Multi-Objective Photoreal Simulation (MOPS) Dataset for Computer Vision in Robotic Manipulation

Maximilian X. Li, Paul Mattes, Nils Blank, Korbinian F. Rudolf, Paul W. Lödige, Rudolf Lioutikov
Intuitive Robots Lab, Karlsruhe Institut for Technology, Germany
{maximilian.li, lioutikov}@kit.edu

Abstract

*Recent advancements in machine learning and computer vision have been driven by benchmark datasets targeting specific downstream tasks. However, computer vision datasets specifically designed for robotics—focusing on relevant scenes and prediction tasks for robot manipulation—are still lacking. We introduce the **M**ulti-**O**bjective **P**hotoreal **S**imulation (MOPS) dataset, a novel dataset that adresses this void by providing photorealistic simulated environments with comprehensive ground truth annotations. MOPS uses a zero-shot asset augmentation pipeline based on large language models to normalize and annotate 3D assets on a part level. MOPS provides pixel-level segmentations for various prediction tasks critical to robotics, including part segmentation and affordance prediction. By combining these detailed annotations with photorealistic simulation, MOPS is able to generate a vast number of diverse indoor scenes, potentially accelerating progress in robot perception, manipulation, and autonomous interaction with real-world environments. The dataset and generation framework will be made publically available.*

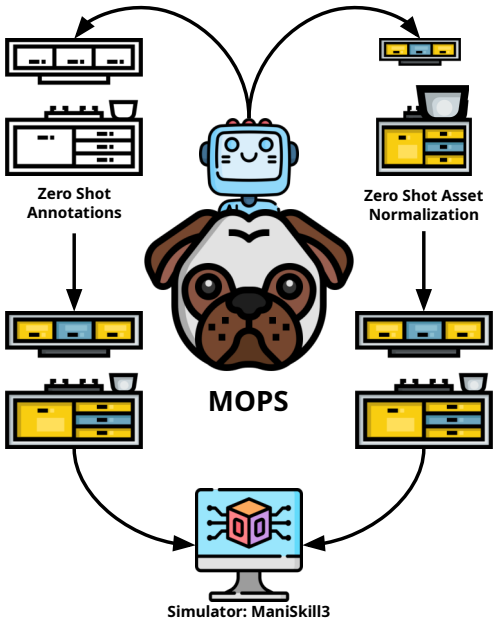


Figure 1. MOPS provides labled and realistic data for robot and vision tasks alike. Leveraging Large Language Models (LLMs) allows for zero-shot annotation of non-labeled 3D assets, as well as zero-shot normalization. The corrected and improved 3D assets are then used in the simulator to create new indoor scenarios and collect new data.

1. Introduction

Training and evaluating machine learning (ML) methods requires data specific to a given problem setting. In the vision domain, such tasks include pixel-wise affordance segmentation [19], 3D Part Segmentation [4], Scene Graph Generation (SGG) [35] or 6D pose estimation [32]. Although some datasets provide video sequences [2], the vast majority are focused on static scenes without dynamic scene interaction over time. Creating datasets and the respective annotations annotations for all of these tasks requires human effort for data collection and labeling. This limitation affects the scale and annotation detail of such datasets.

Nevertheless, these and prior datasets have fueled advancements in training ML models for many computer vision tasks across various domains in recent years. The robotics domain, however, is crucially underrespresented in these datasets. While embodied agents require robust perception of their environments to operate effictively and autonomously, only few computer vision datasets adress the particularities of robot manipulation. Datasets for learning vision for robotic manipulation should ideally fulfill a suspect of the following requirements:

REQ OBJ: Manipulation relevant *Objects*: The datasets should contain objects relevant to robotic manipulation, such as household objects commonly found in living spaces.

REQ ANN: Manipulation relevant *Annotations*: Ad-

Dataset	*Objects*	*Annotations*	*Representations*	*Environment*	*Interaction*	*Robot Trajectories*
Vision Datasets						
CUB-200-2011 [29]		P*	R			
CityScapes [6]		S+I	R			
SemanticKITTI [2]		S+I	R+D+M			
Visual Genome [14]		A	R+G			
PSG [33]		S+A	R+S+G			
ScanNet++ [34]	✓	S+I	R+D+M	✓		
HyperSim [23]	✓	S+I	R+D+M	✓		
RGB-D Part Aff. [19]	✓	A	R+D			
3D AffordanceNet [7]	✓	A	M			
PartNet-Mobility [31]	✓	P	M			
Robotics Datasets						
Open-X [22]	✓		R+D	✓		✓
DROID [11]	✓		R+D	✓		✓
AI2-THOR [13]	✓	S+I	R+D+M	✓	✓	
OmniGibson [15]	✓	S+I	R+D+M	✓	✓	
RoboCasa [20]	✓	S+I	R+D+M	✓	✓	✓
MOPS (Ours)	✓	**S+I+P+A**	**R+D+M+G**	✓	✓	**(✓)**

Note: S: Semantic Segmentation, I: Instance Segmentation, P: Part Segmentation, P*: Part Center Points, A: Affordance Segmentation, R: RGB, D: Depth, M: 3D Meshes or Pointclouds, G: Scene Graphs

Table 1. Comparison of different computer vision and robotics datasets and their relevance to robot manipulation. Please note: MOPS is compatible with demonstrations by RoboCasa, but does not provide new robot trajectories.

ditionally, datasets should include annotations relevant to robotic manipulation, such as *part information*, *affordance labels* or *6D poses*, and ideally in high resolution (pixel-wise).

REQ REP: Manipulation relevant *Representations*: In addition to image observations, other scene representations such as pointclouds or scene graphs are of relevance to robotic manipulation. Such scene graphs can contain information about objects, their affordances and relations to the robot and each other.

REQ ENV: Manipulation relevant and realistic *Environments*: The obervations should be obtained from realistically environments relevant to robot manipulation, such as indoor scenes. The realism refers hereby to both the image quality, i.e., photorealistic rendering, and real setups with clutter and distractors, as opposed to labroratory setups.

REQ INT: Manipulation relevant *Interactions*: The dataset should provide relevant agent-agent, agent-object and object-object interactions. Hereby, an agent refers to both a robotic agent or a human collaborator. While videos can capture these interactions over time, ideally the robotic agent could also directly evaluate its learned behavior, for instance for active perception.

