Pacific Graphics 2025
M. Christie, N. Pietroni, and Y.-S. Wang
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 44 (2025), Number 7

LucidFusion: Reconstructing 3D Gaussians with Arbitrary Unposed Images

Hao He[†1,2], Yixun Liang[†1], Luozhou Wang[1], Yuanhao Cai[3], Xinli Xu[1], Haoxiang Guo[4], Xiang Wen[4], and Yingcong Chen[‡1,2]

[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]The Hong Kong University of Science and Technology
[3]Johns Hopkins University [4]SkyWork AI

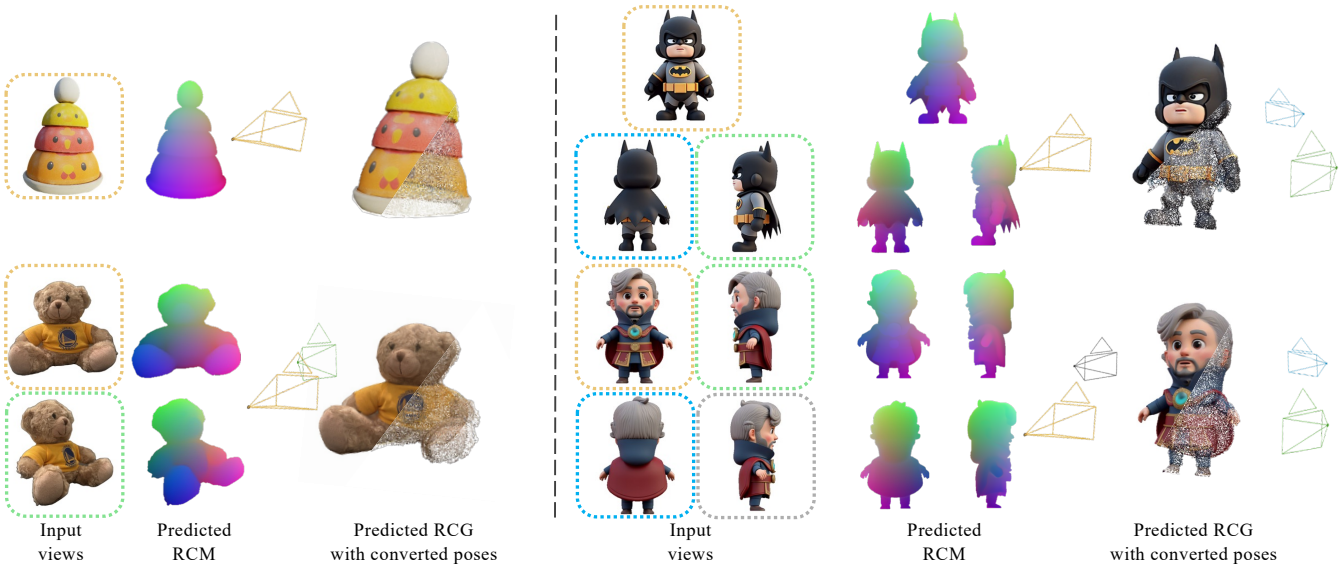

Figure 1: *LucidFusion utilizes Relative Coordinate Gaussian (RCG) representation to achieve 3D reconstruction with pose estimation from unposed, sparse and arbitrary numbers of input views in a feed-forward manner.*

Abstract

Recent large reconstruction models have made notable progress in generating high-quality 3D objects from single images. However, current reconstruction methods often rely on explicit camera pose estimation or fixed viewpoints, restricting their flexibility and practical applicability. We reformulate 3D reconstruction as image-to-image translation and introduce the Relative Coordinate Map (RCM), which aligns multiple unposed images to a "main" view without pose estimation. While RCM simplifies the process, its lack of global 3D supervision can yield noisy outputs. To address this, we propose Relative Coordinate Gaussians (RCG) as an extension to RCM, which treats each pixel's coordinates as a Gaussian center and employs differentiable rasterization for consistent geometry and pose recovery. Our LucidFusion framework handles an arbitrary number of unposed inputs, producing robust 3D reconstructions within seconds and paving the way for more flexible, pose-free 3D pipelines.

CCS Concepts

• ***Computing methodologies*** → ***3D imaging; Reconstruction;***

1. Introduction

Digital 3D objects are increasingly essential in a variety of domains, facilitating immersive visualization, analysis, and interaction with objects and environments that closely mimic real-world

† Equal Contribution
‡ Corresponding Author

experiences. These objects are foundational in fields such as architecture, animation, gaming, and virtual and augmented reality, with broad applications across industries like retail, online conferencing, and education. Despite their growing demand, producing high-quality 3D content remains a resource-intensive task, requiring substantial time, effort, and domain expertise. This challenge has catalyzed the rapid advancement of 3D content generation techniques [MST*21,WWX*21,HLX*23,HZG*23,ZYG*24,HST*24, LYL*24,WLW*24], including methods that reconstruct 3D objects from one or more input images.

Recently, 3D reconstruction methods [HZG*23, TPL*24, ZYG*24] have gained considerable attention, as they can convert single or multiple images, either captured by an external device or generated by diffusion models, into complete 3D objects in content generation workflows. However, these methods inevitably require camera pose as an intermediate step to map image features into 3D: whether explicitly, as in traditional MVS-based methods [YLL*18, CXZ*21], or implicitly, as in LRM-based approaches [HZG*23]. Yet, obtaining accurate poses of the input views is a non-trivial task: current methods often rely on external pose estimation pipelines (e.g., COLMAP [SF16]) or fix the input viewpoint [TCC*24], substantially constraining both the flexibility of the reconstruction process and user experience.

This observation raises a critical problem: *Can we mitigate the pose requirement for 3D reconstruction?* By revisiting the reconstruction problem (detailed in Sec. 3.1), we find that the wrapping from 2D to 3D can be learned via an image-to-image translation approach, if we leverage an intermediate representation such as Canonical Coordinate Maps (CCM) [LCCT23, WWC*24], we can bypass common challenges associated with pose estimation, allowing a more flexible 3D reconstruction pipeline. However, in practice, CCMs are difficult to regress because orientation cues are only implicitly embedded in the color space and such "orientation" information is not well-defined, as shown in Fig. 2. To address this shortcoming, we propose the *Relative Coordinate Map (RCM)*, which transforms each pixel to the camera space of a selected "main" camera (e.g., the first frame in our system), as shown in Fig. 1. This simple yet effective modification retrains CCM's advantage of end-to-end learnability via an image-to-image framework, while mitigating the ambiguities that arise from implicitly encoded orientation information.

