SceneComplete: Open-World 3D Scene Completion in Cluttered Real World Environments for Robot Manipulation

Aditya Agarwal[1], Gaurav Singh[2], Bipasha Sen[1], Tomás Lozano-Pérez[1], Leslie Pack Kaelbling[1]

Fig. 1: (a) takes as input a *single* RGB-D image of a given scene, visualized here as a point cloud; (b) produces high-quality fully completed, accurately segmented object meshes in scenes with substantial occlusion and novel objects; and (c) enables downstream dexterous manipulation that requires accurate complete shape information.

***Abstract*—Careful robot manipulation in every-day cluttered environments requires an accurate understanding of the 3D scene, in order to grasp and place objects stably and reliably and to avoid colliding with other objects. In general, we must construct such a 3D interpretation of a complex scene based on limited input, such as a single RGB-D image. We describe SceneComplete, a system for constructing a complete, segmented, 3D model of a scene from a single view. SceneComplete is a novel pipeline for composing general-purpose pretrained perception modules (vision-language, segmentation, image-inpainting, image-to-3D, visual-descriptors and pose-estimation) to obtain highly accurate results. We demonstrate its accuracy and effectiveness with respect to ground-truth models in a large benchmark dataset and show that its accurate whole-object reconstruction enables robust grasp proposal generation, including for a dexterous hand. We release the code and additional results on our website.**

***Index Terms*—Perception for Grasping and Manipulation, RGB-D Perception, Manipulation Planning.**

I. Introduction

As manipulation robots move from constrained environments such as factories and workshops to open-world environments such as homes and hospitals, they must be able to construct representations of their environment that enable robust, careful manipulation. Such representations need to individuate objects and characterize their shapes, so that the robot can reliably select stable grasps and placements for individual objects and manipulate them without unwanted collisions. These representations must generally be constructed from limited input, such as a single RGB-D image. This problem is fundamentally ill-posed, but we are now in a position to address it using strong priors that have been learned by vision foundation models.

In this paper, we propose a solution to this open-world scene completion problem in the form of a perception pipeline, SceneComplete. It combines multiple large pre-trained vision models into a system that takes a single RGB-D image as input and predicts as output a completed scene, consisting of a set of meshes for all the visible objects, including those that are partially occluded. Crucially, it makes no assumptions about the categories of the objects, their arrangement, or the camera viewpoint. It is constructed from multiple highly capable pre-trained perception components: a vision-language model (VLM) for identifying and generating short descriptions of the objects in a scene, a text-grounded image-segmentation model for localizing objects in the image, a 2D image-inpainting model for predicting the appearance of occluded parts of objects, an image-to-3D model for generating complete object meshes, and visual descriptor and pose-estimation modules to aid in composing individual predicted meshes into a final scene. None of these components can individually solve the problem, but in combination they provide robust object-centric interpretation of complex images, producing a segmented set of object meshes that are suitable for robot planning and manipulation.

To demonstrate the effectiveness of SceneComplete, we conduct extensive quantitative and qualitative evaluations on real-world tabletop scenes. Our quantitative evaluations are on the Graspnet-1B [1] and YCB-Video [2] datasets, which consist of cluttered tabletop scenes. In these scenarios, accurately predicting the full scene——including partially occluded objects——is crucial for stable grasping, collision-free motion with an object in the hand, and reliable placing.

We further illustrate the utility of our shape-reconstruction methods by using them as input to parallel-jaw [3], [4] and

[1]Computer Science and Artificial Intelligence Laboratory, MIT. {adityaag, bise, tlp, lpk}@mit.edu
[2]Department of Computer Science, Brown University. gaurav@brown.edu

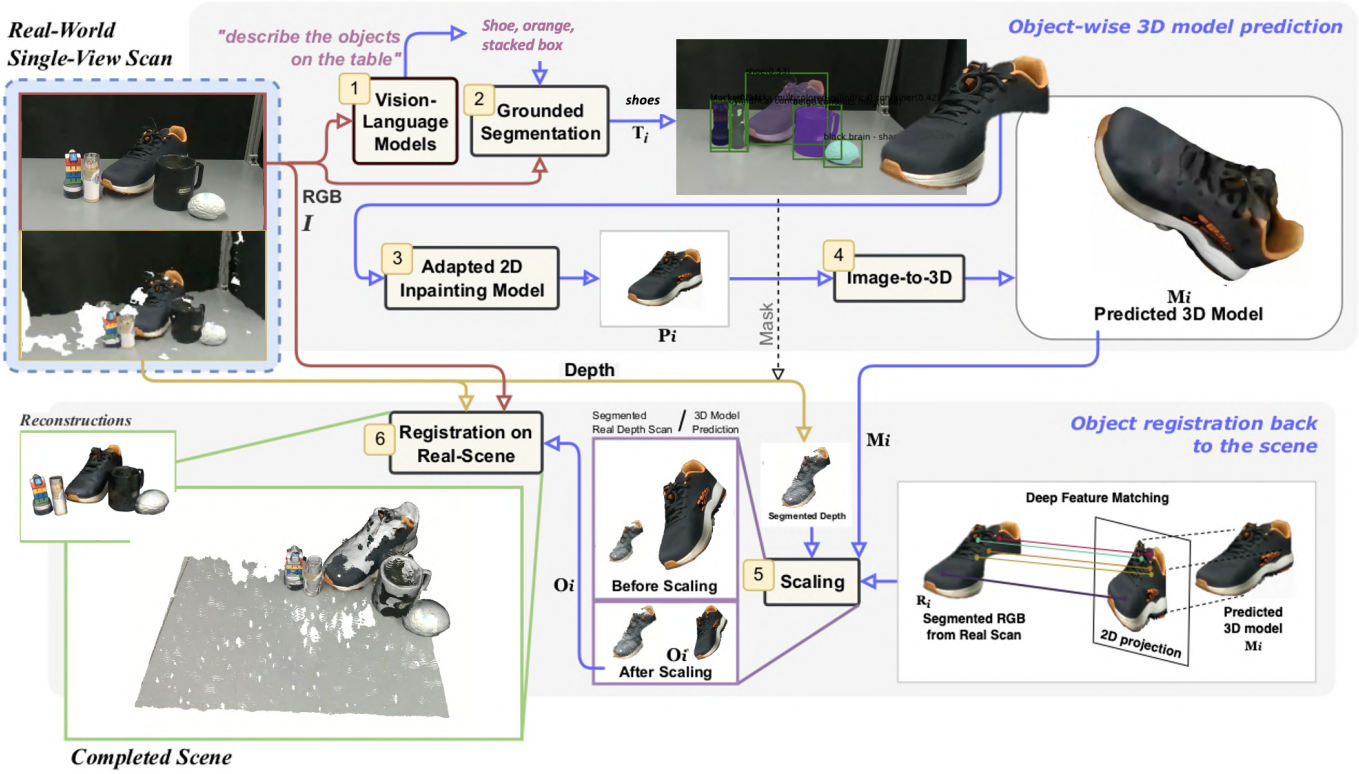

Fig. 2: **Overview of the SceneComplete pipeline.** Starting from a single RGB-D input, the system produces a set of object meshes registered with the input 3D scan, yielding a complete 3D scene reconstruction. The pipeline consists of six key phases: (1) An RGB image is fed into a VLM to enumerate and describe objects, (2) object descriptions and the RGB image are processed by a grounded segmentation model to generate object masks, (3) occluded regions are completed via image inpainting model adapted to output single fully observable objects on a white background, (4) the inpainted 2D images are passed into an image-to-3D model to produce object meshes, (5) object meshes are scaled according to the segmented partial point cloud, and (6) mesh poses are adjusted within the 3D coordinate frame of the original scan using 6DOF pose estimation. Each step leverages pre-trained open world large vision models, enabling scalability and benefiting from future model improvements.

dexterous grasping [5] methods, the latter being especially sensitive to the detailed shape *of the entire object*.

II. Related Work

Feed-forward Scene Reconstruction: Feed-forward multi-object scene completion methods such as [6]–[10] learn end-to-end mappings from single-view RGB-D input to completed object meshes or occupancy grids. These methods are attractive because they are computationally fast at inference time; however, suffer from key limitations: ShAPO [7], FSD [6], and CRISP [10] are closed-set methods, trained extensively on fixed benchmark datasets with predefined object categories, and fail to generalize to novel objects. OctMAE [8] addresses open-set scenes by performing scene-level reconstruction. However, its predictions are surface-level and do not individuate objects, limiting their utility for robotic manipulation. ZeroGrasp [9] performs scene reconstruction with individuated objects for grasp generation, but the reconstruction quality—crucial for collision avoidance—is subpar since the focus is primarily on grasping. In our results, we compare and outperform OctMAE and ZeroGrasp in reconstructing scenes and generating grasps.

Compositional Scene Reconstruction: This line of work composes multiple open-set models with zero-shot generalization into a pipeline for scene reconstruction. While such methods are typically slower than feedforward methods, they can directly handle unseen objects with little or no retraining [11]–[13]. CAST [11] and Gen3DSR [12] both reconstruct full scenes from a single RGB image by first predicting depth maps from monocular depth estimation models. CAST then generates meshes for individual objects and applies a physics-aware correction step to enforce physically consistent placements, while Gen3DSR integrates components such as DreamGaussian to produce full 3D meshes. However, because both methods rely on predicted depth, their pipelines are tightly coupled to synthetic outputs and fail to adapt to raw RGB-D input, where sensor noise is unavoidable. These approaches are primarily designed for asset generation, and their inability to incorporate ground-truth depth limits their applicability in robotics. By contrast, SceneComplete leverages observed sensor depth (often noisy), enabling a more flexible pipeline tailored for manipulation tasks.

A different line of work, Open6DOR [13], introduces a benchmark for language-driven 6-DoF object rearrangement, along with a baseline pipeline for grasp generation. Their approach composes a set of modules similar to those in SceneComplete, but focuses exclusively on predicting object poses. As a result, the pipeline does not perform full scene reconstruction, omits critical steps such as registration, and

restricts predictions to objects that are almost fully visible.

III. Method

Figure 2 illustrates the overall design of SceneComplete. It takes a single RGB-D image as input and produces a set of object meshes that are registered with the input 3D scan. The objective is to provide an accurate 3D reconstruction of the scene, in terms of segmentation into rigid components, and the shape of each component, expressed as a mesh. Importantly, each step in the pipeline makes use of existing pre-trained open-world visual-processing models, with almost no additional training (we do a small amount of low-rank adaptation of the inpainting model). This means that, as improved models become available for each of these tasks, as they inevitably will, we will be able to immediately profit from these improvements. We describe each process in detail in the following subsections.

A. Prompting and segmentation

We begin by using a vision-language model to determine the number and basic description of objects in the scene. In our implementation, we pass the RGB image I into ChatGPT-4o[1] with the prompt *"describe the objects in the image with their generic name and color as prompts in a list."* It produces a text response as a list $t_1, \ldots, t_n$ of text descriptions of objects. E.g., in the example shown in Fig. 2, it returns *"Blue Bowl, Tape, Banana"* and so on.

Next, we obtain an image mask for each object. For each text description t_i, we prompt a grounded segmentation model (in our implementation, GroundedSam2 [14]) using t_i on image I and obtain a candidate set of pairs of masks and confidence values. For example, the prompt "a pear" might return multiple useful masks in a scene with two pears; in other cases, some of the masks will be unhelpful (and hopefully low-confidence). We use the confidence values to greedily select a set of non-overlapping masks, and associate each mask with the text prompt that generated it, producing the set $(R_1, T_1), \ldots, (R_N, T_N)$, R and T denoting the masks and prompts, respectively, for N objects in the scene.

B. Image Inpainting

Fig. 3: In the image inpainting module, occluded objects (blue borders) are transformed into single fully observable objects.

As illustrated in the Fig. 3, the objects represented by the masks in $(R_1, T_1), \ldots, (R_N, T_N)$ could be partially occluded by other objects in the scene. In this step, we use an image inpainting algorithm to fill in the occluded parts of each object's image. This is important for the next step of predicting a 3D mesh, because the image-to-3D models only performs reliably with a complete view of the object. Fig. 4 (a) illustrates the poor results of attempting reconstruction from an incomplete image.

