BEHAVIOR Vision Suite: Customizable Dataset Generation via Simulation

Yunhao Ge[1,2*], Yihe Tang[1*], Jiashu Xu[3*], Cem Gokmen[1*], Chengshu Li[1], Wensi Ai[1], Benjamin Jose Martinez[1], Arman Aydin[1], Mona Anvari[1], Ayush K Chakravarthy[1], Hong-Xing Yu[1], Josiah Wong[1], Sanjana Srivastava[1], Sharon Lee[1], Shengxin Zha[4], Laurent Itti[2], Yunzhu Li[1,7], Roberto Martín-Martín[6], Miao Liu[4], Pengchuan Zhang[5], Ruohan Zhang[1], Li Fei-Fei[1], Jiajun Wu[1]

[1]Stanford University [2]University of Southern California [3]Harvard University [4]GenAI, Meta
[5]FAIR, Meta [6]The University of Texas at Austin [7]The University of Illinois Urbana-Champaign

Abstract

The systematic evaluation and understanding of computer vision models under varying conditions require large amounts of data with comprehensive and customized labels, which real-world vision datasets rarely satisfy. While current synthetic data generators offer a promising alternative, particularly for embodied AI tasks, they often fall short for computer vision tasks due to low asset and rendering quality, limited diversity, and unrealistic physical properties. We introduce the BEHAVIOR Vision Suite (BVS), a set of tools and assets to generate fully customized synthetic data for systematic evaluation of computer vision models, based on the newly developed embodied AI benchmark, BEHAVIOR-1K. BVS supports a large number of adjustable parameters at the scene level (e.g., lighting, object placement), the object level (e.g., joint configuration, attributes such as "filled" and "folded"), and the camera level (e.g., field of view, focal length). Researchers can arbitrarily vary these parameters during data generation to perform controlled experiments. We showcase three example application scenarios: systematically evaluating the robustness of models across different continuous axes of domain shift, evaluating scene understanding models on the same set of images, and training and evaluating simulation-to-real transfer for a novel vision task: unary and binary state prediction. Project website: https://behavior-vision-suite.github.io/

1. Introduction

Large-scale datasets and benchmarks have fueled computer vision research in the past decade [2, 9, 11, 20, 22, 23, 25, 35, 43, 47, 57, 67]. Driven by these datasets and benchmarks, thousands of models and algorithms tackling different perception challenges are being proposed every year, on the topics of object detection [75], segmentation [33], action recognition [62], video understanding [41] and beyond. Despite their success, real-world datasets face inherent limitations. First, the ground-truth object/pixel-level labels are either prohibitively expensive to acquire (e.g., segmentation masks) [42] or suffering from inaccuracies (e.g., depth sensing) [50]. Consequently, each real dataset often only offers limited labels, hindering the development and evaluation of computer vision models that perform a wide range of perception tasks on the same input. Even when annotations are affordable and accurate, real-world datasets are limited by the availability of source images. For example, images of rare events, such as traffic accidents or low-light conditions, might be difficult to acquire from the Internet or real-world sensors. Finally, once collected, these real-world datasets have a fixed data distribution and cannot be easily changed. This makes it challenging for researchers to conduct customized experiments, often leading to models that overfit the datasets and eventually rendering the entire benchmarks obsolete [27, 28, 37, 56].

To avoid this limitation, researchers and practitioners have devised various methods to generate synthetic datasets that complement the real ones [18]. In the realm of indoor scene understanding, 3D reconstruction datasets [4, 53, 66] provide a promising avenue to generate source images from arbitrary viewpoints and free (geometric) annotations. However, due to the imperfect nature of 3D reconstruction techniques, the rendered images are not very realistic. Since each entire scene is a static mesh, these datasets offer very limited customizability beyond camera trajectories. Recent synthetic indoor datasets (often designed by 3D artists) [13, 39, 40, 55] not only offer free geometric and semantic annotations, but also support object layout reconfiguration as objects are usually independent CAD models.

* equal contribution
† correspondence to yunhaoge@cs.stanford.edu, {yihetang, zharu}@stanford.edu