Nevertheless, we observe that naively performing this mapping often results in inconsistent and noisy outputs as shown in Fig. 4, primarily due to the lack of 3D prior supervision. To address this limitation, we further introduce *Relative Coordinate Gaussians (RCG)*, interpreting each pixel's coordinates as the center of a Gaussian. The RCG extension allows differentiable rasterization from arbitrary viewpoints, and can be supervised by ground-truth images rather than solely by per-view coordinate predictions. This additional supervision resolves the noise and misalignment issues that arise under purely RCM training. By re-framing the multi-view reconstruction problem as an image-to-RCG transformation, we can efficiently obtain a complete 3D representation from arbitrary, unposed images, as shown in Fig. 1. Furthermore, since RCG is inherently a 3D representation, it eliminates the common challenge associated with pose estimation and can directly recover

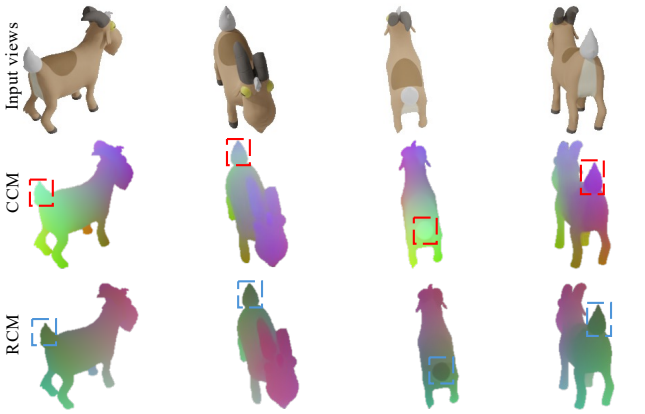

Figure 2: *Pilot study. We compare CCM and RCM given a set of input images. CCM fails to maintain consistency across different input views (red box), while RCM successfully maintains the 2D-3D relation (blue box).*

camera poses. This feature is often missing from other feedforward, Gaussian-based methods. In summary, our contributions are threefold:

- We revisit the reconstruction problem and identify the gap in existing CCM approaches, leading to the proposed RCM and its RCG extension.
- We develop a system, *LucidFusion*, that efficiently maps images to RCG, embedding pixel-wise correspondences across different views into a "main" view and eliminating explicit pose estimation.
- We showcase the superior quality and flexibility of our method, enabling rapid 3D reconstruction and pose estimation within seconds.

2. Related Work

2.1. Multi-View 3D Reconstruction

Multi-view 3D reconstruction typically relies on multi-view stereo (MVS), which reconstructs the visible surface of an object by triangulating between multiple views. MVS-based methods can be broadly classified into three categories: depth map-based methods [CVHC08, SZFP16, CBZ*22, RWZ*23, LHC24], voxel grid-based methods [KS00, YLL*19, CXZ*21], and point cloud-based methods [FP09, CHXS19]. These methods generally operate by taking multi-view images and constructing a 3D cost volume through the unprojection of 2D multi-view features into plane sweeps. However, they all depend on the availability of camera parameters with the input multi-view images, either provided during data acquisition or estimated using Structure-from-Motion (SFM) [SF16,JCT13] for in-the-wild reconstructions. Consequently, these methods often fail when handling sparse-view inputs without known camera poses. In contrast, our approach leverages the RCM representation, enabling 3D generation from uncalibrated and unposed sparse inputs, thereby offering a robust solution for real-world applications.

2.2. Radiance Field Reconstruction

Neural radiance fields (NeRF) [MST*21] have recently driven significant advancements in radiance field methods, achieving state-

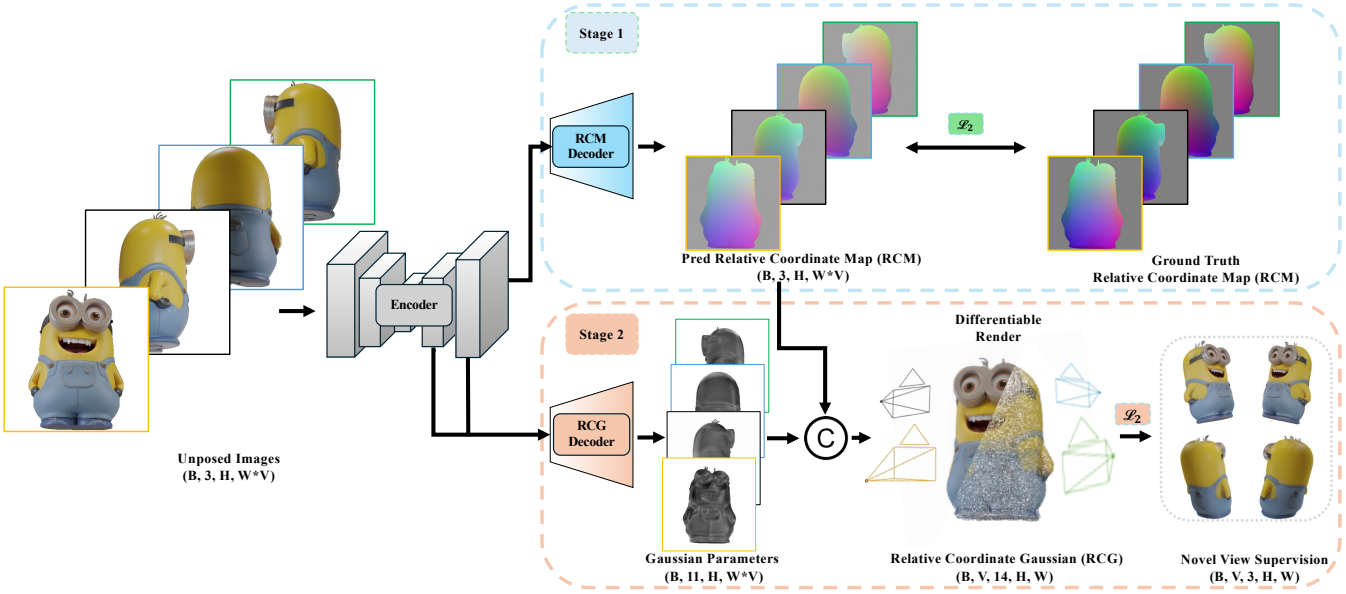

Figure 3: *Pipeline Overview of LucidFusion. Our framework processes a set of sparse, unposed multi-view images as input. The model predicts the RCM representation for the input images. Additionally, the feature map from the final layer of the encoder is fed into a decoder network to extend the RCM representation to RCG. The RCG is then rendered at novel views and supervised with ground truth images.*

