arXiv:submit/6516740 [cs.CV] 5 Jun 2025

RaySt3R: Predicting Novel Depth Maps for Zero-Shot Object Completion

Bardienus P. Duisterhof[1] **Jan Oberst**[1,2] **Bowen Wen**[3]

Stan Birchfield[3] **Deva Ramanan**[1] **Jeffrey Ichnowski**[1]

[1]*Carnegie Mellon University* [2]*Karlsruhe Institute of Technology* [3]*NVIDIA*

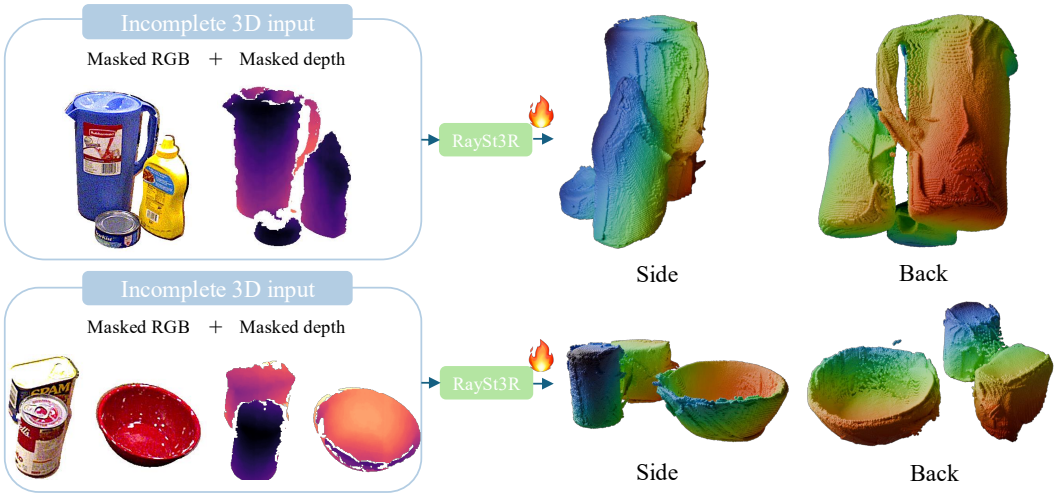


Figure 1: RaySt3R is a method for zero-shot 3D shape completion from a single foreground-masked RGB-D image. It predicts depth maps, object masks, and per-pixel confidence scores for novel viewpoints, and fuses them to reconstruct a complete 3D shape. RaySt3R is able to recover the geometry of full objects in cluttered real-world scenes, despite only being trained on synthetic data.

Abstract

3D shape completion has broad applications in robotics, digital twin reconstruction, and extended reality (XR). Although recent advances in 3D object and scene completion have achieved impressive results, existing methods lack 3D consistency, are computationally expensive, and struggle to capture sharp object boundaries. Our work (RaySt3R) addresses these limitations by recasting 3D shape completion as a novel view synthesis problem. Specifically, given a single RGB-D image and a novel viewpoint (encoded as a collection of query rays), we train a feedforward transformer to predict depth maps, object masks, and per-pixel confidence scores for those query rays. RaySt3R fuses these predictions across multiple query views to reconstruct complete 3D shapes. We evaluate RaySt3R on synthetic and real-world datasets, and observe it achieves state-of-the-art performance, outperforming the baselines on all datasets by up to 44 % in 3D chamfer distance. Project page: `rayst3r.github.io`

This work was generously supported by the Center for Machine Learning and Health (CMLH) at CMU, the NVIDIA Academic Grant Program, and the Pittsburgh Supercomputing Center.

1 Introduction

3D shape completion is an enabling tool for visual reasoning and physical interaction with partially visible objects, and facilitates a wide range of downstream tasks such as robot grasping in cluttered environments [27, 48], obstacle avoidance [30, 39], mechanical search [17], digital-twin reconstruction, and Augmented or Virtual or Extended Reality (AR/VR/XR) applications.

Challenges. We focus on the robotics-driven setting where an RGB-D image is provided as input for multi-object shape completion. While object-centric methods achieve high reconstruction quality for single objects, multi-object scenes require instance segmentation and alignment procedures [1] that tend to be brittle in practice. Generative approaches use 2D image generation models [57, 47] to generate images from novel viewpoints, but these can be sensitive to large viewpoint changes and are computationally inefficient at inference time, which hinders robot and XR deployment. Other methods scale up 3D prediction on abundant synthetic scene data [18], but the resolution for the 3D representations (such as 3D MAE voxel grids) is too coarse to capture sharp object shapes with high-frequency geometry details.

Approach. We propose Ray Stereo 3D Reconstruction (RaySt3R), a novel method for addressing the above challenges. Given a single masked RGB-D image as input, our key insight is to recast shape completion as a novel view-synthesis task, then aggregate multiple view predictions to generate a complete 3D shape. Our approach draws inspiration from recent work that casts 3D reconstruction as point map regression via multi-view transformers [46, 9]. We similarly use a vision transformer (ViT) [20] architecture defined over visual DINOv2 [31] features extracted from the input image. However, instead of requiring a second image as an additional input, we input the novel view to be synthesized in the form of a camera ray map. Specifically, RaySt3R is trained to predict depth maps, confidence maps, and foreground masks for each queried ray via cross-attention. We then merge RaySt3R's geometric predictions from multiple novel views using the per-ray confidence and mask predictions.

Data. Since RaySt3R can be seen as a view-synthesis engine, we can train at scale on pairs of RGB-D images without requiring volumetric 3D supervision (as required by prior work [18]). We train RaySt3R on a large-scale augmented synthetic dataset with 1.1 million scenes and 12 million views. Across synthetic and real-world benchmarks, RaySt3R outperforms prior art by up to 44 % (in shape completion accuracy). Despite never being trained on real data, RaySt3R generalizes well to real-world cluttered scenes.

This paper contributes:

- RaySt3R, a method for view-based 3D shape completion that learns confidence-aware depth maps and object masks from novel views, and uses a novel formulation of merging multi-view prediction.
- A new curated large-scale dataset with 12 million novel depth maps and masks, which we will open-source to facilitate future research.
- Evaluations of RaySt3R on synthetic and real-world datasets that show RaySt3R achieves state-of-the-art accuracy for 3D shape completion, and successfully generalizes to real-world cluttered scenes after training on only synthetic data.

2 Related work

3D shape completion has seen impressive progress over the last years. We explore related works categorized by their reasoning space, i.e., volumetric approaches and view-based approaches.

Volumetric reasoning Volumetric methods operate directly in 3D space and provide strong geometric priors. Some approaches directly predict point clouds [11, 28, 41], while others rely on implicit geometry representations. The latter infer 3D structure at test time by querying the representation with a spatial point and a partial observation (e.g., an RGB or RGB-D scan). Prior work uses signed distance functions for such representations [32, 21] or voxel occupancy grids [3, 4, 33, 16, 29, 51, 22]. OctMAE [18] builds on the idea of MAE [13] from the image synthesis domain, and applies it to next-token prediction natively in 3D. Although these methods yield promising results, their resolution is constrained by the cubic cost of their volumetric resolution, leading to a coarse grid and smoothed structures lacking fine details.

Another strategy decomposes the scene at the object level [1, 56]. SceneComplete [1] constructs a 3D scene by chaining together foundation models for object segmentation, occlusion inpainting, 3D shape retrieval, and pose estimation. Despite its modular design, our experiments (Section 5) suggest that this reliance on multiple components introduces brittleness and several single points of failure.

Recent work like TRELLIS [53] instead learns a 3D latent representation from text or image, which can be decoded into various formats such as meshes. However, as shown in Section 5, experiments suggest it struggles in real-world multi-object scenes. In contrast, RaySt3R adopts a view-based strategy that is specifically designed for robust 3D completion in cluttered environments.

View-based reasoning Diffusion models [14] and their extensions [15, 37, 36] have enabled unprecedented performance in generative tasks such as image synthesis, inpainting, and video prediction. Several works leverage off-the-shelf generative models for 3D generation tasks [25, 57, 42, 52, 12]. ViewCrafter [57] leverages these advances by iteratively completing a scene point cloud using a point-conditioned video diffusion model. More recently, Li et al. [23] predicts layered depth maps for constructing object-level and scene-level 3D geometries. While this method yields promising results on single-object shape completion, it struggles with predicting accurate geometries in real-world scenes containing multiple objects. Unique3D [52] tries to strike a balance between fidelity and inference speed by predicting multi-view images, generating corresponding normal maps and a textured mesh within 30 seconds.

While these models often yield visually appealing results, they lack geometric consistency, especially for cluttered real world environments. The inference time of large diffusion models may also hinder deployment in robotics or XR settings. In contrast, RaySt3R predicts geometrically accurate depth maps from novel views, for fast and accurate 3D shape completion in cluttered real-world scenes.

3 Problem statement

Given a single RGB-D image, $I^{\text{input}} \in \mathbb{R}^{H\times W\times 3}, D^{\text{input}} \in \mathbb{R}^{H\times W}$, foreground mask $M \in \{0,1\}^{H\times W}$, and a camera with known intrinsics, $K^{\text{input}} \in \mathbb{R}^{3\times 3}$, the goal is to predict the full 3D surface geometry of all masked foreground objects.

We frame the prediction goal as a set of points $Q \in \mathbb{R}^{N\times 3}$ that is both *accurate* and *complete* w.r.t. the ground-truth points $Q^{\text{gt}} \in \mathbb{R}^{S\times 3}$, sampled on the surfaces (e.g., meshes) of all objects in the scene.

