DiffusionSfM: Predicting Structure and Motion via Ray Origin and Endpoint Diffusion

Qitao Zhao, Amy Lin, Jeff Tan, Jason Y. Zhang, Deva Ramanan, Shubham Tulsiani

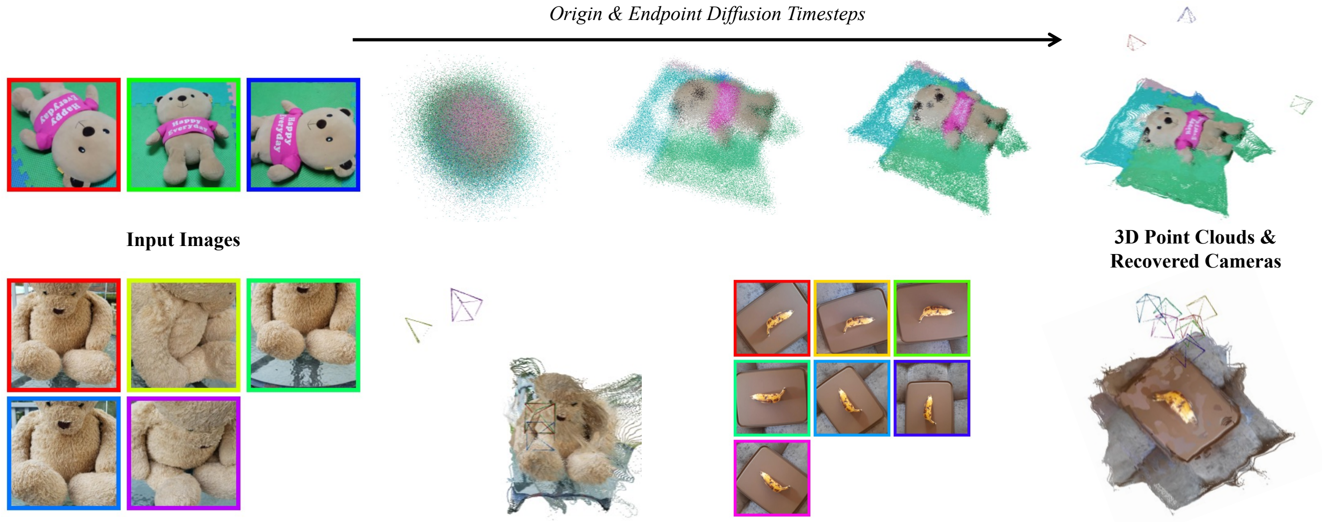
Origin & Endpoint Diffusion Timesteps
Input Images
3D Point Clouds & Recovered Cameras

Figure 1. **DiffusionSfM. Top:** Given a set of multi-view images as input (left), DiffusionSfM parametrizes scene geometry and cameras (right) as pixel-wise ray origins and endpoints in a global frame and learns a denoising diffusion model to infer these from multi-view input. In contrast to current structure from motion pipelines, which often adopt a two-stage approach of pairwise reasoning followed by global optimization, our method unifies both stages into a single end-to-end multi-view reasoning step. **Bottom:** Sample results for the inferred scene geometry and cameras given multi-view input. On the challenging real-world CO3D dataset, we find that DiffusionSfM yields higher-fidelity geometry and cameras compared to prior classical and learning-based methods.

Abstract

Current Structure-from-Motion (SfM) methods often adopt a two-stage pipeline involving learned or geometric pairwise reasoning followed by a global optimization. We instead propose a data-driven multi-view reasoning approach that directly infers cameras and 3D geometry from multi-view images. Our proposed framework, DiffusionSfM, parametrizes scene geometry and cameras as pixel-wise ray origins and endpoints in a global frame, and learns a transformer-based denoising diffusion model to predict these from multi-view input. We develop mechanisms to overcome practical challenges in training diffusion models with missing data and unbounded scene coordinates, and demonstrate that DiffusionSfM allows accurate prediction of 3D and cameras. We empirically validate our approach on challenging real world data and find that DiffusionSfM improves over prior classical and learning-based methods, while also naturally modeling uncertainty and allowing external guidance to be incorporated in inference.

1. Introduction

The task of recovering structure (geometry) and motion (cameras) from multi-view images has long been a focus of the computer vision community, with typical pipelines [21] performing pairwise correspondence estimation followed by global optimization. While classical methods relied on hand-designed features, matching, and optimization, there has been a recent shift towards incorporating learning-based alternatives [3, 4, 12, 20]. More recently, the widely-influential DUSt3R [30] advocates for predicting pairwise 3D pointmaps (instead of only correspondences), demonstrating that this can yield accurate dense geometry and cameras. In order to reconstruct more than 2 views, DUSt3R (and its variants) still require a global optimization reminiscent of classic bundle adjustment. While these methods, both classical and learning-based, have led to impressive improvements in SfM, the overall approach is largely unchanged – learned or geometric pairwise reasoning followed by global optimization. In this work, we seek to develop an alternative approach that directly predicts both structure and motion, while unifying pairwise reasoning and global optimization for data-driven multi-view reasoning.

We are of course not the first to attempt to find unified alternatives to the two-stage SfM pipeline. In the sparse-view setting where conventional correspondence-based methods struggle, several works employ multi-view architectures to jointly reason across input views. SparsePose [22], RelPose++ [10], and PoseDiffusion [28] all leverage multi-view transformers to estimate camera poses for input images, albeit using differing mechanisms such as regression, energy-based modeling, and denoising diffusion. More recently, RayDiffusion [32] argues for a local raymap parameterization of cameras instead of a global extrinsic matrix, and show that existing patch-based transformer architectures can be easily adapted for this task, yielding significantly more accurate pose predictions. Importantly, such methods predict only camera motion and fail to predict scene structure.

In this work, we present DiffusionSfM, an end-to-end multi-view model that directly infers cameras and dense 3D geometry from multiple input images. Instead of inferring rays per pixel (as in RayDiffusion [32]) or 3D points per pixel (as in DUSt3R [30]), DiffusionSfM effectively combines both to predicts ray *origins and endpoints* per pixel, directly reporting both scene geometry (endpoints) and generalized cameras (rays). These can readily be converted back to traditional cameras [32]. Compared to RayDiffusion, our model directly predicts structure as well as motion. Compared to DUSt3R, our model directly predicts motion as well as structure, but even more importantly does so for N views, eliminating the need for memory-intensive global alignment. To model uncertainty, we train a denoising diffusion model but find two key challenges need to be addressed. First, diffusion models require (noisy) ground-truth as input for training but existing real datasets do not have known endpoints for all pixels due to missing depth in multi-view stereo. Second, the 3D coordinates of endpoints can be potentially unbounded, whereas diffusion models require normalized data. We develop mechanisms to overcome these challenges, leveraging additional "mask conditioning" as input to inform the model of missing input data, and parameterizing 3D points in projective space instead of Euclidean space. We find that these allow us to learn accurate predictions for structure and motion.

We train and evaluate our system on the CO3D dataset [18] and find that DiffusionSfM yields more accurate geometry and cameras compared to prior works trained on similar data (see Fig. 1 for sample predictions). We also show that the probabilistic nature of our system allows recovering uncertainty, and the iterative inference allows easily incorporating external guidance without any fine-tuning, *e.g*. off-the-shelf monocular depth prediction to boost the accuracy of the recovered geometry. In summary, we show that DiffusionSfM can serve as a unified multi-view reasoning model for 3D geometry and cameras.

2. Related Work

Structure from Motion. Structure-from-Motion (SfM) systems [21] aim to simultaneously recover geometry and cameras given a set of input images. The typical SfM pipeline extracts pixel correspondences from keypoint matching [1, 14], then performs global bundle adjustment (BA) to optimize sparse 3D points and camera parameters by minimizing reprojection errors. Recently, SfM pipelines have been substantially enhanced by replacing classical subcomponents with learning-based methods, such as neural feature descriptors [4, 6], image matching [13, 20, 26], and bundle adjustment [11, 27].

More recently, an emerging body of research aims to unify the various SfM subcomponents into an end-to-end neural framework. Notably, ACEZero [2] fits a single neural network to input images and learns pixel-aligned 3D coordinates in a self-supervised manner, while FlowMap [23] predicts per-frame cameras and depth maps while using off-the-shelf optical flow as a supervision signal. Though ACEZero and FlowMap are promising attempts to revolutionize SfM pipelines, they both register images incrementally and may suffer under large viewpoint changes. DUSt3R [30] directly regresses 3D pointmaps from image pairs and shows strong generalization ability [9, 18]. On top of DUSt3R, MASt3R [8] adds a feature head to DUSt3R, offering pixel-matching capabilities. MASt3R-SfM [5] is a more scalable SfM pipeline based on MASt3R. While these approaches [5, 8, 30] show impressive performance and robustness under sparse views, they are essentially pair-based, requiring sophisticated global alignment procedures to form a consistent estimate for more than two views.