[1]https://openai.com/index/hello-gpt-4o/

In our implementation we begin with BrushNet [15], which takes an image with explicitly masked out regions and a text prompt, and produces a completed image, with the masked portions filled in. So for each (R_i, T_i) pair, we begin by constructing an image I_i with just the segmented object region R_i on a white background. Then we construct an inpainting mask consisting of the union of the regions R_j for $j \neq i$. Finally we query Brushnet with I_i, $\cup_{j \neq i} R_j$, and T_i and obtain an inpainted image P_i of the completed object.

However, we observed that, out of the box, BrushNet occasionally synthesized additional objects in the occluded areas, as shown in Fig. 4 (b), possibly due to the "artistic" data set on which it was trained. To improve this behavior, we adapt BrushNet to full, single objects. It is important to note that, even though we want to adapt BrushNet to **single fully observed** objects, we want to retain the open world capabilities of the model. To achieve this, we use Low-Rank Adaptation (LoRA) [16] on its learnable layers with the PEFT method, targeting domain-specific improvements on the tabletop YCB dataset [17]. LoRA is an effective method for adapting pretrained models to domain-specific outputs while retaining the inherent generalization of the models. To perform this adaptation, we project the different 3D YCB object meshes from arbitrary poses on a white background and add random brush masks for inpainting as suggested by BrushNet. The adapted model allows us to reliably inpaint individual occluded objects, including those from categories not in the adaptation dataset.

C. Image-to-3D models for object reconstruction

At this point, we have, for each object, a fully observed object image P_i on a white background. The next step is to generate a 3D mesh model for the object. Although methods exist for operating directly on the point-cloud generated from the depth channel of the RGB-D image, the depth information is often very low-resolution and noisy, so these depth-only models tend to be highly tuned to specific categories [7], [18]–[20] and viewpoints, or operate over idealized sensory input. In recent years, substantial improvements have been made in RGB-only methods, which take advantage of the high quality of the RGB signal and the enormous amounts of available training data [21]–[23].

For RGB image as inputs, one option is to use methods that optimize a 3D mesh through differentiable rendering [24]–[26]. Although these models have impressive open world results, they have substantial computational cost. On the other hand, feed-forward methods such as InstantMesh [23] that directly map a single-view RGB image into a 3D mesh are highly accurate, open world, and computationally inexpensive at the time of inference. For these reasons, we use InstantMesh, providing each image P_i as input and obtaining a complete textured 3D mesh M_i as output. The resulting meshes are produced in an arbitrary orientation, at an arbitrary scale, so more work remains to be done, as explained in the subsequent subsections.

D. Mesh Scaling using Dense Correspondence Matching

The next step is to rescale the meshes M_i, using a point-cloud constructed from the region R_i to determine the scale

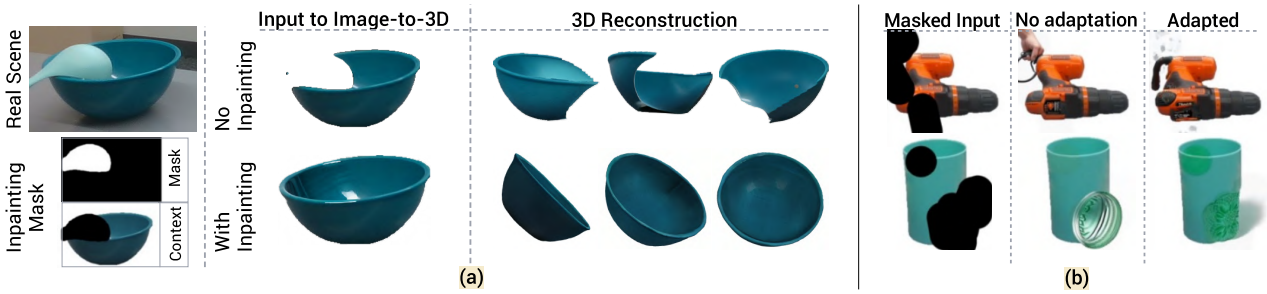

Fig. 4: (a) The impact of inpainting on image-to-3D reconstruction. Without inpainting (top), the image-to-3D model generates incomplete meshes. Inpainting (bottom) fills in occluded parts, producing accurate 3D reconstructions. (b) Comparison of inpainting models. Unadapted BrushNet (middle) introduces artifacts, while the adapted version (right) inpaints occluded parts correctly producing a fully observed object.

factor as follows: (i) Using the default viewpoint v from InstantMesh, generate an image V_i as a 2D projection of mesh M_i. (ii) Following [27], find dense visual descriptors of the segmented original image, R_i and the projected image, V_i, using a pre-trained vision transformer. In our observation, v is usually close to the viewpoint from which R_i was rendered. This enables us to obtain matching visual descriptors across R_i and V_i. (iii) Generate dense pixel-wise correspondences between the descriptor images, so that we have a set of pairs (R_i^j, V_i^j) where R_i^j is a pixel in the original image of the object and V_i^j is a pixel in the synthesized view. (iv) Map these correspondences into 3D, to obtain a pair $(R_i^{j'}, V_i^{j'})$ of points in 3D. (Note that these point sets may not yet be aligned in 3D—we address that problem in the next step.) (v) Center each resulting 3D cloud, compute the average Euclidean distance from the points to the centroid, and compute the ratio of these values as the scale factor. (vi) Apply the scale factor to the mesh M_i to put it in the same scale as the original point cloud to obtain O_i as shown in Fig. 2-(5).

E. 6D Object Pose Estimation and Registration

Now, we have, for each object, an appropriately scaled mesh O_i, and we need to reconstruct the entire scene. To do this, we need to find a 6DOF transform for each mesh that causes it to register well with the observed point cloud. For this task, we use FoundationPose [28], a robust object-pose estimation method designed to operate without being limited to specific object categories. We use its model-based mode, which takes as input a partial point-cloud derived from region R_i of the input RGB-D image and an appropriately scaled textured object mesh O_i, and returns a 6DOF transform τ_i mapping O_i into the coordinate frame of the point cloud.

As a result of this process, we have a set of pairs (O_i, τ_i), where each O_i is a textured complete mesh for object i and τ_i is a pose for that mesh in the camera coordinate frame. This scene reconstruction provides a highly general representation for a wide variety of downstream object-manipulation tasks. Accurate reconstructions of unobserved parts of objects enables a wide variety of manipulation operations, including many types of robust grasping, moving safely in cluttered but unobserved parts of the scene, moving safely when holding a grasped object, etc.

IV. Experiments

We evaluate SceneComplete through three main experimental regimes:

- **Scene Reconstruction and Grasping**: We compare SceneComplete against an existing single-view scene reconstruction method on a large-scale dataset of tabletop scenes, GraspNet-1B [1], captured using a RealSense D435 camera. We also evaluate how the reconstructed scenes contribute in generating collision-free grasps.
- **Object-grasping and Dexterous Manipulation**: We evaluate the effectiveness of SceneComplete in generating grasps for successfully picking up objects inside a simulation environment using a parallel-jaw gripper. We also demonstrate that the reconstructed object models are of sufficient fidelity to enable dexterous grasp proposals for a multi-fingered hand, which depends on having a good estimate of the entire object shape.
- **Real-world Evaluations**: To assess the real-world applicability of SceneComplete, we conduct pick-and-place experiments on a physical robot on a smaller-scale dataset of tabletop scenes collected in our lab (also using a RealSense D435 camera), that include everyday objects which are less likely to have appeared in the training distribution of any of the models used. We evaluate across 15 scenes, each with 4 to 6 objects.

We observe that the runtime of the current implementation of SceneComplete is about 20s per object on a single NVIDIA RTX 4090 GPU; we expect this to improve with advances in models and GPU architectures.

A. Scene Reconstruction and Grasping

We first evaluate SceneComplete on the task of reconstructing tabletop scenes from just a single RGB-D image.

Dataset: We use the GraspNet-1B dataset [1] for our evaluation. This dataset is particularly suitable for our setting, due to its large collection of 190 cluttered tabletop scenes, featuring 88 unique objects in various configurations. It includes ground-truth 3D object models and poses for each scene, as well as real RGB-D images.

Baseline: Our primary baselines for comparison are OctMAE [8] and ZeroGrasp [9]. OctMAE performs reconstruction by combining octree-based representation with a 3D Masked AutoEncoder (MAE). ZeroGrasp performs simultaneous 3D reconstruction and 6D grasp pose prediction using an octree-based CVAE. Both methods expect a single-view RGB-D image along with a corresponding foreground mask as input, and output the reconstructed scene. For our experiments, we

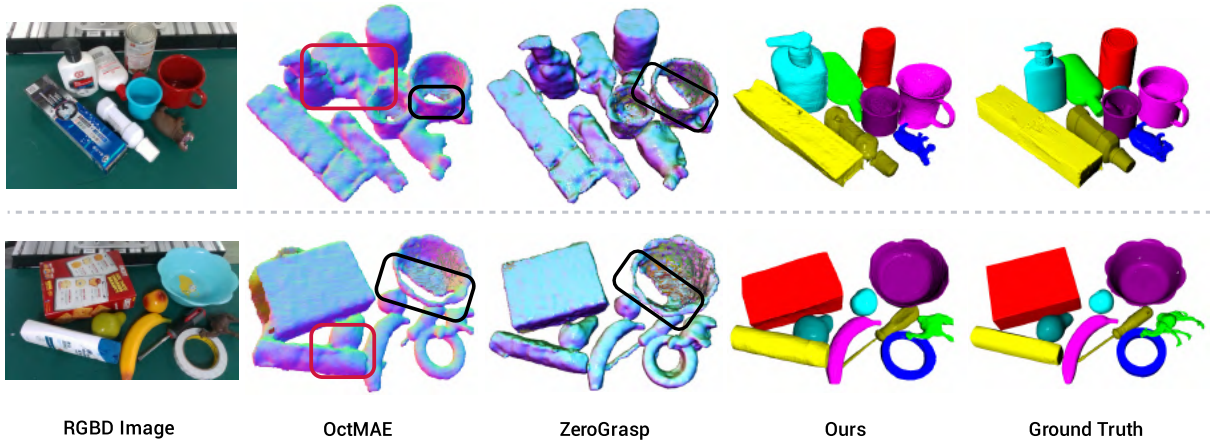

Fig. 5: **Qualitative comparisons** of scene reconstructions on the GraspNet-1B dataset. For each scene we show, the input RGB-D image, OctMAE reconstruction (rendered as normal maps as it predicts scene-level occupancy values), ZeroGrasp reconstruction (rendered as normal maps), our reconstruction (visualized as individually reconstructed object meshes color-matched to the ground truth), and ground-truth object meshes. Highlighted regions indicate missing area (black) or spurious region connecting distinct objects (red).

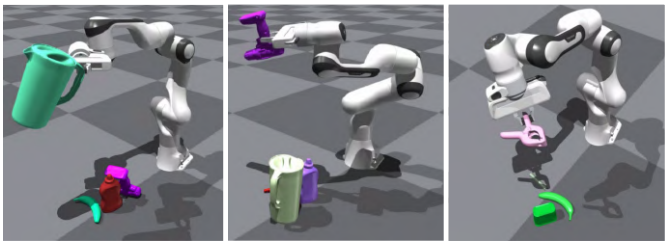

Fig. 6: Evaluating *GSR* on the YCB-V dataset in Isaac Gym. We show a close-up of 3 distinct objects being picked up.

Fig. 7: We demonstrate dexterous grasps using both Shadow Hands and Allegro Hands [5] on objects from the GraspNet-1B dataset, highlighting improved manipulation of complex objects with complete 3D reconstructions.

provide OctMAE and ZeroGrasp with the ground-truth segmentation masks. We also report numbers against the partial point cloud (PartialDecomp).