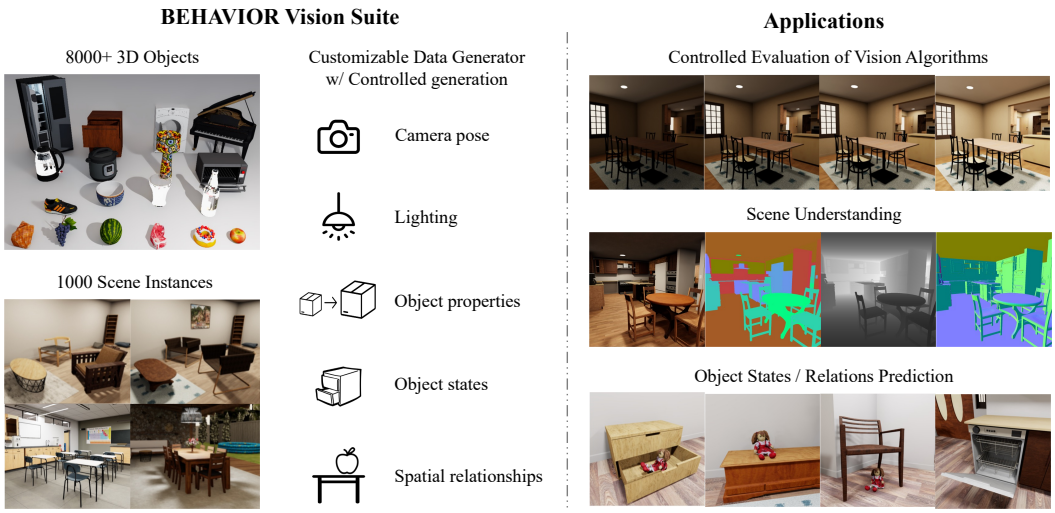


Figure 1. Overview of BEHAVIOR Vision Suite (BVS), our proposed toolkit for computer vision research. BVS builds upon the extended object assets and scene instances from BEHAVIOR-1K [37], and provides a customizable data generator that allows users to generate photorealistic, physically plausible labeled data in a controlled manner. We demonstrate BVS with three representative applications.

However, these datasets do not guarantee physical plausibility, as object penetration and levitation occur frequently, and offer no customization capability beyond changing object poses. 3D simulators [7, 14, 34, 36, 56, 60], on the other hand, guarantee physical plausibility with their underlying physics engines. They allow users to customize the joint configuration of articulated objects and even more advanced object states such as "cooked" or "sliced" [34, 36]. Yet these 3D simulators generally cater to embodied AI and robotics researchers, and as a result, they lack photorealism compared to the synthetic datasets mentioned before (often due to speed constraints), and do not offer ready-made tools to generate customized image/video datasets for computer vision researchers.

To overcome the aforementioned challenges, we propose BEHAVIOR Vision Suite (BVS), a customizable data generation tool that enables systematic evaluation and understanding of computer vision models (see Fig. 1 for an overview). To do so, we expand the 3D asset library in BEHAVIOR-1K [37], focusing on enhancing both object diversity and scene variety, as well as adding features to increase the value of the assets for vision tasks. We also introduce a customizable dataset generator, which uses the simulator from the BEHAVIOR-1K benchmark [37, 59] to generate custom vision datasets. We build a versatile and customizable toolbox to generate high-quality synthetic data for systematic model evaluation and understanding.

In summary, BEHAVIOR Vision Suite possesses the following unique combination of desirable features:

- BVS offers image/object/pixel-level labels (scene graph, point cloud, depth, segmentation, etc.);
- BVS covers a wide variety of indoor scenes and objects (8K+ objects, 1K scene instances, fluid, soft bodies);
- BVS provides physical plausibility and photorealism;
- BVS supports customization in terms of object models, poses, joint configurations, semantic states, lighting, texture, material, camera setting, etc.;
- BVS includes easy-to-use tooling to generate customized data for new use cases.

To demonstrate the usefulness of BVS, we show three example applications: 1) parametrically evaluating model robustness across different conditions such as lighting and occlusion, 2) evaluating different types of representative computer vision models on the same set of images, and 3) training and evaluating sim2real transfer for object state and relation prediction. We hope that BVS can unlock more possibilities for the computer vision community.

2. Related works

In this section, we compare BEHAVIOR Vision Suite with other real RGB-D datasets, 3D reconstruction datasets, synthetic datasets, and 3D simulators in terms of customizability and visual quality (see Tab. 1).

Real Indoor Scene RGB-D Datasets. RGB-D image datasets of real indoor scenes [1, 5, 50, 58, 70] have driven advances in 3D perception and holistic scene understanding, with recent additions like ARKitScenes [1] and ScanNet++ [70] offering dense semantic and 3D annotations. Despite having minimum domain gaps with respect to real-world applications, these real datasets are expensive to annotate and inherently static, limiting users' ability to generate images from novel camera views, acquire new types of annotations, or alter scenes. Our work complements these limitations by offering a fully customizable generator for photorealistic synthetic data.

Dataset Category	Customizability				Visual Quality
	Camera	Obj. Pose	Obj. State	Toolkit	
Real RGB-D Datasets	✗	✗	✗	N/A	Real
3D Reconstruction Datasets	✓	✗	✗	N/A	Medium
Synthetic Datasets	✓	✓	✗	∼	High
3D Simulators	✓	✓	✓	✗	Low
BEHAVIOR Vision Suite (ours)	✓	✓	✓	✓	High

Table 1. Comparison of real and different types of synthetic datasets with BEHAVIOR Vision Suite. 'Camera' denotes the ability to render images from any viewing angle. 'Obj. Pose' refers to the modifiability of the object layout. 'Obj. State' indicates whether an object's physical states (e.g., open/close, folded) and semantic states (e.g., cooked, soaked) can be modified. 'Toolkit' indicates the availability of utility functions for sampling object layout and camera poses under specified conditions (e.g., viewing half-open kitchen cabinets filled with grocery items). 'Visual Quality' evaluates the photorealism of rendered images.

