

---

# Sparse-view Pose Estimation and Reconstruction via Analysis by Generative Synthesis

---

Qitao Zhao      Shubham Tulsiani

Carnegie Mellon University

Project page: [qitaozhao.github.io/SparseAGS](http://qitaozhao.github.io/SparseAGS)

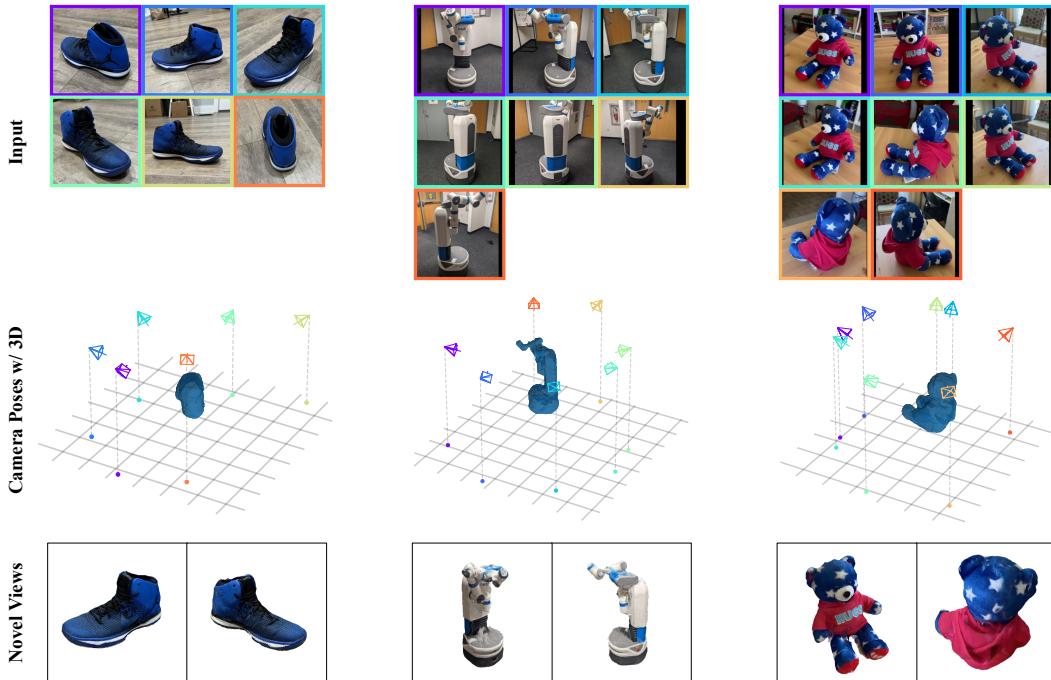


Figure 1: Given a set of unposed input images, SparseAGS jointly infers the corresponding camera poses and underlying 3D, allowing high-fidelity 3D inference in the wild.

## Abstract

Inferring the 3D structure underlying a set of multi-view images typically requires solving two co-dependent tasks – accurate 3D reconstruction requires precise camera poses, and predicting camera poses relies on (implicitly or explicitly) modeling the underlying 3D. The classical framework of analysis by synthesis casts this inference as a joint optimization seeking to explain the observed pixels, and recent instantiations learn expressive 3D representations (*e.g.*, Neural Fields) with gradient-descent-based pose refinement of initial pose estimates. However, given a sparse set of observed views, the observations may not provide sufficient direct evidence to obtain complete and accurate 3D. Moreover, large errors in pose estimation may not be easily corrected and can further degrade the inferred 3D. To allow robust 3D reconstruction and pose estimation in this challenging setup, we propose SparseAGS, a method that adapts this analysis-by-synthesis approach by: a) including novel-view-synthesis-based generative priors in conjunction with photometric objectives to improve the quality of the inferred 3D, and b) explicitly

reasoning about outliers and using a discrete search with a continuous optimization-based strategy to correct them. We validate our framework across real-world and synthetic datasets in combination with several off-the-shelf pose estimation systems as initialization. We find that it significantly improves the base systems’ pose accuracy while yielding high-quality 3D reconstructions that outperform the results from current multi-view reconstruction baselines.

## 1 Introduction

Consider the images of the robot shown in Fig. 1. From just these few images, we humans can easily understand the 3D structure of this object – it has a cylindrical base supporting a tall body from which an arm extends to the front. We do this by aggregating the information across images into a consistent 3D mental model, *e.g.*, the “front” view informs us of the width of the body and the “side” view(s) about the extended arm. But how do we know which image is to the “front” or to the “side” to begin with? As evidenced in seminal research of mental rotation [30], we understand viewpoints by forming mental 3D models. Thus, to form mental 3D models, we need to understand the (relative) viewpoints across images, but doing so in turn requires a mental 3D model!

This co-dependency in inferring shape and pose is one that any computational approach aiming to recover 3D from multiple views also needs to deal with. Indeed, classical approaches like Structure-from-Motion (SfM) [29] tackle the two together and infer 3D and camera viewpoints. However, these correspondence-based methods can only infer sparse 3D representations and are not robust given a small set of images with limited overlap. To allow 3D inference in such sparse-view settings, recent learning-based approaches have pursued sparse-view reconstruction approaches [52, 45], but assuming known precise camera poses. Separately, there have been several methods [18, 47, 48, 31] which predict camera viewpoints given a set of images. While these methods have led to impressive results for both 3D reconstruction and pose inference, their singular focus on only one task without tackling the other limits their utility – the 3D reconstruction methods requiring precise cameras cannot be easily used in real-world applications, and pose estimation methods that do not model 3D are typically limited in their accuracy.

We present SparseAGS, a framework that unifies the advances in learning-based pose estimation and 3D reconstruction – using the former as an initialization and building on the latter for obtaining accurate 3D reconstruction. Specifically, we adopt an “analysis by synthesis” approach where we jointly optimize pose and 3D to explain the observed pixels. However, unlike prior methods [19, 42] which simply leverage photometric-error-driven gradient-based updates for pose and 3D, we additionally leverage generative priors [8, 33] for more complete (and accurate) 3D despite input images that may only partially capture the object. However, current off-the-shelf novel-view generative models [20] only allow 3-DoF camera parameterization which is insufficient beyond synthetic settings, we finetune a SoTA model to allow 6-DoF camera variation when querying novel views. We find that such generative priors not only contribute to the 3D reconstruction quality but also result in more accurate camera poses. Moreover, we also explicitly account for large possible errors in initial camera estimation and prevent these from degrading 3D reconstruction via identifying outliers, and also improve poses via a combination of a discrete search and continuous optimization.

Compared to prior joint reconstruction and pose estimation methods that are designed to improve near-perfect initial cameras [19, 38], SparseAGS can leverage off-the-shelf pose estimates, thereby allowing robust inference in real-world scenarios. We demonstrate its efficacy using both, real-world and synthetic datasets in conjunction with several state-of-the-art pose estimation methods as initialization. We find that our approach significantly improves the initial camera estimates and yields high-fidelity 3D reconstructions (and novel view renderings). In summary, our contributions are:

- We introduce an analysis-by-generative-synthesis framework that jointly estimates 3D and camera viewpoints given a sparse set of input images, by integrating a 6-DoF novel-view generative prior in an analysis-by-synthesis approach
- Our approach allows leveraging any off-the-shelf pose estimation system and can robustly estimate 3D and viewpoints despite large errors in the initial estimates.
- We present results across datasets and initializations and show clear improvements over the initializations as well as outperform prior sparse-view 3D reconstruction baselines.

## 2 Related Work

**Sparse-view Pose Estimation.** Traditional correspondence-based Structure-from-Motion [32, 29] methods often fail to estimate camera poses in sparse-view settings. Several approaches instead seek to leverage data-driven priors, for example learning energy-based [48, 18] or denoising diffusion [39] models to predict cameras. While these approaches predict global camera representations, some works have demonstrated the benefit of denser camera parametrizations by predicting raymaps [47] or pointmaps [41, 17]. As an alternative paradigm to direct camera prediction, some recent methods [3, 43, 34] instead estimate relative poses by inverting the view-conditioned synthesis capabilities of diffusion models [20]. While these methods have led to remarkable improvements in camera estimation, these are still susceptible to some imprecision and occasional outliers which our 3D-reasoning-based approach can correct.