Pose estimation with global reasoning. For the task of sparse-view pose estimation, learning-based methods equipped with global reasoning show favorable robustness where traditional SfM methods [21, 24] fail. This line of research includes energy-based [10, 31], regression-based [22], and diffusion-based pose estimators [28, 32]. Among them, diffusion-based methods show a better ability to handle uncertainty. Closest to our work, RayDiffusion [32] leverages a denoising diffusion model [7, 16] with a patch-aligned ray representation to predict generic cameras. Our method goes further and pursues a generic representation

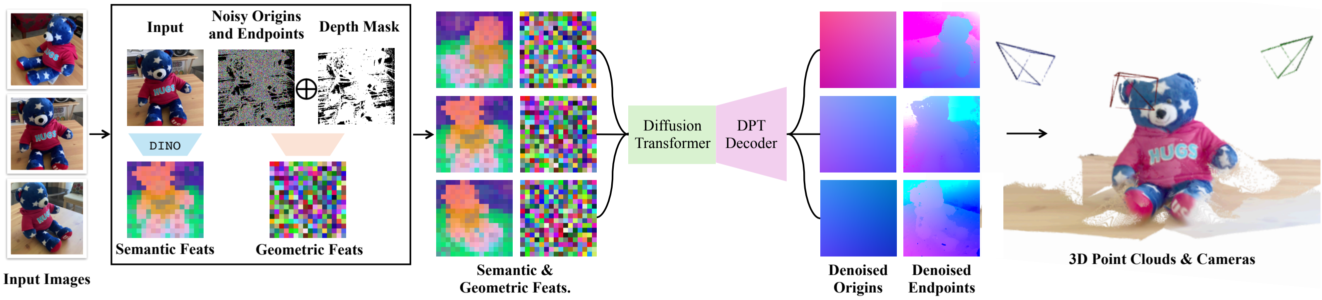

Figure 2. **Method.** Given sparse multi-view images as input, DiffusionSfM predicts pixel-wise ray origins and endpoints for each input image in a global frame (Sec. 3.1), via a denoising diffusion process. For each image, we compute off-the-shelf patch-wise image embeddings using DINOv2 [15]. We use a single downsampling convolutional layer to embed noisy ray origins and endpoints into noisy ray latents, while matching the spatial footprint of the image embeddings. We implement a diffusion transformer architecture that predicts clean ray origin and endpoint latents from noisy samples. A convolutional DPT head outputs full-resolution denoised ray origins and endpoints. The ray endpoints can be directly visualized in 3D, or further post-processed to recover camera extrinsics, camera intrinsics, and multi-view consistent depth maps. At inference, the depth mask is set to all ones so that the diffusion model predicts origins and endpoints for all pixels.

for both cameras and geometry, in the form of as ray origins and endpoints for each pixel. In addition to resulting in a richer output, this joint geometry and pose prediction also yields improvements for pose estimation.

3. Method

Given a set of sparse (*i.e*. 2-8) input images, DiffusionSfM predicts the geometry and cameras of a 3D scene in a global coordinate frame. In Sec. 3.1, we propose to represent 3D scenes as dense pixel-aligned ray origins and endpoints. To predict such scene representations from sparse input images while modeling uncertainty, Sec. 3.2 proposes a denoising diffusion architecture for dense ray origin and endpoint prediction. We then discuss some key practical challenges in training such a model in Sec. 3.3 and then discuss how such a diffusion model can also incorporate additional external cues for inference in Sec. 3.4.

3.1. 3D Scenes as Ray Origins and Endpoints

Given an input image $\mathbf{x}$ with depth map $\mathbf{D}$, camera intrinsics $\mathbf{K} \in \mathbb{R}^{3\times3}$, and world-to-camera extrinsics $\mathbf{P} \in \mathbb{R}^{4\times4}$ (equivalently, rotation $\mathbf{R} \in SO(3)$ and translation $\mathbf{T} \in \mathbb{R}^3$), each 2D image pixel $\mathbf{p}_i = [u_i, v_i]$ corresponds to a ray that travels from the camera center $\mathbf{c}$ through the pixel's projected position on the image plane, terminating at the object's surface as specified by the depth map $\mathbf{D}$. The endpoint of the ray associated with image pixel $\mathbf{p}_i$ is given by:

$$\mathbf{e}_i = \mathbf{P}^{-1} h \left(\mathbf{D}_i \cdot \mathbf{K}^{-1} [u_i, v_i, 1]^T\right) \tag{1}$$

where h maps the 3D point into homogeneous coordinates. The shared ray origin $\mathbf{o}_i$ is equivalent to the camera center $\mathbf{c}$, and can be computed as:

$$\mathbf{o}_i = \mathbf{c} = h\left(-\mathbf{R}^{-1}\mathbf{T}\right) \tag{2}$$

In summary, we associate each image pixel with a ray origin and endpoint $\mathbf{s}_i = \langle \mathbf{o}_i, \mathbf{e}_i \rangle$ in world coordinates, describing the location of the observing camera and the observed 3D point on the object surface. Given a bundle of ray origins and endpoints, we can extract the corresponding camera pose and depth maps: the supplemental includes the conversion details.

Learning Over-Parameterized Representations. While representing scenes as ray origins and endpoints appears redundant, it facilitates leveraging the distributed deep features learned by state-of-the-art vision backbones [32], such as DINOv2 [15], which encode image information in a patch-wise manner. It is worth mentioning that although the ray origins $\{\mathbf{o}_i\}$ should ideally be identical for all pixels i, we predict ray origins densely alongside ray endpoints $\{\mathbf{e}_i\}$. This approach encourages the predicted ray origins to be close to each other within the same image, serving as regularization during model training.

3.2. DiffusionSfM

We propose a denoising Diffusion Transformer (DiT) architecture [16] that predicts ray origins and endpoints (Sec. 3.1) via a denoising diffusion process. An overview of DiffusionSfM is given in Fig. 2.

Diffusion Framework. Given a set of N input images $\{\mathbf{x}^{(i)}\}_{i=1}^{N}$, with corresponding pixel-aligned ray origins and endpoints $\mathcal{S} = \{\mathbf{s}^{(i)}\}_{i=1}^{N}$, we apply a forward diffusion process [7, 25] that adds time-dependent Gaussian noise to the origin and endpoint pointmaps. Let $\mathcal{S}_t$ denote the set of time-dependent noisy origins and pointmaps, where $\mathcal{S}_0$ is the clean sample and $\mathcal{S}_T$ is the noisy sample at the largest diffusion timestep T (approximating Gaussian noise). The