Metrics: To measure 3D reconstruction quality and fidelity, we report well-known 3D metrics such as *Chamfer distance (CD)* and *Earth Mover's distance-maximum mean discrepancy (MMD-EMD)* metric, similar to existing works [8], [19], [29]. Both CD and MMD-EMD metrics expect pointclouds as input. For SceneComplete, we sample points uniformly from the reconstructed object meshes. Since OctMAE predicts occupancy values, we use the reconstructed point cloud produced by its occupancy values, normal vectors, and SDF. We also report the *Mesh Intersection-over-Union (MIoU)* metric, which is based on comparing the ground-truth meshes with the

	Reconstruction			Grasping
	MIoU↑	CD↓	MMD-EMD↓	GC↓
PartialDecomp	0.166	3.16	3.32	53.5
OctMAE [8]	0.445	1.73	3.11	20.3
ZeroGrasp [9]	0.440	1.86	3.07	18.9
SceneComplete	**0.478**	**1.54**	**3.06**	**16.4**

TABLE I: Comparison of shape reconstruction methods on GraspNet-1B. Higher MIoU indicates better shape fidelity. Lower CD, MMD-EMD, and GC indicate more accurate and feasible reconstructions for downstream grasping. CD and MMD-EMD are scaled by 10^4 and 10^2 respectively.

reconstructed meshes. Specifically, let U^* be the union of volumes enclosed by the ground truth object meshes, and let $\widehat{U}$ be the union of the volumes enclosed by the meshes produced by a reconstruction algorithm. Then the intersection-over-union metric between the meshes is

$$\text{MIoU}(U^*, \widehat{U}) = \frac{U^* \cap \widehat{U}}{U^* \cup \widehat{U}} \quad .$$

MIoU explicitly penalizes both under-reconstructions (missing parts), over-reconstructions (excess geometry), and registration errors. To compute MIoU, we make the reconstructed meshes produced by SceneComplete, ZeroGrasp, and OctMAE water-tight using ManifoldPlus [30].

Collision-free Grasping: We perform a basic test of the utility of SceneComplete on an important downstream task of grasping, by using an antipodal grasp generation method [3] to generate grasps on the objects in the reconstructed scene. To illustrate the importance of whole-scene reconstruction on grasping, we ask the question: of these potential grasps, which ones are *in collision with the ground-truth scene*? Concretely, for a reconstructed scene, we (a) sample a set G of collision-free grasps using antipodal sampling, (b) evaluate grasps in G that would cause a collision in the ground truth scene to obtain a subset G', and compute the *Grasp Collision metric GC* as $(|G'|/|G|)$ which is the percentage of *grasps that the reconstruction would allow, that in fact collide*, similar to the metric adopted by [19], [31]. In our case, G is set to 40.

	Contact-GraspNet GSR	Antipodal Grasping GSR	Overall GSR↑
PartialDecomp	0.46 ± 0.34	0.17 ± 0.13	0.32
SceneComplete	$\mathbf{0.81 \pm 0.2}$	$\mathbf{0.73 \pm 0.18}$	**0.77**

TABLE II: Overall Grasp Success Rate (GSR↑) on the YCB-V dataset for two grasping methods.

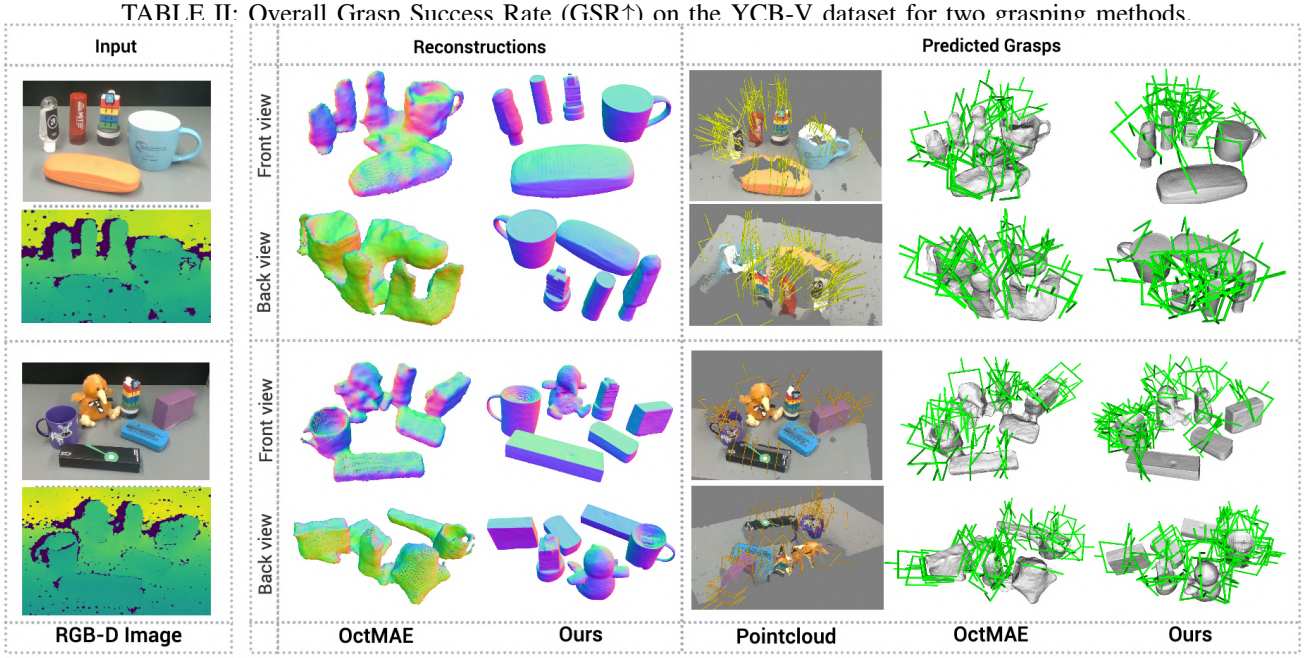

Fig. 8: **Qualitative comparisons** of scene reconstructions on the scans collected in our lab. For each scene we show, the input RGB-D image, OctMAE reconstructions, our reconstructions, grasp proposals on input partial point cloud, grasp proposals on OctMAE's reconstructions, and grasp proposals on our reconstructions. We show the scene from a front and back viewpoint.

Results: As shown in Table IV, our method outperforms the baseline methods in all metrics. We also visualize the comparisons in Figure 5. While OctMAE produces visually plausible reconstructions, it sometimes fails to recover parts of objects that are occluded by other objects, or are partially observable due to the viewpoint. This also results in higher grasp collisions as shown in Table IV, while our method recovers such missing regions. Moreover, since OctMAE directly predicts a scene-level reconstruction, it often hallucinates geometry connecting distinct objects (highlighted in Fig. 5), which results in a lower MIoU. In contrast, our object-centric approach reconstructs each object individually, preserving a clear separation between them. We find that on average, only **16%** of grasps generated by SceneComplete are in collision with real objects, as opposed to 20% of those generated by OctMAE, 19% of those generated by ZeroGrasp, and 53% of those generated by PartialDecomp.

B. Object Grasping and Dexterous Manipulation

We next evaluate the utility of SceneComplete for object grasping and dexterous manipulation in simulation.

Dataset: We use a subset of the YCB-Video [2] dataset consisting of 30 cluttered tabletop scenes with 4 to 5 objects per scene from the YCB [17] dataset for simulated grasping experiments using a parallel jaw gripper. For dexterous manipulation, we evaluate on 20 object instances chosen arbitrarily from the GraspNet-1B [1] dataset.

Baseline: We compare against the raw input partial point cloud (PartialDecomp) as the baseline. Since this experiment is object-centric, its not compatible with OctMAE, which produces scene-level reconstruction without per-object separation. While instance segmentation could in principle be used to individuate objects, they do not reliably recover the extent or shape of each reconstructed object, making fair comparison infeasible for object-centric manipulation tasks. We show comparisons with OctMAE on a scene-level grasping task in the next section.

Metrics: To assess the effectiveness of SceneComplete for object manipulation, we evaluate the *Grasp Success Rate (GSR)* using simulated grasp attempts (a common metric adopted by prior works [32], [33]) in Isaac Gym. Our scene consists of a Franka Emika Panda arm with a parallel jaw gripper, and scenes from the YCB-V dataset [2]. We compare the GSR achieved by SceneComplete against PartialDecomp and adopt two established and distinct grasping methods for generating grasp proposals:

- Antipodal grasping using [3]: This method samples antipodal points to generate collision-free grasps on the object meshes. To make the objects compatible with simulation and to improve efficiency, we perform an approximate convex decomposition of the watertight object meshes into convex hulls using CoACD [34].
- Contact-GraspNet [4]: This method generates grasp proposals directly from point clouds and was trained to predict grasps from partial observations. We generate grasp proposals on both the reconstructed point cloud (from SceneComplete) and the raw partial point cloud, and evaluate these grasps on the ground-truth scenes.

For each grasping method, we first generate candidate

grasps on both SceneComplete's reconstructed scene and PartialDecomp. For each grasp, we simulate a pick attempt on the reconstructed meshes in Isaac Gym and retain only those grasps for which the object remains securely held in the gripper. From these filtered grasps, we randomly select upto 40 grasps per object and evaluate them on the ground-truth object meshes in Isaac Gym. Each evaluation consists of successfully picking up the object from its computed grasp and holding it in the air without dropping, as shown in Fig. 6. GSR is computed as the proportion of successful grasp attempts out of the 40 evaluated grasps. Objects that do not yield valid grasps due to constraints such as exceeding the gripper width are excluded from our evaluation.

Dexterous Grasping and Stability: A significant test of the utility of our approach is whether the object reconstructions support the computation of good dexterous grasps for a multi-fingered hand. We evaluate object reconstruction as:

- Pass a *reconstructed* object mesh O_i into DexGraspNet [5], configured for a dexterous hand (we used both Shadow and Allegro hands), to obtain a grasp g_i.
- Instantiate Isaac Gym with the selected hand and the *ground truth* object mesh.
- Similar to other methods that evaluate dexterous grasping [5], [35], we lift the hand and rotate it within the simulation, and detect whether the object is dropped using PhysX as the physics engine. We visualize dexterous grasps on representative objects using both Shadow and Allegro hands in Fig. 7.

We calculate the percentage of such tests that succeed. For evaluation, we selected 20 objects from the GraspNet-1B dataset and evaluated SceneComplete against PartialDecomp on ground-truth objects—an important upper bound illustrating the reliability of DexGraspNet for selecting such grasps.

Results: As shown in Table II, reconstructing scenes with SceneComplete significantly improves grasp success rates in simulation using a parallel jaw gripper. Across both grasping methods, SceneComplete achieves over **twice** the number of successful grasps compared to those from partial input alone. For dexterous grasping, we observe that the number of stable grasps sampled by DexGraspNet varies with object geometry. On average, we evaluate up to 100 random grasps per object and find that SceneComplete enables **twice** as many valid dexterous grasps compared to PartialDecomp. These improvements show that reconstructing scenes enables more reliable grasping and manipulation in cluttered environments.

C. Real Robot Experiments

We validate the real-world applicability of SceneComplete through experiments on a physical robot. Our experimental setup includes a Franka Emika Panda arm equipped with a wrist-mounted Intel RealSense D435 camera.

Dataset: We evaluate our method on 15 distinct tabletop scenes collected in our lab, each containing 4 to 6 everyday objects, specifically chosen to minimize their likelihood of appearing in the training distribution of the methods.

Baseline: We compare against PartialDecomp and OctMAE in our evaluation.

Metrics: For each scene, we capture an initial RGB-D image using the wrist-mounted camera and randomly select and execute kinematically feasible and collision-free grasps on each object in the scene. We measure the success rate as the percentage of objects that were successfully picked up and report the results in Table III.

Results: Our method achieves a success rate of $\sim$**73%**, significantly outperforming OctMAE and grasps generated directly from the input partial point cloud alone. We show qualitative comparisons with OctMAE in Fig. 8 and additional results in the supplementary. OctMAE struggles to produce plausible reconstructions on these everyday objects, often hallucinating geometry between distinct objects, leading to no valid grasps being generated for some objects. SceneComplete on the other hand, produces object reconstructions that align closely with the ground truth scene. We note that for SceneComplete, most grasping failures occur due to inaccuracies in the estimated object size, which subsequently leads to errors in registration. In general, SceneComplete allows robust manipulation of objects in cluttered real-world settings.

	Partial	OctMAE [8]	SceneComplete
Success Rate↑	36.7 ± 9.9	59.6 ± 15.3	$\mathbf{73.3 \pm 15.2}$

TABLE III: **Real robot** success rate (%) measured as the percentage of objects picked successfully by the real robot for each method.

