

iSplat: Iterative Learning for Fine-Grained Gaussian Splatting

Haifeng Wu, Wei Long, Shuhang Gu, Lixin Duan, Wen Li*

University of Electronic Science and Technology of China

haifengwu205@gmail.com, lwsch5940@163.com, {shuhangu, lxduan, liwenbnu}@gmail.com

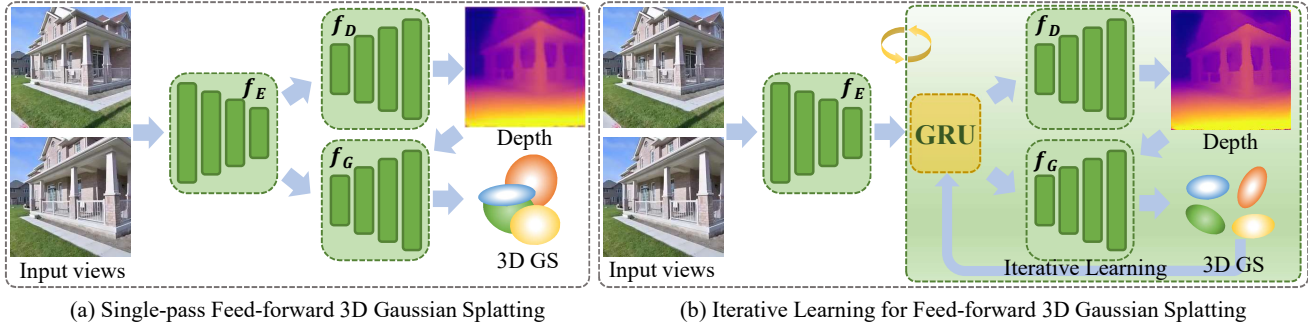


Figure 1. **Comparison of single-pass and iterative feed-forward 3D Gaussian splatting.** (a) Existing approaches [5, 33] estimate 3D Gaussian parameters through a single-pass, depth-conditioned regression pipeline. Lacking any mechanism for refinement, initial geometric errors are irreversibly propagated, often leading to suboptimal reconstructions. (b) Our proposed iSplat reformulates reconstruction as an iterative refinement process. It employs a GRU-based recurrent optimizer to synergistically update geometry and appearance over multiple passes. This allows the model to progressively self-correct through our proposed design choices, yielding more robust reconstructions.

Abstract

Recent advances in feed-forward 3D Gaussian splatting have demonstrated remarkable efficiency by reconstructing scenes in a single pass. However, the reconstruction fidelity of these methods lags behind that of traditional optimization-based approaches, which gradually correct reconstruction flaws through a lengthy iterative process. In this paper, we leverage the strengths of both paradigms and introduce **iSplat**, a novel framework that reformulates reconstruction as an iterative feed-forward process involving multiple (typically three) passes. Central to iSplat is a recurrent GRU-based optimizer that refines both geometry and appearance in a synergistic loop. To address geometric inaccuracies, we propose an uncertainty-driven depth refinement strategy that progressively narrows the search space for each Gaussian based on its estimated uncertainty from the previous step. To further improve appearance details, we design a region-aware enhancement mechanism that applies targeted multi-view and monocular feature aggregation to resolve ambiguities in both overlapping and non-overlapping areas. We validate iSplat’s robustness and generalization on in-domain (RealEstate10K, ACID) and cross-dataset (DTU, ACID) benchmarks. With only **42.6M** parameters, iSplat surpasses DepthSplat (354M) on RealEstate10K (PSNR: **27.67** vs. **27.47 dB**). Crucially,

on the cross-dataset DTU benchmark, it further boosts the PSNR by 2.88 dB (**18.26** vs. **15.38 dB**), showcasing exceptional generalization. These results highlight the significant potential of iterative learning to overcome the inherent limitations of one-shot approaches. Our code is available at <https://github.com/haifengwu205/iSplat>.

1. Introduction

Recent advances in feed-forward 3D Gaussian Splatting (3DGS) have established a new paradigm for efficient and generalizable 3D scene reconstruction. Unlike optimization-based 3DGS approaches that require per-scene fitting [14], feed-forward models directly predict the parameters of 3D Gaussians such as position, rotation, scale, color, and opacity from sparse multi-view observations through a single forward pass [4, 5, 15, 18, 33]. This direct inference eliminates the need for time-consuming scene-specific optimization and volumetric sampling as used in Neural Radiance Fields (NeRFs) [20], enabling real-time, high-fidelity rendering. By leveraging large-scale training datasets, these methods learn powerful geometric and photometric priors that support robust generalization across diverse environments, representing a crucial step toward scalable deployment in robotics, augmented reality, and autonomous perception systems.

Despite their efficiency, existing feed-forward 3DGS ap-

* Corresponding author.

proaches are fundamentally constrained by their single-pass pipeline (Fig. 1 (a)). These methods typically first estimate a depth map as a geometric scaffold and then predict all Gaussian parameters in a single, subsequent step [4, 5, 15, 18, 33]. This one-shot formulation, while computationally lean, presents a critical scientific challenge: it establishes a brittle dependency on the initial depth scaffold, where any estimation errors are irreversibly propagated into the final Gaussian attributes. By collapsing the complex task of 3D reconstruction into a single predictive step, these models are deprived of any mechanism for self-correction or progressive refinement. Consequently, their capacity to resolve geometric ambiguities such as those in poorly constrained, non-overlapping regions and render fine-grained appearance details is inherently limited, as the initial, coarse geometric estimate dictates the final, detailed representation. In contrast, optimization-based methods [14, 32] achieve superior fidelity through iterative refinement, albeit at a substantial computational cost. This highlights a compelling need for a new paradigm that integrates the adaptive precision of optimization within an efficient, learnable feed-forward framework.

