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

Bardienus P. Duisterhof
Carnegie Mellon University

Jan Oberst
Carnegie Mellon University

Bowen Wen
NVIDIA

Stan Birchfield
NVIDIA

Deva Ramanan
Carnegie Mellon University

Jeffrey Ichnowski
Carnegie Mellon University

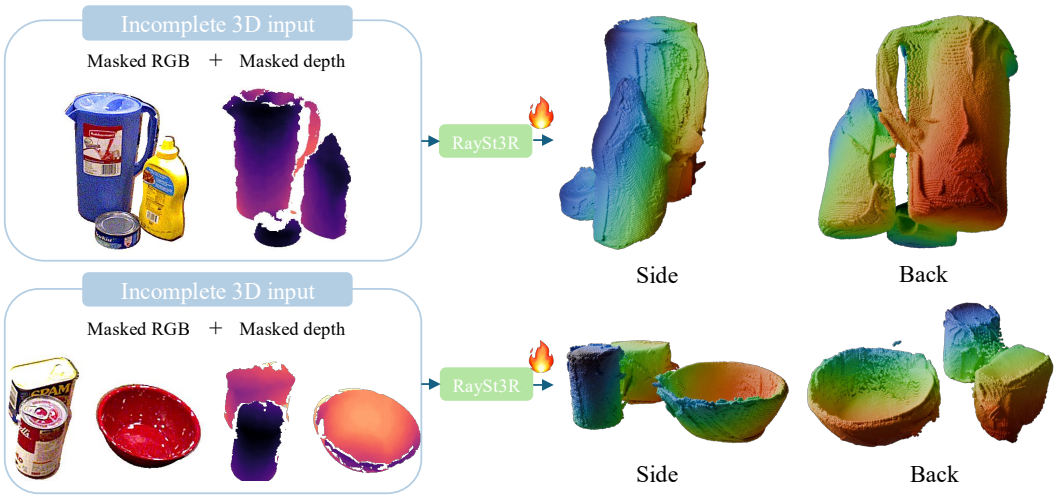

Figure 1: RaySt3R is a method for zero-shot 3D shape completion from a single foreground-masked RGB-D image. It predicts depth maps, object masks, and per-pixel confidence scores for novel viewpoints, and fuses them to reconstruct a complete 3D shape. RaySt3R is able to recover the geometry of full objects 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`

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

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 [29, 50], obstacle avoidance [32, 41], 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 [59, 49] 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 [48, 9]. We similarly use a vision transformer (ViT) [22] architecture defined over visual DINOv2 [33] 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 251 k unique scenes and 11 million novel depth maps (training pairs). 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 11 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, 30, 43], 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 [34, 23] or voxel occupancy grids [3, 4, 35, 16, 31, 53, 24]. 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

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.

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, 58]. 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 [55] 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, 39, 38] 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 [27, 59, 44, 54, 12]. ViewCrafter [59] leverages these advances by iteratively completing a scene point cloud using a point-conditioned video diffusion model. Similar to RaySt3R, LVSM [20] synthesizes novel views by querying a transformer with Plücker Rays and by conditioning on input views. RaySt3R predicts depths and object masks instead of images, and does not require full Plücker rays for querying novel views. RayZer [19] is a self-supervised large view synthesis model, using its self-predicted camera poses to eliminate the need for any ground-truth camera annotations.

In the object-centric domain, Li et al. [25] 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 [54] 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 [48] and successors [47, 49, 45]. 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