V. Discussion

a) Limitations: Although our results are very promising, there are of course many failure modes in a composition of so many modules, which can have cascading effects on overall system performance. We outline some failure modes and opportunities for improvement. **Prompting and segmentation:** Occasionally the VLM fails to detect some of the object(s) in the image. Tuning the prompt mitigates the problem, as can prompting multiple ways for multiple hypotheses. **Segmentation:** Grounded-SAM occasionally segments parts of an object along with the full object which leads to multiple reconstruction hypotheses for the same object. We mitigate this partially using IoU-based de-duplication. **Inpainting:** Our current inpainting strategy operates on a relatively isolated object, which removes some important context. We mitigate this by slightly increasing the bounding box and adapting the model, but there is room for improvement. **Image-to-3D:** Although remarkable, these models can sometimes fail to generate plausible reconstructions when given images from highly unusual viewpoints. **Scaling and registration:** Our scaling method is naive and would be improved in some cases by making it non-isotropic. Registration sometimes fails on uniformly-textured objects, where it is difficult to find distinctive features. **Seed value:** The performance of the image-to-3D and inpainting models may sometimes vary depending on the seed value selected.

b) Conclusion & Future Work: We have presented a system that solves full-scene reconstruction from a single real-world RBG-D input in cluttered, occluded scenes with no assumptions about object categories. We have built on an incredibly strong foundation of existing general-purpose

open-domain perception models and believe our approach will be able to adapt to and profit from future advances in such models. One goal of this paper is to emphasize the importance of this problem for robot manipulation in real open-world environments and to encourage others to propose alternative solution strategies. We hope that overall advances in scene understanding in realistic manipulation settings will enable much more robust and capable robot manipulation systems. One important strategy for making the system less error-prone is to move to a more generative setting with quantified uncertainty. If each module could generate multiple hypotheses, conditioned on its inputs, it would be possible to search for an interpretation that is collectively high-probability for all the modules.

APPENDIX
ADDITIONAL RESULTS AND ANALYSES

A. Failure Cases and Limitations of the Modular Pipeline

While having a modular pipeline is positive in several regards, errors can accumulate at each step. This section discusses the limitations of each component in the pipeline, highlights where the overall pipeline is most expected to fail, and includes representative failure scenarios.

Error accumulation is an inherent challenge in our modular pipeline. Certain failure cases such as missed detections and segmentation, scale mismatch, and registration errors do arise. We provide representative failure scenarios and approaches to mitigate these. We also discuss these limitations in detail in Section IV of the manuscript.

In Figures 9, 10, and 11, we show some suboptimal results for the individual modules. In Fig. 12, we show an example of a suboptimal result in the inpainting model cascading to the image-to-3D model. Finally, in Figures 13 and 14, we show failure scenarios that we mitigate using verifiers and simple heuristics at a minimal additional cost, to prevent them from propagating into the later stages of the pipeline, as further detailed in our analysis of verifier-based mitigation strategies. We will add these failure cases and subsequent mitigation to the website.

However, as we show later via both qualitative and quantitative metrics, SceneComplete outperforms existing state-of-the-art methods and works robustly on in-the-wild scenes without any additional training or with minimal adaptation. Moreover, the modular nature of SceneComplete allows us to swap out individual modules in the existing pipeline with improved and more efficient versions as and when they become available.

B. Comparison with Recent Baselines

This section discusses and compares SceneComplete with several important recent baselines, including Gen3DSR [12], ZeroGrasp [9], CAST [11], and FSD [6]. In the main manuscript, the Related Work section groups prior methods into feed-forward multi-object full-scene reconstruction approaches such as FSD [6], ShaPO [7], OctMAE [8], ZeroGrasp [9], CRISP [10] etc., and compositional methods such as Open6DOR [13], Gen3DSR [12], CAST [11], and SceneComplete, and highlights the advantages of our method and how it differs from existing approaches.

Regarding comparisons (these results and comparisons are also included on the project website):

- **Gen3DSR**: This method generates scene reconstructions directly from a single RGB image and relies on a monocular depth estimation model for geometry. We do not report quantitative comparisons with Gen3DSR because the model-predicted output is not scaled in 3D (ground-truth metric depth)—robotics not being its primary use case. We attempted to adapt the released code to accept ground-truth partial metric depth (similar setting as ours) with modified camera intrinsics, but the reconstruction was unstable, probably due to the real-world metric depth being noisy. We therefore show qualitative comparisons against Gen3DSR in Fig. 15 **after manually adjusting their scale using isotropic scaling**, and note that their modular framework is conceptually similar to ours but technically different.
- **ZeroGrasp**: ZeroGrasp's code was released just one week prior to our RA-L submission, which prevented inclusion into the original manuscript. In the current version, we have added quantitative comparisons against ZeroGrasp on the GraspNet-1B dataset (Table IV). ZeroGrasp also outputs grasps along with the reconstruction, and hence we evaluated their predicted grasp proposals when computing the grasp collision (GC) metric. We noticed that a large percentage of their predicted grasps collided with the reconstructed scene, so we filtered them before computing collisions with the ground-truth scene, and still observed the GC metric to be fairly high ($\sim$ 43% grasp collisions). For a fair comparison, we generated antipodal grasps on ZeroGrasp's reconstructed scene and computed the GC again, and reported the metrics (these are also included in the main manuscript). We show these qualitatively in Fig. 17. We also show qualitatively in Fig. 16 ZeroGrasp's reconstructions against SceneComplete on the GraspNet-1B dataset, and observe that our reconstructions preserve geometry better and generate more plausible reconstructions. We additionally run SceneComplete on the public scans released by ZeroGrasp and show representative examples in Fig. 18, where our reconstructions are visually more plausible and preserve better geometry.
- **CAST** and **FSD**: Unfortunately, neither of these works have released their codebases, preventing direct experimental comparison. Nevertheless, they are discussed in detail in the Related Work section of the main manuscript, where it is clarified how SceneComplete differs in formulation and execution.

C. Runtime Analysis and Speed–Accuracy Trade-offs

Timing analysis is important for understanding the trade-off between reconstruction quality and efficiency. This section reports the runtime of SceneComplete at the module level and for the overall pipeline (Table V), together with runtime comparisons against representative feed-forward and compositional baselines (Table VI).

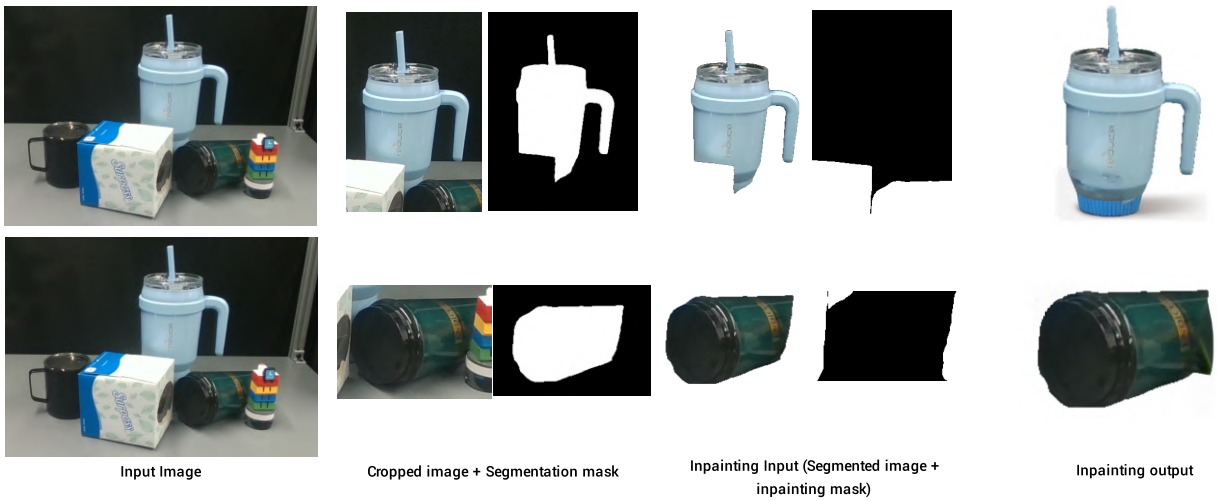

Fig. 9: Some examples of suboptimal inpainting are shown. Most errors occur when the full spatial extent of the object is not visible. As shown, our inpainting model takes as input the segmented image of the target object and an inpainting mask constructed from the 2D bounding box around the object. Within this box, all pixels belonging to other objects are marked as the region to be inpainted. However, if the object is heavily occluded—e.g., fully hidden from one side, the mask does not capture the true extent of the occluded region. As a result, the model is forced to infer the object's missing parts within the bounded region, resulting in suboptimal results.

Fig. 10: Representative examples of suboptimal image-to-3D generations for a few representative examples for transparent objects or objects in unusual viewpoints.

	Reconstruction			Grasping	
	MIoU↑	**CD↓**	**MMD-EMD↓**	**GC↓**	**ZGC↓**
PartialDecomp	0.166	3.16	3.32	53.5	-
OctMAE [8]	0.445	1.73	3.11	20.3	-
ZeroGrasp [9]	0.440	1.86	3.07	18.9	43.14
SceneComplete	**0.478**	**1.54**	**3.06**	**16.4**	-

TABLE IV: Comparison of shape reconstruction methods on GraspNet-1B. Higher MIoU indicates better shape fidelity. Lower CD, MMD-EMD, and GC indicate more accurate and feasible reconstructions for downstream grasping. CD and MMD-EMD are scaled by 10^4 and 10^2 respectively. ZGC is evaluated on grasps predicted directly by ZeroGrasp.

TABLE V: SceneComplete module-wise runtime and peak VRAM. Times reported are inference time in seconds (forward-pass).

Module	**Time/obj (4090)** (s)	**Peak VRAM** (GB)
VLM prompt generation (GPT-4o)	*2.5s*	-
Grounded segmentation (GroundedSAM2)	*0.5s*	*2GB*
2D inpainting (BrushNet+LoRA)	*3.5s*	*4.5GB*
Image-to-3D (InstantMesh)	*6s*	*18GB*
Scale estimation (correspondence matching)	*5s*	*2.5GB*
Registration (FoundationPose)	*1.2*	*7GB*
Total (per object)	*19.7*	*18GB*

While the current implementation is slower than feed-forward methods such as FSD, ShaPO, Oct-MAE, and ZeroGrasp, this is largely due to the modular design (which is a characteristic of recent modular methods that are slower [11]–[13] than feed-forward approaches but are much more generalizable). SceneComplete is, however, approximately 2.5× faster than Gen3DSR (another compositional approach). A key advantage of this modular design is that as newer and more efficient foundation models become available, the pipeline can immediately benefit without retraining. This effect is illustrated by substituting alternate modules in Table VII. Moreover, due to its compositional design, SceneComplete works robustly on in-the-wild scenes without any additional retraining or with minimal adaptation, whereas feed-forward approaches are typically closed-set and need to be trained extensively on specific datasets.

The runtime can be further improved with hardware parallelization. On GPUs with larger VRAM (e.g., A100s), all modules after the initial VLM step can be executed in parallel across multiple objects (with peak VRAM being the primary

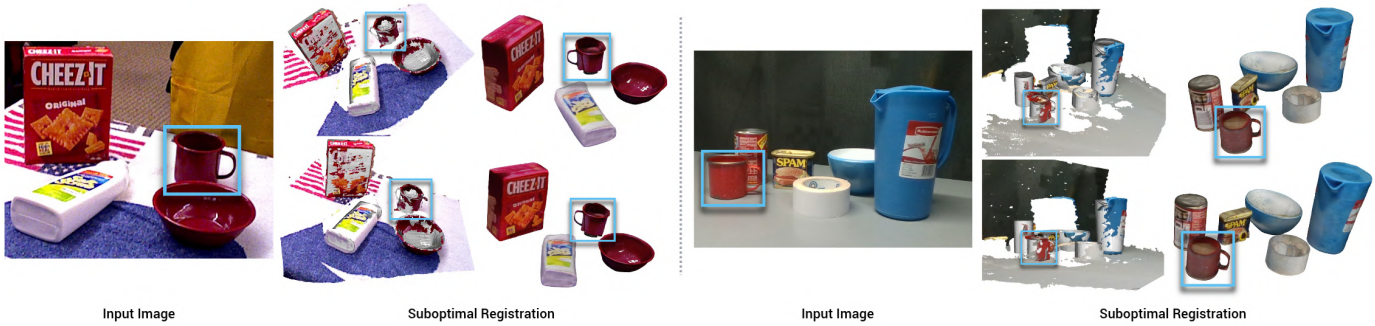

Fig. 11: Representative examples of suboptimal registration using FoundationPose. Notice the misaligned handle of the mug, and the inverted mug in registration.