**Sparse-view 3D Reconstruction.** This line of work aims to recover 3D from sparsely sampled views, aiming to infer complete 3D representations that faithfully reflect the content captured by the input images while making reasonable guesses for invisible areas. The progress of diffusion models [8, 33] has greatly advanced this direction, as they are capable of learning strong natural image priors from data. Inspired by DreamFusion [23], which generates 3D scenes given textual descriptions leveraging a text-to-image diffusion model [26], SparseFusion [52] learns a view-conditioned diffusion model on multi-view image collections for novel view synthesis and then distills the learned novel-view distributions into a single consistent 3D representation. DreamSparse [45] further improves the performance by utilizing internet-scale natural image priors learned by Stable Diffusion [25]. Although these methods present impressive results, they assume precise camera poses are available, which limits their applications. FORGE [11] addresses this by jointly inferring both camera poses and 3D structure in a single forward pass, though the quality of its reconstructions remains constrained by pose estimation accuracy without further refinement or correction.

**Pose-free Sparse-view 3D Reconstruction.** Some recent works [27, 40, 12, 13] have attempted to bypass the reasoning about camera poses and directly infer novel views or 3D representations from unposed images. An unposed variant of the Scene Representation Transformer [27] encodes a set of input images into latent features and synthesizes novel views given the corresponding query rays (w.r.t. the viewpoint of the first image) using a transformer encoder and decoder. UpFusion [13] improved upon this by learning a diffusion model and distilling a consistent 3D representation via Score Distillation Sampling [35], whereas LEAP [12] and PF-LRM [40] can directly predict (volumetric or triplane) 3D representations in a feedforward manner. While these methods demonstrate promising results, their geometry-free approach cannot easily capture the specific details across input images and they struggle to improve the 3D estimation with additional input images.

**Analysis-by-synthesis Approaches.** Approaching visual perception as an inverse graphics task is classical idea in computer vision [15, 46], and has been leveraged for inferring scene properties (*e.g.*, object pose) by synthesizing visual content as close to observations as possible [2, 16, 51, 50]. Closer to our setup, prior approaches jointly optimize camera pose and 3D representation (*e.g.*, NeRF [22]) to explain the observed images [19, 42] but these methods are designed for dense observations and only handle small pose errors. Closer to our work, SPARF [38] focuses on the sparse-view setup, leveraging estimated pixel correspondence [37] as prior knowledge in addition to the standard photometric loss. However, reliably extracting such correspondences can be challenging, and false match estimates may even confuse pose refinement, leading to degraded 3D reconstruction.

## 3 Method

### 3.1 Overview

**Analysis by Synthesis.** Given a set of sparse-view images, denoted as  $\mathbb{I} \equiv \{\mathbf{I}_i\}_{i=1}^N$ , our goal is to reconstruct the underlying 3D structure  $\theta$  and infer the camera poses corresponding to the input images  $\Pi \equiv \{\boldsymbol{\pi}_i\}_{i=1}^N$ . This can be done by solving an *analysis-by-synthesis* problem

$$\min_{\theta, \Pi} \sum_{i=1}^N \|\mathbf{I}_i - f_{\theta}(\boldsymbol{\pi}_i)\|^2 \quad (1)$$

where  $f_{\theta}$  is a rendering function. Eq. 1 demonstrates that we want to find a scene description consisting of a 3D representation  $\theta$  and camera configurations  $\Pi$  that well explain the observed input

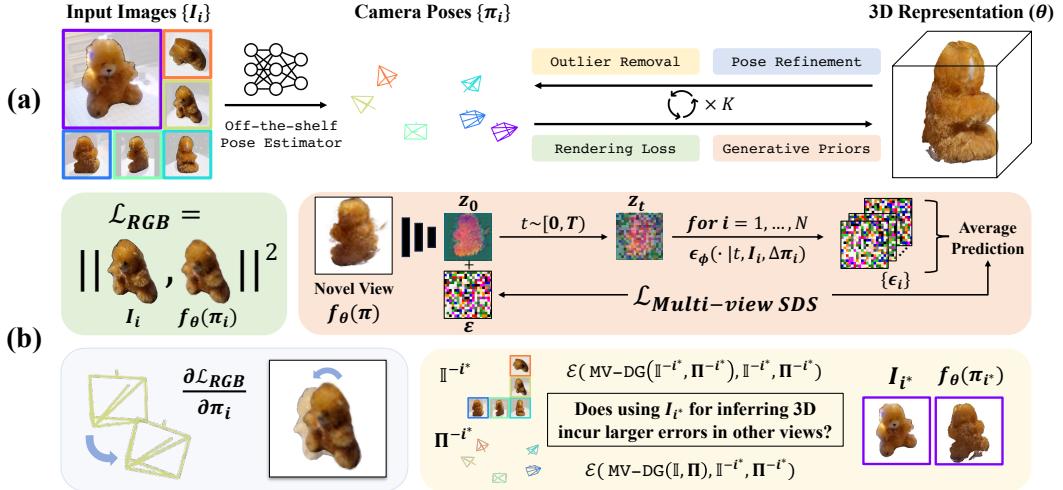


Figure 2: (a) **Overview of SparseAGS**: Given estimated camera poses from off-the-shelf models, our method iteratively reconstructs 3D and optimizes poses leveraging diffusion priors. (b) **Detailed View of Each Component**: We use rendering loss and multi-view SDS loss for 3D reconstruction while the rendering loss is propagated back to refine camera poses. At the end of each reconstruction iteration, we identify outliers by checking if their involvement in 3D inference yields larger errors in other views, implying the inconsistency of their poses with others.

images. If  $f_\theta$  is differentiable, we can jointly optimize the 3D representation and camera poses via gradient descent [19, 42]. However, this approach may not work well in the sparse-view setting [38] (*i.e.*,  $N$  is small) as the 3D representation can overfit to the input images without forming a plausible structure, degrading both, pose estimation and 3D reconstruction.

**Analysis by Generative Synthesis.** To address this issue, we propose to introduce generative priors into analysis by synthesis, so we term our method *analysis by generative synthesis*. In addition to the known-view objective (Eq. 1), we leverage diffusion priors [8, 33] to optimize renderings from randomly sampled novel views ( $\pi$ ) as well

$$\min_{\theta} \mathbb{E}_{\pi} - \log p_\phi(f_\theta(\pi) | \pi, \mathbb{I}, \Pi) \quad (2)$$

where  $p_\phi$  is the likelihood of the novel view rendering conditioned on the viewpoint  $\pi$  and inputs  $(\mathbb{I}, \Pi)$ , modeled by the diffusion model  $\phi$ . The gradients for this objective can be obtained via Score Distillation Sampling (SDS) [23], and intuitively, they encourage the renderings of the 3D representation to be plausible based on image distributions learned by the diffusion model.

In the following, we first introduce a few preliminaries about an efficient single-view-to-3D approach (Sec. 3.2), on which we build our multi-view reconstruction method, MV-DreamGaussian, enabling analysis by generative synthesis in the wild (Sec. 3.3). Then, we present our complete framework that involves dealing with imprecise cameras (Sec. 3.4). An illustration of our approach is in Fig. 2.

### 3.2 Preliminaries: DreamGaussian

DreamGaussian [36] generates 3D from a single image with a two-stage approach, achieving a satisfactory trade-off between speed and fidelity. The first stage optimizes 3D Gaussians [14] (parameterized by  $\theta$ ) using a combination of photometric loss (Eq. 1, except that the camera pose is not optimized) and SDS loss (Eq. 3) with a view-conditioned diffusion model, Zero-1-to-3 [20]. Specifically, for a randomly sampled novel view  $\pi$ , scheduled noise  $\epsilon$  at timestep  $t$  is added to the latent of its rendering (the noisy latent is denoted by  $\mathbf{z}_t$ ). The training objective minimizes the difference between the predicted noise and the added noise, approximating the negative log-likelihood of the rendered image. The gradient of SDS loss is given by

$$\nabla_{\theta} \mathcal{L}_{SDS} = \lambda_{SDS} \mathbb{E}_{t, \pi, \epsilon} \left[ w(t)(\epsilon_\phi(\mathbf{z}_t; t, \mathbf{I}_1, \Delta\pi) - \epsilon) \frac{\partial f_\theta(\pi)}{\partial \theta} \right] \quad (3)$$

where  $w(t)$  is a weighting function,  $\epsilon_\phi(\cdot)$  is a U-Net trained to predict the added noise given the noisy latent  $\mathbf{z}_t$ , conditioned on the timestep  $t$ , reference image  $\mathbf{I}_1$ , and the relative camera pose  $\Delta\pi$ . This stage efficiently builds the geometry of the object with rough texture, which takes 500 training steps (in about 1 minute). In the second stage, 3D Gaussians are converted to a textured mesh with Marching Cubes [21], and only its texture is optimized. This stage takes another 50 steps and can finish within 30 seconds on a single GPU.