3D Reconstruction Datasets. 3D reconstruction datasets such as Gibson, Matterport, and HM3DSem [4, 53, 66, 69] allow the rendering of novel views. While these datasets have tremendously benefited the embodied navigation community, their utility for broader computer vision applications remains limited. Each scene, being a single 3D mesh, restricts further customization, such as modifying the object layout. Moreover, the visual quality of rendered novel views depends on the reconstruction's fidelity, often resulting in artifacts. While Taskonomy [71] and Omnidata [10] have extended mid-level visual cues such as surface normal for these datasets, semantic label acquisition remains expensive. In contrast, our work offers the flexibility to generate images with customized object layouts with consistent visual quality, while also providing comprehensive labels at no additional cost.

Synthetic Datasets. Synthetic datasets offer an alternative approach that eliminates the need for manual semantic labeling by rendering realistic images from interior scenes composed of independent object models [6, 8]. Object layouts are usually created by artists [13, 39, 55] or parsed from real scans [40], offering semantic realism of the scenes [19, 64]. Methods like OpenRooms [40] and Unity Synthetic Homes [61] also allow users to configure rendering options, such as lighting. However, despite their photorealism, the rendered images often lack physical plausibility, with common issues like object penetration or slight levitation. In addition, the object models are mostly fully rigid and support very limited semantic states. Our generator not only ensures the physical plausibility of images created, but also supports broader relationship customization (e.g., "cooked" or "filled") and more granular control over the sampled state, such as openness level through joint limit annotations.

3D Simulators. A large number of 3D simulators with physical realism have been developed recently. Kubric [24] focuses on generating physically plausible object clusters without full-scene simulation. iGibson [36, 56] and Habitat 2.0 [60] offer reconfigurable indoor scenes with articulated assets, the former notably supporting extended object states such as wetness level. ThreeDWorld [14] emphasizes physical prediction, especially with non-rigid objects. ProcTHOR [7] automates the large-scale generation of semantically plausible virtual environments. Since these 3D simulators cater to the embodied AI and robotics community, their visual quality is often not prioritized. In contrast, we use OmniGibson, a new simulator that surpasses the photorealism of the aforementioned ones, according to a user study [37], positioning our work as more suitable for computer vision research. Moreover, we provide a range of utility functions in this work, allowing for easy creation of diverse images tailored to specific needs—a feature most existing 3D simulators lack.

