat: Connecting gaussian splatting and depth. In: CVPR. pp. 16453–16463 (2025). https://doi.org/10.1109/CVPR52734.2025.01534
36. Xu, H., Zhang, J., Cai, J., Rezatofighi, H., Yu, F., Tao, D., Geiger, A.: Unifying flow, stereo and depth estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(11), 13941–13958 (2023)
37. Yang, Y., Shao, J., Li, X., Shen, Y., Geiger, A., Liao, Y.: Prometheus: 3d-aware latent diffusion models for feed-forward text-to-3d scene generation. In: CVPR. pp. 2857–2869 (2025)
38. Yang, Z., Gao, X., Zhou, W., Jiao, S., Zhang, Y., Jin, X.: Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In: CVPR. pp. 20331–20341 (2024). https://doi.org/10.1109/CVPR52733.2024.01922
39. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: ECCV. pp. 767–783 (2018)
40. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: CVPR. pp. 4576–4585 (2021). https://doi.org/10.1109/CVPR46437.2021.00455
41. Yu, Z., Chen, A., Huang, B., Sattler, T., Geiger, A.: Mip-splatting: Alias-free 3d gaussian splatting. In: CVPR. pp. 19447–19456 (2024). https://doi.org/10.1109/CVPR52733.2024.01839
42. Zhang, C., Zou, Y., Li, Z., Yi, M., Wang, H.: Transplat: Generalizable 3d gaussian splatting from sparse multi-view images with transformers. In: AAAI. pp. 9869–9877 (2025). https://doi.org/10.1609/aaai.v39i9.33070
43. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR. pp. 586–595 (2018)
44. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR. pp. 586–595 (2018). https://doi.org/10.1109/CVPR.2018.00068
45. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: learning view synthesis using multiplane images. ACM Transactions on Graphics **37**(4) (2018). https://doi.org/10.1145/3197517.3201323

A Implementation Details

We implement SeeU in PyTorch. Unless otherwise specified, all input images are resized to 256×256, and two sparse input views are used. With a downsampling factor of $s = 4$, the latent resolution is 64×64. For semantic feature extraction, we use a frozen ViT-Base encoder [7] pretrained with DINOv3 [30]. The Conditional Gaussian Transformer follows the TinyDiT architecture [8] and comprises four layers with a hidden size of 256, four attention heads, and an MLP ratio of 2. We train the model for 300,000 iterations on one NVIDIA H800 GPU using AdamW [20] with a batch size of 8.

B Additional Discussion

B.1 Zero-Shot Evaluation on ACID

We additionally evaluate the RealEstate10K-trained models on ACID without fine-tuning. As shown in Table 7, SeeU achieves the best PSNR and SSIM, improving over HiSplat by 0.04 dB and 0.003, respectively, while HiSplat retains a marginal 0.001 advantage in LPIPS. Figure 8 provides the corresponding qualitative comparison under this domain shift.

Table 7: Zero-shot evaluation on ACID. All methods are trained on RealEstate10K and evaluated on ACID without fine-tuning.

Method	PSNR↑	SSIM↑	LPIPS↓
pixelSplat [5]	27.64	0.830	0.160
MVSplat [6]	28.15	0.841	0.147
TranSplat [42]	28.17	0.842	0.146
HiSplat [32]	28.66	0.850	**0.137**
SeeU (Ours)	**28.70**	**0.853**	0.138

B.2 Generalization to More Input Views

To assess the generalizability of our model under varying numbers of sparse input views, we conduct a zero-shot evaluation by directly applying the model trained with 2-view inputs in RealEstate10K to a 3-view input setting on DTU. As reported in Table 8, our method demonstrates robust performance when evaluated with more input views.