We measure accuracy as the shortest distance from a predicted point to the nearest ground-truth point, averaged over all predicted points. We measure completeness as the shortest distance from a ground-truth point to the nearest predicted point, averaged over all ground truth points.

4 Methods

An overview of our approach is illustrated in Figure 2. We propose to train a transformer that, given the partial capture from a single RGB-D image and foreground mask, predicts depth maps and per-pixel confidence scores, and foreground masks for novel views. We first present the model architecture (Section 4.1), then the training objectives (Section 4.2). We then describe the procedure for querying novel views (Section 4.3) and conclude with the prediction merging strategy (Section 4.4).

4.1 Network architecture

The RaySt3R network architecture is inspired by DUSt3R [46] and successors [45, 47, 43]. Here, we leverage a ViT with point map, ray map, and depth map representations for 3D object completion.

The inputs to RaySt3R (Figure 2) are a foreground-masked RGB-D image and a novel query view. First, we unproject the input depth map D^{input} to a point map $X^{\text{input}} \in \mathbb{R}^{H\times W\times 3}$ using the given input image intrinsics K^{input}, thus $X^{\text{input}}_{i,j} = (K^{\text{input}})^{-1}[iD^{\text{input}}_{i,j}, jD^{\text{input}}_{i,j}, D^{\text{input}}_{i,j}]^{\mathsf{T}}$. We convert the query view into a ray map $R \in \mathbb{R}^{H\times W\times 2}$, with $R_{i,j} = [(i-c_x)/f_x, (j-c_y)/f_y]^{\mathsf{T}}$, where c_x, c_y is the image center and f_x, f_y are the focal lengths of the novel target view.

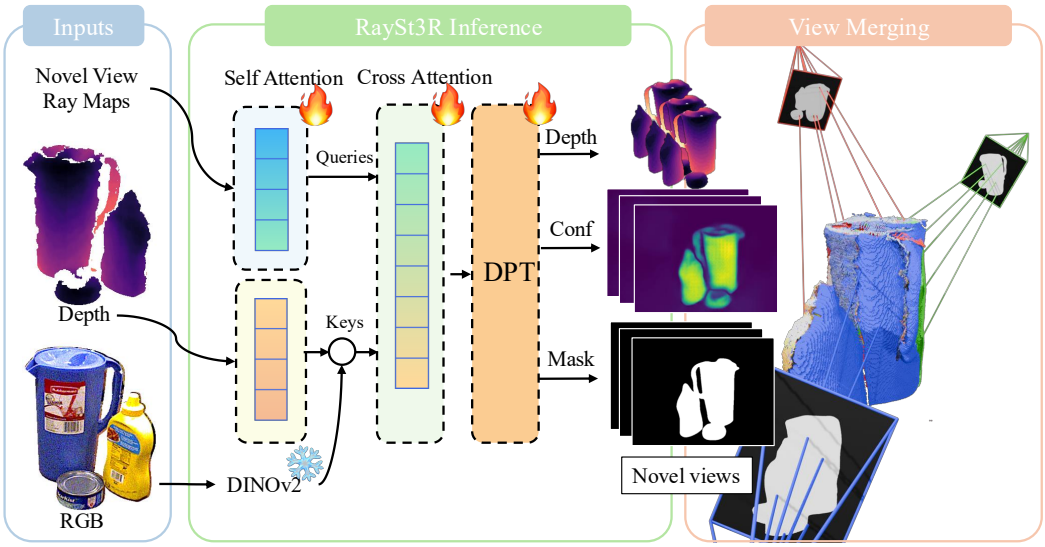


Figure 2: The architecture of RaySt3R. RaySt3R takes a single RGB-D image and foreground mask as input, and predicts depth maps, object masks, and per-pixel confidence scores for novel views. First, we apply the foreground mask to the RGB and point map input. Next, we use self-attention layers for the point map and ray map inputs, and feed the RGB image into the frozen DINOv2 [31] encoder. We feed all features into cross-attention layers followed by two separate DPT heads [34] for depth and mask predictions. Finally, we provide confidence- and occlusion-aware multi-view merging formulation.

Because we would like to pass information between the input and query views via cross attention, we transform the input point map X_{input} into the target camera coordinate frame to ease information sharing. That is, $X^{\text{context}} = P_{\text{input}}^{\text{query}} h(X^{\text{input}})$, where $P_{\text{input}}^{\text{query}} \in \mathbb{R}^{3\times4}$ transforms from the input to query camera coordinate frame, and $h : (x, y, z) \mapsto (x, y, z, 1)$.

We compute features for the point map and ray query with L layers of self-attention (SA).

$$F^{\text{point_map}} = \text{SA}(X^{\text{context}}), \quad F^{\text{ray}} = \text{SA}(R) \tag{1}$$

We mask out the background in X^{context} and replace it with a single learned background token. We process the RGB image by first masking out the background, and subsequently passing it through a frozen DINOv2 [31] encoder. Recent work has shown that a combination of features from different layers of a pre-trained ViT is useful for downstream tasks [10], hence we concatenate the features from intermediate layers of the DINOv2 encoder, and use a linear layer to project them to F^{DINO}.

For the cross-attention (CA) layers, we construct the keys of the first layer by concatenating $F^{\text{point_map}}$ and F^{DINO}. The queries are the ray features F^{ray}.

$$G = \text{CA}(F^{\text{ray}}, \text{concat}(F^{\text{point_map}}, F^{\text{DINO}})) \tag{2}$$

Finally, we use a DPT head [34] to predict depth maps, and its confidence scores. A separate DPT head predicts the object mask.

4.2 Training objectives

We train RaySt3R to predict confidence-aware depth maps and object masks.

Depth loss Inspired by DUSt3R [46], we define the confidence-aware depth loss:

$$\mathcal{L}_{\text{depth}} = \sum_{i\in[0,W-1]} \sum_{j\in[0,H-1]} M_{i,j}^{\text{gt}} \left(C_{i,j} \left\| d_{i,j} - d_{i,j}^{\text{gt}} \right\|_2 - \alpha \log C_{i,j} \right). \tag{3}$$

Here, $C_{i,j}$ is the confidence score of each pixel in the predicted depth map, α is a hyper parameter, $d_{i,j}$ is the predicted depth, $d_{i,j}^{\text{gt}}$ is the ground-truth depth, and $M_{i,j}^{\text{gt}}$ is the ground-truth mask. The confidence scores are enforced to be strictly positive by setting $C_{i,j} \leftarrow 1 + \exp(C_{i,j})$. This enables a confidence estimate without explicit supervision.

We also predict binary object masks from novel viewpoints, and use a binary cross entropy loss to supervise it during training.

$$\mathcal{L}_{\text{mask}} = \sum_{i \in [0,W-1]} \sum_{j \in [0,H-1]} \left(-m^{\text{gt}}_{i,j} \log(m_{i,j}) - (1 - m^{\text{gt}}_{i,j}) \log(1 - m_{i,j}) \right) \tag{4}$$

Here, $m_{i,j}$ is the predicted object mask after a sigmoid operation, and $m^{\text{gt}}_{i,j}$ is the ground-truth object mask. Finally, we combine the depth and mask losses with a sum weight λ_{mask}.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{depth}} + \lambda_{\text{mask}} \mathcal{L}_{\text{mask}} \tag{5}$$

4.3 View sampling

To construct a set of query views to sample, we fit a tight bounding box to the input view point map and sample points on a sphere with radius $\lambda_{bb} r_{bb}$ around the box's center. Here, λ_{bb} is a tunable parameter, and r_{bb} is half the length of the bounding-box diagonal. We found degenerate cases where the bounding box is too small, thus we clip the radius to be at least $\lambda_{\text{cam}} r_{\text{cam}}$, where r_{cam} is the distance from the camera to the center of the bounding box and λ_{cam} is a tunable hyperparameter. We sample points evenly on a cylindrical equal-area projection of the sphere to improve the coverage of the scene. We don't include the input point map in our predictions as it likely contains noise and artifacts. Instead, query RaySt3R with the input view and include it in our predictions.

4.4 Merging predictions

After predicting depth maps and object masks for all novel views, we merge them to produce a complete 3D shape. We merge the depth maps by accounting for occlusions, RaySt3R's predicted masks, and the confidence scores.

Occlusion handling: First, we filter the points in each novel view to only parts of the scene that were not visible in the input image (i.e., those points occluded by the input view's foreground mask M^{input} and depth map D^{input}). Each point $q_{n,i,j}$ is defined as the point predicted by the n-th novel view at pixel (i, j). Its projection in the input view is given by $p_{n,i,j} = K_{\text{input}} P^{\text{input}}_n h(q_{n,i,j})$, where P^{input}_n transforms points from the n-th novel view to the input view. We define each entry of the mask as:

$$m^{\text{occ}}_{n,i,j} = \begin{cases} 1 & \text{if } (p_{n,i,j})_z > D^{\text{input}}_{i,j} \text{ and } M^{\text{input}}_{i,j} = 1 \\ 0 & \text{otherwise} \end{cases}. \tag{6}$$

RaySt3R predicted masks: Even with the occlusion constraint, the object mask from a novel view is largely unknown. For example, any observed surface could be a thin plate or a rich 3D object. We use the predicted mask from RaySt3R to filter the points, by thresholding the predicted mask $m^{\text{RaySt3R}}_{i,j} \in [0, 1]$ at 0.5.