Our newly introduced MOPS dataset addresses all of these requirements. In contrast exiting vision datasets fullfill merely a subset. Human-centric video datasets [25] provide dynamic videos of human-object interaction in realistic environments, but are not suitable to train a robot manipulation policy. Realistic and dynamic environments and temporal continutity are often found in human-centric video datasets, which are unfit for training a robot manipulation policy. Datasets built around task relevant objects such as affordance detection [19] or 6D pose estimation [32], are limited in the realism of the captured environments and do not provide any scene dynamics.

On the other hand, robot datasets [11, 22] are built around behavior, showing dynamic and realistic scenes. However, if any ground truth labels for the observations are provided, their quality does not fullfill the same standards as for vision-based datasets, requiring either manual labeling or sophisticated post-hoc annotation pipelines like NILS [3].

We propose a new dataset generation framework with pixel-level ground truth for **M**ulti-**O**bjective **P**hotoreal **S**imulation (MOPS). The MOPS dataset generator bridges the gap between interactive robotics datasets for learning behavior and high-quality pixel-level annotations from vision datasets. This makes the MOPS framework poten-

tially interesting to both the vision and robotics community and enables learning computer vision for robotic manipulation at scale. MOPS uses assets from PartNet-Mobility [31] and RoboCasa [20] to show scenes with articulated objects for manipulation (REQ OBJ) in photorealistically rendered scenes (REQ ENV). MOPS leverages a zero-shot asset augmentation pipeline built on GPT-4o [21] to normalize assets, and to provide manipulation relevant annotations such as affordances (REQ ANN). Thus, MOPS can provide pixel-level ground truth masks for relevant labels such as class segmentations, part segmentations, instance segmentations and affordance labels. MOPS can provide relevant geometric information such as normal maps and 6D poses, provide different sensor modalities such as RGB-D or pointclouds and generate scene graphs with an LLM (REQ REP) on demand. MOPS uses the Maniskill3 [26] simulator to enable dynamic and interactive scenes, ready for evaluation of learned robot behavior or recording demonstrations via teleoperation (REQ INT).

2. Related Work

Vision Datasets: In order to develop and evaluate new models for common computer vision tasks, the computer vision community has proposed several specialized datasets. For example, CUB-200-2011 [29] is a widely used image dataset for fine-grained image classification. The RGB images in this dataset depict 200 different bird species with point-based annotations for body parts and attributes like head color or wing shape. As such, it is widely used for evaluating image classification models, including interpretable approaches like prototypical part networks [5] or concept bottleneck models [12]. For semantic segmentation, several datasets have been proposed with a focus on autonomous driving, such as Cityscapes [6] or SemanticKITTI [2]. In Scene Graph Generation (SGG) datasets such as Visual Genome (VG) [14] are used, which focus on the collection of many data samples, but less on the data quality. A more clean and segmentation oriented SGG dataset is Panoptic Scene Graphs (PSG) [33]. Although these datasets are ideal to train and test novel models specialized on the specific task, the dataset content is less relevant to robot manipulation (REQ OBJ).

ScanNet++ [34] is a dataset of RGB-D voxels of indoor scenes and provides ground truth semantic and instance segmentation annotations. Hypersim [23] provides photorealisticly rendered indoor scenes created by 3D artists, also with semantic and instance segmentations. Such indoor environments are ideal for training household robot agents (REQ ENV). However, they only provide class and part segmentations and lack other interesting ground truth annotations such as 6D poses or affordances (REQ ANN).

The RGB-D Part Affordance dataset [19] consists of single object RGB-D images and three cluttered scenes with affordance annotations (REQ ANN) in front of a uniformly blue background. This object-centric focus can be used to train a specialized affordance detection stage after an object detector with bounding box crops. However, it is limited in the realism of the depicted scenes with regard to background and amount of clutter (REQ ENV). Like most pure vision datasets, they all lack interactivity (REQ INT). Our MOPS dataset aims to adress all these issues by providing procedural, synthetic scenes of cluttered indoor environments with multiple, pixel-wise ground truth annotations. This requires access to high-quality rendering assets.

3D AffordanceNet [7] is built on the 3D object mesh dataset ShapeNet [4]. It provides nearly 23k 3D point clouds across 23 object categories and 18 affordance labels (REQ OBJ & REQ ANN). However, 3D AffordanceNet lacks material information crucial to photorealistic rendering (REQ ENV). PartNet-Mobility [31] provides a subset of ShapeNet objects with material information and modeled articulation (REQ ENV), so that it could be used in a physics simulator (REQ INT). However, PartNet-Mobility does not provide affordances (REQ ANN). Like most 3D asset libraries, PartNet-Mobility models are not modeled to a common reference scale. On direct import, they appear larger than the robotic arm (see Figure 2).

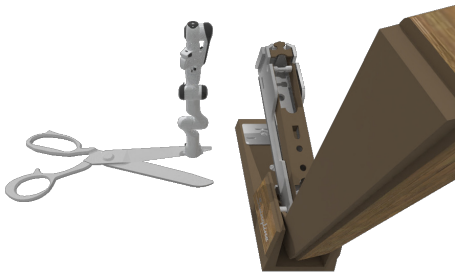


Figure 2. 3D assets such as PartNet-Mobility are often not modeled to the same reference scale. MOPS normalizes all objects to realistic size relative to the robot according to scale ranges provided by an LLM.

We propose a zero-shot augmentation pipeline based on an LLM to provide affordance annotations for each the used 3D assets. Additionnally, the augmentation pipeline provides realistic scale ranges such that 1.0 simulation units equal 1.0 meters for all objects. By using a full physics simulator based on ManiSkill3 [26] to generate the MOPS dataset, it offers full interactivity for evaluating robot behavior, which is impossible on real-world RGB images or videos.

Robotics Datasets: Datasets for robotics are typically real-world or simulated datasets. Each sample is a pair of sensor inputs and robot movement trajectories. The Open X-Embodiment dataset [22] is one of the largest and

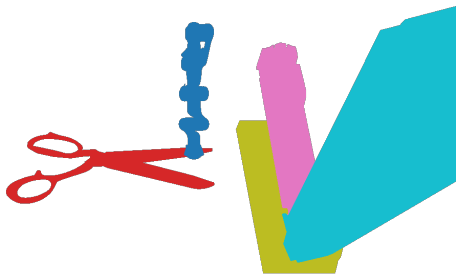


Figure 3. MOPS queries an Large Language Model (LLM) to provide a list of affordances for 3D assets on a part and object level. For ease of visibility, we show affordance masks for the unscaled assets.

most diverse dataset collections, consisting of more than 70 individual datasets across different labs. This, however, leads to issues regarding consistency and data quality (REQ ANN. The DROID dataset [11] provides a standardized data collection platform, resulting in more consistent data. However, increasing real-robot datasets to a scale of typical vision/language datasets remains expensive due to the required time in collecting robot trajectories and obtaining ground truth annotations, even when aided by zero-shot labeling frameworks such as NILS [3].