of-the-art performance [CXZ*21, WWG*21, GHZ*23]. These approaches optimize radiance field representations through differentiable rendering, diverging from traditional MVS pipelines, yet they still rely on dense sampling for precise reconstruction. To address sparse-view challenges in NeRF, recent works have incorporated regularization terms [NBM*22, WCLL23] or leveraged geometric priors [CXZ*21, YPW23]. However, these methods continue to require image samples with known camera poses. Another research direction explores SDS-based optimization techniques, distilling detailed information from 2D diffusion models into 3D representations [PJBM22, WLW*24, LYL*24], which enables the rendering of high-fidelity scenes but requires lengthy optimization for each individual scene. In contrast, our approach eliminates the need for known camera poses and operates in a feed-forward manner, supporting generalizable 3D generation without extensive optimization.

2.3. Unconstrained Reconstruction

Large Reconstruction Model (LRM) [HZG*23] introduced a triplane-based approach combined with volume rendering, demonstrating that a regression model can robustly predict a neural radiance field from a single-view image and thus reduce dependence on camera poses. Subsequent works [LLZ*23, SCZ*23, SWY*23, XTL*23, TCC*24, ZBT*24] have leveraged diffusion models to extend single-view inputs to multi-view inputs, bypassing the need for camera poses. However, many of these approaches rely on fixed viewpoints (e.g., *front, back, left, right*), limiting their applicability in real-world scenarios.

Another line of research explores pose-free 3D reconstruction using uncalibrated images as direct input. Several approaches [LZRT24, JJGZ24] regress camera poses through network predictions, while PF-LRM [WTB*23] adapts LRM by incorporating a differentiable PnP module to predict poses from multi-view images for 3D reconstruction. iFusion [WCS*23] leverages Zero123 [LXJ*24] predictions within an optimization-based pipeline to align poses. SpaRP [XLC*24] employs a coordinate-map representation with a generative diffusion model but relies on an additional PnP solver for refinement and is limited to no more than 6 views. Dust3R [WLC*24] and MASt3R [LCR24] generate point cloud representations and showcase outstanding 3D reconstruction performance from multi-view images. However, both of them are only generalizable in local pair-wise views and require global alignment to merge the point clouds. In contrast, our regression-based method accommodates an arbitrary number of unposed inputs, providing a more efficient rendering pipeline while maintaining high-quality results for practical 3D reconstruction.

3. Method

LucidFusion is a feed-forward 3D reconstruction model that processes one to N unposed images, recovering pose and object Gaussians. In Sec. 3.1, we first examine how existing reconstruction models are formulated. Building on these insights, Sec. 3.2 introduces the *Relative Coordinate Map (RCM)*, a novel representation directly regressed from input images that enables pose estimation and 3D reconstruction without explicit pose information. Sec.3.3 extends RCM into *Relative Coordinate Gaussians (RCG)* via 3D Gaussian Splatting [KKLD23], enforcing global 3D consistency through a rendering loss. Finally, Sec. 3.4 presents our two-stage training strategy for efficient 3D reconstruction.

3.1. Preliminary

Extending a reconstruction pipeline from a single image to multiple images introduces several challenges. We abstract the 3D reconstruction problem as a mapping task: with a single image, the

Figure 4: *Visualization of stage 1 and stage 2 results.*

primary goal is to extract geometric information for object generation, whereas with multiple images, both mapping and scaling issues arise. This mapping can be performed explicitly, as in traditional MVS-based methods [YLL*18,CXZ*21], or implicitly, as in LRM-based approaches [HZG*23]. However, both strategies typically rely on pose estimation, where images must either be preposed or restricted to specific viewpoints, limiting the pipeline's flexibility. In contrast, we propose a method that performs the mapping end-to-end without relying on explicit pose information.

We argue that a key challenge in multi-view reconstruction is ensuring consistent geometric feature estimation across different viewpoints, while also preserving scale-wrapping relationships. From this perspective, pose is merely an intermediate variable that performs the mapping. If pose information is embedded in the regression objective itself, it can be bypassed, thereby improving overall usability and reducing the pipeline's complexity.

Building on this idea, *Canonical Coordinate Map (CCM)* [LCCT23] represents a natural approach by embedding pose information directly into an image's pixel values. However, when regressing CCM from multi-view inputs, the model must operate under a world-coordinate convention and therefore simultaneously infer both orientation and geometry. This limitation becomes evident in our pilot study, where we regress a model using CCM (see the middle row of Fig. 2): the same object parts—such as a sheep's head and tail—should retain consistent colors across all views. This semantic information is crucial for indicating an object's orientation in world space. Any misalignment suggests that the model fails to accurately align the 2D multi-view inputs in 3D space.

3.2. Relative Coordinate Map

For a reconstruction task, however, it is more important to maintain 3D consistency across input views than to learn an object's canonical orientation. Hence, we propose the *Relative Coordinate Map (RCM)*, which transforms each view's coordinates to align with the coordinate system of a selected "main" view. As shown in the bottom row of Fig. 2, this transformation resolves orientation ambiguities in our pilot study, making it more suitable for the reconstruction task.

Let $\{\boldsymbol{I}_i\}_{i=1}^{N}$ be a set of N input images, each $\boldsymbol{I}_i \in \mathbb{R}^{H\times W\times 3}$. We define *RCM* for each image as $\boldsymbol{M}_i \in \mathbb{R}^{H\times W\times 3}$, where $\boldsymbol{M}_i$ contains the 3D coordinates corresponding to each pixel in $\boldsymbol{I}_i$. To help the model learn these coordinates from arbitrary viewpoints, we project all N images into the coordinate system of a randomly chosen input view. This random selection encourages generalization of different viewpoints.

