MV-DUSt3R+: Single-Stage Scene Reconstruction from Sparse Views In 2 Seconds

Zhenggang Tang,[1,2] Yuchen Fan,[1] Dilin Wang,[1] Hongyu Xu,[1] Rakesh Ranjan,[1]
Alexander Schwing,[2] Zhicheng Yan[1]
[1]Meta Reality Labs: {zgtang, zyan3}@meta.com
[2]University of Illinois Urbana-Champaign: {zt15, aschwing}@illinois.edu

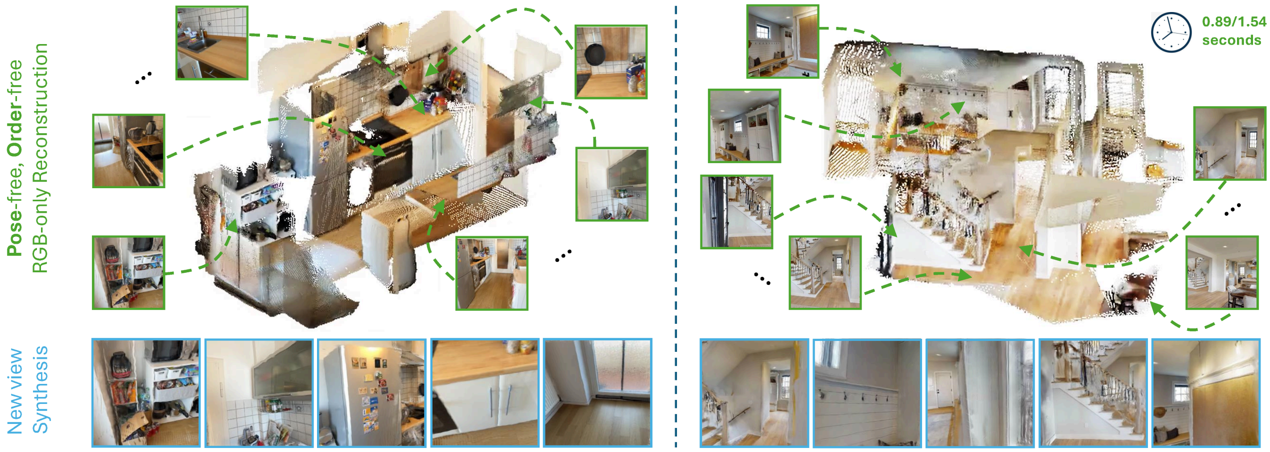

Figure 1. The proposed Multi-View Dense Unconstrained Stereo 3D Reconstruction Prime (MV-DUSt3R+) is able to reconstructs large scenes from multiple pose-free RGB views. **Top row**: one single-room scene and one large multi-room scene reconstructed by MV-DUSt3R+ in 0.89 and 1.54 seconds using 12 and 20 input views respectively (only a subset is shown for visualization). **Bottom row**: MV-DUSt3R+ is able to synthesize novel views by predicting pixel-aligned Gaussian parameters. Reconstruction of such large scenes are challenging for prior methods (*e.g.*, DUSt3R [68]). See Fig. 6 and appendix for more results with comparison.

Abstract

Recent sparse multi-view scene reconstruction advances like DUSt3R and MASt3R no longer require camera calibration and camera pose estimation. However, they only process a pair of views at a time to infer pixel-aligned pointmaps. When dealing with more than two views, a combinatorial number of error prone pairwise reconstructions are usually followed by an expensive global optimization, which often fails to rectify the pairwise reconstruction errors. To handle more views, reduce errors, and improve inference time, we propose the fast single-stage feed-forward network MV-DUSt3R. At its core are multi-view decoder blocks which exchange information across any number of views while considering one reference view. To make our method robust to reference view selection, we further propose MV-DUSt3R+, which employs cross-reference-view blocks to fuse information across different reference view choices. To further enable novel view synthesis, we extend both by adding and jointly training Gaussian splatting heads. Experiments on multi-view stereo reconstruction, multi-view pose estimation, and novel view synthesis confirm that our methods improve significantly upon prior art. Code released.[1]

1. Introduction

Multi-view scene reconstruction as shown in Fig. 1 has been a fundamental task in 3D computer vision for decades [27]. It is widely applicable in mixed reality [2], city reconstruction [62], autonomous driving simulation [33, 75], robotics [87] and archaeology [46]. Classic methods decompose multi-view scene reconstruction into sub-tasks, including camera calibration [7, 90], pose estimation [8, 48], feature detection and matching [40, 50, 84], structure from motion (SfM) [53], bundle adjustment [9, 61], *etc*., and assemble individual components into a pipeline. Recent approaches adopt a learning-based paradigm for those subtasks [23], explore different neural scene representations (*e.g*., Neural Signed Distance Functions [23, 41, 45], Neural Radiance Fields [11, 42], Gaussian Splatting [30, 32]), and build more end-to-end pipelines to reconstruct objects [60, 77] and scenes [13, 15, 88]. While these ap-

[1] https://mv-dust3rp.github.io/

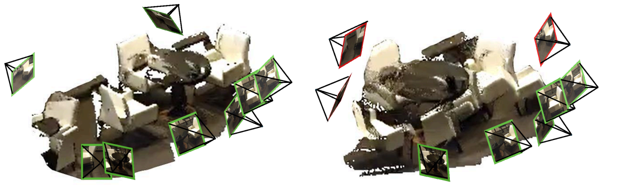

Figure 2. **Left**: Groundtruth scene with 8 views: Three chairs surrounding one table and one more chair next to another table. **Right**: reconstruction and pose estimation of DUSt3R with global optimization: all chairs incorrectly surround one table. Wrong poses are marked in red.

proaches have enjoyed quite some success, they often require prior knowledge or nontrivial pre-processing to obtain camera parameters and poses.

More recently, novel multi-view scene reconstruction approaches, such as DUSt3R [68] and MASt3R [36], directly process an unordered set of unposed rgb views, *i.e*., camera intrinsics and poses are unknown. These methods process two views, *i.e*., a chosen reference view and another source view, at a time, and directly infer pixel-aligned 3D pointmaps in the reference view's camera coordinate system. To handle a larger set of input views, a combinatorial number of pairwise inferences is followed by a second stage of global optimization to align the local pairwise reconstructions into a single global coordinate system.

While evaluation results of these methods are promising on object-centric DTU data [1], we point out inefficiencies in reconstructing scenes, as stereo cues in 2-view input could be ambiguous. Further, despite plausible reconstructions of individual 2-view inputs, conflicts often arise when aligning them in a global coordinate system. Such conflicts can be challenging to resolve by a global optimization, which only rotates pairwise predictions but doesn't rectify wrong pairwise matches. As a consequence, scene reconstructions exhibit misaligned pointmaps. One example is shown in Fig. 2.

To address the aforementioned issues, we propose the single-stage network **Multi-View Dense Unconstrained Stereo 3D Reconstruction** (**MV-DUSt3R**), which jointly processes a large number of input views in one feed-forward pass, and completely removes the cascaded global optimization used in prior arts. To achieve this, we employ multi-view decoder blocks, which jointly learn not only all pairwise relationships between a chosen reference view and all the other source views, but also appropriately address pairwise relationships among all source views. Moreover, our training recipe encourages the predicted per-view pointmaps to adhere to the same reference camera coordinate system, which waives the need for a subsequent global optimization.

When reconstructing a large scene from sparse multi-view images, the stereo cues between the one selected reference view and certain source views could be insufficient. This is because significant changes in camera poses make it difficult to directly infer the relation between the reference view and those source views. Therefore, long-range information propagation is required for those source views. To handle this efficiently, we further present **MV-DUSt3R+**. It operates on *a set of reference views* and employs Cross-Reference-View attention blocks for effective long-range information propagation. See Fig. 1 for its reconstructions.