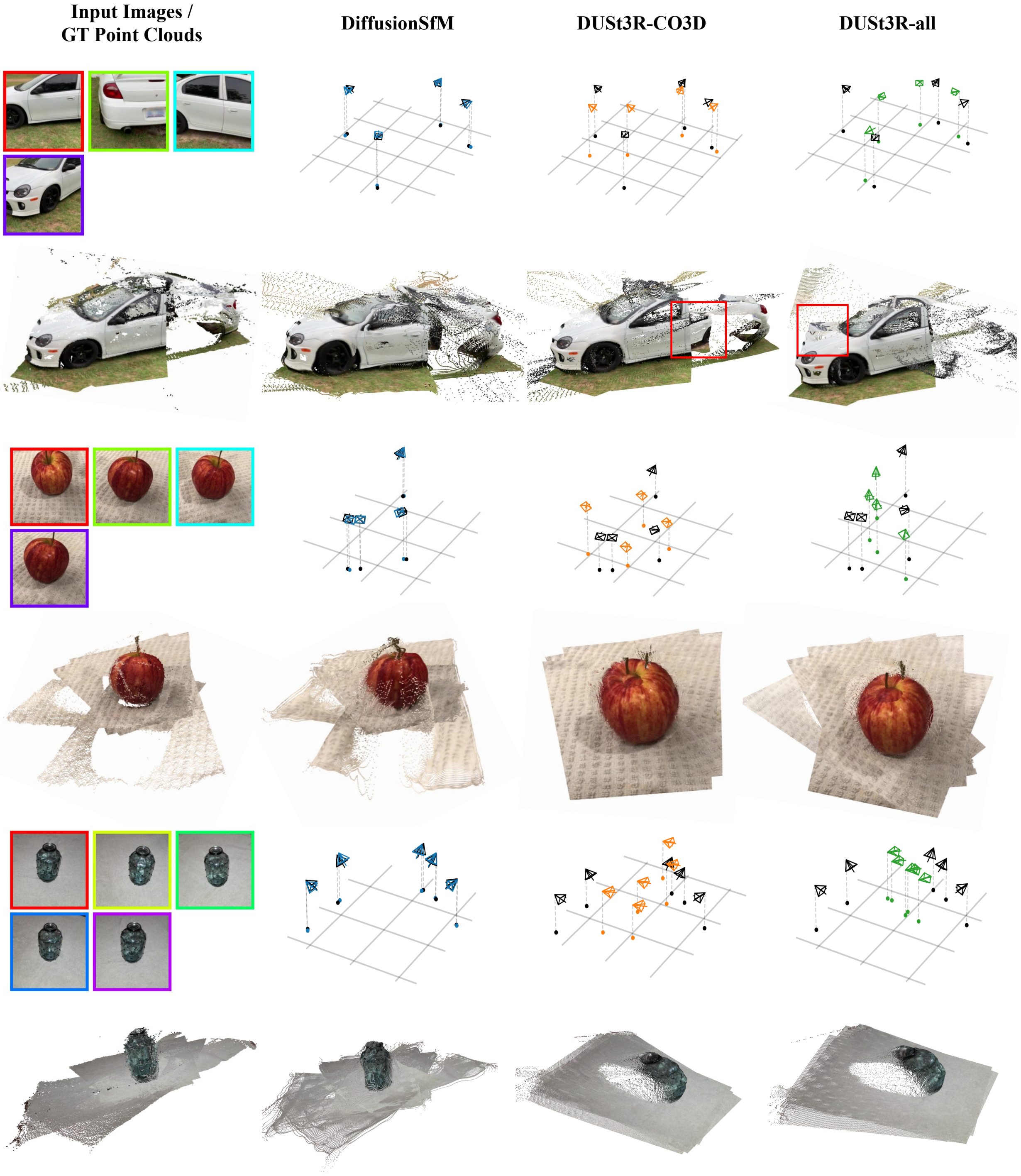

Figure 3. **Qualitative Comparison on Camera Pose Accuracy and Predicted Geometry.** For each method, we plot the ground-truth cameras in black and the predicted cameras in other colors. DiffusionSfM yields more accurate camera poses than prior art, and produces point clouds that more closely match the ground-truth point clouds on the left. Compared to DUSt3R, which sometimes fails to register images in a consistent manner, DiffusionSfM consistently yields a coherent global prediction.

forward diffusion process is defined by:

$$\mathcal{S}_t = \sqrt{\bar{\alpha}_t}\,\mathcal{S}_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon \tag{3}$$

where $t \sim \text{Uniform}(0, T]$, $\epsilon \sim \mathcal{N}(0, I)$, and $\bar{\alpha}_t$ follows a pre-defined noise schedule that controls the strength of added noise at each timestep. To perform the reverse diffusion process, which progressively reconstructs the clean sample given noisy observations, we train a diffusion model f_θ that takes the set $\mathcal{S}_t$ as input and optionally incorporates additional conditioning information $\mathcal{C}$. The model is trained using the following loss function:

$$\mathcal{L}_{\text{Diffusion}} = \mathbb{E}_{t,\mathcal{S}_0,\epsilon} \| \mathcal{S}_0 - f_\theta(\mathcal{S}_t, t, \mathcal{C}) \|^2 \tag{4}$$

Architecture. We implement f_θ using a DiT [16] architecture conditioned on deep image features $\mathcal{C} \in \mathbb{R}^{N \times h \times w \times c_1}$ from DINOv2 [15], where h and w are the patch resolution and c_1 is the embedding dimension. To align pixels to the spatial information learned by DINOv2, we apply a convolutional layer with kernel size and stride equal to ViT patch size: this spatially downsamples the noisy origins and endpoints $\mathcal{S}$ to match the DINOv2 features while increasing their feature dimension:

$$\mathcal{F} = \texttt{Conv}(\mathcal{S}_t) \in \mathbb{R}^{N \times h \times w \times c_2} \tag{5}$$

The combined DiT input is constructed by concatenating these two feature sets along the channel dimension: $\mathcal{F} \oplus \mathcal{C}$.

While the DiT operates on low-resolution features, our objective is to produce pixel-aligned dense ray origins and endpoints. To achieve this, we employ a DPT (Dense Prediction Transformer) [17] decoder, which takes intermediate feature maps from both DINOv2 and DiT as inputs. The DPT decoder progressively increases the feature resolution through several convolutional layers. The final ray origins and endpoints are decoded from the DPT output using a single linear layer. During inference, we apply the trained model in the reverse diffusion process to iteratively denoise a randomly initialized Gaussian sample.

3.3. Practical Training Considerations

Homogeneous Coordinates for Unbounded Geometry. Real-world scenes tend to exhibit large variations in scale. However, neural networks tend to train most effectively when working with bounded inputs and outputs (*e.g.* between -1 and 1). To stabilize training across the large scale variations present in 3D scene datasets, we propose to represent ray origins and endpoints in homogeneous coordinates.

Specifically, given any 3D point, we first apply a homogeneous transform from $\mathbb{R}^3 \to \mathbb{P}^3$:

$$(x, y, z) \to \frac{1}{w}(x, y, z, 1) \tag{6}$$

where w is an arbitrary scale factor. To encourage bounded coordinates in practice, we choose w such that the homogeneous coordinate is unit-norm:

$$w := \sqrt{x^2 + y^2 + z^2 + 1} \tag{7}$$

Unit normalization allows homogeneous coordinates to serve as a bounded representation for unbounded scene geometry. For example, $(x, y, z, 0)$ is a point at infinity in the direction of (x, y, z). In the supplemental material, we analyze the impact of homogeneous representations on training stability.

Training with Incomplete Ground Truth. A significant challenge in diffusion training is that ground truth depth values from real-world datasets often contain invalid or missing data. It is highly undesirable for missing ray endpoints derived from incomplete data to be interpreted as part of the target distribution. Unlike regression models which can simply mask the loss at invalid pixels, diffusion models gradually map entire images from the noise distribution to the target distribution and thus require dense supervision. Many real-world datasets such as CO3D [18] and MegaDepth [9] only provide sparse SfM [21] point clouds, resulting in incomplete depth information.

To mitigate this issue, we further apply depth masks $\mathcal{M} \in \mathbb{R}^{N \times H \times W}$ to the DiT inputs, where zero values indicate pixels with invalid depth. During training, we multiply DiT inputs with depth masks element-wise, then concatenate along the channel dimension: $\mathcal{S}'_t = (\mathcal{M} \cdot \mathcal{S}_t) \oplus \mathcal{M}$. Then, we only compute the diffusion loss in Eq. 4 only over unmasked pixels. By implementing these strategies, we encourage the model to focus on regions with valid ground truth values during training. During inference, however, we would like the diffusion process to estimate endpoints at all pixels, so we always use depth masks with values set to one.

Sparse-to-Dense Training. In practice, we find that training the entire model from scratch leads to slow convergence and suboptimal performance. To address this, we propose a sparse-to-dense training approach. First, we train a sparse version of the model, where the DPT decoder is removed, and the output ray origins and endpoints have the same spatial resolution as the DINOv2 features. Unlike Eq.5, no spatial downsampling is required, so this sparse model uses a single linear layer to embed the noisy ray origins and endpoints. Once the sparse model is trained, we initialize the dense model DiT with the learned weights from the sparse model. This two-stage approach significantly improves performance: see supplementary for comparisons.

3.4. Guidance for Diffusion Inference

One benefit of a diffusion formulation is that the iterative denoising process allows us to guide intermediate sample predictions towards external signals, refining the predictions to be more precise while maintaining multi-view consistency. After each step of diffusion, we push the depth values of predicted clean ray endpoints $\{\mathbf{e}_i\}_{i=1}^N$ towards depths derived from unprojecting off-the-shelf estimates from monocular depth networks. Let $\{\hat{\mathbf{d}}_i\}$ be the depth of each predicted ray endpoint after projecting into the estimated camera frame, and let $\{\mathbf{d}_i^{\text{MoGe}}\}$ be the output of MoGe [29] after 1D optimal alignment to match the scale of $\{\hat{\mathbf{d}}_i\}$. We calculate the updated ray endpoints after guidance as:

$$\hat{\mathbf{d}}_i' = (1-\lambda)\hat{\mathbf{d}}_i + \lambda \mathbf{d}_i^{\text{MoGe}} \tag{8}$$

where $\lambda = 0.2$. We show in Tab. 4 that diffusion guidance helps improve pose and geometry accuracy.

4. Experiments

4.1. Experimental Setup

Dataset. We train and evaluate our model on the CO3D [18] dataset, featuring turntable video sequences of object categories. Following prior work [10, 32], we train our model on 41 object categories and hold out 10 unseen categories to evaluate generalization.

Baselines and Metrics. To evaluate camera pose accuracy in the sparse-view setup, we compare with Ray Diffusion [32] and DUSt3R [30], along with previous methods [10, 19, 28]. It is important to note that DUSt3R is trained on a blend of eight datasets, including samples from all CO3D categories, whereas our model is exclusively trained on a subset of CO3D. To ensure a fair comparison, we re-train DUSt3R on the 41-10 split of CO3D, using the authors' official implementation and hyperparameters (referred to as DUSt3R-CO3D). The model trained on all eight datasets is referred to as DUSt3R-all. To evaluate camera predictions, we follow prior work [32] and convert predicted rays back to traditional extrinsic matrices and report two pose accuracy metrics: (1) Camera Rotation Accuracy which compares the predicted relative camera rotation between images against ground truth and (2) Camera Center Accuracy which compares predicted camera centers to the ground truth after a similarity alignment. To evaluate the estimated geometry (*i.e.* ray endpoints), we report Chamfer Distance (CD) and compare our method with both versions of DUSt3R. We use 2-8 images as input across evaluations.