We find DreamGaussian to be a suitable starting point to perform *analysis by generative synthesis*, but note that it has some key limitations: (1) **3-DoF Camera Parameterization**. DreamGaussian adopts a 3-DoF camera parameterization (*i.e.*, radius, elevation, and azimuth) to accommodate the camera definition in Zero-1-to-3 [20]. While this parameterization is sufficient for synthetic data, it cannot well represent the 6-DoF camera poses of real-world images. (2) **Single Input Image**. DreamGaussian is designed for the single-view-to-3D task. In contrast, we aim for the reconstructed 3D to reflect the details captured by multiple input images faithfully. This requires an approach to handling information from multi-view images.

### 3.3 Leveraging Generative Priors for Sparse-view 3D in the Wild

We adapt DreamGaussian’s two-stage method and extend it to (1) handle real-world images with 6-DoF camera parameters and (2) utilize sparse-view images as input.

**Generative Priors in the Wild.** Zero-1-to-3 [20] offers desirable generative priors that enable single-view-to-3D generation of DreamGaussian. However, it assumes no in-plane camera rotation and that all possible camera poses are strictly directed toward a common origin. We find these assumptions are over-restrictive for real-world images. Therefore, we propose to replace the 3-DoF camera condition in Zero-1-to-3 with a 6-DoF one, represented as an 18-dimensional vector:

$$[\text{Flatten}(\boldsymbol{\pi}_{\text{rel}}), \log(f_{\text{rel}}^x), \log(f_{\text{rel}}^y)] \quad (4)$$

where  $\boldsymbol{\pi}_{\text{rel}}$  is the relative extrinsic matrix ( $4 \times 4$ ) between the source view and target view, and  $f_{\text{rel}}^x$  ( $f_{\text{rel}}^y$ ) is the ratio of the focal length along the  $x$ - ( $y$ -) axis between them. We include the focal length term to account for the object scale change due to cropping. This simple camera parameterization effectively represents 6-DoF cameras in the wild. Details regarding finetuning Zero-1-to-3 for 6-DoF camera conditioning are deferred to Sec. C in the appendix. We note that recent work, ZeronVNS [28], also discussed this 3-DoF issue of Zero-1-to-3 and proposed a “6-DoF+1” camera parameterization for scene-level novel view synthesis. However, this approach is not directly applicable to our object-centric setting, as it is trained using images with complex backgrounds and leverages depth priors to address scale ambiguity.

**Leveraging the Generative Priors from Multiple Views.** DreamGaussian only uses the generative priors from a single reference image via SDS loss. To make  $\mathcal{L}_{\text{SDS}}$  aware of the visual cues from multiple input images, we modify Eq. 3 as

$$\nabla_\theta \mathcal{L}_{\text{Multi-view SDS}} = \lambda_{\text{SDS}} \mathbb{E}_{t, \boldsymbol{\pi}, \boldsymbol{\epsilon}} \left[ w(t) (\bar{\epsilon}_\phi - \epsilon) \frac{\partial f_\theta(\boldsymbol{\pi})}{\partial \theta} \right], \text{ where} \quad (5)$$

$$\bar{\epsilon}_\phi = \frac{1}{N} \sum_{i=1}^N \epsilon_\phi(\mathbf{z}_t; t, \mathbf{I}_i, \Delta\pi_i) \quad (6)$$

$N$  is the total number of input views,  $\mathbf{I}_i$  is the  $i^{\text{th}}$  input image, and  $\Delta\pi_i$  is its relative camera pose w.r.t. the sampled novel view  $\boldsymbol{\pi}$ . We average the noise predictions from all input views that share the same timestep  $t$ . This method draws inspiration from the implementation of Stable-Dreamfusion [35], but we do not weigh the predicted noises based on the relative closeness of camera poses. The rationale behind this is that the camera poses in our setting are not always reliable, and relying too heavily on “close” views could introduce significant conflicts during the 3D optimization process.

With these modifications, our multi-view reconstruction approach, termed *MV-DreamGaussian*, is capable of reconstructing 3D from sparse images in the wild by leveraging diffusion priors. When describing its use in our overall framework, we use the notation  $\theta = \text{MV-DG}(\mathbb{I}, \Pi)$  to denote the 3D representation inferred via this pipeline given a set of input images  $\mathbb{I}$  and associated viewpoints  $\Pi$ .

### 3.4 3D Reconstruction with Imperfect Cameras

Here, we introduce the complete framework of SparseAGS (see Fig. 2 for an overview) that: a) leverages off-the-shelf pose estimation methods and b) incorporates our multi-view reconstruction

approach MV-DG (Sec. 3.3) to jointly infer accurate 3D and camera viewpoints. A key challenge we seek to overcome is that the estimated camera viewpoints may have significant errors and that naively using all images to infer 3D can result in suboptimal estimates.

**Pose Refinement via Gradient Descent.** During 3D reconstruction via MV-DreamGaussian, we back-propagate gradients from the photometric loss (Eq. 1) back to update camera poses (implementing custom CUDA kernels to enable this gradient computation). This process allows the camera poses to become more precisely aligned as 3D reconstruction progresses. We denote by  $\Pi' = \text{GD}(\mathbb{I}, \theta, \Pi)$  the resulting camera viewpoints from this optimization given the set of input images  $\mathbb{I}$ , and 3D reconstruction  $\theta$  and initial poses  $\Pi$ .

With this pose-and-3D co-optimization, we can instantiate a version of our analysis-by-generative-synthesis framework by iteratively refining poses and reconstructing 3D given initial pose estimates  $\Pi_0$  from an off-the-shelf system:

$$\text{For } k = 1 \dots K : \quad \theta_k = \text{MV-DG}(\mathbb{I}, \Pi_{k-1}); \quad \Pi_k = \text{GD}(\mathbb{I}, \theta_k, \Pi_{k-1}) \quad (7)$$

For clarity, we present separate formulas for the reconstructed 3D  $\theta_k$  and the updated poses  $\Pi_k$ , though they are part of the same optimization process. Notably, in each iteration, we initialize the camera poses using the output from the previous iteration ( $\Pi_{k-1}$ ), while the 3D representation ( $\theta_k$ ) is reset and reconstructed from scratch.

**Dealing with Outliers.** Although the above iterative optimization framework can allow us to infer consistent poses and 3D reconstructions, it is susceptible to local optima and not robust to large errors in initial pose estimates. To overcome this, we additionally detect “*outliers*”, *i.e.*, images with possibly large pose errors that degrade the quality of 3D reconstruction. We then modify our approach to leverage only the estimated inliers for 3D reconstruction while also separately performing a discrete search to update the outlier viewpoints.

*Iterative Outlier Identification.* Our key insight is that an “outlier” image not only exhibits high reprojection error, making it difficult to reconstruct on its own, but also that including it as a training image for 3D reconstruction degrades the overall quality, thus leading to poorer reconstruction even from other views! We operationalize this insight by classifying an image as an outlier if removing it from training significantly improves performance on other images. More formally, let  $\mathbb{I}^{-i}$  denote the set of images after removing the  $i^{\text{th}}$  one and let  $\mathcal{E}(\theta, \mathbb{I}, \Pi)$  denote the average reprojection error of a 3D representation  $\theta$  over images  $\mathbb{I}$  with (predicted) poses  $\Pi$ . We consider an image  $i$  as an outlier if

$$\mathcal{E}(\text{MV-DG}(\mathbb{I}, \Pi), \mathbb{I}^{-i}, \Pi^{-i}) >> \mathcal{E}(\text{MV-DG}(\mathbb{I}^{-i}, \Pi^{-i}), \mathbb{I}^{-i}, \Pi^{-i}) \quad (8)$$

*i.e.*, adding the image to training set significantly increases the error for other views. For efficiency, instead of considering all images as outlier candidates, we iterate over images in decreasing order of reprojection error. Given this procedure to detect outliers, at each iteration (except  $k = 1$ ) we modify the above framework to first filter out the outliers found in previous iterations (along with the new “*outlier candidate*” that gives the largest reprojection error at the last iteration):

$$\mathbb{I}_{k-1}^{\text{inlier}}, \Pi_{k-1}^{\text{inlier}} \equiv \text{filter-outliers}(\mathbb{I}, \theta_{k-1}, \Pi_{k-1}) \quad (9)$$

and only use the estimated inliers for optimizing 3D:  $\theta_k = \text{MV-DG}(\mathbb{I}_{k-1}^{\text{inlier}}, \Pi_{k-1}^{\text{inlier}})$ . This filter-and-reconstruct loop stops when either the selected outlier candidate is determined to be an inlier (*i.e.*, the condition 8 is not satisfied) or the number of remaining inliers falls below a threshold (*e.g.*, 4).