B.3 More Visual Comparisons

We provide additional qualitative comparisons on the RealEstate10K dataset, as shown in Figure 9. PixelSplat and MVSplat frequently exhibit severe blurring and geometric distortions, while HiSplat produces sharper outputs but still

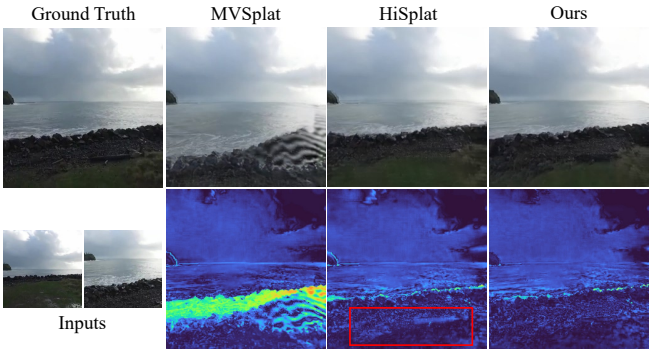

Fig. 8: Zero-shot qualitative comparison on ACID. All models are trained on RealEstate10K and evaluated on ACID without fine-tuning.

Table 8: Quantitative comparison of 3-view cross-dataset generalization.

Methods	PSNR ↑	SSIM ↑	LPIPS ↓
pixelSplat [5]	12.52	0.367	0.585
MVSplat [6]	14.30	0.508	0.371
HiSplat [32]	16.34	0.674	**0.286**
SeeU (Ours)	**16.44** (+0.1)	**0.729** (+0.055)	0.298 (+0.012)

fails to recover boundaries and occluded regions. In contrast, SeeU consistently generates sharper novel views, preserving furniture outlines and textile patterns.

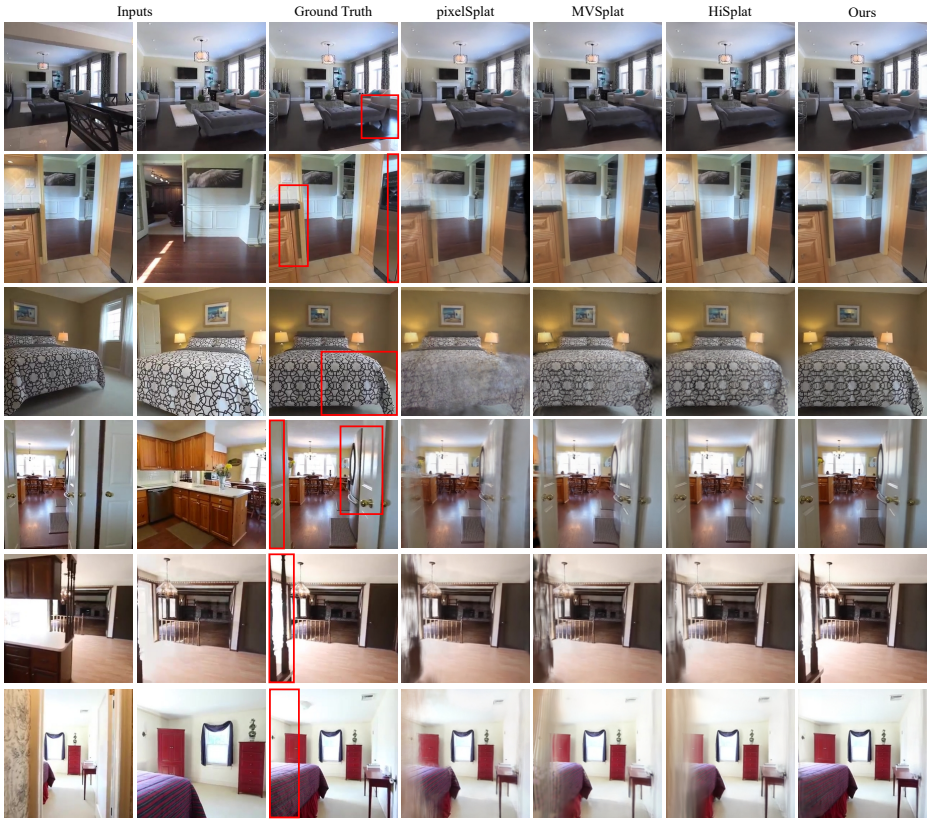

Fig. 9: More comparisons on RealEstate10K. Qualitative comparison of novel view synthesis results across different methods. Red boxes highlight challenging regions where baseline methods exhibit artifacts or texture loss. In contrast, SeeU produces sharper and more structurally consistent renderings, refining object contours and details (e.g., furniture edges, patterned bedspreads).