Fig. 12: Suboptimal inpainting outputs cascades to suboptimal image-to-3D generation.

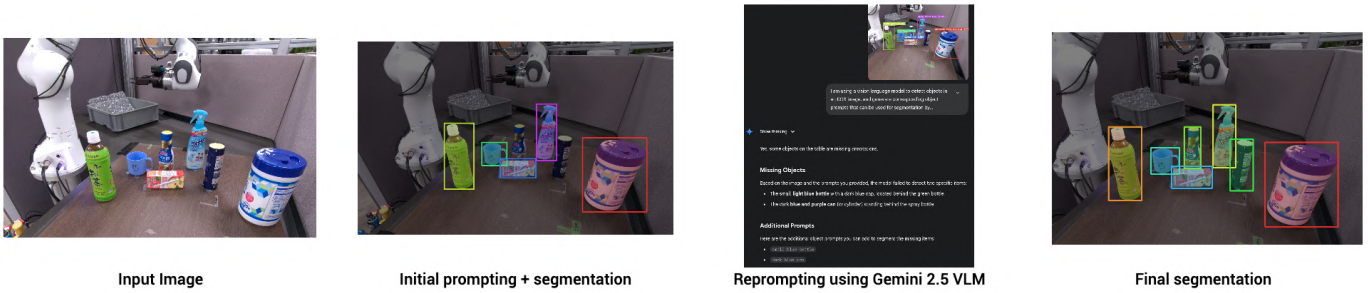

Fig. 13: Initial bad segmentation due to all objects not being detected by the GPT-4o VLM. These are fixed by prompting Gemini 2.5 Flash VLM with the annotated image, and detected objects, and subsequently all objects are segmented.

TABLE VI: Runtime comparison across different methods. Openness and retraining needs are included to contextualize speed/accuracy trade-offs. For OctMAE and ZeroGrasp, the reported time includes the time taken to generate detections and segmentation masks.

Method	**Open-set** (Y/N)	**Retraining** Required?	**Time** (s / obj or scene)	**Peak VRAM** (GB)
OctMAE [8]	N	Y	*3+0.5*	*1GB*
ZeroGrasp [9]	N	Y	*3+0.6*	*10GB*
Gen3DSR [12]	Y	No / minimal	*50*	*19GB*
SceneComplete	Y	No / minimal	*19.7*	*18GB*

bottleneck), substantially reducing the total processing time. In practice, reconstruction of 5 objects can be run in parallel on an A100 with 80 GB VRAM, significantly reducing the overall running time for a scene. These additional results and comparisons, along with qualitative examples, are also provided on the project website.

TABLE VII: SceneComplete (alternative configuration) module-wise runtime and peak VRAM. Times are inference time (forward pass) in seconds on a single GPU (RTX 4090). Totals are shown for two image-to-3D choices: TriPoSR [36] and Trellis [37].

Module (Alt Config)	**Time/obj (s)**	**Peak VRAM (GB)**
VLM detection (Gemini 2.5 Flash)	*2s*	-
Grounded segmentation (GroundedSAM2)	*0.5s*	*2GB*
2D inpainting (LaMa [38])	*0.2s*	*3GB*
Image-to-3D (TriPoSR [36])	*0.9s*	*8GB*
Image-to-3D (Trellis [37])	*7s*	*19GB*
Scale estimation (bounding-box heuristic)	*≈0s*	-
Registration (FoundationPose)	*1.2s*	*7GB*
Total (TriPoSR)	*4.8*	*8GB*
Total (Trellis)	*10.9*	*19GB*

D. Novelty of the SceneComplete Pipeline

While the idea of composing multiple modules is not entirely new [11]–[13], the specific composition and adaptation

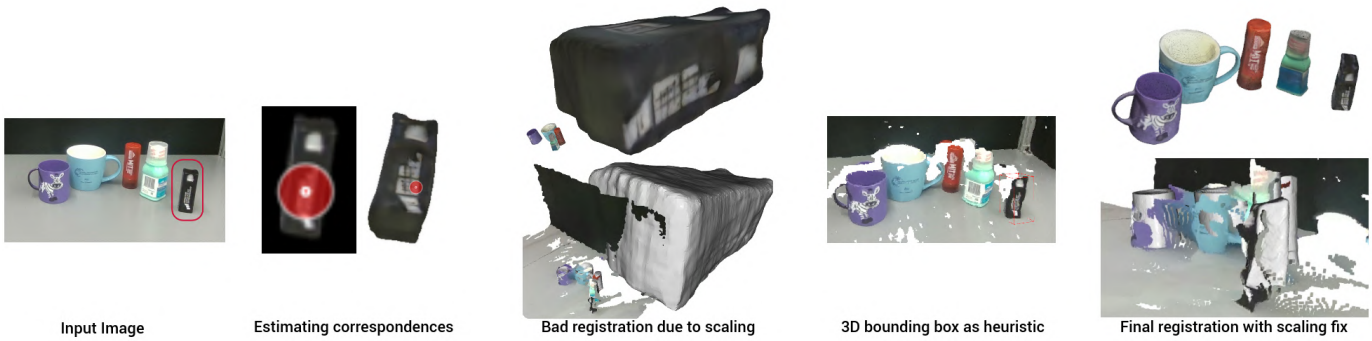

Fig. 14: When correspondence matching for scale-estimation produces suboptimal scaling, this cascades to the registration step. We use the 3D bounding box (estimated from the object's partial point cloud) as a heuristic to estimate the scale factor.

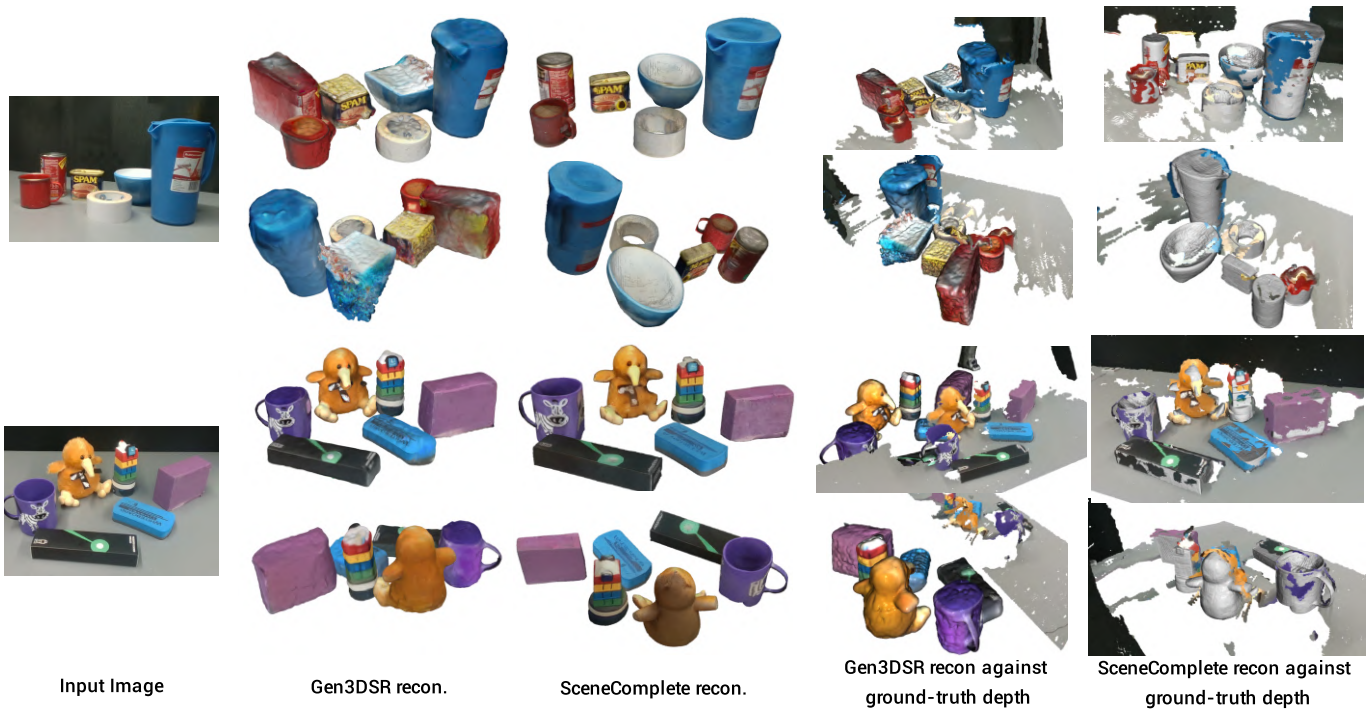

Fig. 15: We compare SceneComplete reconstructions against Gen3DSR [12] on a few tabletop scans. SceneComplete generates much more visually plausible reconstructions. Additionally, Gen3DSR generates reconstructions from just an RGB image, and uses a monocular depth estimation model for depth prediction, resulting in misalignment with the ground-truth depth. **We make a best effort to manually align** Gen3DSR's reconstructions with the ground-truth by estimating an isotropic scaling factor, but note that the registrations and scaling are still poor. Comparatively, SceneComplete directly produces reconstructions that are aligned with the ground-truth depth.

proposed in SceneComplete is, to our knowledge, the first system to achieve robust object-centric 3D scene completion from a single RGB-D input in open-world cluttered settings. Identifying a successful configuration required extensive experimentation with different modules, and the final pipeline represents a carefully chosen integration that balances accurate reconstruction and generalization to novel scenes.

A key contribution is the adaptation of recent image-to-3D models [21]–[23], [36], [37], [39], [40], originally developed and evaluated primarily for visual quality in computer vision, into a robotics pipeline where geometry and structural fidelity are critical. By coupling these models with inpainting, dense correspondence scaling, and pose registration, SceneComplete enables reconstructions that directly support robotic tasks such as grasping (dexterous and parallel-jaw), manipulation, and motion planning. This goes beyond visual reconstruction and demonstrates a novel way of leveraging vision foundation models for robotic applications.

The results show consistent improvements over existing end-to-end baselines across multiple reconstruction and grasping metrics (Table IV), as well as qualitatively in Fig. 16 and Fig. 18. SceneComplete also exhibits superior performance compared to a state-of-the-art modular pipeline, Gen3DSR [12], as shown qualitatively in Fig. 15. Importantly, unlike prior feed-forward approaches that are closed-set and require extensive dataset-specific training, SceneComplete op-