*Correcting Outlier Poses.* While identifying the outliers allows us to prevent them from influencing the 3D inference, the finally recovered model may not capture the details from all images. We thus also attempt to “correct” the pose estimates for the outliers via a discrete search (followed by continuous optimization). Using the currently estimated 3D (reconstructed from only the inliers), for each image in the outlier set, we re-estimate its camera pose via render-and-compare. We first densely sample pose candidates on a sphere and render images from the current 3D. We rank the pose candidates by measuring both pixel-space error (*i.e.*, MSE) and perception error (*i.e.*, LPIPS [49]). The pose candidate with the highest cumulative rank is selected as the optimal solution. Once all identified outliers are corrected, another reconstruction is performed to form a consistent 3D representation with all images using the updated poses.

Our overall framework is very efficient (largely due to an efficient implementation of the reconstruction step), typically taking 5-10 minutes given 8 input images, with increased inference time depending on the number of estimated outliers. We include a brief analysis of the inference time of our system in Sec. B.

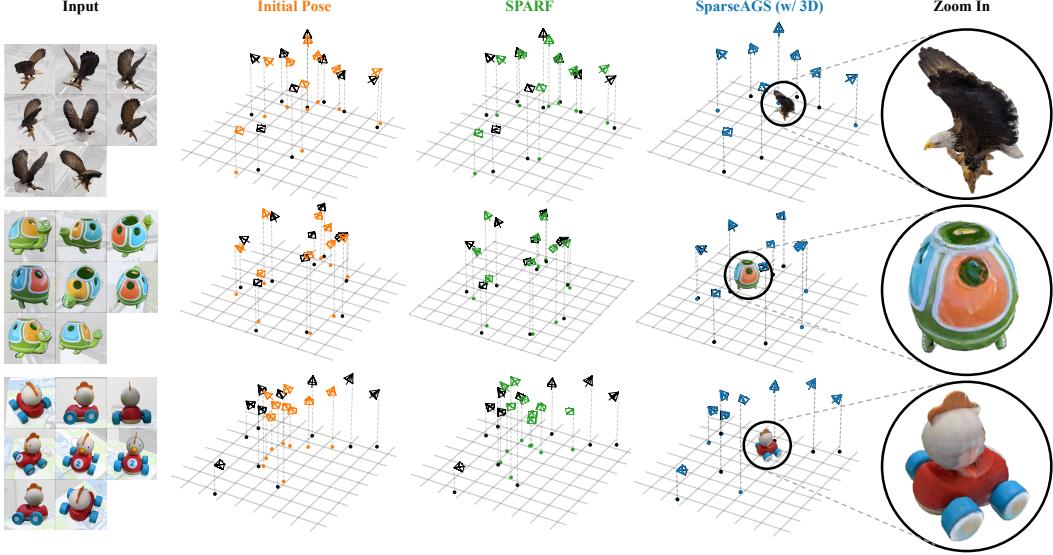


Figure 3: **Qualitative Comparison on Camera Pose Accuracy.** Given **initial** poses from off-the-shelf methods (top to bottom: DUS3R [41], Ray Diff. [47] and RelPose++ [18]), the refined poses from **SPARF** [38] are compared with the output of **SparseAGS**. The estimated cameras are aligned with ground truth (in black) with an optimal similarity transform. More results are available in Fig. 8.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets.** We primarily evaluate our method on a real-world multi-view object-centric dataset NAVI [9]. This dataset includes high-quality foreground masks, precise camera poses, and 3D meshes. For each of the 35 objects in NAVI, we randomly select 5 multi-view sequences for pose estimation and reconstruction. Additionally, we assess our method on synthetic datasets, including GSO [7], ABO [4], and OmniObject3D [44]. Results for the synthetic datasets are provided in Sec. E of the appendix.

**Baselines.** To evaluate camera pose accuracy, we select three sparse-view pose estimation baseline methods: RelPose++ [18], Ray Diffusion [47], and DUS3R [41]. The first two are trained exclusively on CO3D [24], while DUS3R is trained on a mixture of eight datasets, representing different levels of precision in initial camera poses. Our method initializes and improves the pose estimates from these baselines, and we also compare with SPARF [38], a sparse-view pose-NeRF co-optimization method. For evaluation of novel view synthesis, we mainly compare our method with unposed sparse-view reconstruction approaches, LEAP [12] and UpFusion [13] (we include comparison with SPARF in Sec. D). We conduct experiments with varying numbers of input images ( $N = 6, 8, 10, 16$ ).

**Metrics.** For pose accuracy, we follow prior works [18, 47] and report the following metrics: (1) Rotation accuracy: we compare pairwise relative rotation between the predicted cameras and ground truth. We report the proportion of samples with errors less than a specified threshold, such as 5 and 15 degrees. (2) Camera center accuracy: we align the predictions and ground truth using an optimal similarity transform and report the proportion of camera centers within 10% of the scene

Table 1: **Comparison of Camera Rotation and Center Accuracy with SPARF [38].** We use three pose estimation baselines (RelPose++ [18], Ray Diff. [47], DUS3R [41]) and measure rotation accuracy at two thresholds (5 and 15 degrees) and camera center accuracy at a threshold of 0.1 (of the scene scale). Eight images are used.

Method	Rot. @ 5° ↑	Rot. @ 15° ↑	CC @ 0.1 ↑
<b>RelPose++</b>	10.9	56.4	26.0
w/ SPARF	28.6(+17.7)	51.9(-4.5)	37.9(+11.9)
w/ SparseAGS	42.1(+31.2)	67.6(+11.2)	53.3(+27.3)
<b>Ray Diff.</b>	13.5	73.5	38.3
w/ SPARF	46.0(+32.5)	76.1(+2.6)	65.8(+27.5)
w/ SparseAGS	60.3(+46.8)	88.2(+14.7)	80.4(+42.1)
<b>DUS3R</b>	52.3	93.8	82.2
w/ SPARF	59.7(+7.4)	87.8(-6.0)	81.9(-0.3)
w/ SparseAGS	83.7(+31.4)	96.2(+2.4)	93.5(+11.3)



Figure 4: **Qualitative Comparison with LEAP [12] on Novel View Synthesis.** We use two pose estimation baselines (Ray Diffusion [47] and DUS3R [41]). SparseAGS better preserves details from the input images and shows enhanced performance with more accurate initial camera poses. More results are available in Fig. 9 of the appendix.

scale relative to the ground truth. We evaluate our 3D representation via novel-view synthesis and report PSNR and LPIPS [49] for the rendered views. In our ablation study, we also assess the 3D geometry using the F1 score, comparing our recovered geometry against the ground truth 3D meshes.

Table 2: **Evaluation of Camera Pose Accuracy with Varying Numbers of Input Images on NAVI [9].** Here we use the same evaluation protocols as Tab. 1.

