

# SelfSplat: Pose-Free and 3D Prior-Free Generalizable 3D Gaussian Splatting

Gyeongjin Kang<sup>1\*</sup> Jisang Yoo<sup>1\*</sup> Jihyeon Park<sup>1</sup> Seungtae Nam<sup>2</sup> Hyeonsoo Im<sup>3</sup>  
 Sangheon Shin<sup>3</sup> Sangpil Kim<sup>4</sup> Eunbyung Park<sup>2†</sup>

Sungkyunkwan University<sup>1</sup> Yonsei University<sup>2</sup> Hanhwa Systems<sup>3</sup> Korea University<sup>4</sup>

<https://gynjn.github.io/selfsplat/>

## Abstract

We propose SelfSplat, a novel 3D Gaussian Splatting model designed to perform pose-free and 3D prior-free generalizable 3D reconstruction from unposed multi-view images. These settings are inherently ill-posed due to the lack of ground-truth data, learned geometric information, and the need to achieve accurate 3D reconstruction without finetuning, making it difficult for conventional methods to achieve high-quality results. Our model addresses these challenges by effectively integrating explicit 3D representations with self-supervised depth and pose estimation techniques, resulting in reciprocal improvements in both pose accuracy and 3D reconstruction quality. Furthermore, we incorporate a matching-aware pose estimation network and a depth refinement module to enhance geometry consistency across views, ensuring more accurate and stable 3D reconstructions. To present the performance of our method, we evaluated it on large-scale real-world datasets, including RealEstate10K, ACID, and DL3DV. SelfSplat achieves superior results over previous state-of-the-art methods in both appearance and geometry quality, also demonstrates strong cross-dataset generalization capabilities. Extensive ablation studies and analysis also validate the effectiveness of our proposed methods.

## 1. Introduction

The recent introduction of Neural Radiance Fields (NeRF) [39] and 3D Gaussian Splatting (3D-GS) [27] had marked a significant advancement in computer vision and graphics, particularly in 3D reconstruction and novel view synthesis. By training on images taken from various viewpoints, these methods can produce geometrically consistent photo-realistic images, providing beneficial for various ap-

plications, such as virtual reality [25, 63], robotics [16, 41], and semantic understanding [72, 73]. Despite their impressive capability in 3D scene representation, training NeRF and 3D-GS requires a large set of accurately posed images as well as iterative per-scene optimization procedures, which limits their applicability for broader use cases.

To bypass the iterative optimization steps, various learning-based generalizable 3D reconstruction models [3, 14, 50, 60, 69] have been proposed. These models can predict 3D geometry and appearance from a few posed images in a single forward pass. Leveraging large-scale synthetic and real-world 3D datasets, they used pixel-aligned features to extract scene priors from input images and generate novel views through differentiable rendering methods such as volume rendering [38] or rasterization [30]. The generated images are then supervised with ground truth images captured from the same camera poses. While this approach enables 3D scene reconstruction without iterative optimization steps, a key limitation remains as follows: it relies on calibrated images (with accurate camera poses) for both training and inference, thereby constraining its use with less controlled, “in-the-wild” images or videos.

Recent efforts have integrated camera pose estimation with 3D scene reconstruction, combining multiple tasks within a single framework. By relaxing the constraint of a posed multi-view setup, pose-free generalizable methods [6, 22, 32, 46] aim to learn reliable 3D geometry from uncalibrated images and generate accurate 3D representations in a single forward pass. While these approaches have demonstrated promising results, they still face significant challenges. For example, [46] relies on error-prone pre-trained flow model for pose estimation, often leading to inaccuracies and performance degradation. [6, 32] achieve impressive results but require a per-scene fine-tuning stage, making them computationally expensive for real-world applications. Furthermore, both [46] and [6] inherit the limitation of NeRF-based approaches, demanding substantial computational costs due to the volumetric rendering.

\*Equal contribution

†Corresponding author

In this work, we present SelfSplat, a novel training framework for pose-free, generalizable 3D representations from monocular videos without pretrained 3D prior models or further scene-specific optimizations. We build upon the 3D-GS representation and leverage the pixel-aligned Gaussian estimation pipeline [3, 50], which has demonstrated fast and high-quality reconstruction results. By integrating 3D-GS representations with self-supervised depth and pose estimation techniques, the proposed method jointly predicts depth, camera poses, and 3D Gaussian attributes within a unified neural network architecture.

3D-GS, as an explicit 3D representation, is highly sensitive to minor errors in 3D positioning. Even slight misplacements of Gaussians can disrupt multi-view consistency, significantly degrading rendering quality [3, 7]. This makes the simultaneous prediction of Gaussian attributes and camera poses especially challenging. The proposed approach, SelfSplat, mitigates this issue by leveraging the strengths of both self-supervised learning and 3D-GS. Exploiting the geometric consistency inherent in self-supervised learning techniques effectively guides the positioning of 3D Gaussians, leading to improved reconstruction accuracy in the absence of camera pose information. Also, harnessing 3D-GS representation and its superior view synthesis capabilities help enhance the accuracy of camera pose estimation, which would otherwise depend solely on 2D image features derived from CNNs [18, 48] or Transformers [9, 42].

While the proposed method is encouraging, simply combining self-supervised learning with explicit 3D geometric supervision has yielded suboptimal results, particularly in predicting accurate camera poses and generating multi-view consistent depth maps. This often results in misaligned 3D Gaussians and inferior 3D structure reconstructions. To address issues from pose estimation errors, we introduce a matching-aware pose network that incorporates additional cross-view knowledge to improve geometric accuracy. By leveraging contextual information from multiple views, this network improves pose accuracy and ensures more reliable estimates across views. Additionally, to support consistent depth estimation, crucial for accurate 3D scene geometry, we develop a depth refinement network. This module uses estimated poses as embedding features which contains spatial information from surrounding views, to achieve accurate and consistent 3D geometry representations.

Once trained in a self-supervised manner, SelfSplat is equipped to perform several downstream tasks, including (1) pose, depth estimation, and (2) 3D reconstruction, including fast novel view synthesis. We demonstrate the efficacy of our method on RealEstate10k [75], ACID [36], and DL3DV [35] datasets providing higher appearance and geometry quality as well as better cross-dataset generalization performance. Extensive ablation studies and analyses also show the effectiveness of our proposed method. The main

contributions can be summarized as follows:

- We propose SelfSplat, a pose-free and 3D prior-free self-supervised learner from large-scale monocular videos.
- We propose to unify self-supervised learning with 3D-GS representation, harnessing the synergy of both frameworks to achieve robust 3D geometry estimation.
- To address pose estimation errors and inconsistent depth predictions, we introduce the matching-aware pose network and depth refinement module, which enhance geometry consistency across views, ensuring more accurate and stable 3D reconstructions.
- We have conducted comprehensive experiments and ablation studies on diverse datasets, and the proposed SelfSplat significantly outperforms the previous methods.

## 2. Related work

### 2.1. Pose-free Neural 3D Representations

In the absence of camera pose information, recent efforts have aimed to jointly optimize camera poses and 3D scenes. Starting with optimization-based methods, BARF [33] and subsequent research [2, 17, 26] addressed this challenge by training poses along with implicit or explicit scene representations. Also, in a generalizable setting with NeRF representations, FlowCam [46] utilized pretrained flow estimation model, RAFT [51], and find the rigid-body motion between 3D point clouds using the Procrustes algorithm [11]. DBARF [6] extended the previous optimization-based method [33] and utilized recurrent GRU [10] network for pose and depth estimation. Based on 3D-GS, several methods [23, 32, 66] showed pose-free generalizable method by employing explicit 3D representation. However, these methods face practical limitations due to their reliance on pretrained models [6, 32, 46, 66], the need for additional fine-tuning stages [6, 32], and the computationally intensive volume rendering process [6, 46], all of which hinder their scalability and efficiency in real-world applications. Also, CoPoNeRF [22] provides poses and radiance fields estimation at the inference stage, it still requires ground-truth pose supervision during training. In contrast, our method can reconstruct 3D scenes and synthesize novel views from unposed images, mitigating the preceding challenges, and offering a more scalable and efficient solution.

### 2.2. Self-supervised Learning for 3D Vision

Masked Autoencoder (MAE) [20, 53] is one of self-supervised representation learning framework on video datasets, leveraging their consistency in space and over time. The main objective of MAE is to reconstruct masked patch of pixels or latent features, thereby learning spatiotemporal continuity without any 3D inductive bias. Recently, CroCo [58, 59], a cross-view completion method which extends previous single-view approaches,

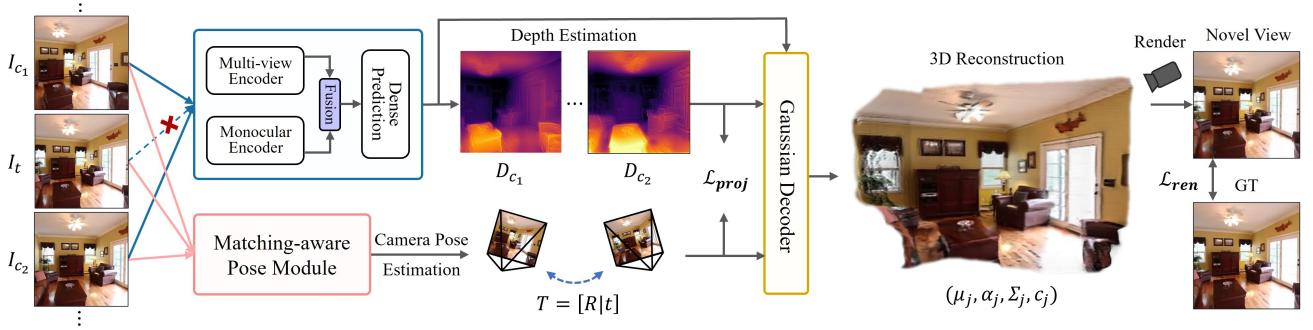


Figure 1. Overview of SelfSplat. Given unposed multi-view images as input, we predict depth and Gaussian attributes from the images, as well as the relative camera poses between them. We unify a self-supervised depth estimation framework with explicit 3D representation achieving accurate scene reconstruction.

has demonstrated a pretraining objective well-suited for geometric downstream tasks, such as optical flow and stereo matching. Expanding on this, DUSt3R [56] and MAS3R [31] introduce a novel paradigm for dense 3D reconstruction from multi-view image collections.

Another area of self-supervised learning for 3D vision is monocular depth estimation. Without ground-truth depth and camera pose annotations, they utilized the information from consecutive temporal frames using warped image reconstruction as a signal to train their networks. Starting with [74], which first introduced the method, and subsequent works [1, 8, 18] have developed upon this field. In this paper, we also follow the framework of self-supervised depth estimation, but different from previous methods, we combine 3D representation learning, which improves the depth estimation and enables novel view synthesis with resulting 3D scene representations.