On the three benchmark scene-level datasets HM3D [49], ScanNet [17], and MP3D [12], we demonstrate that MV-DUSt3R achieves significantly better results on the tasks of Multi-View Stereo (MVS) reconstruction and Multi-View Pose Estimation (MVPE) while being $48 \sim 78\times$ faster than DUSt3R. In addition, MV-DUSt3R+ is able to improve the reconstruction quality especially on harder settings, while still inferring one order of magnitude faster than DUSt3R.

To extend both of our methods towards Novel View Synthesis (NVS), we further attach lightweight prediction heads which regress to 3D Gaussian attributes. The predicted per-view Gaussian primitives are transformed into the coordinate of a target view before splatting-based rendering [32]. Using this, we also show that our models outperform DUSt3R with heuristically designed 3D Gaussian parameters under the standard photometric evaluation protocol. The gains can be attributed to the more accurate predictions of the Gaussian locations by our method. We summarize our contributions as follows:

- We present **MV-DUSt3R**, a novel feed-forward network for pose-free scene reconstruction from sparse multi-view input. It not only runs $48 \sim 78\times$ faster than DUSt3R for $4 \sim 24$ views, but also reduces Chamfer distance on 3 challenging evaluation datasets HM3D [49], ScanNet [17], and MP3D [12] by $2.8\times$, $2\times$ and $1.6\times$ for smaller scenes of average size 2.2, 7.5, 19.3 (m^2) with 4-view input, and $3.2\times$, $1.9\times$ and $2.1\times$ for larger scenes of average size 3.3, 17.9, 37.3 (m^2) with 24-view input.
- We present **MV-DUSt3R+**, which improves MV-DUSt3R by using multiple reference views, addressing the challenges which occur when inferring relations between all input views via a single reference view. We validate, MV-DUSt3R+ performs well across all tasks, number of views, and on all three datasets. For example, for MVS reconstruction, it further reduces Chamfer distance on 3 datasets by $2.6\times$, $1.6\times$, $1.8\times$ for large scenes with 24-view input, while still running $14\times$ faster than DUSt3R.
- We extend both networks to support NVS by adding Gaussian splatting heads [82] to predict per-pixel Gaussian attributes. With joint training of all layers using both reconstruction loss and view rendering loss, we demonstrate that the model outperforms a DUSt3R-based baseline significantly.

2. Related Work

Structure-from-Motion (SfM). SfM methods reconstruct sparse scene geometry from a set of images and estimate individual camera poses. For this, SfM is often addressed in a few independent steps, including detecting/describing/matching local features across multiple views (*e.g.*, SIFT [40], ORB [50], LIFT [84]), triangulating features to estimate sparse 3D geometry and camera poses (*e.g.*, COLMAP [53]), applying bundle adjustment over many views (see Triggs et al. [61] for an overview), *etc*. Though steady progress has been made in the past decades [44], and a large number of applications have been enabled [10, 31, 71], the classic SfM pipeline solves sub-tasks individually and sequentially, accumulating errors. More recent SfM methods improve traditional pipeline with learnable components [28, 65]. MASt3R-SfM [20] extends MASt3R [36], which only produces local reconstructions for 2-view input, to perform global optimization for aligning local reconstructions via gradient descent to minimize 3D matching loss.

Multi-View Stereo. MVS reconstructs dense 3D scene geometry from multiple views [24], often in the form of 3D points. In the classic PatchMatch-based framework [91], per-pixel depth in the reference image is estimated from a set of unstructured source images via patch matching under a homography transform [54]. Subsequent work has substantially improved feature matching [63, 69, 92] and depth estimation [25, 52, 76]. More recent learning-based approaches [67, 78] often build an end-to-end pipeline, where deep models extract visual features, model cross-view correspondences (*e.g.*, cost volume [26]), and regress depth maps [79]. Note, with few exceptions [80], most approaches require prior knowledge of camera intrinsics from SfM or camera calibration. Our MV-DUSt3R network also processes sparse multi-view input, but does not require prior knowledge of camera parameters.

Neural Scene Reconstruction. Compared to classic methods, which reconstructs a scene using either explicit representations (*e.g.*, 3D point, mesh) or implicit representations (*e.g.*, signed distance function [43]), recent approaches adopt different neural representations [55], including Neural Distance Fields [16, 38], Neural Radiance Fields (NeRFs) [3, 4, 34, 42, 62], Gaussian Splatting [30, 32, 86], and their combination [93]. Many of them require slow per-scene optimization to attain accurate results, while more recent methods explore the use of feed-forward networks for generalizable reconstruction at a fraction of the time, including those for generating Neural Distance Functions [16, 45, 56], NeRFs [14, 19, 85], and Gaussians [13, 15, 59, 72]. Note, neural scene reconstruction often requires input views with known camera poses, albeit quite a few exceptions exist, such as CoPoNeRF [29], Splatt3R [57], and NoPoSplat [81]. For example, NoPoSplat predicts 3D Gaussians in the same camera coordinates, akin to the key idea of DUSt3R. However, those pose-free methods primarily focus on inference with 2 input views. It is not clear how they perform when processing sparse multi-view input. In contrast, our models MV-DUSt3R, MV-DUSt3R+ equipped with 3D Gaussian splatting heads, not only waive the need for camera pose, but also reconstruct large scenes from multiple views in a single feed-forward pass.

Dense Unconstrained Scene Reconstructions from Multi-View Input. To bypass estimation of camera parameters and poses, recent works like DUSt3R [68] and MASt3R [36] propose a new approach: directly regress pixel-aligned 3D pointmaps for pairs of input views. An expensive 2$^{\text{nd}}$ stage global optimization is required to align all pairwise reconstructions in the same coordinate system. Both DUSt3R and MASt3R are only evaluated on object-centric DTU data [1] where all views are concentrated in a small region. Notably, methods are not validated if their 2-stage pipeline excels at reconstructing larger scenes captured with sparse multi-view input. Subsequently, Spann3R [64] augments DUSt3R with a spatial memory to process an ordered set of images. Although capable of performing online scene reconstruction for object-centric scenes, for larger scenes, Spann3R is more likely to drift, generating a misaligned reconstruction due to the limited size of the spatial memory and the lack of globally aligning reconstructions. In contrast, our MV-DUSt3R+ performs offline scene reconstruction by processing all input views (up to 24 in our experiments) at once. Different from DUSt3R, it does not require global optimization because the predicted per-view pointmaps are already globally aligned.

Generative models for 3D reconstruction. Reconstructing scenes from a small number of views is challenging, in particular for unseen areas. Recent advances such as InFusion [39], ZeroNVS [51], Reconfusion [73], and ReconX [37] exploit priors encoded in image and video generative models [5, 6, 58]. We leave benefit from image priors of diffusion models as future work.

3. Method

Our goal is to densely reconstruct a scene given a sparse set of rgb images with unknown camera intrinsics and poses. Following DUSt3R, our model predicts 3D pointmaps aligned with 2D pixels for each view. Different from DUSt3R, our model jointly predicts 3D pointmaps for any number of input views in a single forward pass. Formally, given N input image views of a scene $\{I^v\}_{v=1}^N$, where $I^v \in \mathbb{R}^{H\times W\times 3}$, from which we select one reference $r \in \{1,\dots,N\}$, our goal is to predict per-view 3D pointmaps $\{X^{v,r}\}_{v=1}^N$. Note, the 3D pointmap $X^{v,r} \in \mathbb{R}^{H\times W\times 3}$ denotes the coordinates of 3D points for image I^v in the camera coordinate system of the reference view r.