Concretely, for each input view, we have a camera pose $\boldsymbol{P}_i \in \mathbb{R}^{4\times 4}$ and an intrinsic matrix $\boldsymbol{K} \in \mathbb{R}^{4\times 4}$ (both in homogeneous form), as well as a depth map $\boldsymbol{D}_i \in \mathbb{R}^{H\times W}$. We then randomly select one of these poses, $\boldsymbol{P}_{main}$, as the main camera pose. We define the main camera's RCM as:

$$\boldsymbol{M}_{main} = \boldsymbol{P}_{main}\boldsymbol{P}_{main}^{-1}\boldsymbol{K}^{-1} * \boldsymbol{D}_{main}, \tag{1}$$

which simplifies to

$$\boldsymbol{M}_{main} = \boldsymbol{K}^{-1} * \boldsymbol{D}_{main}, \tag{2}$$

within its own camera coordinate frame. For the remaining $N-1$ views, we transform each one into the main camera's coordinates:

$$\boldsymbol{M}_j = \boldsymbol{P}_{main}\boldsymbol{P}_j^{-1}\boldsymbol{K}^{-1} * \boldsymbol{D}_j, \quad j = 1,2,3,\ldots,N-1, \tag{3}$$

with the RCM values constrained to $[-1,1]$. To further enforce 3D consistency across multiple views, we concatenate all input images along the width dimension W, allowing the model to use self-attention to integrate multi-view information.

The RCM representation offers several key advantages. First, as an *image-based* representation, it benefits from pre-trained foundation models, thereby simplifying the learning process. Second, RCM preserves a one-to-one mapping between image pixels and their corresponding 3D points, effectively capturing the geometry as a point cloud. Finally, since each RCM explicitly represents the position (x,y,z) of every pixel, we can compute the pose ξ_i for each view M_i using a standard Perspecitive-n-Point (PnP) solver [Tea25], enabling relative pose estimation.

3.3. Relative Coordinate Gaussians

Building on relative coordinate maps, one could train a 2D image-to-image model directly for unconstrained 3D reconstruction. However, we observe that naively performing this mapping often results in inconsistent and noisy outputs as shown in Fig. 4, primarily due to the lack of 3D prior supervision that is crucial for maintaining 3D consistency. To address this, we integrate 3D Gaussians [SRV24] with the relative coordinate map, forming what we call the *Relative Coordinate Gaussians (RCG)*.

Specifically, we take the relative coordinates as the center of each Gaussian point. Beyond simply regressing the 3D position, we also regress the Gaussian parameters. Since the RCG is pixel-aligned, we can seamlessly expand the network's output channels from 3 to 14. These additional channels encode the scale $\boldsymbol{s}$ (3 channels), the rotation quaternion $\boldsymbol{rot}$ (4 channels), and the opacity $\boldsymbol{\sigma}$ (1 channel). With these Gaussian parameters, we employ differentiable rasterization from arbitrary viewpoints, supervised by ground-truth images rather than solely by per-view coordinate predictions. This global rendering loss enforces consistency across views and yields smoother, more coherent reconstructions, as shown in Fig. 4.

3.4. Two Stage Training

We observe that jointly optimizing both the Relative Coordinate Map (RCM) and the rendering objective often leads to training instability. As illustrated in Fig. 5, the network fails to localize the object geometry accurately and maintain multi-view consistency, resulting in misalignments or empty holes of the object. This arises because the model must simultaneously reason about per-pixel alignment and global 3D consistency, creating conflicting objectives during training. To overcome this challenge, we adopt a two-stage training scheme. In Stage 1, we train the network on the RCM representation and using stable diffusion–based prior similar to [HLY*24], enabling it to learn robust mappings from the input images to the RCM. In Stage 2, we expand the learned RCM into the RCG representation and incorporate a differentiable rendering loss to enforce 3D consistency. By decoupling these learning stages, we alleviate the tension between local pixel alignment and global geometry constraints, substantially stabilizing the training process.

3.4.1. Stage 1 training

We try to train a network to learn RCM representation. Let E be the network mapping N RGB images $\{\mathbf{I}_i\}_{i=1}^{N}$, where $\mathbf{I}_i \in \mathbb{R}^{H\times W\times 3}$, to their corresponding RCMs $\mathbf{M}_i \in \mathbb{R}^{H\times W\times 3}$. Formally,

$$\hat{\mathbf{M}}_i = E(\mathbf{I}_i), \quad i = 1,2,3,\ldots,N. \tag{4}$$

We obtain ground truth RCMs from Eq. 3 and supervise the predicted RCMs $\hat{\mathbf{M}}_i$ via MSE loss:

$$\mathcal{L}_{rcm} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{MSE}(\hat{\mathbf{M}}_i, \mathbf{M}_i). \tag{5}$$

After Stage 1, the network E serves as a base model that reliably transforms input images into RCMs.

3.4.2. Stage 2 training

We then extend the output layer to introduce RCGs as Sec. 3.3. Specifically, We extract an intermediate feature map $\mathbf{f}_i \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times l}$ from E, which is passed to a decoder G to predict the 14-channel RCGs $\mathbf{\Theta}_i$:

$$\mathbf{\Theta}_i = G(\mathbf{f}_i), \tag{6}$$

$$\mathbf{\Theta}_i = (\hat{\mathbf{M}}_i, \mathbf{I}_i + \boldsymbol{\delta}_i^c, \mathbf{s}_i, \mathbf{rot}_i, \boldsymbol{\sigma}_i). \tag{7}$$

We render I_i supervision views using a differentiable renderer [KKLD23], and supervise it with its ground-truth view $\mathbf{I}_i$. To enforce visual fidelity, we adopt a combination of MSE loss, SSIM loss from [KKLD23], and VGG-based LPIPS loss [ZIE*18]:

$$\begin{aligned}\mathcal{L}_{rgb} = (1-\lambda)\mathcal{L}_{MSE}(\hat{\mathbf{I}}_i, \mathbf{I}_i) \\ + \lambda\mathcal{L}_{SSIM}(\hat{\mathbf{I}}_i, \mathbf{I}_i) \\ + \mathcal{L}_{LIPIS}(\hat{\mathbf{I}}_i, \mathbf{I}_i),\end{aligned} \tag{8}$$

where $\lambda = 0.2$, following [KKLD23]. To further accelerate convergence and enhance object boundaries, we also apply an MSE loss to the alpha channel [TCC*24]:

$$\mathcal{L}_{\alpha} = \mathcal{L}_{MSE}(\hat{\mathbf{I}}_i^{\alpha}, \mathbf{I}_i^{\alpha}). \tag{9}$$