### 3. Preliminary

#### 3.1. Self-supervised Depth and Pose Estimation

The self-supervised depth and pose estimation method is a geometric representation learning method from videos or unposed images, which does not require ground-truth depth and pose annotations [68, 74]. Typically, two separate networks are employed for each depth and pose estimation, though these networks may share common representations. Given a triplet of consecutive frames  $I_{c_1}, I_t, I_{c_2} \in \mathbb{R}^{H \times W \times 3}$ , the pose network predicts the relative camera pose between two frames and the depth network produces the depth maps for each frame. While there exist many variants, a typical loss function,  $\mathcal{L}_{proj}$ , to train two networks is

$$\mathcal{L}_{proj} = \text{pe}(I_t, I_{c_1 \rightarrow t}) + \text{pe}(I_t, I_{c_2 \rightarrow t}), \quad (1)$$

$$\text{pe}(I_a, I_b) = \frac{\omega}{2} (1 - \text{SSIM}(I_a, I_b)) + (1 - \omega) \|I_a - I_b\|_1, \quad (2)$$

where  $I_{c_1 \rightarrow t} \in \mathbb{R}^{H \times W \times 3}$  denotes the projected image from  $I_{c_1}$  onto  $I_t$  using the predicted camera pose and the depth

map.  $\text{pe}(\cdot, \cdot)$  is a photometric reconstruction error, usually calculated using a combination of L1 and SSIM [57] losses, and  $\omega$  is a hyperparameter that controls the weighting factor between them [18].

#### 3.2. Feed-forward 3D Gaussian Splatting

Feed-forward 3D Gaussian Splatting methods infer 3D scene structure from input images through a single network evaluation, predicting Gaussian attributes based on pixel-, feature-, or voxel-level tensors. Each Gaussian, defined as  $g_j = (\mu_j, \alpha_j, \Sigma_j, c_j)$ , includes attributes such as a mean  $\mu_j$ , a covariance  $\Sigma_j$ , an opacity  $\alpha_j$ , and spherical harmonics (sh) coefficients  $c_j$ . In particular, our framework adopts a pixel-aligned approach, predicting per-pixel Gaussian primitives along with accurate depth estimations, achieving high-quality 3D reconstruction and fast novel view synthesis. Given multiple input views, the model generates pixel-aligned Gaussians for each image, and combine them to represent the full 3D scene [3, 50].

### 4. Methods

#### 4.1. Self-supervised Novel View Synthesis

We begin with a triplet of unposed images,  $I_{c_1}, I_t, I_{c_2} \in \mathbb{R}^{H \times W \times 3}$ , which are taken from different viewpoints. Building on the recent pixel-aligned 3D Gaussian Splatting methods, our goal is to predict dense per-pixel Gaussian parameters from input view images,

$$\mathcal{G}_{c_1}, \mathcal{G}_{c_2} = f_\theta(I_{c_1}, I_t, I_{c_2}), \quad (3)$$

where  $f_\theta$  is a feed-forward network with learnable parameters  $\theta$ , and  $\mathcal{G}_{c_1} = \{(\mu_j, \alpha_j, \Sigma_j, c_j)\}_{j=1}^{HW}$  is a generated Gaussians for the input image  $I_{c_1}$ . Note that we only generate pixel-aligned Gaussians for two input views  $I_{c_1}$  and  $I_{c_2}$  while excluding the target view  $I_t$ . This design encourages the network to generalize to novel views  $I_t$  during training. In addition, we train a pose network  $f_\phi$  to estimate a relative transformation between two images,  $T_{c_1 \rightarrow c_2} = f_\phi(I_{c_1}, I_{c_2})$ , where  $T_{c_1 \rightarrow c_2} \in SE(3)$  consists of

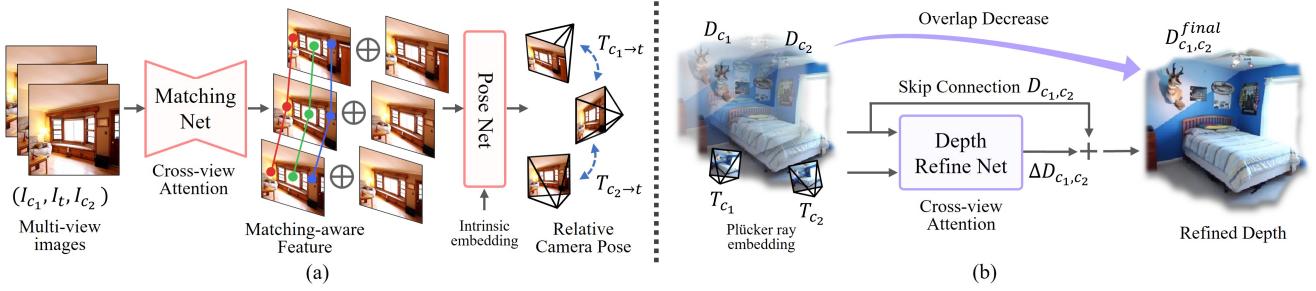


Figure 2. Matching-aware pose network (a) and depth refinement module (b). We leverage cross-view features from input images to achieve accurate camera pose estimation, and use these estimated poses to further refine the depth maps with spatial awareness.

rotation,  $R_{c_1 \rightarrow c_2} \in \mathbb{R}^{3 \times 3}$ , and translation,  $t_{c_1 \rightarrow c_2} \in \mathbb{R}^{3 \times 1}$ , between two images,  $I_{c_1}$  and  $I_{c_2}$ . We utilize the estimated camera poses to transform the Gaussian positions in each frame’s local coordinate system into an integrated global space. Then, we construct the 3D Gaussian representations for a scene by union of the generated Gaussians as follows,

$$\mathcal{G} = \text{TR}(\mathcal{G}_{c_1}, T_{c_1 \rightarrow t}) \cup \text{TR}(\mathcal{G}_{c_2}, T_{c_2 \rightarrow t}), \quad (4)$$

where  $\text{TR}(\mathcal{G}_{c_1}, T_{c_1 \rightarrow t})$  transforms the generated Gaussian  $\mathcal{G}_{c_1}$  into the  $I_t$ ’s coordinate system, and  $\mathcal{G}$  is the final 3D Gaussians that are used to render images. The final loss function to jointly train both  $f_\theta$  and  $f_\phi$  is defined as follows,

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{proj}} + \lambda_2 \mathcal{L}_{\text{ren}}, \quad (5)$$

$$\mathcal{L}_{\text{ren}} = \sum_{I_k \in \{I_{c_1}, I_{c_2}, I_t\}} \gamma_1 (1 - \text{SSIM}(I_k, \hat{I}_k)) + \gamma_2 \|I_k - \hat{I}_k\|_2, \quad (6)$$

where  $\mathcal{L}_{\text{proj}}$  is the reprojection loss (Eq. (2)) and  $\mathcal{L}_{\text{ren}}$  is the rendering loss that computes the error between input view images,  $I_k$ , and the rendered images,  $\hat{I}_k$ , from the constructed Gaussians  $\mathcal{G}$ . Note that in  $\mathcal{L}_{\text{proj}}$ , we use the rendered depth for  $I_t$  to maintain a consistent scale with estimated depth maps from the context images,  $I_{c_1}$  and  $I_{c_2}$ . In accordance with the prior pose-free generalizable methods, we assume that the camera intrinsic parameters are given from the camera sensor metadata [6, 22, 32, 54].

## 4.2. Architecture

As illustrated in Fig. 1, the proposed SelfSplat consists of four components: a multi-view and monocular encoder, a fusion and dense prediction block, a matching-aware pose estimation network, and a Gaussian decoder.

**Multi-view and monocular encoder.** For multi-view feature extraction from input view images, we begin by processing each image independently through a weight-sharing CNN architecture, followed by a multi-view Transformer to exchange information across different views. Specifically, a ResNet-like architecture [19] is used to extract 4x down-sampled features for each view. These features are then refined by a six-block Swin Transformer [37], which utilizes efficient local window self- and cross-attention mechanisms. The resulting cross-view-aware features are denoted

as  $F_{c_1}^{\text{mv}}, F_{c_2}^{\text{mv}} \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times C^{\text{mv}}}$ , where  $C^{\text{mv}}$  is the dimension. These features are subsequently processed to generate Gaussian attributes for rendering. As discussed in Sec. 4.1, since we do not generate Gaussian attributes for  $I_t$ ,  $I_t$  is excluded from the feature extraction in this module.

Despite substantial advancements in multi-view feature matching-based depth estimation methods, such as those leveraging epipolar sampling [3, 21] or plane-sweep techniques [4, 7, 65], these approaches continue to face challenges in handling occlusions, texture-less regions, and reflective surfaces. To address these limitations, we incorporate a monocular feature extractor, which has demonstrated robust performance across various downstream tasks [13, 52]. Specifically, we utilize a shared-weight Vision Transformer (ViT) model, CroCo [58, 59], as a monocular feature extractor. More specifically, input images are divided into non-overlapping patches with a patch size of 16 and processed by multi-head self-attention blocks and feed-forward networks in parallel. Then, we obtain robust monocular Transformer features  $F_{c_1}^{\text{mono}}, F_{c_2}^{\text{mono}} \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C^{\text{mono}}}$ , where  $C^{\text{mono}}$  denotes the channel dimension. Similar to the multi-view feature extraction, we do not extract the monocular feature from  $I_t$ . It is important to note that, unlike previous methods [62, 70] which use a pretrained DepthAnything [64] model as a ViT backbone and thus incorporate 3D priors, we employ CroCov2 [59] weights, allowing us to maintain a fully self-supervised framework

**Feature fusion and dense prediction.** To achieve consistent and fine-grained prediction of Gaussian primitives, we combine the multi- and single-view features, leveraging complementary information from both perspectives to enhance depth accuracy and robustness in complex scenes. We build our feature fusion block with Dense Prediction Transformer (DPT) [40] module. As the spatial resolutions between the two features are different, we first downsample the multi-view features by four to match with monocular ones. Then, CNN-based pyramidal architecture [34] is adopted to produce features at four different levels. Four intermediate outputs are pulled out from the encoder blocks for the monocular features. These are then simply concatenated at each level and used to produce dense predictions

through a combination of reassemble and fusion blocks.

Given the merged features,  $F_{c_1}^{\text{cat}}, F_{c_2}^{\text{cat}}$ , we utilize two branches of dense prediction module, one for the depth of 3D Gaussians,  $\text{DPT}_{\text{depth}}$ , and the other for the remaining Gaussian attributes  $\text{DPT}_g$ ,

$$\tilde{D}_k = \text{DPT}_{\text{depth}}(F_k^{\text{cat}}), \tilde{\mathcal{G}}_k = \text{DPT}_g(F_k^{\text{cat}}), k \in \{c_1, c_2\}, \quad (7)$$

where  $\tilde{\mathcal{G}}_k = \{(\delta x_j, \delta y_j, \alpha_j, \Sigma_j, c_j)\}_{j=1}^{HW}$  is a set of the predicted Gaussians with all attributes except ‘z’ coordinates and  $\tilde{D}_k$  is the predicted depth, further processed by the depth refinement module.  $\delta x_j$  and  $\delta y_j$  are the predicted offsets for each Gaussian, and the Gaussians for each input image in its coordinate system,  $\mathcal{G}_{c_1}, \mathcal{G}_{c_2}$ , can be obtained by adding the offsets to the pixel coordinates and unprojecting to 3D space using the refined depth  $D_{c_1}, D_{c_2}$ . Then, we construct the unified Gaussians by transforming them into a target coordinate system with the predicted poses (Eq. (4)).