Method	N = 6			N = 10			N = 16		
	Rot. @ 5°	Rot. @ 15°	CC @ 0.1	Rot. @ 5°	Rot. @ 15°	CC @ 0.1	Rot. @ 5°	Rot. @ 15°	CC @ 0.1
<b>RelPose++</b>	11.0	57.0	28.6	/	/	/	/	/	/
+ SparseAGS	27.3(+16.3)	60.2(+3.2)	47.0(+18.4)	/	/	/	/	/	/
<b>Ray Diff.</b>	13.3	74.3	44.1	12.9	73.4	36.1	12.6	74.0	34.0
+ SparseAGS	44.9(+31.6)	83.6(+9.3)	73.6(+29.5)	70.0(+57.1)	89.6(+16.2)	83.4(+47.3)	82.3(+69.7)	93.2(+19.2)	89.0(+55.0)
<b>DUS3R</b>	50.3	93.4	82.1	52.9	95.0	84.4	55.5	94.9	84.2
+ SparseAGS	74.3(+24.0)	95.1(+1.7)	92.3(+10.2)	87.0(+34.1)	97.3(+2.3)	94.3(+9.9)	91.1(+35.6)	97.7(+2.8)	95.0(+10.8)

## 4.2 Evaluation

**Camera Pose Accuracy.** We compare SparseAGS with SPARF [38] on pose accuracy given eight input images quantitatively in Tab. 1 (numbers are in percentage) and qualitatively in Fig. 3. We find that SparseAGS consistently yields larger improvements than SPARF, which sometimes even leads to degraded accuracy (marked by red numbers). We attribute this to the unreliable correspondences extracted by SPARF (we include an example in Fig. 7), as the input images in NAVI may exhibit more significant viewpoint changes compared to scene-level datasets, *e.g.*, DTU [10] where SPARF

Table 3: **Quantitative Comparison of 3D Reconstruction on NAVI [9]**. We compare our method with two unposed approaches: LEAP [12] and UpFusion [13], using varying numbers of input images (N). We adopt two pose initializations (Ray Diff. [47], DUS3R [41]) reporting PSNR and LPIPS.

Method	Initial Cam. Pose	N = 6		N = 8		N = 10	
		PSNR $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	LPIPS $\downarrow$
LEAP	×	12.84	0.2918	12.93	0.2902	12.98	0.2890
UpFusion	×	13.30	0.2747	13.27	0.2744	/	/
SparseAGS	Ray Diff.	13.63	0.2697	15.30	0.2304	16.80	0.1960
SparseAGS	DUS3R	<b>15.56</b>	<b>0.2173</b>	<b>17.03</b>	<b>0.1870</b>	<b>18.03</b>	<b>0.1660</b>

is originally tested. Note that training SPARF (or other NeRF-based methods) is far more expensive than ours, and it may take more than 10 hours. Whereas our method typically finishes in 5-10 minutes. More analysis and detailed comparisons with SPARF on pose accuracy and novel view synthesis are in Sec. D.

We vary the number of input images (N = 6, 10, 16) and report camera pose accuracy in Tab. 2 (we only test RelPose++ [18] with six images as inference with more than eight images is not supported). SparseAGS consistently enhances baseline performance for both rotation and camera center accuracy, with particularly significant gains for stricter metrics (*e.g.*, Rot. @ 5°). Moreover, the improvements tend to further increase with the number of input images. These results demonstrate that our method is robust to varying levels of initial camera poses and generalizes well across different input numbers.

**3D Reconstruction.** In Table 3, we compare SparseAGS with two unposed approaches, LEAP [12] and UpFusion [13], reporting metrics for 3D reconstruction (novel view synthesis). Our method consistently outperforms both baselines across different numbers of input images and with two pose estimation initializations. While SparseAGS shows continuous improvements with an increasing number of input images, the performance of LEAP and UpFusion nearly saturates in terms of both PSNR and LPIPS. We hypothesize that unposed methods struggle to utilize additional input images beyond their training capacity without further training adjustments (LEAP is trained using five views, while UpFusion is trained with a maximum of six images). In contrast, our method is flexible w.r.t. the number of input images, eliminating the need for re-training. A qualitative comparison with LEAP and UpFusion is presented in Fig. 4 and Fig. 5, respectively. The results show that SparseAGS better preserves the details in input images by explicitly modeling cameras and produces higher-quality novel view synthesis with more precise initial camera poses.

Table 4: **Ablation Study.** Using initial poses from Ray Diffusion [47] for eight input images, we ablate the effect of each proposed component of our approach.

Method	Rot. @ 5° $\uparrow$	Rot. @ 15° $\uparrow$	CC @ 0.1 $\uparrow$	PSNR $\uparrow$	LPIPS $\downarrow$	F1 @ 0.01 $\uparrow$
Ray Diffusion	13.5	73.5	38.3	/	/	/
+ Pose-3D Co-opt. (w/o SDS)	(1)	28.4	79.9	57.7	12.72	0.3100
+ SDS (vanilla Zero123 [20])	(2)	30.2	78.3	57.3	13.04	0.2999
+ SDS (Our 6-DoF Zero123)	(3)	34.6	83.1	65.3	13.44	0.2793
+ Outlier Removal & Correction	(4)	60.3	88.2	80.4	15.30	0.2304
						68.2

### 4.3 Ablation Study

We ablate the effectiveness of each component in our approach (Tab. 4) using initial pose estimates from Ray Diffusion [47] with eight input images. In addition to camera pose metrics and novel-view

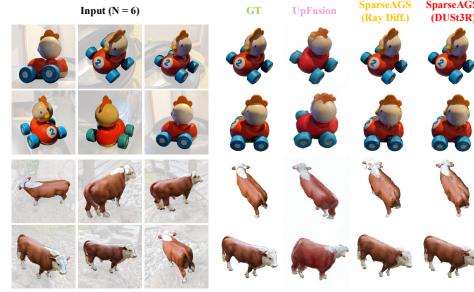


Figure 5: **Qualitative Comparison with UpFusion [13] on Novel View Synthesis.** We use two pose estimation baselines (Ray Diffusion [47] and DUS3R [41]) as in Fig. 4. Note that the left eye and symbol ② of the Chicken Racer is missing in UpFusion’s output, probably because of the “first-image bias”, while SparseAGS preserves these details.

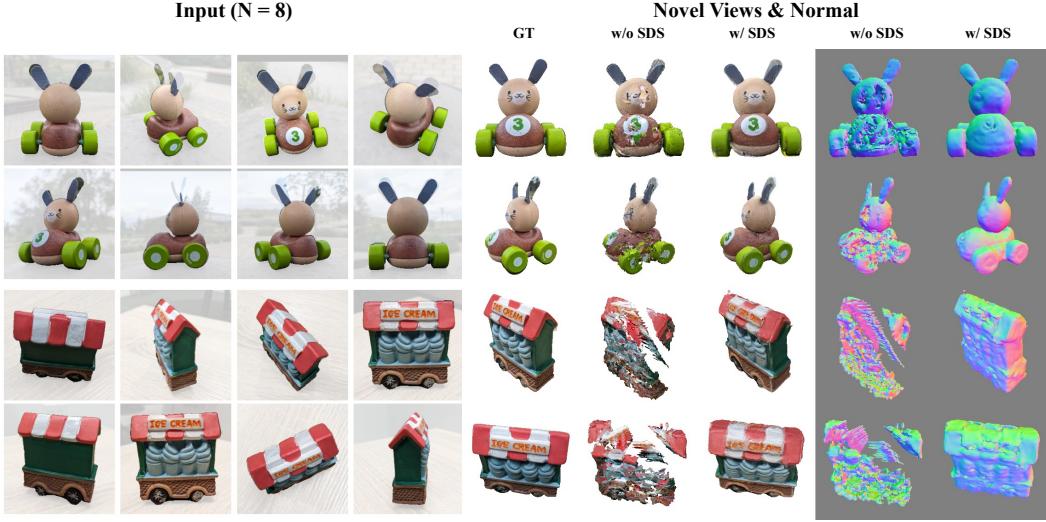


Figure 6: **Qualitative Comparison with No-Generative-Piror Setup (N = 8).** Novel Views (NV) & Normal: From left to right – GT, NV w/o SDS, NV w/ SDS, Normal w/o SDS, Normal w/ SDS. Leveraging generative priors in the form of SDS contributes to a consistent 3D representation.

metrics, we report the F1 score of reconstructed meshes, which reflects their alignment with ground truth meshes.

**(Appropriate) Generative Priors Improve Analysis by Synthesis.** We find that adding generative priors (Eq. 2) to naive pose-3D co-optimization (Eq. 1) improves both pose accuracy and 3D reconstruction quality (comparison between (1) and (3) shows consistent improvements in all metrics). However, vanilla Zero-1-to-3 [20] is not suitable for providing such priors in real-world scenarios, as we observed a drop in camera rotation and center accuracy (compare (1) with (2) in Rot. @ 15° and CC@0.1). This is because 3-DoF camera parameterization cannot well represent the camera poses in the wild. Although the numerical improvements may appear marginal (*e.g.*, in PSNR), Fig. 6 presents a qualitative comparison of 3D reconstruction with and without our 6-DoF novel-view generative priors. Supervision on novel views via SDS helps form a consistent 3D representation.