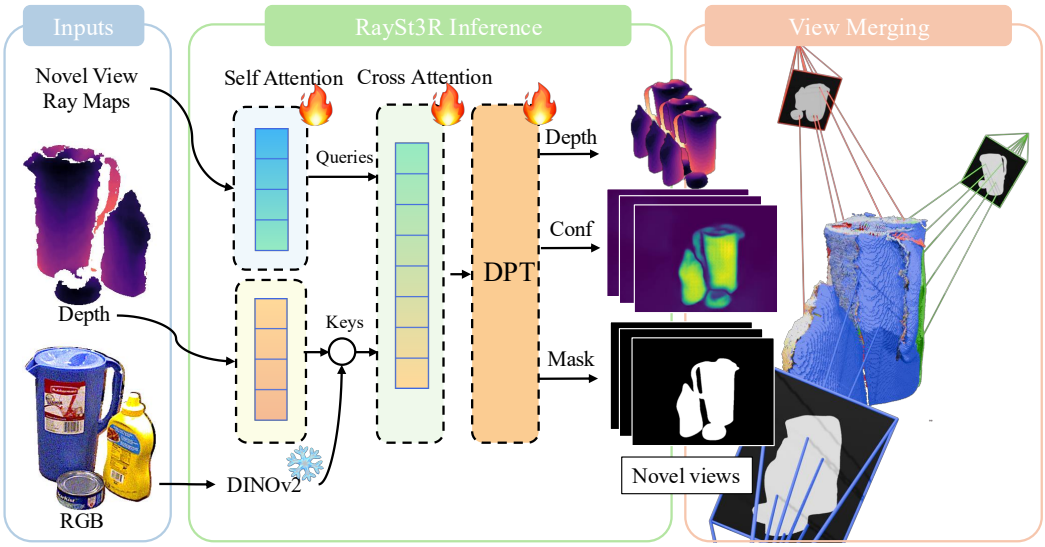

Figure 2: The architecture of RaySt3R. RaySt3R takes a single RGB-D image and foreground mask as input, and predicts depth maps, object masks, and per-pixel confidence scores for novel views. First, we apply the foreground mask to the RGB and point map input. Next, we use self-attention layers for the point map and ray map inputs, and feed the RGB image into the frozen DINOv2 [33] encoder. We feed all features into cross-attention layers followed by two separate DPT heads [36] for depth and mask predictions. Finally, we provide confidence- and occlusion-aware multi-view merging formulation.

input image intrinsics K^{input}, thus $X^{\text{input}}_{i,j} = (K^{\text{input}})^{-1}[iD^{\text{input}}_{i,j}, jD^{\text{input}}_{i,j}, D^{\text{input}}_{i,j}]^\intercal$. We convert the query view into a ray map $R \in \mathbb{R}^{H\times W\times 2}$, with $R_{i,j} = [(i-c_x)/f_x, (j-c_y)/f_y]^\intercal$, where c_x, c_y is the image center and f_x, f_y are the focal lengths of the novel target view.

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, $X^{\text{context}} = P^{\text{query}}_{\text{input}} h(X^{\text{input}})$, where $P^{\text{query}}_{\text{input}} \in \mathbb{R}^{3\times 4}$ 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 [33] 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 [36] 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 [48], we define the confidence-aware depth loss:

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

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, $d^{\text{gt}}_{i,j}$ is the ground-truth depth, and $M^{\text{gt}}_{i,j}$ 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 $p_{n,i,j} = K_{\text{input}} P_n^{\text{input}} h(q_{n,i,j})$, where P_n^{input} 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 [48, 45], 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 [52] 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 251 k unique scenes, 12 k objects, and 11 M novel depth maps rendered for supervision.

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 [56] (900 frames), HOPE [42] (50 frames), and HomebrewedDB [21] (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 [52, 51, 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 [28]. 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 [33].

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 [26], ShapeFormer [57], MCC [53], ConvONet [35], POCO [3], AICNet [24], Minkowski [5], and OCNN [46].

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 [37] 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 [54] and TRELLIS [55], which are recent image to 3D models. Finally, we compare against 'Layered Ray Intersections' (LaRI) [25], 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 [40]. Note that we do not perform such a registration for RaySt3R, but provide this to baselines to give them the benefit of the doubt. LaRI [25], Unique3D [54] and TRELLIS [55] were not trained on cluttered scenes.

While LaRi [25] and Unique3D [54] do not require foreground masks for reconstructing objects, we observe that TRELLIS [55] 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 [59] 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 [25], Unique3D [54], and TRELLIS [55] show better performance than SceneComplete [1], but also do not produce competitive results. Feeding masked RGB images to TRELLIS [55] outperforms raw image inputs on all datasets except HomebrewedDB [21]. 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 [54] (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 [56], HOPE [42], and HomebrewedDB [21]. 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 [25], Unique3D [54], and TRELLIS [55] with the ground truth mesh, we apply brute force search followed by ICP [40]. We evaluate TRELLIS [55] with masked and unmasked RGB inputs. SceneComplete [1] runs out of VRAM on HOPE [42]. The results suggest RaySt3R outperforms all baselines.