4.2. Evaluation on CO3D

Camera Pose Accuracy. We present the quantitative results in Tab. 1. Across both seen and unseen categories, DiffusionSfM achieves comparable camera rotation accuracy to DUSt3R-all despite using less training data, and outperforms baselines trained on CO3D such as DUSt3R-CO3D and Ray Diffusion. Notably, our method demonstrates strong generalization, achieving significant improvements over CO3D-trained baselines on unseen categories. For camera center accuracy, our approach consistently surpasses other methods, including DUSt3R-all. Additionally, the qualitative results in Fig. 3 illustrate that DiffusionSfM produces robust predictions under partial observations (*e.g.* the car example) and for symmetric structures (*e.g.* the apple and vase examples), where DUSt3R often produces inaccurate results. We attribute this improvement to our model's probabilistic modeling capability derived from diffusion, as well as its multi-view reasoning abilities, which together effectively handle these challenging scenarios.

Predicted Geometry. To evaluate predicted geometry, we compute Chamfer Distance (CD) and show comparisons against baselines in Tab. 2. We compute CD in two setups (with and without foreground object mask), and find that our method performs best without foreground masking. In this setup, the predicted ray endpoints corresponding to the background image pixels tend to have larger scale variations than foreground ones, and therefore dominate CD. This result indicates that our model provides more accurate predictions for complex image backgrounds. In terms of CD with masking, DUSt3R-all achieves the best result, while our approach outperforms DUSt3R-CO3D. See Fig. 3 for visualization.

Inference Speed. Though our method requires iterative diffusion denoising at inference, we can speed this up by performing early stopping. Specifically, we can treat the x_0-prediction from early timesteps as our output instead of iterating over all denoising timesteps. Consistent with observations by Zhang *et al.* [32], this in fact yields more accurate predictions than the final-step diffusion outputs. As a result we only require 10 denoising diffusion timesteps for inference, taking 1.91 seconds on a single A5000 GPU with 8 input images. In contrast, DUSt3R takes 8.73 seconds to run the complete pairwise inference and global alignment procedure. We provide additional analysis in the supplementary material.

4.3. Ablation Study

Homogeneous Coordinates. The homogeneous representation for ray origins and endpoints is critical for stable model training. Specifically, we experiment with replacing the proposed homogeneous representation with origins and endpoints in $\mathbb{R}^3$. We include more details regarding this experiment in the supplementary material.

Depth Mask Conditioning. To assess the effectiveness of the proposed depth mask conditioning, we train a baseline model without depth mask conditioning. For missing values in the ground truth depth maps, we apply nearest-neighbor interpolation to fill in invalid pixels. This experiment is performed on an early checkpoint of our model for efficient training evaluation, with results presented in Tab. 3. The findings indicate that omitting depth mask conditioning significantly degrades both camera pose accuracy and

		Rotation Accuracy (↑, @ 15°)							Center Accuracy (↑, @ 0.1)						
	# of Images	2	3	4	5	6	7	8	2	3	4	5	6	7	8
Seen Categories	COLMAP [19]	30.7	28.4	26.5	26.8	27.0	28.1	30.6	100	34.5	23.8	18.9	15.6	14.5	15.0
	PoseDiffusion [28]	75.7	76.4	76.8	77.4	78.0	78.7	78.8	100	77.5	69.7	65.9	63.7	62.8	61.9
	RelPose++ [10]	81.8	82.8	84.1	84.7	84.9	85.3	85.5	100	85.0	78.0	74.2	71.9	70.3	68.8
	Ray Diffusion [32]	91.8	92.4	92.6	92.9	93.1	93.3	93.3	100	94.2	90.5	87.8	86.2	85.0	84.1
	DUSt3R-CO3D [30]	86.7	87.9	88.0	88.2	88.6	88.8	88.9	100	92.0	86.8	83.8	82.0	81.1	80.4
	DUSt3R-all [30]	91.7	92.7	**93.3**	**93.6**	**93.8**	**94.0**	**94.3**	100	93.0	85.7	81.9	79.6	77.8	76.8
	DiffusionSfM (Ours)	**92.4**	**93.0**	**93.3**	93.5	93.6	93.8	93.8	100	**95.2**	**92.1**	**90.5**	**89.2**	**88.7**	**87.8**
Unseen Categories	COLMAP [19]	34.5	31.8	31.0	31.7	32.7	35.0	38.5	100	36.0	25.5	20.0	17.9	17.6	19.1
	PoseDiffusion [28]	63.2	64.2	64.2	65.7	66.2	67.0	67.7	100	63.6	50.5	45.7	43.0	41.2	39.9
	RelPose++ [10]	69.8	71.1	71.9	72.8	73.8	74.4	74.9	100	70.6	58.8	53.4	50.4	47.8	46.6
	Ray Diffusion [32]	83.5	85.6	86.3	86.9	87.2	87.5	88.1	100	87.7	81.1	77.0	74.1	72.4	71.4
	DUSt3R-CO3D [30]	79.8	81.5	82.6	82.7	83.0	83.3	83.7	100	83.6	77.2	71.8	70.0	68.1	67.0
	DUSt3R-all [30]	**90.8**	**92.6**	**93.6**	**93.6**	**93.8**	**93.6**	**93.4**	100	87.9	79.8	74.3	71.7	69.4	67.8
	DiffusionSfM (Ours)	90.1	91.0	91.8	92.6	92.9	93.0	93.1	100	**90.9**	**85.7**	**83.7**	**82.4**	**80.9**	**80.7**

Table 1. **Camera Rotation and Center Accuracy on CO3D.** On the left, we report the proportion of relative camera rotations within 15° of the ground truth. On the right, we report the proportion of camera centers within 10% of the scene scale. To align the predicted camera centers to ground-truth, we apply an optimal similarity transform (s, $\mathbf{R}$, $\mathbf{t}$), hence the alignment is perfect at $N = 2$ but worsens with more images. DiffusionSfM outperforms all other methods for camera center accuracy, and outperforms all methods trained on equivalent data for rotation accuracy.

# of Images	2	3	4	5	6	7	8
DUSt3R* [30]	0.036	0.037	0.040	0.040	0.037	0.036	0.039
DUSt3R [30]	0.021	0.023	**0.024**	0.024	0.025	0.025	**0.023**
DiffusionSfM	**0.020**	**0.022**	**0.024**	**0.023**	**0.022**	**0.023**	**0.023**
DUSt3R* [30]	0.038	0.036	0.036	0.036	0.034	0.033	0.034
DUSt3R [30]	**0.023**	**0.022**	**0.019**	**0.020**	**0.019**	**0.020**	**0.020**
DiffusionSfM	0.031	0.027	0.026	0.026	0.026	0.025	0.026

Table 2. **Chamfer Distance (↓) on CO3D Unseen Categories.** Top: CD on all scene points. Bottom: CD on foreground points only. DUSt3R* is trained on only CO3D, while DUSt3R-all is trained on multiple datasets. DiffusionSfM outperforms all other methods on full scene geometry, and outperforms both DUSt3R-CO3D on foreground geometry.

# of Images	2	3	4	5	6	7	8
w/o Mask	82.4	84.1	84.7	85.6	85.7	86.0	85.9
DiffusionSfM	**87.1**	**89.0**	**90.1**	**90.7**	**90.9**	**90.9**	**91.2**
w/o Mask	100.0	89.0	82.2	78.9	76.5	74.3	72.9
DiffusionSfM	100.0	**89.7**	**84.6**	**82.0**	**79.9**	**78.8**	**78.1**
w/o Mask	0.029	0.029	0.031	0.030	0.030	0.030	0.029
DiffusionSfM	**0.024**	**0.028**	**0.029**	**0.027**	**0.028**	**0.026**	**0.027**

Table 3. **Ablation Study on Depth Mask Conditioning.** Top: Camera rotation accuracy (↑). Middle: Camera center accuracy (↑). Bottom: Chamfer distance (↓). Experiments are conducted on unseen categories from CO3D, using an early model checkpoint for training efficiency. Adding mask conditioning to indicate missing data during training significantly improves camera accuracy and geometry quality.

predicted geometry. While interpolation can be effective for filling missing depth within object regions, it can introduce substantial noise in the background (*e.g.* the sky). Such noise adversely impacts diffusion model training, as the entire input ray origin and endpoint map are used.

