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

Maximilian Xiling Li [1] **Paul Mattes** [1] **Nils Blank** [1] **Rudolf Lioutikov** [1]

Abstract

Datasets bridging computer vision and robotics by providing high-quality visual annotations in manipulation-relevant scenes remain limited. This work introduces the Multi-Object Photoreal Simulation (MOPS) dataset, which provides comprehensive ground truth annotations for photorealistic simulated environments. MOPS employs a zero-shot asset augmentation pipeline based on Large Language Models (LLM) to automatically normalize 3D object scale and generate part-level affordances. The dataset features pixel-level segmentations for tasks crucial to robotic perception, including fine-grained part segmentation and affordance prediction (e.g., "graspable" or "pushable"). By combining detailed annotations with photorealistic simulation, MOPS generates a vast, diverse collection of scenes to accelerate progress in robot perception and manipulation. We validate MOPS through vision and robot learning benchmarks. The dataset and generation framework will be made publicly available.

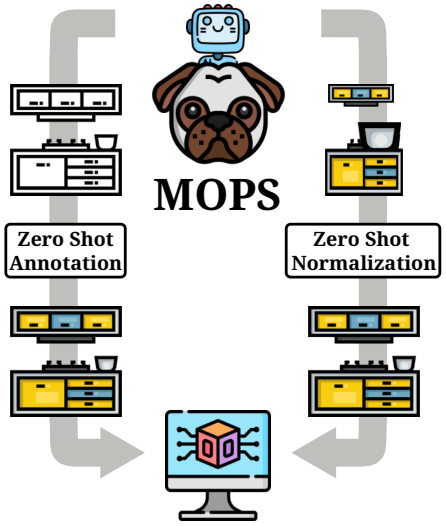


Figure 1. MOPS provides labeled, realistic data for robotics and vision tasks through Large Language Models (LLMs)-enabled zero-shot annotation and normalization of 3D assets, which are then used to create new indoor scenarios for data collection.

1. Introduction

Machine learning methods in computer vision rely on task-specific datasets for training and evaluation, spanning applications from affordance segmentation (Myers et al., 2015) and 3D part segmentation (Chang et al., 2015) to scene graph generation (Zellers et al., 2018) and 6D pose estimation (Xiang et al., 2017). While some datasets include video sequences (Behley et al., 2019), the majority focus on static scenes without dynamic interaction over time. Creating comprehensive datasets with detailed annotations requires substantial human effort for data collection and labeling, limiting both scale and annotation granularity.

Despite these datasets driving significant advancements across computer vision domains, the robotics domain remains underrepresented. Embodied agents require robust environmental perception for effective autonomous operation, including the ability to identify actionable object parts and manipulation opportunities. Yet, few computer vision datasets address the requirements of robot manipulation. While MOPS does not directly train manipulation policies, it provides the visual perception foundation necessary for such systems, serving both the computer vision community (through rich annotations) and robotics community (through interactive simulation).

Datasets for learning computer vision for robot manipulation should ideally fulfill several key requirements:

1. **Manipulation-relevant Objects** (REQ OBJ): Common household items found in living spaces.

2. **Manipulation-relevant Annotations** (REQ ANN): High-resolution labels including part information, visual affordance labels, and 6D poses.

[1]Intuitive Robots Lab, Karlsruhe Institute for Technology, Karlsruhe, Germany. Correspondence to: Maximilian Li <m.li@kit.edu>, Rudolf Lioutikov <lioutikov@kit.edu>.

Preprint. March 3, 2026.

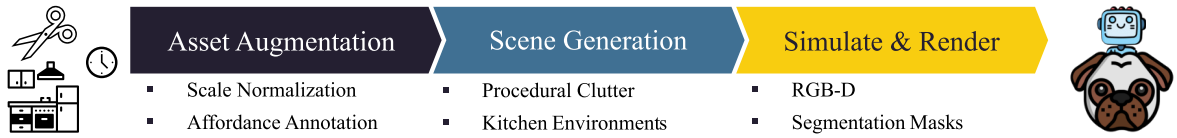


Figure 2. The three stages of the MOPS dataset creation process, which uses 3D assets from RoboCasa and PartNet-Mobility.

3. **Manipulation-relevant Representations** (REQ REP): Beyond images, incorporating other scene representations such as pointclouds or scene graphs.
4. **Manipulation-relevant and Realistic Environments** (REQ ENV): Photorealistic rendering and real-world scenes with natural clutter, beyond controlled laboratory conditions.
5. **Manipulation-relevant Interactions** (REQ INT): Capturing agent-agent, agent-object and object-object interactions over time, ideally supporting direct policy evaluation for manipulation or active perception.

Existing vision datasets fulfill merely a subset of these requirements, creating significant gaps for robotics applications. Human-centric video datasets (Soomro et al., 2012) provide dynamic videos of human-object interaction in realistic environments but are unsuitable for training robot manipulation policies. Datasets focused on task-relevant objects, such as those for affordance detection (Myers et al., 2015) or 6D pose estimation (Xiang et al., 2017), are limited in environmental realism and lack scene dynamics. Conversely, robot datasets (O'Neill et al., 2024; Khazatsky et al., 2024) feature dynamic and realistic scenes but provide ground truth labels of insufficient quality for vision tasks, requiring manual labeling or sophisticated post-hoc annotation pipelines like NILS (Blank et al., 2024).

This work proposes a new dataset generation framework with pixel-level ground truth for **M**ulti-**O**bjective **P**hotoreal **S**imulation (MOPS) addressing all requirements, unlike existing datasets. MOPS bridges the gap between interactive robotics datasets and high-quality vision annotations, making it valuable for both vision and robotics communities. The dataset generation pipeline is illustrated in Figure 2. MOPS uses assets from PartNet-Mobility (Xiang et al., 2020) and RoboCasa (Nasiriany et al., 2024) to create scenes with articulated household objects (REQ OBJ) in photorealistically rendered environments (REQ ENV). The framework normalizes assets and provides manipulation-relevant annotations (REQ ANN) using a zero-shot augmentation pipeline built on GPT-4o (OpenAI, 2024). MOPS provides pixel-level ground truth for class, part, and instance segmentations alongside affordance labels, geometric information (normal maps, 6D poses) and multiple sensor modalities (RGB-D, pointclouds) (REQ REP). MOPS uses the Maniskill3 (Tao et al., 2024) simulator to enable dynamic, interactive scenes suitable for evaluating learned robot behavior or recording teleoperated demonstrations (REQ INT). MOPS combines high-quality vision annotations with interactive robotics scenarios, enabling scalable learning of computer vision for robotic manipulation. In summary, our contributions are:

- A zero-shot augmentation pipeline using LLMs for automatic scale normalization and affordance annotation of 3D assets without manual labeling
- A dataset generation framework producing unlimited photorealistic scenes with pixel-perfect ground truth for 56 affordances across 137 object categories
- Three benchmark datasets (MOPS-Object, MOPS-Clutter, MOPS-Kitchen) with vision and robot learning baselines demonstrating dataset difficulty and downstream utility
- Full compatibility with physics simulation enabling interactive robot learning and teleoperated demonstration recording

2. Related Work

Vision Datasets: The computer vision community has developed specialized datasets for various tasks, including fine-grained image classification with CUB-200-2011 (Wah et al., 2011), which provides RGB images of 200 bird species with point-based annotations for body parts and attributes. For semantic segmentation, datasets like Cityscapes (Cordts et al., 2016) and SemanticKITTI (Behley et al., 2019) focus on autonomous driving scenarios. While these datasets excel for their specific tasks, their content is less relevant to robot manipulation (REQ OBJ).