**Matching-aware pose estimation.** To enable high-quality rendering and reconstruction, it is crucial to predict accurate camera poses since it defines the transformation in 3D space. We begin by employing the CNN-based pose network from previous studies [18, 29] and introduce our matching-awareness module as a novel encoding strategy. As shown in Fig 2-(a), we use a 2D U-Net [44] with cross-attention blocks to extract multi-view aware features from unposed images. Unlike the dense prediction module, we also incorporate the target view  $I_t$  as input and predict the relative camera poses for  $(I_{c_1}, I_t)$  and  $(I_{c_2}, I_t)$ . First, the matching network processes the triplet,

$$F_{c_1}^{\text{ma}}, F_t^{\text{ma}}, F_{c_2}^{\text{ma}} = \text{MatchingNet}(I_{c_1}, I_t, I_{c_2}), \quad (8)$$

where the matching aware features,  $F_k^{\text{ma}} \in \mathbb{R}^{H \times W \times 3}$ , have the same sizes as input images and inject these features into the pose network.

$$T_{c_1 \rightarrow t} = \text{PoseNet}([F_{c_1}^{\text{ma}}; I_{c_1}; E^{\text{int}}], [F_t^{\text{ma}}; I_t; E^{\text{int}}]), \quad (9)$$

where  $[.; .; .]$  concatenates input tensors along with the channel dimension, and  $E^{\text{int}} \in \mathbb{R}^{H \times W \times 3}$  is ray embedding of camera intrinsic matrix for scale-awareness. More specifically,  $E_{x,y}^{\text{int}} = K^{-1}p(x, y) \in \mathbb{R}^3$ , where  $K \in \mathbb{R}^{3 \times 3}$  is a camera intrinsic matrix and  $p(x, y) \in \mathbb{R}^3$  is a homogeneous coordinate of a pixel coordinate  $x, y$ . Note that the camera intrinsic parameters vary for different scenes but remain the same across different input views within the same scene.

**Pose-aware depth refinement.** In this module, we refine the estimated depth map,  $\tilde{D}_{c_1}, \tilde{D}_{c_2}$ , derived from the dense prediction module,  $\text{DPT}_{\text{depth}}$ , to improve the quality of rendering and reconstruction. The initial depth estimation,  $\tilde{D}_{c_1}, \tilde{D}_{c_2}$ , yields inconsistent estimation between input views that negatively impact the overall accuracy of the reconstruction, e.g., incorrectly overlapping Gaussians. To resolve the limitation, we propose our refine module that

leverages cross-view information with spatial awareness. While a few recent works have proposed depth refinement approaches [7, 70], our method uniquely differs by utilizing the predicted camera pose as additional information to resolve inconsistencies in the estimated depths across multiple input views. We employ a lightweight 2D U-Net, which takes current depth predictions, input images, and estimated poses as input and outputs residual depths for each view. The operation is defined as follows,

$$\begin{aligned} \Delta D_{c_1}, \Delta D_{c_2} &= \text{Refine}([\tilde{D}_{c_1}; I_{c_1}; E^{\text{ext}}(T_{c_1 \rightarrow t})], \quad (10) \\ &\quad [\tilde{D}_{c_2}; I_{c_2}; E^{\text{ext}}(T_{c_2 \rightarrow t})]), \end{aligned}$$

where  $\Delta D_k$  is the residual for each view depth, and the final depth  $D_k = \tilde{D}_k + \Delta D_k$  is obtained by adding the residual to the initial depth estimation for each view. Similar to our pose estimation module, there are cross-attention blocks in U-Net, and we utilize Plücker ray embedding to densely encode our estimated pose into a higher-dimensional representation space, e.g.,  $E^{\text{ext}}(T_{c_1 \rightarrow t}) \in \mathbb{R}^{H \times W \times 6}$  (Fig. 2-(b)).

	Pose-Free	w/o 3D prior	Depth	Fast	w/o Finetune
VAE [29]	✓	✓	✗	✓	✓
DBARF [6]	✓	✗ <sup>2</sup>	✓	✗	✗
FlowCAM [46]	✓	✗	✓	✗	✓
CopoNeRF [22]	✗ <sup>1</sup>	✓	✓	✗	✓
SelfSplat (Ours)	✓	✓	✓	✓	✓

Table 1. Baseline attributes compared to our proposed method.<sup>1</sup> CopoNeRF offers pose-free inference, but requires ground-truth pose supervision during training. <sup>2</sup> DBARF is trained from the pretrain generalizable NeRF, IBRNet [55] that has a 3D prior.

## 5. Experiments

### 5.1. Experiment Setup

We train and evaluate our model on three large-scale datasets: RealEstate10K (RE10k) [75], ACID [36], and DL3DV [35], which include diverse indoor and outdoor real estate videos, aerial outdoor nature scenes, and diverse real-world videos, respectively. For RE10k, we use 67,477 training and 7,289 testing scenes; for ACID, 11,075 training and 1,972 testing scenes, consistent with previous works [3, 7]. Lastly, for DL3DV, we use subsets of the dataset amounting to 2,000 scenes (3K and 4K) for training and testing on 140 benchmark scenes, following PF3plat [23]. We assess our model’s performance in reconstructing intermediate video frames between two context frames.

**Baselines.** We compare our model against existing pose-free generalizable novel view synthesis methods, includ-

	RE10k											
Method	Average			Small			Medium			Large		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
VAE [29]	20.65	0.643	0.325	18.78	0.585	0.414	20.56	0.646	0.319	23.13	0.701	0.241
DBARF [6]	12.57	0.494	0.474	10.48	0.497	0.522	12.39	0.487	0.475	15.55	0.513	0.415
FlowCAM [46]	22.29	0.711	0.313	20.74	0.679	0.375	22.15	0.709	0.313	24.58	0.754	0.241
CoPoNeRF [22]	21.03	0.693	0.256	19.70	0.670	0.348	20.99	0.695	0.285	22.70	0.715	0.217
Ours	<b>24.22</b>	<b>0.813</b>	<b>0.188</b>	<b>21.63</b>	<b>0.749</b>	<b>0.257</b>	<b>24.21</b>	<b>0.820</b>	<b>0.181</b>	<b>27.25</b>	<b>0.864</b>	<b>0.132</b>

Table 2. Quantitative results of novel view synthesis on RE10k dataset.

	ACID											
Method	Average			Small			Medium			Large		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
VAE [29]	24.02	0.666	0.287	23.76	0.679	0.306	24.21	0.664	0.295	23.99	0.658	0.263
DBARF [6]	15.48	0.572	0.397	13.87	0.618	0.422	15.51	0.584	0.398	16.72	0.521	0.378
FlowCAM [46]	25.59	0.721	0.294	25.37	0.730	0.312	25.64	0.714	0.305	25.73	0.723	0.267
CoPoNeRF [22]	23.30	0.668	0.278	22.92	0.692	0.304	23.42	0.667	0.283	23.45	0.649	0.251
Ours	<b>26.71</b>	<b>0.801</b>	<b>0.196</b>	<b>25.65</b>	<b>0.776</b>	<b>0.237</b>	<b>26.87</b>	<b>0.797</b>	<b>0.198</b>	<b>27.33</b>	<b>0.826</b>	<b>0.159</b>

Table 3. Quantitative results of novel view synthesis on ACID dataset.

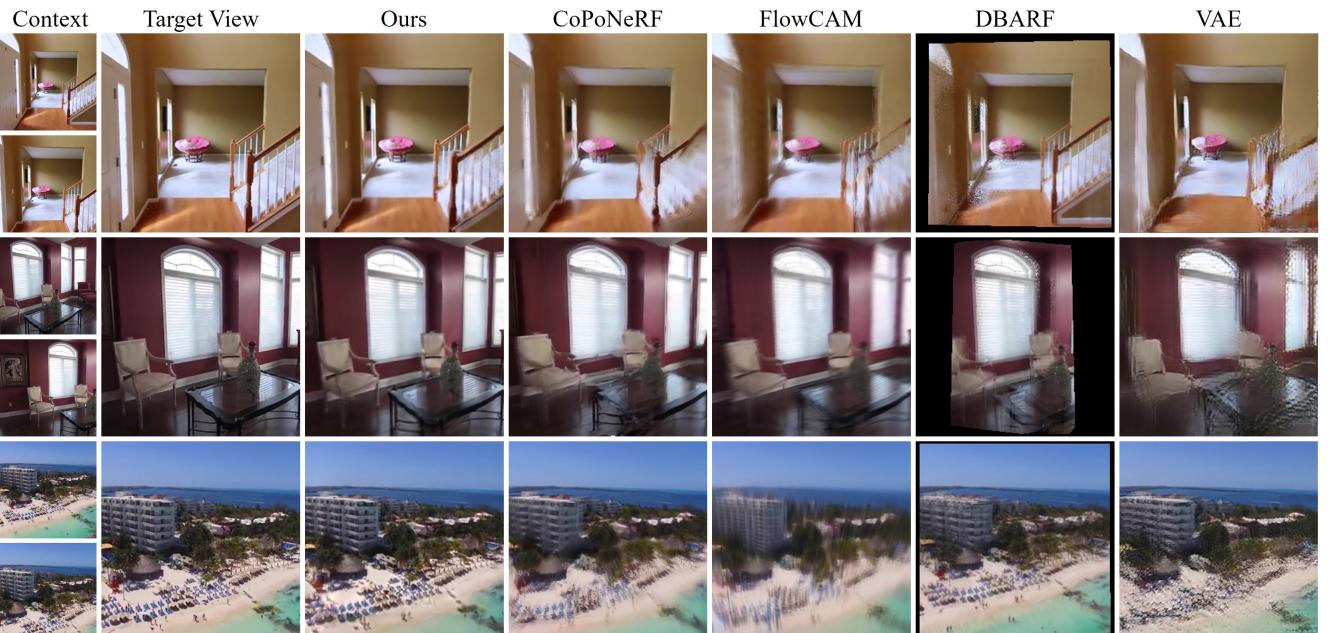


Figure 3. Qualitative comparison of novel view synthesis on RE10k (top two rows) and ACID (bottom row) datasets.

ing VAE [29], DBARF [6], FlowCAM [46], and CoPoNeRF [22], on two different tasks: novel view synthesis and relative camera pose estimation. We train all methods, including ours, using the same training curriculum, where the frame distance between context views increases gradually. We also provide an attribute overview in Tab. 1, showing the distinct features of our proposed method.