In Sec. 3.1, we introduce our Multi-View Dense Unconstrained Stereo 3D Reconstruction (**MV-DUSt3R**) network to efficiently processes all input views in one pass and without subsequent global optimization, while considering a single chosen reference view. In Sec. 3.2, we present **MV-DUSt3R+**, which processes all input views while consid-

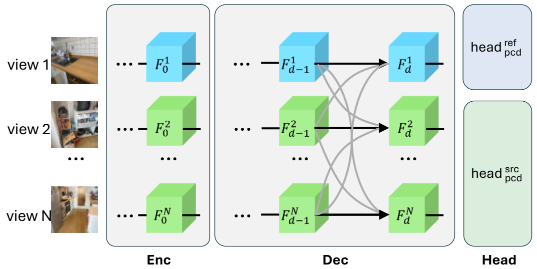

Figure 3. Overview of MV-DUSt3R. Visual tokens for the reference view and other source views are shown in Blue and Green. Black straight solid lines indicate the primary token flow while gray lines indicate secondary token flow.

ering multiple reference views. Finally, to support novel view synthesis, in Sec. 3.3, we augment our networks with Gaussian heads to predict pixel-aligned 3D Gaussians.

3.1. MV-DUSt3R

A Multi-View Model Architecture. As shown in Fig. 3, MV-DUSt3R consists of an encoder to transform images into visual tokens, decoder blocks to fuse tokens across views, and regression heads to predict per-view 3D pointmaps aligned with 2D pixels. Different from DUSt3R, our network uses decoder blocks to fuse tokens across *all* views rather than independently fusing only tokens for two views at a time. Concretely, a ViT [18] encoder with shared weights, denoted as $\texttt{Enc}$, is first applied on input views $\{I^v\}_{v=1}^N$ to compute initial visual tokens $\{F_0^v\}_{v=1}^N$, *i.e.*, $F_0^v = \texttt{Enc}(I^v)$. Note, the resolution of the encoder output features is $16\times$ smaller than the input image before being flattened into a sequence of tokens.

To fuse the tokens, two types of decoders are used, one for the chosen reference view and one for the remaining source views. They share the same architecture but their weights differ. Each decoder consists of D decoder blocks referred to as $\texttt{DecBlock}_d^{\text{ref}}$ and $\texttt{DecBlock}_d^{\text{src}}$ for $d \in \{1, \ldots, D\}$. Their difference is, $\texttt{DecBlock}_d^{\text{ref}}$ is dedicated to update reference view tokens F^r, while $\texttt{DecBlock}_d^{\text{src}}$ updates tokens $\{F^v\}_{v \neq r}$ from *all other* source views. Each decoder block takes as input a set of primary tokens from one view, and a set of secondary tokens from other views. In each block, a self-attention layer is applied to primary tokens only, and a cross-attention layer fuses primary tokens with secondary tokens before a final MLP is applied on the primary tokens. Layer norm is also applied before both attentions and the MLP. Using those, the decoder computes the final token representations F_D^v via

$$F_d^v = \begin{cases} \texttt{DecBlock}_d^{\text{ref}}(F_{d-1}^v, \mathcal{F}_{d-1}^{-v}) \text{ if } v = r, \\ \texttt{DecBlock}_d^{\text{src}}(F_{d-1}^v, \mathcal{F}_{d-1}^{-v}) \text{ otherwise.} \end{cases} \tag{1}$$

Here, the secondary tokens $\mathcal{F}_d^{-v} = \{F_d^1, \ldots, F_d^{v-1}, F_d^{v+1}, \ldots, F_d^N\}$ subsume tokens from all views other than the view of the primary tokens F_d^v.

To finally predict the per-view 3D pointmaps, we use two heads: $\texttt{Head}_{\text{pcd}}^{\text{ref}}$ for the reference view and $\texttt{Head}_{\text{pcd}}^{\text{src}}$ for all other views. They share the same architecture but use different weights. Each consists of a linear projection layer and a pixel shuffle layer with an upscale factor of 16 to restore the original input image resolution. As in DUSt3R, the head predicts 3D pointmaps $X^{v,r} \in \mathbb{R}^{H \times W \times 3}$ and confidence maps $C^{v,r} \in \mathbb{R}^{H \times W}$ via

$$X^{v,r}, C^{v,r} = \begin{cases} \texttt{Head}_{\text{pcd}}^{\text{ref}}(F_D^v) \text{ if } v = r, \\ \texttt{Head}_{\text{pcd}}^{\text{src}}(F_D^v) \text{ otherwise.} \end{cases} \tag{2}$$

Note that DUSt3R is a special case of MV-DUSt3R if the number of views $N = 2$. However, for multiple input views, MV-DUSt3R will update primary tokens using a much larger set of secondary tokens. Hence, it is able to benefit from many more views. Importantly, as our architecture components and structure only differ slightly from those in DUSt3R (additional skip connection and conv net), we have only marginally more trainable parameters. Since, the number of parameters in MV-DUSt3R is almost identical to DUSt3R, MV-DUSt3R can beneficially be initialized using pre-trained DUSt3R weights.

Training Recipe. Inspired by DUSt3R, we use a confidence-aware pointmap regression loss $\mathcal{L}_{\text{conf}}$, *i.e.*,

$$\mathcal{L}_{\text{conf}} = \sum_{v \in \{1,\ldots,N\}} \sum_{p \in P^v} C_p^{v,r} \ell_{\text{regr}}(v, p) - \beta \log C_p^{v,r}, \tag{3a}$$

$$\text{where} \quad \ell_{\text{regr}}(v, p) = \left\| \frac{1}{z} X_p^{v,r} - \frac{1}{\bar{z}} \bar{X}_p^{v,r} \right\|. \tag{3b}$$

Here, P_v denotes the set of valid pixels in view v where groundtruth 3D points are well defined. β controls the weight of the regularization term. The pointmap regression loss ℓ_{regr} measures the difference between predicted and groundtruth 3D points after normalization, which is needed to resolve the scale ambiguity between prediction and groundtruth. It uses $\bar{X}_p^{v,r}$, the groundtruth 3D point of pixel p of view v in the reference view r. The scale normalization factor $z = \text{norm}(\mathcal{X}^{\{v\},r})$ and $\bar{z} = \text{norm}(\bar{\mathcal{X}}^{\{v\},r})$ are computed as the average distance of valid 3D points to the coordinate origin in all views, for prediction and groundtruth, respectively.

3.2. MV-DUSt3R+

As shown in Fig. 4, for different reference view choices, the quality of the scene reconstructed by MV-DUSt3R varies spatially. The predicted pointmap for an input source view tends to be better when the viewpoint change to the reference view is small, and deteriorates as the viewpoint change increases. However, to reconstruct a large scene with a sparse set of input views, a single reference view with only moderate viewpoint changes to all other source views is unlikely to exist. Therefore, it is difficult to reconstruct scene geometry

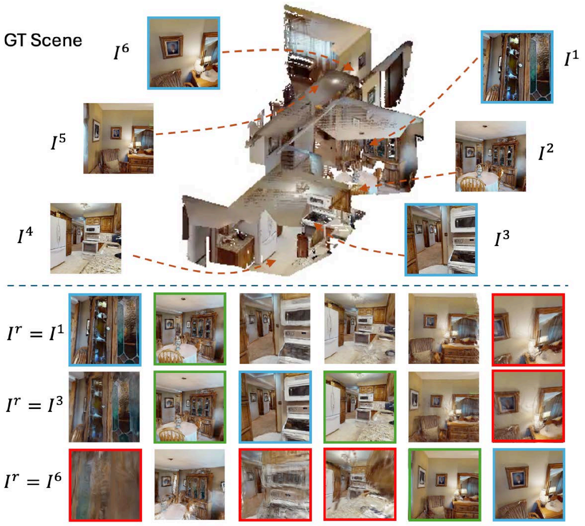

Figure 4. **Top**: A multi-room scene: 16 views are sampled as input to MV-DUSt3R. For clarity, only 6 are shown. 3 of them are reference view candidates, highlighted in blue. **Bottom**: In each row, we select a different reference view and render the reconstructed scene from 6 input views. Renderings in good and poor quality are highlighted in green and red. As the viewpoint change between the input view and the reference view increases, quality of the reconstructed scene geometry in that input view decreases.