In this paper, we propose iSplat, an iterative feed-forward framework designed to bridge the gap between single-pass efficiency and optimization-based fidelity (Fig. 1 (b)). Instead of relying on a one-shot prediction, iSplat formulates 3D Gaussian estimation as a learnable recurrent process that progressively refines both geometry and appearance over multiple iterations. This design enables the network to correct its own errors and achieve higher fidelity without per-scene optimization. Specifically, to shatter the brittle dependency on a static depth scaffold, we introduce an uncertainty-driven depth refinement strategy. This mechanism dynamically adjusts the search space for each Gaussian based on its evolving uncertainty, facilitating robust and precise geometric localization. To further improve appearance details, we design a region-aware enhancement mechanism that applies targeted multi-view and monocular feature aggregation to resolve ambiguities in both overlapping and non-overlapping areas. These two components are orchestrated by a central gated recurrent unit (GRU)-based recurrent optimizer, which creates a virtuous cycle: improved geometry provides a more reliable scaffold for appearance refinement, while enhanced features, in turn, guide more precise geometric corrections in subsequent passes. Through this paradigm, iSplat effectively combines the adaptability of optimization-based refinement with the efficiency of feed-forward inference, enabling fine-grained and high-fidelity 3D Gaussian splatting.

Our main contributions can be summarized as follows:

- We propose iSplat, a novel iterative learning framework for feed-forward 3D Gaussian Splatting. It recasts one-shot prediction as a recurrent process that progressively

enhances both geometry and appearance, bridging the gap between efficient feed-forward models and high-fidelity optimization-based methods.

- We introduce an uncertainty-driven depth refinement strategy that dynamically adjusts the depth search space, enabling fine-grained and robust geometric localization for each Gaussian.
- We present a region-aware enhancement mechanism that leverages the iteratively improved depth to distinguish between overlapping and non-overlapping regions, applying targeted aggregation to improve both geometric and appearance parameters.
- iSplat achieves superior reconstruction quality and strong cross-dataset generalization with only 42.6M parameters, surpassing DepthSplat on RealEstate10K and delivering a 2.88 dB PSNR gain on DTU, demonstrating the efficacy of iterative learning over one-shot approaches.

2. Related Work

Depth Estimation. Depth estimation includes multi-view and monocular approaches. Multi-view methods [3, 7, 9, 10, 26, 35] exploit cross-view consistency for accurate geometry but struggle in occluded or textureless regions. Monocular methods [2, 22, 29, 34] leverage semantic priors and generalize well, yet suffer from scale ambiguity and view inconsistency. These complementary characteristics inspire our design to combine the structural precision of multi-view estimation with the semantic and uncertainty awareness of monocular cues, enabling more robust and fine-grained geometry learning for 3D Gaussian splatting.

Feed-Forward Gaussian Splatting. Such methods [4, 5, 15, 18, 33] reconstruct scenes by predicting a depth scaffold followed by direct Gaussian regression. However, their single-pass design propagates initial geometric errors without correction. Recent works introduce refinement, *e.g.*, HiSplat [24] (coarse-to-fine) and ReSplat [32] (iterative residual learning), but rely on explicit rendering errors, which incur high computational complexity. In contrast, iSplat adopts a lightweight GRU-based recurrent optimization framework, where uncertainty and hidden states guide iterative refinement of geometry and appearance, achieving a better balance between quality and efficiency.

Iterative Refinement. Iterative refinement improves predictions in tasks such as optical flow [12, 25], stereo [11], and depth estimation [1, 22]. These methods iteratively update predictions using shared weights and contextual feedback. Inspired by this paradigm, we extend iterative refinement to feed-forward 3D Gaussian splatting. iSplat employs a GRU-based optimizer to progressively refine both geometry and appearance, enabling self-correction and bridging the gap between efficient one-shot methods and optimization-based approaches.

3. Method

3.1. Iterative Learning Framework Overview

The fundamental limitation of existing feed-forward 3DGS methods, as illustrated in Fig. 1 (a), lies in their monolithic one-shot prediction pipeline. Given a set of source images $\{I_i\}_{i=1}^N$, these approaches first extract multi-view features using an encoder f_E : $F_i = f_E(I_i)$. Subsequently, a depth head f_D and a Gaussian head f_G operate in a strict cascade:

$$D_i = f_D(F_i), \quad G_i = f_G(F_i, D_i), \quad (1)$$

where D_i denotes the estimated depth and $G_i = \{\mu, \alpha, c, \Sigma\}_i$ represents the corresponding Gaussian parameters. This single-pass mapping from image features to Gaussians is efficient but non-corrective: any depth errors are irrevocably propagated to appearance attributes, leading to suboptimal geometry and rendering fidelity. Moreover, the absence of a refinement mechanism prevents the model from progressively reasoning about uncertainty or correcting earlier mistakes.

Iterative Learning for Fine-grained Reconstruction. To overcome these limitations, we propose **iSplat**, which reformulates 3D Gaussian estimation as a *learnable iterative learning process*, as shown in Fig. 1 (b). At the core of iSplat is a GRU-based recurrent optimizer that maintains a hidden state tensor H_k , capturing the accumulated geometric and photometric context over iterations. This recurrent design enables the model to iteratively refine both geometry and appearance, progressively improving reconstruction fidelity without per-scene optimization.

For a given view (omitting index i for clarity), the process unfolds over K iterations. Starting from an initial hidden state H_0 and gaussian features G_0 initialized from the image features F , the iterative learning is expressed as:

$$H_k = f_{\text{GRU}}(H_{k-1}, F, G_{k-1}), \quad (2)$$

$$D_k = f_D(H_k), \quad (3)$$

$$G_k = f_G(H_k, F, D_k). \quad (4)$$

Here, the updated hidden state H_k serves as a dynamic memory that preserves scene priors and accumulates correction signals across iterations. It plays a dual role: first, H_k guides the depth head f_D to produce a refined depth map D_k (Eq. 3); second, it informs the Gaussian head f_G to update appearance parameters conditioned on the latest geometric estimate (Eq. 4). This shared latent state establishes a tight coupling between geometry and appearance estimation, forming a synergistic refinement loop where improved depth enhances appearance prediction, and the refined features, in turn, guide subsequent depth corrections.