	Synthetic		Real					
	OctMAE [18]		YCB-Video [56]		HB [21]		HOPE [42]	
Method	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑
VoxFormer [26]	44.54	0.382	30.32	0.438	34.84	0.366	47.75	0.323
ShapeFormer [57]	39.50	0.401	38.21	0.385	40.93	0.328	39.54	0.306
MCC [53]	43.37	0.459	35.85	0.289	19.59	0.371	17.53	0.357
ConvONet [35]	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 [24]	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 [46]	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 [25]	39.22	0.283	11.41	0.658	22.23	0.414	18.64	0.528
Unique3D [54]	44.62	0.244	17.56	0.468	25.41	0.329	26.37	0.322
TRELLIS (w/ mask) [55]	65.43	0.224	22.44	0.454	36.12	0.364	19.46	0.470
TRELLIS (w/o mask) [55]	69.45	0.221	31.45	0.345	29.98	0.360	20.87	0.438
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 [55], Unique3D [54] and LaRI [25] 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.

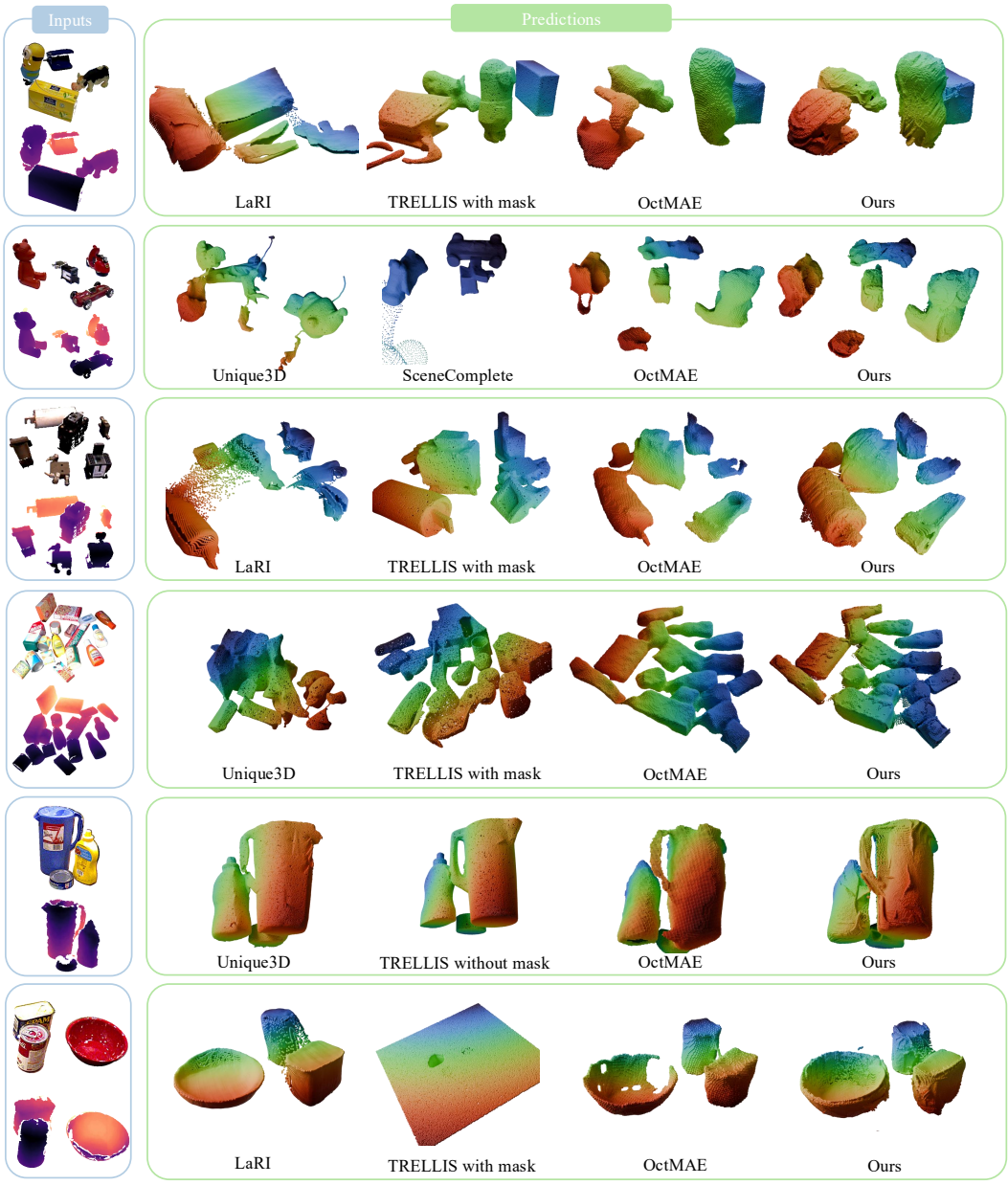

Figure 3: Qualitative results of RaySt3R in real-world multi-object scenes. For each scene, we select a subset of methods to preserve visual clarity, but we share all method predictions in the supplemental material. The results suggest RaySt3R produces the most geometrically accurate shapes compared 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 [55], Unique3D [54], and LaRI [25] 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.