Indoor scene datasets offer greater relevance for robotics applications. ScanNet++ (Yeshwanth et al., 2023) provides RGB-D voxel scans of indoor scenes with semantic and instance segmentation annotations. Hypersim (Roberts et al., 2021) offers photorealistically rendered indoor scenes (REQ ENV) created by 3D artists, but provides only class and instance segmentations, lacking other critical annotations such as 6D poses or affordances (REQ ANN). The RGB-D Part Affordance dataset (Myers et al., 2015) addresses affordance detection with single object RGB-D images and three cluttered scenes (REQ ANN), but suffers from limited environmental realism with uniform blue backgrounds (REQ

Table 1. **Related Work Overview.** Comparison of computer vision and robotics datasets. MOPS bridges the gap by providing comprehensive part-level affordance annotations (S+I+P+A) across multiple modalities (R+D+M).

Dataset	Objects	Annotations	Represent.	Env.	Interact.	Trajectory
Vision Datasets						
CUB-200-2011 (Wah et al., 2011)	—	P*	R	—	—	—
CityScapes (Cordts et al., 2016)	—	S+I	R	—	—	—
SemanticKITTI (Behley et al., 2019)	—	S+I	R+D+M	—	—	—
ScanNet++ (Yeshwanth et al., 2023)	✓	S+I	R+D+M	✓	—	—
HyperSim (Roberts et al., 2021)	✓	S+I	R+D+M	✓	—	—
RGB-D Part Aff. (Myers et al., 2015)	✓	A	R+D	—	—	—
3D AffordanceNet (Deng et al., 2021)	✓	A	M	—	—	—
PartNet-Mobility (Xiang et al., 2020)	✓	P	M	—	—	—
Robotics Datasets						
2-Handed (Heidinger et al., 2025)	✓	S+I	R+D	✓	—	—
Open-X (O'Neill et al., 2024)	✓	—	R+D	✓	—	✓
DROID (Khazatsky et al., 2024)	✓	—	R+D	✓	—	✓
AI2-THOR (Kolve et al., 2017)	✓	S+I	R+D+M	✓	✓	—
OmniGibson (Li et al., 2023)	✓	S+I	R+D+M	✓	✓	—
RoboCasa (Nasiriany et al., 2024)	✓	S+I	R+D+M	✓	✓	✓
MOPS (Ours)	✓	**S+I+P+A**	**R+D+M**	✓	✓	(✓)

Abbreviations: **S**: Semantic Seg., **I**: Instance Seg., **P**: Part Seg., **P***: Part Centers, **A**: Affordance Seg., **R**: RGB, **D**: Depth, **M**: 3D (Mesh/PC).(✓) denotes compatibility with external trajectories (RoboCasa).

ENV). Like most vision datasets, these lack interactivity (REQ INT).

3D Asset Datasets: 3D AffordanceNet (Deng et al., 2021) provides nearly 23k 3D point clouds across 23 object categories with 18 affordance labels (REQ OBJ& REQ ANN) but lacks material information crucial for photorealistic rendering (REQ ENV). PartNet-Mobility (Xiang et al., 2020) addresses this gap by providing ShapeNet objects with material information and modeled articulation suitable for physics simulation (REQ INT), yet lacks affordance annotations (REQ ANN). A critical limitation of these 3D asset libraries is inconsistent scaling—objects are not modeled to a common reference scale and often appear disproportionate relative to robotic manipulators. MOPS addresses these limitations through a zero-shot asset augmentation pipeline based on large language models to enrich PartNet-Mobility assets with affordance annotations and normalize scaling to realistic proportions for simulation.

Robotics Datasets: Robotics datasets typically consist of real-world or simulated sensor inputs paired with robot movement trajectories. The Open X-Embodiment dataset (O'Neill et al., 2024) represents one of the largest collections, comprising over 70 individual datasets across different laboratories, though this diversity leads to consistency and data quality issues (REQ ANN). The DROID dataset (Khazatsky et al., 2024) addresses some consistency concerns through standardized data collection platforms. However, scaling real-robot datasets to match typical vision and language dataset sizes remains prohibitively expensive due to trajectory collection time and ground truth annotation requirements, even with zero-shot labeling frameworks like NILS (Blank et al., 2024).

Simulation frameworks offer more scalable alternatives. AI2-THOR (Kolve et al., 2017), OmniGibson (Li et al., 2023), and RoboCasa (Nasiriany et al., 2024) provide complex room-scale scenes with photorealistic rendering capabilities (REQ ENV& REQ REP). AI2-THOR offers extensive procedurally generated scene variety, while OmniGibson employs high-quality physics simulation including deformable objects and fluids for complex tasks. RoboCasa provides over 100k training trajectories obtained through human demonstrations and MimicGen (Mandlekar et al., 2023) generation. Generative simulation frameworks such as RoboGen (Wang et al., 2023) and Genesis (Genesis, 2024) leverage generative AI for near-unlimited scene variety, though they perform best with simple task prompts rather than cluttered environments, limiting their effectiveness for vision applications (REQ ENV).

While simulated datasets provide easier access to ground truth annotations for vision tasks like depth and 6D pose estimation, or instance segmentation (REQ ANN), as well as point clouds (REQ REP), obtaining basic annotations like object classes for detection or semantic segmentation remains challenging. To the best of our knowledge, no simulation

framework provides comprehensive pixel-wise ground truth annotations for semantic concepts and affordances out of the box.

MOPS addresses these limitations by combining RoboCasa's high scene variety (Nasiriany et al., 2024) with zero-shot augmented assets from PartNet-Mobility (Xiang et al., 2020), enabling generation of virtually unlimited realistic scenes with pixel-wise ground truth annotations, including affordances, in cluttered environments across multiple representations. Built on the ManiSkill3 simulator (Tao et al., 2024), MOPS improves visual quality and generation speed through raytracing and GPU parallelization, similar to Isaac Lab (Mittal et al., 2023). This positions MOPS as an efficient and valuable resource for both robot learning and computer vision communities. Table 1 provides a comprehensive comparison of current datasets against manipulation-relevant requirements.

3. Methodology Overview

The MOPS dataset generation pipeline consists of three main stages, as illustrated in Figure 2:

1. **Asset Augmentation**: We apply a zero-shot LLM-based pipeline to normalize object scales and generate affordances (Section 4),
2. **Scene Generation**: We procedurally combine RoboCasa kitchen environments with augmented PartNet-Mobility objects to create realistic, cluttered indoor scenes (Section 5).
3. **Rendering and Annotation**: Using ManiSkill3's photorealistic renderer, we generate multi-modal observations with pixel-perfect ground truth including segmentations, 6D poses and multilabel affordances (Section 5).