**Outlier Removal and Correction.** The presence of outlier initial cameras introduces significant challenges to pose-3D co-optimization. Our iterative outlier removal and correction pipeline effectively addresses this issue. For instance, comparing (3) with (4) shows a substantial improvement: Rot. @ 5° increased from 34.6% to 60.3% (**25.7%** absolute improvement), PSNR improved from 13.44 to 15.30, and F1@0.01 increased from 57.2 to 68.2 (**11.0** point absolute improvement). These results confirm the effectiveness of our approach.

## 5 Conclusion

In this work, we presented SparseAGS, a framework for joint pose estimation and 3D reconstruction – combining off-the-shelf pose estimation methods with a novel-view synthesis generative prior for robust inference in real-world sparse-view captures.

**Limitations.** While our experiments demonstrated clear improvements over initializations and stronger performance compared to prior 3D reconstruction methods, there are several challenges that remain. First, our approach does rely on some *reasonable* off-the-shelf pose estimates and cannot succeed if a large fraction of the predictions have a large error. Secondly, SparseAGS (similar to existing baselines) does not deal with truncation or occlusion and cannot be directly applied to scenarios with close-up images of parts of objects or cluttered scenes with one object occluding the other. Finally, we focused here on an object-centric setting, and it would be interesting to extend our approach to broader settings, *e.g.*, deploying our framework in conjunction with methods that learn novel-view generative priors for scenes.

## Acknowledgements

We thank Zihan Wang and the members of the Physical Perception Lab at CMU for their valuable discussions. We especially appreciate Amy Lin and Zhizhuo (Z) Zhou for their assistance in creating figures, as well as Yanbo Xu and Jason Zhang for their feedback on the draft.

This work was supported in part by NSF Award IIS-2345610. This work used Bridges-2 [1] at Pittsburgh Supercomputing Center through allocation CIS240166 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

## References

- [1] Shawn T Brown, Paola Buitrago, Edward Hanna, Sergiu Sanielevici, Robin Scibek, and Nicholas A Nystrom. Bridges-2: A platform for rapidly-evolving and data intensive research. In *Practice and experience in advanced research computing*, 2021.
- [2] Xu Chen, Zijian Dong, Jie Song, Andreas Geiger, and Otmar Hilliges. Category level object pose estimation via neural analysis-by-synthesis. In *ECCV*, 2020.
- [3] Weihao Cheng, Yan-Pei Cao, and Ying Shan. Id-pose: Sparse-view camera pose estimation by inverting diffusion models, 2023. arXiv preprint arXiv:2306.17140.
- [4] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *CVPR*, 2022.
- [5] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. In *NeurIPS*, 2023.
- [6] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, 2023.
- [7] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *ICRA*, 2022.
- [8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.
- [9] Varun Jampani, Kevins-Kokitsi Maninis, Andreas Engelhardt, Arjun Karpur, Karen Truong, Kyle Sargent, Stefan Popov, André Araujo, Ricardo Martin Brumalla, Kaushal Patel, et al. Navi: Category-agnostic image collections with high-quality 3d shape and pose annotations. In *NeurIPS*, 2023.
- [10] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *CVPR*, 2014.
- [11] Hanwen Jiang, Zhenyu Jiang, Kristen Grauman, and Yuke Zhu. Few-view object reconstruction with unknown categories and camera poses. In *3DV*, 2024.
- [12] Hanwen Jiang, Zhenyu Jiang, Yue Zhao, and Qixing Huang. LEAP: Liberate sparse-view 3d modeling from camera poses. In *ICLR*, 2024.
- [13] Bharath Raj Nagoor Kani, Hsin-Ying Lee, Sergey Tulyakov, and Shubham Tulsiani. Upfusion: Novel view diffusion from unposed sparse view observations. In *ECCV*, 2024.
- [14] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. In *ACM TOG*, 2023.
- [15] Daniel Kersten, Pascal Mamassian, and Alan Yuille. Object perception as bayesian inference. In *Annu. Rev. Psychol.*, 2004.

[16] Yann Labb  , Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. Megapose: 6d pose estimation of novel objects via render & compare. In *CoRL*, 2022.

[17] Vincent Leroy, Yohann Cabon, and J  r  me Revaud. Grounding image matching in 3d with mast3r, 2024. arXiv preprint arXiv:2406.09756.

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

[19] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021.

[20] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*, 2023.

[21] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM SIGGRAPH*, 1987.

[22] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[23] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2023.

[24] 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.

[25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bj  rn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[26] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022.

[27] Mehdi SM Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lu  c  , Daniel Duckworth, Alexey Dosovitskiy, et al. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *CVPR*, 2022.

[28] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, et al. Zeronvs: Zero-shot 360-degree view synthesis from a single real image. In *CVPR*, 2024.

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

[30] Roger N Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. In *Science*, 1971.

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

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

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

[34] Yujing Sun, Caiyi Sun, Yuan Liu, Yuexin Ma, and Siu Ming Yiu. Extreme two-view geometry from object poses with diffusion models, 2024. arXiv preprint arXiv:2402.02800.

[35] Jiaxiang Tang. Stable-dreamfusion: Text-to-3d with stable-diffusion, 2022. <https://github.com/ashawkey/stable-dreamfusion>.

[36] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *ICLR*, 2024.

[37] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *CVPR*, 2021.

[38] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. In *CVPR*, 2023.

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

[40] Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. PF-LRM: Pose-free large reconstruction model for joint pose and shape prediction. In *ICLR*, 2024.

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

[42] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf-: Neural radiance fields without known camera parameters, 2021. arXiv preprint arXiv:2102.07064.

[43] Chin-Hsuan Wu, Yen-Chun Chen, Bolivar Solarte, Lu Yuan, and Min Sun. ifusion: Inverting diffusion for pose-free reconstruction from sparse views, 2023. arXiv preprint arXiv:2312.17250.

[44] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *CVPR*, 2023.

[45] Paul Yoo, Jiaxian Guo, Yutaka Matsuo, and Shixiang Shane Gu. Dreamsparse: Escaping from plato’s cave with 2d diffusion model given sparse views. In *NeurIPS*, 2023.

[46] Alan Yuille and Daniel Kersten. Vision as bayesian inference: analysis by synthesis? In *Trends in cognitive sciences*, 2006.

[47] 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.

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

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

[50] Chen Zhao, Tong Zhang, and Mathieu Salzmann. 3d-aware hypothesis & verification for generalizable relative object pose estimation. In *ICLR*, 2024.

[51] Guangyao Zhou, Nishad Gothoskar, Lirui Wang, Joshua B Tenenbaum, Dan Gutfreund, Miguel Lázaro-Gredilla, Dileep George, and Vikash K Mansinghka. 3d neural embedding likelihood: Probabilistic inverse graphics for robust 6d pose estimation. In *ICCV*, 2023.

[52] Zhizhuo Zhou and Shubham Tulsiani. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. In *CVPR*, 2023.

## Appendix

- Sec. A: Broader social impacts.
- Sec. B: Analysis of inference time.
- Sec. C: Implementation details.
- Sec. D: More detailed comparisons with SPARF [38].
- Sec. E: Additional results on three synthetic datasets.
- Sec. F: More qualitative comparison on camera pose accuracy with SPARF [38].
- Sec. G: More qualitative comparison on novel view synthesis with LEAP [12].

### A Broader Impacts

Our method leverages generative priors from diffusion models, enabling 3D reconstruction and pose estimation in the wild. This may benefit the generation of 3D assets for common users. However, we acknowledge that the web-scale data used for training these diffusion models may include content with potential negative social impacts, such as biased representations or harmful stereotypes. Therefore, while our approach benefits from the richness of the data, we must remain vigilant about the ethical implications and strive to mitigate any adverse effects.

### B Analysis of Inference Time

For 8-image inference using a single RTX A5000 GPU, one reconstruction with MV-DreamGaussian takes about 2 minutes to complete, and the “render-and-compare” for each outlier takes around a minute. Our full pipeline (using RayDiffusion initialization) detected an average of 0.94 outliers per sequence on NAVI, resulting in an inference time of around 9 minutes. We believe additional engineering efforts can further optimize and reduce inference time.