Confidence scores: RaySt3R's architecture enables unsupervised confidence scores for each pixel in the predicted depth maps. Confidence scores are typically used to reduce edge-bleeding in dense ViT predictions [46, 43], or to exclude out-of-distribution objects such as specularities. With the same objective, we threshold $c^{\text{RaySt3R}}_{i,j}$ at τ for all experiments, more analysis is provided in Section 5.8.

Final mask: The final valid mask for a given novel view is obtained by setting

$$m_{n,i,j} = m^{\text{occ}}_{n,i,j} \cdot \mathbf{1}\left[m^{\text{RaySt3R}}_{n,i,j} > 0.5\right] \cdot \mathbf{1}\left[c^{\text{RaySt3R}}_{n,i,j} > \tau\right] \tag{7}$$

We obtain our final 3D reconstruction by aggregating valid points across all novel views.

5 Results

5.1 Training dataset

RaySt3R's training procedure requires a large number of camera pairs to scale zero-shot to the real world. It requires an RGB image for the input view and depth maps, intrinsics, and extrinsics for

both cameras. We leverage existing synthetic datasets from FoundationPose [50] and OctMAE [18]. OctMAE [18] places GSO [8] and Objaverse [7] objects in synthetic scenes, and provide a single rendered image and depth map for each scene. FoundationPose has separate GSO and Objaverse splits, we only use the GSO split. For both datasets, we use the Objaverse and GSO meshes to render depth maps from novel views. Our dataset spans 1.1 million scenes with 12 million views in total.

5.2 Evaluation datasets

We evaluate RaySt3R on synthetic and real-world datasets. Following OctMAE [18], we evaluate on subsets of evaluation splits of the YCB-Video [54] (900 frames), HOPE [40] (50 frames), and HomebrewedDB [19] (1,000 frames) datasets. They are real-world 6D pose estimation datasets with noisy depth maps and imperfect masks, including common objects such as boxes and cylinders, as well as items of complex geometries such as metal parts. For results on synthetic data, we evaluate on evaluation split of the OctMAE [18] (1,000 frames) dataset test split. We notice edge artifacts in the masks introduced due to data compression in the original work [18].

5.3 Data augmentation

Synthetic training data lacks noise and other artifacts as present in the real world. We therefore apply data augmentation during training to better bridge the sim-to-real gap. Inspired by [50, 49, 6], we apply a set of augmentations to the input views at training time. For depth maps, we randomly apply Gaussian noise, add holes, and shift the pixel coordinates [2, 6]. For the RGB image, we randomly vary brightness and contrast, and apply a per-channel salt and pepper noise and Gaussian noise.

5.4 Implementation details

We train RaySt3R on 8$\times$ 80-GB A100 GPUs for 18 epochs, totaling approximately 20 million scene iterations. We set the batch size to 10 per GPU, and a learning rate of 1.5×10^{-4} with a half-cosine learning-rate schedule, starting with one warm-up epoch and using an AdamW optimizer [26]. We use a ViT-B model with patch size 16, embedding dimension 768, 12 heads, 12 cross-attention layers, but 4 self-attention layers to save on compute. We select the ViT-L with registers for DINOv2 [31].

We set $\lambda_{bb} = 1.3$ and $\lambda_{\text{cam}} = 0.7$ for all real-world datasets, and $\lambda_{bb} = 2.5$ and $\lambda_{\text{cam}} = 1.2$ for the OctMAE dataset. The parameters are chosen to be larger for the OctMAE dataset, as the input view is typically placed very close to the objects with severe occlusions. We set the confidence threshold $\tau = 5$ for all experiments, and sample 22 views in total. During training we set the confidence parameter $\alpha = 0.2$, $\lambda_{\text{mask}} = 0.1$. Inference takes less than 1.2 seconds on a single RTX 4090 GPU, and can be further reduced by querying fewer views.

5.5 Baselines

We compare RaySt3R against the state-of-the-art in 3D shape completion. OctMAE introduced a novel 3D MAE algorithm, and also trained prior shape completion models on their novel dataset. We compare against OctMAE, and the prior works they trained, i.e., VoxFormer [24], ShapeFormer [55], MCC [51], ConvONet [33], POCO [3], AICNet [22], Minkowski [5], and OCNN [44].

We also compare against SceneComplete [1], which uses a combination of foundation models to produce complete geometry. The authors leverage a VLM and Grounded-SAM [35] to produce object-level masks, image inpainting to fill in occluded regions, an image-to-3D model to produce a 3D mesh, and finally FoundationPose for 3D alignment.

We also benchmark against Unique3D [52] and TRELLIS [53], which are recent image to 3D models. Finally, we compare against 'Layered Ray Intersections' (LaRI) [23], which introduced the concept of layered point maps to predict multiple points on each camera ray. Unique3D, Trellis, and Lari predict points in canonical coordinates, we align the predictions with the ground truth 3D using first a brute-force search for a similarity transform, followed by ICP [38]. Note that we do not perform such a registration for RaySt3R, but provide this to baselines to give them the benefit of the doubt. While LaRi [23] and Unique3D [52] do not require foreground masks for reconstructing objects, we observe that TRELLIS [53] tends to reconstruct the entire scene. Therefore, we compare TRELLIS with a masked image input and a raw image input. We also attempted to evaluate ViewCrafter [57] on

this task, but the images produced by the video diffusion model were of too poor quality to perform the evaluation. We provide more details in the supplementary material.

5.6 Quantitative results

Following prior work [18], we evaluate the zero-shot generalization performance of all methods using chamfer distance (CD) and F1-Score@10mm (F1). Detailed formulations of the metrics are provided in the supplemental material. We present the quantitative results of this evaluation in Table 1. RaySt3R consistently outperforms all baselines across both synthetic and real-world scenes. The strongest baseline, OctMAE [18], performs competitively, however RaySt3R surpasses it across all metrics, by 20 % to 44 % in CD. SceneComplete [1] proves to be a fragile pipeline, therefore not yielding competitive results. We were unable to produce SceneComplete results on HOPE, as the pipeline requires an intractable amount of VRAM for cluttered scenes. LaRI [23], Unique3D [52], and TRELLIS [53] show better performance than SceneComplete [1], but also do not produce competitive results. Feeding masked RGB images to TRELLIS [53] outperforms raw image inputs on all datasets except HomebrewedDB [19]. We show common failure modes in Section 5.7.

We also compute the standard deviation of the chamfer distance across all real-world datasets for each method and observe that our model exhibits the lowest standard deviation (1.74 mm), followed by OctMAE [18] (2.38 mm) and Unique3D [52] (8.11 mm).

Table 1: Quantitative evaluation of multi-object scene completion on synthetic and real-world datasets. We evaluate on the test split of OctMAE [18], and the BoP benchmarks YCB-Video [54], HOPE [40], and HomebrewedDB [19]. We report chamfer distance (CD) [mm] and F1-Score@10mm (F1). The first section contains numbers copied from OctMAE [18], the second section contains recent works we evaluated. For alignment of LaRI [23], Unique3D [52], and TRELLIS [53] with the ground truth mesh, we apply brute force search followed by ICP [38]. We evaluate TRELLIS [53] with masked and unmasked RGB inputs. SceneComplete [1] runs out of VRAM on HOPE [40]. The results suggest RaySt3R outperforms all baselines.

	Synthetic		Real					
	OctMAE [18]		YCB-Video [54]		HB [19]		HOPE [40]	
Method	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑
VoxFormer [24]	44.54	0.382	30.32	0.438	34.84	0.366	47.75	0.323
ShapeFormer [55]	39.50	0.401	38.21	0.385	40.93	0.328	39.54	0.306
MCC [51]	43.37	0.459	35.85	0.289	19.59	0.371	17.53	0.357
ConvONet [33]	23.68	0.541	32.87	0.458	26.71	0.504	20.95	0.581
POCO [3]	21.11	0.634	15.45	0.587	13.17	0.624	13.20	0.602
AICNet [22]	15.64	0.573	12.26	0.545	11.87	0.557	11.40	0.564
Minkowski [5]	11.47	0.746	8.04	0.761	8.81	0.728	8.56	0.734
OCNN [44]	9.05	0.782	7.10	0.778	7.02	0.792	8.05	0.742
OctMAE [18]	6.48	0.839	6.40	0.800	6.14	0.819	6.97	0.803
LaRI [23]	39.22	0.283	11.41	0.658	22.23	0.414	18.64	0.528
Unique3D [52]	44.62	0.244	17.56	0.468	25.41	0.329	26.37	0.322
TRELLIS (w/ mask) [53]	61.74	0.227	22.94	0.443	35.29	0.354	20.25	0.443
TRELLIS (w/o mask) [53]	65.74	0.225	31.74	0.338	30.71	0.348	22.05	0.416
SceneComplete [1]	81.57	0.289	96.63	0.359	85.81	0.416	N/A	N/A
RaySt3R (ours)	**5.21**	**0.893**	**3.56**	**0.930**	**4.75**	**0.889**	**3.92**	**0.926**

5.7 Qualitative results

Figure 3 shows qualitative results of RaySt3R on real-world datasets. The results suggest that RaySt3R is capable of generating high-quality 3D predictions, generating more consistent and complete results compared to the baselines. The most competitive baseline, OctMAE [18], produces viable but oversmoothed object shapes due to its low-resolution 3D MAE grid. Furthermore, TRELLIS [53], Unique3D [52] and LaRI [23] struggle with relative object placement and aspect ratios. They also fail for certain out-of-distribution objects, and occasionally fail to register to the ground truth point cloud. Interestingly, TRELLIS [53] may predict table surfaces even for masked input images with no visible table. SceneComplete [1] proves to be brittle to single-point failures such as missing object masks.