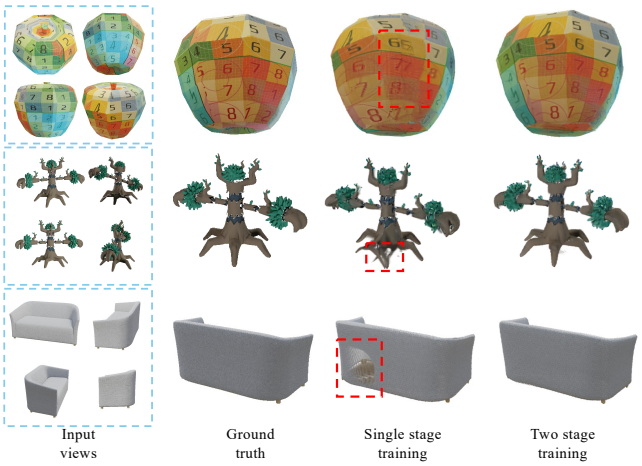

Figure 5: *Comparison with single and two-stage training. For single stage, the model struggles to predict Gaussian locations.*

Thus, the overall loss for Stage 2 is given by

$$\mathcal{L} = \mathcal{L}_{rgb} + \mathcal{L}_{\alpha}. \tag{10}$$

3.4.3. Pose Estimation

As we discussed, since the center of RCG is defined as the spatial coordinates of each pixel, we can estimate the camera pose by minimizing the reprojection error of 3D–2D point correspondences. Assume $\mathbf{q}_{i,j}$ represents 3D point location (x,y,z) in RCM view i, and $\mathbf{p}_{i,j}$ represents 2D pixel location at j of the RCM view i, we have:

$$\xi_i = argmin\sum_{j=1}^{N}||Proj(R_i\cdot\mathbf{q}_{i,j}+t_i) - \mathbf{p}_{i,j}||^2, \tag{11}$$

where R_i, t_i are the rotation and translation matrix, and N represents number of pixels in each of the RCM M_i. We use the RANSAC scheme in OpenCV [Tea25] and filter out non-informative white background points from affecting the pose prediction. We present these results in Sec. 4.2.

4. Experiment

In this section, we first explain our training and testing datasets in Sec. 4.1. We then make both quantitative and qualitative comparisons against different baselines in Sec. 4.2. Finally, we explain our design choice in Sec. 4.3.

4.1. Experimental Setting

4.1.1. Datasets

We train our model on a subset of Objaverse [DSS*23] dataset, as there are many low-quality 3D shapes in the original set. The final training data contains approximately 98K 3D objects. For each 3D object, we generate a total of 90 views with different elevations. During training, N views are randomly sampled from these 90 images. The rendered images have a resolution of 512×512 and are generated under uniform lighting conditions.

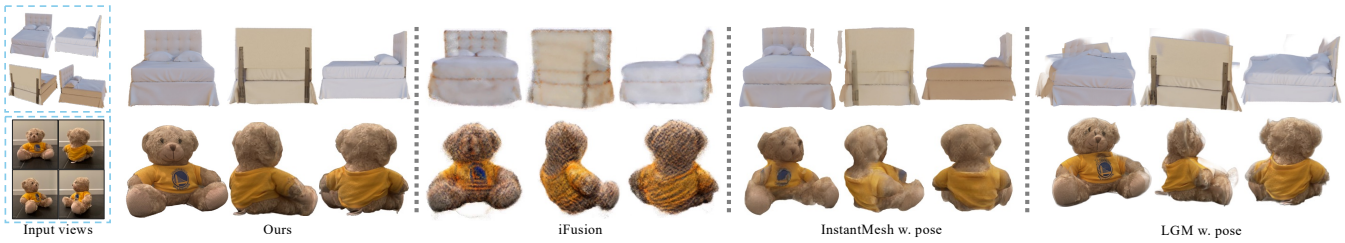

Figure 6: *Qualitative comparison with iFusion [WCS*23], InstantMesh [XCG*24] and LGM [TCC*24] under sparse view setting.*

Figure 7: *Qualitative comparison with InstantMesh [XCG*24], CRM [WWC*24] and LGM [TCC*24] under standard single-image-to-3D paradim.*

4.1.2. Stage 1

Similar to [HLY*24], we empirically found that using a pre-trained Stable Diffusion model [RBL*22] in a purely feedforward manner, bypassing the need for multiple diffusion steps, achieves the best result. The feature map f is extracted before the final output layer and used by the decoder to generate Gaussian splats in Stage 2. The feature map f has a shape of $\{N, 320, \frac{H}{8}, \frac{W}{8}\}$, where H and W denote the image resolution. During training, we unfreeze the VAE decoder and UNet components, training the SD model in a feedforward manner without utilizing diffusion steps. Specifically, we set the text prompt to an empty string ("") and use $t = 999$ for the scheduler. The RGB input views are set to 5.

4.1.3. Stage 2

For Stage 2, the SD VAE decoder is adapted to generate Gaussian splats. We modify the SD VAE decoder to accept a channel size of 320 and output 11-channel Gaussian splat predictions, which are then processed by a Gaussian renderer to generate supervision views. The number of Gaussian splats is proportional to the number of input images. Specifically, the total number of splats equals $N \times H \times W \times ch$, where N is the number of input images. During training, we randomly sample between 1 and 5 input views and render additional novel views to produce a total of 8 views for supervision. The SD and VAE decoder are trained simultaneously using only the rendering loss.

We conducted the training on 8 NVIDIA A800 GPUs for both Stage 1 and Stage 2. In Stage 1, we train the model on images with a resolution of 512×512. The batch size for Stage 1 is set to 4 per GPU, resulting in an effective batch size of 32. We train for 40 epochs and Stage 1 training takes approximately 3 days. For Stage 2, we use a batch size of 2 per GPU, resulting in an effective batch size of 16, with training taking around 4 days for 20 epochs. The output 3D Gaussians are rendered at a resolution of 512×512. We utilize the AdamW optimizer [LH17] with a learning rate of 3×10^{-5} for stage 1 and 2.