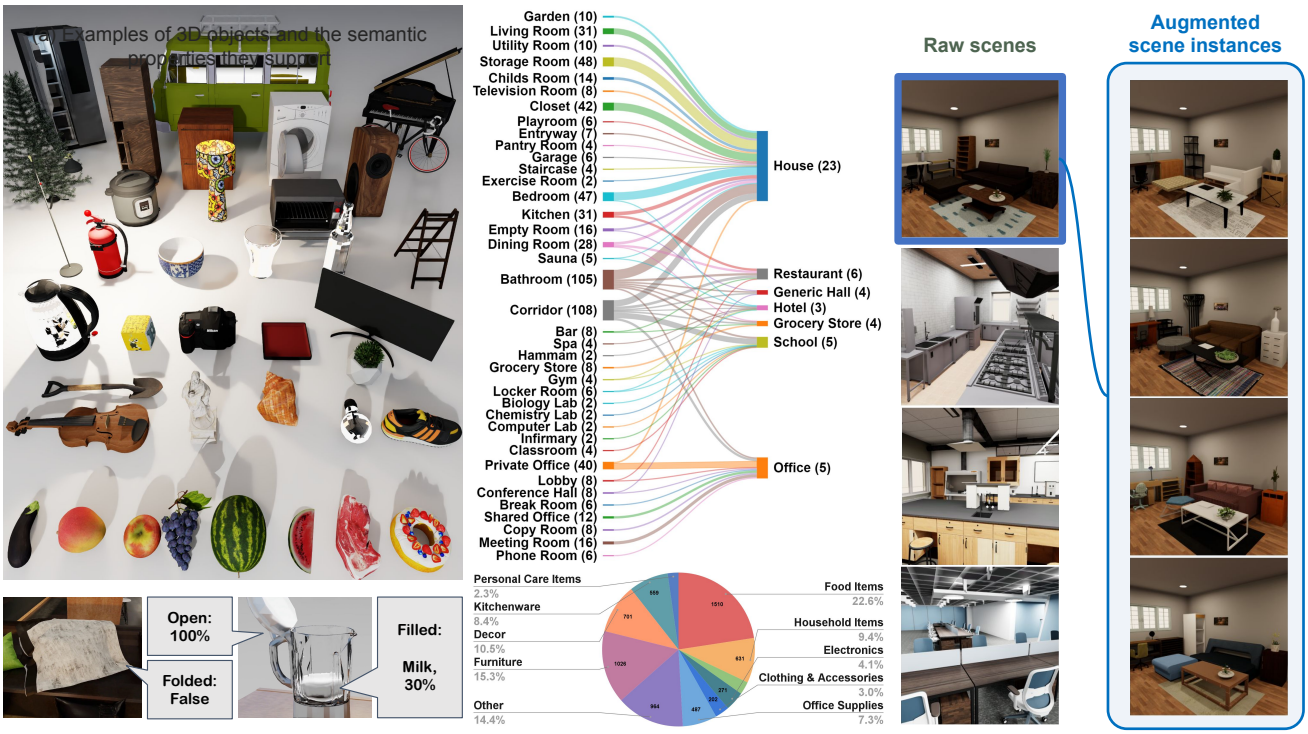
3. BEHAVIOR Vision Suite

BEHAVIOR Vision Suite contains two main components (Fig. 1): the extended BEHAVIOR-1K assets and the customizable dataset generator. The assets serve as the foundation, while the generator leverages these assets to create vision datasets tailored to downstream tasks of interest.

3.1. Extended BEHAVIOR-1K Assets

The extended BEHAVIOR-1K assets comprises a diverse collection of 8,841 object models and 1,000 scene instances, derived from 51 artist-designed raw scenes. Of these objects, 2,156 are structural elements like walls, floors, and ceilings, while the remaining 6,685 nonstructural items span 1,937 categories, including food, tools, electronics, clothing, and office supplies, among others. This categorization is detailed in Fig. 2. Predominantly indoor, the 51 raw scenes also incorporate outdoor elements such as gardens and encompass a wide variety of environments: houses (23), offices (5), restaurants (6), grocery stores (4), hotels (3), schools (5), and generic halls (4), as well as a simulated twin of a mock apartment in our research lab. This collection of assets is the result of a year-long effort to extend the BEHAVIOR-1K [37] assets to enhance their applicability in computer vision.

We expanded the object collection from 5,215 to 8,841 by adding more everyday objects, segmenting building structures into individual objects for more precise 3D bounding box labels, and procedurally generating sliced food. In addition, we have developed functionality that enables the generation of diverse scene variations by altering furniture object models and incorporating additional everyday objects. We will release 1000 scene instances augmented from the 51 raw scenes.



(b) Distribution of scenes, room types and objects

(c) Examples of raw scenes and augmented scene instances

Figure 2. Overview of extended BEHAVIOR-1K assets: Covering a wide range of object categories and scene types, our 3D assets have high visual and physical fidelity and rich annotations of semantic properties, allowing us to generate 1,000+ realistic scene configurations.

To improve physical realism, we refined collision meshes using V-HACD [45] and CoACD [65], manually selecting the best parameters to ensure a balance between physical accuracy, affordance preservation, and simulation efficiency. For more than 2,000 objects, where this method was insufficient, we manually designed their collision meshes.

We enhanced lighting realism by annotating actual light source objects, such as lamps and ceiling lights, to mimic real-world illumination. For more detailed semantic properties, we annotated appropriate container fillable volumes (e.g., cups, pots) and fluid source/sink locations (e.g., faucets, drains, sprayers), enabling us to spawn fluids in the scene realistically. Scene objects were annotated if they cannot be freely moved, e.g., when they physically support other objects. Cluttered objects were distinctly annotated, allowing them to be replaced with alternative clutters.

Altogether, we designed the assets to form a strong basis for custom data generation (discussed in §3.2), with a functional organization that allows accurate object randomization, and the annotations to provide a large number of modifiable parameters at both the object and scene levels.

3.2. Customizable Dataset Generator

The customizable dataset generator, the software component of the BEHAVIOR Vision Suite, is designed to generate synthetic datasets tailored to particular specifications. Built on OmniGibson [37], it leverages NVIDIA Omniverse's photorealistic, real-time renderer and OmniGibson's procedural sampling functions for object states to generate custom images and videos that satisfy arbitrary requirements. The produced datasets include rich, comprehensive annotations–segmentation masks, 2D/3D bounding boxes, depth, surface normals, flows, and point clouds–at no additional cost. Crucially, it empowers users with extensive control over the dataset generation process, allowing them to specify requirements on scene layouts, object states, camera angles, and lighting conditions, all while ensuring physical plausibility through the physics engine.

Capabilities. The generator has the following capabilities:

- *Scene Object Randomization:* It can swap scene objects with alternative models from the same category, which are grouped based on visual and functional similarities. This randomization significantly varies scene appearances while maintaining layouts' semantic integrity.
- *Physically Realistic Pose Generation:* The generator can procedurally change the physical states of objects to satisfy certain predicates. This includes 1) placing objects with respect to other objects in the scene in a certain way (e.g., inside, on top of, or under), 2) opening or closing articulated objects, 3) filling containers with fluids, and

4) folding or unfolding pieces of cloth. The generator can generate multiple valid configurations for the same predicate and ensures physical plausibility.