Interestingly, TRELLIS [55] 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.

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

Table 2: Ablation study on RaySt3R training on the YCB-Video dataset [56]. The results suggest data scale, data diversity, DINOv2 [33] 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 [33]	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

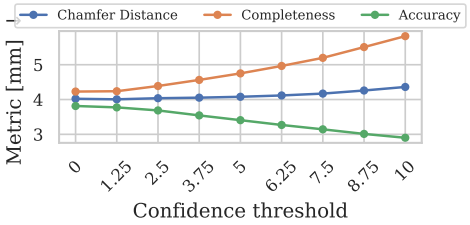

Figure 4: Confidence threshold vs error metrics averaged over all real-world datasets. The results suggest increasing the confidence threshold improves accuracy and degrades completeness.

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.

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 [33] inputs causes an additional decline, underscoring the benefit of pretrained features. We also train a model exclusively on the OctMAE split of our dataset, and find it achieves a CD of 5.87, and an F1 of 0.893 on the OctMAE test set. This result suggests RaySt3R outperforms the baselines even when trained on the same dataset.

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.

While the results suggest our view-based approach generates high-quality reconstructions, it also suffers from the common edge bleeding issue, adding noise to predictions. Our formulation also requires inferring high-quality novel view poses, which is a non-trivial task beyond objects on table tops. An advantage of RaySt3R's view-based approach is that it enables training on real-world data without the need for ground truth meshes. Training on real-world data may help generalize to more objects and adapt to real-world noise. Future work may also explore scaling up compute by exploring other architectures like diffusion transformers.

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, Sergey Zakharov, Nikhil Varma Keetha, Jeff Tan and all members of the Momentum Robotics lab at CMU for providing useful feedback.

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] H. Jiang, H. Tan, P. Wang, H. Jin, Y. Zhao, S. Bi, K. Zhang, F. Luan, K. Sunkavalli, Q. Huang, and G. Pavlakos. Rayzer: A self-supervised large view synthesis model. 2025.

[20] H. Jin, H. Jiang, H. Tan, K. Zhang, S. Bi, T. Zhang, F. Luan, N. Snavely, and Z. Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

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

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

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

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

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

[49] 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.

[50] 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.

[51] 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.

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

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

[54] 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.

[55] 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.

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

[57] 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.

[58] 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.

[59] 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.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: Yes, we have precisely highlighted the contribution of our model without overclaiming.

   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: Yes, we discuss the limitation of our method in the conclusion section.

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: We do not provide any theorems or proofs in this paper.

   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 will provide all necessary hyper parameters and details to reproduce the results. We also plan to release the code, model checkpoints, and dataset.

   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: [Yes]

   Justification: We plan to release the code, model checkpoints and dataset of this paper, with useful instructions to help adoption.

   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, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, we believe we have provided all necessary details to reproduce our results.

   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: [Yes]

   Justification: We present metrics suitable for the experiments we execute. For training the model, rerunning the training several times would be an impractical cost in terms of compute and energy expenditure. For evaluation, we partially adopt results from prior papers, and unfortunately do not have access to the raw metrics or checkpoints necessary to compute error bars. For the methods we compute metrics for, we discuss the methods with the largest variance in our writing and provide a full set of error bars in the supplemental material.

   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: Yes, we provide the compute required for training our model.

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: Yes, our work, to the best of our knowledge, conforms with the NeurIPS 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 potential impact on robotics and extended reality. There is an ongoing debate about the positive and negative societal impact of these fields.

    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: [NA]

Justification: We do not foresee a need for safeguards in our system.

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 have accredited the original owners of the assets used in this work, specifically Objaverse, GSO, OctMAE, and FoundationPose.

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` 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 do not release any 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: No crowdsourcing or human subjects involved.

    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: No crowdsourcing or human subjects involved.

    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.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: We do not involve LLMs in our method.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.