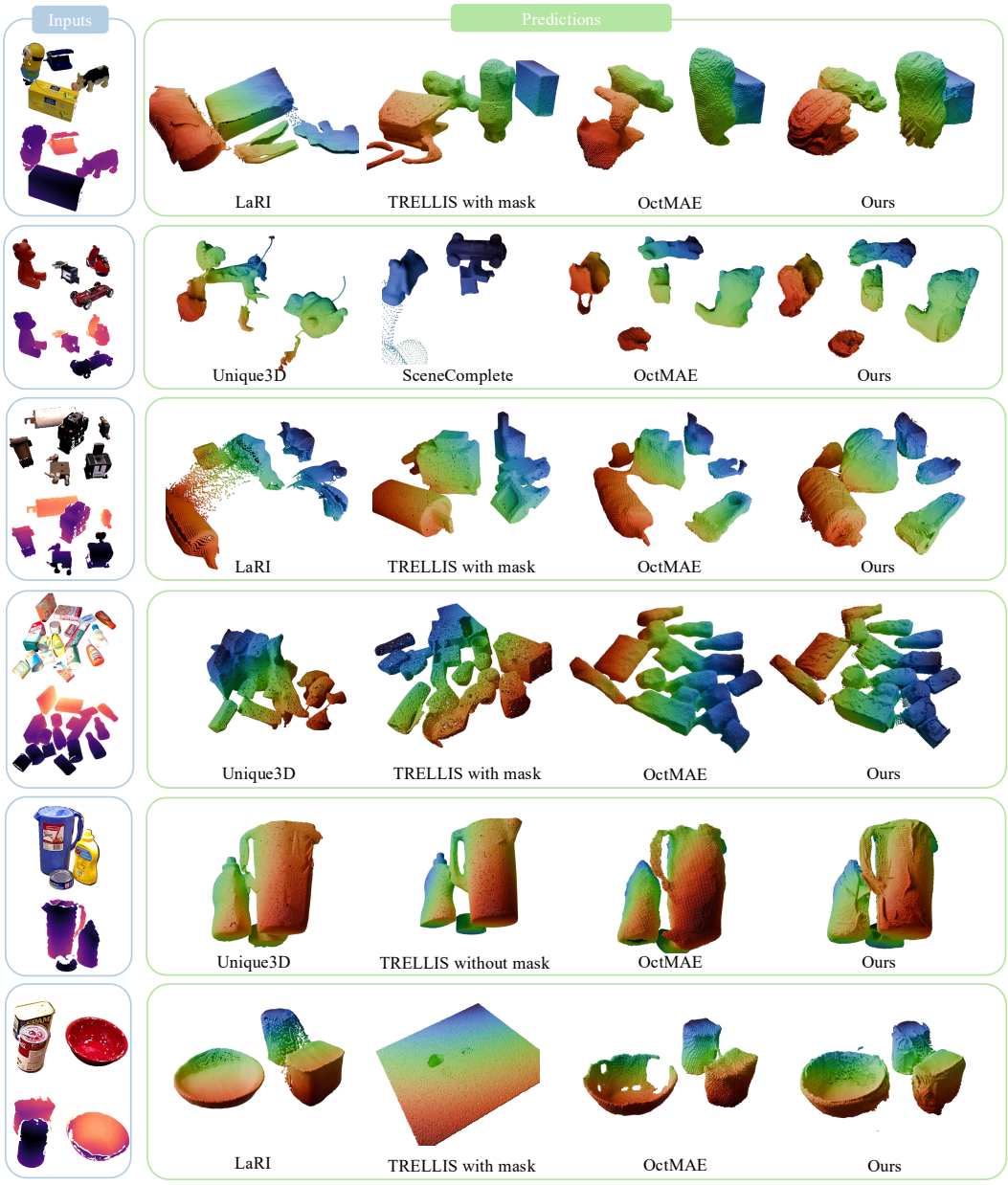


Figure 3: Qualitative results of RaySt3R in real-world multi-object scenes. For each scene, we select a subset of methods to preserve visual clarity, but we share all method predictions in the supplemental material. The results suggest RaySt3R produces the most geometrically accurate shapes compared to the baselines. The most competitive baseline, OctMAE [18], tends to predict softer shapes as a result of its coarse 3D MAE grid. We observe TRELLIS [53], Unique3D [52], and LaRI [23] struggle with relative object placing, aspect ratios, and out-of-distribution objects, occasionally leading to incorrect registration to the ground truth points. Finally, SceneComplete [1] proves to be brittle to single points of failure such as missing object masks.

5.8 Ablation studies

Training ablations: Table 2 summarizes our training ablation results, with all models trained under the same setup for roughly 20 million scene iterations. Training a ViT-S model (384-dim, 6 heads) leads to a performance drop. Disabling data augmentation further degrades zero-shot generalization to real-world data, highlighting its importance. We also compare training on 100k uniformly sampled scenes versus the full 226k GSO set. The smaller but more diverse 100k set performs better, emphasizing the value of data diversity. Finally, removing DINOv2 [31] inputs causes an additional decline, underscoring the benefit of pretrained features.

Table 2: Ablation study on RaySt3R training on the YCB-Video dataset [54]. The results suggest data scale, data diversity, DINOv2 [31] features, data augmentation, and model size affect RaySt3R's performance.

Method name	CD ↓	F1 ↑
RaySt3R (proposed)	**3.56**	**0.930**
ViT-S	3.70	0.920
No data augmentation	3.89	0.916
Train on 100k scenes	4.30	0.894
w/o DINOv2 [31]	4.81	0.877
Train on 226k GSO scene	5.34	0.864

Table 3: Ablation study on RaySt3R view merging on the OctMAE test dataset [18]. We ablate querying the input view (Section 4.3), occlusion masking with the input mask, and finally RaySt3R's predicted masks. The results suggest all steps contribute to performance, especially the predicted masks.

Query Input	Occ. Mask	Pred. Mask	CD↓	F1↑
✓	✓	✓	**5.21**	**0.893**
✗	✓	✓	7.55	0.836
✓	✗	✓	7.69	0.855
✓	✓	✗	10.12	0.825
✗	✗	✗	73.17	0.641

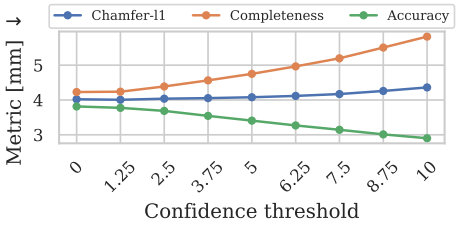


Figure 4: Confidence threshold vs error metrics averaged over all real-world datasets. The results suggest increasing the confidence threshold improves accuracy and degrades completeness.



Figure 5: The impact of the confidence threshold on the predicted 3D points. This experiment suggests increasing the confidence threshold can aid in reducing the edge bleeding issue.

View merging ablations: Table 3 shows the ablations on our view merging formulation. We ablate querying the input view (Section 4.3), using the input mask to detect occluded regions, and finally the use of RaySt3R's predicted mask. The results suggest that each component of our formulation contributes to shape completion performance, especially the predicted masks.

Confidence RaySt3R predicts a per-pixel confidence value, which can be used to filter the predictions. To understand the impact of the confidence threshold, we report chamfer distance, Completeness, and Accuracy for a range of thresholds, as depicted in Figure 4. The results suggest that confidence is a good proxy for error and that confidence can be effectively used to trade off accuracy and completeness. Depending on the application, the threshold can therefore be tuned accordingly, as some applications may be less tolerant to outliers requiring high accuracy, while others may require more complete predictions. For all prior experiments, we set confidence threshold τ to 5 for a balance between accuracy and completeness. Figure 5 shows a visualization of the impact of changing the confidence threshold on the predicted 3D points.

6 Conclusion and future work

We present RaySt3R, a novel approach to 3D shape completion from a single RGB-D image and foreground mask. RaySt3R learns to predict depth maps, object masks, and per-pixel confidence scores for novel viewpoints, which are fused to reconstruct complete 3D shapes. We benchmark RaySt3R on real-world and synthetic datasets, and compare it to the state-of-the-art in volumetric and view-based methods. The results suggest that RaySt3R is capable of generating high-quality 3D predictions, outperforming the baselines across the board.

RaySt3R's results could be further improved by training on real-world data and scaling up compute. The current synthetic dataset might be too limited to capture increasingly complex objects. Future work may also explore scaling up compute by exploring other architectures like diffusion transformers.

Table 4: Standard deviations (STD) of Chamfer Distance (CD) [mm] and F1-Score@10mm (F1) for multi-object scene completion methods evaluated on synthetic (OctMAE [18]) and real datasets (YCB-Video [54], HOPE [40], and HomebrewedDB [19]). Metric values are provided for context, standard deviations are denoted as subscripts. The results suggest RaySt3R consistently achieves lower metrics and STD.