- *Predicate-Based Rich Labeling:* Beyond usual labels (semantic & instance segmentation, bounding boxes, surface normals, depth, etc.), the generator also provides annotations including unary states of an object (e.g., whether an articulated object is open, or an appliance is toggled on), binary predicates between two objects (e.g., if one is touching, on top of, next to another) or between an object and a substance (e.g., if an object is filled/covered/soaked with a substance), and continuous labels (e.g., joint openness for articulated objects, filled fraction for containers).
- *Camera Pose and Trajectory Sampling:* Finding proper camera pose in a 3D scene is a challenging but crucial step in the rendering pipeline: the camera shall not be occluded and points at the subject of interest. The generator uses occupancy grids and hand-crafted heuristics to generate both static camera poses and plausible traversal trajectories that satisfy these constraints to curate image or scene traversal video datasets.
- *Configurable Rendering:* Through a user-friendly API, the generator allows for the customization of rendering parameters, including lighting and camera specifics such as aperture and field of view.

Dataset Generation. Images in the BVS dataset can be generated as follows. First, we select one of the 51 raw scenes from the user-configured scene category (say, an office). Scene objects are randomized with instances from the same category. Depending on the user configuration, we determine additional objects to add to the scene. We place the objects using the pose generation capabilities based on user-specified requirements. This might include cluttering certain areas (e.g., filling a fridge with perishables) or individually manipulating object states (e.g., making a cabinet open or a table covered with water) for predicate prediction.

We then generate a camera pose (or a sequence of poses as a camera trajectory), as well as randomize the scene's lighting parameters and the camera's intrinsics based on the user's specifications. Finally, we render an image (or a sequence of images) and record it alongside all relevant labels requested by the user, including additional modalities (depth/segmentation/etc.), bounding boxes, and predicate and object state values.

4. Applications and Experiments

We present three applications and corresponding experiments to demonstrate the utility of BVS: first, systematically evaluating model robustness against various continuous domain shifts, such as the lighting condition (§4.1); second, assessing various scene understanding models using a consistent set of images with comprehensive annotations (§4.2); and third, training a model for a new vision task, object states and relations prediction, on synthesized data and evaluating its simulation-to-real transfer capability. (§4.3).

Axis	# scenes	# video clips
Articulation	17	237
Lighting	16	441
Visibility	14	211
Zoom	9	215
Pitch	16	268

Table 2. We generate up to 200–500 short video clips with diverse scene configurations for parametric evaluation (§4.1). Each video clip varies along one axis of distribution shift with a single target object. On average, each video has 300 frames.

4.1. Parametric Model Evaluation

Parametric model evaluation is essential for developing and understanding perception models, enabling a systematic assessment of performance robustness against various domain shifts. Previous efforts, such as 3DCC [32], have explored image corruption generation using 3D information, yet their scope is constrained by the static nature of input meshes, limiting the type and extent of possible variations. Leveraging the flexibility of the simulator, our generator extends parametric evaluation to more diverse axes, including scene, camera, and object state changes.

Task Design and Dataset Generation. We focus on five key parameters difficult to rigorously control in real-world datasets yet significantly influence model performance: object articulation, lighting, object visibility, camera zoom, and camera pitch. Each parameter varies along a continuous axis for evaluating baseline models. For instance, object visibility varies from fully occluded to fully visible.

We generate 200 to 500 videos for each axis (Tab. 2), using our collection of more than 8,000 3D assets. Each video includes a target object with changes focused on a single parameter under examination. Fig. 3 shows examples of target objects with variations along each axis. We maintained consistency in other aspects of the environment, systematically synthesizing images to isolate the main parameter's impact.

To validate our findings in real-world conditions and further assess the sim2real transfer capability of parametric evaluation, we collected a smaller-scale real dataset for each of the 5 axes and replicated the evaluation. For setup details and additional results, please refer to the appendix.

Baselines and Metrics. We explore two vision tasks: *open-vocabulary detection* and *open-vocabulary segmentation*, hypothesizing that models for these tasks may be sensitive to object-centric domain shifts. For baselines, we select the current state-of-the-art (SOTA) models on real datasets: GLIP [38], RAM [73], and Grounding DINO [44] for detection, and ODISE [72], OpenSeeD [68], and Grounding SAM [29] for segmentation.