**Evaluation metrics.** For novel view synthesis, we use standard metrics: PSNR, SSIM [57], and LPIPS [71]. Pose estimation is evaluated based on geodesic rotation and translation angular error, following [22]. For RE10k and ACID, we categorize test context pairs by image overlap ratios to evaluate performance across small(0.05-0.6), medium(0.6-

0.8), and large(0.8+) overlap, identified by a pretrained image matching method [15]. For DL3DV, overlap categories are defined by frame intervals between context images: 6 frames for large and 10 frames for small overlap.

**Implementation details.** We employ the encoder part of pretrained CroCo [59] model as our monocular encoder, which is trained in a self-supervised manner, and utilized adapter [5] block designed to efficiently adapt pretrained ViT models to downstream tasks. For the Gaussian rasterizer, we implement it using gsplat [67], an open-source library for Gaussian Splatting [27], offering efficient computation and memory usage. We train RE10k and ACID with  $256 \times 256$  resolution, and for DL3DV, we use  $256 \times$

	RE10k															
Method	Average				Small				Medium				Large			
	Rotation		Translation		Rotation		Translation		Rotation		Translation		Rotation		Translation	
	Avg. $\circ \downarrow$	Med. $\circ \downarrow$														
VAE [29]	3.859	2.818	115.746	113.598	6.971	6.936	118.211	116.226	3.595	3.037	112.444	109.959	1.127	0.525	123.840	132.181
DBARF [6]	2.471	1.471	36.069	24.308	4.771	3.009	38.292	25.822	2.043	1.440	32.749	20.899	1.222	0.914	44.500	35.788
FlowCAM [46]	1.438	1.160	22.524	15.917	2.483	1.741	26.044	19.583	1.264	1.137	19.889	14.355	0.801	0.745	27.179	19.328
CoPoNeRF [22]	0.839	0.587	9.175	5.538	<b>1.281</b>	0.830	<b>10.173</b>	<b>6.047</b>	0.784	0.583	8.431	5.179	0.511	0.326	10.487	6.232
Ours	<b>0.750</b>	<b>0.362</b>	<b>9.095</b>	<b>4.438</b>	1.523	<b>0.701</b>	14.954	7.267	<b>0.603</b>	<b>0.346</b>	<b>7.593</b>	<b>3.976</b>	<b>0.344</b>	<b>0.206</b>	<b>7.281</b>	<b>3.799</b>

Table 4. Quantitative results of pose estimation on RE10k dataset.

Method	ACID															
	Average				Small				Medium				Large			
	Rotation		Translation													
Avg. $\circ \downarrow$	Med. $\circ \downarrow$															
VAE [29]	1.631	0.461	86.180	79.987	2.789	0.617	81.056	72.060	1.568	0.497	85.372	77.761	0.806	0.355	91.187	87.526
DBARF [6]	1.975	0.860	49.906	95.399	2.899	1.060	52.275	36.895	1.879	0.751	49.396	36.702	1.373	0.745	48.693	33.672
FlowCAM [46]	3.846	2.819	92.786	82.976	4.787	3.574	84.599	76.659	4.297	3.001	91.262	81.019	2.548	2.168	101.074	91.248
CoPoNeRF [22]	1.058	0.354	19.888	9.757	<b>1.731</b>	0.503	<b>20.572</b>	9.449	1.025	0.365	19.617	9.401	0.575	0.261	19.694	10.578
Ours	<b>0.981</b>	<b>0.199</b>	<b>16.329</b>	<b>4.535</b>	1.787	<b>0.362</b>	21.631	<b>5.681</b>	<b>0.952</b>	<b>0.219</b>	<b>16.006</b>	<b>4.518</b>	<b>0.386</b>	<b>0.134</b>	<b>12.592</b>	<b>3.828</b>

Table 5. Quantitative results of pose estimation on the ACID dataset.

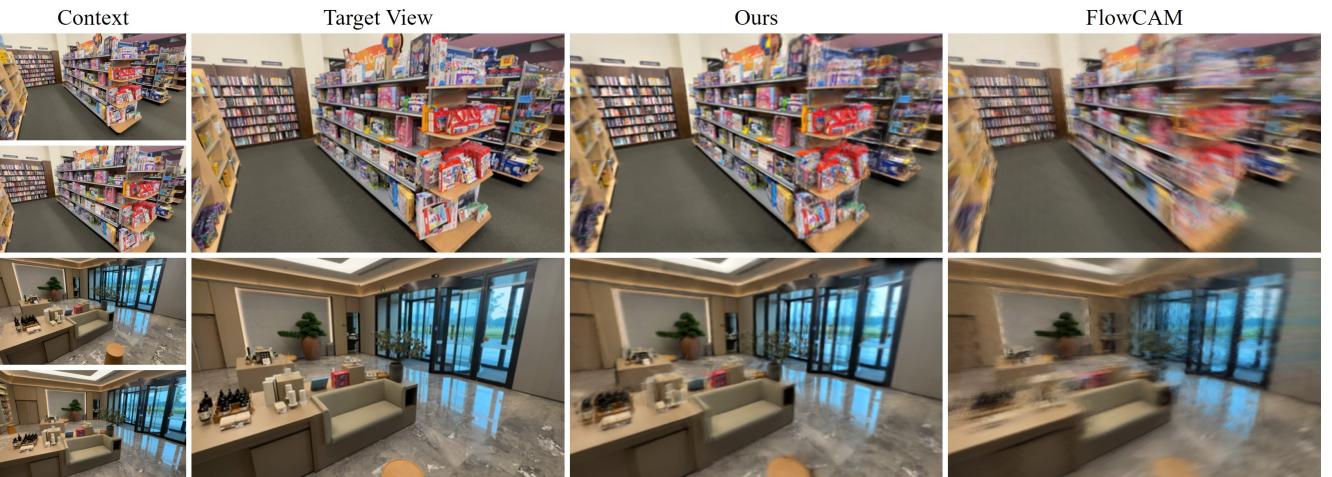


Figure 4. Qualitative comparison of novel view synthesis on DL3DV dataset.

Method	DL3DV																	
	Small (10 frame)								Large (6 frame)									
	Rotation			Translation			Rotation			Translation			Rotation			Translation		
PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	
FlowCAM [46]	21.01	0.608	0.411	1.138	1.011	22.385	16.432	23.52	0.710	0.314	0.705	0.626	28.681	18.214				
Ours	<b>21.91</b>	<b>0.723</b>	<b>0.279</b>	<b>0.985</b>	<b>0.573</b>	<b>9.681</b>	<b>5.164</b>	<b>24.82</b>	<b>0.822</b>	<b>0.200</b>	<b>0.634</b>	<b>0.256</b>	<b>12.057</b>	<b>6.998</b>				

Table 6. Quantitative results of novel view synthesis and pose estimation on DL3DV dataset.

448 to accommodate the wider view in our experiments.

## 5.2. Results

**Novel view synthesis.** We report quantitative results in Tab. 2, 3 and qualitative results in Fig. 3 for RE10k and ACID datasets, while Tab. 6 and Fig. 4 for DL3DV dataset. Our method outperforms the baselines on all metrics, especially in terms of perceptual distance. These observations can be further confirmed by the rendering results that our method effectively captures fine details of 3D structure.

**Relative pose estimation.** Tab. 4, 5, and 6 present the quantitative results for camera pose estimation between two images across the datasets. Our approach consistently achieves lower errors in both average and median devia-

tions, highlighting its accuracy and robustness. The qualitative results in Fig. 5, visualizing epipolar lines from the estimated poses also demonstrates the effectiveness of our approach in capturing accurate geometric alignments.

**Cross-Dataset Generalization.** To evaluate the generalization performance on out-of-distribution scenes, we train the models on RE10k (ACID) dataset and test them on ACID (RE10k) dataset without additional finetuning. As shown in Tab. 7, SelfSplat outperforms previous methods on unseen datasets, demonstrating robust generalization capabilities.

## 5.3. Ablations and Analysis

We provide quantitative and qualitative results on ablations studies in Tab. 8 and Fig. 6. All methods are trained for

Method	RE10k → ACID							ACID → RE10k						
				Rotation		Translation					Rotation		Translation	
	PSNR ↑	SSIM ↑	LPIPS ↓	Avg.° ↓	Med.° ↓	Avg.° ↓	Med.° ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Avg.° ↓	Med.° ↓	Avg.° ↓	Med.° ↓
FlowCAM [46]	25.31	0.715	0.297	1.607	1.074	32.316	21.135	21.13	0.667	0.329	4.426	3.995	74.873	63.667
CoPoNeRF [22]	23.56	0.683	0.287	2.301	0.619	19.620	8.642	18.89	0.607	0.364	4.159	2.893	20.435	13.572
Ours	<b>26.60</b>	<b>0.793</b>	<b>0.206</b>	<b>1.119</b>	<b>0.249</b>	<b>18.607</b>	<b>5.864</b>	<b>21.65</b>	<b>0.728</b>	<b>0.242</b>	<b>1.618</b>	<b>0.867</b>	<b>17.993</b>	<b>10.228</b>

Table 7. Cross-dataset generalization. We train the models on RE10k (ACID) dataset and directly evaluate on ACID (RE10k) dataset.

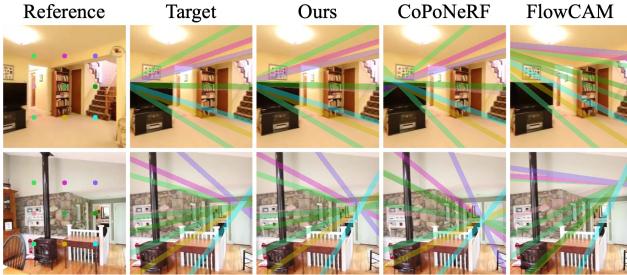


Figure 5. Epipolar lines visualization. We draw the lines from reference to target frame using relative camera pose.

50,000 iterations on RE10k dataset for a fair comparison.

#### Importance of matching awareness in pose estimation.

To measure the importance of adopting cross-view features in our pose network, we conduct a study (“No Matching awareness”) by removing it from the pose network. Quantitatively, it leads to a drop in pose metrics: translation error increases by 1.5 degrees, which also negatively impacts rendering scores, decreasing PSNR by 0.4 dB. These results highlight that our encoding methods with multi-view awareness help capture relationships between frames, improving both pose estimation and novel view synthesis.

**Importance of depth refinement module.** We conduct a study (“No Depth Refine”) on our depth refinement module to validate its effectiveness. The results indicate a clear decline across all metrics: PSNR drops by 0.6 dB, and translation discrepancy increases by 1.1 degrees. Additionally, misalignment of overlapping Gaussians leads to degrading in visual quality, such as motion blur artifacts, demonstrating that our refinement scheme enhances the multi-view consistency of depth predictions.