On the simulation side, frameworks like AI2-THOR [13], OmniGibson [15] and RoboCasa [20] provide complex room-scale scenes with photorealistic rendering capabilities (REQ ENV & REQ REP). AI2-THOR offers a large variety of procedurally generated scenes [13]. OmniGibson employs a high-quality physics simulation including deformable objects, fluids and state machines, which enables it to simulate, for example, cooking tasks [15]. Robocasa provides over 100k training trajectories [20] obtained via human demonstrations and generation via MimicGen [16]. Newly proposed generative simulation frameworks such as RoboGen [30] and Genesis [1] offer a near unlimited variety of scenes by employing generative AI. However, they work best for task prompts without cluttered scenes, reducing the effectiveness of generated environments for (robotic) vision applications (REQ ENV).

The synthetic nature of simulated datasets provide easy access to ground truth annotations for vision tasks like depth estimation, 6D pose estimation or instance segmentation (REQ ANN). Additionally, they are able to easily provide additional representations such as pointclouds (REQ REP). Nevertheless, even object classes or categories required for object detection, semantic segmentation or scene graph generation are often difficult to obtain. Finally, to the best of our knowledge, no simulation framework provides pixel-wise ground truth annotations for semantic concepts, affordances, or scene graphs out of the box.

Our MOPS dataset adresses the shortcomings in environment realism, ground truth annotations and representations by combining the high scene variety offered by Robocasa [20] with zero-shot augmented assets imported from PartNet-Mobility [31]. Thus, MOPS can generate a virtually unlimited number of realistic scenes with pixel-wise ground truth annotations, including affordances, in cluttered environments and across represenations, including scene graphs.

The existing simulation datasets rely on CPU based physic simulators such as MuJoCo [27] or Unity [28]. By implementing the MOPS dataset generation pipeline with the ManiSkill3 simulator [26], we improve visual quality and generation speed due to raytracing and GPU parallelization features, similar to Isaac Lab [17]. Hence, the MOPS dataset represents an efficient, valuable and promising asset for the robot learning and computer vision community. Table 1 provides a concise overview of current computer vision and robotics datasets and their compliance with respect to the manipulation relevant dataset requirements.

3. Zero Shot 3D Asset Augmentation

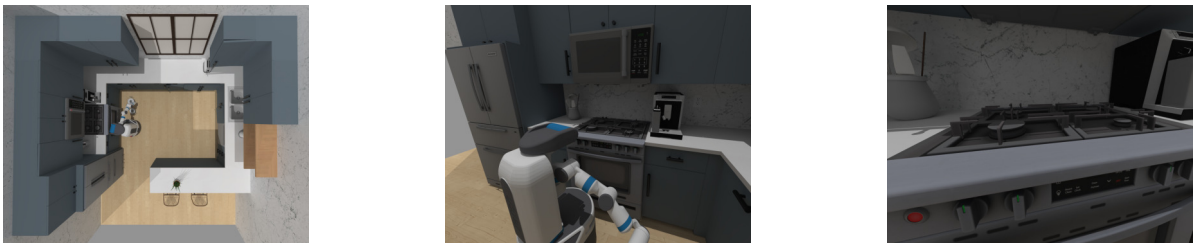
MOPS creates photorealisticly rendered scenes of indoor environments for robot manipulation (REQ ENV) by using the ManiSkill3 [26] simulator and renderer. To populate the scenes with realistic and interactive objects (REQ OBJ) MOPS uses 3D assets provided by RoboCasas [20] and PartNet-Mobility [31]. However, the 3D assets must first be augmented for use in MOPS. Firstly, 3D objects are not modeled on the same reference scale, breaking realism (REQ OBJ & REQ ENV). Secondly, the 3D assets must be annotated with affordance annotations in order to easily provide pixel-wise ground truth masks (REQ ANN). In order to adress these issues, MOPS employs a zero-shot augmentation pipeline based on GPT-4o [21].

3.1. Zero-Shot Asset Normalization (REQ OBJ)

MOPS uses 3D assets of common household objects relevant to robot manipulation. However, most 3D assets, including PartNet-Mobility, are not modeled on the same reference scale. This results in unrealistic relative sizes between different assets or in relation to a simulated robot. We propose a zero-shot asset normalizaton stage to adress this issue.

We first verify the internal, spatial scale of the simulation by comparing the Denavit-Hartenberg parameters of a Franka Emika Panda 7-DoF arm [8] with the endeffector coordinates in simulation. This confirms a simulation scale of 1.0 simulation units equaling 1.0 meters.

Each asset in PartNet-Mobility is annotated with XYZ-Bounding boxes and an object category, e.g., *Remote*, *Microwave*, *Coffemachine*. Through the common-sense knowledge embedded in recent Large Language Models (LLMs), MOPS can obtain realistic object scales. In par-



(a) Birdseye Camera (b) External Camera (c) Ego Camera

Figure 4. MOPS provides several camera views to cover a wide range of robot learning tasks, from mobile navigation to manipulation.

ticular, GPT-4o is queried to output standard dimensions for each object category. MOPS queries the LLM for realistic minimum and maximum Width $\times$ Height $\times$ Depth (WHD) sizes for each object category. As the orientation of each asset is unknown, the calculation process divides the largest bounding box dimension with the largest WHD dimension to obtain a realistic scaling factor. When loading an asset into the simulation, a random sample from a uniform distribution is used to scale between the minimum and maximum realistic range to further increase the variety in simulated objects. Figure 2 presents an example scene before scale normalization. This approach enables easy addition of new object categories and models into the dataset, even from other datasets.

3.2. Zero-Shot Affordance Annotation (REQ ANN)

MOPS integrates the PartNet-Mobility dataset, a collection of articulated, part-based 3D assets. It is a subset of the PartNet asset dataset [18], which in turn is a subset of ShapeNet [4]. The fully articulated PartNet-Mobility assets provide a wide range of interactions (REQ INT), but are missing manipulation relevant annotations such as affordances (REQ ANN).