### C Implementation Details

**Finetuning Zero-1-to-3 with 6-DoF Camera Conditioning.** To learn the 6-DoF camera conditioning for novel view synthesis in the wild, we first initialize the weights of Zero-1-to-3 using the Zero123-XL checkpoint [5] and replace the original camera condition with ours. We then only finetune the layers associated with camera conditioning (*i.e.*, the linear projection and cross-attention layers) while freezing all other layers. This approach is more efficient than all-layer finetuning. To alleviate the synthetic data bias learned by the vanilla Zero-1-to-3, we include the training samples from CO3D [24] along with the Objaverse [6] renderings provided by Liu *et al.* [20] for finetuning. For computational resources, we used 8 V100 GPUs, setting a batch size of 36 per GPU with a gradient accumulation of 6. The model was trained for 23,500 iterations, taking ~4 days.

**Iterative Outlier Removal Details.** For the outlier condition specified in inequality 8, we employ LPIPS as the reprojection error metric, applying a threshold of 0.05. The reconstruction loop terminates when the average reprojection error reduction falls below this threshold or if the number of estimated inliers drops below a predefined count. Specifically, we use a threshold of 4 inliers for  $N = 6$  and  $N = 8$ , 6 inliers for  $N = 10$ , and 12 inliers for  $N = 16$ . These iteration counts generally suffice to handle outliers given the current capabilities of state-of-the-art pose estimation systems.

**Comparing with Pose-free Methods.** To compare with pose-free methods, we follow these steps to obtain their novel view renderings: First, we normalize the ground truth camera poses to match the scale of the coordinate systems used by these methods. Next, we render target images from novel views using their relative camera poses with respect to the first input image. Additionally, we adjust the camera intrinsics (focal length and principal point) during inference to align the foreground mask of the rendered images with the ground truth to reduce scale difference.

### D More Detailed Comparisons with SPARF

In addition to our main text comparing to SPARF for pose estimation, here we also present Novel View Synthesis (NVS) metrics. We report the results in Tab. 5, using DUST3R pose ( $N = 6, 8$ ) as initialization. Due to SPARF’s long training time (about 10 hours per instance), we could only include

Table 5: **Expanded Comparison of Pose Accuracy and Novel View Synthesis with SPARF [38].** In addition to the primary metrics presented in the main text, we report Average Rotation Error and Improvement Rate (IR), which indicates the percentage of sequences with reduced pose error. See Sec. D for further analysis and detailed explanations.

N = 6	Pose Metrics				NVS Metrics			
	Avg Rot. Err	Rot. @ 5°	CC @ 0.1	IR	PSNR	LPIPS	*PSNR	*LPIPS
DUST3R	7.90	48.5	80.2	/	/	/	/	/
w/ SPARF	17.07	47.9	67.6	0.41	12.80	0.3201	15.30	0.2478
w/ SparseAGS	5.82	73.9	92.1	0.80	15.52	0.2179	16.23	0.1844
N = 8	Pose Metrics				NVS Metrics			
	Avg Rot. Err	Rot. @ 5°	CC @ 0.1	IR	PSNR	LPIPS	*PSNR	*LPIPS
DUST3R	8.71	51.6	79.5	/	/	/	/	/
w/ SPARF	18.17	57.3	68.6	0.54	13.42	0.3059	15.44	0.2584
w/ SparseAGS	5.99	81.7	92.5	0.93	17.02	0.1874	17.48	0.1695

70 sequences (2 sequences per object) in these two experiments. Notably, for a direct comparison on NVS, we removed backgrounds from the input images, whereas no masking was applied in Tab. 1. This may slightly disadvantage SPARF, as backgrounds provide additional cues for pose registration and correspondences.

The results indicate that SparseAGS outperforms SPARF in pose accuracy and novel view quality as well. In fact, we found that SPARF can often make the poses worse compared to the (relatively accurate) DUST3R initialization (measured via *Improvement Rate* that indicates the percentage of sequences with reduced pose error). This is likely because the correspondences leveraged by SPARF in its optimization are not robust and are susceptible to false matches – see Fig. 7 for an example.

As an attempt to compare novel view quality despite the difference in pose accuracy, we report \*PSNR and \*LPIPS, which are measured *only on the sequences where SPARF improves pose accuracy* and find that even in these, our approach outperforms it. We also observed that while SPARF works well on novel views close to the input, floaters constantly appear with significant viewpoint changes. In contrast, our generative prior leads to a more consistent 3D representation.

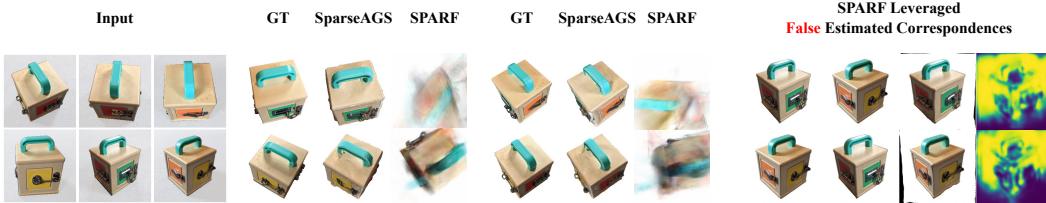


Figure 7: **SPARF Fails When Incorrect Correspondence is Leveraged.** Rightmost Section: From left to right – Source, Target, Warped Source to Target Based on Estimated Correspondence, Confidence Map (yellow indicates high confidence). The estimated false correspondence due to symmetric patterns causes pose optimization to fail, leading to degraded novel views in SPARF.

## E Evaluation on Synthetic Datasets

Table 6: **Evaluation of Rotation Accuracy on Three Synthetic Datasets (GSO [7], ABO [4] and OmniObject3D [44]).** We test our method on ID-Pose [3] with eight images as input. We measure rotation accuracy at two thresholds (15 and 30 degrees).

Method	GSO		ABO		OmniObject3D	
	Rot. @ 15°	Rot. @ 30°	Rot. @ 15°	Rot. @ 30°	Rot. @ 15°	Rot. @ 30°
ID-Pose	52.5	59.2	47.3	52.7	55.1	62.4
w/ SparseAGS	68.8(+16.3)	74.8(+15.6)	64.4(+17.1)	69.0(+16.3)	75.1(+20.0)	79.2(+16.8)

Though our main focus is real-world data, we apply our approach to three synthetic datasets (GSO, ABO, OmniObject3D) for a complete evaluation of our approach. We use ID-Pose [3] as a baseline,