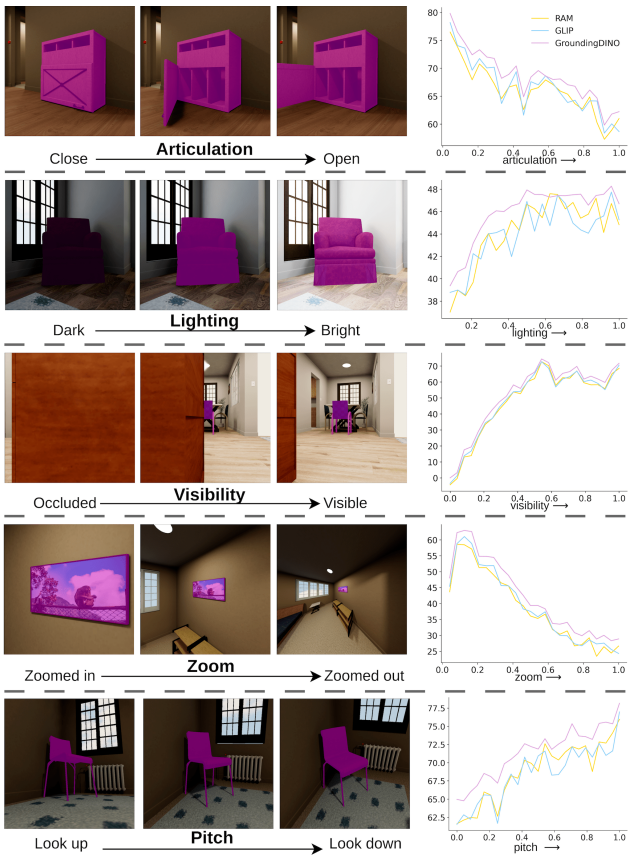


Figure 3. Parametric evaluation of object detection models on five example video clips. Selected frames from these clips are shown on the left, with the target object highlighted in magenta. Average Precisions (APs) for our baseline models in §4.2 are plotted on the right. Since BVS allows for full customization of scene layout and camera viewpoints, we can systematically evaluate model robustness against variations in object articulation, lighting conditions, visibility, zoom (object proximity), and pitch (object pose). As illustrated, current SOTA models exhibit limited robustness to these axes of variation.

Results and Analysis. In Fig. 3 and Fig. 4, we present example images when varying each parameter as well as respective detection Average Precision (AP) performance. AP, calculated exclusively for the target object (highlighted in magenta), assesses the model's recognition accuracy. Detailed analyses reveal the following:

• *Articulation* varies the joint angles of the articulated target object, from fully closed to fully open, including processes such as opening/closing drawers or doors, and folding/unfolding laptops. A notable negative correlation between the degree of articulation and model performance suggests that models, typically trained or evaluated on existing benchmarks featuring mostly closed articulated objects (e.g., closed washing machines and microwaves), struggle with recognizing objects in open states.

• *Lighting* adjusts global illumination of the environment from dark to bright. We observed improving model performance up to a midpoint brightness level of 0.5, beyond which it plateaus. This suggests that, while current models suffer from low-light conditions, their performance saturates once the brightness level surpasses a certain threshold.

Figure 4. Mean performance of open-vocab object detection and segmentation models across five axes. The larger a model's colored envelope, the more robust it is. Through BVS, new vision models can be systematically tested for their robustness along these five dimensions and beyond: our users can easily add new axes of domain shift with just a few lines of code.

• *Visibility* shifts the visibility of the target object from fully occluded to fully visible, which is computed as the ratio of visible to total pixels of the target object. We observe a steep decline in model performance as visibility drops below 0.5, which highlights a significant opportunity to enhance model robustness to partial occlusions.

• *Zoom* controls camera zoom from zoomed-in to zoomed-out. Results show that extremely close views, where a partial view of the target object occupies the entire image, hinder performance due to a lack of contextual information. In contrast, too-zoomed-out views make the object too small for models to detect it effectively. Optimal performance is achieved at moderate zoom levels.

• *Pitch* varies camera pitch from looking up to looking down. We find that models perform inconsistently with seemingly benign changes in camera viewpoint, generally showing improved performance when the camera looks down at the target object. One potential explanation is that objects in large-scale real datasets are often captured from above, making this perspective more familiar to the models.

To summarize, we observe significant performance discrepancies across three models on all five axes, with our parallel experiments in real settings (see the appendix) confirming that these trends observed in synthetic data mirror those in real-world scenarios. This underscores the lack of robustness of the current SOTA models in extreme or out-of-distribution test environments. By generating large-scale synthetic datasets with controlled variability, BVS provides a unique and powerful test bed to evaluate model performance. Furthermore, consistent with §4.2, the relative performance remains steady across the five axes, highlighting

Figure 5. Holistic Scene Understanding Dataset. We generated extensive traversal videos across representative scenes, each with 10+ camera trajectories. For each image, BVS generates various labels (e.g., scene graphs, segmentation masks, depth) as shown on the right.

Open-vocab Det.	AP ↑	AP_{small} ↑	AP_{medium} ↑	AP_{large} ↑	COCO (AP)
GLIP [38]	41.4	7.0	27.5	61.8	60.8
RAM [73]	41.3	6.4	27.8	63.9	61.4
Grounding DINO [44]	**44.7**	**11.9**	**31.2**	**66.3**	63.0

Open-vocab Seg.	AP ↑	AP_{small} ↑	AP_{medium} ↑	AP_{large} ↑	ADE (AP)
ODISE [68]	57.1	41.0	53.2	65.0	13.9
OpenSeeD [72]	57.3	42.0	54.1	64.8	15.0
Grounding SAM [29]	**59.2**	**42.9**	**54.4**	**65.1**	14.8

Depth Est.	RMS ↓	AbsRel ↓	Log10 ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑	NYUv2 (δ_1)
DPT [54]	0.66	0.14	0.05	0.09	0.15	0.20	0.90
NVDS [63]	0.58	**0.13**	**0.04**	0.10	0.15	0.21	0.93
iDisc [51]	**0.49**	**0.13**	**0.04**	**0.12**	**0.19**	**0.22**	0.94

Point Cloud Recon.	Comp. Ratio ↑	Comp. ↑	Acc. ↓	Replica (C.R.)
GradSLAM [31]	50.0	**14.8**	29.8	67.9
NICE-SLAM [74]	**66.3**	12.0	**23.5**	89.3

Table 3. A comprehensive evaluation of SOTA models on four vision tasks. Our synthetic dataset can be a faithful proxy for real datasets as the relative performance between different models closely correlates to that of the real datasets.

the predictive value of the datasets generated by BVS.