MOPS again leverages the common-sense reasoning capabilities of LLMs to generate multi-label affordances for each object on a part and object level. Particularly, GPT-4o is queried to output a list of affordances for each object and object part. For 3D assets without part annotations, e.g., the RoboCasa assets, the zero-shot annotation stage labels the entire object with all applicable affordances.

Finally, MOPS clusters the affordances with a sentence embedding model to filter duplicates such as *closable* and *close* and aligns semantic clusters like *heatable* and *warmup-able*.

Figure 3 illustrates the multi-label part affordance masks for a simple scene with two PartNet-Mobility assets. Future work could extend the annotation stage to produce region-level affordance annotations similar to the manually labeled 3D AffordanceNet[7]. Meanwhile, the already presetented zero-shot annotation stage alleviates the human-labeling effort signficantly.

4. MOPS Dataset Generation

MOPS can generate a virtually unlimited number of simulation scenes by combining 120 realistic indoor environments from RoboCasa and its assets with 2,300 articulated objects provided by PartNet-Mobility and augmentations from a zero-shot asset augmentation pipeline. To increase the relevance for robot manipulation, MOPS provides the following technical enhancements.

4.1. MOPS Ground Truth Masks (REQ ANN)

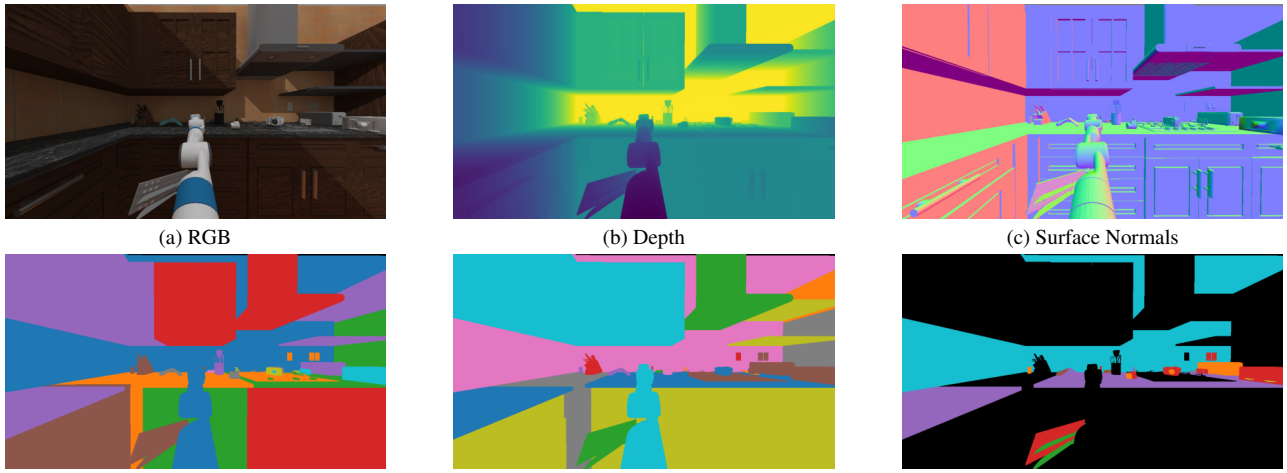
In order to provide a good dataset for learning computer vision tasks relevant to robot manipulation, MOPS provides multiple scene representations. This begins with the position of cameras. MOPS provides several camera views to facilitate training for a range of tasks. A birdseye view can provide ground truth for SLAM reconstruction tasks in mobile manipulation. External over-the-shoulder, ego, and in-hand cameras mimic typically used camera setups in robot manipulation. Figure 4 presents raytraced RGB images for a RoboCasa scene with birdseye, external and ego view for an empty RoboCasa scene.

MOPS cameras provide commonly used image modalities. The RGB renderings of the simulated scene can use raytracing for enhanced realism, or employ rasterized rendering for faster computation. The simulation provides depth images with distance to the camera in milimeters, providing 2.5D RGB-D inputs or ground truth data for learning depth estimation. Additionally, the underlying simulation provides ground truth surface normal maps and part level segmentation masks for the simulated assets.

MOPS can provide additional ground truth modalities from the rendering, such as 6D poses for objects in the camera frame, or instance and semantic segmentation masks. These masks are generated via look-up tables MOPS populates during scene creation and provides pixel-perfect ground truth data.

By obtaining affordance labels from its zero-shot asset augmentation pipeline, MOPS also provides pixel-perfect affordance annotations for each camera.

Figure 5 presents different, visual camera modalities from an ego camera in a sparsely populated RoboCasa



(a) RGB (b) Depth (c) Surface Normals

(d) Part Segmentation (e) Class Segmentation (f) Multilabel Affordances

Figure 5. MOPS provides multiple, pixelwise ground truth maps in addition to RGB-D perception.

scene. The RGB modality is rendered via raytracing, resulting in realistic shadows, reflections and even light transmission for correctly modelled glasware assets. The depth modality shows the distance of each pixel to the virtual camera. The surface normal modality encodes the XYZ components of each normal in the RGB values of the image. For the part segmentation, each pixel stores the simulation internal part ID, whereas the class segmentation performs a table lookup to retrieve the class for each part ID. Similarly, for instance segmentation, a different lookup-table storing object instances is used. The difference between these maps is most notable for articulated objects, like the cabinet door or robot arm, where each articulation link is a seperate part, but share the same class. Finally, the affordance annotation map returns a binary list for each pixel and affordance. It indicates, wheter this affordance is present, as affordances are modeled in MOPS as a multi-label annotation.

4.1.1. Extended Representations (REQ REP)

MOPS can query an LLM to generate scene graphs on demand, based on the pixel-wise affordance annotations and object class and instances.

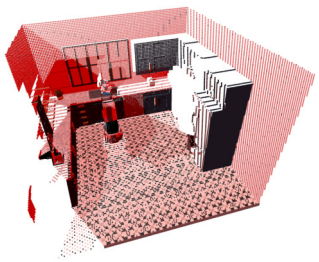


Figure 6. Pointcloud of a RoboCasa scene with red ambient lighting.

The underlying ManiSkill3 simulation also supports point cloud generation by merging the 2.5D images of all cameras. The observations can be further customized by changing the extrinsic and intrinsic parameters of the virtual cameras or the lighting configuration. Figure 6 shows a scene point-cloud with red ambient lighting.