Dataset	Method	Rot. error↓	Acc. @15° ↑	Acc. @30° ↑	T.error↓
GSO	RelPose++ [LZRT24]	101.24	0.014	0.087	1.75
	iFusion [WCS*23]	107.29	0.011	0.086	1.05
	Ours	**11.50**	**0.93**	**0.99**	**0.16**
ABO	RelPose++ [LZRT24]	103.23	0.016	0.092	1.74
	iFusion [WCS*23]	102.68	0.016	0.094	1.13
	Ours	**19.40**	**0.77**	**0.84**	**0.17**
OO3D	RelPose++ [LZRT24]	104.23	0.017	0.092	1.78
	iFusion [WCS*23]	106.95	0.012	0.086	1.18
	Ours	**12.91**	0.85	**0.97**	**0.13**

Table 1: *Performance on pose prediction task. We compare cross-dataset generalization on GSO [DFK*22], ABO [CGD*22] and OminiObject3D (OO3D) [WZF*23] with baselines RelPose++ [LZRT24], iFusion [WCS*23].*

To evaluate our model's generalization ability across different datasets, we utilize GSO [DFK*22], ABO [CGD*22], and OmniObject3D [WZF*23]. We randomly choose 200 objects to evaluate our model's performance given sparse images as input. For each object, we randomly render 24 views at different elevations, and randomly choose 4 of them as our input images to our model to predict pose and novel view rendering quality. For more details, please see the Appendix.

4.2. Experiment Results

4.2.1. Reconstruction

We first compare our methods under sparse view settings with a recent open-source pose-free method iFusion [WCS*23], and feed-forward based methods LGM [TCC*24] and InstantMesh [XCG*24]. Since LGM and InstantMesh can only work when camera poses are given, we supply them with the ground truth pose for 3D reconstruction. We report PSNR, SSIM and LPIPS metrics for measuring the image quality. As we show in Tab. 2, our model consistently outperforms baselines with a large margin. In addition, we visualize the result in Fig. 6, where the top row

Figure 8: *Our method generates high-resolution 3D Gaussians on various input types. The blue box shows that input views are generated using a Text-to-Image diffusion model, where the yellow box shows sparse view inputs.*

	GSO			ABO		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
iFusion [WCS*23]	17.21	0.852	0.180	17.54	0.853	0.180
LGM [TCC*24]	19.61	0.872	0.131	19.89	0.873	0.131
InstantMesh [XCG*24]	20.75	0.894	0.127	20.98	0.901	0.129
Ours	**25.97**	**0.930**	**0.070**	**25.98**	**0.917**	**0.088**

Table 2: *Performance comparison against baselines on GSO [DFK*22] and ABO [CGD*22] for 4 views input.*

	GSO			ABO		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
CRM [WWC*24]	16.74	0.858	0.177	19.23	0.871	0.169
LGM [TCC*24]	14.31	0.824	0.186	16.03	0.861	0.181
InstantMesh [XCG*24]	16.84	**0.864**	0.177	**19.73**	0.873	0.168
Ours	**16.91**	0.862	**0.177**	19.51	**0.873**	**0.168**

Table 3: *Performance comparison against baselines on GSO [DFK*22] and ABO [CGD*22] for single-image-to-3D setting.*

shows an object from ABO [CGD*22], and the bottom row shows an in-the-wild captured image. In the wild capture, we supply LGM and InstantMesh with our predicted poses, we notice that even with poses, LGM and InstantMesh still struggle to reconstruct the object, as they have overfitted their inputs to the fixed camera position, where our method gives better geometry and visual quality.

We then follow the standard single-image-to-3D paradigm to evaluate our method to demonstrate the flexibility of our method with the standard approach. Specifically, we use the off-shelf Flux [Lab24] diffusion model to generate multi-views. As shown in Tab. 3, our approach works effectively within the existing single-image-to-3D paradigm, delivering on-par performance with current baselines. In addition, as we show in Fig. 7, our method can utilize the multi-view diffusion model and faithfully produce results at 512×512.

In Fig. 8, we demonstrate our model's generalization ability across different data sources. In the top 3 rows, we showcase where our method pairs with a Text-to-Image (T2I) Flux [Lab24] model, and in the bottom two rows, we show results with scanned objects and in-the-wild captured objects. Our model produces high-quality results at a resolution of 512×512, demonstrating the capability for real-world applications with an arbitrary number of sparse input views.

4.2.2. Pose Estimation

We compare LucidFusion with feed-forward approach RelPose++ [LZRT24] and recently open sourced optimization based approach iFusion [WCS*23]. We follow iFusion [WCS*23] and measure median error in rotation and translation. We also report the relative rotation accuracy below thresholds 15° and 30°. As shown

Figure 9: *Point cloud visualization for stage 1 and 2. We also show their chamfer distance.*

in Tab. 1, our method consistently outperforms baseline models across different datasets. It is worth noting that optimization-based pipeline iFusion introduces a 5-minute optimization time for each of the objects, where our method recovers pose and object shapes with a single feed-forward pass, demonstrating our superior performance for real-world applications. Please see more results in the Appendix.

4.3. Ablation Study

4.3.1. Importance of RCG

As detailed in Sec. 3.3, the RCG representation enforces global 3D consistency across all input views. Fig. 9 illustrates several examples of point clouds derived from the RCG representation, where we extract both position and RGB data for visualization. We also compute the Chamfer Distance between each stage's output and the ground-truth point cloud. The results clearly show that incorporating the RCG representation produces smoother, more coherent reconstructions. For more details please refer to the Appendix.

4.3.2. Training Scheme

As we explained in Sec. 3.4, jointly optimizing the model with RCM and rendering supervision leads to misalignment and empty holes, as the network struggles to localize the object geometry and maintain multi-view consistency, as we show in Fig. 5. These artifacts reflect the incorrect position extracted from the predicted RCGs. However, in the two-stage training scheme, we first learn the per-pixel alignment using the RCM supervision, and extend the RCM to RCG to utilize rendering supervision to ensure 3D consistency across multi-views.