This iterative framework introduces two major advantages. (1) It brings *temporal consistency* into feed-forward reconstruction, each GRU update step functions

as a lightweight optimization iteration, progressively refining geometry and appearance. (2) The shared hidden state H_k acts as a latent bridge between f_D and f_G , ensuring bidirectional information flow and enabling the model to reason about geometric uncertainty and fine-grained appearance jointly. Through this design, iSplat combines the adaptability of optimization-based refinement with the efficiency of feed-forward inference, achieving fine-grained, high-fidelity 3D Gaussian Splatting.

The overall architecture of our proposed iSplat is illustrated in Fig. 2. To realize this effective iterative process, we propose several key components. In the following sections, we will first detail the feature extraction backbone (Sec.3.2), and then elaborate on the specific designs of the core components within the iterative loop: our uncertainty-driven depth refinement (Sec.3.3), the region-aware feature enhancement (Sec.3.4), and the recurrent update mechanism that predicts the final Gaussian attributes (Sec.3.5).

3.2. Feature Extraction

Our iterative framework builds on a rich collection of complementary image features that capture both multi-view geometric consistency and single-view structural priors. We thus employ a dual-branch feature encoder.

Multi-view Features. Following standard practices in multi-view stereo [35], we first extract per-view features from the N input images $\{I_i\}_{i=1}^N$ using a shared-weight CNN backbone [31] (f_S). These initial features are then processed by a multi-view Transformer [19, 31] to facilitate cross-view information exchange. This yields a set of geometrically-aware high-level feature maps, denoted as $\{F_{i,m}^h\}_{i=1}^N \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times C_1}$, which are crucial for establishing correspondence in overlapping regions. We also retain the initial low-level CNN features $\{F_{i,m}^l\}_{i=1}^N \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times C_2}$ for contextual information.

Monocular Features. While powerful, multi-view features degrade in texture-less or non-overlapping regions. To provide robust priors in such ambiguous areas, we leverage a pre-trained monocular depth network [34] (f_M) to extract high-level structural features $\{F_{i,s}^h\}_{i=1}^N$ and corresponding low-level features $\{F_{i,s}^l\}_{i=1}^N$ from each image. These features encode rich semantic and plausible geometric cues where multi-view consistency is unavailable. For the remainder of this section, we omit the view index i for clarity.

3.3. Uncertainty-Driven Depth Refinement

To break the brittle dependency on single-shot depth prediction, we introduce a progressive refinement strategy that iteratively localizes depth with dynamically adjusted search ranges. This process is driven by the hidden state H_{k-1} of our recurrent optimizer.

Initialization ($k = 1$). At the first iteration, we establish a coarse depth context. We discretize a global depth range

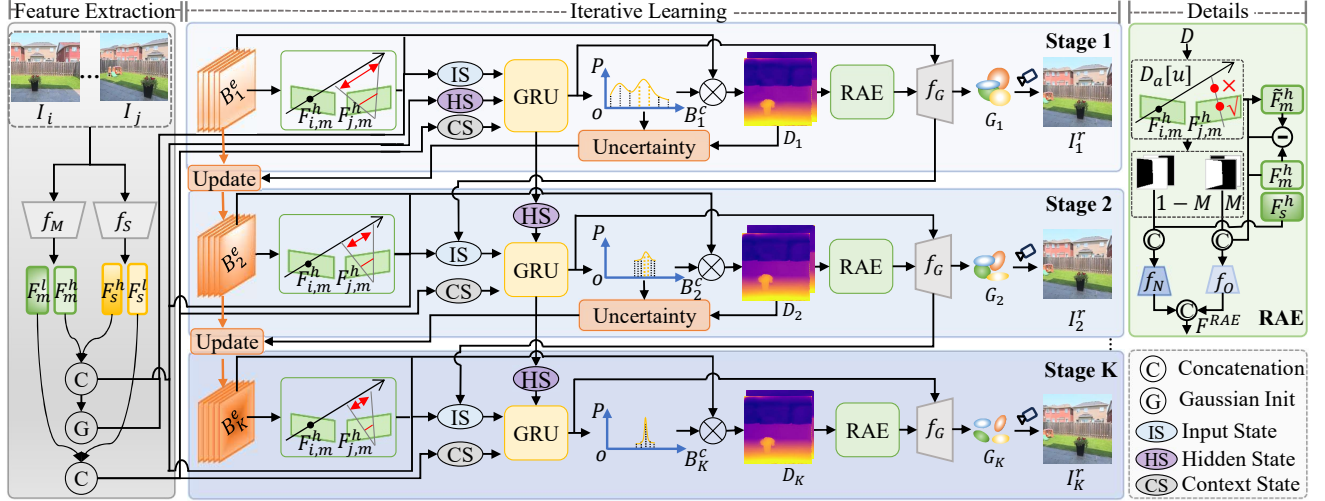


Figure 2. **Overview of the proposed iSplat framework.** It is composed of two primary components: a feature extraction module and an iterative learning module. (Left) The feature extraction module processes input images to extract a rich and complementary set of features, providing cues for both multi-view geometric consistency and single-view structural priors. (Center) Central to iSplat is the iterative learning module, which employs a recurrent GRU-based optimizer to refine both geometry and appearance in a synergistic loop. Each refinement stage consists of three key components: uncertainty-driven depth refinement, region-aware feature enhancement (RAE), and recurrent state update and prediction, which collectively allow the model to progressively improve the 3D reconstruction.