The SAPIEN renderer used by ManiSkill3 provides realistically simulated stereo depth sensors with active IR lighting, including simulated sensor noise [24]. Unfortunately, the implementation of these simulated stereo depth sensors is not yet fully available in the in the newest release of ManiSkill3[1], and, hence can not be shown in the acmops version presented here. However, inlcuding them once they are available is straightforward, based on the experience with ManiSkill2.

4.2. MOPS Realistic Environments (REQ ENV)

The goal of MOPS is to generate a vision dataset of realistic environments relevant to robot manipulation. The high visual quality of the used RoboCasa assets aids in generating realistic indoor kitchen scenes. Additionally, the high-quality raytracing implementation used by ManiSkill3 results in photorealistic lighting and rendering.

In order to create realistically cluttered scenes with potential object overlap and distractors, MOPS procedurally places augmented PartNet-Mobility asssets in the RoboCasa kitchen scenes. First, MOPS obtains the location of the kitchen countertops from the simulation. Then, it computes the available space by observing the bounding box of the collision mesh. Finally, it generates random positions within the available space and drops simulation objects. While this is a rather heuristic approach compared

[1]ManiSkill v3.0.0b19, retrieved 2025-03-07 from GitHub, Commit *91e1396*

to trained models such as ClutterGen [9], it can be easily applied to novel scenes and environments outside of RoboCasa, without requiring any training. Figure 7 shows one especially messy example. More scenes are presented in the Appendix.

4.3. Interactive Simulation (REQ INT)

MOPS leverages the ManiSkill3 simulation to provide full interactivity. Robot trajectories such as the RoboCasa demonstrations can be rolled out to record observations in dynamic environments.

Additionally, new demonstrations can be recorded using an teleoperation interface. Thus, MOPS supports teaching new robot behavior, e.g., for active perception. Figure 8 illustrates two steps in a robot trajectories recorded using a mouse-and-keyboard teleoperation interface provided by ManiSkill3. In this demonstration, the robot explores object articulations in a cluttered tabletop scene by depressing the hinged top of a stapler. The recorded robot trajectory can be replayed to record the ground truth segmentations provided by MOPS.

5. Affordance Analysis

The zero-shot annotation pipeline produces **24** part level annotations for the **23,048** 3D parts of **2,346** individual objects across **46** object categories offered by PartNet-Mobility [31] , making MOPS the most diverse dataset for affordance detection for both images and 3D meshes. Table 2 presents an overview of labeled affordances and the number of labeled parts per affordance.

For the used RoboCasa assets, MOPS provides object-level affordance annotations. As these assets do not provide part informations, the affordance are applied to the entire object, even if actions such as *grasp* apply to specific areas on the object. The Zero-Shot Annotation pipeline discovered **44** affordances for **1008** objects across **101** categories. Table 3 presents the number of objects for each affordance.

Figure 7. A RoboCasa kitchen filled with PartNet-Mobility clutter.

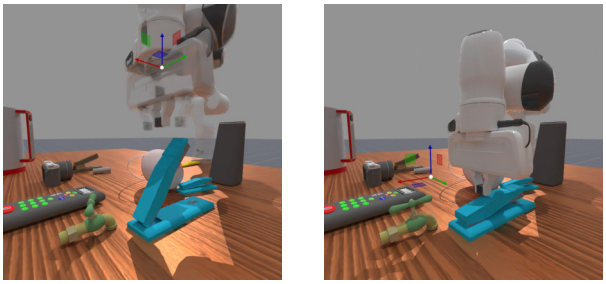


Figure 8. MOPS uses the mouse-and-keyboard teleoparation interface by ManiSkill3. The semitransparent robot arm indicates to the user the new target position.

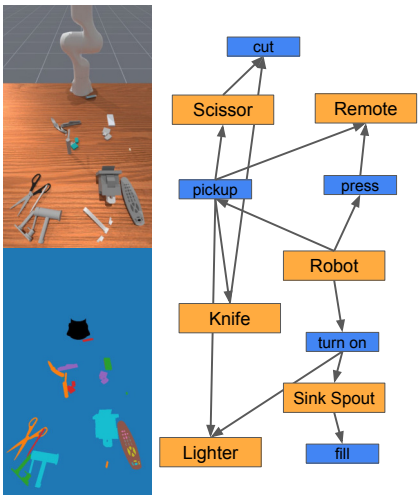


Figure 9. Sub-graph of a possible Scene Graph (SG) generated using the affordance information provided by MOPS.

Although 3D-AffordanceNet provides more individual samples, MOPS provides one of the most varied affordance learning datasets with regard to object categories, number of affordances and sensor modalities, all embedded in realistic and cluttered scenes.

5.1. Scene Graph Generation

MOPS also includes Scene Graph (SG) information, which can be used for Scene Graph Generation (SGG). Compared to traditional SGG datasets [14], MOPS provides high quality SGs with non-redundant relation annotations. The VG dataset includes more than 40k relation classes, resulting in an high amount of duplicates. MOPS removes duplicate relations by the clustering post-processing step in the zero-shot affordance annotation stage. Furthermore, the relations in VG are dominated by frequent occuring classes, like *next to*, *has* or *beside* [35]. MOPS solves these issues through information rich relation classes based on the detected affordances. MOPS generates dual-purpose SGs through its

Table 2. Part-Level Affordances for PartNet-Mobility dataset

Affordance	*Count*	*Affordance*	*Count*	*Affordance*	*Count*	*Affordance*	*Count*	*Affordance*	*Count*
close	**106**	grasp	**1897**	move	**1996**	pull	**1647**	sittable	**128**
connect	**11**	insert	**51**	openable	**2079**	pushable	**1638**	stab	**105**
contain	**1650**	layable	**1381**	pourable	**171**	remove	**51**	support	**1907**
cut	**208**	lift	**245**	press	**6658**	rotate	**694**	turn	**7**
display	**189**	listen	**2**	foldable	**32**	wear	**195**		

Table 3. Object Level Affordances for RoboCasa

Affordance	*Count*	*Affordance*	*Count*	*Affordance*	*Count*	*Affordance*	*Count*	*Affordance*	*Count*
adjustable	**15**	contain	**99**	grasp	**994**	mixable	**64**	recyclable	**98**
breakable	**68**	cookable	**213**	hold	**565**	mountable	**16**	roastable	**98**
close	**33**	coolable	**56**	juicy	**122**	move	**376**	rotate	**37**
connect	**2**	cuttable	**575**	lift	**177**	openable	**150**	scoopable	**25**
consumable	**110**	decorative	**44**	lockable	**4**	organize	**47**	sealable	**116**
display	**44**	drinkable	**157**	mashable	**61**	peelable	**181**	shareable	**73**
edible	**483**	fillable	**188**	placeable	**75**	pointable	**39**	spreadable	**88**
press	**10**	pourable	**252**	stackable	**225**	store	**105**	squeezable	**127**
support	**85**	throwable	**100**	toastable	**50**	washable	**220**		

affordance modeling: part-level affordances create robot-centric SGs for manipulation tasks, while object-level affordances produce vision-centric SGs for scene understanding.