4. Zero Shot 3D Asset Augmentation

MOPS creates photorealistically rendered scenes of indoor environments for robot manipulation (REQ ENV) by using the ManiSkill3 (Tao et al., 2024) simulator and renderer. To populate the scenes with realistic and interactive objects (REQ OBJ) MOPS uses 3D assets provided by RoboCasa (Nasiriany et al., 2024) and PartNet-Mobility (Xiang et al., 2020). 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 address these issues, MOPS employs a zero-shot augmentation pipeline based on GPT-4o (OpenAI, 2024).

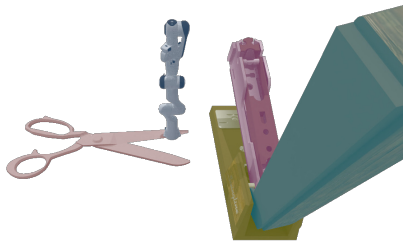


Figure 3. The MOPS pipeline scales 3D assets (e.g., PartNet-Mobility) to a realistic range relative to the robot, driven by scale estimates from an LLM. Affordances for parts and objects are also generated by the LLM (shown as colored overlays).

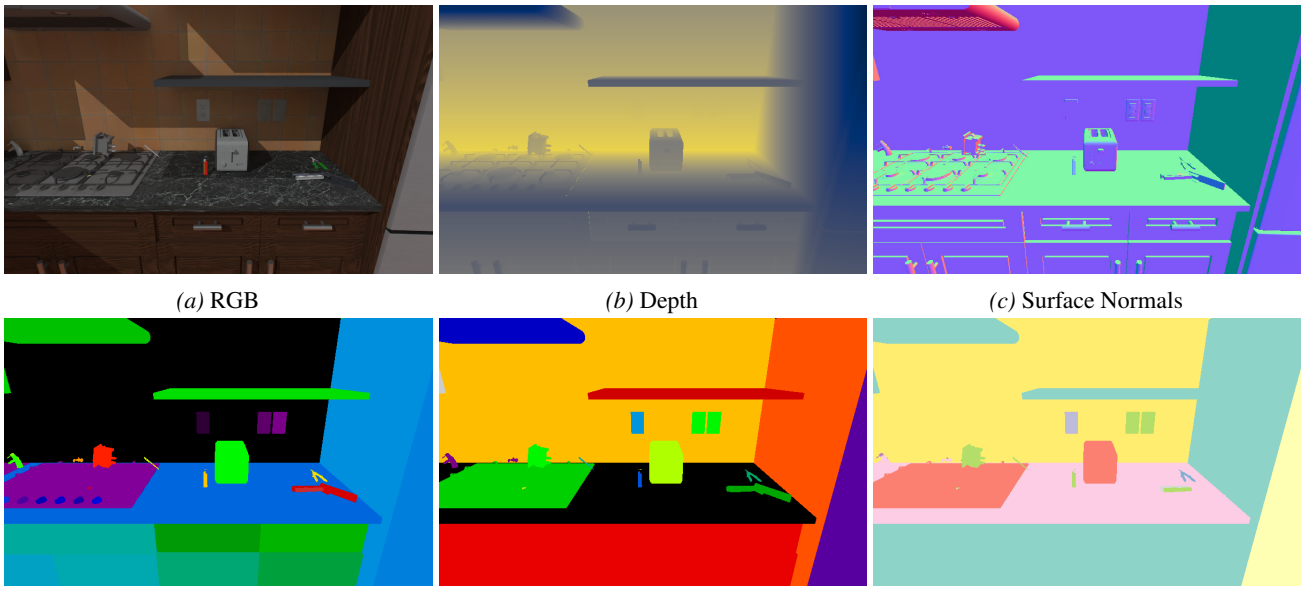
4.1. Zero-Shot Asset Normalization (REQ OBJ)

MOPS uses 3D assets of common household objects. However, most 3D assets 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 normalization stage to address 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 (Franka Robotics) 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. Through the common-sense knowledge embedded in recent Large Language Models (LLMs), MOPS can obtain realistic object scales. In particular, 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 realistic scaling factors $s_{\min}$ and $s_{\max}$. When loading an asset instance into the simulation, we sample a random scaling factor from the range $[s_{\min}, s_{\max}]$ to increase visual variety within a scene. Figure 3 presents an example scene before scale normalization. This approach enables easy addition of new object categories and models into the dataset without human intervention.

4.2. 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 (Mo et al., 2019), which in turn is a subset of ShapeNet (Chang et al., 2015). The fully articulated PartNet-Mobility assets provide a wide range of interactions (REQ INT), but are missing manipulation annotations such as affordances (REQ ANN).



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

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

Figure 4. MOPS provides multiple, pixelwise ground truth maps in addition to RGB-D perception.

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 *warmupable*. Future work could extend the annotation stage to produce region-level affordance annotations similar to the manually labeled 3D AffordanceNet (Deng et al., 2021). Meanwhile, the already presented zero-shot annotation stage alleviates the human-labeling effort significantly. The exact prompts for the zero shot 3D asset augmentation pipeline are detailed in Appendix A.

5. MOPS Dataset Generation

MOPS can generate an unlimited number of 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.

Throughout this work, *interaction* refers to physics-enabled object manipulation (e.g., opening drawers, articulating joints), while *robot trajectory* refers to sequences of robot actions, such as joint configurations, end-effector poses and gripper commands during task execution.

5.1. MOPS Ground Truth Masks (REQ ANN)

MOPS provides multiple scene representations and camera views to facilitate training for robot manipulation tasks. Camera configurations include birds-eye view (for SLAM reconstruction in mobile manipulation), and external over-the-shoulder, ego, and in-hand views (mimicking typical robot manipulation setups).

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 millimeters, 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 4 presents different, visual camera modalities from an ego camera in a sparsely populated RoboCasa scene. The RGB modality is rendered via raytracing, resulting in realistic shadows and reflections. The depth modality shows the distance of each pixel to the virtual camera. The sur-

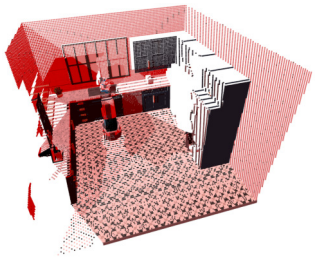


Figure 5. Pointcloud of a RoboCasa scene with red ambient lighting.

Figure 6. A RoboCasa kitchen filled with PartNet-Mobility clutter.

face normal modality encodes the XYZ components of each surface 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 separate part, but share the same class. Finally, affordances are modeled in MOPS as a multi-label annotation; consequently, the annotation map output is a per-pixel binary vector indicating the presence or absence of each affordance type.

5.1.1. Extended Representations (Req Rep)

The underlying ManiSkill3 simulation 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 5 shows a scene pointcloud with red ambient lighting.