#### How self-supervised depth estimation method and 3D-GS representation can make reciprocal improvement?

We explore the benefits of combining self-supervised depth estimation with explicit 3D representation by comparing SelfSplat with two variants (“No Reprojection Loss”, “No Rendering Loss”). Training without reprojection loss shows a significant performance decline across all metrics, particularly a 1.5 dB drop in PSNR, indicating challenges in accurately positioning Gaussians—a crucial factor for precise 3D reconstruction and novel view synthesis. In the “No Rendering Loss” variant, we replaced the rendered depth of the target view previously used in reprojection loss with an estimated depth map from the image using a dense prediction module. To validate the impact of incorporating 3D-GS, we also account for gradients of rotation,  $R \in \mathbb{R}^{3 \times 3}$ ,

and translation,  $t \in \mathbb{R}^{3 \times 1}$ , in camera poses. The rendering loss gradients with respect to translation and rotation are:

$$\frac{\delta \mathcal{L}_{\text{ren}}}{\delta t} = -\sum_j \frac{\delta \mathcal{L}_{\text{ren}}}{\delta \tilde{\mu}_j}, \quad \frac{\delta \mathcal{L}_{\text{ren}}}{\delta R} = -\left[ \sum_j \frac{\delta \mathcal{L}_{\text{ren}}}{\delta \tilde{\mu}_j} (\mu_j - t)^\top \right] R, \quad (11)$$

where  $\tilde{\mu}_j$  is a splatted Gaussian in rendering viewspace. Excluding rendering loss results in degraded pose metrics, a common issue in self-supervised depth estimation methods with limited image overlap. Our framework effectively addresses this by combining explicit 3D-GS representation with rendering loss, improving depth and pose estimation.



Figure 6. Ablation studies on our proposed component.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	Rot. Avg.° ↓	Trans. Avg.° ↓
Ours	<b>22.65</b>	<b>0.764</b>	<b>0.222</b>	<b>1.036</b>	<b>13.705</b>
No Matching Network	22.21	0.735	0.241	1.308	15.171
No Depth Refine	22.05	0.744	0.224	1.064	14.8
No Reprojection Loss	21.12	0.672	0.293	2.236	28.584
No Rendering Loss	-	-	-	8.581	64.436

Table 8. Ablations. Our methods achieves better alignment of 3D Gaussians, with accurate pose and consistent depth estimations.

## 6. Conclusion

We present SelfSplat, a pose-free generalizable 3D Gaussian Splatting model that does not require pretrained 3D priors or an additional fine-tuning stage. Our method effectively integrates a 3D-GS representation with self-supervised depth estimation techniques to recover 3D geometry and appearance from unposed monocular videos. We conduct extensive experiments on diverse real-world datasets to demonstrate its effectiveness, showcasing its ability to produce photorealistic novel view synthesis and accurate camera pose estimation. We believe that SelfSplat represents a significant step forward in 3D representation learning, offering a robust solution for various applications.

# SelfSplat: Pose-Free and 3D Prior-Free Generalizable 3D Gaussian Splatting

## Supplementary Material

### A. Additional Details

#### A.1. Architectural Details

For the prediction of 3D Gaussians [27], we utilize the monocular, multi-view encoder and the fusion block. Unlike previous methods that utilize DepthAnything [64] as a monocular encoder [62, 70] or UniMatch [61] as a multi-view encoder [7, 47], we only employ the encoder part of Croco [58] as our monocular encoder which is trained in a fully self-supervised manner. For the multi-view encoder, we adopt the backbone of [61] with randomly initialized weights. Then, we unify features from monocular and multi-view encoders using DPT [40] block. For a detailed architecture for the fusion module, see Fig. 7.

#### A.2. Implementation Details

For our monocular encoder, we utilized Adapter [5], which keeps the model parameters frozen while training additional residual networks for each layer. Specifically, a residual MLP block, comprising a down-projection layer and an up-projection layer, is introduced within each layer of the transformer encoder. Considering the channel dimension of the original encoder,  $C^{\text{mono}} = 1024$ , we set the low rank hidden dimension of AdaptMLP,  $C^{\text{adapt}} = 32$ , to efficiently reduce computational overhead while maintaining sufficient capacity for adaptation.

For 3D Gaussian primitives, we set the order of spherical harmonics expansion to 1, enabling the representation to extend beyond the Lambertian color model. When warping the color model from each frame’s local coordinate system into an integrated global space which requires the Wigner matrices in general case, we simplify the rotation of the first level of spherical harmonics,  $Y_1(r_d) = [Y_1^{-1}(r_d), Y_1^0(r_d), Y_1^1(r_d)]$ , as follows:

$$Y_1(r_d) = \sqrt{\frac{3}{4\pi}} \Pi r_d, \quad \Pi = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix},$$

where  $r_d \in \mathbb{S}^2$  is the viewing direction derived from the estimated camera poses. We adopt this warping protocol from Splatter Image [50] which is a pose-required generalizable 3D reconstruction model using 3D Gaussian Splatting.

#### A.3. Training Details

We train all baseline models, including ours, using custom data loaders. For RealEstate10K [75] (RE10k) and ACID [36] datasets, the distance between context frames is progressively increased from 5 to 25, and target frames

are randomly selected between the context frames within this range. Each model is trained for 200K iterations and for baselines we used the default hyperparameter settings provided by the respective authors. The only exception is DBARF [6], which is trained for 400K iterations due to its official implementation supporting only a batch size of one. We provide our detailed training hyperparameters in Tab. 9 and we train our model on a single H100 GPU, which takes approximately for 3 days. For the experiment on DL3DV [35] dataset, we initialize the model with pretrained weights from RE10k dataset and train it for 50K iterations on a single H100 GPU with a batch size of 6. The distance between context frames is gradually increased from 2 to 10. This procedure is applied to FlowCAM [46] in the same way which is the baseline model on DL3DV dataset.

For VAE [29], which was initially designed for novel view synthesis from a single image, we modify its architecture following the approach in [69] to handle multi-view input images. Specifically, we employ two separate encoders and use their mean output as the input to the decoder which synthesizes novel view images. All other hyperparameters remain the same as the official implementation.

SelfSplat	
Config	Value
optimizer	Adam [28]
scheduler	Linear
learning rate	1e-4
gradient clipping	0.5
batch size	12
total iters.	200,000
warmup iters.	2,000

Table 9. Training hyperparameters.

#### A.4. Evaluation Details

During the evaluation on RE10k and ACID datasets, we set the interval between context frames to 40 and select the middle frame as the target view point. This target frame is used as the ground truth for metric evaluations in novel view synthesis and camera pose estimation. For the overlap categories, we utilize the pretrained feature matching model, RoMa [15], to estimate the overlap ratios between the first context frame and the target frame.

For RE10k dataset, the split proportions are 18.26% for large, 60.56% for medium, and 21.17% for small categories. In ACID dataset [36], the proportions are 33.05% for large, 41.15% for medium, and 25.80% for small.

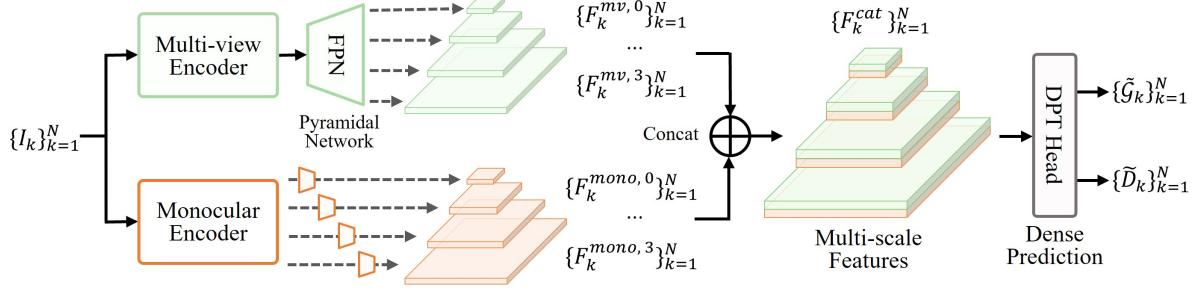


Figure 7. Detailed 3D Gaussian prediction architecture. This module takes only context images as input.

## B. Additional Experiment Analysis

### B.1. Inference Cost

We report the memory and time consumption required to synthesize a single  $256 \times 256$  image during the inference stage in Table 10. Memory usage is measured as the peak memory during inference, while the number of rays per batch is adjusted if necessary. Except for VAE [29], which generates novel view images without rendering operations (utilize 2D CNN blocks) and thus fail to reconstruct interpretable 3D scene representations, our method achieves significantly lower memory usage and faster rendering speed with explicit 3D representations, demonstrating its efficiency and practical usage in real-world scenarios.

Method	Mem. (GB)	Time (s)
VAE [29]	4.694	<b>0.0003</b>
DBARF [6]	2.079	0.254
FlowCAM [46]	16.644	0.801
CoPoNeRF [22]	16.802	5.624
Ours	<b>1.795</b>	0.002

Table 10. Memory and time consumption analysis. All baselines including ours are measured on a single NVIDIA RTX 4090 GPU.

### B.2. Using N Context Views

We further evaluate the model’s performance across various numbers of input views, considering its practical application where more than two views are commonly used. The total number of frames is evenly divided based on the number of context views, and target frames are sampled between the context frames. Additionally, we generate a camera trajectory using the selected view points (context and target), and the Absolute Trajectory Error (ATE) is measured to validate the accuracy of the reconstructed camera path. We evaluate on RE10k dataset with 3 context views (80 frames) and 4 context views (120 frames) settings. As shown in Tab. 11 and Fig. 8, our method demonstrates superior performance in both novel view synthesis and camera

trajectory estimation, as well as its ability to scale effectively with multiple input views and estimations over extended ranges without any further finetuning.

Method	3 views (80 frames)				4 views (120 frames)			
	PSNR↑	SSIM↑	LPIPS↓	ATE↓	PSNR↑	SSIM↑	LPIPS↓	ATE↓
FlowCAM [46]	19.75	0.630	0.412	0.048	18.91	0.606	0.449	0.081
Ours	<b>21.12</b>	<b>0.761</b>	<b>0.241</b>	<b>0.031</b>	<b>19.52</b>	<b>0.717</b>	<b>0.283</b>	<b>0.053</b>

Table 11. Quantitative results of using different numbers of context views on RE10k dataset.

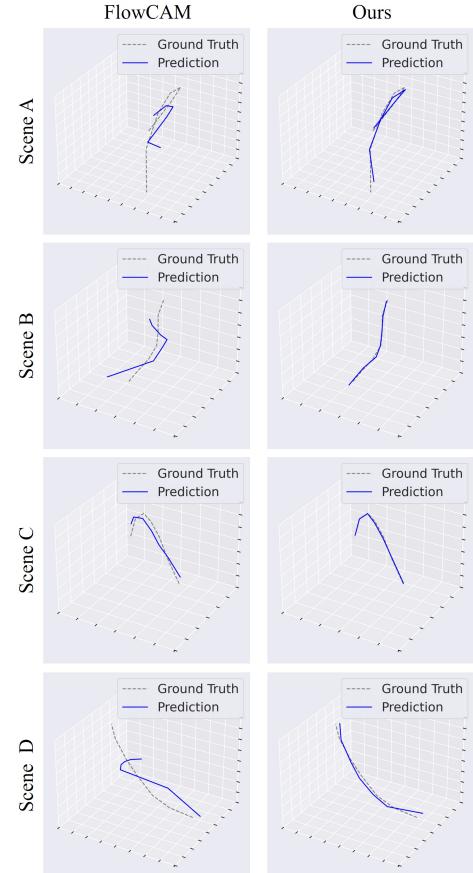


Figure 8. Visualization of camera trajectory on RE10k dataset. Construction of trajectory only consider the translation part of the estimated camera poses.