	Synthetic		Real					
	OctMAE [18]		YCB-Video [54]		HB [19]		HOPE [40]	
Method	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑
OctMAE [18]	$6.48_{\pm 28.52}$	$0.839_{\pm 0.104}$	$6.40_{\pm 2.269}$	$0.800_{\pm 0.066}$	$6.14_{\pm 2.394}$	$0.819_{\pm 0.074}$	$6.97_{\pm 3.446}$	$0.803_{\pm 0.082}$
LaRI [23]	$39.22_{\pm 23.89}$	$0.283_{\pm 0.143}$	$11.41_{\pm 7.422}$	$0.658_{\pm 0.195}$	$22.23_{\pm 9.650}$	$0.414_{\pm 0.146}$	$18.64_{\pm 14.23}$	$0.528_{\pm 0.202}$
Unique3D [52]	$44.62_{\pm 26.99}$	$0.244_{\pm 0.122}$	$17.56_{\pm 7.180}$	$0.468_{\pm 0.144}$	$25.41_{\pm 6.951}$	$0.329_{\pm 0.074}$	$26.37_{\pm 8.313}$	$0.322_{\pm 0.075}$
TRELLIS (w/ mask) [53]	$61.74_{\pm 69.06}$	$0.227_{\pm 0.135}$	$22.94_{\pm 12.13}$	$0.443_{\pm 0.207}$	$35.29_{\pm 62.04}$	$0.354_{\pm 0.154}$	$20.25_{\pm 10.75}$	$0.443_{\pm 0.155}$
TRELLIS (w/o mask) [53]	$65.74_{\pm 104.9}$	$0.225_{\pm 0.135}$	$31.74_{\pm 53.39}$	$0.338_{\pm 0.141}$	$30.71_{\pm 73.71}$	$0.348_{\pm 0.092}$	$22.05_{\pm 11.47}$	$0.416_{\pm 0.121}$
SceneComplete [1]	$81.57_{\pm 219.9}$	$0.289_{\pm 0.135}$	$96.63_{\pm 100.6}$	$0.359_{\pm 0.120}$	$85.81_{\pm 188.6}$	$0.416_{\pm 0.168}$	N/A	N/A
RaySt3R (ours)	$\mathbf{5.21}_{\pm \mathbf{7.836}}$	$\mathbf{0.893}_{\pm \mathbf{0.085}}$	$\mathbf{3.56}_{\pm \mathbf{1.194}}$	$\mathbf{0.930}_{\pm \mathbf{0.038}}$	$\mathbf{4.75}_{\pm \mathbf{1.976}}$	$\mathbf{0.889}_{\pm \mathbf{0.070}}$	$\mathbf{3.92}_{\pm \mathbf{0.9229}}$	$\mathbf{0.926}_{\pm \mathbf{0.043}}$

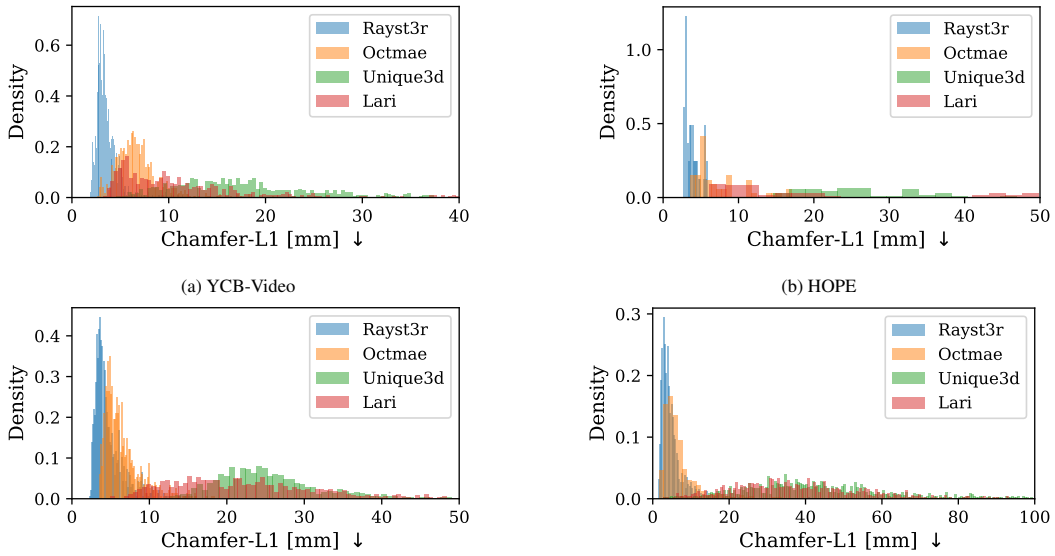
7 Acknowledgements

This work was generously supported by the Center for Machine Learning and Health (CMLH) at CMU, the NVIDIA Academic Grant Program, and the Pittsburgh Supercomputing Center. The authors would like to thank Mandi Zhao, Shun Iwase, Balázs Gyenes, Gerhard Neumann, and all memebers of the Momentum Robotics lab at CMU for providing useful feedback.

8 Supplemental material

8.1 Quantitative analysis

Here we provide additional quantitative results of RaySt3R evaluated against the baselines. Figure 6 shows the distributions of chamfer distance, visualized with histograms. The results suggest RaySt3R consistently produces more accurate predictions.



(a) YCB-Video

(b) HOPE

(c) HomeBrewedDB

(d) OctMAE

Figure 6: RaySt3R chamfer distance distribution compared against several baselines. The results suggest RaySt3R consistently produces a more favorable distribution, with higher density at smaller chamfer distance.

Table 4 shows the standard deviation for the baselines with public checkpoints. The results suggest RaySt3R not only produces the most competitive results on average, but also consistently does so with the smallest standard deviation. Table 5 shows the results on each scene individually. The results suggest RaySt3R outperforms the baselines in all scenes.

Table 5: RaySt3R evaluated on each real-world scene individually. The results suggest RaySt3R outperforms all baselines on all scenes.

Dataset	Scene	Unique3D [52]		TRELLIS (w/ mask) [53]		TRELLIS (w/o mask) [53]		LaRI [23]		OctMAE [18]		RaySt3R (ours)	
		CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑	CD ↓	F1 ↑
HB [19]	000001	35.53	0.24	68.57	0.18	62.15	0.21	30.70	0.31	7.59	0.77	**5.99**	**0.85**
	000002	33.39	0.28	62.81	0.18	55.64	0.20	31.27	0.31	7.32	0.79	**5.36**	**0.88**
	000003	24.39	0.34	33.12	0.35	26.45	0.31	26.59	0.34	4.74	0.88	**3.43**	**0.94**
	000004	30.69	0.30	72.98	0.24	29.18	0.31	27.95	0.34	4.99	0.87	**3.40**	**0.94**
	000005	22.06	0.37	23.24	0.41	23.73	0.39	18.66	0.43	5.65	0.84	**4.27**	**0.90**
	000006	25.22	0.30	34.58	0.31	23.25	0.37	16.50	0.51	6.19	0.82	**4.19**	**0.91**
	000007	24.72	0.33	23.01	0.43	24.87	0.35	23.55	0.36	6.79	0.81	**4.66**	**0.89**
	000008	24.35	0.33	20.59	0.46	22.67	0.40	12.61	0.60	6.32	0.81	**4.15**	**0.91**
	000009	22.36	0.36	50.75	0.30	20.26	0.39	17.16	0.50	5.32	0.86	**3.56**	**0.93**
	000010	25.22	0.31	26.99	0.37	51.21	0.34	28.26	0.32	5.19	0.86	**3.95**	**0.92**
	000011	26.58	0.32	21.30	0.42	25.45	0.35	18.11	0.47	5.04	0.87	**3.97**	**0.92**
	000012	17.96	0.40	24.56	0.40	18.55	0.46	19.75	0.43	9.75	0.71	**7.69**	**0.78**
	000013	19.16	0.39	20.47	0.44	19.58	0.43	20.30	0.42	9.63	0.72	**7.40**	**0.78**
HOPE [40]	000001	24.85	0.34	12.88	0.55	19.34	0.43	10.72	0.63	5.65	0.83	**3.88**	**0.93**
	000002	31.93	0.29	15.33	0.51	14.34	0.49	6.82	0.79	4.09	0.92	**2.74**	**0.98**
	000003	35.12	0.25	41.98	0.20	42.50	0.24	47.11	0.20	5.61	0.84	**3.37**	**0.94**
	000004	31.79	0.25	24.08	0.36	26.36	0.36	19.83	0.43	11.47	0.66	**5.54**	**0.87**
	000005	20.11	0.36	16.96	0.46	21.11	0.41	16.02	0.45	7.56	0.78	**3.89**	**0.93**
	000006	19.46	0.40	11.40	0.58	14.94	0.50	7.94	0.74	6.23	0.79	**3.15**	**0.95**
	000007	22.83	0.33	14.42	0.51	18.29	0.43	13.31	0.57	8.41	0.73	**5.39**	**0.84**
	000008	22.82	0.32	19.88	0.41	14.80	0.47	9.38	0.66	5.23	0.87	**3.64**	**0.96**
	000009	38.66	0.24	35.41	0.23	37.06	0.26	44.74	0.18	4.60	0.87	**3.18**	**0.95**
	000010	16.17	0.45	10.20	0.64	11.78	0.56	10.54	0.62	15.63	0.67	**4.48**	**0.89**
YCB-Video [54]	000048	15.17	0.49	15.96	0.54	26.39	0.38	10.90	0.65	9.07	0.74	**4.29**	**0.91**
	000049	17.50	0.45	25.13	0.42	27.45	0.34	9.56	0.71	9.15	0.74	**5.04**	**0.89**
	000050	20.22	0.43	22.15	0.42	55.73	0.28	17.95	0.46	9.07	0.72	**4.25**	**0.90**
	000051	21.93	0.37	35.79	0.28	39.82	0.26	8.71	0.69	6.66	0.77	**2.82**	**0.96**
	000052	8.92	0.71	5.79	0.86	18.34	0.50	9.12	0.71	5.61	0.83	**2.98**	**0.95**
	000053	14.87	0.48	27.61	0.34	26.73	0.32	4.95	0.88	4.29	0.89	**2.43**	**0.98**
	000054	18.07	0.41	16.30	0.47	23.96	0.40	13.32	0.53	6.96	0.78	**3.93**	**0.91**
	000055	31.03	0.27	38.87	0.26	44.97	0.23	9.02	0.70	6.64	0.77	**3.68**	**0.92**
	000056	11.95	0.59	14.29	0.56	20.55	0.43	7.55	0.79	6.27	0.80	**3.34**	**0.94**
	000057	24.99	0.34	33.36	0.32	47.29	0.21	28.94	0.31	5.49	0.84	**3.71**	**0.92**
	000058	12.21	0.55	19.72	0.43	25.45	0.32	7.79	0.77	6.49	0.79	**3.13**	**0.94**
	000059	13.94	0.53	20.37	0.42	24.17	0.38	9.08	0.70	5.60	0.84	**3.10**	**0.94**

8.2 Ablations

8.2.1 Baseline ground truth alignment

Unique3D [52], LaRI [23] and TRELLIS [53] predict shapes in a canonical space. We fit their predictions to the ground truth for evaluation. Note that we do not perform an alignment step for RaySt3R, by allowing the baselines access to the ground truth points, we give them the benefit of the doubt. All other baselines predict points directly in the coordinate frame of the input camera.