4.2. Holistic Scene Understanding

One of the major advantages of synthetic datasets, including BVS, is that they offer various types of labels (segmentation masks, depth maps, and bounding boxes) for the same sets of input images. We believe that this feature can fuel the development of versatile vision models that can perform multiple perception tasks at the same time in the future. Since such models are not currently available, we instead evaluate the current SOTA methods on a subset of the tasks that BVS supports (see below). This will also serve as a validation of the photorealism of our datasets, i.e., models trained on real datasets should perform reasonably without fine-tuning.

Task Design and Dataset Generation. Equipped with BVS's powerful generator (see §3.2), we generated 100+ full scene traversal videos with a total of 266240 frames with per-frame ground truth annotations in multiple modalities. Fig. 5 shows an overview of the generated dataset.

Baselines and Metrics. In Tab. 3, we assess 11 models in four tasks. Specifically, we consider *Detection* and *Segmentation* tasks, both in the challenging open vocabulary setting [44, 72], as well as *Depth Estimation* and *Point Cloud Reconstruction*, with standard metrics used.

Results and Analysis. We summarize all our evaluation results in Tab. 3. We observe that the relative performance of these models on our synthetic dataset has a high correlation with that on real datasets such as MS COCO [42] or NYUv2 [50], indicating that our generated synthetic datasets can be a faithful proxy for real datasets.

In summary, we provide a comprehensive benchmark to score and understand a wide range of existing models for each of the four tasks on exactly the same images. Although most current vision models focus on a single output modality, we hope BEHAVIOR Vision Suite could motivate researchers and practitioners to develop versatile models that concurrently predict multiple modalities in the future, where our benchmarking results for single-task SOTA methods in this section could serve as a useful reference.

4.3. Object States and Relations Prediction

BVS's capabilities extend beyond model evaluation shown in §4.1 and §4.2. Users can also leverage BVS to generate training data with specific object configurations that are difficult to accumulate or annotate in the real world. This section illustrates BVS's practical application in synthesizing a dataset that facilitates the training of a vision model capable of zero-shot transfer to real-world images on the task of object relationship prediction [3, 15–18, 26]. Additional

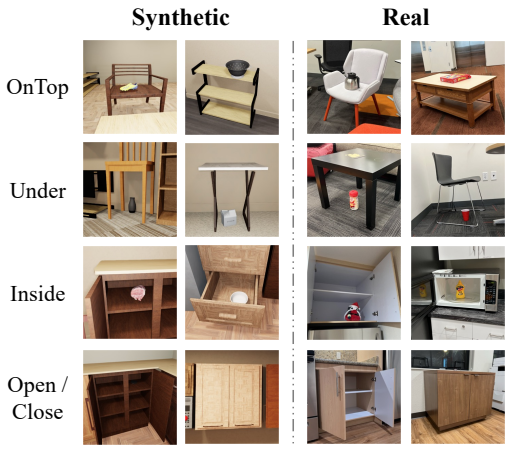


Figure 6. Sample images of each class from our generated synthetic and collected real datasets.

Test on		Open	Close	Ontop	Inside	Under	Avg.
Synthetic	Precision	0.962	0.897	0.947	0.989	0.874	0.932
	Recall	0.822	0.978	0.913	0.995	0.949	0.929
Real	Precision	0.943	0.958	0.545	0.906	0.948	0.863
	Recall	0.757	0.915	0.913	0.776	0.703	0.817

Table 4. Classification results on held-out synthetic eval set and real test set for our method adapted from [48].

Method	Precision	Recall	F1
Zero-shot CLIP	0.293	0.282	0.271
Ours	**0.863**	**0.817**	**0.839**

Table 5. Classification results on the real test set. Task-specific training on synthetic data boosts performance on real images.

experiments focusing on unary object states, `filled` and `folded`, are detailed in the appendix.

Task Design and Dataset Generation. Predicting object relationships, such as `open` and `inside`, is a crucial yet challenging perception task due to the difficulties in collecting such data in the real world, let alone the costly annotations[35, 46, 48]. We use our generator to synthesize 12.5k images with five labels (`open`, `close`, `ontop`, `inside`, `under`), depicting relationships between target objects. We also collected and labeled 910 real images with unseen object instances and scenes to test sim2real performance. Examples are shown in Fig. 6.

Baselines and Metrics. Adapting from [48], our model takes an image and target objects' bounding boxes as input, and outputs a five-way classification over the five labels. We specifically define `open/close` as a binary relationship between the movable link and the unmovable base of an articulated object, enabling fine-grained articulation state assessment. For example, the model can be queried for the open or closed status of individual drawers of a cabinet. Detailed model architecture is available in the appendix.