$[d_{\min}, d_{\max}]$ into B bins with edges:

$$B_{1,b}^e = d_{\min} + b \cdot \frac{d_{\max} - d_{\min}}{B}, \quad b = 0, \dots, B, \quad (5)$$

and define the bin centers:

$$B_{1,b}^c = \frac{1}{2}(B_{1,b}^e + B_{1,b+1}^e), \quad b = 0, \dots, B-1. \quad (6)$$

Given hidden state H_1 (Sec. 3.1), a lightweight prediction head predicts a probability distribution $P_1 \in \mathbb{R}^B$ over these bins. The initial depth D_1 is computed as the expected value (i.e., a linear combination) over these bin centers:

$$D_1(u) = \sum_{b=0}^{B-1} P_{1,b}(u) B_{1,b}^c, \quad (7)$$

where $u = (x, y)$ denotes the pixel coordinate.

Uncertainty Modeling. For subsequent iterations, a fixed search range around the previous estimate D_{k-1} is suboptimal, as it cannot recover from significant errors. We therefore propose an adaptive strategy where the search space is adjusted based on the prediction’s uncertainty. We quantify the geometric uncertainty $\mathcal{U}_{k-1}$ as the standard deviation of the previous probability distribution P_{k-1} :

$$\mathcal{U}_{k-1}(u) = \sqrt{\sum_{b=0}^{B-1} P_{k-1,b}(u) (B_{k-1,b}^c(u) - D_{k-1}(u))^2}, \quad (8)$$

where $B_{k-1,b}^c$ are the bin centers from the previous iteration. In our modeling, the uncertainty is used to control the width of the new search space.

Iterative Refinement ($k > 1$). Let b^* satisfy $B_{k-1,b^*}^e \leq D_{k-1} < B_{k-1,b^*+1}^e$. Instead of using this bin verbatim, we elastically expand it by the uncertainty. Therefore, we

define the new, refined bins $[d_{\min}^{(k)}, d_{\max}^{(k)}]$ for iteration k as:

$$[d_{\min}^{(k)}, d_{\max}^{(k)}] = [B_{k-1,b^*}^e - \varphi \cdot \mathcal{U}_{k-1}, B_{k-1,b^*+1}^e + \varphi \cdot \mathcal{U}_{k-1}], \quad (9)$$

where φ is a hyperparameter controlling the elasticity. We then re-discretize this interval into B bins to obtain $\{B_{k,b}^e, B_{k,b}^c\}$ and predict P_k with f_D guided by the current state H_k (Eq. 3). The refined depth D_k again follows Eq. (7). This uncertainty-driven approach allows the model to intelligently “zoom in” on confident regions while maintaining a wider search for uncertain ones, enabling robust geometric self-correction.

3.4. Region-Aware Feature Enhancement

To leverage our dual feature sources, we introduce a Region-Aware Enhancement (RAE) module that processes features differently based on their geometric reliability. At each iteration k , the RAE module uses the just-refined depth map D_k to create an explicit geometric prior by overlap.

Overlap Mask. Specifically, we compute a binary overlap mask $M_k \in \{0, 1\}^{H \times W}$. For each pixel u in reference view i , we project it into other source views j using $D_k(u)$ and camera intrinsics and extrinsics ($\mathbf{K}_i, \mathbf{T}_{i \rightarrow j}$):

$$u_j' \sim \mathbf{K}_j \mathbf{T}_{i \rightarrow j} \mathbf{K}_i^{-1} \tilde{u}_i D_k(u_i), \quad \tilde{u} = [x, y, 1]^\top, \quad (10)$$

and is valid if u_j' falls inside image bounds. Therefore, the binary mask is defined as:

$$M_k(u) = \max_{j \neq i} \mathbb{I}[\text{is_valid}(u_j')], \quad (11)$$

where $M_k = 1$ marks overlapping regions and 0 otherwise.

Region-specific Enhancement. Our enhancement strategy operates in a region-specific manner, adapting its fea-

ture processing based on the geometric reliability indicated by the overlap mask M_k . For reliable overlapping regions ($M_k = 1$), we first establish multi-view correspondence by warping features F_m^h from source views into the reference view to obtain $\tilde{F}_m^h$ (Eq.(10)). The resulting discrepancy, $F_m^h - \tilde{F}_m^h$, which encodes correspondence errors, is then processed by a dedicated submodule f_O alongside the original features:

$$F_O = f_O(F_m^h, \tilde{F}_m^h, F_m^h - \tilde{F}_m^h, M_k). \quad (12)$$

For non-overlapping regions ($M_k = 0$) where multi-view consistency is unavailable, a second submodule f_N is employed to leverage strong monocular priors from F_s^h :

$$F_N = f_N(F_s^h, 1 - M_k). \quad (13)$$

Finally, the outputs from both specialized pathways are concatenated to form the comprehensive region-aware representation F_k^{RAE} , which is then fed to the Gaussian head f_G :

$$F_k^{\text{RAE}} = \text{Concat}(F_O, F_N). \quad (14)$$

This design allocates capacity adaptively: enforcing multi-view consistency where available and falling back to learned single-view priors otherwise.

3.5. Recurrent State Update and Prediction

The core of our framework is the recurrent optimizer that integrates all information to update its belief about the scene, represented by the hidden state H_k . This update is performed by a Convolutional GRU (ConvGRU) [6].

At each iteration k , the ConvGRU takes the previous hidden state H_{k-1} and a context-motion input X_k to produce the new hidden state H_k . The input X_k is a concatenation of static context features (derived from F_m^l and F_s^l) and dynamic features derived from the current iteration's computation, including the cost volume from the depth estimation, the region-aware features F_k^{RAE} , and gaussian features G_{k-1} . The GRU update proceeds as:

$$Z_k = \sigma(\text{Conv}([H_{k-1}, X_k])) \quad (15)$$

$$R_k = \sigma(\text{Conv}([H_{k-1}, X_k])) \quad (16)$$

$$\tilde{H}_k = \tanh(\text{Conv}([R_k \odot H_{k-1}, X_k])) \quad (17)$$

$$H_k = (1 - Z_k) \odot H_{k-1} + Z_k \odot \tilde{H}_k, \quad (18)$$

where σ is the sigmoid function and $\odot$ is element-wise multiplication. The initializer H_0 is a learned projection of $\text{Concat}(F_m^h, F_s^h)$. The updated hidden state H_k now encapsulates the most comprehensive understanding of the scene at this iteration. It is then used by two lightweight prediction heads.