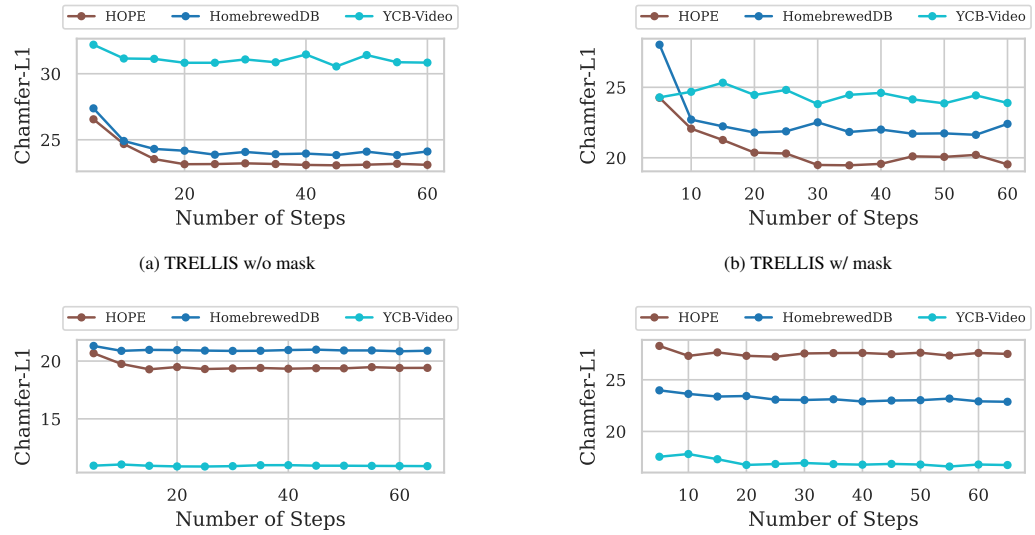
To align points from a canonical frame, we first scale the prediction such that its oriented bounding-box (OBB) extent matches that of the ground truth points. Second, we align the prediction's OBB center with that of the ground truth points. Third, we do two passes of brute force rotational orientation search in a grid search manner to minimize the chamfer distance. Finally, we apply iterative closest point (ICP) [38] for the local alignment of the predictions. In the following ablations, we randomly sample 30 scenes from each real-world dataset for computational tractability.

Rotational grid search resolution: We ablate the number of steps during the first brute-force search for the optimal rotational alignment with the ground truth point cloud. Figure 7 shows the results. The average chamfer distance declines with an increasing number of steps. A value of 20 steps marks a point beyond which further increases yield diminishing improvements in chamfer distance.

Scale grid search: We use PCA to estimate the parameters of the OBB, which may not be the optimal scale. For this ablation, we first scale the predictions using the OBB as described above. Then, we scale the object with a constant and apply the rotational alignment grid search with 20 steps. We evaluate scales in the range from 0.65 to 1.35 with a step size of 0.05, the results are shown in Figure 8.

8.2.2 Mask ablation

We have demonstrated RaySt3R outperforms the baselines in the real world, even with imperfect masks. It remains unclear how sensitive our method is to the input mask. Figure 9 shows a sensitivity analysis on false positive and false negative entries in the mask, evaluated on the OctMAE dataset. False positives are pixels in the background marked as foreground, while false negatives are pixels in



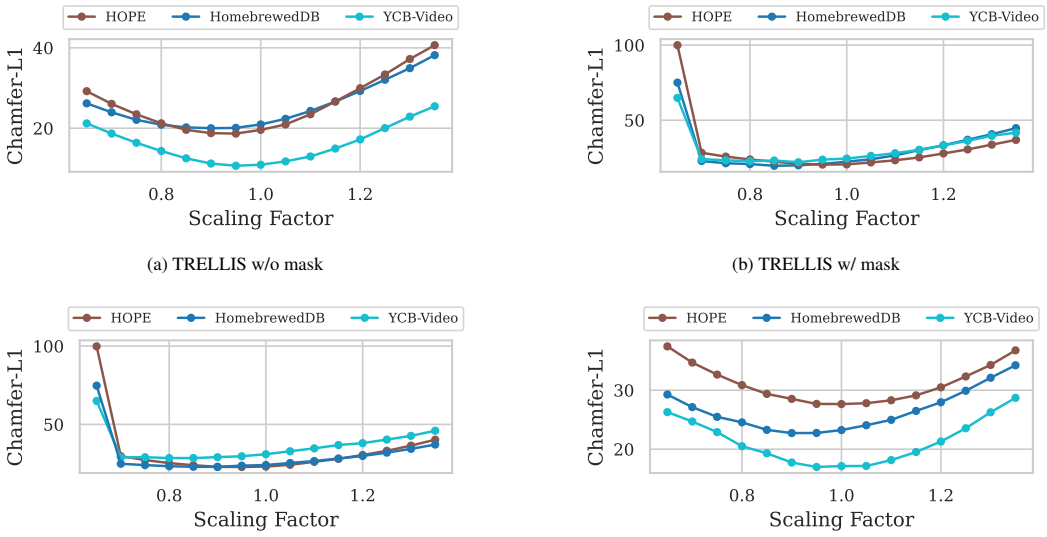
(a) TRELLIS w/o mask

(b) TRELLIS w/ mask

(c) LaRI

(d) Unique3D

Figure 7: **Rotation ablation**: Chamfer Distance averaged over the evaluation split of HomebrewedDB [19], HOPE [40], and YCB-Video [54] with different numbers of steps during the rotational alignment grid search.



(a) TRELLIS w/o mask

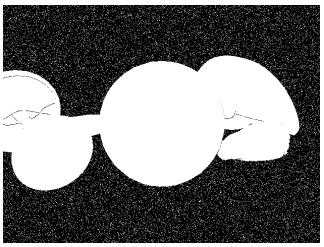
(b) TRELLIS w/ mask

(c) LaRI

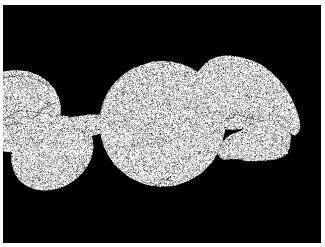
(d) Unique3D

Figure 8: **Scaling ablation**: Chamfer distance averaged over the evaluation split of HomebrewedDB [19], HOPE [40], and YCB-Video [54] for different initial scaling factors.

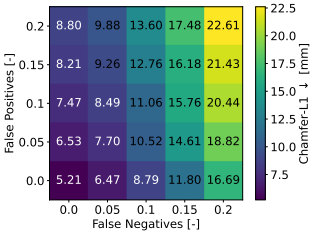
the foreground marked as background. We study noise in the range of 0 to 0.2, meaning 0% to 20% of the foreground and background pixels are incorrect. The results suggest RaySt3R is more robust to false positives, likely thanks to the confidence and mask filtering steps. The performance degrades substantially with a large number of false negatives.



(a) False Positives = 0.2



(b) False Negatives = 0.2



(c) Chamfer distance heatmap

Figure 9: RaySt3R input mask noise ablation. The results suggest RaySt3R is more robust against false positive noise, as compared to false negative noise. Colors in heatmap altered for visual clarity only.

8.3 Implementation details

8.3.1 Evaluation metrics

Consistent with prior work [18], we adopt the Chamfer Distance (CD) and F1-Score (F1) metrics for evaluation.