4.4. Leveraging Diffusion Denoising Process

Diffusion Guidance. The iterative denoising process enables using external signals to make better predictions for each timestep, bringing a more precise final prediction. Here, we leverage the monocular depth estimates from MoGe [29] in two ways: (1) Linear interpolation guidance and (2) Direct replacement (Tab. 4). The results show that monocular depth guidance helps improve the predicted geometry while direct replacement degrades the predicted geometry as the MoGe estimates for each input view may not be multi-view consistent.

Multi-modality from Multiple Diffusion Sampling. One benefit of using diffusion models is that we can produce diverse samples given challenging input images. For example, in Fig. 4, the vase has symmetrical patterns, and we show two different predicted endpoints from DiffusionSfM: both samples explain the flowers in the images in different ways. Compared to regression models *e.g.* DUSt3R [6], DiffusionSfM excels at handling uncertainty. To visualize the uncertainty of our model, we can generate uncertainty maps for DiffusionSfM by running multiple inferences and then taking the variances of the predictions and setting a

# of Images	2	3	4	5	6	7	8
DiffusionSfM	0.024	0.029	0.036	**0.028**	0.031	**0.026**	**0.027**
w/ Guidance	**0.020**	**0.026**	**0.031**	0.029	**0.027**	**0.026**	**0.027**
w/ Replacing	0.037	0.046	0.061	0.057	0.052	0.046	0.046

Table 4. **Effect of Monocular Depth Diffusion Guidance on Predicted Geometry.** We conduct experiments on (a smaller subset of instances from) unseen categories from CO3D and report chamfer distance (↓) to assess the quality of recovered geometry. Adding diffusion guidance improves the quality of predicted geometry in most cases. In contrast, naively replacing the diffusion ray endpoints with unprojected monocular depth results in substantially worse performance, highlighting the benefit of multi-view reasoning for geometry inference.

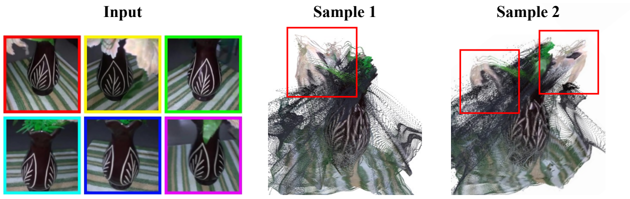

Figure 4. **Multi-modality of DiffusionSfM.** We show two distinct samples from DiffusionSfM, starting from the same input images but different random noise. Sample 1 explains the input images by putting all flowers on the left side, while Sample 2 places one flower on each side. DiffusionSfM is able to predict multi-modal geometry distributions when the scene layout may be ambiguous in the input images.

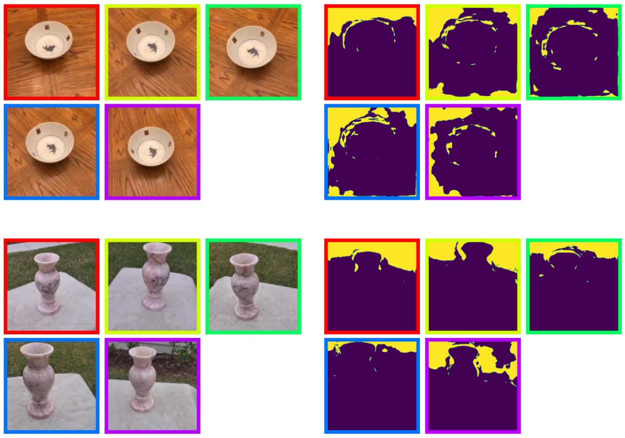

Figure 5. **Uncertainty Maps from Multiple DiffusionSfM Samples.** We can generate uncertainty maps for DiffusionSfM by running multiple inferences and then computing the variances of the predicted origins and endpoints. We find that the model is typically uncertain around depth boundaries and regions with low texture.

threshold (see Fig. 5).

5. Discussion

We present DiffusionSfM and demonstrate that it recovers accurate predictions of both cameras and geometry from multi-view input. Although our results are promising, several challenges and open questions remain. In particular, current pointmap prediction models such as DUSt3R are trained at large scale across multiple datasets. It would be interesting to similarly scale our training, with some careful consideration on how to sample sets of overlapping views as opposed to just pairs. Moreover, the compute requirement in multi-view transformers scales quadratically with the number of input images: one would require masked attention to deploy systems like ours for a large set of input images. Despite these challenges, we believe that our work highlights the potential of a unified approach for multi-view geometry tasks. We envision that our approach can be built upon to train a common system across related geometric tasks, such as SfM (input images with unknown origins and endpoints), registration (some images have known origins and endpoints whereas others don't), mapping (known rays but unknown endpoints), and view synthesis (unknown pixel values for known rays).

References

[1] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006. 2

[2] Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer. In *ECCV*, 2024. 2

[3] Ruojin Cai, Joseph Tung, Qianqian Wang, Hadar Averbuch-Elor, Bharath Hariharan, and Noah Snavely. Doppelgangers: Learning to disambiguate images of similar structures. In *ICCV*, 2023. 1

[4] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR*, 2018. 1, 2

[5] Bardienus Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jerome Revaud. Mast3r-sfm: a fully-integrated solution for unconstrained structure-from-motion, 2024. arXiv preprint arXiv:2409.19152. 2

[6] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Robust Dense Feature Matching. In *CVPR*, 2024. 2, 7

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2, 3

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

[9] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 2, 5

[10] Amy Lin, Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose++: Recovering 6d poses from sparse-view observations. In *3DV*, 2024. 2, 6, 7

[11] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-perfect structure-from-motion with featuremetric refinement. In *ICCV*, 2021. 2

[12] Philipp Lindenberger, Paul-Edouard Sarlin, Viktor Larsson, and Marc Pollefeys. Pixel-Perfect Structure-from-Motion with Featuremetric Refinement. In *ICCV*, 2021. 1

[13] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023. 2

[14] David G Lowe. Distinctive image features from scale-invariant keypoints. In *IJCV*, 2004. 2

[15] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision, 2023. arXiv preprint arXiv:2304.07193. 3, 5

[16] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 2, 3, 5, 4

[17] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 5, 4

[18] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *ICCV*, 2021. 2, 5, 6, 1

[19] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 6, 7

[20] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 1, 2

[21] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 2, 5

[22] Samarth Sinha, Jason Y Zhang, Andrea Tagliasacchi, Igor Gilitschenski, and David B Lindell. Sparsepose: Sparse-view camera pose regression and refinement. In *CVPR*, 2023. 2

[23] Cameron Smith, David Charatan, Ayush Tewari, and Vincent Sitzmann. Flowmap: High-quality camera poses, intrinsics, and depth via gradient descent. In *ECCV*, 2024. 2

[24] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM SIGGRAPH*, 2006. 2

[25] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 3

[26] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-Free Local Feature Matching with Transformers. In *CVPR*, 2021. 2

[27] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment network. In *ICLR*, 2019. 2

[28] Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *ICCV*, 2023. 2, 6, 7

[29] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision, 2024. arXiv preprint arXiv:2410.19115. 6, 7, 1, 4, 5

[30] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 1, 2, 6, 7, 5

[31] Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose: Predicting probabilistic relative rotation for single objects in the wild. In *ECCV*, 2022. 2

[32] Jason Y. Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Sparse-view pose estimation via ray diffusion. In *ICLR*, 2024. 2, 3, 6, 7, 1, 5

DiffusionSfM: Predicting Structure and Motion via Ray Origin and Endpoint Diffusion

Supplementary Material

Overview

The supplementary material includes sections as follows:

- Section A: Additional analysis on integrating Ray Diffusion [32] camera poses with MoGe [29] monocular depth estimates.
- Section B: More qualitative comparisons of predicted geometry and camera poses against baseline methods.
- Section C: Visualizations illustrating the effect of mono-depth diffusion guidance.
- Section D: Details and evaluation of the sparse-to-dense training strategy employed in DiffusionSfM.
- Section E: Inference details.
- Section F: More analysis of the homogeneous representation.
- Section G: Converting predicted ray origins and endpoints into camera poses.

A. Additional Analysis: Ray Diffusion + MoGe

We analyze whether the combination of an off-the-shelf sparse-view pose estimation method and a monocular depth estimation model is sufficient to infer the 3D geometry of scenes from multiple images. Here, we conduct an additional experiment that combines the state-of-the-art pose estimation method Ray Diffusion [32] with the monocular depth model MoGe [29].

Specifically, we adopt the predicted camera poses (intrinsics and extrinsics) predicted by Ray Diffusion and unproject image pixels using the mono-depth estimates from MoGe for each input image, essentially inferring the scene structure from multiple images. To minimize the scale difference for the predicted camera poses and depth to form a single consistent output, we follow these procedures: (1) We match the MoGe depth with the ground truth depth using a 1D optimal alignment (thus giving this baseline some privileged information). (2) We align the predicted camera centers from Ray Diffusion with ground truth cameras using an optimal similarity transform. (3) Finally, we unproject image pixels using the updated camera parameters and the aligned depth. We compare the predicted geometry of this approach with our method and DUSt3R [30] in Tab. 5. The results show that a naive combination of Ray Diffusion and MoGe yields poor Chamfer Distance, even though Ray Diffusion estimates relatively accurate focal length. This is because the MoGe depth estimates for different input views are inconsistent with each other. Therefore, to predict consistent 3D geometry from multiple images, the model must learn to reason over the entire set of views, rather than relying on mono-depth predictions from individual images. We also include visualizations of the predicted geometry for this approach in Fig. 6, where duplicated structures are observed due to significant pose errors or minor misalignment between views.