### B.3. Additional Comparison

For the reader’s reference, we provide a comparison with Splatt3R [45], which is also a pose-free, feed-forward Gaussian Splatting method for 3D reconstruction and novel view synthesis from stereo pairs. We omitted this model in the main paper because it requires ground-truth depth and camera pose annotations during training, which are not available in the datasets we used: RE10k, ACID [36], and DL3DV [35]. Acknowledging the differences in training data—Splatt3R was trained on ScanNet++ [12], whereas our model was trained on RE10k—we evaluate them on the DTU [24] dataset, which is an out-of-distribution dataset for both. As shown in Tab. 12 and Fig. 9, our method achieves better performance than the baseline in both evaluation metrics and visual quality, and also outperforms pixelSplat [3] which is a pose-required method in training and evaluation stage. The main reason Splatt3R cannot estimate a consistent scene scale is its reliance on a fixed pretrained MASt3R [31] model, which is trained using metric camera poses, and difference between estimated intrinsic parameters and ground truth intrinsic parameters. Thus, using the DTU dataset, which consists of unseen novel scenes, they fail to align consistent 3D Gaussians.

Training Data	Method	PSNR↑	SSIM↑	LPIPS↓
ScanNet++ [12]	Splatt3R [45]	10.24	0.295	0.629
	pixelSplat [3]	12.89	0.382	0.560
RE10k [75]	MVSplat [7]	13.94	0.473	0.385
	Ours	<b>13.14</b>	<b>0.425</b>	<b>0.448</b>

Table 12. Quantitative results of novel view synthesis on DTU dataset. While pixelSplat [3] and MVSplat [7] are pose-required methods, we include them for the reader’s reference.

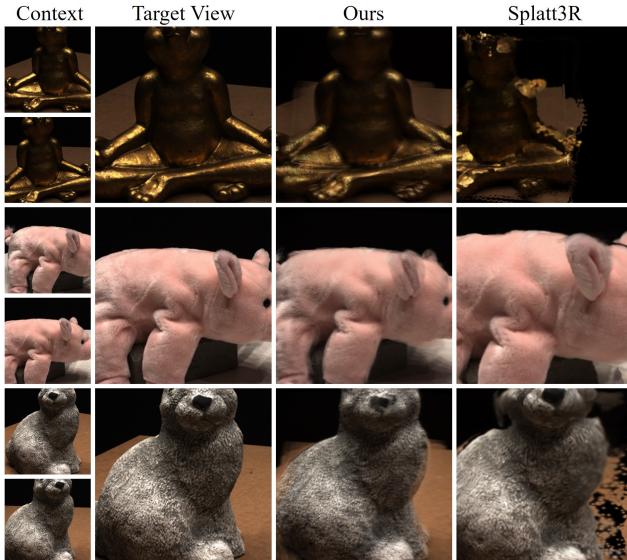


Figure 9. Qualitative comparison of novel view synthesis on DTU dataset.

### B.4. Baseline Comparisons

We provide additional baseline results on cross-dataset generalization in Tab. 13.

Method	RE10k → ACID				ACID → RE10k			
	PSNR ↑	SSIM ↑	Rot.° ↓	Trans.° ↓	PSNR ↑	SSIM ↑	Rot.° ↓	Trans.° ↓
VAE	23.67	0.649	1.619	117.721	18.98	0.537	3.744	65.884
DBARF	23.56	0.644	1.772	51.969	12.60	0.502	3.306	47.851
Ours	<b>26.60</b>	<b>0.793</b>	<b>1.119</b>	<b>18.607</b>	<b>21.65</b>	<b>0.728</b>	<b>1.618</b>	<b>17.993</b>

Table 13. Additional comparison on cross-dataset generalization.

### B.5. Additional Ablation and Analysis

We provide additional ablation studies and analyses, focusing on our encoder module. All methods are trained on RE10k [75] for 50k iterations, following the same procedure as in the main paper. As shown in Tab. 14, our feature fusion module with CroCo [58] initialization shows superior results in evaluation metrics.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	Rot. Avg.° ↓	Trans. Avg.° ↓
Ours	<b>22.65</b>	<b>0.764</b>	<b>0.222</b>	<b>1.036</b>	<b>13.705</b>
No CroCo [58] Init	22.15	0.738	0.240	1.091	14.042
No Monocular Encoder	21.88	0.733	0.247	1.394	17.125
No Multi-view Encoder	21.47	0.731	0.243	1.233	16.327

Table 14. Ablation studies on the encoder module design.

**Pretrained weight.** Since our goal is to use only unposed raw video datasets without 3D priors, we utilized CroCo, trained in a self-supervised manner. While DUST3R [56] or MASt3R [31] pre-trained weight could enhance performance, we focus on demonstrating that 3D foundation models can be trained without costly 3D annotations.

### B.6. Architectural and Evaluation Design

We designed our evaluation protocol assuming that there are no given poses, so we made a separate pose block (context and target) and a Gaussian branch (only context) independently. Thus, target images are used to estimate camera poses for following novel view synthesis evaluations. All baselines follow this protocol in their original implementations, except for CoPoNeRF [22] which utilizes given camera poses, so we substitute these poses in CoPoNeRF with estimated ones for a fair comparison.

### B.7. Depth Visualization

We also provide the visualization of depth maps generated through rendering, which is essential for producing interpretable 3D representations. By comparing the results of our method with previous approaches, as shown in Fig. 10, SelfSplat demonstrates robust and reliable depth maps derived from 3D scene structures.

### C. Limitations

While we demonstrate high-quality 3D geometry estimation in this work, the current framework still possesses limitations. First, further technical improvements are needed to

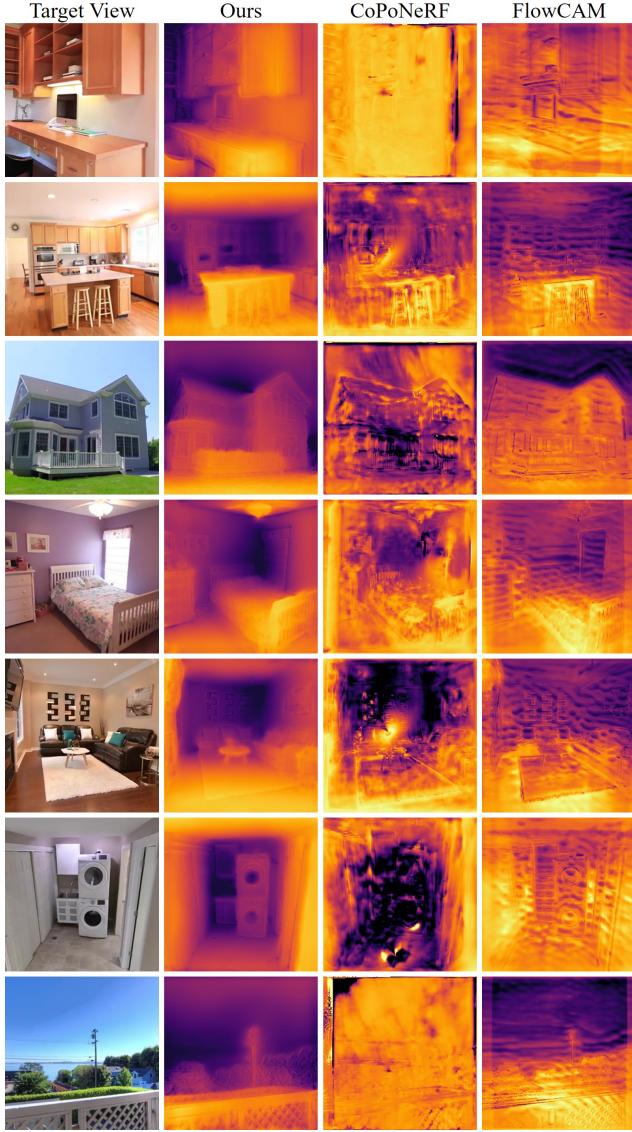


Figure 10. Qualitative comparison of depth visualization on RE10k dataset. Depth maps are obtained following the rendering process.

support wider baseline scenarios, such as a  $360^\circ$  scene reconstruction from unposed images in a single forward pass. Second, our framework struggles with dynamic scenes where both camera and object motion are present. Addressing these complex scenarios may benefit from incorporating multi-modal priors [43, 49] for robust and consistent alignment across wide and dynamic scene space.

## D. Additional Results

We provide additional results on the following pages including novel view synthesis and epipolar line visualizations.