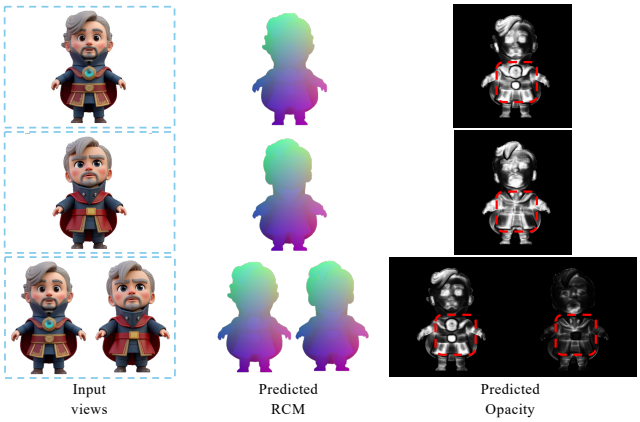

Figure 10: *Visualization of predicted RCM and opacity extracted from RCG as confidence map.*

4.3.3. Gaussians Opacity as Confidence

As we illustrated in Sec. 3.3, extending RCM to RCG not only enables supervision via rendering but also enforces 3D consistency across views, leading to a globally optimized 3D representation. Without RCG refinement, multi-view misalignment can cause conflicts, as the model maps image pixels directly to 3D points, leading to geometric ambiguities. However, RCG's opacity serves as a confidence measure, filtering out conflicting regions across input images and improving multi-view fusion. As shown in Fig. 10, the predicted opacity maps reflect confidence in different regions, allowing the model to lower opacity in conflicting areas and preserve object rendering quality.

4.4. Limitation

Despite the promising results, our model has some limitations. First, it can only render objects positioned at the center of the scene, without backgrounds. We hypothesize that incorporating background information into the RCG representation during training could address this issue, which we leave for future work. Additionally, our current model is trained only with Objaverse data. Future work could explore training on a wider variety of settings to enhance the robustness of the RCG representation.

5. Conclusion

In this work, we propose LucidFusion, a flexible end-to-end feed-forward framework that leverages the *Relative Coordinate Gaussians (RCG)*, a novel representation designed to align geometric features coherently across different views. Our model first maps RGB inputs to *Relative Coordinate Map (RCM)* representations and extends it to RCG for simultaneously reconstructing the object and recovering pose, all in a feedforward manner. This approach alleviates the pose requirements in the 3D reconstruction pipelines and delivers high-quality outputs across a range of scenarios. LucidFusion can also integrate seamlessly with the original single-image-to-3D pipeline, making it a versatile tool for 3D object reconstruction. We hope this work will open new avenues for future research in the field of 3D reconstruction.

References

[CBZ*22] CHANG D., BOŽIČ A., ZHANG T., YAN Q., CHEN Y., SÜSSTRUNK S., NIESSNER M.: Rc-mvsnet: Unsupervised multi-view stereo with neural rendering. In *European conference on computer vision* (2022), Springer, pp. 665–680. 2

[CGD*22] COLLINS J., GOEL S., DENG K., LUTHRA A., XU L., GUNDOGDU E., ZHANG X., VICENTE T. F. Y., DIDERIKSEN T., ARORA H., ET AL.: Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), pp. 21126–21136. 6, 7

[CHXS19] CHEN R., HAN S., XU J., SU H.: Point-based multi-view stereo network. In *Proceedings of the IEEE/CVF international conference on computer vision* (2019), pp. 1538–1547. 2

[CVHC08] CAMPBELL N. D., VOGIATZIS G., HERNÁNDEZ C., CIPOLLA R.: Using multiple hypotheses to improve depth-maps for multi-view stereo. In *Computer Vision–ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part I 10* (2008), Springer, pp. 766–779. 2

[CXZ*21] CHEN A., XU Z., ZHAO F., ZHANG X., XIANG F., YU J., SU H.: Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF international conference on computer vision* (2021), pp. 14124–14133. 2, 3, 4

[DFK*22] DOWNS L., FRANCIS A., KOENIG N., KINMAN B., HICKMAN R., REYMANN K., MCHUGH T. B., VANHOUCKE V.: Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)* (2022), IEEE, pp. 2553–2560. 6, 7

[DSS*23] DEITKE M., SCHWENK D., SALVADOR J., WEIHS L., MICHEL O., VANDERBILT E., SCHMIDT L., EHSANI K., KEMBHAVI A., FARHADI A.: Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 13142–13153. 5

[FP09] FURUKAWA Y., PONCE J.: Accurate, dense, and robust multi-view stereopsis. *IEEE transactions on pattern analysis and machine intelligence 32*, 8 (2009), 1362–1376. 2

[GHZ*23] GE W., HU T., ZHAO H., LIU S., CHEN Y.-C.: Ref-neus: Ambiguity-reduced neural implicit surface learning for multi-view reconstruction with reflection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 4251–4260. 3

[HLX*23] HE H., LIANG Y., XIAO S., CHEN J., CHEN Y.: Cp-nerf: Conditionally parameterized neural radiance fields for cross-scene novel view synthesis. *Computer Graphics Forum 42*, 7 (2023), e14940. 2

[HLY*24] HE J., LI H., YIN W., LIANG Y., LI L., ZHOU K., LIU H., LIU B., CHEN Y.-C.: Lotus: Diffusion-based visual foundation model for high-quality dense prediction. *arXiv preprint arXiv:2409.18124* (2024). 5, 6

[HST*24] HUANG Z., STOJANOV S., THAI A., JAMPANI V., REHG J. M.: Zeroshape: Regression-based zero-shot shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 10061–10071. 2

[HZG*23] HONG Y., ZHANG K., GU J., BI S., ZHOU Y., LIU D., LIU F., SUNKAVALLI K., BUI T., TAN H.: Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400* (2023). 2, 3, 4

[JCT13] JIANG N., CUI Z., TAN P.: A global linear method for camera pose registration. In *Proceedings of the IEEE international conference on computer vision* (2013), pp. 481–488. 2

[JJGZ24] JIANG H., JIANG Z., GRAUMAN K., ZHU Y.: Few-view object reconstruction with unknown categories and camera poses. In *2024 International Conference on 3D Vision (3DV)* (2024), IEEE, pp. 31–41. 3

[KKLD23] KERBL B., KOPANAS G., LEIMKÜHLER T., DRETTAKIS G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph. 42*, 4 (2023), 139–1. 3, 5

[KS00] KUTULAKOS K. N., SEITZ S. M.: A theory of shape by space carving. *International journal of computer vision 38* (2000), 199–218. 2

[Lab24] LABS B. F.: Flux. https://github.com/black-forest-labs/flux, 2024. 7

[LCCT23] LI W., CHEN R., CHEN X., TAN P.: Sweetdreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d. *arXiv preprint arXiv:2310.02596* (2023). 2, 4

[LCR24] LEROY V., CABON Y., REVAUD J.: Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision* (2024), Springer, pp. 71–91. 3

[LH17] LOSHCHILOV I., HUTTER F.: Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017). 6