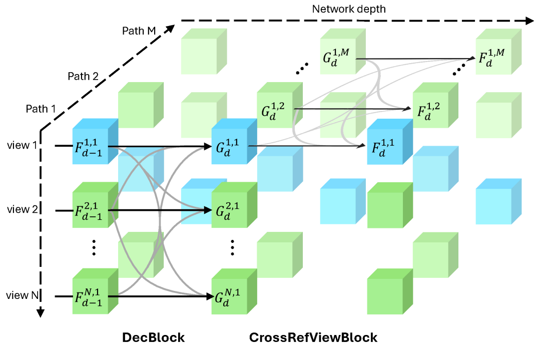

Figure 5. `DecBlock` and `CrossRefViewBlock` in MV-DUSt3R+: tokens of the reference and other views are highlighted in blue and green, respectively. Each model path uses a different reference view. For clarity, only 1 of stacked `DecBlock` and `CrossRefViewBlock` are shown.

equally well everywhere with a single selected reference view. To address this, we propose **MV-DUSt3R+**, which selects multiple views as the reference view, and jointly predicts pointmaps for all input views in the camera coordinate of each selected reference view. We hypothesize: while pointmaps of certain input views are difficult to predict for one reference view, they are easier to predict for a different reference view (*e.g.*, smaller viewpoint change, more salient matching patterns). To holistically improve the pointmap prediction of all input views, we include a novel Cross-Reference-View block into MV-DUSt3R+.

A Multi-Path Model Architecture. Let $R = \{r^m\}_{m=1}^{M}$ denote a set of M reference views randomly chosen from an unordered set of input views. We adopt the same decoder blocks from MV-DUSt3R, deploy them in a multi-path model architecture (see Fig. 5), and use them to compute a reference-view dependent intermediate representation $G_d^{v,m}$ at decoder layer d for input view v *and reference view* r^m:

$$G_d^{v,m} = \begin{cases} \texttt{DecBlock}_d^{\text{ref}}(F_{d-1}^{v,m}, \mathcal{F}_{d-1}^{-v,m}) \text{ if } v = r^m, & (4) \\ \texttt{DecBlock}_d^{\text{src}}(F_{d-1}^{v,m}, \mathcal{F}_{d-1}^{-v,m}) \text{ otherwise}, & (5) \end{cases}$$

$$F_d^{v,m} = \texttt{CrossRefViewBlock}_d(G_d^{v,m}, \mathcal{G}_d^{v,-m}). \quad (6)$$

Here, $\mathcal{F}_{d-1}^{-v,m} = \{F_{d-1}^{1,m}, \ldots, F_{d-1}^{v-1,m}, F_{d-1}^{v+1,m}, \ldots, F_{d-1}^{N,m}\}$. As shown in Fig. 5, we fuse and update per-view tokens computed under different reference views by adding a *Cross-Reference-View* block after each decoder block (Eq. (6)), where $\mathcal{G}_d^{v,-m} = \{G_d^{v,1}, \ldots, G_d^{v,m-1}, G_d^{v,m+1}, \ldots, G_d^{v,M}\}$. Following Eq. (2), we compute per-view pointmaps $X^{v,m}$ and confidence map $C^{v,m}$ under each reference view r^m.

Training Recipe. Compared with MV-DUSt3R, MV-DUSt3R+ only adds a small number of additonal trainable parameters via the Cross-Reference-View blocks (see appendix). During training, a random subset of M input views are selected as the reference views. We average the pointmap regression losses in Eq. (3a) for all reference views.

Model Inference. At inference time, we uniformly select a subset of M input views as the reference views, while the 1st input view is always selected. A model with M paths is used but the final per-view pointmap predictions are computed using the heads in the 1st path.

3.3. MV-DUSt3R(+) for Novel View Synthesis

Next we extend our networks to support NVS with Gaussian primitives [32]. For clarity, below we use MV-DUSt3R+ as an example. MV-DUSt3R can be extended similarly.

Gaussian Head. We add a separate set of heads to predict per-pixel Gaussian parameters, including scaling factor $S^{v,m} \in \mathbb{R}^{H \times W \times 3}$, rotation quaternion $q^{v,m} \in \mathbb{R}^{H \times W \times 4}$, and opacity $\alpha^{v,m} \in \mathbb{R}^{H \times W}$. We add Gaussian heads $\texttt{Head}_{\text{3DGS}}^{\text{ref}}$ and $\texttt{Head}_{\text{3DGS}}^{\text{src}}$ for reference and other views:

$$S^{v,m}, q^{v,m}, \alpha^{v,m} = \begin{cases} \texttt{Head}_{\text{3DGS}}^{\text{ref}}(F_D^{v,m}) \text{ if } v = r^m, & (7) \\ \texttt{Head}_{\text{3DGS}}^{\text{src}}(F_D^{v,m}) \text{ otherwise}. & (8) \end{cases}$$

For other Gaussian parameters, we use the predicted pointmap $X^{v,m}$ as the center, the pixel color I^v as the color and fix the spherical harmonics degree to be 0.

Training Recipe. During training, for a chosen reference view r^m, we perform differentiable splatting-based rendering [32] to generate rendering predictions for both input views and novel views. Following prior approaches [13, 15, 59], we use a weighted sum of L^2 pixel difference loss and

Dataset	Eval setting	Scene type
HM3D	Supervised	multi-room
ScanNet	Supervised	single-room
MP3D	Zero-shot	multi-room & outdoor

Table 1. **Evaluation datasets comparison**.

perceptual similarity loss LPIPS as the rendering loss $\mathcal{L}_{\text{render}}$ to train the Gaussian heads. The final training loss includes both $\mathcal{L}_{\text{conf}}$ and $\mathcal{L}_{\text{render}}$ (for details see appendix).

4. Experiments

4.1. Datasets

Our training data includes ScanNet [17], ScanNet++ [83], HM3D [49], and Gibson [74]. Note, all of them are also used by DUSt3R. For evaluation, we use datasets MP3D [12], HM3D [49], and ScanNet [17]. While ScanNet scenes are often small single-room sized and with low diversity, scenes in MP3D and HM3D are often large multi-room sized and with high diversity. MP3D also contains outdoor scenes. See Tab. 1 to compare evaluation datasets. We use the same train/test split as DUSt3R, and our training data is a subset of DUSt3R's training data (for details see appendix).

Trajectory Generation. To generate a set of input views $\{I^v\}_{v=1}^N$ for $N > 2$, we first randomly select one frame and initialize the current scene point cloud using its data. Then we sequentially sample more candidate frames. We retain a candidate frame and add its corresponding point cloud to the current scene, if the overlap between the candidate frame's point cloud and the current scene point cloud is between a lower threshold t_{min} and an upper bound t_{max}.

Training Trajectories. To sample the training set trajectories, we employ two choices of thresholds: $(t_{\text{min}}, t_{\text{max}}) \in \{(30\%, 70\%), (30\%, 100\%)\}$. From ScanNet and ScanNet++, we sample 1K trajectories of 10 views per scene, and a total of 3.2M trajectories. On HM3D and Gibson, where the scene is often larger, we sample 6K trajectories per scene with 10 views each, and a total of 7.8M trajectories.

Test Trajectories. For the test set, we generate $1K$ trajectories per dataset. To support evaluation with a larger number of inputs views, we sample 30 views per trajectory.

4.2. Implementation Details

We process input views at resolution 224 × 224. We utilize 64 Nvidia H100 GPUs for the model training. To initialize, DUSt3R model weights are used. We use the first $N = 8$ views of each trajectory as input views, and randomly select 1 view as the reference view for MV-DUSt3R and $M = 4$ views for MV-DUSt3R+. We train for 100 epochs using 150K trajectories per epoch, which takes 180 hours. For MVS reconstruction evaluation, to assess the performance of each method in reconstructing scenes of variable sizes, we report results with input views ranging from 4 to 24 views. For NVS evaluation, we use the remaining 6 views as novel views. Below we report results on all evaluation datasets for all choices of the number of input views using only one MV-DUSt3R model and one MV-DUSt3R+ model.