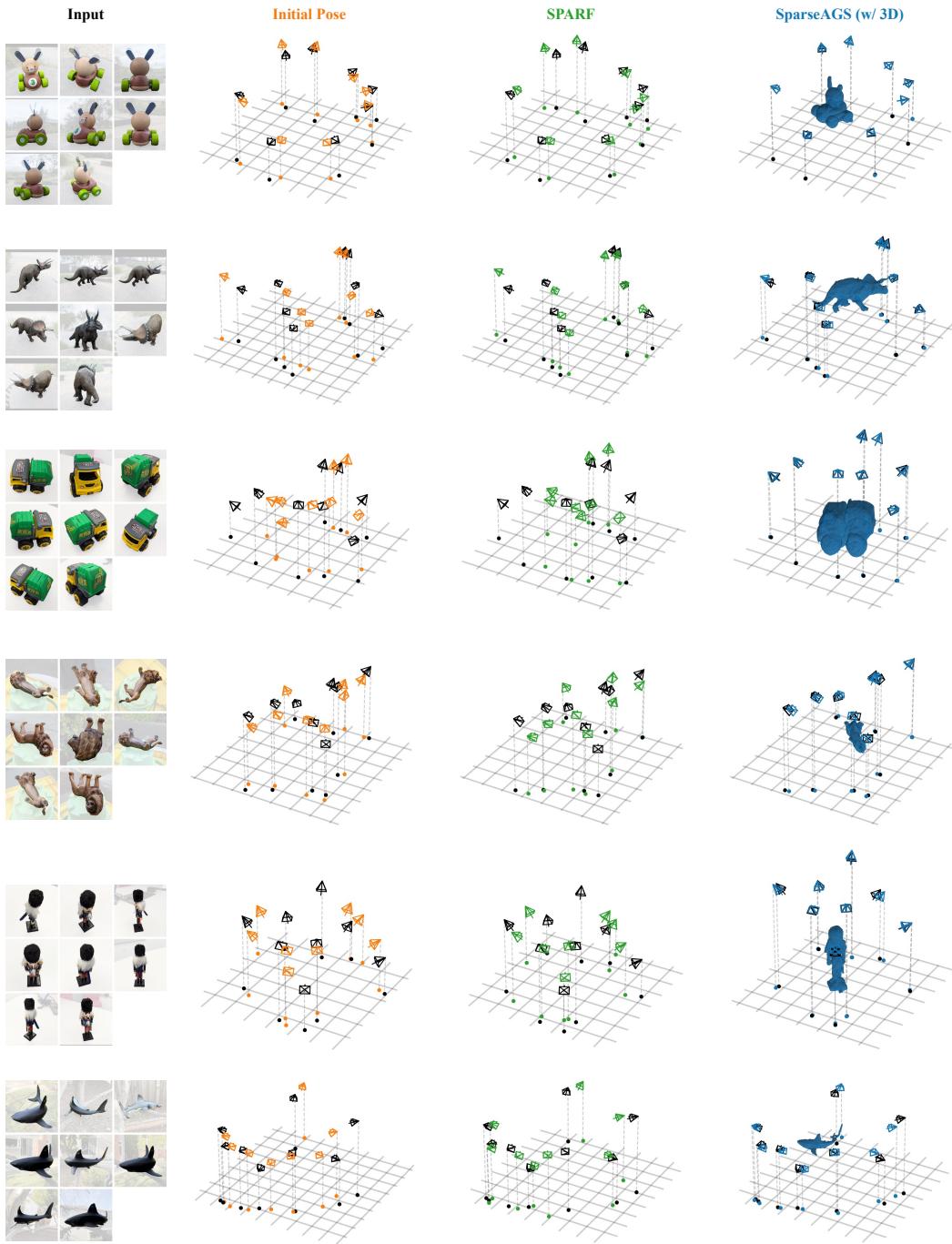
which inverses the novel-view-synthesis ability of Zero-1-to-3 [20] for pose estimation and adopts a 3-DoF camera parameterization. Here, we also use the vanilla Zero-1-to-3 for multi-view SDS loss to accommodate this camera definition and for fair comparison. We report the results on camera rotation accuracy in Tab. 6. Across these datasets, our approach consistently improves performance on two metrics, even though ID-Pose uses the same backbone model (vanilla Zero-1-to-3) as we do. These results demonstrate that our approach more effectively leverages the generative priors from Zero-1-to-3, achieving better pose accuracy. Plus, we also show that our method is applicable to synthetic data, showing strong generalization abilities across different datasets.

## F Additional Visualizations for Inferred Poses

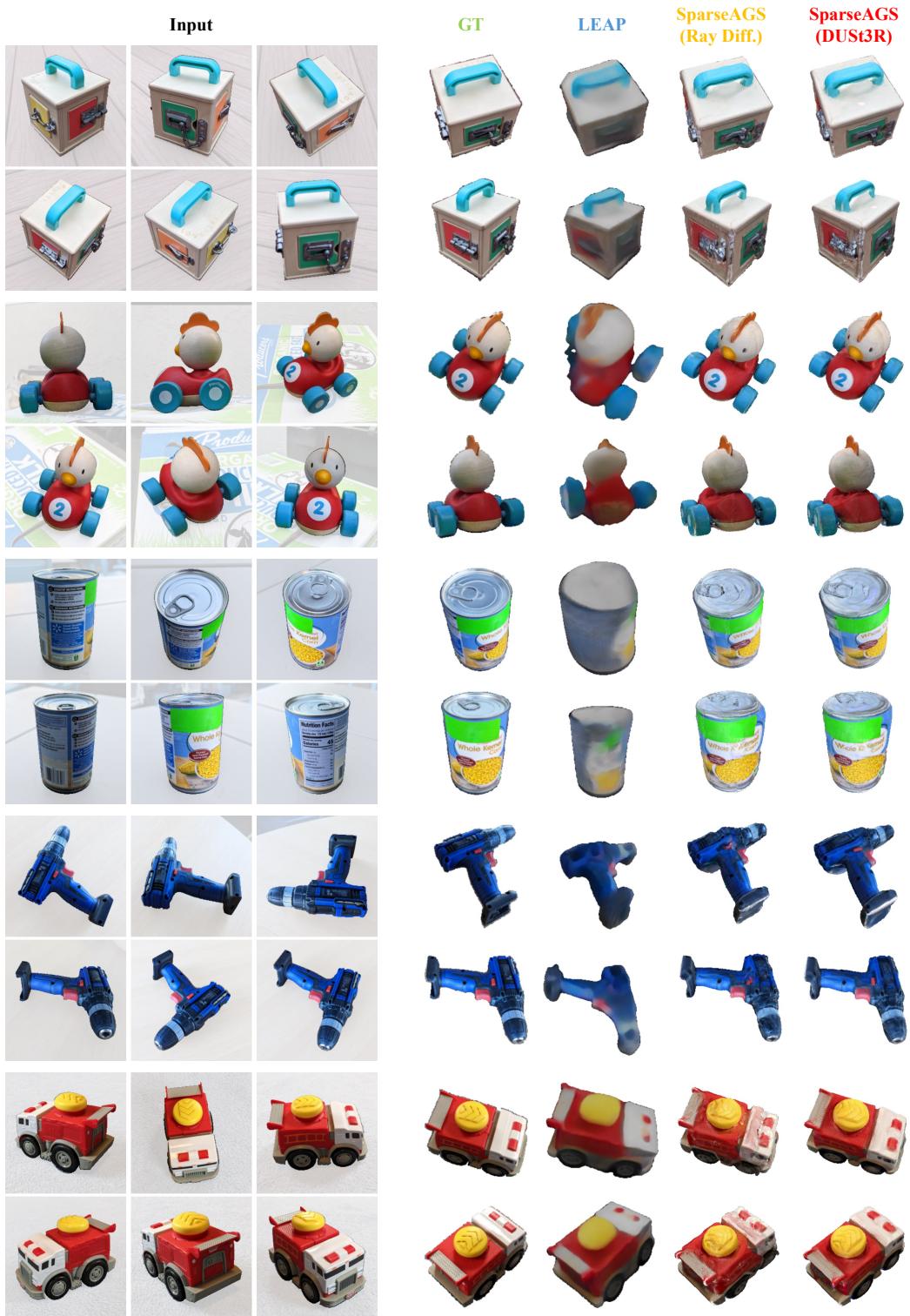
See Fig. 8.

## G Additional Visualizations for Novel View Synthesis

See Fig. 9.



**Figure 8: More Qualitative Comparison on Camera Pose Accuracy.** Given **initial** camera poses from off-the-shelf methods, the refined poses from **SPARF** [38] are compared with the output of **SparseAGS**. We align the estimated cameras with ground truth (in black) with an optimal similarity transform.



**Figure 9: More Qualitative Comparison with LEAP [12] on Novel View Synthesis.** We use two pose estimation baselines (Ray Diffusion [47] and DUS3R [41]). SparseAGS better preserves details in the input images and shows enhanced performance with more accurate initial camera poses.

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: **[Yes]**

Justification: We provide extensive experiments in Sec. 4 to support our claims.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: **[Yes]**

Justification: We include the discussions of our limitations in Sec. 5.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: **[NA]**

Justification: Our paper does not include theoretical results, and we conduct experiments to verify the effectiveness of our method.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

#### 4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: **[Yes]**

Justification: We include our experiment setup in Sec. 4.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

#### 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: **[No]**

Justification: We, unfortunately, are not able to clear data and codes before the camera-ready deadline. However, we will release our code as well as the data used soon.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: **[Yes]**

Justification: We include our experiment setup in Sec. 4.1 and details of our method in Sec. C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: **[No]**

Justification: We follow the standard practice in the community to evaluate our methods, which do not include error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

## 8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: **[Yes]**

Justification: We include such information in Sec. B and Sec. C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

## 9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [\[Yes\]](#)

Justification: We followed the code of ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

## 10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [\[Yes\]](#)

Justification: We discuss the social impacts in Sec. A.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [\[No\]](#)

Justification: Although our method leverages diffusion priors, we do not generate images directly. Instead, we reconstruct the 3D images from user-specified images. We will require the users to strictly follow the guidelines when our models are publicly available.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring

that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#)

Justification: We give proper credit to the resources we used mainly via citations and textual descriptions.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

## 13. New Assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[NA\]](#)

Justification: We currently do not introduce new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

## 14. Crowdsourcing and Research with Human Subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: Our paper does not include such content.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

**15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Our work does not include the related content.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.