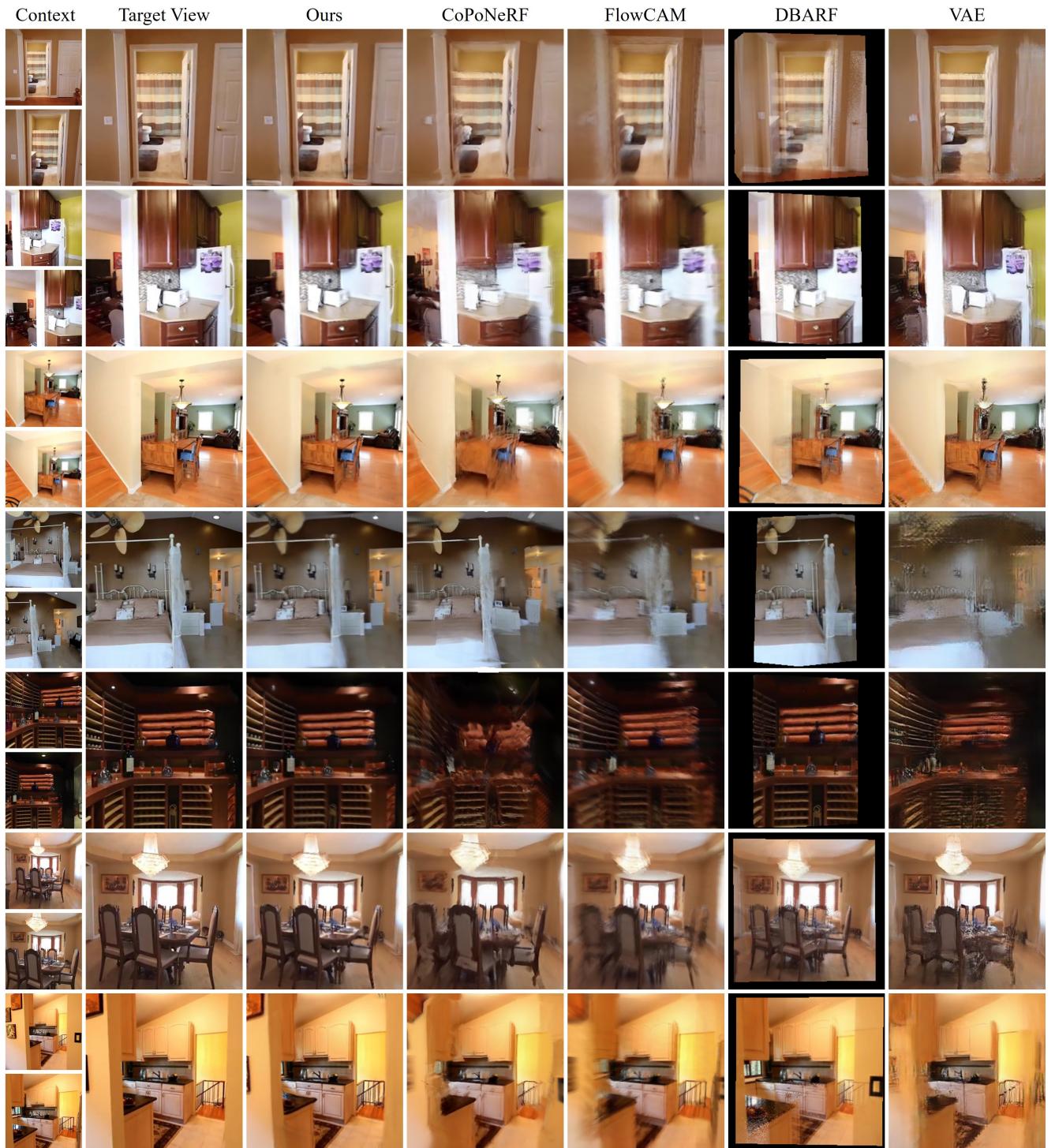


Figure 11. Qualitative comparison of novel view synthesis on RE10k dataset.

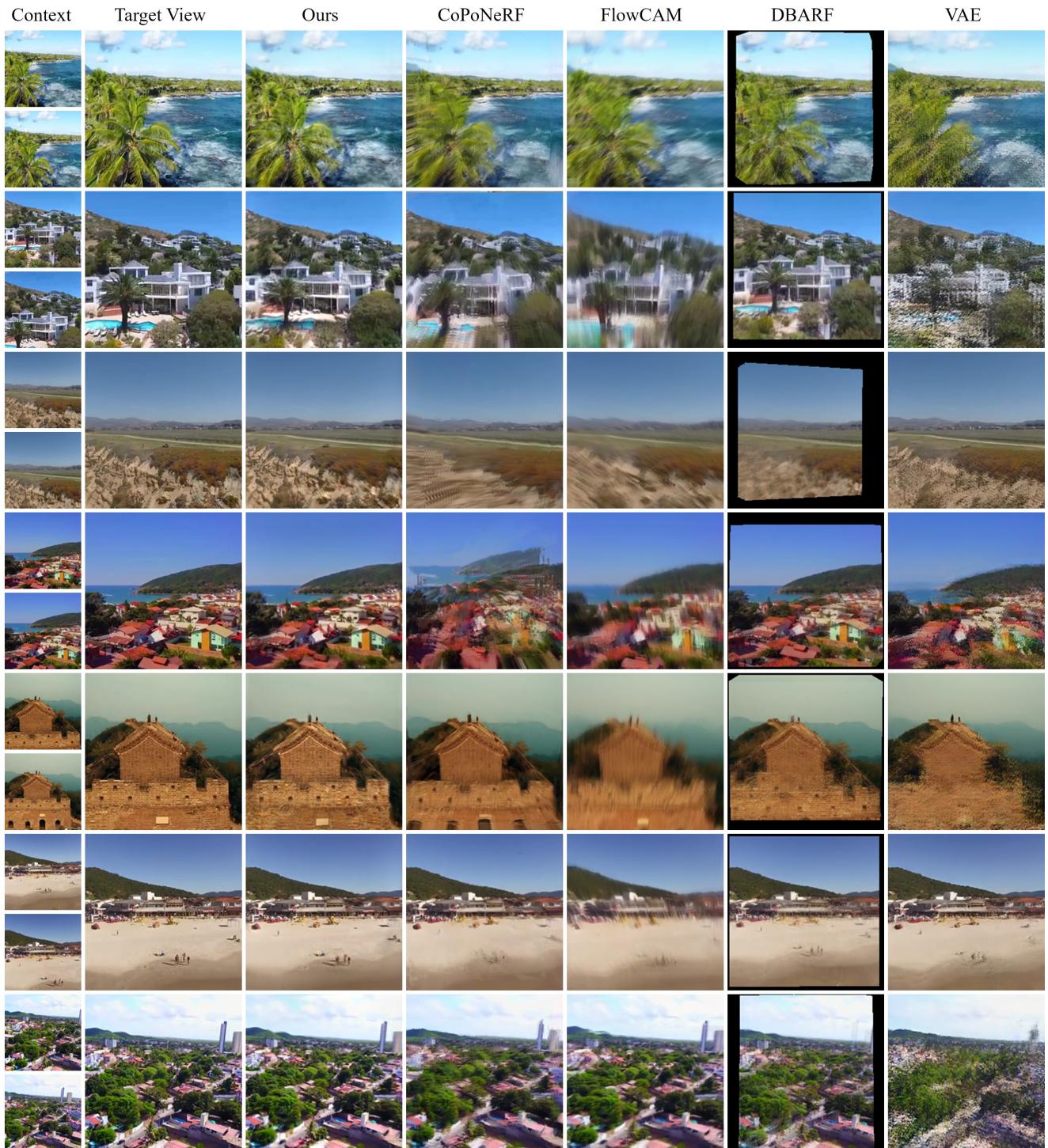


Figure 12. Qualitative comparison of novel view synthesis on ACID dataset.

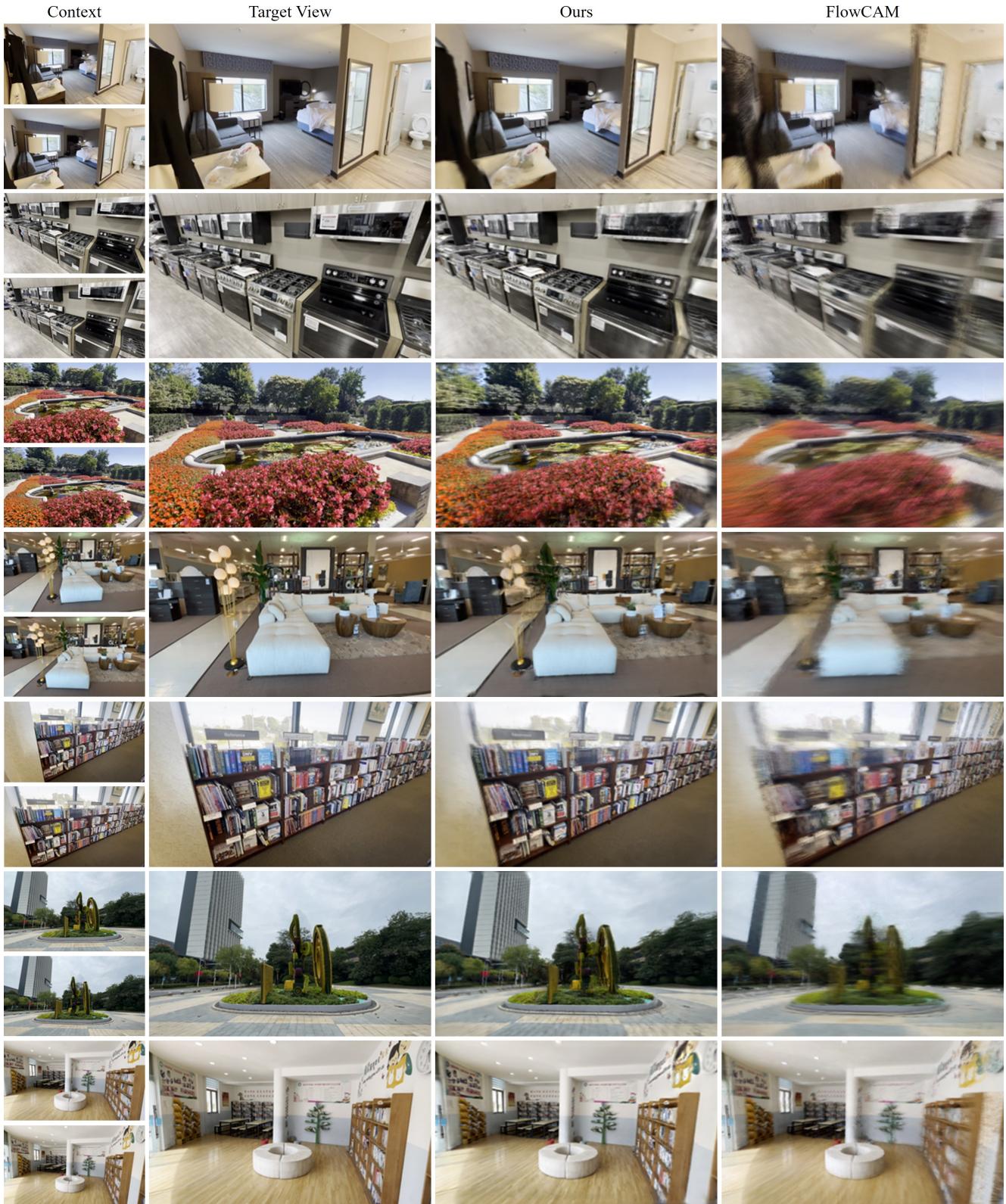


Figure 13. Qualitative comparison of novel view synthesis on DL3DV dataset.