Object nodes in a SG are connected if the detected affordances are connected on a logical basis. For example, if an object is cut-able and the scene includes a knife, the object and the knife will be connected in the SG. These logical connection can either be hand-crafted or queried through an LLM.

An example sub-graph of a SG can be seen in Figure 9. In contrast to the VG dataset, MOPS does not suffer from the long-tailed distribution problem [35], because relations and their occurence can be filtered.

6. Conclusion

We introduce MOPS, a dataset generation pipeline for learning computer vision for robotics. MOPS provides photorealistically rendered samples of common household objects relevant to robot manipulation with mulitple, pixel level ground truth annotations. Hereby, the ground truth annotations focus on tasks relevant to computer vision and robotics, such as class, part and instance segmentation, affordance labels and 6D poses. By using assets from PartNet-Mobility and RoboCasa, MOPS generates cluttered object ensembles and realistic indoor kitchen scenes. The employed ManiSkill3 simulator provides full interactivity for recording, learning and evaluating robot behavior.

MOPS uses a zero-shot asset augmentation pipeline by leveraging GPT-4o to generate one of the most diverse affordance learning datasets with over 40 labels across 100+ object categories. This augmentation pipeline can be quickly applied to new asset libraries or novel semantic labels for even more variety.

Limitations. Although MOPS leverages many advantages provided by the underlying ManiSkill3 simulator, MOPS is also limited by the simulators functionality. For example, the raytraced rendering is not usable in conjunction with GPU parallelizaton, limiting the maximum performance. Future work could adapt MOPS to other simulators such as Isaac Lab [17] or Genesis [1] for enhanced rendering and parallelization capabilities.

Future Work. MOPS relies on 3D assets with realistic material information and textures. Future work could explore extending the presented zero-shot asset augmentation pipeline to also generate materials and textures via GenAI. This would enable MOPS to import the entire PartNet [18] and ShapeNet [4] datasets.

Lastly, MOPS is fully compatible to robot demonstrations provided by RoboCasa. However, MOPS currently does not provide new demonstrations showing robot-object interactons. A future version will explore teleoperation interfaces such as IRIS [10] for recording new robot trajectories in AR/VR.

References

[1] Genesis Authors. Genesis: A universal and generative physics engine for robotics and beyond, 2024. 4, 8

[2] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019. 1, 2, 3

[3] Nils Blank, Moritz Reuss, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Wenzel, Oier Mees, and Rudolf Lioutikov. Scaling robot policy learning via zero-shot labeling with foundation models. In *8th Annual Conference on Robot Learning*, 2024. 2, 4

[4] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1, 3, 5, 8

[5] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K. Su. This looks like that: Deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019. 3

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3

[7] Shengheng Deng, Xun Xu, Chaozheng Wu, Ke Chen, and Kui Jia. 3d affordancenet: A benchmark for visual object affordance understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1778–1787, 2021. 2, 3, 5

[8] Franka Robotics. Franka Denavit–Hartenberg parameters. Franka FCI Documentation. 4

[9] Yinsen Jia and Boyuan Chen. Cluttergen: A cluttered scene generator for robot learning. In *8th Annual Conference on Robot Learning*, 2024. 7

[10] Xinkai Jiang, Qihao Yuan, Enes Ulas Dincer, Hongyi Zhou, Ge Li, Xueyin Li, Julius Haag, Nicolas Schreiber, Kailai Li, Gerhard Neumann, and Rudolf Lioutikov. Iris: An immersive robot interaction system. *arXiv preprint arXiv:2502.03297*, 2025. 8

[11] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset, 2024. 2, 4

[12] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020. 3

[13] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. 2, 4

[14] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017. 2, 3, 7

[15] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pages 80–93. PMLR, 2023. 2, 4

[16] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *Conference on Robot Learning*, pages 1820–1864. PMLR, 2023. 4

[17] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6): 3740–3747, 2023. 4, 8

[18] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5, 8

[19] Austin Myers, Ching L. Teo, Cornelia Fermüller, and Yiannis Aloimonos. Affordance detection of tool parts from geometric features. In *ICRA*, 2015. 1, 2, 3, 11

[20] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems*, 2024. 2, 3, 4

[21] OpenAI. Gpt-4o system card, 2024. 3, 4

[22] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024. 2, 3

[23] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *International Conference on Computer Vision (ICCV) 2021*, 2021. 2, 3

[24] Sapien Developers. Sapien 3.0 documentation, 2024. Sapien StereoDepthSensor Documentation. 6

[25] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild, 2012. 2

[26] Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu,

Tse-kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 2024. 3, 4

[27] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. 4

[28] Unity Technologies. Unity, 2023. Game development platform. 4

[29] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Caltech-ucsd birds-200-2011. Technical Report CNS-TR-2011-001, 2011. 2, 3

[30] Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation, 2023. 4

[31] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3, 4, 7

[32] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. 1, 2

[33] Jingkang Yang, Yi Zhe Ang, Zujin Guo, Kaiyang Zhou, Wayne Zhang, and Ziwei Liu. Panoptic scene graph generation, 2022. 2, 3

[34] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 2, 3

[35] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5831–5840, 2018. 1, 7, 8

Multi-Objective Photoreal Simulation (MOPS) Dataset for Computer Vision in Robotic Manipulation

Supplementary Material

7. Example Single Object Images

Figure 10 shows example observations from the single object configuration, which mimics single object datasets with uniform backgrounds like RGB-D Part Affordance [19].