Depth Distribution Prediction. A small convolutional head takes H_k as input to predict the probability distribution P_{k+1} over the depth candidates for the *next* iteration. This prediction drives the uncertainty-driven refinement described in Sec. 3.3.

Gaussian Attribute Prediction. A second head uses the hidden state H_k and the region-aware features F_k^{RAE} to pre-

dict the remaining Gaussian attributes. The 3D position μ is determined by unprojecting the most accurate depth map D_k . An attribute head then regresses the opacity α , covariance matrix Σ , and color c from the concatenated features $[H_k, F_k^{\text{RAE}}, D_k]$. This ensures the final appearance is conditioned on the fully refined geometric and feature context.

3.6. Training Loss

Our framework is trained end-to-end using a rendering loss that jointly supervises both the geometric positions (via depth) and the appearance attributes (opacity, covariance, color) of the 3D Gaussians. Since our progressive learning paradigm produces a set of Gaussian primitives at each stage k , we apply supervision across all K stages to ensure stable and effective convergence. The total training loss is a weighted sum of rendering losses across all iterations:

$$L_{\text{total}} = L_1^{\text{render}} + \sum_{k=2}^K \gamma^{K-k} \cdot L_k^{\text{render}}, \quad (19)$$

where $\gamma < 1$ is an exponential decay factor, giving higher weight to later, more refined predictions. The per-iteration rendering loss L_k^{render} is a combination of an L1 loss and the Learned Perceptual Image Patch Similarity (LPIPS) [37] between the rendered image I_k^{render} and the ground truth I^{gt} :

$$L_k^{\text{render}} = \|I_k^{\text{render}} - I^{\text{gt}}\|_1 + \lambda \cdot \text{LPIPS}(I_k^{\text{render}}, I^{\text{gt}}), \quad (20)$$

where λ balances the two terms. This multi-stage supervision strategy is crucial for guiding the synergistic refinement of both geometry and appearance throughout the iterative learning process.

4. Experiments

4.1. Experimental Setup

Datasets. We follow recent Gaussian splatting works [5, 18] and use three standard benchmarks. Our model is trained on RealEstate10K [38] (67,477 training / 7,289 test indoor sequences) and ACID [17] (11,075 training / 1,972 test aerial sequences), both providing accurate camera poses and diverse scene variation. For cross-domain evaluation, we perform zero-shot testing on the DTU dataset [13], reporting results on 16 validation scenes with four novel viewpoints each to assess generalization across domains.

Metrics. For quantitative evaluation, we adopt standard image-quality metrics, including pixel-level PSNR, patch-level SSIM [27], and feature-level LPIPS [37]. We additionally report model parameter counts to assess computational efficiency. Following prior work [5, 33], all experiments are conducted at a resolution of 256×256 for fair comparison.

Implementation Details. Our iSplat framework is implemented in PyTorch, with rendering performed using an off-the-shelf CUDA implementation of 3D Gaussian Splatting. We adopt the simplest configuration of DepthSplat [33] using single-scale depth and a ViT-Small backbone for

Table 1. **Quantitative comparisons on the RealEstate10K [38] and ACID [17] datasets.** We report PSNR, SSIM, and LPIPS for reconstruction quality, along with model parameters for efficiency. DepthSplat_S², DepthSplat_L², and DepthSplat_S¹ denote three variants of DepthSplat, where ^s indicates the depth scale and the subscript *S/L* refers to the size of the monocular backbone [34]. **The simplest variant is used as our baseline.** “-” denotes that the corresponding result is not reported in the original paper. The best results are in bold.

Method	RealEstate10K [38]			ACID [17]			Params (M)↓
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	
Du <i>et al.</i> [8]	24.78	0.820	0.213	26.88	0.799	0.218	125.1
GPNR [23]	24.11	0.793	0.255	25.28	0.764	0.332	9.6
pixelNeRF [36]	20.43	0.589	0.550	20.97	0.547	0.533	28.2
MuRF [30]	26.10	0.858	0.143	28.09	0.841	0.155	5.3
pixelSplat [4]	26.09	0.863	0.136	28.27	0.843	0.146	125.4
latentSplat [28]	23.07	0.825	0.182	24.95	0.782	0.207	187.0
MVSplat [5]	26.39	0.869	0.128	28.25	0.843	0.144	12.0
eFreeSplat [21]	26.45	0.865	0.126	28.30	0.851	0.140	-
TranSplat [15]	26.69	0.875	0.125	28.35	0.845	0.143	-
MonoSplat [18]	26.68	0.875	0.123	28.63	0.864	0.138	30.3
DepthSplat _S ² [33]	27.12	0.884	0.119	-	-	-	41.0
DepthSplat _L ² [33]	27.47	0.889	0.114	-	-	-	354.0
Base(DepthSplat _S ¹ [33])	26.84	0.878	0.122	-	-	-	38.0
iSplat (Ours)	27.67	0.891	0.110	29.01	0.875	0.126	42.6