The SAPIEN renderer provides realistic stereo depth sensors with active IR lighting and simulated sensor noise (Sapien Developers, 2024). While these sensors are not yet fully available in the current ManiSkill3 release[1], their integration into MOPS will be straightforward once released, based on prior experience with ManiSkill2.

5.2. 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 assets in the RoboCasa kitchen scenes. First, MOPS obtains the location of the kitchen counter tops 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 (see Figure 6). While this is a rather heuristic approach compared to trained models such as ClutterGen (Jia & Chen, 2024), it can be easily applied to novel scenes and environments outside of RoboCasa, without requiring any training. More scenes are presented in Appendix F.

5.3. Interactive Simulation (Req Int)

MOPS leverages the ManiSkill3 simulation to provide full interactivity. While MOPS primarily uses existing robot trajectories from the RoboCasa dataset (see Table 1), the framework also supports recording new demonstrations through a teleoperation interface for researchers who wish to collect additional interaction data.

Robot trajectories such as the RoboCasa demonstrations can be rolled out to record observations in dynamic environments with full physics simulation, capturing object-object and robot-object interactions. Additionally, new demonstrations can be recorded using the teleoperation interface provided by ManiSkill3. Figure 7 illustrates a teleoperated demonstration where 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 generate the comprehensive ground truth segmentations provided by MOPS.

6. Affordance Analysis

MOPS provides comprehensive affordance annotations across multiple granularities. We derive part-level annotations from PartNet-Mobility (Xiang et al., 2020), contributing 24 affordance types for 14,100 parts across 2,345 objects. To extend the semantic reach of our benchmark, we further incorporate object-level annotations for RoboCasa assets with 44 affordance types across 101 categories. By combining the both asset sets, **MOPS offers 56 affordance labels for 137 different object categories**.

[1] ManiSkill v3.0.0b22, retrieved 2026-01-22 from GitHub

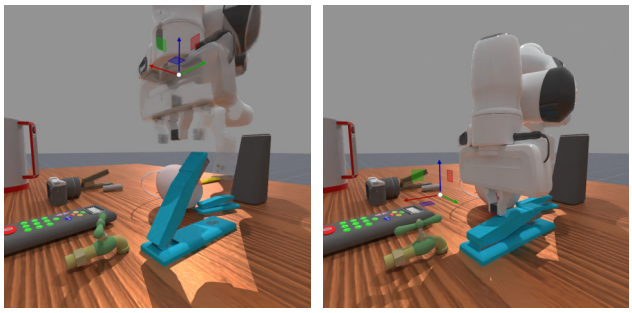


Figure 7. MOPS uses the mouse-and-keyboard teleoperation interface by ManiSkill3. The semitransparent robot arm previews the new target position.

While existing benchmarks like 3D-AffordanceNet (Deng et al., 2021) provide a higher volume of individual object instances, MOPS prioritizes taxonomic diversity. As shown in Table 2, MOPS offers 3.5× more affordance types and over 6× the number of object categories compared to prior work. This makes MOPS the most diverse dataset available for affordance detection, shifting the focus from instance-level repetition to the long-tail distribution of real-world object interactions.

Table 2. Comparison of dataset diversity. While 3D-AffordanceNet has more instances, MOPS provides significantly higher taxonomic coverage across object categories and affordance types.

Dataset	Level	Aff. Labels	Obj. Cat.	Objects
RGB-D Part	Part	7	17	105
3D-AffNet	Part	16	23	**22,949**
MOPS-Partnet	Part	24	46	2,345
MOPS-Robocasa	Object	44	101	1,008
MOPS (Total)	**Mixed**	**56**	**137**	3,353

7. Experiments

We construct three benchmark datasets with systematic differences in scene composition and visual complexity. We use a randomized train-test splits per asset class, ensuring no 3D model overlap between splits to test generalization capabilities of the models. Each 512×512 image uses randomized lighting (intensity, color, direction), object poses, and camera positions. To ensure physical plausibility, images are captured only after a physics settling period. Our generation pipeline achieves a throughput of ≈ 1000 images/hour on consumer-grade hardware. Qualitative examples for all settings are provided in Appendix F.

MOPS-Object consists of isolated assets centered on uniform backgrounds. It contains 30 images per category (15 train/15 test) across 46 balanced part-level classes, mirroring the structure of the RGB-D Part Affordance Dataset (Myers et al., 2015).

Table 3. **Object Classification Baselines on MOPS.** All CLIP results use the ViT-B/32 architecture.

Model	Training	Top-1 Acc. (%)
ResNet-50	Full	61.3
ViT-B/16	Full	63.5
CLIP	Frozen	40.7
CLIP	Full	**72.2**

MOPS-Clutter evaluates affordance recognition in unstructured tabletop scenes. It features randomized arrangements of multiple assets (3000 training, 1000 test images).

MOPS-Kitchen represents our most challenging setting, situating PartNet-Mobility assets within complex RoboCasa kitchen environments alongside various distractor objects (3000 training, 1000 test images).

7.1. Vision Benchmark

We validate MOPS's utility through two complementary tasks: single-object classification on MOPS-Object (Table 3) and multi-label affordance segmentation on MOPS-Clutter and MOPS-Kitchen (Table 4). Comprehensive training hyperparameters are detailed in Appendix C.

Classification. We benchmark standard vision backbones, including ResNet-50 (He et al., 2015) and ViT-B/16 (Dosovitskiy et al., 2021), against CLIP (Radford et al., 2021) (ViT-B/32) to evaluate the difficulty of the MOPS-Object dataset. We evaluate these models in two states: frozen and fully trained. While CLIP in a frozen state provides a baseline for general knowledge, we find that even the best-performing model requires full training on our dataset to achieve high accuracy, yielding a $+31.5\%$ absolute gain over the frozen baseline. This significant performance gap highlights the domain-specific challenges inherent in our benchmark.

Affordance Segmentation. To evaluate the complexity of the 56-class affordance task, we benchmark DeepLabV3 (Chen et al., 2017), Segformer (Xie et al., 2021), and DINOv2. We report mIoU macro-averaged across all classes, comparing fully trained models against a frozen backbone. The results demonstrate a clear distribution shift: While fully trained models like Segformer achieve strong performance in the controlled Clutter setting (mIoU: 16.2%), their performance degrades in the more complex Kitchen environment (mIoU: 22.1%). Interestingly, frozen DINOv2 shows superior robustness in the Kitchen setting (mIoU: 34.9% vs 22.1%), suggesting that self-supervised pretraining captures more generalizable features for complex, cluttered scenes than task-specific fine-tuning on simpler environments.

Table 4. **Multi-Label Segmentation Baselines.** Models are evaluated on 56 affordance labels using macro-averaged mIoU and F1 scores.

		Clutter Setting		Kitchen Setting	
Model	**Training**	**mIoU**	**F1**	**mIoU**	**F1**
DeepLabV3	Full	15.2	16.5	17.9	20.3
Segformer	Full	**16.2**	**17.9**	22.1	23.9
DINOv2	Frozen	13.5	14.3	**34.9**	**38.9**

Table 5. **Robot Manipulation Performance.** Imitation learning results on 24 RoboCasa tasks, evaluated over 10 environment seeds each.