	Method	GO	HM3D			ScanNet			MP3D			Time
			ND ↓	DAc ↑	CD ↓	ND ↓	DAc ↑	CD ↓	ND ↓	DAc ↑	CD ↓	(sec)
4 views	Spann3R	×	37.1	0.0	225(184)	8.9	19.5	54.7(50.1)	42.7	0.0	248(202)	0.36
	DUSt3R	✓	1.9	75.1	5.6(2.3)	1.3	89.8	4.0(0.4)	3.9	41.7	40.0(5.3)	2.42
	MV-DUSt3R	×	1.1	92.2	2.0(1.1)	1.0	93.3	2.0(0.4)	2.5	62.4	25.3(4.1)	0.05
	MV-DUSt3R+	×	1.0	95.2	1.5(0.9)	0.8	94.9	1.5(0.3)	2.2	68.0	19.9(3.4)	0.29
	MV-DUSt3R$_{\text{oracle}}$	×	1.0	94.6	1.5(0.7)	0.8	95.5	1.3(0.3)	2.3	66.6	20.7(4.0)	-
	MV-DUSt3R+$_{\text{oracle}}$	×	0.9	96.5	1.4(0.7)	0.7	95.8	1.2(0.2)	2.1	70.6	17.9(3.3)	-
12 views	Spann3R	×	32.6	0.0	125(113)	9.1	16.3	36.6(31.2)	35.0	0.0	138(112)	1.34
	DUSt3R	✓	3.9	30.7	18.1(3.4)	1.9	82.6	4.1(0.6)	6.6	12.0	49.6(8.3)	8.28
	MV-DUSt3R	×	1.6	79.5	3.0(1.2)	1.4	86.8	2.3(0.8)	3.4	41.3	22.6(5.5)	0.15
	MV-DUSt3R+	×	1.2	91.5	1.8(0.7)	1.2	88.4	1.8(0.7)	2.6	55.0	15.1(3.8)	0.89
	MV-DUSt3R$_{\text{oracle}}$	×	1.3	88.8	1.8(0.9)	1.0	90.6	1.3(0.7)	2.9	51.3	16.4(4.0)	-
	MV-DUSt3R+$_{\text{oracle}}$	×	1.1	94.8	1.4(0.7)	1.0	90.9	1.3(0.5)	2.5	59.8	13.6(3.5)	-
24 views	Spann3R	×	41.7	0.0	139(121)	11.4	1.6	37.4(35.5)	46.6	0.0	151(121)	2.73
	DUSt3R	✓	6.8	7.3	32.4(5.2)	2.4	72.6	5.1(1.0)	11.4	2.5	80.9(14.3)	27.21
	MV-DUSt3R	×	3.4	36.7	10.0(3.5)	2.2	75.2	2.7(0.9)	6.3	12.2	38.6(13.9)	0.35
	MV-DUSt3R+	×	2.1	64.5	3.9(2.0)	1.6	81.2	1.7(0.7)	4.3	26.7	22.0(5.9)	1.97
	MV-DUSt3R$_{\text{oracle}}$	×	2.1	58.9	3.5(2.1)	1.4	82.9	1.4(0.7)	4.4	22.0	19.9(5.1)	-
	MV-DUSt3R+$_{\text{oracle}}$	×	1.8	77.9	2.6(1.3)	1.3	85.1	1.3(0.6)	3.6	33.1	15.1(4.4)	-

Table 2. **MVS reconstruction results**. Best results are marked in red. For clarity, metric ND is scaled up by 10×, DAc is in percent %, and CD is scaled by 100× with its median given in parentheses. Results of our **oracle** methods are obtained by manually selecting the best single reference view from reference view candidates to report a possibly achievable performance. In the rightmost column, each method's time for scene reconstruction is reported.

4.3. Multi-View Stereo Reconstruction

Metrics. We report Chamfer Distance (CD), as in prior work [1, 68], as well as 2 additional metrics. **Normalized-Distance (ND)**: ℓ_{regr} with zero-centering to make it scale and translation invariant; **DistanceAccu@0.2 (DAc)**: proportion of pixels where the corresponding normalized distance between prediction and groundtruth in pointmap is ≤ 0.2.

Baselines. We compare with baselines that reconstruct scenes from input rgb views without knowing camera intrinsics and poses. We evaluate the DUSt3R model trained at input resolution 224 × 224 with Global Optimization (GO), and also the Spann3R [64] model on Github.

Results are shown in Tab. 2. In the supervised setting, we compare DUSt3R and our methods on HM3D and ScanNet. On HM3D (multi-room scenes), **MV-DUSt3R** consistently outperforms DUSt3R, as the scene size increases and more input views are sampled (from 4 to 24 views). For example, MV-DUSt3R reduces ND by 1.7× and increases DAc by 1.2× for 4-view input. For 24-view input, MV-DUSt3R improves ND by 2× and DAc by 5.3×. This confirms: as more input views are available, single-stage MV-DUSt3R exploits multi-view cues to infer 3D scene geometry better than DUSt3R, which only exploits pairwise stereo cues at a time. Furthermore, **MV-DUSt3R+** substantially improves upon MV-DUSt3R, especially when the scene size is large and many more input views are used. With 12-view input, MV-DUSt3R+ improves ND by 1.3× and DAc by 1.2×. With 24-view input, the improvements are more significant, with a 1.6× lower ND and a 1.8× higher DAc. The multi-path architecture enables MV-DUSt3R+ to more effectively fuse multi-view cues across different choices of reference

	Method	GO	HM3D			ScanNet			MP3D		
			RRE ↓	RTE ↓	mAE ↓	RRE ↓	RTE ↓	mAE ↓	RRE ↓	RTE ↓	mAE ↓
4 views	DUSt3R	✓	2.4	3.1	12.5	3.0	20.0	30.7	3.5	3.8	13.3
	MV-DUSt3R	×	1.5	1.5	5.5	2.3	16.8	27.0	1.2	1.0	5.4
	MV-DUSt3R+	×	1.2	1.1	4.9	1.4	16.1	26.2	0.8	0.8	4.6
	MV-DUSt3R$_{\text{oracle}}$	×	0.0	0.1	2.8	0.9	7.0	18.9	0.1	0.1	2.9
	MV-DUSt3R+$_{\text{oracle}}$	×	0.0	0.0	2.4	0.9	6.8	18.7	0.1	0.0	2.4
12 views	DUSt3R	✓	3.7	8.3	20.1	4.6	22.6	34.2	4.5	8.4	19.8
	MV-DUSt3R	×	1.5	2.6	8.4	3.7	14.7	26.1	1.6	2.6	8.2
	MV-DUSt3R+	×	0.6	1.2	5.2	2.5	11.6	22.9	0.5	1.0	4.9
	MV-DUSt3R$_{\text{oracle}}$	×	0.4	0.6	4.9	1.7	7.8	20.2	0.6	0.7	5.1
	MV-DUSt3R+$_{\text{oracle}}$	×	0.3	0.3	3.4	1.7	6.0	17.9	0.3	0.3	3.3
24 views	DUSt3R	✓	8.8	18.1	30.9	8.1	26.6	38.9	10.0	18.2	30.5
	MV-DUSt3R	×	8.9	12.8	23.7	8.2	21.9	34.2	8.2	11.1	21.4
	MV-DUSt3R+	×	3.0	6.5	15.8	4.6	16.7	29.4	3.3	6.0	14.6
	MV-DUSt3R$_{\text{oracle}}$	×	3.2	4.4	14.7	3.4	13.1	26.7	3.4	4.2	14.0
	MV-DUSt3R+$_{\text{oracle}}$	×	1.4	2.4	11.1	2.6	9.9	23.7	1.8	2.4	10.6

Table 3. **Multi-View Pose Estimation results**. Metrics are reported in percent %.

frames, which holistically improves the scene geometry reconstruction in all input views. For a qualitative comparisons, see Fig. 6. For zero-shot evaluation on the challenging MP3D data, all methods have worse results. However, both of our methods consistently improve upon DUSt3R across different numbers of input views.