B. More Qualitative Comparisons

We include more qualitative comparisons with baselines on the predicted geometry (in Fig. 6) and camera poses (in Fig. 7).

Discussions. We show that DiffusionSfM can handle challenging input images where objects present highly symmetric patterns (*e.g.* the tennis ball example in Fig. 6 and the donut example in Fig. 7), while Ray Diffusion [32] and DUSt3R [30] fail to predict correct camera poses. Compared to Ray Diffusion, our approach leverages the prediction of *dense* scene geometry (*i.e.* pixel-aligned ray origins and endpoints) rather than relying on patch-wise "distance-agnostic rays." When compared to DUSt3R, despite being trained exclusively on CO3D [18], our model benefits from attending to all input images simultaneously and utilizing a diffusion framework to effectively manage the high uncertainties inherent to this task (see also Fig. 3 for examples where DUSt3R gives degraded results). Additionally, we observe that DUSt3R often predicts precise camera rotations but struggles with camera centers in many cases (*e.g.* the keyboard and backpack examples in Fig. 6). This observation aligns with our quantitative results for camera center evaluation, presented in Tab. 1.

C. Mono-Depth Diffusion Guidance Visualization

In Fig. 8, we visualize the impact of mono-depth diffusion guidance (Sec. 3.4). By utilizing the relatively precise and sharp monocular depth estimates from MoGe [29], we effectively reduce floaters along object boundaries and enhance the predicted geometry (see Tab. 4 for numerical results). Additionally, we compare mono-depth diffusion guidance with direct depth replacement in Fig. 9. Although replacing the predicted depth from DiffusionSfM with MoGe estimates produces cleaner geometry, it compromises multi-view consistency. For instance, this approach may result in artifacts such as multiple ground planes instead of maintaining a single consistent surface. To clarify, while diffusion guidance improves our method,

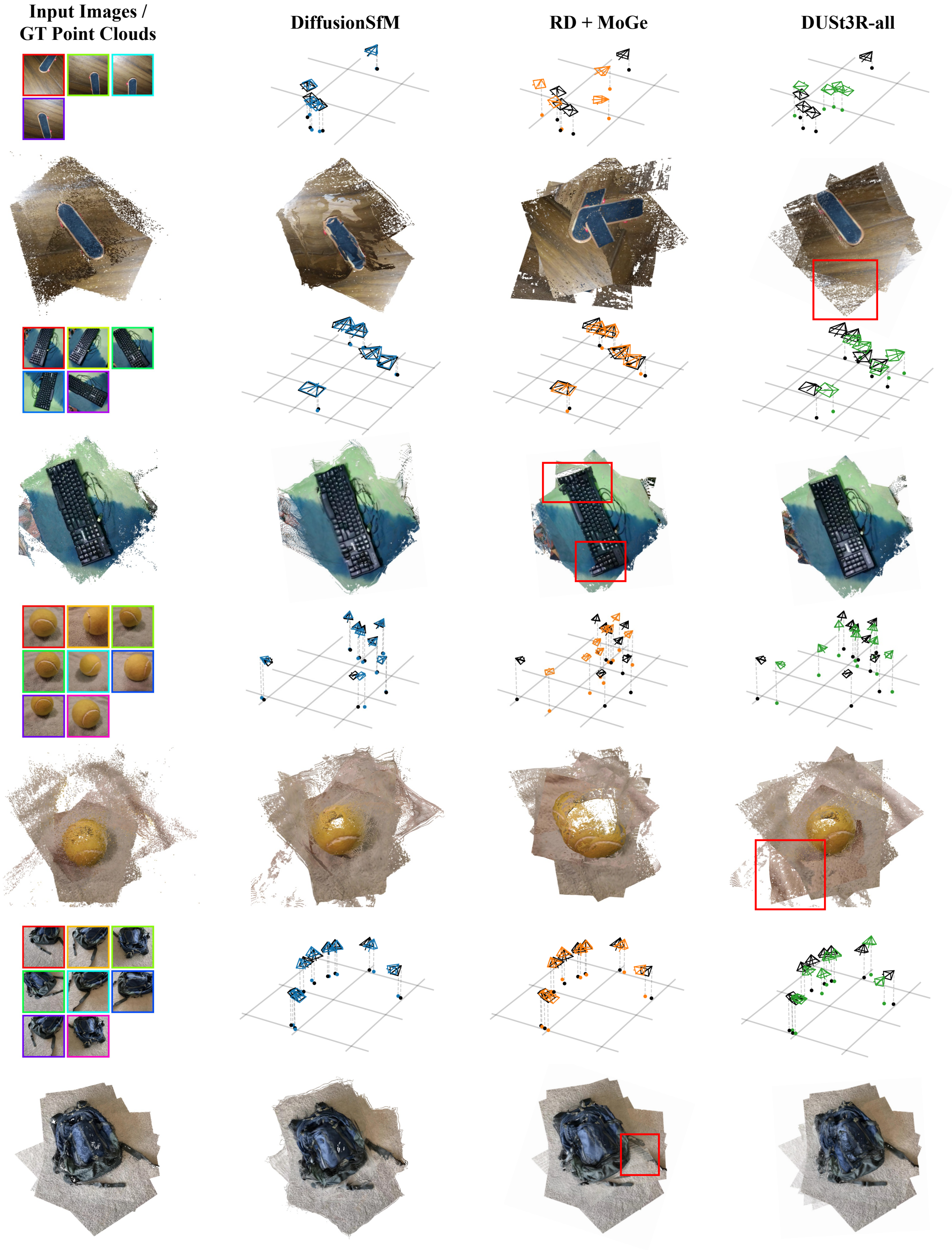

Figure 6. **More Qualitative Comparisons on Predicted Geometry and Camera Poses.** DiffusionSfM shows superior capabilities in handling challenging samples, *e.g*. the skateboard and tennis ball. Additionally, while we observe that DUSt3R-all can predict highly precise camera rotations, it often struggles with camera centers (see the keyboard and backpack examples).

we consistently use the *raw output* from DiffusionSfM for comparisons with other baselines and in our ablation studies.