[LHC24] LIANG Y., HE H., CHEN Y.: Retr: Modeling rendering via transformer for generalizable neural surface reconstruction. *Advances in Neural Information Processing Systems 36* (2024). 2

[LLZ*23] LI S., LI C., ZHU W., YU B., ZHAO Y., WAN C., YOU H., SHI H., LIN Y.: Instant-3d: Instant neural radiance field training towards on-device ar/vr 3d reconstruction. In *Proceedings of the 50th Annual International Symposium on Computer Architecture* (2023), pp. 1–13. 3

[LXJ*24] LIU M., XU C., JIN H., CHEN L., VARMA T M., XU Z., SU H.: One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems 36* (2024). 3

[LYL*24] LIANG Y., YANG X., LIN J., LI H., XU X., CHEN Y.: Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 6517–6526. 2, 3

[LZRT24] LIN A., ZHANG J. Y., RAMANAN D., TULSIANI S.: Relpose++: Recovering 6d poses from sparse-view observations. In *2024 International Conference on 3D Vision (3DV)* (2024), IEEE, pp. 106–115. 3, 6, 7

[MST*21] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHI R., NG R.: Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM 65*, 1 (2021), 99–106. 2

[NBM*22] NIEMEYER M., BARRON J. T., MILDENHALL B., SAJJADI M. S., GEIGER A., RADWAN N.: Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 5480–5490. 3

[PJBM22] POOLE B., JAIN A., BARRON J. T., MILDENHALL B.: Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022). 3

[RBL*22] ROMBACH R., BLATTMANN A., LORENZ D., ESSER P., OMMER B.: High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), pp. 10684–10695. 6

[RWZ*23] REN Y., WANG F., ZHANG T., POLLEFEYS M., SÜSSTRUNK S.: Volrecon: Volume rendering of signed ray distance functions for generalizable multi-view reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 16685–16695. 2

[SCZ*23] SHI R., CHEN H., ZHANG Z., LIU M., XU C., WEI X., CHEN L., ZENG C., SU H.: Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110* (2023). 3

[SF16] SCHONBERGER J. L., FRAHM J.-M.: Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 4104–4113. 2

[SRV24] SZYMANOWICZ S., RUPPRECHT C., VEDALDI A.: Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 10208–10217. 4

[SWY*23] SHI Y., WANG P., YE J., LONG M., LI K., YANG X.: Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512* (2023). 3

[SZFP16] SCHÖNBERGER J. L., ZHENG E., FRAHM J.-M., POLLEFEYS M.: Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14* (2016), Springer, pp. 501–518. 2

[TCC*24] TANG J., CHEN Z., CHEN X., WANG T., ZENG G., LIU Z.: Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054* (2024). 2, 3, 5, 6, 7

[Tea25] TEAM O. D.: Opencv: Open source computer vision library, 2025. Version 4.12.0. URL: https://opencv.org/. 4, 5

[TPL*24] TOCHILKIN D., PANKRATZ D., LIU Z., HUANG Z., LETTS A., LI Y., LIANG D., LAFORTE C., JAMPANI V., CAO Y.-P.: Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151* (2024). 2

[WCLL23] WANG G., CHEN Z., LOY C. C., LIU Z.: Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 9065–9076. 3

[WCS*23] WU C.-H., CHEN Y.-C., SOLARTE B., YUAN L., SUN M.: ifusion: Inverting diffusion for pose-free reconstruction from sparse views. *arXiv preprint arXiv:2312.17250* (2023). 3, 6, 7

[WLC*24] WANG S., LEROY V., CABON Y., CHIDLOVSKII B., REVAUD J.: Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 20697–20709. 3

[WLW*24] WANG Z., LU C., WANG Y., BAO F., LI C., SU H., ZHU J.: Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems 36* (2024). 2, 3

[WTB*23] WANG P., TAN H., BI S., XU Y., LUAN F., SUNKAVALLI K., WANG W., XU Z., ZHANG K.: Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction. *arXiv preprint arXiv:2311.12024* (2023). 3

[WWC*24] WANG Z., WANG Y., CHEN Y., XIANG C., CHEN S., YU D., LI C., SU H., ZHU J.: Crm: Single image to 3d textured mesh with convolutional reconstruction model. *arXiv preprint arXiv:2403.05034* (2024). 2, 6, 7

[WWG*21] WANG Q., WANG Z., GENOVA K., SRINIVASAN P. P., ZHOU H., BARRON J. T., MARTIN-BRUALLA R., SNAVELY N., FUNKHOUSER T.: Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2021), pp. 4690–4699. 3

[WWX*21] WANG Z., WU S., XIE W., CHEN M., PRISACARIU V. A.: Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064* (2021). 2

[WZF*23] WU T., ZHANG J., FU X., WANG Y., REN J., PAN L., WU W., YANG L., WANG J., QIAN C., ET AL.: Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 803–814. 6

[XCG*24] XU J., CHENG W., GAO Y., WANG X., GAO S., SHAN Y.: Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191* (2024). 6, 7

[XLC*24] XU C., LI A., CHEN L., LIU Y., SHI R., SU H., LIU M.: Sparp: Fast 3d object reconstruction and pose estimation from sparse views. *arXiv preprint arXiv:2408.10195* (2024). 3

[XTL*23] XU Y., TAN H., LUAN F., BI S., WANG P., LI J., SHI Z., SUNKAVALLI K., WETZSTEIN G., XU Z., ET AL.: Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. *arXiv preprint arXiv:2311.09217* (2023). 3

[YLL*18] YAO Y., LUO Z., LI S., FANG T., QUAN L.: Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)* (2018), pp. 767–783. 2, 4

[YLL*19] YAO Y., LUO Z., LI S., SHEN T., FANG T., QUAN L.: Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 5525–5534. 2

[YPW23] YANG J., PAVONE M., WANG Y.: Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2023), pp. 8254–8263. 3

[ZBT*24] ZHANG K., BI S., TAN H., XIANGLI Y., ZHAO N., SUNKAVALLI K., XU Z.: Gs-lrm: Large reconstruction model for 3d gaussian splatting. *arXiv preprint arXiv:2404.19702* (2024). 3

[ZIE*18] ZHANG R., ISOLA P., EFROS A. A., SHECHTMAN E., WANG O.: The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 586–595. 5

[ZYG*24] ZOU Z.-X., YU Z., GUO Y.-C., LI Y., LIANG D., CAO Y.-P., ZHANG S.-H.: Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 10324–10335. 2