Chamfer Distance (CD). The chamfer distance $\text{CD}(Q, Q^{\text{gt}})$ between a predicted point cloud Q and a ground truth point cloud Q^{gt} is defined as the average of two asymmetric terms:

The accuracy (forward chamfer term) measures how well the predicted points approximate the ground truth:

$$A = \frac{1}{|Q|} \sum_{q_{\text{pd}} \in Q} \min_{q_{\text{gt}} \in Q^{\text{gt}}} \|q_{\text{pd}} - q_{\text{gt}}\| \tag{8}$$

The completeness (backward chamfer term) measures how well the ground truth is covered by the predicted points:

$$C = \frac{1}{|Q^{\text{gt}}|} \sum_{q_{\text{gt}} \in Q^{\text{gt}}} \min_{q_{\text{pd}} \in Q} \|q_{\text{gt}} - q_{\text{pd}}\| \tag{9}$$

The full Chamfer distance is:

$$\text{CD}(Q, Q^{\text{gt}}) = \frac{A + C}{2} \tag{10}$$

F1-Score. The F1-Score@ η evaluates the geometric match between predicted and ground truth point clouds under a distance threshold η, using:

Precision (prediction accuracy under threshold):

$$P = \sum_{q_{\text{pd}} \in Q} \frac{(\min_{q_{\text{gt}} \in Q^{\text{gt}}} \|q_{\text{gt}} - q_{\text{pd}}\|) < \eta}{|Q|} \tag{11}$$

Recall (ground truth completeness under threshold):

$$R = \sum_{q_{\text{gt}} \in Q^{\text{gt}}} \frac{(\min_{q_{\text{pd}} \in Q} \|q_{\text{gt}} - q_{\text{pd}}\|) < \eta}{|Q^{\text{gt}}|} \tag{12}$$

The final F1 score is:

$$\text{F1} = \frac{2PR}{P + R} \tag{13}$$

8.3.2 Visual features from DINOv2

We use DINOv2 [31] to extract visual features, and select the ViT-L with registers. As prior work has shown [10], aggregating features from several intermediate layers may lead to a perfromance improvement over only considering the last layer. We aggregate features from layers 4, 11, 17, 23 and project them to the ViT token size with a linear layer.

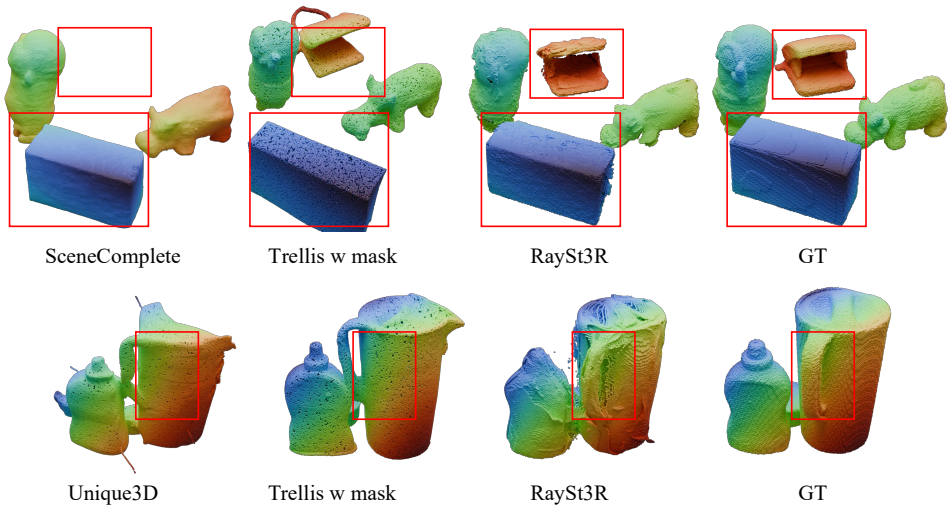


Figure 10: Highlight on relative object placement and aspect ratio. The examples suggest related works such as TRELLIS [53] and Unique3D [52] may produce misaligned geometries in their predictions. We observe minor misalignment in SceneComplete [1], and objects missing, as we describe in the paper.

8.4 Qualitative evaluation

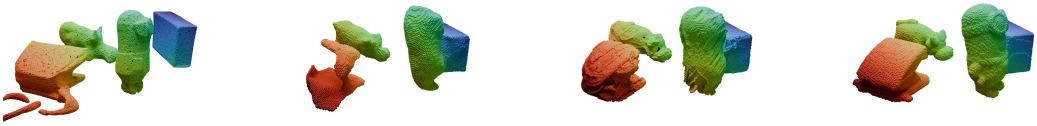
8.4.1 Object placement and aspect ratio

We describe our findings of misalignment and incorrect aspect ratios in the paper. It can be challenging to observe these deficiencies, we highlight them in Figure 10. The examples suggest related works such as TRELLIS [53] and Unique3D [52] may produce misaligned geometries in their predictions. We observe minor misalignment in SceneComplete [1], and objects missing, as we describe in the paper.

8.4.2 Baseline comparison

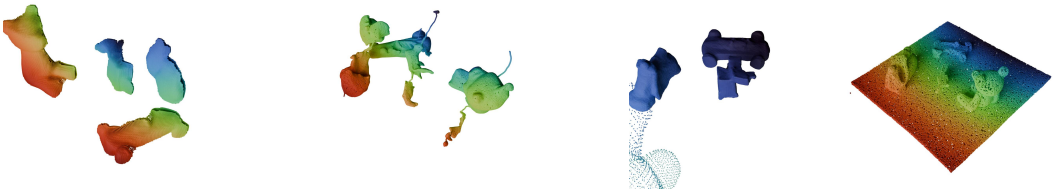


(a) LaRI (b) Unique3D. (c) SceneComplete. (d) TRELLIS w/o mask.



(e) TRELLIS w/ mask. (f) OctMAE. (g) RaySt3R (ours). (h) Ground truth.

Figure 11: Comparison of different methods on scene `000003`, frame `000061` of HomebrewedDB [19].



(a) LaRI (b) Unique3D. (c) SceneComplete. (d) TRELLIS w/o mask.

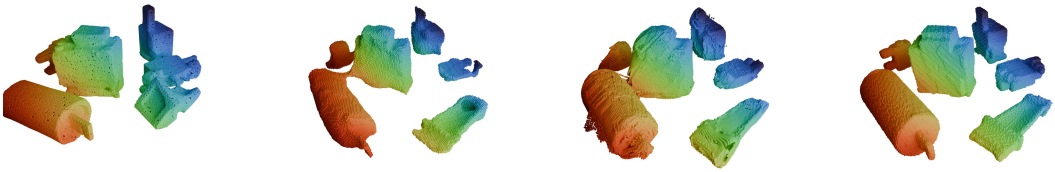


(e) TRELLIS w/ mask. (f) OctMAE. (g) RaySt3R (ours). (h) Ground truth.

Figure 12: Comparison of different methods on scene `000004`, frame `000176` of HomebrewedDB [19].



(a) LaRI (b) Unique3D. (c) SceneComplete. (d) TRELLIS w/o mask.



(e) TRELLIS w/ mask. (f) OctMAE. (g) RaySt3R (ours). (h) Ground truth.

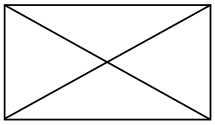
Figure 13: Comparison of different methods on scene `000009`, frame `000096` of HomebrewedDB [19].



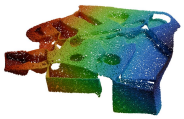
(a) LaRI



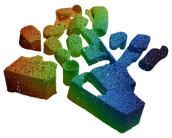
(b) Unique3D.



(c) SceneComplete (ran out of memory).



(d) TRELLIS w/o mask.



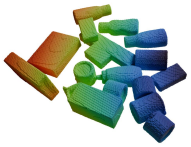
(e) TRELLIS w/ mask.



(f) OctMAE.



(g) RaySt3R (ours).



(h) Ground truth.

Figure 14: Comparison of different methods on scene 000002, frame 000002 of HOPE [40].



(a) LaRI



(b) Unique3D.



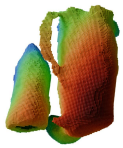
(c) SceneComplete.



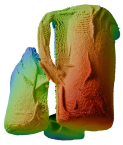
(d) TRELLIS w/o mask.



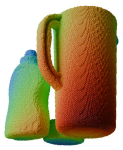
(e) TRELLIS w/ mask.



(f) OctMAE.

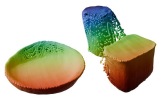


(g) RaySt3R (ours).



(h) Ground truth.

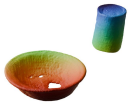
Figure 15: Comparison of different methods on scene 000052, frame 000650 of YCB-Video [54].



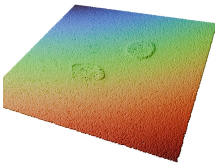
(a) LaRI



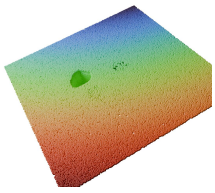
(b) Unique3D.



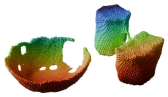
(c) SceneComplete.



(d) TRELLIS w/o mask.



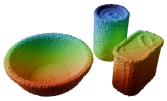
(e) TRELLIS w/ mask.



(f) OctMAE.



(g) RaySt3R (ours).



(h) Ground truth.

Figure 16: Comparison of different methods on scene 000053, frame 000075 of YCB-Video [54].