8. Tabletop Clutter Images

Figure 11 shows example observations from randomly generated, cluttered tabletop scenes. These images show interactive scenes including the robot base, ready for learning robot behavior. For a purely vision-based dataset, the environment geometry with table, floor, robot and background could be easily disabled.

9. Example Cluttered Kitchen Images

Figure 12 shows example observations from the cluttered RoboCasa kitchen configuration.

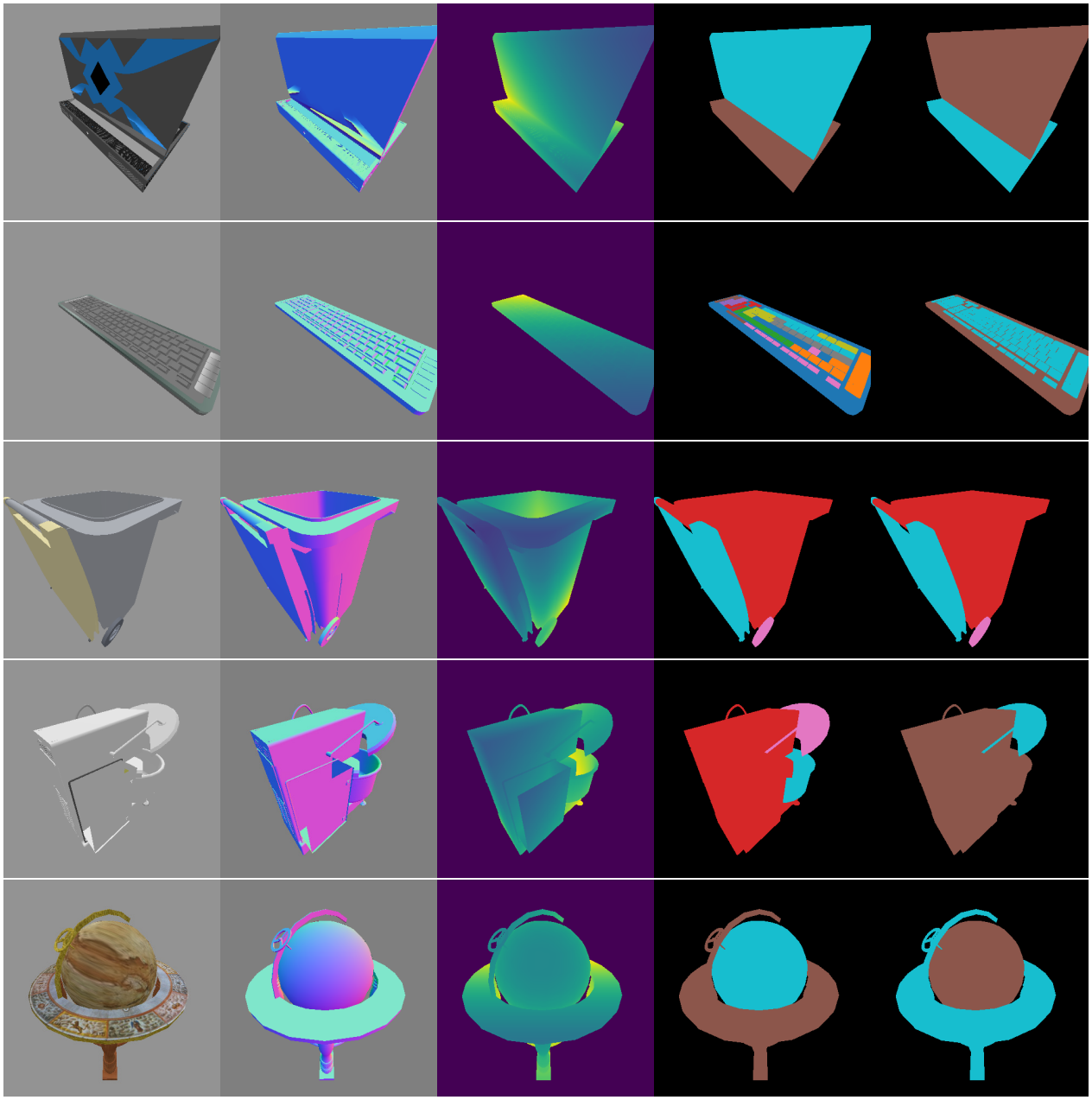


Figure 10. Single Object images. From Left to Right: RGB Image, Normal Map, Depth Image, Part Segmentation, Affordance Segmentation. Please note that the colors for Affordance Segmentation visualizaton only provide contrast and do not share meaning across images.