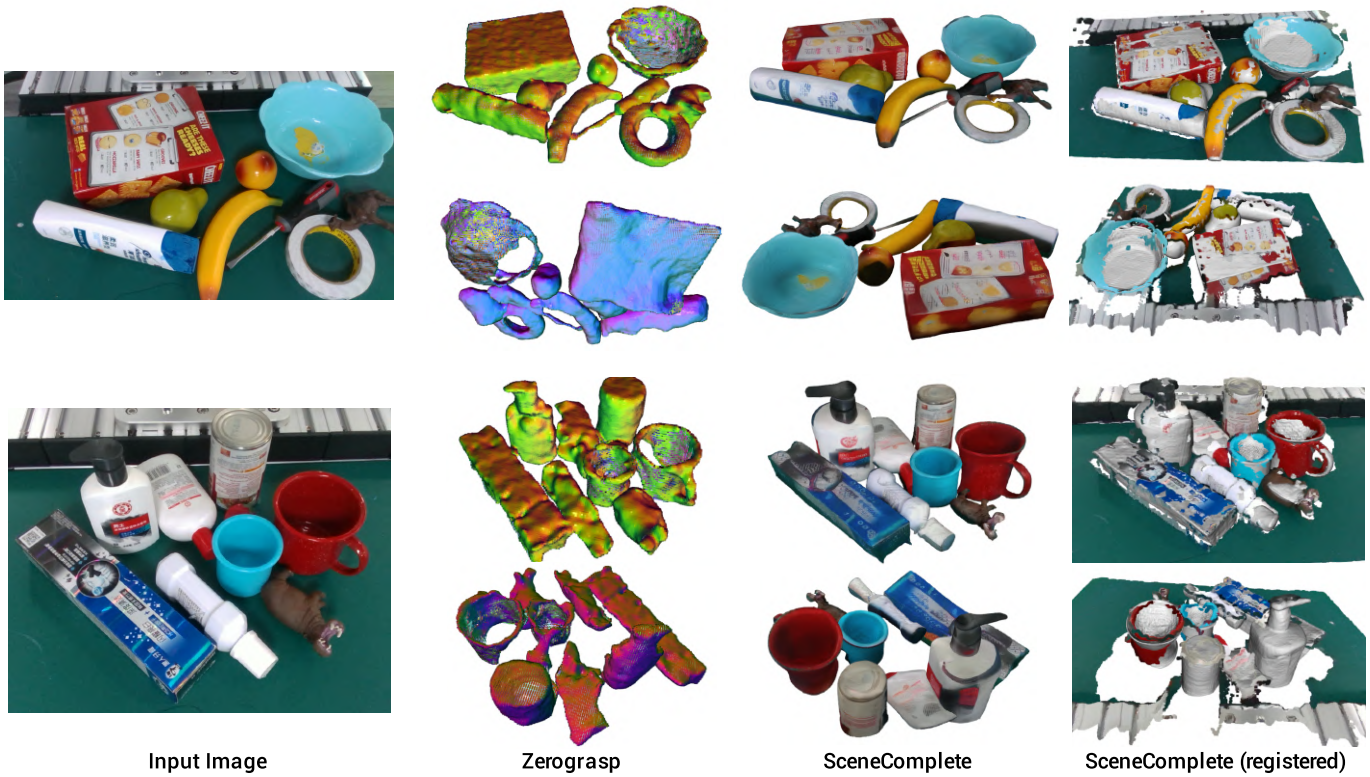

Fig. 16: Representative examples comparing reconstructions generated by SceneComplete and ZeroGrasp on the GraspNet-1Billion [1] dataset. SceneComplete generates more plausible reconstructions and preserves geometry better.

erates in an open-set manner and works out-of-the-box in real-world settings.

Furthermore, because of its modular design, the pipeline immediately benefits from advances in foundation models, so that as newer models become available, the overall system becomes more accurate and efficient. The codebase will be open-sourced upon publication to support reproducibility, and additional qualitative and quantitative comparisons will be included on the project website.

E. Module Choices and Ablation Studies

The choice of modules in SceneComplete is deliberate and based on extensive evaluation of alternatives, with a focus on open-set generalization and robustness in cluttered, real-world settings. A justification of the key modules is provided below:

- **Dense Correspondence Matching for scale estimation:** The meshes produced by InstantMesh (the image-to-3D stage) are not aligned with metric depth (Fig. 2 of the main manuscript), since they are generated purely from RGB appearance. Because the setting is open-set, where no prior model of the object is available, the only reliable source of metric dimensions is the partial point cloud of the observed object. A straightforward alternative explored was enclosing the partial point cloud and the reconstructed mesh in bounding boxes and using their dimensions to compute the scale factor (highlighted as a mitigation step in Fig. 14 when correspondence-matching for scale estimation fails). However, this approach has two fundamental issues: (i) the partial cloud and the reconstructed mesh are generally not axis-aligned, making box-based comparisons inaccurate, and (ii) the partial point cloud is itself incomplete due to occlusion and viewpoint limitations, which typically leads to underestimation of object size. Dense Correspondence Matching overcomes these problems by directly establishing correspondences between the partially observable points and the reconstructed mesh and then estimating the relative spread of points in 3D. This produces a more consistent and geometry-aware scale estimate in open-world settings.
- **FoundationPose for 6D pose estimation:** FoundationPose is selected instead of traditional registration methods (e.g., ICP or PnP-RANSAC based alignment) because it is category-agnostic, fast, and shows strong generalization in in-the-wild settings. Classical registration techniques rely on iterative optimization over many steps, which is both computationally expensive and prone to local minima, especially when the partial point cloud is noisy or incomplete. FoundationPose, in contrast, uses a render-and-compare strategy that directly matches candidate object views against the observed frame. Despite not being purely feed-forward, it is highly optimized and significantly faster than classical optimization-based approaches, while producing more accurate alignment. This is highlighted via an example in Fig. 19. Importantly, FoundationPose requires a CAD model of the object at inference time to operate in an open-set setting. This fits naturally in the pipeline, since SceneComplete directly

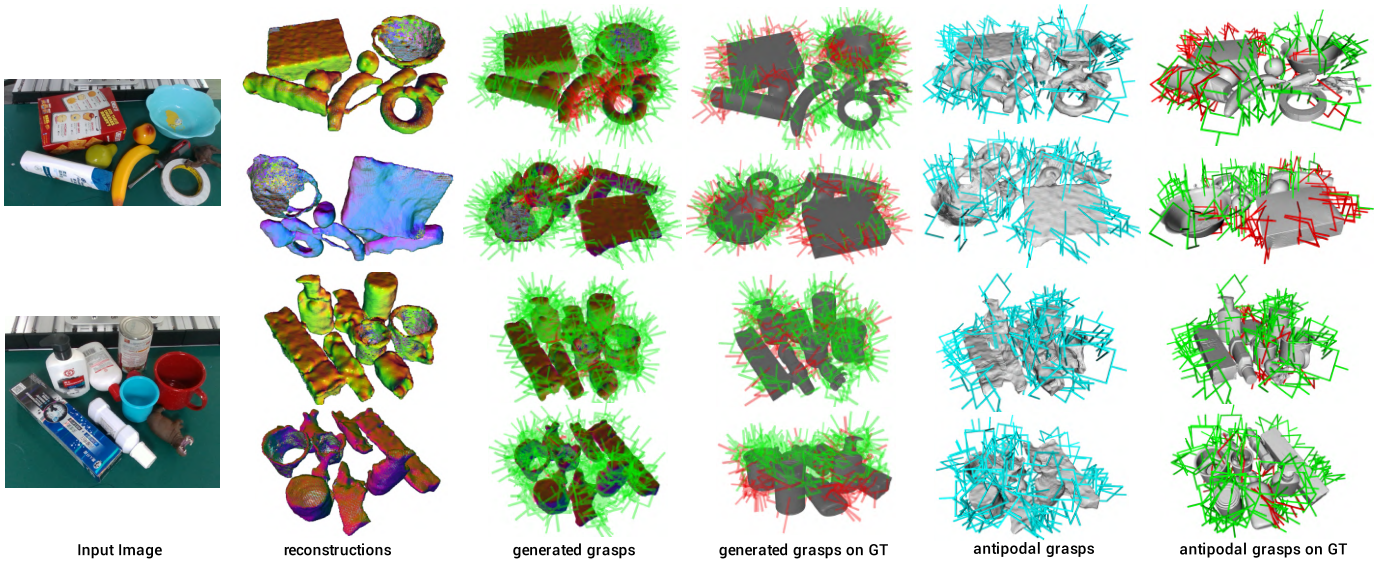

Fig. 17: ZeroGrasp reconstructions on the GraspNet-1B dataset with the model-predicted grasps visualized. Even after filtering the predicted grasps with the reconstructed scene, a large number of grasps collide with the ground-truth scene (visualized in red), making them not suitable for direct manipulation. For a fair comparison, antipodal grasps (similar to those used in SceneComplete) are generated and the colliding grasps on the ground-truth scene are visualized.

generates an unaligned, scaled 3D mesh for each object as input. This allows alignment of reconstructed meshes in an open-set manner, supporting novel objects without the need for category-specific training. The quantitative comparisons in the FoundationPose paper [28] further compare against ICP-based optimization methods. Additionally, FoundationPose is observed to be highly robust to slight errors in the estimation of the scale factor, whereas ICP-based algorithms are more susceptible to such miscalculations.

- **Other module choices:** For the individual components, vision foundation models with open-ended capabilities are intentionally selected, ensuring that the pipeline can generalize to unseen objects and scenes. The primary contribution is not in the selection of any single module, but rather in orchestrating a composition of modules that can work together to achieve open-world scene completion suitable for robotic manipulation. To validate these choices, ablations against recent models and heuristic-based baselines are provided. Given the modular approach, these models can be incorporated with minimal adaptation.

  Qualitative results on alternate choices are shown in this appendix, with references to the relevant papers for quantitative comparisons (these will also be included on the project website). Concretely:

  – For VLM detection, GPT-4o is compared against Gemini-2.5 Flash. Gemini-2.5 Flash produces slightly worse detections, but is much less expensive to query (in terms of tokens required) compared to GPT-4o.

  – For image inpainting, BrushNet (with LoRA adaptation) is compared against LaMa image inpainting [38]. BrushNet is a diffusion-based generative architecture, while LaMa is an encoder–decoder architecture. These comparisons are shown in Fig. 20. Both architectural styles have merits: LaMa is more robust to oddly shaped inpainting masks that arise from occlusions, whereas BrushNet has stronger generative capabilities due to its diffusion-based architecture and initialization from Stable Diffusion. Overall, LoRA adaptation increases BrushNet's robustness to oddly shaped masks while retaining its generative capabilities (as shown in Fig. 22), and it performs qualitatively better compared to LaMa inpainting across most scenes.

  – For image-to-3D reconstruction, InstantMesh [23] is compared against two recent state-of-the-art open-source methods, Trellis [37] and TriPoSR [36], in Fig. 21. While TriPoSR is an order of magnitude faster than InstantMesh (Tables VII and V), it produces much worse reconstructions. Trellis has similar compute requirements to InstantMesh, but in evaluations, InstantMesh produces slightly better reconstructions in in-the-wild settings.

  – For pose estimation, FoundationPose is compared against traditional optimization-based methods such as ICP-based optimization, with a representative example shown in Fig. 19.

  These ablations reinforce two points: (i) the module choices are empirically justified, and (ii) the modular design allows new models to be incorporated with minimal adaptation. This ensures that SceneComplete will continue to improve as stronger foundation models are released. In the forthcoming code release, these module variants will be exposed as configuration options, allowing users to select among different models depending on their priorities for speed, accuracy, or resource availabil-