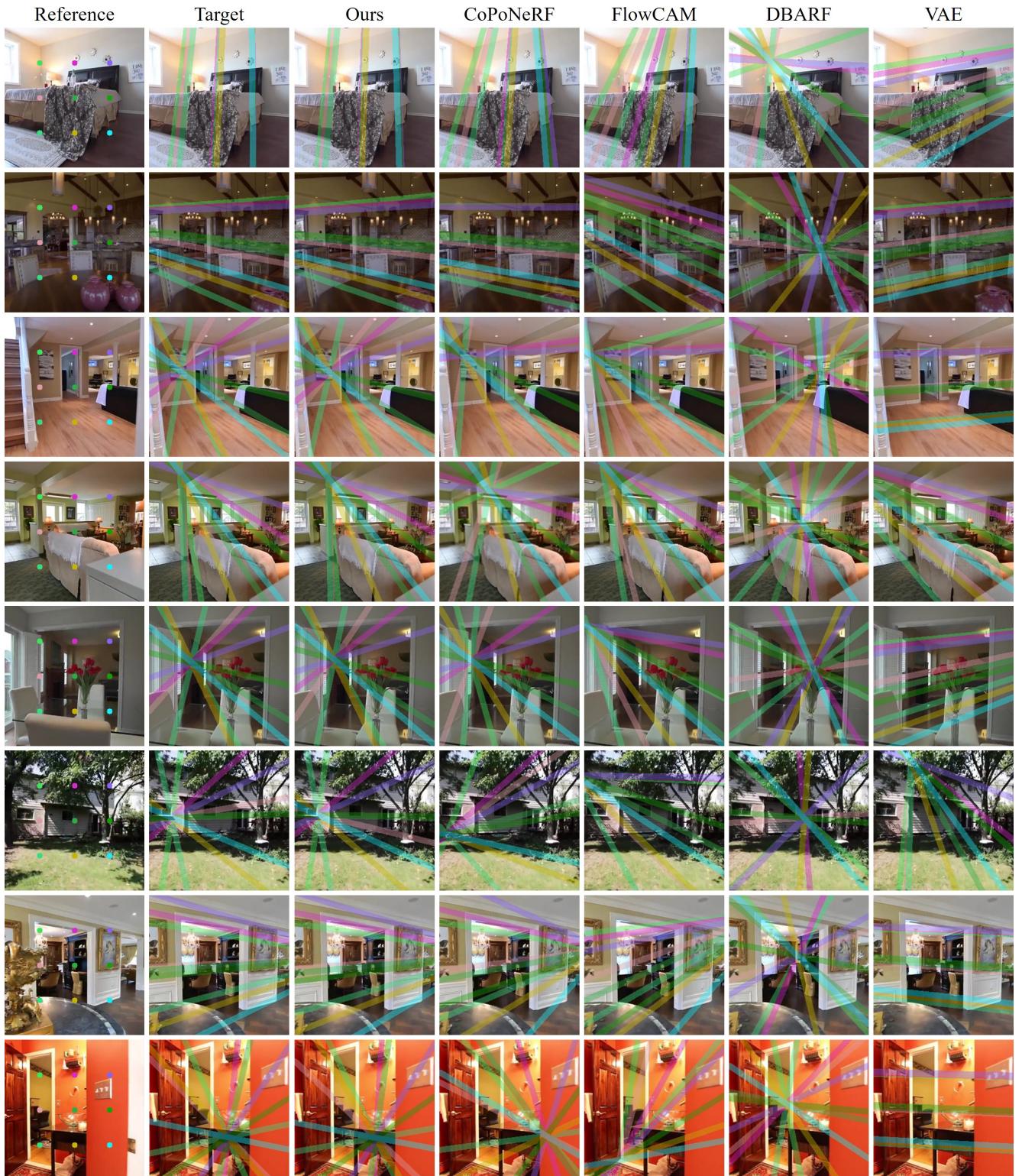


Figure 14. Epipolar lines visualization on RE10k dataset. We draw the lines from reference to target frame using relative camera pose.

## References

- [1] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. *Advances in neural information processing systems*, 32, 2019. 3
- [2] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4160–4169, 2023. 2
- [3] 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, 3, 4, 5, 11
- [4] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14124–14133, 2021. 4
- [5] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022. 6, 9
- [6] Yu Chen and Gim Hee Lee. Dbarf: Deep bundle-adjusting generalizable neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24–34, 2023. 1, 2, 4, 5, 6, 7, 9, 10
- [7] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. *arXiv preprint arXiv:2403.14627*, 2024. 2, 4, 5, 9, 11
- [8] Boris Chidlovskii and Leonid Antsfeld. Self-supervised pretraining and finetuning for monocular depth and visual odometry. *arXiv preprint arXiv:2406.11019*, 2024. 3
- [9] Boris Chidlovskii and Leonid Antsfeld. Self-supervised pretraining and finetuning for monocular depth and visual odometry. *arXiv preprint arXiv:2406.11019*, 2024. 2
- [10] Kyunghyun Cho. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014. 2
- [11] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2514–2523, 2020. 2
- [12] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 11
- [13] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4
- [14] 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. 1
- [15] Johan Edstedt, Qiyu Sun, Georg Bökmä, Mårten Wadenbäck, and Michael Felsberg. Roma: Robust dense feature matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19790–19800, 2024. 6, 9
- [16] Irving Fang, Kairui Shi, Xujin He, Siqi Tan, Yifan Wang, Hanwen Zhao, Hung-Jui Huang, Wenzhen Yuan, Chen Feng, and Jing Zhang. Fusionsense: Bridging common sense, vision, and touch for robust sparse-view reconstruction. *arXiv preprint arXiv:2410.08282*, 2024. 1
- [17] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros, and Xiaolong Wang. Colmap-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20796–20805, 2024. 2
- [18] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3828–3838, 2019. 2, 3, 5
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4
- [20] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 2
- [21] Yihui He, Rui Yan, Katerina Fragkiadaki, and Shouo-I Yu. Epipolar transformers. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 7779–7788, 2020. 4
- [22] Sunghwan Hong, Jaewoo Jung, Heeseong Shin, Jiaolong Yang, Seungryong Kim, and Chong Luo. Unifying correspondence, pose and nerf for pose-free novel view synthesis from stereo pairs. *arXiv preprint arXiv:2312.07246*, 2023. 1, 2, 4, 5, 6, 7, 8, 10, 11
- [23] Sunghwan Hong et al. Pf3plat: Pose-free feed-forward 3d gaussian splatting. *arXiv:2410.22128*, 2024. 2, 5
- [24] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413, 2014. 11
- [25] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, et al. Vr-gs: a physical dynamics-aware interactive gaussian splatting system in virtual reality. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–1, 2024. 1
- [26] Nikhil Keetha, Jay Karhade, Krishna Murthy Jatavallabhula, Gengshan Yang, Sebastian Scherer, Deva Ramanan, and Jonathon Luiten. Splatamat: Splat track & map 3d gaussians

- for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21357–21366, 2024. 2
- [27] 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, 6, 9
- [28] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 9
- [29] Zihang Lai, Sifei Liu, Alexei A Efros, and Xiaolong Wang. Video autoencoder: self-supervised disentanglement of static 3d structure and motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9730–9740, 2021. 5, 6, 7, 9, 10
- [30] Christoph Lassner and Michael Zollhofer. Pulsar: Efficient sphere-based neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1440–1449, 2021. 1
- [31] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. *arXiv preprint arXiv:2406.09756*, 2024. 3, 11
- [32] Hao Li et al. Ggrt: Towards generalizable 3d gaussians without pose priors in real-time. *arXiv:2403.10147*, 2024. 1, 2, 4
- [33] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5741–5751, 2021. 2
- [34] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4
- [35] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024. 2, 5, 9, 11
- [36] 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. 2, 5, 9, 11
- [37] 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. 4
- [38] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1995. 1
- [39] 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
- [40] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 4, 9
- [41] Adam Rashid, Satvik Sharma, Chung Min Kim, Justin Kerr, Lawrence Yunliang Chen, Angjoo Kanazawa, and Ken Goldberg. Language embedded radiance fields for zero-shot task-oriented grasping. In *7th Annual Conference on Robot Learning*, 2023. 1
- [42] Chris Rockwell, Justin Johnson, and David F Fouhey. The 8-point algorithm as an inductive bias for relative pose prediction by vits. In *2022 International Conference on 3D Vision (3DV)*, pages 1–11. IEEE, 2022. 2
- [43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 12
- [44] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III* 18, pages 234–241. Springer, 2015. 5
- [45] Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs. *arXiv preprint arXiv:2408.13912*, 2024. 11
- [46] Cameron Smith, Yilun Du, Ayush Tewari, and Vincent Sitzmann. Flowcam: training generalizable 3d radiance fields without camera poses via pixel-aligned scene flow. *arXiv preprint arXiv:2306.00180*, 2023. 1, 2, 5, 6, 7, 8, 9, 10
- [47] Libo Sun, Jia-Wang Bian, Huangying Zhan, Wei Yin, Ian Reid, and Chunhua Shen. Sc-depthv3: Robust self-supervised monocular depth estimation for dynamic scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2023. 9
- [48] Libo Sun, Jia-Wang Bian, Huangying Zhan, Wei Yin, Ian Reid, and Chunhua Shen. Sc-depthv3: Robust self-supervised monocular depth estimation for dynamic scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2
- [49] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative pretraining in multi-modality. *arXiv preprint arXiv:2307.05222*, 2023. 12
- [50] Stanislaw Szymanicowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10208–10217, 2024. 1, 2, 3, 9
- [51] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II* 16, pages 402–419. Springer, 2020. 2
- [52] Hans Thisanke, Chamli Deshan, Kavindu Chamith, Sachith Seneviratne, Rajith Vidanaarachchi, and Damayanthi

- Herath. Semantic segmentation using vision transformers: A survey. *Engineering Applications of Artificial Intelligence*, 126:106669, 2023. 4
- [53] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022. 2
- [54] Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. Pf-irm: Pose-free large reconstruction model for joint pose and shape prediction. *arXiv preprint arXiv:2311.12024*, 2023. 4
- [55] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2021. 5
- [56] Shuzhe Wang et al. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 3, 11
- [57] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. 3, 6
- [58] Philippe Weinzaepfel, Vincent Leroy, Thomas Lucas, Romain Brégier, Yohann Cabon, Vaibhav Arora, Leonid Antsfeld, Boris Chidlovskii, Gabriela Csurka, and Jérôme Revaud. Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion. *Advances in Neural Information Processing Systems*, 35:3502–3516, 2022. 2, 4, 9, 11
- [59] Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. Croco v2: Improved cross-view completion pre-training for stereo matching and optical flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17969–17980, 2023. 2, 4, 6
- [60] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7467–7477, 2020. 1
- [61] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 9
- [62] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. *arXiv preprint arXiv:2410.13862*, 2024. 4, 9
- [63] Lining Xu, Vasu Agrawal, William Laney, Tony Garcia, Ayush Bansal, Changil Kim, Samuel Rota Bulò, Lorenzo Porzi, Peter Kortschieder, Aljaž Božič, et al. Vr-nerf: High-fidelity virtualized walkable spaces. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–12, 2023. 1
- [64] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024. 4, 9
- [65] 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. 4
- [66] Botao Ye, Sifei Liu, Haofei Xu, Li Xuetong, Marc Pollefeys, Ming-Hsuan Yang, and Peng Songyou. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. *arXiv preprint arXiv:2410.24207*, 2024. 2
- [67] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, et al. gsplat: An open-source library for gaussian splatting. *arXiv preprint arXiv:2409.06765*, 2024. 6
- [68] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1983–1992, 2018. 3
- [69] 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. 1, 9
- [70] Chuanrui Zhang, Yingshuang Zou, Zhuoling Li, Minmin Yi, and Haoqian Wang. Transplat: Generalizable 3d gaussian splatting from sparse multi-view images with transformers. *arXiv preprint arXiv:2408.13770*, 2024. 4, 5, 9
- [71] 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. 6
- [72] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15838–15847, 2021. 1
- [73] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024. 1
- [74] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017. 3
- [75] 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. 2, 5, 9, 11