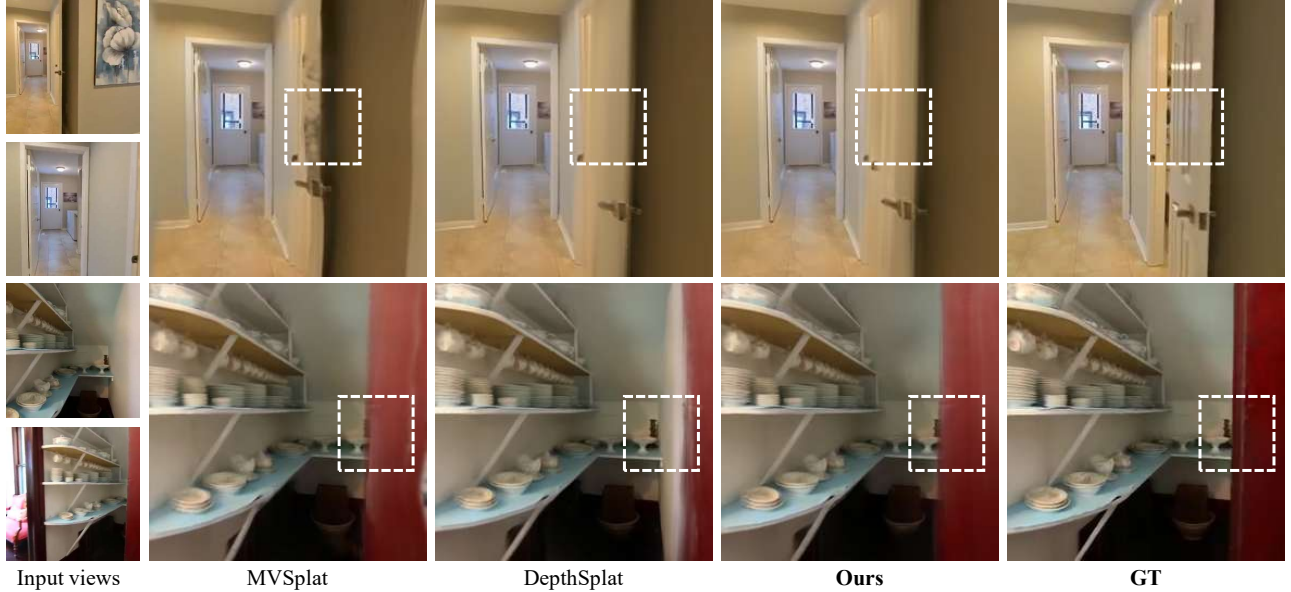


Figure 3. **The qualitative comparisons of novel view synthesis on RealEstate10K [38].** Models are trained with a collection of scenes from RealEstate10K, and tested on novel scenes from the test splits. Our iSplat yields more accurate geometry and finer appearance through iterative learning in challenging regions.

monocular feature extraction as our baseline. All models are trained for 300,000 iterations with a total batch size of 16 on four NVIDIA H800 GPUs using the Adam optimizer [16]. Inference speed and memory usage are measured on a single RTX 3090 GPU. The hyperparameters φ and γ are set to 0.5 and 0.85, respectively.

Comparison Methods. We compare our approach against

a broad spectrum of leading methods for scene-level novel view synthesis, covering three major categories. (i) *Light Field Network* models, including Du *et al.* [8] and GPNR [23]. (ii) *NeRF-based* generalizable radiance field approaches, represented by pixelNeRF [36] and MuRF [23]. (iii) *Feed-forward 3D Gaussian Splatting* methods, which constitute the current state of the art, including spixel-

Table 2. **Quantative comparisons of cross-dataset generalization.** We perform zero-shot tests on DTU [13] and ACID [17] datasets, using models trained solely on RealEstate10K [38].

Method	Re10k $\rightarrow$ DTU			Re10k $\rightarrow$ ACID		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
pixelSplat [4]	12.89	0.382	0.560	27.64	0.830	0.160
MVSplat [5]	13.94	0.473	0.385	28.15	0.841	0.147
TranSplat [15]	14.93	0.531	0.326	28.17	0.842	0.146
MonoSplat [18]	15.25	0.605	0.291	28.24	0.848	0.145
DepthSplat [33]	15.38	0.415	0.442	28.37	0.847	0.141
iSplat (Ours)	18.26	0.698	0.256	28.65	0.852	0.137

Splat [4] and latentSplat [28], MVSplat [5], eFreeSplat [21], TranSplat [15], MonoSplat [18], and DepthSplat [33].

4.2. Main Results

Comparisons on RealEstate10K and ACID. We present extensive quantitative comparisons of iSplat against state-of-the-art feed-forward methods on the RealEstate10K and ACID datasets in Table 1. Our method demonstrates superior performance across all standard evaluation metrics. By introducing iterative learning into the feed-forward paradigm, iSplat effectively mitigates the error accumulation characteristic of single-pass methods, yielding substantial gains in reconstruction fidelity. Compared to DepthSplat_L² (354.0M), our model achieves +0.2 dB PSNR on RealEstate10K with only 42.6M parameters ($\approx 1/8$ size). Moreover, adding only 4.6M parameters to DepthSplat_S¹ yields a +0.83 dB (3.09%) PSNR gain, demonstrating the efficiency of our recurrent optimizer. Fig. 3 further demonstrates that iSplat delivers noticeably sharper geometry and more coherent appearance compared to existing feed-forward methods. The highlighted areas show that iSplat recovers high-frequency structures with significantly improved accuracy, producing renderings that closely match ground-truth views. The compared methods produce blurred textures, misaligned structures, or shading inconsistencies in these challenging regions, our iterative learning strategy progressively corrects geometry errors and enhances view-consistent details.

Cross-Dataset Generalization. We present a zero-shot generalization analysis in Table 2, training on RealEstate10K and evaluating on two distinct domains. On the challenging Re10k $\rightarrow$ DTU task, which evaluates generalization from indoor scenes to object-centric indoor scans, iSplat achieves 18.26 PSNR, 0.698 SSIM, and 0.256 LPIPS, consistently outperforming existing methods. These results highlight the strength of our iterative learning mechanism in learning domain-robust geometric and appearance priors that remain stable under significant distribution shifts. On Re10k $\rightarrow$ ACID task, which evaluates gener-

Table 3. **Runtime comparison on the RealEstate10K dataset.**

Method	Iters	Params (M)	Mem. (MB) $\downarrow$	Time (s) $\downarrow$	PSNR $\uparrow$
MVSplat	0	12.0	1719	0.056	26.39
DepthSplat	0	354.0	3257	0.098	27.47
iSplat	1	42.6	1823	0.059	26.95
	2	42.6	2080	0.097	27.60
	3	42.6	2337	0.124	27.67

Table 4. **The ablation study on RealEstate10K [38].** UDR: Uncertainty-driven depth refinement. RAE: Region-Aware feature enhancement. IL: Iterative Learning.