Policy Inputs	**Success Rate**	**Gain**
RGB only	13.33%	–
RGB + MOPS Affordances	**21.25%**	+7.92

7.2. Robot Manipulation

To evaluate the utility of the MOPS-derived annotations for downstream robotics tasks, we conduct imitation learning experiments on 24 single-stage RoboCasa tasks (Nasiriany et al., 2024) spanning pick-and-place, articulated object manipulation and multi-object rearrangement scenarios.

Experimental Setup. We train DiTFlow policies (Shafiullah & San José Pro, 2024), a Diffusion Transformer policy (Dasari et al., 2024) using Flow Matching, within the LeRobot framework (Cadene et al., 2024). Each of the 24 tasks uses 50 human teleoperated demonstrations for training.

We compare two conditions: (1) an RGB-only baseline DiTFlow policy conditioned on observations from three cameras (in-hand, left-shoulder, right-shoulder), and (2) an affordance-aware policy that additionally receives ground-truth MOPS affordance segmentation masks for each camera view. Each task is evaluated across 10 randomized environment seeds with different object poses and clutter configurations. Training hyperparameters are detailed in Appendix D

Results. Table 5 presents the average success rates across all 24 tasks. The affordance-aware policy achieves 19.58% success rate compared to 13.33% for the RGB-only baseline, a +6.25 percentage point absolute improvement. This demonstrates that incorporating ground-truth affordance information leads to measurable improvements in manipulation performance, validating that MOPS provides supervision for a capability that transfers to robot behavior.

8. 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 multiple, 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 using GPT-4o to generate a diverse affordance learning datasets with 56 affordance labels across 137 object categories. This augmentation pipeline can be quickly applied to new asset libraries or novel semantic labels for even more variety. We validate MOPS through vision benchmarks and demonstrate that ground-truth affordances improve robot manipulation success rates by 7% compared to RGB-only baselines, confirming the dataset provides actionable supervision for robotics applications.

Limitations and Future Directions. Although MOPS leverages many advantages provided by the underlying ManiSkill3 simulator, it is also limited by the simulators functionality. For example, the raytraced rendering is not usable in conjunction with GPU parallelization, limiting the maximum throughput. Future work could adapt MOPS to other simulators such as Isaac Lab (Mittal et al., 2023) or Genesis (Genesis, 2024) for enhanced rendering and parallelization capabilities.

MOPS currently relies on 3D assets with existing material information and textures. The zero-shot asset augmentation pipeline could be extended to also generate materials and textures via generative AI, which would enable import of the entire PartNet (Mo et al., 2019) and ShapeNet (Chang et al., 2015) datasets, significantly expanding object diversity.

While our current robot learning experiments utilize ground-truth affordances from an oracle to demonstrate the downstream utility of the MOPS dataset, they do not constitute a final robot policy. Future work will focus on the development of robust affordance prediction models that bridge the gap from RGB observations to end-to-end manipulation, a path directly supported by the diversity of our training data.

Impact Statement

This paper presents work aimed at advancing robotic manipulation and computer vision through synthetic data generation. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

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

Blank, N., Reuss, M., Rühle, M., Yağmurlu, Ö. E., Wenzel, F., Mees, O., and Lioutikov, R. Scaling robot policy learning via zero-shot labeling with foundation models. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=EdVNB2kHv1.

Cadene, R., Alibert, S., Soare, A., Gallouedec, Q., Zouitine, A., Palma, S., Kooijmans, P., Aractingi, M., Shukor, M., Aubakirova, D., Russi, M., Capuano, F., Pascal, C., Choghari, J., Moss, J., and Wolf, T. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. https://github.com/huggingface/lerobot, 2024.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. Rethinking atrous convolution for semantic image segmentation, 2017. URL https://arxiv.org/abs/1706.05587.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Dasari, S., Mees, O., Zhao, S., Srirama, M. K., and Levine, S. The ingredients for robotic diffusion transformers. *arXiv preprint arXiv:2410.10088*, 2024.

Deng, S., Xu, X., Wu, C., Chen, K., and Jia, K. 3d affordancenet: A benchmark for visual object affordance understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1778–1787, 2021.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

Franka Robotics. Franka Denavit–Hartenberg parameters. URL https://frankaemika.github.io/docs/control_parameters.html#denavithartenberg-parameters. Franka FCI Documentation.

Genesis. Genesis: A universal and generative physics engine for robotics and beyond, December 2024. URL https://github.com/Genesis-Embodied-AI/Genesis.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

Heidinger, M., Jauhri, S., Prasad, V., and Chalvatzaki, G. 2handedafforder: Learning precise actionable bimanual affordances from human videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14743–14753, October 2025.

Jia, Y. and Chen, B. Cluttergen: A cluttered scene generator for robot learning. In *8th Annual Conference on Robot Learning*, 2024.

Khazatsky, A., Pertsch, K., Nair, S., Balakrishna, A., Dasari, S., Karamcheti, S., Nasiriany, S., Srirama, M. K., Chen, L. Y., Ellis, K., et al. Droid: A large-scale in-the-wild robot manipulation dataset, 2024.

Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Deitke, M., Ehsani, K., Gordon, D., Zhu, Y., et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

Li, C., Zhang, R., Wong, J., Gokmen, C., Srivastava, S., Martín-Martín, R., Wang, C., Levine, G., Lingelbach, M., Sun, J., et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pp. 80–93. PMLR, 2023.

Mandlekar, A., Nasiriany, S., Wen, B., Akinola, I., Narang, Y., Fan, L., Zhu, Y., and Fox, D. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *Conference on Robot Learning*, pp. 1820–1864. PMLR, 2023.

Mittal, M., Yu, C., Yu, Q., Liu, J., Rudin, N., Hoeller, D., Yuan, J. L., Singh, R., Guo, Y., Mazhar, H., Mandlekar, A., Babich, B., State, G., Hutter, M., and Garg, A. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023. doi: 10.1109/LRA.2023.3270034.

Mo, K., Zhu, S., Chang, A. X., Yi, L., Tripathi, S., Guibas, L. J., and Su, H. 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)*, June 2019.

Myers, A., Teo, C. L., Fermüller, C., and Aloimonos, Y. Affordance detection of tool parts from geometric features. In *ICRA*, 2015.

Nasiriany, S., Maddukuri, A., Zhang, L., Parikh, A., Lo, A., Joshi, A., Mandlekar, A., and Zhu, Y. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems*, 2024.

OpenAI. Gpt-4o system card, 2024. URL https://arxiv.org/abs/2410.21276.

O'Neill, A., Rehman, A., Maddukuri, A., Gupta, A., Padalkar, A., Lee, A., Pooley, A., Gupta, A., Mandlekar, A., Jain, A., 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)*, pp. 6892–6903. IEEE, 2024.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Roberts, M., Ramapuram, J., Ranjan, A., Kumar, A., Bautista, M. A., Paczan, N., Webb, R., and Susskind, J. M. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *International Conference on Computer Vision (ICCV) 2021*, 2021.