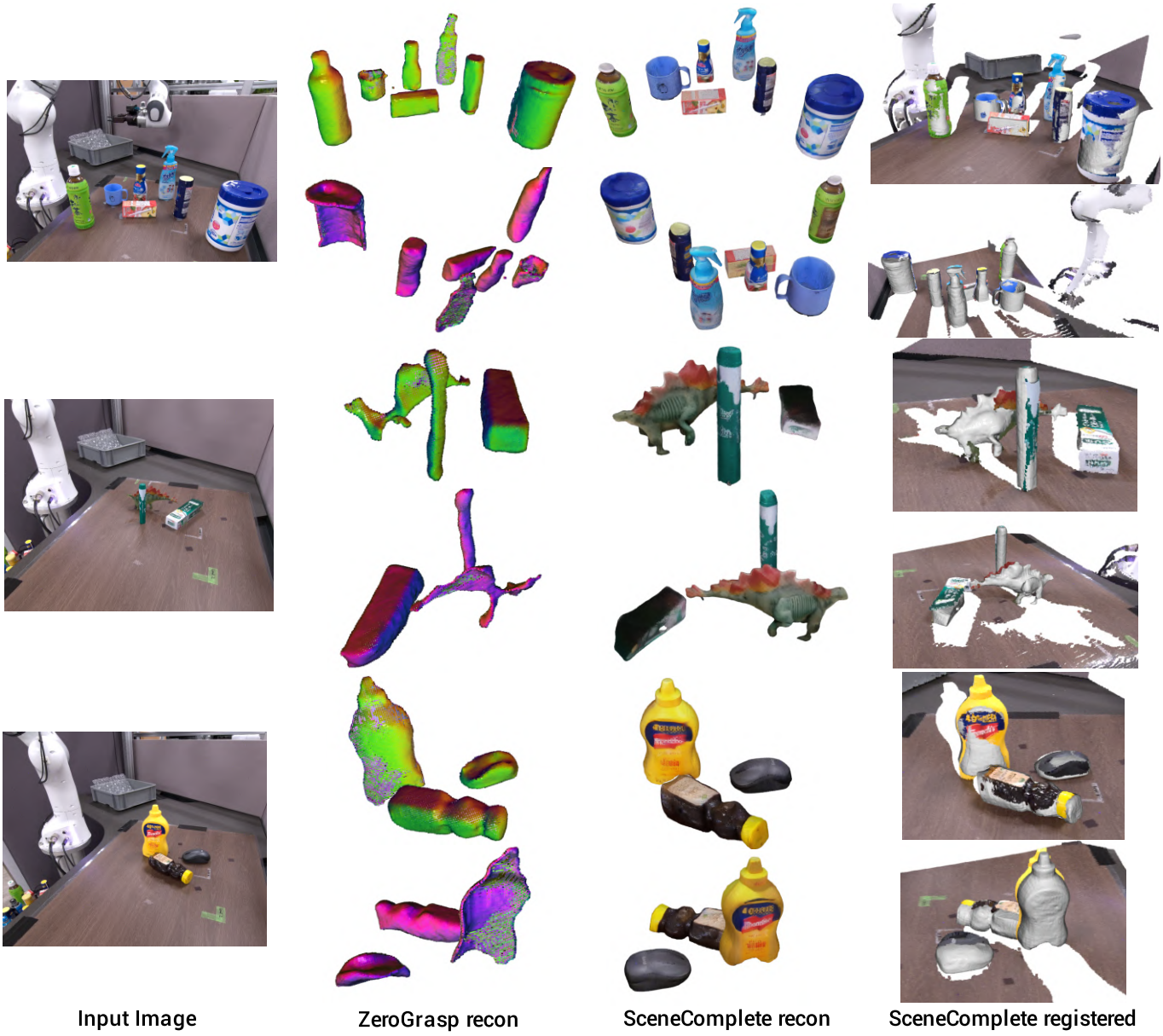

Fig. 18: SceneComplete comparisons with ZeroGrasp on ZeroGrasp's collected scenes.

ity.

In addition, further analysis of the LoRA adaptation for the inpainting module is provided:

- The LoRA-adapted BrushNet and the pretrained BrushNet model are evaluated on tabletop objects (YCB + scans recorded in the lab). These dataset-specific artifacts and the corresponding fixes are shown in Fig. 22. Similar outputs generated by the adapted and the pretrained model are shown in Fig. 23.
- To address generalizability, the adapted model is evaluated against the pretrained BrushNet model on the original Laion-Aesthetic [41] dataset (on which the model was trained), as shown in Fig. 24. The results indicate that the model retains its general-purpose inpainting ability while adapting to tabletop scenes, confirming that generalization is preserved.

F. Scalability and Runtime Considerations

Runtime is an important consideration, especially for larger scenes. A detailed timing analysis, as well as the peak VRAM memory consumption for each individual module in the pipeline, is provided in Table V. As shown there, the most time-consuming step is the image-to-3D stage, which is a natural target for further research and optimization. Alternative module choices are also integrated as drop-in replacements in the pipeline, with their timings reported in Table VII.

A key limiting factor for end-to-end runtime is the available GPU memory, since many of the foundation models used in SceneComplete are memory-intensive. On an RTX 4090 GPU with 24 GB of VRAM, modules must be executed sequentially, which increases total latency. To explore scalability, experiments are also conducted on an NVIDIA A100 with 80 GB of VRAM, which allows multiple modules to be executed in

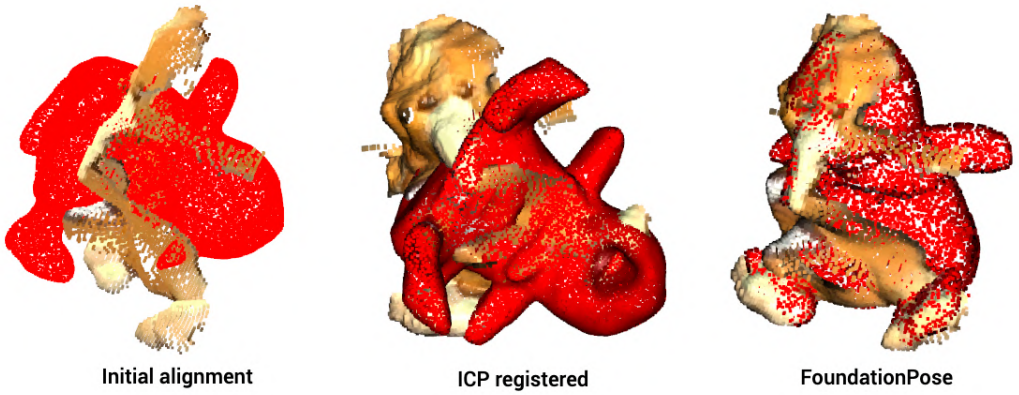

Fig. 19: Comparing registration between ICP and FoundationPose.

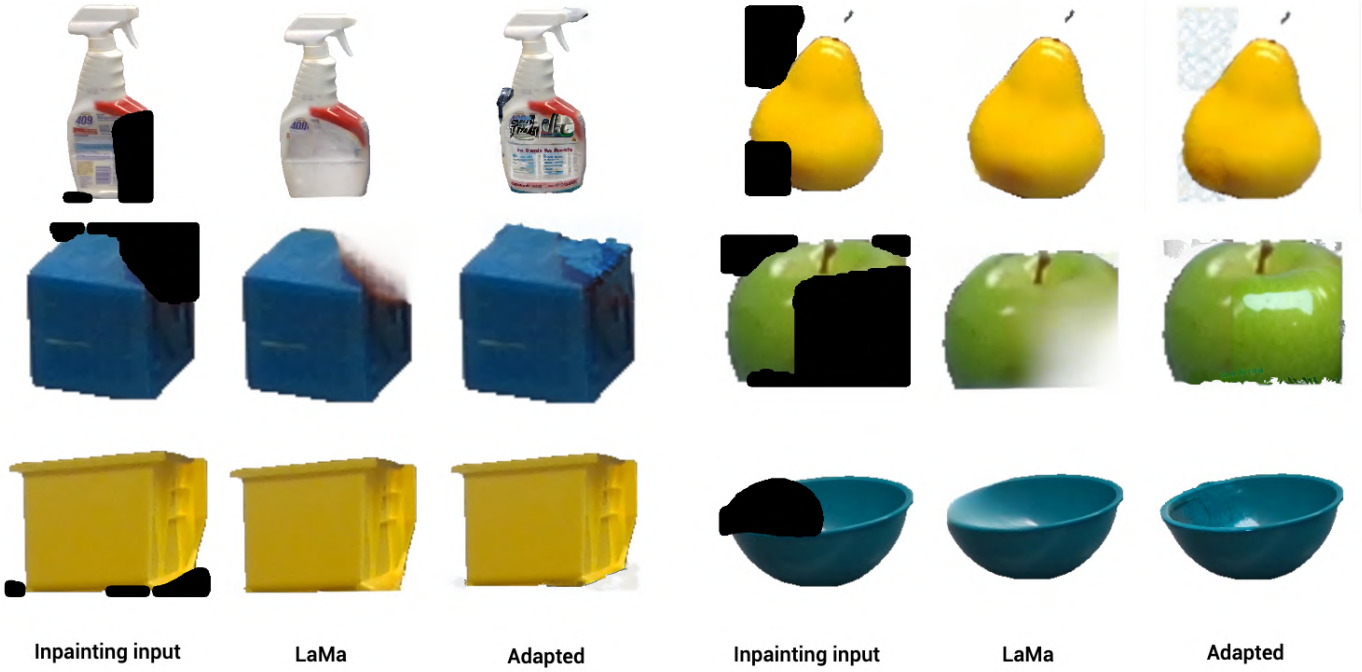

Fig. 20: Comparing the different inpainting models.

parallel. In this setting, completions for up to 5 objects can be run concurrently, substantially reducing the effective per-object runtime.

While SceneComplete is slower than feed-forward approaches such as OctMAE and ZeroGrasp (Table VI), it offers a complementary set of benefits: the pipeline works out-of-the-box in open-set conditions and requires no dataset-specific retraining. In contrast, feed-forward approaches often require hundreds of GPU-hours of training on modern hardware to produce competitive results, and the resulting models are typically closed-set, restricted to the benchmarks that they were trained on. SceneComplete therefore provides a more general and readily deployable solution with the flexibility to integrate newer and faster models as they become available.

Finally, in line with current hardware and model trends, both inference time and memory usage are expected to continue to decline as more efficient backbones are released and more powerful GPUs become available, further improving the practicality of SceneComplete.

G. Generalizability of LoRA-based Inpainting Adaptation

The use of LoRA fine-tuning on BrushNet [15] is intended as a lightweight adaptation to improve performance on tabletop scenes without limiting generalization. BrushNet is trained on BrushData, which is constructed from the Laion-Aesthetic subset of Laion-5B and contains generic web images. While this makes BrushNet broadly capable in open-set inpainting, it was observed to introduce artifacts (hallucinations) that reflect biases present in its training dataset. To bridge this gap, BrushNet is fine-tuned with LoRA using a small set of representative tabletop examples consisting of YCB objects. Crucially, LoRA preserves the generalization ability of the pre-trained backbone while enabling adaptation to tabletop scenes. As described in [16], LoRA achieves this by injecting low-rank matrices into the frozen weight space of the model. This approach allows the visual priors from Laion-Aesthetic to

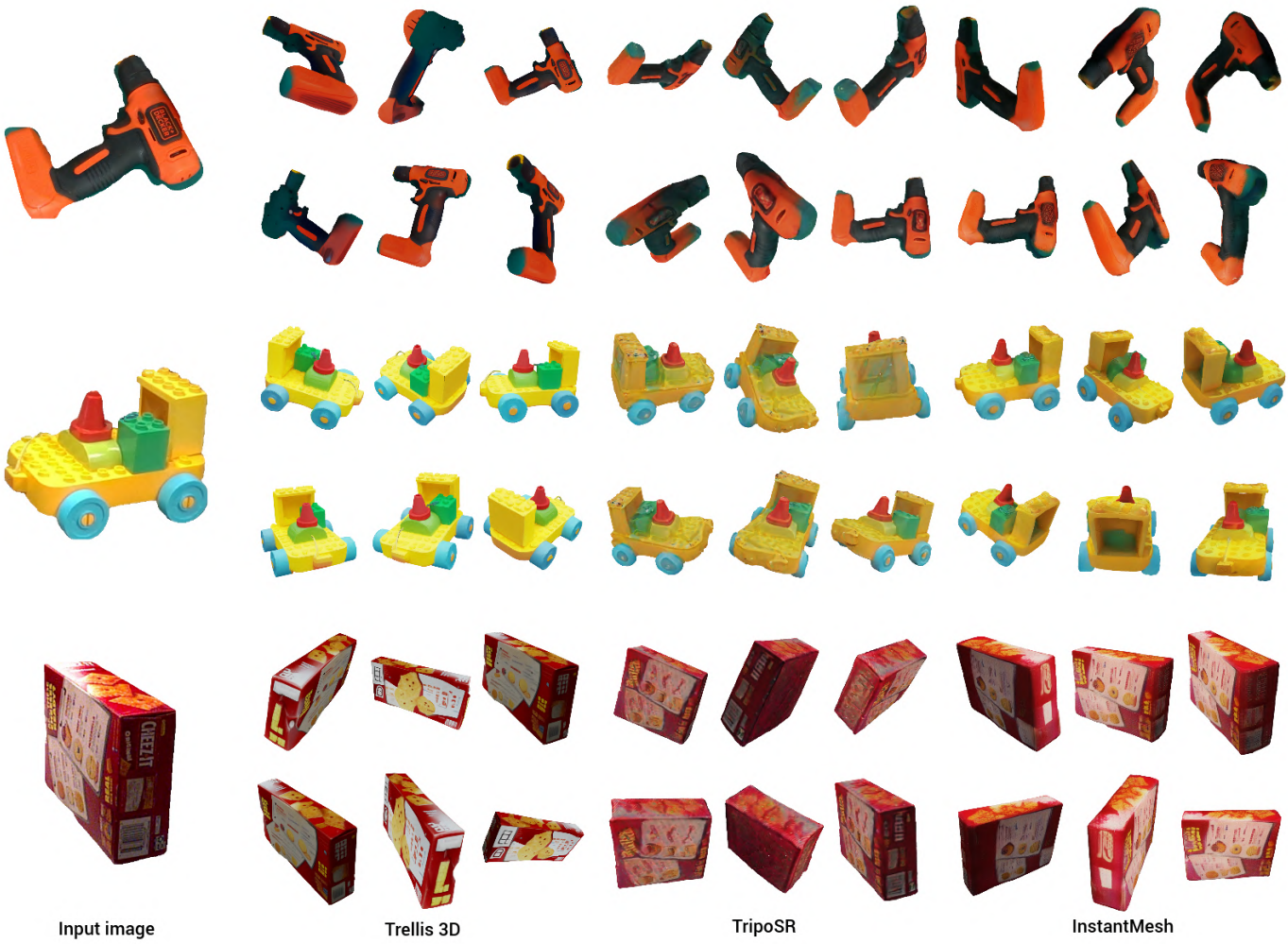

Fig. 21: Image-to-3D comparison for different state-of-the-art models.

remain untouched, while the low-rank updates provide a small adaptation to the structured occlusion patterns encountered in the tabletop setting.