Method	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
iSplat	27.67	0.891	0.110
w/o UDR	27.10	0.883	0.120
w/o RAE	27.42	0.886	0.115
w/o IL	26.95	0.880	0.119



Figure 4. **Qualitative comparisons on RealEstate10k [38] with different design choices.**

alization to aerial outdoor scenes, iSplat again attains the best performance across all metrics with 28.65 PSNR, 0.852 SSIM, and 0.137 LPIPS. The consistent improvements on both near-domain (ACID) and far-domain (DTU) transfers demonstrate that iSplat’s iterative learning paradigm enable strong resilience to domain changes, yielding superior cross-dataset generalization.

Runtime. Fig. 3 compares the runtime-quality trade-off of different methods. iSplat achieves higher reconstruction quality than MVSplat and DepthSplat at comparable runtimes, using 1 and 2 iterations, respectively. All runtimes are measured on a NVIDIA RTX 3090 GPU with input resolution of 256×256 and batch size 1. Increasing the number of iterations further improves performance with only moderate additional cost, demonstrating a favorable balance between efficiency and reconstruction quality.

4.3. Ablation Study

We conduct ablation experiments on RealEstate10K [38] to examine the contribution of each core component in iSplat. The quantitative results are shown in Table 4 and the qualitative results are shown in Fig. 4.

Effect of Uncertainty-Driven Depth Refinement (UDR). we first investigate the significance of our adaptive geometric refinement strategy. In the “w/o UDR” variant, we re-

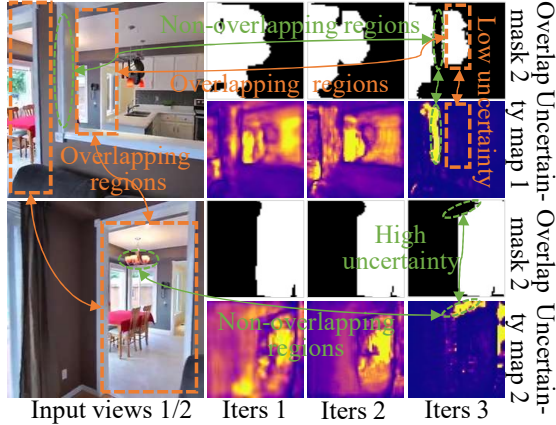


Figure 5. Uncertainty maps and region-aware representations.

placed the dynamic, uncertainty-guided search space adjustment with a static refinement range (by setting a fixed uncertainty value of 2). This modification led to a marked decline in reconstruction quality, confirming that a naive, uniform refinement strategy is suboptimal. By dynamically adjusting the search bins based on the previous step’s uncertainty, our UDR module intelligently allocates the model’s refinement capacity. It focuses on regions with high uncertainty where significant geometric corrections are needed, while maintaining precision in already confident areas, as shown in Fig. 5. This targeted optimization is critical for efficiently resolving geometric inaccuracies, resulting in a more coherent and accurate 3D representation.

Effect of Region-Aware Feature Enhancement (RAE). Next, we analyze the role of the RAE module in enhancing appearance details. The performance drop in the “w/o RAE” ablation underscores the importance of adaptively fusing complementary feature sources. Multi-view features provide powerful correspondence cues in well-constrained, overlapping regions, but their reliability degrades significantly in texture-less or non-overlapping areas. Conversely, single-view features encode rich semantic and contextual priors that offer plausible details where multi-view consistency is weak or unavailable. The RAE module acts as an intelligent arbitrator, using the refined depth map as a geometric prior to selectively emphasize the most reliable feature source for each region, as shown in Fig. 5. Without it, the model is forced into a naive fusion, leading to blurred textures and inconsistent details. The RAE is thus essential for rendering photorealistic appearances.

Effect of Iterative Learning (IL). Finally, We investigate the cornerstone of our approach: the iterative learning process. By removing the iterative loop (“w/o IL”), our model degenerates into a single-pass architecture. This configuration yielded the most substantial performance degradation across all metrics. As visualized in Fig. 4, this quantitative drop translates directly to severe visual artifacts, including

Table 5. Ablation experiment for the iterations on the RealEstate10K. $a \rightarrow b$ represents a : #iters at training and b : #iters at test time.

#Iters	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
1 $\rightarrow$ 1	26.95	0.880	0.119
2 $\rightarrow$ 1	27.53	0.886	0.114
2 $\rightarrow$ 2	27.60	0.890	0.111
3 $\rightarrow$ 1	27.58	0.889	0.113
3 $\rightarrow$ 2	27.61	0.890	0.111
3 $\rightarrow$ 3	27.67	0.891	0.110
4 $\rightarrow$ 4	27.65	0.891	0.110

a blurry appearance and inconsistent geometry. This result provides strong evidence for our central thesis that a single feed-forward pass is inherently insufficient for resolving the complex geometric and photometric ambiguities present in sparse-view scene reconstruction. The iterative framework, guided by the GRU-based optimizer, creates a crucial feedback loop. It enables the model to progressively identify and correct its own estimation errors from previous passes, thereby achieving a level of fidelity that one-shot approaches struggle to attain.

Analysis of Iteration Count. We further analyze the impact of the number of iterative steps, with results presented in Table 5. Reconstruction quality progressively improves as the iteration count increases from one to three, validating our hypothesis that each step corrects errors from the previous one. However, performance gains saturate beyond this point. We thus adopt three iterations as our default, striking an optimal balance between fidelity and efficiency. Results also show that performance is maximized when training and testing iterations are identical ($a \rightarrow a$), as a mismatch disrupts the optimizer’s learned refinement trajectory.