Sapien Developers. Sapien 3.0 documentation, 2024. URL https://sapien-sim.github.io/docs/user_guide/rendering/depth_sensor.html. Sapien StereoDepthSensor Documentation.

Shafiullah, N. M. and San José Pro, D. DiT-Flow policy in LeRobot. GitHub Pull Request #680, https://github.com/huggingface/lerobot/pull/680, 2024. Accessed: 2026-01-26.

Soomro, K., Zamir, A. R., and Shah, M. Ucf101: A dataset of 101 human actions classes from videos in the wild, 2012. URL https://arxiv.org/abs/1212.0402.

Tao, S., Xiang, F., Shukla, A., Qin, Y., Hinrichsen, X., Yuan, X., Bao, C., Lin, X., Liu, Y., Chan, T.-k., Gao, Y., Li, X., Mu, T., Xiao, N., Gurha, A., Huang, Z., Calandra, R., Chen, R., Luo, S., and Su, H. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 2024.

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

Wang, Y., Xian, Z., Chen, F., Wang, T.-H., Wang, Y., Fragkiadaki, K., Erickson, Z., Held, D., and Gan, C. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation, 2023.

Xiang, F., Qin, Y., Mo, K., Xia, Y., Zhu, H., Liu, F., Liu, M., Jiang, H., Yuan, Y., Wang, H., Yi, L., Chang, A. X., Guibas, L. J., and Su, H. SAPIEN: A simulated part-based interactive environment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

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

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

Yeshwanth, C., Liu, Y.-C., Nießner, M., and Dai, A. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023.

Zellers, R., Yatskar, M., Thomson, S., and Choi, Y. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5831–5840, 2018.

A. GPT-4o Labeling Prompts

For reproducibility, we provide the exact prompts used for GPT-4o annotation.

A.1. Scale Annotation

System prompt:

> I will provide you with the name of the object. Output the typical lengths, widths, and heights of the object. Output json with the following format:
>
> ```
> {"length": [low,high],
> "width": [low,high],
> "height": [low,high]}
> ```

A.2. Affordance Annotation

System prompt:

> You will be provided with an object and parts of the object. Assign an affordance to each object part. Examples for affordances are the following: grasp, contain, lift, openable, layable, sittable, support, pourable, move, display, pushable, pull, listen, wear, press, cut, stab. You are not restricted to these affordances. Output valid json

Few-shot example:

User: Object: Remote
Parts: button, remote_base, knob, rotation_button

Assistant:

```
{
  "button": [
    "press"
  ],
  "remote_base": [
    "grasp", "moveable", "layable"
  ],
  "knob": [
    "press", "rotate"
  ],
  "rotation_button": [
    "rotate"
  ]
}
```

A.3. Detailed Affordances

Table 6. **Part-Level Affordance Statistics (PartNet-Mobility).** Distribution of affordance types across the 46 part-level categories in the MOPS 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 7. **Object-Level Affordance Statistics (RoboCasa).** Distribution of affordance types across the 101 object-level categories in the MOPS dataset.

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		

B. Compute Resources

All Experiments and Data Generation were performed on a single Desktop Workstation with Intel i7-12700 20-Core CPU, NVidia RTX 3070 (8GB) GPU, 32 GB RAM, Python 3.14 and Pytorch 2.9.1

C. Vision: Training Details

For reproducibility, we provide detailed training configurations for all baseline models evaluated on the MOPS datasets.

Online Data augmentation: Random horizontal flip, random rotation (±10°), color jitter.

C.1. Image Classification

Table 8. Training hyperparameters for image classification models on MOPS-Object.

Parameter	**ResNet50**	**ViT-B/16**	**CLIP-FT**
Batch size	64	32	32
Optimizer	AdamW	AdamW	AdamW
Learning rate	1e-4	1e-4	1e-5
LR schedule	None	Cosine w/ warmup	None
Warmup epochs	-	12	-
Weight decay	0.01	0.05	0.05
Epochs	40	120	10
Augmentation	Yes	Yes	Yes
Loss function	Cross-entropy	Cross-entropy	Cross-entropy
Pretrained weights	ImageNet-1K	ImageNet-1K	OpenAI CLIP

C.2. Affordance Segmentation

Table 9. Training hyperparameters for segmentation models on MOPS-Clutter and MOPS-Kitchen.

Parameter	**DeepLabV3**	**SegFormer-B2**	**DinoV2**
Backbone	ResNet50	Transformer	ViT-B/14
Batch size	16	16	16
Optimizer	AdamW	AdamW	AdamW
Learning rate	1e-4	1e-4	1e-3
LR schedule	OneCycle	OneCycle	None
Weight decay	0.01	0.01	0.01
Epochs	200	200	20
Training mode	Full	Full	Linear probing
Loss function		Binary Cross Entropy	
Pretrained weights	COCO	ImageNet-1K	DINOv2

C.3. Evaluation

We evaluate our models using the following metrics and data preprocessing steps:

Classification Top-1 accuracy on a balanced 46-class test set.

Segmentation Mean IoU (mIoU) across 56 affordance labels.

Data Preprocessing All images are resized to 512×512. For classification, ImageNet normalization is applied, while segmentation models normalize images to $[0, 1]$.

D. Imitation Learning Details

Our imitation learning experiments use DiTFlow policies (Shafiullah & San José Pro, 2024) implemented in the LeRobot framework (Cadene et al., 2024). All experiments were performed on a single Desktop Workstation with AMD Ryzen 9 9950X CPU, Nvidia RTX 5090 (32GB) GPU, 128 GB RAM, Python 3.14 and Pytorch 2.9.1.

Table 10. Training hyperparameters for DiTFlow on robot learning tasks.

Parameter	**Value**
Backbone	ResNet34
Batch size	32
Optimizer	Adam
Learning rate	1e-4
LR schedule	Cosine with warmup
Loss function	Flow Matching

The policies receive the RGB observations from the inhand, left-shoulder and right-shoulder cameras with resolutions 256×256. The affordance aware policy additionally receives ground truth affordance segmentation masks for each camera sensor. Each model is trained for 50 epochs using 50 human teleoperated demonstrations for each of Robocasa's 24 single stage tasks.

E. MOPS Dataset Structure

Our datasets are stored in the *Hierarchical Data Format 5* (HDF5) to allow for efficient access and compression of large-scale, image-based data. Each dataset is structured as a single HDF5 file with a clear, hierarchical organization. The file's root contains several primary groups: `images`, `masks`, `metadata`, and, for single-object datasets, a `labels` group.

E.1. HDF5 Group and Dataset Structure

The dataset is organized into a clear hierarchy that mirrors a directory structure, simplifying access. The overall structure is defined as follows.