In all settings, Spann3R performs worse than all other methods, and often fails to reconstruct a scene from sparse views, a setting which is more challenging than the sequential video frames used for evaluation in the original paper.

4.4. Multi-View Pose Estimation

For both baseline and our methods, we estimate the relative camera pose for all pairs of input views from a given set of input views. We use the Weiszfeld algorithm [47] to estimate camera intrinsics, and RANSAC [22] with PnP [35] to estimate camera pose (see appendix for more details).

Baselines. We compare with other pose-free methods including DUSt3R and PoseDiffusion [66], a recent diffusion based method for camera pose estimation.

Metrics. Prior methods [68] report Relative Rotation Accuracy (**RRA@15**), Relative Translation Accuracy (**RTA@15**) under threshold 15 degrees, and mean Average Accuracy (**mAA@30**) under threshold 30 degrees. For clarity, we report Relative Rotation Error (**RRE@15** = 1.0 − RRA@15), Relative Translation Error (**RTE@15** = 1.0 − RTA@15) and mean Average Error (**mAE@30** = 1.0 − mAA@30).

Results. The comparisons with DUSt3R are presented in Tab. 3, while comparisons with PoseDiffusion are included in the appendix, as we find PoseDiffusion significantly underperforms other methods on our evaluation datasets. As shown in Tab. 3, in the supervised setting on HM3D, our method MV-DUSt3R achieves a 2.3× lower mAE for 4-view input, and a 1.3× lower mAE for 24-view input, compared with DUSt3R. MV-DUSt3R+ performs best, achieving a 2.6× lower mAE for 4-view input, and a 2.0× lower mAE for 24-view input, compared with DUSt3R. For another supervised setting ScanNet and the zero-shot MP3D, our method MV-DUSt3R+ performs best, while MV-DUSt3R consistently outperforms DUSt3R under all circumstances.

	Method	GO	HM3D			ScanNet			MP3D		
			PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
4 views	DUSt3R	✓	16.0	5.0	3.7	17.0	6.0	3.0	15.5	4.6	4.0
	MV-DUSt3R	×	19.9	6.0	2.0	21.9	7.1	1.6	19.6	5.8	2.1
	MV-DUSt3R+	×	20.2	6.1	1.9	22.2	7.1	1.5	19.9	5.9	2.0
	MV-DUSt3R$_{\text{oracle}}$	×	21.0	6.5	1.6	22.8	7.4	1.4	20.6	6.2	1.8
	MV-DUSt3R+$_{\text{oracle}}$	×	21.4	6.6	1.5	23.0	7.4	1.4	21.0	6.3	1.7
12 views	DUSt3R	✓	15.1	4.4	4.9	16.3	5.4	3.6	14.7	4.0	5.3
	MV-DUSt3R	×	18.9	5.6	2.7	20.1	6.5	2.2	18.4	5.3	2.8
	MV-DUSt3R+	×	19.4	5.8	2.4	20.4	6.6	2.1	19.0	5.5	2.6
	MV-DUSt3R$_{\text{oracle}}$	×	19.9	5.9	2.2	21.0	6.8	1.9	19.3	5.6	2.4
	MV-DUSt3R+$_{\text{oracle}}$	×	20.4	6.1	2.0	21.3	6.8	1.8	19.9	5.8	2.2
24 views	DUSt3R	✓	14.3	4.2	5.6	15.2	5.0	4.2	13.8	3.6	6.1
	MV-DUSt3R	×	17.8	5.3	3.6	18.4	6.0	2.9	17.3	4.9	3.8
	MV-DUSt3R+	×	18.4	5.4	3.2	18.6	6.0	2.8	17.9	5.1	3.5
	MV-DUSt3R$_{\text{oracle}}$	×	18.5	5.5	3.1	19.3	6.2	2.5	18.0	5.1	3.3
	MV-DUSt3R+$_{\text{oracle}}$	×	19.0	5.6	2.9	19.4	6.2	2.5	18.5	5.3	3.1

Table 4. **Novel View Synthesis results**. For clarity, we scale up metrics SSIM and LPIPS by 10×.

4.5. Novel View Synthesis

Metrics. Following prior works [13, 21, 57], we report Peak Signal-to-Noise Ratio (**PSNR**), Structural Similarity Index Measure (**SSIM**) [70], and Learned Perceptual Image Patch Similarity (**LPIPS**) [89].

Baseline. We compare with a DUSt3R-based baseline, which generates per-pixel Gaussian parameters as follows. We use the pointmap predicted by DUSt3R as the Gaussian center, use pixel RGB color I^v as the color, a constant 0.001 for the scale factor $S^{v,m}$, an identity transform, 1.0 for opacity, and spherical harmonics with zero-degree. See appendix for more details on rendering.

Results. As shown in Tab. 4, MV-DUSt3R improves upon the DUSt3R baseline across all evaluation datasets under all choices of input views. The improvements are also confirmed qualitatively in Fig. 6: the novel views synthesized by MV-DUSt3R better infer 3D geometry of objects and background (*e.g.*, walls, ceiling). MV-DUSt3R+ further improves in challenging situations, such as a scene with multiple close-by objects of similar appearance (*e.g.*, chairs). As an example, consider the ScanNet scene with 20 views in Fig. 6. Using multiple reference views, and fusing features computed in different model paths help resolve ambiguity in inferring the spatial relations between input views.

4.6. Scene Reconstruction Time

We compare the time of MVS reconstruction in Tab. 2. Our single-stage feed-forward networks entirely run on a GPU, without Global Optimization (GO). Compared with DUSt3R, our MV-DUSt3R runs 48× to 78× faster than DUSt3R, while the more performant MV-DUSt3R+ runs 8× to 14× faster, when considering 4 to 24 input views. *MV-DUSt3R+ reconstructs 24-view input for scenes of average size* 17.9 m^2 *on HM3D and* 37.3 m^2 *on MP3D in less than 2 seconds*.

4.7. Ablation Studies

Number of input views at training time. We compare 1-stage and 2-stage trained MV-DUSt3R+ on the HM3D evaluation set. For 1-stage training, we choose the first 4 or 8 views of the trajectory. For 2-stage training, we finetune