5. Conclusion

We present iSplat, an iterative feed-forward framework that addresses the fidelity limitations of single-pass 3D Gaussian Splatting. By reformulating reconstruction as a learnable recurrent process, iSplat progressively refines both geometry and appearance through a GRU-based optimizer, uncertainty-driven depth refinement, and region-aware enhancement. Experiments on multiple benchmarks demonstrate that iSplat achieves state-of-the-art performance with strong efficiency and generalization. Our results show that iterative feed-forward refinement is an effective way to combine the speed of direct inference with the precision of optimization-based methods.

Acknowledgements

This work is supported by the New Generation Artificial Intelligence-National Science and Technology Major Project (No. 2025ZD0123002), the National Natural Science Foundation of China (No. 62476051 and No. 82121003), the Sichuan Natural Science Foundation (No. 2024NSFTD0041), and the Science, Technology and Innovation Project of Shenzhen Longhua District (No. 20260309G23410662).

References

- [1] Antyanta Bangunharcana, Ahmed Magd, and Kyung-Soo Kim. Dualrefine: Self-supervised depth and pose estimation through iterative epipolar sampling and refinement toward equilibrium. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 726–738, 2023. 2
- [2] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4009–4018, 2021. 2
- [3] 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. 2
- [4] 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*, pages 19457–19467, 2024. 1, 2, 6, 7
- [5] Yuedong Chen, Haoifei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pages 370–386. Springer, 2024. 1, 2, 5, 6, 7
- [6] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 5
- [7] Yikang Ding, Wentao Yuan, Qingtian Zhu, Haotian Zhang, Xiangyue Liu, Yuanjiang Wang, and Xiao Liu. Transmvsnet: Global context-aware multi-view stereo network with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8585–8594, 2022. 2
- [8] 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*, pages 4970–4980, 2023. 6
- [9] Arda Duzceker, Silvano Galliani, Christoph Vogel, Pablo Speciale, Mihai Dusmanu, and Marc Pollefeys. Deepvideomvs: Multi-view stereo on video with recurrent spatiotemporal fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15324–15333, 2021. 2
- [10] 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*, pages 2495–2504, 2020. 2
- [11] Xiaodong Gu, Weihao Yuan, Zuozhuo Dai, Siyu Zhu, Chengzhou Tang, Zilong Dong, and Ping Tan. Dro: Deep recurrent optimizer for video to depth. *IEEE Robotics and Automation Letters*, 8(5):2844–2851, 2023. 2
- [12] Junhwa Hur and Stefan Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5754–5763, 2019. 2
- [13] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanaes. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413, 2014. 5, 7
- [14] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 1, 2
- [15] Jeongyun Kim, Jeongho Noh, Dong-Guw Lee, and Ayoung Kim. Transplat: Surface embedding-guided 3d gaussian splatting for transparent object manipulation. *arXiv preprint arXiv:2502.07840*, 2025. 1, 2, 6, 7
- [16] Diederik Kinga, Jimmy Ba Adam, et al. A method for stochastic optimization. In *International conference on learning representations (ICLR)*. California, 2015. 6
- [17] 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*, pages 14458–14467, 2021. 5, 6, 7
- [18] Yifan Liu, Keyu Fan, Weihao Yu, Chenxin Li, Hao Lu, and Yixuan Yuan. Monosplat: Generalizable 3d gaussian splatting from monocular depth foundation models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21570–21579, 2025. 1, 2, 5, 6, 7
- [19] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 3
- [20] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1
- [21] Zhiyuan Min, Yawei Luo, Jianwen Sun, and Yi Yang. Epipolar-free 3d gaussian splatting for generalizable novel view synthesis. *Advances in Neural Information Processing Systems*, 37:39573–39596, 2024. 6, 7
- [22] Shuwei Shao, Zhongcai Pei, Xingming Wu, Zhong Liu, Weihai Chen, and Zhengguo Li. Iebins: Iterative elastic bins for

- monocular depth estimation. *Advances in Neural Information Processing Systems*, 36:53025–53037, 2023. 2
- [23] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *European Conference on Computer Vision*, pages 156–174. Springer, 2022. 6
- [24] Shengji Tang, Weicai Ye, Peng Ye, Weihao Lin, Yang Zhou, Tao Chen, and Wanli Ouyang. Hisplat: Hierarchical 3d gaussian splatting for generalizable sparse-view reconstruction. *arXiv preprint arXiv:2410.06245*, 2024. 2
- [25] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 2
- [26] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, and Marc Pollefeys. Itermvs: Iterative probability estimation for efficient multi-view stereo. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8606–8615, 2022. 2
- [27] 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. 5
- [28] Christopher Wewer, Kevin Raj, Eddy Ilg, Bernt Schiele, and Jan Eric Lenssen. latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. In *European conference on computer vision*, pages 456–473. Springer, 2024. 6, 7
- [29] Haifeng Wu, Shuhang Gu, Lixin Duan, and Wen Li. Geodepth: From point-to-depth to plane-to-depth modeling for self-supervised monocular depth estimation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11525–11535, 2025. 2
- [30] Han Xu, Jiteng Yuan, and Jiayi Ma. Murf: Mutually reinforcing multi-modal image registration and fusion. *IEEE transactions on pattern analysis and machine intelligence*, 45(10):12148–12166, 2023. 6
- [31] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezaatofghi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13941–13958, 2023. 3
- [32] Haofei Xu, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Resplat: Learning recurrent gaussian splats. *arXiv preprint arXiv:2510.08575*, 2025. 2
- [33] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16453–16463, 2025. 1, 2, 5, 6, 7
- [34] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiao-gang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024. 2, 3, 6
- [35] 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)*, pages 767–783, 2018. 2, 3
- [36] 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*, pages 4578–4587, 2021. 6
- [37] 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*, pages 586–595, 2018. 5
- [38] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 5, 6, 7