- `MOPS-dataset/` (HDF5 Root)
  - `images/`
    - `image_XXXXXX` (e.g., `image_000000`, `image_000001`, etc.)
  - `masks/`
    - `affordance/`: Affordance Segmentation for each `image_XXXXXX`.
    - `bbox/`: Bounding Box Annotation for each `image_XXXXXX`.
    - `depth/`: Depth Map for each `image_XXXXXX`.
    - `instance/`: Instance Segmentation for each `image_XXXXXX`.
    - `normal/`: Normal Map for each `image_XXXXXX`.
    - `part/`: Part Segmentation for each `image_XXXXXX`.
    - `semantic/`: Semantic Class Segmentation for each `image_XXXXXX`.
  - `metadata/`
    - `image_info`
    - `splits`
    - `split_counts`
    - `total_images`
  - `labels/` (Optional, for single-object datasets)
    - `class_names`
    - `class_labels`
    - `class_counts`

E.2. Dataset Contents

The following table details the contents of the primary groups and their respective datasets. Each image record has a unique ID, `image_XXXXXX`, that links corresponding data across different groups.

Table 11. Structure and contents of the HDF5 dataset.

Group	Dataset/Item	Description
`images`	`image_*`	RGB images as 3-channel, 8-bit arrays (linked by `image_ID`).
`masks`	`semantic`	Compressed, pixel-level semantic segmentation masks.
	`instance`	Compressed, pixel-level instance segmentation masks.
	`part`	Compressed, pixel-level part segmentation masks.
	`affordance`	Compressed, pixel-level affordance segmentation masks.
	`depth`	Compressed depth maps in a high-precision format.
	`normal`	Compressed surface normal maps.
	`bbox`	Bounding box data in `[x, y, w, h, class_id]` format.
`metadata`	`image_info`	Variable-length string dataset (JSON-formatted dictionary per image).
	`splits`	Boolean array indicating dataset split (e.g., train/test) for each image.
	`split_counts`	JSON object with total number of images in each split.
	`total_images`	Single value indicating the total number of images in the dataset.
`labels`	`class_names`	Array of unique class name strings (Single-object datasets only).
	`class_labels`	Integer array of class indices for each image.
	`class_counts`	JSON object detailing the distribution of each class in the dataset.

F. MOPS Dataset Example Images

The following pages showcase some random training samples from the three generated Datasets MOPS-Object (Figure 8), MOPS-Clutter (Figure 9 and MOPS-Kitchen (Figure 10).

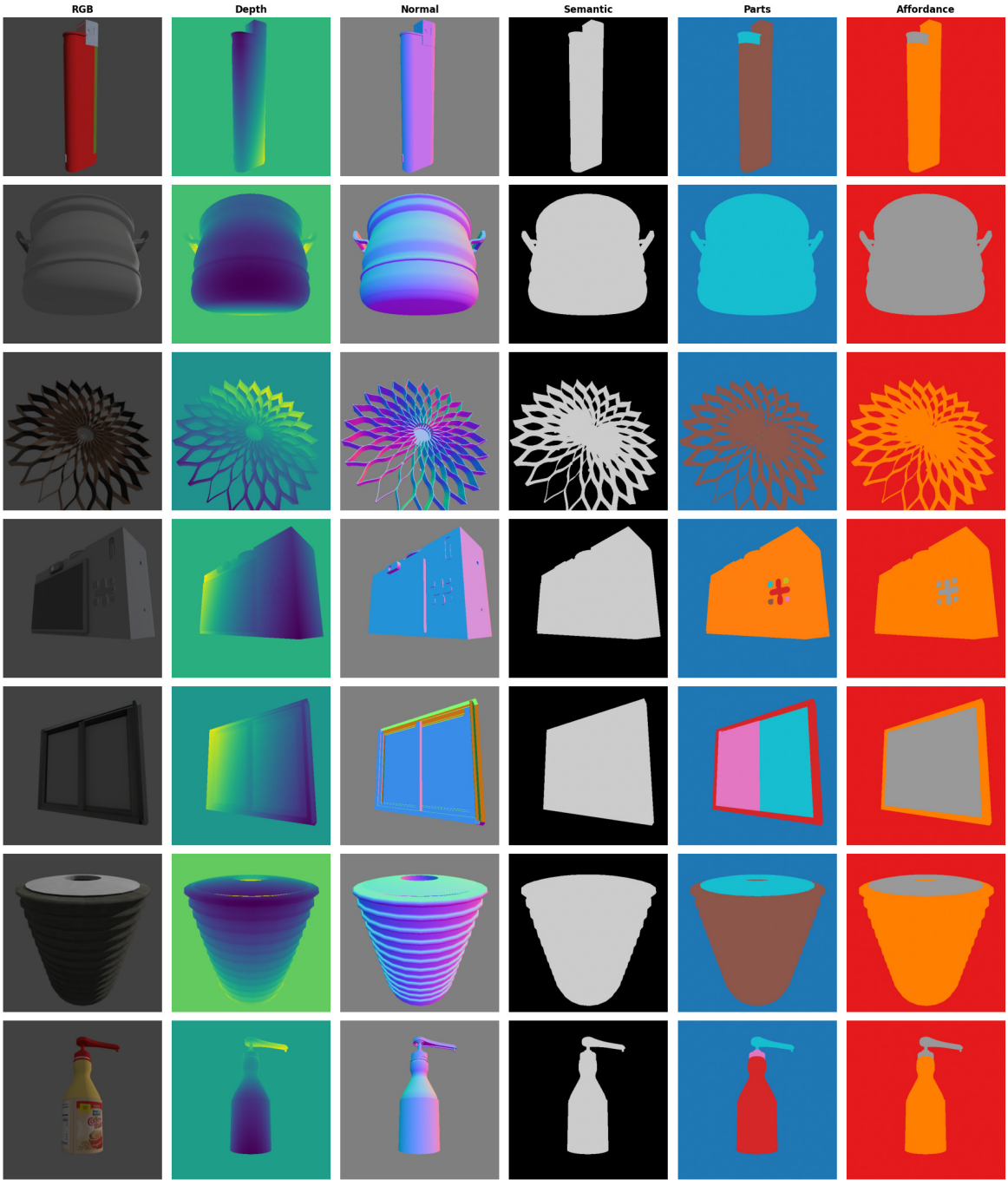


Figure 8. MOPS-Object dataset examples.