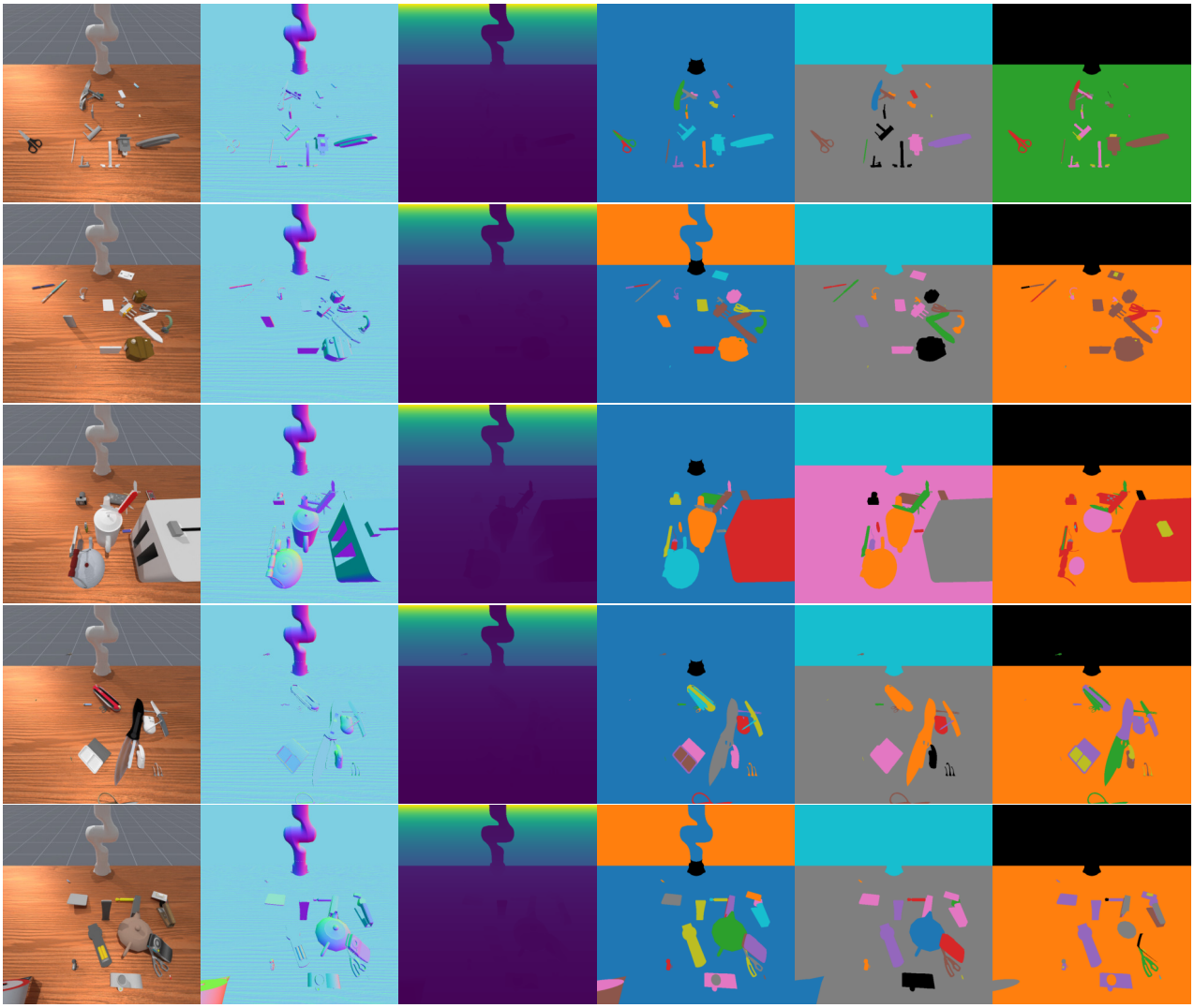


Figure 11. Cluttered Tabletop. From Left to Right: RGB Image, Normal Map, Depth Image, Part Segmentation, Class Segmentation, Affordance Segmentation. Please note that the colors for segmentation visualizations only provide contrast and do not share meaning across images. The depth images lack visual detail in this illustration, due to the floor in the background going towards infinity.

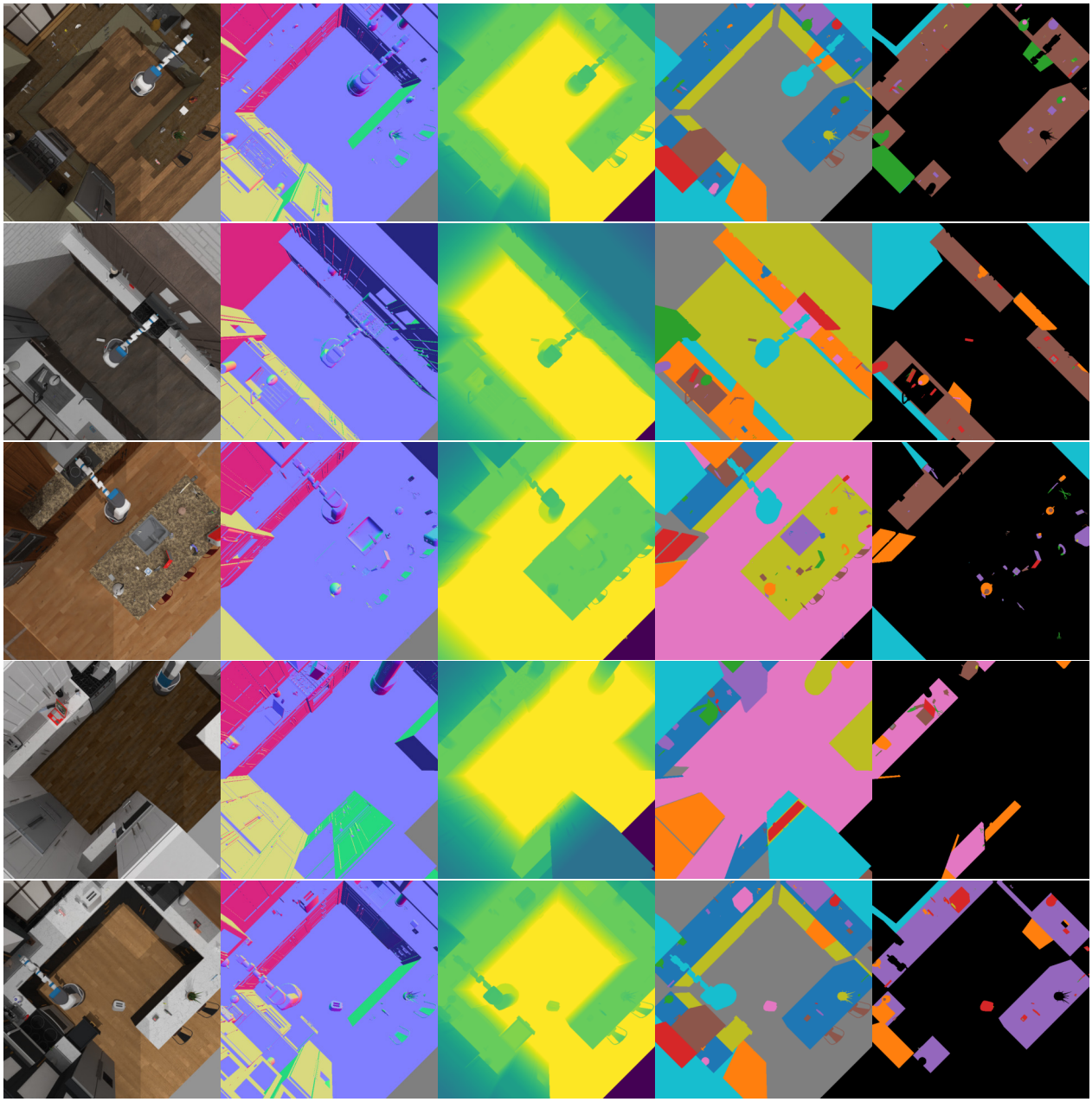


Figure 12. RoboCasa Kitchens with clutter. From Left to Right: RGB Image, Normal Map, Depth Image, Class Segmentation, Affordance Segmentation. Please note that the colors for Affordance Segmentation visualizaton only provide contrast and do not share meaning across images.