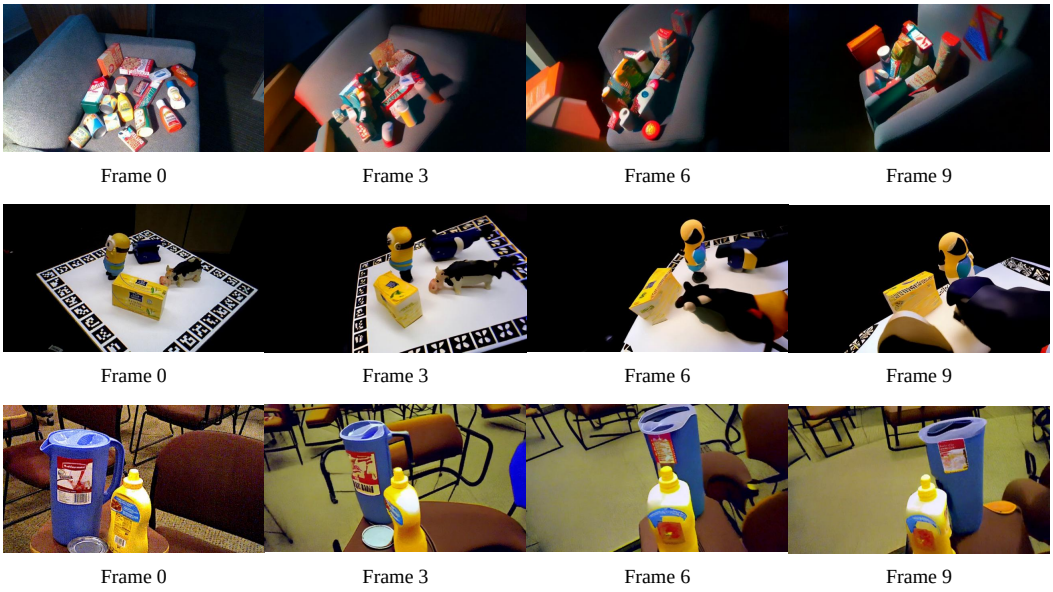


Figure 17: Qualitative results for RGB predictions from ViewCrafter [57] on a HOPE [40] (top), HomebrewedDB [19] (middle), and YCB-Video [54] (bottom) scene. The rendered trajectory is a circle around the center of the scene. The results suggest ViewCrafter [57] is unable to produce 3D-consistent images.

8.4.3 ViewCrafter qualitative results

Recent work has demonstrated video diffusion models may be used to render novel views, and subsequently synthesize geometry [57]. In this context, we observe the images synthesized by ViewCrafter [57] on scenes from the three real-world datasets used in this paper (YCB-Video [54], HOPE [40], and HomebrewedDB [19]). We define the camera trajectory as a circle on the sphere centered in the scene. Figure 17 shows frames produced by ViewCrafter [57]. The results suggest ViewCrafter [57] is unable to reconstruct geometrically consistent images.

References

[1] A. Agarwal, G. Singh, B. Sen, T. Lozano-Pérez, and L. P. Kaelbling. Scenecomplete: Open-world 3d scene completion in complex real world environments for robot manipulation, 2024.

[2] J. T. Barron and J. Malik. Intrinsic scene properties from a single rgb-d image. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 17–24, 2013.

[3] A. Boulch and R. Marlet. Poco: Point convolution for surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6302–6314, 2022.

[4] A. Bozic, P. Palafox, J. Thies, A. Dai, and M. Nießner. Transformerfusion: Monocular rgb scene reconstruction using transformers. *Advances in Neural Information Processing Systems*, 34:1403–1414, 2021.

[5] C. Choy, J. Gwak, and S. Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, 2019.

[6] M. Dalal, M. Liu, W. Talbott, C. Chen, D. Pathak, J. Zhang, and R. Salakhutdinov. Local policies enable zero-shot long-horizon manipulation. *International Conference of Robotics and Automation*, 2025.

[7] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi. Objaverse: A universe of annotated 3d objects. *CVPR*, 2022.

[8] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *ICRA*, 2022.

[9] B. Duisterhof, L. Zust, P. Weinzaepfel, V. Leroy, Y. Cabon, and J. Revaud. Mast3r-sfm: a fully-integrated solution for unconstrained structure-from-motion, 2024.

[10] A. El-Nouby, M. Klein, S. Zhai, M. A. Bautista, V. Shankar, A. Toshev, J. M. Susskind, and A. Joulin. Scalable pre-training of large autoregressive image models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[11] H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[12] R. Gao*, A. Holynski*, P. Henzler, A. Brussee, R. Martin-Brualla, P. P. Srinivasan, J. T. Barron, and B. Poole*. Cat3d: Create anything in 3d with multi-view diffusion models. *Advances in Neural Information Processing Systems*, 2024.

[13] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. *CVPR*, 2022.

[14] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[15] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

[16] J. Hou, A. Dai, and M. Nießner. Revealnet: Seeing behind objects in rgb-d scans. In *CVPR*, 2020.

[17] H. Huang, L. Fu, M. Danielczuk, C. M. Kim, Z. Tam, J. Ichnowski, A. Angelova, B. Ichter, and K. Goldberg. Mechanical search on shelves with efficient stacking and destacking of objects. In *Robotics Research*, pages 205–221, Cham, 2023. Springer Nature Switzerland.

[18] S. Iwase, K. Liu, V. Guizilini, A. Gaidon, K. Kitani, R. Ambrus, and S. Zakharov. Zero-shot multi-object scene completion, 2024.

[19] R. Kaskman, S. Zakharov, I. Shugurov, and S. Ilic. Homebreweddb: Rgb-d dataset for 6d pose estimation of 3d objects. *ICCVW*, 2019.

[20] A. Kolesnikov, A. Dosovitskiy, D. Weissenborn, G. Heigold, J. Uszkoreit, L. Beyer, M. Minderer, M. Dehghani, N. Houlsby, S. Gelly, T. Unterthiner, and X. Zhai. An image is worth 16x16 words: Transformers for image recognition at scale. 2021.

[21] L. Ladicky, O. Saurer, S. Jeong, F. Maninchedda, and M. Pollefeys. From point clouds to mesh using regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3893–3902, 2017.

[22] J. Li, K. Han, P. Wang, Y. Liu, and X. Yuan. Anisotropic convolutional networks for 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3351–3359, 2020.

[23] R. Li, B. Zhang, Z. Li, F. Tombari, and P. Wonka. Lari: Layered ray intersections for single-view 3d geometric reasoning. In *arXiv preprint arXiv:2504.18424*, 2025.

[24] Y. Li, Z. Yu, C. Choy, C. Xiao, J. M. Alvarez, S. Fidler, C. Feng, and A. Anandkumar. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In *CVPR*, 2023.

[25] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023.

[26] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[27] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg. Learning ambidextrous robot grasping policies. *Science Robotics*, 4(26):eaau4984, 2019.

[28] L. Melas-Kyriazi, C. Rupprecht, and A. Vedaldi. Pc2: Projection-conditioned point cloud diffusion for single-image 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12923–12932, 2023.

[29] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019.

[30] I. Mishani, H. Feddock, and M. Likhachev. Constant-time motion planning with anytime refinement for manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, page 10337–10343. IEEE, May 2024.

[31] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[32] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

[33] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer, 2020.

[34] R. Ranftl, A. Bochkovskiy, and V. Koltun. Vision transformers for dense prediction. In *ICCV*, 2021.

[35] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

[36] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[37] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

[38] A. Somani, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 698–700, 1987.

[39] Z. Tang, B. Sundaralingam, J. Tremblay, B. Wen, Y. Yuan, S. Tyree, C. Loop, A. Schwing, and S. Birchfield. Rgb-only reconstruction of tabletop scenes for collision-free manipulator control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1778–1785. IEEE, 2023.

[40] S. Tyree, J. Tremblay, T. To, J. Cheng, T. Mosier, J. Smith, and S. Birchfield. 6-dof pose estimation of household objects for robotic manipulation: An accessible dataset and benchmark. In *International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[41] A. Vahdat, F. Williams, Z. Gojcic, O. Litany, S. Fidler, K. Kreis, et al. Lion: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems*, 35:10021–10039, 2022.

[42] V. Voleti, C.-H. Yao, M. Boss, A. Letts, D. Pankratz, D. Tochilkin, C. Laforte, R. Rombach, and V. Jampani. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. In *European Conference on Computer Vision*, pages 439–457. Springer, 2024.

[43] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[44] P.-S. Wang, Y. Liu, and X. Tong. Deep octree-based cnns with output-guided skip connections for 3d shape and scene completion. In *CVPRW*, 2020.

[45] Q. Wang, Y. Zhang, A. Holynski, A. A. Efros, and A. Kanazawa. Continuous 3d perception model with persistent state. *arXiv preprint arXiv:2501.12387*, 2025.

[46] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024.

[47] E. Weber, N. Müller, Y. Kant, V. Agrawal, M. Zollhöfer, A. Kanazawa, and C. Richardt. Fillerbuster: Multi-view scene completion for casual captures, 2025. arXiv:2502.05175.

[48] B. Wen, W. Lian, K. Bekris, and S. Schaal. Catgrasp: Learning category-level task-relevant grasping in clutter from simulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6401–6408. IEEE, 2022.

[49] B. Wen, C. Mitash, B. Ren, and K. E. Bekris. se (3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10367–10373. IEEE, 2020.

[50] B. Wen, W. Yang, J. Kautz, and S. Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *CVPR*, 2024.

[51] C.-Y. Wu, J. Johnson, J. Malik, C. Feichtenhofer, and G. Gkioxari. Multiview compressive coding for 3D reconstruction. In *CVPR*, 2023.

[52] K. Wu, F. Liu, Z. Cai, R. Yan, H. Wang, Y. Hu, Y. Duan, and K. Ma. Unique3d: High-quality and efficient 3d mesh generation from a single image, 2024.

[53] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, and J. Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.

[54] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. 2018.

[55] X. Yan, L. Lin, N. J. Mitra, D. Lischinski, D. Cohen-Or, and H. Huang. Shapeformer: Transformer-based shape completion via sparse representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[56] K. Yao, L. Zhang, X. Yan, Y. Zeng, Q. Zhang, L. Xu, W. Yang, J. Gu, and J. Yu. CAST: Component-aligned 3d scene reconstruction from an RGB image. In *arXiv:2502.12894*, 2025.

[57] W. Yu, J. Xing, L. Yuan, W. Hu, X. Li, Z. Huang, X. Gao, T.-T. Wong, Y. Shan, and Y. Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024.