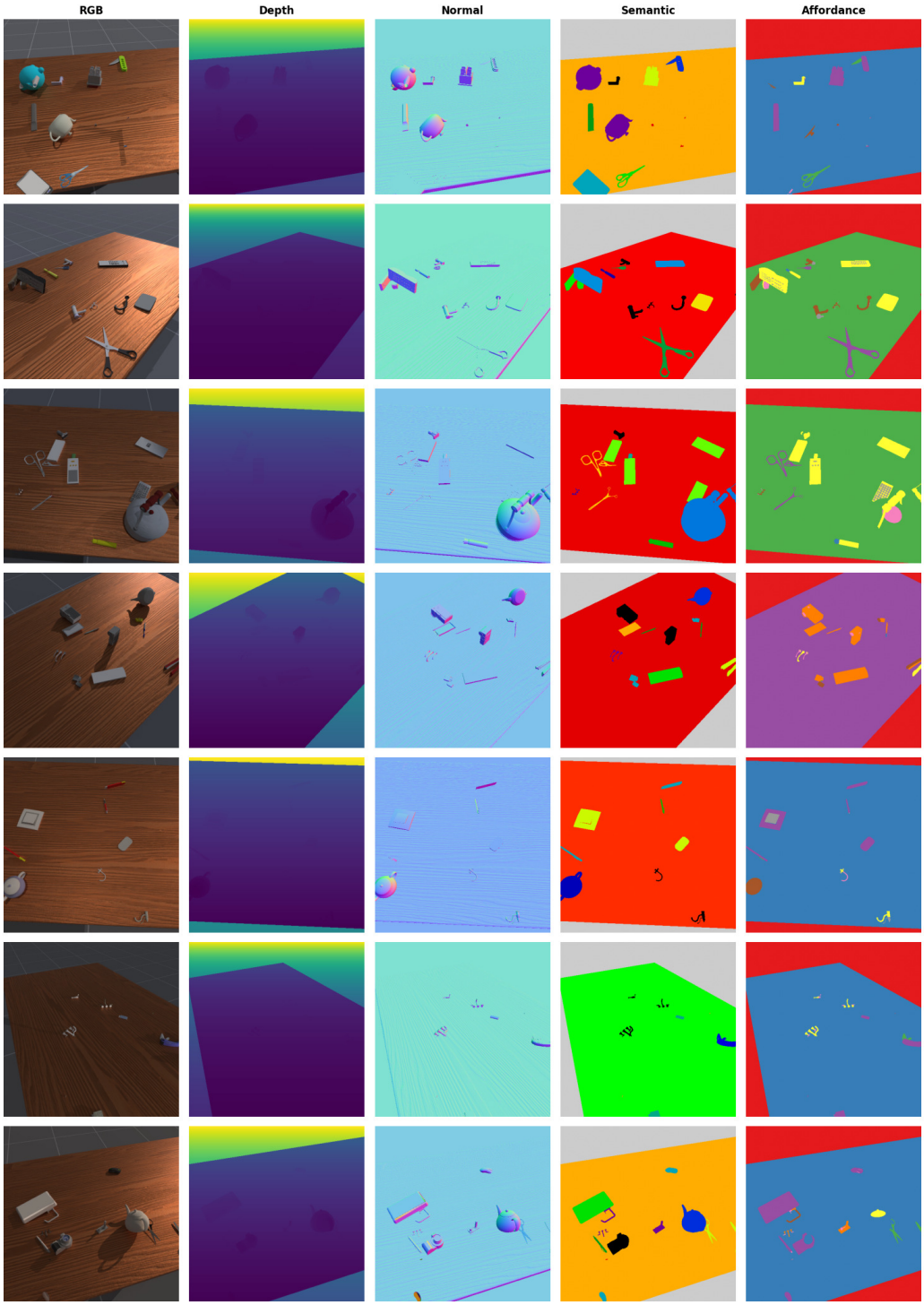


Figure 9. MOPS-Clutter dataset examples.

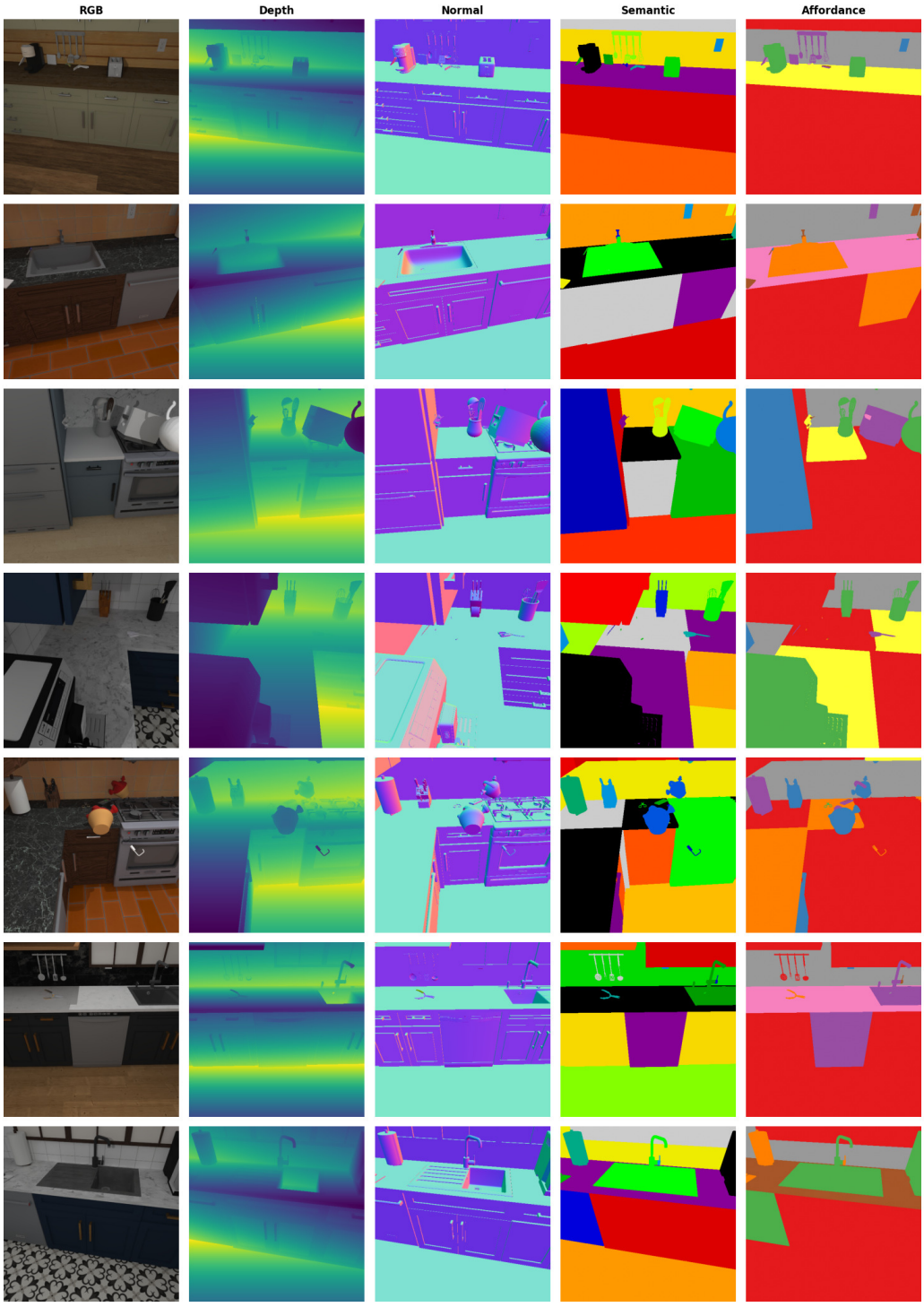


Figure 10. MOPS-Kitchen dataset examples.