To strengthen this discussion, generalizability is explicitly analyzed in complementary settings, and the corresponding results are made available on the project website.

In summary, while LoRA-adapted BrushNet provides a boost on YCB scenes, the low-rank adaptation helps the model infer structural properties of inpainting needed for general tabletop occlusions, without restricting it to tabletop settings. SceneComplete therefore remains open-set and generalizable by design.

References

[1] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 444–11 453.

[2] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[3] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, p. eaau4984, 2019.

[4] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," 2021.

[5] R. Wang, J. Zhang, J. Chen, Y. Xu, P. Li, T. Liu, and H. Wang, "Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 359–11 366.

[6] M. Lunayach, S. Zakharov, D. Chen, R. Ambrus, Z. Kira, and M. Z. Irshad, "Fsd: Fast self-supervised single rgb-d to categorical 3d objects," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 14 630–14 637.

[7] M. Z. Irshad, S. Zakharov, R. Ambrus, T. Kollar, Z. Kira, and A. Gaidon, "Shapo: Implicit representations for multi-object shape, appearance, and pose optimization," in *European Conference on Computer Vision*. Springer, 2022, pp. 275–292.

[8] S. Iwase, K. Liu, V. Guizilini, A. Gaidon, K. Kitani, R. Ambruş, and S. Zakharov, "Zero-shot multi-object scene completion," in *European Conference on Computer Vision*. Springer, 2024, pp. 96–113.

[9] S. Iwase, M. Z. Irshad, K. Liu, V. Guizilini, R. Lee, T. Ikeda, A. Amma, K. Nishiwaki, K. Kitani, R. Ambrus *et al.*, "Zerograsp: Zero-shot shape reconstruction enabled robotic grasping," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 17 405–17 415.

[10] J. Shi, R. Talak, H. Zhang, D. Jin, and L. Carlone, "Crisp: Object pose and shape estimation with test-time adaptation," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 11 644–11 653.

[11] 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," *ACM Transactions on Graphics (TOG)*, vol. 44, no. 4, pp. 1–19, 2025.

[12] A. Ardelean, M. Özer, and B. Egger, "Gen3dsr: Generalizable 3d scene reconstruction via divide and conquer from a single view," *arXiv preprint*

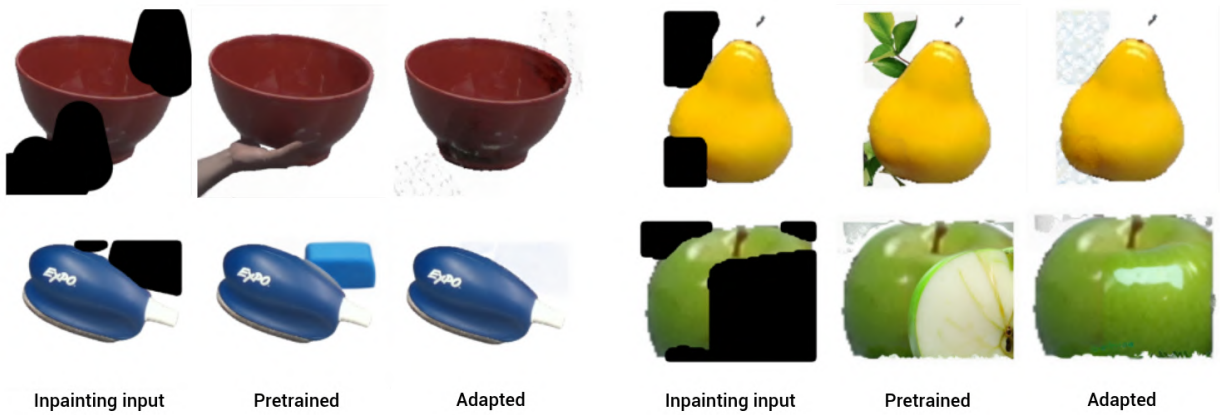

Fig. 22: Differences in the inpainting model with and without adaptation using LoRA.

Fig. 23: Similarities in the inpainting model with and without LoRA adaptation.

arXiv:2404.03421, 2024.

[13] Y. Ding, H. Geng, C. Xu, X. Fang, J. Zhang, S. Wei, Q. Dai, Z. Zhang, and H. Wang, "Open6dor: Benchmarking open-instruction 6-dof object rearrangement and a vlm-based approach," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 7359–7366.

[14] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan *et al.*, "Grounded sam: Assembling open-world models for diverse visual tasks," *arXiv preprint arXiv:2401.14159*, 2024.

[15] X. Ju, X. Liu, X. Wang, Y. Bian, Y. Shan, and Q. Xu, "Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion," *arXiv preprint arXiv:2403.06976*, 2024.

[16] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[17] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols," *arXiv preprint arXiv:1502.03143*, 2015.

[18] J. Zhang, X. Chen, Z. Cai, L. Pan, H. Zhao, S. Yi, C. K. Yeo, B. Dai, and C. C. Loy, "Unsupervised 3d shape completion through gan inversion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1768–1777.

[19] B. Sen, A. Agarwal, G. Singh, B. Brojeshwar, S. Sridhar, and M. Krishna, "Scarp: 3d shape completion in arbitrary poses for improved grasping," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3838–3845.

[20] 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*. Springer, 2020, pp. 523–540.

[21] 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*, 2023, pp. 9298–9309.

[22] M. Liu, C. Xu, H. Jin, L. Chen, M. Varma T, Z. Xu, and H. Su, "One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[23] J. Xu, W. Cheng, Y. Gao, X. Wang, S. Gao, and Y. Shan, "Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models," *arXiv preprint arXiv:2404.07191*, 2024.

[24] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," *arXiv preprint arXiv:2209.14988*, 2022.

[25] H. Wang, X. Du, J. Li, R. A. Yeh, and G. Shakhnarovich, "Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 619–12 629.

[26] J. Seo, W. Jang, M.-S. Kwak, H. Kim, J. Ko, J. Kim, J.-H. Kim, J. Lee, and S. Kim, "Let 2d diffusion model know 3d-consistency for robust text-to-3d generation," *arXiv preprint arXiv:2303.07937*, 2023.

[27] S. Amir, Y. Gandelsman, S. Bagon, and T. Dekel, "Deep vit features as dense visual descriptors," *arXiv preprint arXiv:2112.05814*, vol. 2, no. 3, p. 4, 2021.

[28] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 868–17 879.

[29] D. W. Shu, S. W. Park, and J. Kwon, "3d point cloud generative adversarial network based on tree structured graph convolutions," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 3859–3868.

[30] J. Huang, Y. Zhou, and L. Guibas, "Manifoldplus: A robust and scalable watertight manifold surface generation method for triangle soups," *arXiv preprint arXiv:2005.11621*, 2020.

[31] J. Carvalho, A. T. Le, P. Jahr, Q. Sun, J. Urain, D. Koert, and J. Peters, "Grasp diffusion network: Learning grasp generators from

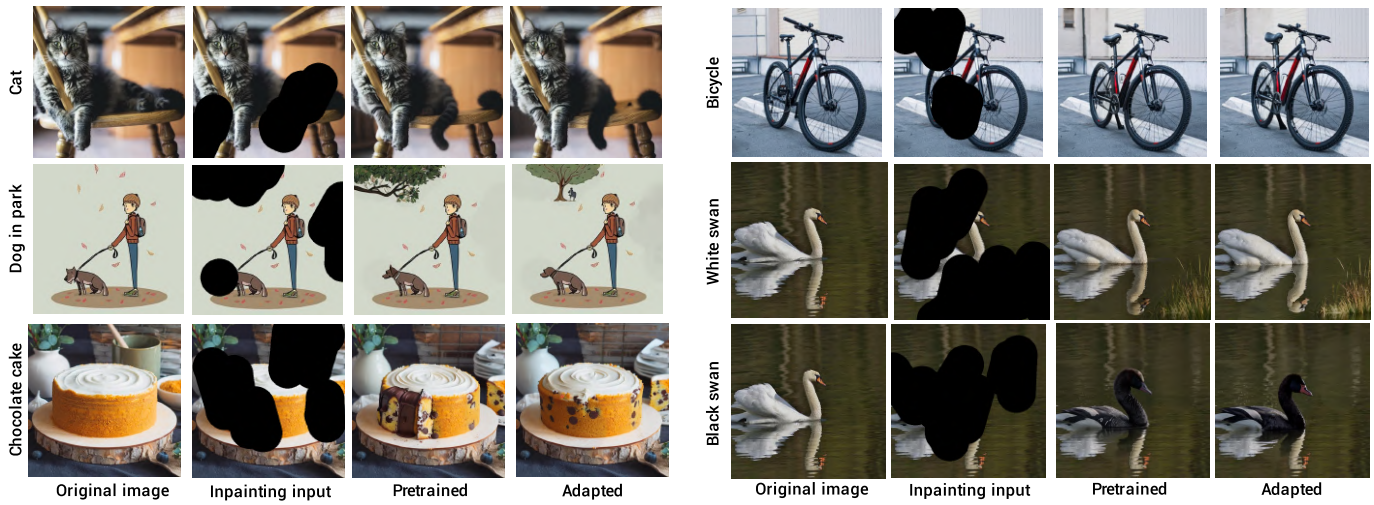

Fig. 24: Inpainting results on the Laion-Aesthetic dataset [41] for the pretrained and adapted inpainting model, showcasing generalizability.

partial point clouds with diffusion models in so (3) xr3," *arXiv preprint arXiv:2412.08398*, 2024.

[32] H. Zhang, S. Christen, Z. Fan, O. Hilliges, and J. Song, "Graspxl: Generating grasping motions for diverse objects at scale," in *European Conference on Computer Vision*. Springer, 2024, pp. 386–403.

[33] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, "Synergies between affordance and geometry: 6-dof grasp detection via implicit representations," *arXiv preprint arXiv:2104.01542*, 2021.

[34] X. Wei, M. Liu, Z. Ling, and H. Su, "Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–18, 2022.

[35] Y. Shao and C. Xiao, "Bimanual grasp synthesis for dexterous robot hands," *IEEE Robotics and Automation Letters*, 2024.

[36] D. Tochilkin, D. Pankratz, Z. Liu, Z. Huang, A. Letts, Y. Li, D. Liang, C. Laforte, V. Jampani, and Y.-P. Cao, "Triposr: Fast 3d object reconstruction from a single image," *arXiv preprint arXiv:2403.02151*, 2024.

[37] 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," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 21 469–21 480.

[38] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky, "Resolution-robust large mask inpainting with fourier convolutions," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2022, pp. 2149–2159.

[39] R. Shi, H. Chen, Z. Zhang, M. Liu, C. Xu, X. Wei, L. Chen, C. Zeng, and H. Su, "Zero123++: a single image to consistent multi-view diffusion base model," *arXiv preprint arXiv:2310.15110*, 2023.

[40] M. Liu, R. Shi, L. Chen, Z. Zhang, C. Xu, X. Wei, H. Chen, C. Zeng, J. Gu, and H. Su, "One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 072–10 083.

[41] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman *et al.*, "Laion-5b: An open large-scale dataset for training next generation image-text models," *Advances in neural information processing systems*, vol. 35, pp. 25 278–25 294, 2022.