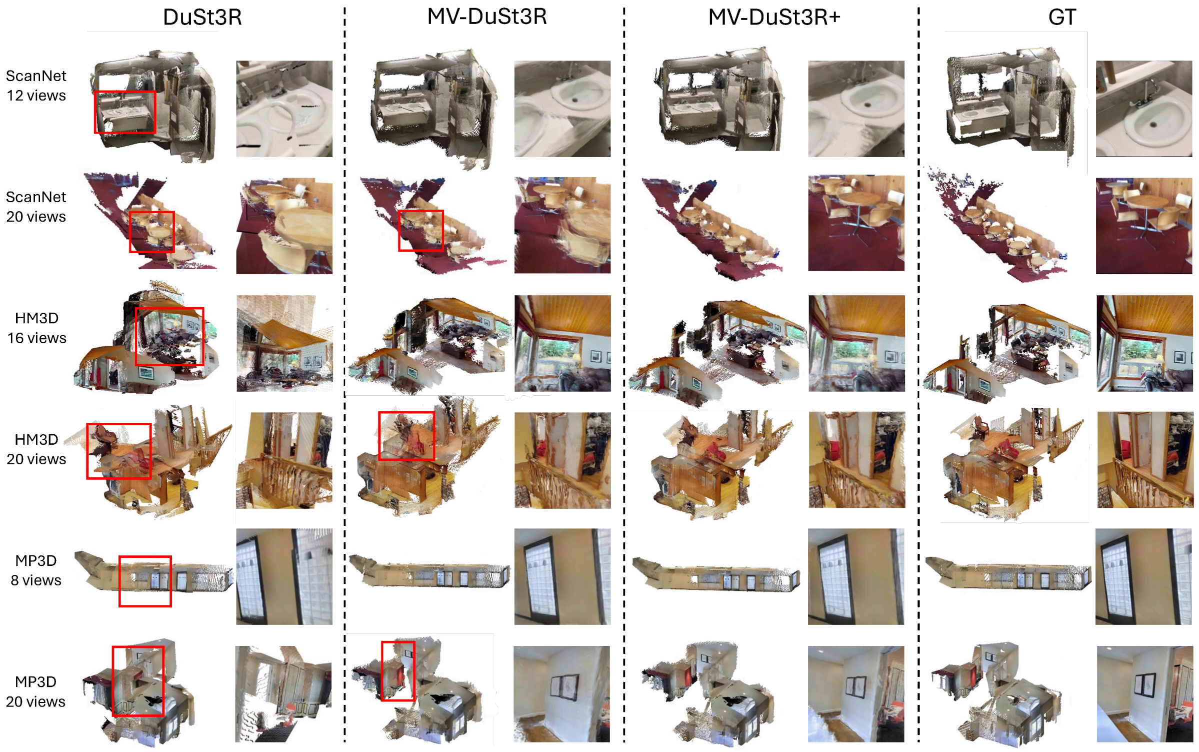

Figure 6. **MVS reconstruction and NVS qualitative results.** We show one method in each column, which includes the reconstructed pointcloud and 1 rendered new view. Incorrectly reconstructed geometry is highlighted in red boxes. DUSt3R often introduces incorrect pairwise reconstructions when the scene has multiple objects with similar appearance (*e.g.*, windows, chairs, doors), which can not be recovered by the global optimization. MV-DUSt3R is more robust overall but still sometimes fails to reconstruct geometry accurately in regions far away from the reference view, while MV-DUSt3R+ predicts geometry more evenly across the space.

Test	Training recipe	MVS Reconstruction			MVPE			NVS		
		ND ↓	DAc ↑	CD ↓	RRE ↓	RTE ↓	mAE ↓	PSNR ↑	SSIM ↑	LPIPS ↓
4 views	1-stage, 4 views	1.0	94.4	1.7(1.2)	0.9	0.9	2.6	20.7	6.3	1.7
	1-stage, 8 views	1.0	95.2	1.5(0.9)	1.2	1.1	4.9	20.2	6.1	1.9
	2-stage, mixed views	0.9	95.5	1.5(0.8)	0.8	0.7	2.0	20.7	6.3	1.8
12 views	1-stage, 4 views	6.3	0.3	23.8(18.2)	15.1	17.5	34.1	16.5	4.9	4.4
	1-stage, 8 views	1.2	91.5	1.8(0.7)	0.6	1.2	5.2	19.4	5.8	2.4
	2-stage, mixed views	1.2	92.2	1.5(1.0)	0.4	0.8	3.8	19.5	5.9	2.2
24 views	1-stage, 4 views	17.7	0.0	81.4(55.5)	45.5	47.4	63.2	14.5	4.6	6.2
	1-stage, 8 views	2.1	64.5	3.9(2.0)	3.0	6.5	15.8	18.4	5.4	3.2
	2-stage, mixed views	1.7	81.4	2.6(1.3)	1.4	3.0	9.1	19.1	5.7	2.7

Table 5. Impact of # of input views at training time on the performance of MV-DUSt3R+ on the HM3D evaluation set.

	Method	GS	HM3D			ScanNet			MP3D		
			ND ↓	DAc ↑	CD ↓	ND ↓	DAc ↑	CD ↓	ND ↓	DAc ↑	CD ↓
4 views	MV-DUSt3R	×	1.1	93.5	1.9(1.4)	1.0	93.3	2.0(0.4)	2.6	61.6	25.4(4.9)
		✓	1.1	92.2	2.0(1.1)	1.0	93.3	2.0(0.4)	2.5	62.4	25.3(4.1)
	MV-DUSt3R+	×	0.9	95.2	1.5(1.1)	0.8	94.8	1.4(0.4)	2.2	68.5	18.9(3.5)
		✓	1.0	95.2	1.5(0.9)	0.8	94.9	1.5(0.3)	2.2	68.0	19.9(3.4)
24 views	MV-DUSt3R	×	3.3	37.6	10.0(5.1)	2.2	75.8	2.7(0.7)	6.4	11.0	43.3(13.0)
		✓	3.4	36.7	10.0(3.5)	2.2	75.2	2.7(0.9)	6.3	12.2	38.6(13.9)
	MV-DUSt3R+	×	2.1	68.1	4.4(2.5)	1.4	84.9	1.5(0.5)	4.3	26.5	22.6(5.6)
		✓	2.1	64.5	3.9(2.0)	1.6	81.2	1.7(0.7)	4.3	26.7	22.0(5.9)

Table 6. Impact of adding Gaussian (GS) heads on MVS reconstruction performance on HM3D, ScanNet, and MP3D datasets.

upon MV-DUSt3R+'s 1-stage 8-view training, by using a mixed set of inputs with the number of views uniformly sampled between 4 and 12. As shown in Tab. 5, 1-stage training on 4 views doesn't generalize well to more views, 1-stage training on 8 views performs decent, and 2-stage training outperforms 1-stage training on almost all tasks on HM3D. See appendix for more 2-stage results.

Upper bound performance of our networks. We study oracle performance if the best reference view is chosen based on groundtruth. For MV-DUSt3R, we consider all input views as reference view candidates, and manually select the one with best MVS reconstruction (**MV-DUSt3R$_{\textbf{oracle}}$**). For MV-DUSt3R+, we choose the model path with best MVS reconstruction (**MV-DUSt3R+$_{\textbf{oracle}}$**). As shown in Tabs. 2 to 4, the performance gap between MV-DUSt3R+ and MV-DUSt3R+$_{\text{oracle}}$ is significantly smaller than that between MV-DUSt3R and MV-DUSt3R$_{\text{oracle}}$. This validates our multi-path MV-DUSt3R+ architecture.

Impact of adding Gaussian head on MVS reconstruction performance. In Tab. 6, we compare MV-DUSt3R and MV-DUSt3R+ models with and without Gaussian heads, while keeping other settings identical. As shown in Tab. 6, adding Gaussian heads does not significantly improve or degrade performance of our models on MVS reconstruction.

5. Conclusion

We propose fast single-stage networks MV-DUSt3R and MV-DUSt3R+ to reconstruct scenes from up to 24 input views in one feed-forward pass without requiring camera intrinsics and poses. We extensively evaluate results on 3 datasets in both supervised and zero-shot settings, and confirm compelling results and efficiency over prior art.

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *IJCV*, 2016. 2, 3, 6

[2] Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, et al. Scenescript: Reconstructing scenes with an autoregressive structured language model. *arXiv preprint arXiv:2403.13064*, 2024. 1

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022. 3

[4] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *CVPR*, 2023. 3

[5] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 3

[6] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023. 3

[7] Sylvain Bougnoux. From projective to euclidean space under any practical situation, a criticism of self-calibration. In *ICCV*, 1998. 1

[8] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *CVPR*, 2017. 1

[9] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 2016. 1

[10] Jonathan L Carrivick, Mark W Smith, and Duncan J Quincey. *Structure from Motion in the Geosciences*. John Wiley & Sons, 2016. 3

[11] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, 2022. 1

[12] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 2, 6

[13] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, 2024. 1, 3, 5, 7

[14] 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 *ICCV*, 2021. 3

[15] 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. In *ECCV*, 2024. 1, 3, 5

[16] Julian Chibane, Gerard Pons-Moll, et al. Neural unsigned distance fields for implicit function learning. *Neurips*, 2020. 3

[17] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 2, 6

[18] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4

[19] Yilun Du, Cameron Smith, Ayush Tewari, and Vincent Sitzmann. Learning to render novel views from wide-baseline stereo pairs. In *CVPR*, 2023. 3