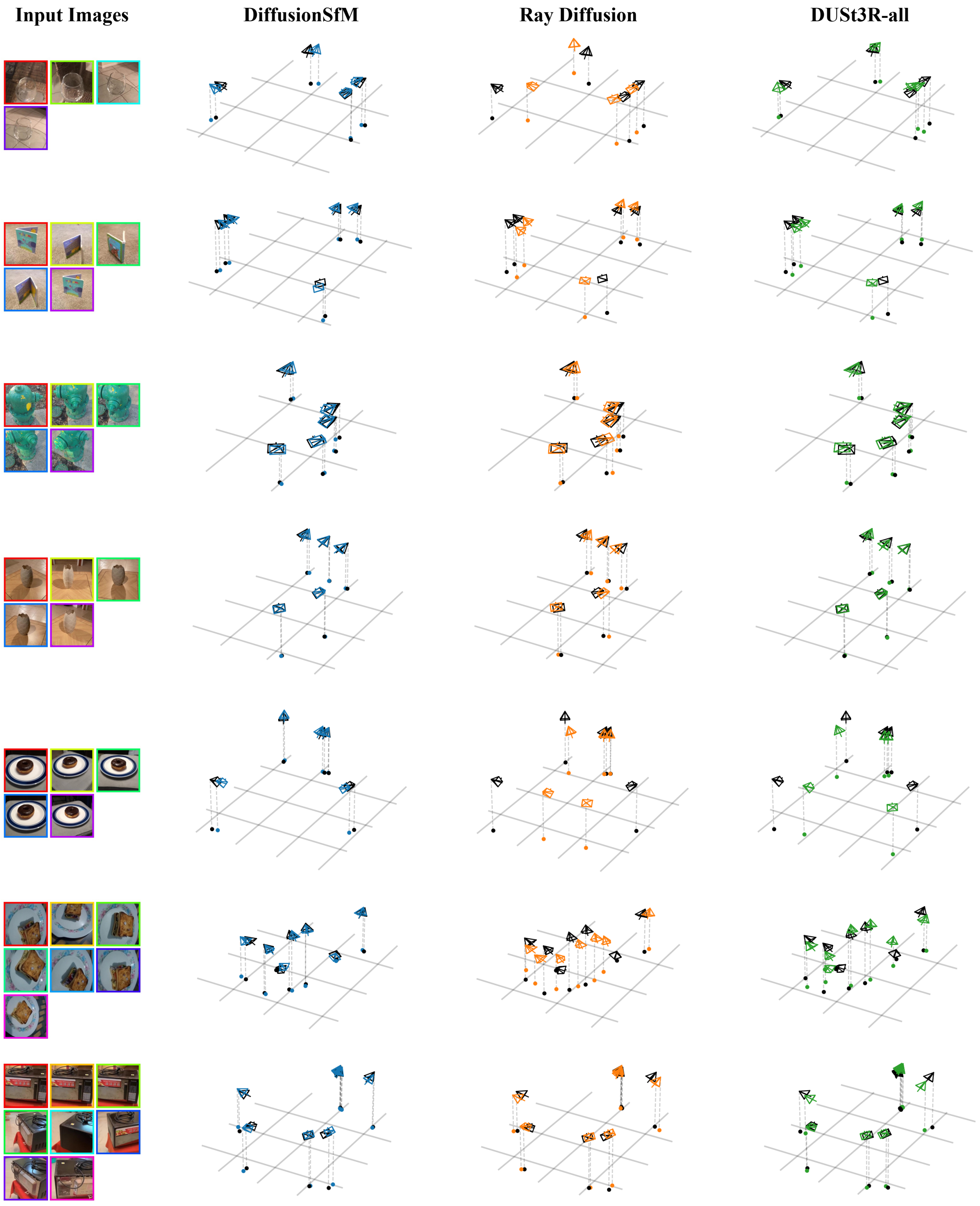

Figure 7. **More Qualitative Comparisons on Predicted Camera Poses.**

D. Sparse-to-Dense Training Details and Evaluation

As outlined in Sec. 3.3, we follow a sparse-to-dense strategy to train our model as we find that training the high-resolution model (*i.e.* dense model) from scratch yields sub-optimal performance. We visualize the output of the sparse model and dense model in Fig. 10. In the following, we introduce the details and resources to train DiffusionSfM.

Details. Our model leverages DINOv2-ViTs14 [15] as the feature backbone and takes 224×224 images as input. This results in 16×16 image patches, each with patch size 14.

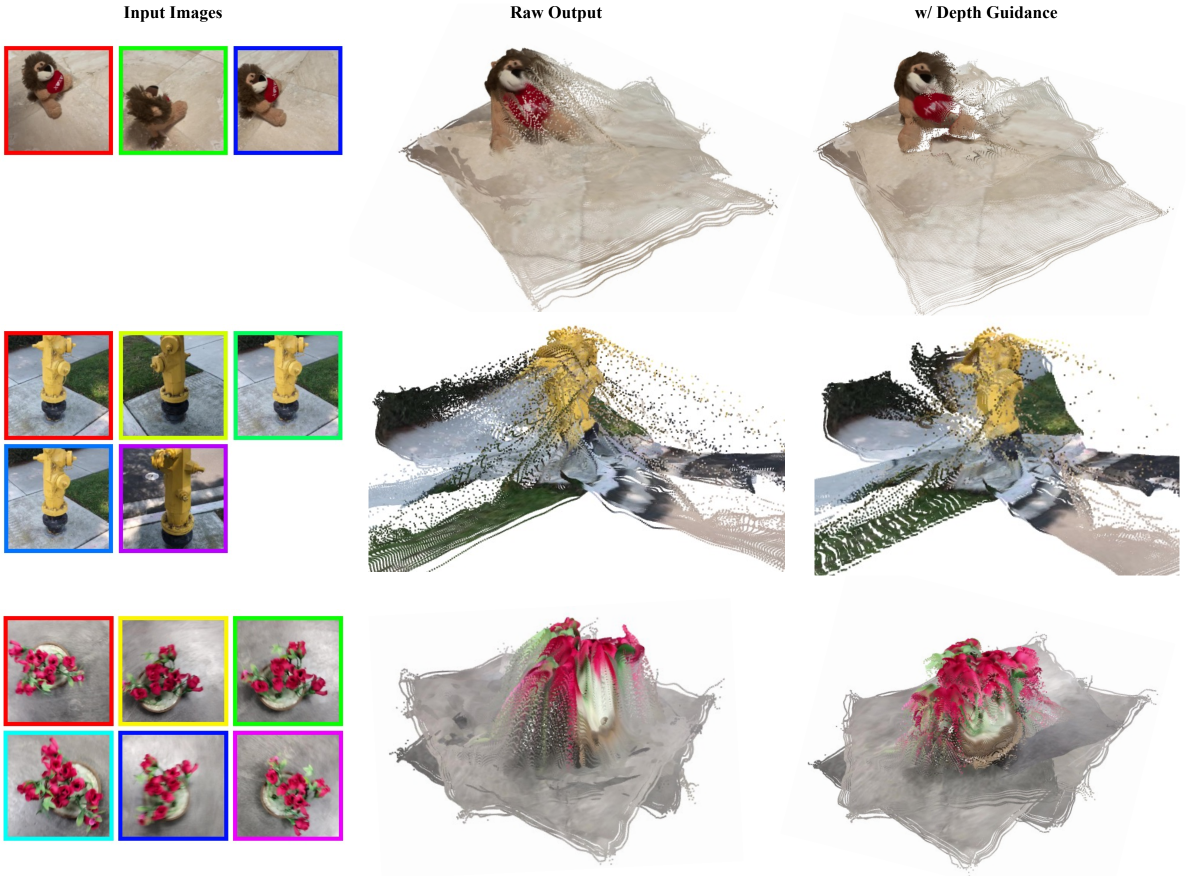

Figure 8. **Visualizations of the Effect of Mono-Depth Diffusion Guidance.** We utilize mono-depth estimates from MoGe [29] to guide the x_0-prediction from our model towards more accurate, clean estimates. This guidance enhances the quality of the predicted geometry while preserving multi-view consistency.

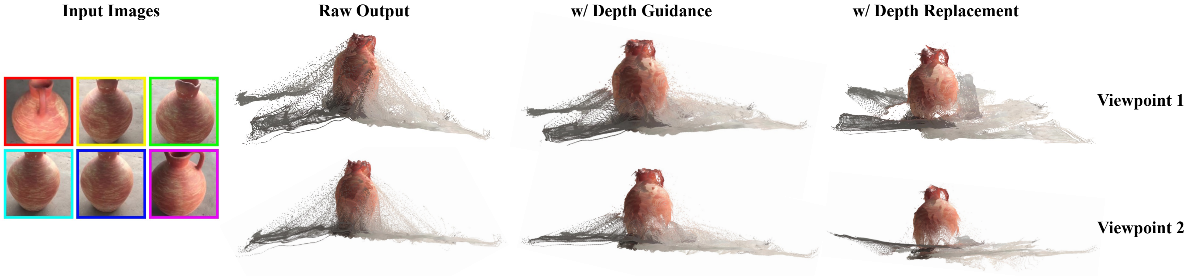

Figure 9. **Qualitative Comparison between Depth Diffusion Guidance and Direct Depth Replacement.** Replacing the predicted depth with MoGe estimates yields the cleanest results. However, this approach disrupts the multi-view consistency learned by DiffusionSfM, leading to artifacts such as the presence of multiple ground planes.

We first train a sparse model that outputs patch-wise (*i.e.* 16 × 16) ray origins and endpoints. Since the spatial resolution of the ground truth ray origins and endpoints for the sparse model aligns with the DINOv2 feature map, we use a single linear layer to embed the noisy ray origins and endpoints (without spatial downsampling), rather than a convolutional layer as shown in Eq. 5. We also remove the DPT [17] decoder in our sparse model. Subsequently, we initialize our dense model from the pre-trained sparse model to predict dense (*i.e.* 256 × 256) ray origins and endpoints. We copy-paste the DiT [16] weights from the sparse model. Whereas for the convolutional layer used to embed ray origins and endpoints, we duplicate the linear-layer weights by 16 × 16 (as the patch size of the conv-layer is 16) and then

	# of Images	2	3	4	5	6	7	8
CD	RD+MoGe [29, 32]	0.059	0.064	0.071	0.062	0.063	0.061	0.061
	DUSt3R-CO3D [30]	0.036	0.037	0.040	0.040	0.037	0.036	0.039
	DUSt3R-all [30]	0.021	0.023	**0.024**	0.024	0.025	0.025	**0.023**
	DiffusionSfM	**0.020**	**0.022**	**0.024**	**0.023**	**0.022**	**0.023**	**0.023**
CD w/ Mask	RD+MoGe [29, 32]	0.071	0.075	0.068	0.067	0.066	0.064	0.064
	DUSt3R-CO3D [30]	0.038	0.036	0.036	0.036	0.034	0.033	0.034
	DUSt3R-all [30]	**0.023**	**0.022**	**0.019**	**0.020**	**0.019**	**0.020**	**0.020**
	DiffusionSfM	0.031	0.027	0.026	0.026	0.026	0.025	0.026
Focal Length	Ray Diffusion [32]	0.705	0.709	0.716	0.717	0.720	0.724	0.723
	DUSt3R-CO3D [30]	0.679	0.672	0.674	0.673	0.673	0.671	0.668
	DUSt3R-all [30]	0.589	0.594	0.595	0.599	0.603	0.600	0.597
	DiffusionSfM	**0.754**	**0.753**	**0.750**	**0.750**	**0.749**	**0.752**	**0.753**

Table 5. **Evaluation of the Predicted Geometry and Focal Length on CO3D Unseen Categories.** Top: Chamfer Distance (CD) computed over all scene points. Middle: CD computed on foreground points only. Bottom: Percentage of predicted focal lengths within 15% of the ground truth. RD+MoGe refers to the Ray Diffusion camera pose, with depth estimates from MoGe aligned to the ground truth. DUSt3R-CO3D is trained solely on CO3D, while DUSt3R-all is trained on multiple datasets. DiffusionSfM outperforms all methods in terms of full scene geometry and estimated focal length, and also outperforms both RD+MoGe and DUSt3R-CO3D on foreground geometry.