[20] Bardienus Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jerome Revaud. Mast3r-sfm: a fully-integrated solution for unconstrained structure-from-motion. *arXiv preprint arXiv:2409.19152*, 2024. 3

[21] Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds. *arXiv preprint arXiv:2403.20309*, 2, 2024. 7

[22] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 7

[23] Qiancheng Fu, Qingshan Xu, Yew Soon Ong, and Wenbing Tao. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *Neurips*, 2022. 1

[24] Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2015. 3

[25] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *ICCV*, 2015. 3

[26] 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 *CVPR*, 2020. 3

[27] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003. 1

[28] Xingyi He, Jiaming Sun, Yifan Wang, Sida Peng, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Detector-free structure from motion. In *CVPR*, 2024. 3

[29] Sunghwan Hong, Jaewoo Jung, Heeseong Shin, Jiaolong Yang, Seungryong Kim, and Chong Luo. Unifying correspondence pose and nerf for generalized pose-free novel view synthesis. In *CVPR*, 2024. 3

[30] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH*, 2024. 1, 3

[31] Jakob Iglhaut, Carlos Cabo, Stefano Puliti, Livia Piermattei, James O'Connor, and Jacqueline Rosette. Structure from motion photogrammetry in forestry: A review. *Current Forestry Reports*, 2019. 3

[32] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 2023. 1, 2, 3, 5

[33] Mustafa Khan, Hamidreza Fazlali, Dhruv Sharma, Tongtong Cao, Dongfeng Bai, Yuan Ren, and Bingbing Liu. Autosplat: Constrained gaussian splatting for autonomous driving scene reconstruction. *arXiv preprint arXiv:2407.02598*, 2024. 1

[34] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *CVPR*, 2022. 3

[35] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Ep n p: An accurate o (n) solution to the p n p problem. *IJCV*, 2009. 7

[36] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. *arXiv preprint arXiv:2406.09756*, 2024. 2, 3

[37] Fangfu Liu, Wenqiang Sun, Hanyang Wang, Yikai Wang, Haowen Sun, Junliang Ye, Jun Zhang, and Yueqi Duan. Reconx: Reconstruct any scene from sparse views with video diffusion model. *arXiv preprint arXiv:2408.16767*, 2024. 3

[38] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *CVPR*, 2020. 3

[39] Zhiheng Liu, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jie Xiao, Kai Zhu, Nan Xue, Yu Liu, Yujun Shen, and Yang Cao. Infusion: Inpainting 3d gaussians via learning depth completion from diffusion prior. *arXiv preprint arXiv:2404.11613*, 2024. 3

[40] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1, 3

[41] Baorui Ma, Junsheng Zhou, Yu-Shen Liu, and Zhizhong Han. Towards better gradient consistency for neural signed distance functions via level set alignment. In *CVPR*, 2023. 1

[42] 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*, 2021. 1, 3

[43] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMARISMAR*, 2011. 3

[44] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion*. *Acta Numerica*, 2017. 3

[45] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 1, 3

[46] MV Peppa, JP Mills, KD Fieber, I Haynes, S Turner, A Turner, M Douglas, and PG Bryan. Archaeological feature detection from archive aerial photography with a sfm-mvs and image enhancement pipeline. *ISPRS*, 2018. 1

[47] Frank Plastria. The weiszfeld algorithm: proof, amendments, and extensions. *Foundations of location analysis*, 2011. 7

[48] Long Quan and Zhongdan Lan. Linear n-point camera pose determination. *IEEE TPAMI*, 1999. 1

[49] Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021. 2, 6

[50] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, 2011. 1, 3

[51] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, et al. Zeronvs: Zero-shot 360-degree view synthesis from a single real image. *arXiv preprint arXiv:2310.17994*, 2023. 3

[52] Aron Schmied, Tobias Fischer, Martin Danelljan, Marc Pollefeys, and Fisher Yu. R3d3: Dense 3d reconstruction of dynamic scenes from multiple cameras. In *ICCV*, 2023. 3

[53] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 3

[54] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 3

[55] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Neurips*, 2019. 3

[56] Vincent Sitzmann, Eric Chan, Richard Tucker, Noah Snavely, and Gordon Wetzstein. Metasdf: Meta-learning signed distance functions. *Neurips*, 2020. 3

[57] Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3r: Zero-shot gaussian splatting from uncalibarated image pairs. *arXiv preprint arXiv:2408.13912*, 2024. 3, 7

[58] Liangchen Song, Liangliang Cao, Hongyu Xu, Kai Kang, Feng Tang, Junsong Yuan, and Zhao Yang. Roomdreamer: Text-driven 3d indoor scene synthesis with coherent geometry and texture. In *ACM MM*, 2023. 3

[59] Stanislaw Szymanowicz, Chrisitian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *CVPR*, 2024. 3, 5

[60] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *ECCV*, pages 1–18. Springer, 2024. 1

[61] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms Corfu, Greece, September 21–22, 1999 Proceedings*, 2000. 1, 3

[62] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs. In *CVPR*, 2022. 1, 3

[63] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In *CVPR*, 2021. 3

[64] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. *arXiv preprint arXiv:2408.16061*, 2024. 3, 6

[65] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Visual geometry grounded deep structure from motion. *arXiv preprint arXiv:2312.04563*, 2023. 3

[66] Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *ICCV*, 2023. 7

[67] Kaixuan Wang and Shaojie Shen. Mvdepthnet: Real-time multiview depth estimation neural network. In *3DV*, 2018. 3

[68] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 1, 2, 3, 6, 7

[69] Yuesong Wang, Zhaojie Zeng, Tao Guan, Wei Yang, Zhuo Chen, Wenkai Liu, Luoyuan Xu, and Yawei Luo. Adaptive patch deformation for textureless-resilient multi-view stereo. In *CVPR*, 2023. 3

[70] 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*, 2004. 7

[71] Matthew J Westoby, James Brasington, Niel F Glasser, Michael J Hambrey, and Jennifer M Reynolds. 'structure-from-motion'photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 2012. 3

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

[73] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. In *CVPR*, 2024. 3

[74] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *CVPR*, 2018. 6

[75] Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, et al. Pandaset: Advanced sensor suite dataset for autonomous driving. In *ITSC*, 2021. 1

[76] Qingshan Xu and Wenbing Tao. Multi-scale geometric consistency guided multi-view stereo. In *CVPR*, 2019. 3

[77] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. *arXiv preprint arXiv:2403.14621*, 2024. 1

[78] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, 2018. 3

[79] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *CVPR*, 2019. 3

[80] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Neurips*, 2020. 3

[81] Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. *arXiv preprint arXiv:2410.24207*, 2024. 3

[82] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. gsplat: An open-source library for Gaussian splatting. *arXiv preprint arXiv:2409.06765*, 2024. 2

[83] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023. 6

[84] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *ECCV*, 2016. 1, 3

[85] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 3

[86] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *CVPR*, 2024. 3

[87] Rui Zeng, Yuhui Wen, Wang Zhao, and Yong-Jin Liu. View planning in robot active vision: A survey of systems, algorithms, and applications. *Computational Visual Media*, 2020. 1

[88] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *ECCV*, pages 1–19. Springer, 2024. 1

[89] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 7

[90] Zhengdong Zhang, Yasuyuki Matsushita, and Yi Ma. Camera calibration with lens distortion from low-rank textures. In *CVPR*, 2011. 1

[91] Enliang Zheng, Enrique Dunn, Vladimir Jojic, and Jan-Michael Frahm. Patchmatch based joint view selection and depthmap estimation. In *CVPR*, 2014. 3

[92] Lei Zhou, Siyu Zhu, Zixin Luo, Tianwei Shen, Runze Zhang, Mingmin Zhen, Tian Fang, and Long Quan. Learning and matching multi-view descriptors for registration of point clouds. In *ECCV*, 2018. 3

[93] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *CVPR*, 2024. 3