		Rotation Accuracy (↑, @ 15°)							Center Accuracy (↑, @ 0.1)						
	# of Images	2	3	4	5	6	7	8	2	3	4	5	6	7	8
Seen	Sparse Model	88.8	89.6	89.8	90.1	90.0	90.3	90.2	100	93.7	89.6	87.2	85.5	84.5	83.4
	Dense Model (1)	84.9	84.1	83.8	84.3	84.2	84.2	83.7	100	92.5	87.4	84.1	81.8	79.8	76.9
	Dense Model (2)	92.4	93.0	93.3	93.5	93.6	93.8	93.8	100	95.2	92.1	90.5	89.2	88.7	87.8
Unseen	Sparse Model	82.5	84.2	85.2	86.1	86.5	86.6	86.6	100	87.9	81.5	77.9	75.9	73.8	72.7
	Dense Model (1)	77.8	79.0	79.6	80.5	80.5	80.7	79.9	100	86.2	78.6	74.1	71.5	68.6	66.1
	Dense Model (2)	90.1	91.0	91.8	92.6	92.9	93.0	93.1	100	90.9	85.7	83.7	82.4	80.9	80.7

Table 6. **Camera Rotation and Center Accuracy on CO3D at Different Training stages.** On the left, we report the proportion of relative camera rotations within 15° of the ground truth. On the right, we report the proportion of camera centers within 10% of the scene scale. To align the predicted camera centers to ground truth, we apply an optimal similarity transform (s, $\mathbf{R}$, $\mathbf{t}$). Hence the alignment is perfect at $N = 2$ but worsens with more images.

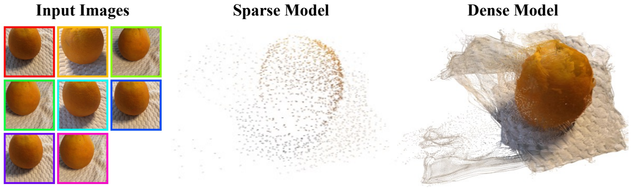

Figure 10. **Qualitative Comparison of Sparse and Dense Model Outputs.** The sparse model predicts the ray origin and endpoint for each image patch, limiting its ability to capture the fine-grained details of the scene.

divide them by 256 to account for the patch-wise addition. While the DiT in the dense model has learned meaningful representations, the DPT decoder is initialized from scratch. To avoid breaking the learned DiT weights in early training iterations, we freeze its weights while only training the convolutional embedding layer and the DPT decoder for a few iterations. After that, we train the whole model together, including the DINOv2 encoder as well (which was frozen in the previous stage). We compare the performance of DiffusionSfM at each stage in Tab. 6.

Training Resources. We train our sparse model on 8 RTX A5000 GPUs for 400,000 iterations, which takes approximately 5 days. To "warm up" our dense model, we freeze its DiT weights and train it for 55,000 iterations on 6 RTX A6000 GPUs, requiring an additional 16 hours. Finally, we unfreeze the entire model and continue training for 430,000 iterations on 4 H100 GPUs, which takes roughly 3 days.

E. Inference Details

DiffusionSfM utilizes x_0-parameterization to predict the clean ray origin and endpoint map as the model output, em-

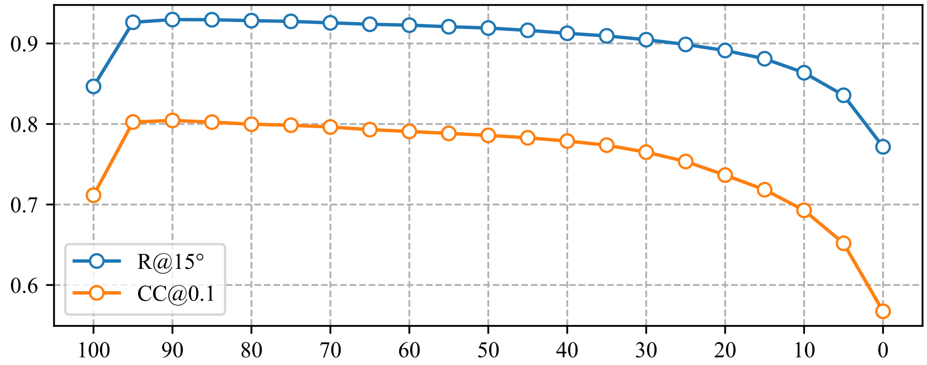

Figure 11. **Performance of x_0-Prediction Across Diffusion Denoising Timesteps (N = 8).** The X-axis represents the diffusion denoising timesteps, with $T = 100$ indicating predictions starting from Gaussian noise and $T = 0$ corresponding to the clean sample. The Y-axis shows the accuracy for camera rotation (blue) and camera center (orange). Notably, DiffusionSfM achieves peak performance at $T = 90$. As a result, in inference, we perform only 10 diffusion steps, significantly improving inference speed.

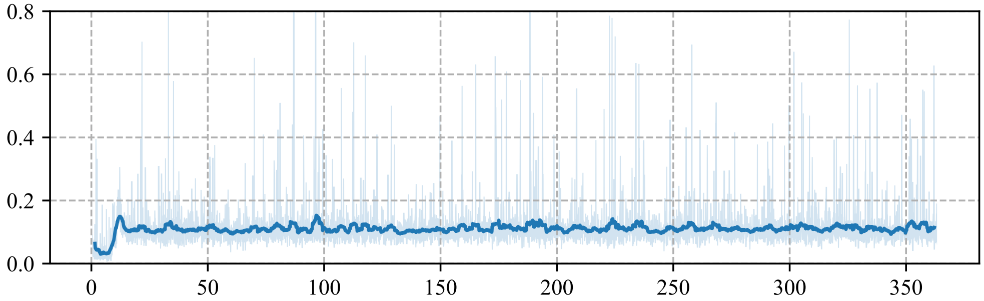

Figure 12. **Training Loss Curve for DiffusionSfM without Homogeneous Representation.** The X-axis represents training iterations (in thousands, k), and the Y-axis denotes the loss value. Without incorporating a homogeneous representation for ray origins and endpoints, the model struggles to train effectively due to significant scale differences across various scene components.

ploying 100 diffusion denoising timesteps. In Fig. 11, we evaluate the accuracy of x_0-prediction at each timestep with eight input images. Interestingly, we find that DiffusionSfM achieves its most accurate clean sample predictions at an early timestep ($T = 90$), rather than at the final denoising step ($T = 0$). This observation remains consistent across different numbers of input images (Zhang *et al.* [32] also have a similar observation that early stopping helps improve performance). To capitalize on this property, we limit inference to 10 denoising steps and use the x_0-prediction at $T = 90$ as the final output, significantly reducing inference time. Throughout our experiments, we consistently adopt this approach to report all results presented in our paper.

F. The Effect of Homogeneous Representation

To underscore the importance of the proposed homogeneous representation for ray origins and endpoints, we train a variant of DiffusionSfM using these components directly in $\mathbb{R}^3$ (*i.e.* without using homogeneous coordinates). For this model, we employ a scale-invariant loss function, as used in DUSt3R [30]. The training loss curve for this model is shown in Fig. 12. Notably, the model fails to converge, with the training loss remaining persistently high. This failure occurs because our diffusion-based approach assumes input data within a reasonable range, as the Gaussian noise added during training has a fixed standard deviation of 1. Consequently, training scenes with substantial scale differences across components disrupt the model's learning process. In contrast, employing homogeneous coordinates enables the normalization of the input data to a unit norm, which not only stabilizes training and facilitates convergence but also provides an elegant representation of unbounded scene geometry.

G. Converting Ray Origins and Endpoints to Camera Poses

The camera centers for each input image are recovered by averaging the corresponding predicted ray origins. To determine camera rotations and intrinsics, we follow the method proposed by Zhang *et al.* [32], which involves solving for the optimal homography that aligns the predicted ray directions with those of an identity camera. For additional details, we refer readers to Zhang *et al.* [32].