We compare our model with zero-shot CLIP, which is not trained on our synthetic dataset, in terms of precision, recall, and F1, on the synthetic evaluation set and the real test set. Specifically, by harnessing CLIP's zero-shot capabilities [52], this baseline outputs a five-way classification prediction by comparing the image embeddings with the five verbalized prompts' text embeddings.

Results and Analysis. Tab. 4 presents the quantitative results on the held-out synthetic dataset and the real dataset for our method. Although there is some performance gap, our model trained on only synthetic data can transfer to real images with promising overall accuracy. Additionally, unary state prediction experiments, detailed in the appendix, also reveal high accuracy in both domains. These results underscore that BVS offers a promising way to obtain realistic synthetic data that researchers can use not only for evaluation (as shown in §4.1 and §4.2), but also for training models that can then be transferred to the real world. In fact, from Tab. 5, we observe that task-specific training on synthetic data is a very effective method to obtain good performance on real images.

5. Conclusion

We have introduced the BEHAVIOR Vision Suite (BVS), a novel toolkit designed for the systematic evaluation and comprehensive understanding of computer vision models. BVS enables researchers to control a wide range of parameters across scene, object, and camera levels, facilitating the creation of highly customized datasets. Our experiments highlight BVS's versatility and efficacy through three key applications. First, we show its ability to evaluate model robustness against various domain shifts, underscoring its value in systematically assessing model performance under challenging conditions. Second, we present comprehensive benchmarking of scene understanding models on a unified dataset, illustrating the potential for developing multi-task models using a single BVS dataset. Lastly, we investigate BVS's role in facilitating sim2real transfer for novel vision tasks, including object states and relations prediction. BVS highlights synthetic data's promise in advancing the field, offering researchers the means to generate high-quality, diverse, and realistic datasets tailored to specific needs.

Acknowledgments. We are grateful to SVL members for their helpful feedback and insightful discussions. The work is in part supported by the Stanford Institute for Human-Centered AI (HAI), NSF CCRI #2120095, RI #2338203, ONR MURI N00014-22-1-2740, N00014-21-1-2801, Amazon, Amazon ML Fellowship, and Nvidia.

References

[1] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. 2

[2] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015. 1

[3] Paola Cascante-Bonilla, Khaled Shehada, James Seale Smith, Sivan Doveh, Donghyun Kim, Rameswar Panda, Gul Varol, Aude Oliva, Vicente Ordonez, Rogerio Feris, et al. Going beyond nouns with vision & language models using synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20155–20165, 2023. 7

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 1, 3

[5] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 2

[6] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. *arXiv preprint arXiv:2212.08051*, 2022. 3

[7] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied ai using procedural generation. *Advances in Neural Information Processing Systems*, 35:5982–5994, 2022. 2, 3

[8] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. *arXiv preprint arXiv:2307.05663*, 2023. 3

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[10] Ainaz Eftekhar, Alexander Sax, Roman Bachmann, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans, 2021. 3

[11] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010. 1

[12] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *arXiv preprint arXiv:2303.11331*, 2023. 19

[13] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, 129:3313–3337, 2021. 1, 3

[14] Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, et al. Threedworld: A platform for interactive multi-modal physical simulation. *arXiv preprint arXiv:2007.04954*, 2020. 2, 3

[15] Yunhao Ge, Harkirat Behl, Jiashu Xu, Suriya Gunasekar, Neel Joshi, Yale Song, Xin Wang, Laurent Itti, and Vibhav Vineet. Neural-sim: Learning to generate training data with nerf. In *European Conference on Computer Vision*, pages 477–493. Springer, 2022. 7, 19

[16] Yunhao Ge, Jiashu Xu, Brian Nlong Zhao, Laurent Itti, and Vibhav Vineet. Em-paste: Em-guided cut-paste with dall-e augmentation for image-level weakly supervised instance segmentation, 2022.

[17] Yunhao Ge, Jiashu Xu, Brian Nlong Zhao, Neel Joshi, Laurent Itti, and Vibhav Vineet. Dall-e for detection: Language-driven compositional image synthesis for object detection. *arXiv preprint arXiv:2206.09592*, 2022.

[18] Yunhao Ge, Jiashu Xu, Brian Nlong Zhao, Neel Joshi, Laurent Itti, and Vibhav Vineet. Beyond generation: Harnessing text to image models for object detection and segmentation. *arXiv preprint arXiv:2309.05956*, 2023. 1, 7, 19

[19] Yunhao Ge, Hong-Xing Yu, Cheng Zhao, Yuliang Guo, Xinyu Huang, Liu Ren, Laurent Itti, and Jiajun Wu. 3d copy-paste: Physically plausible object insertion for monocular 3d detection. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[20] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 1

[21] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 18

[22] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017. 1

[23] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Pro-*

ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 18995–19012, 2022. 1

[24] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: a scalable dataset generator. 2022. 3

[25] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617, 2018. 1

[26] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and XIAOJUAN QI. Is synthetic data from generative models ready for image recognition? In *The Eleventh International Conference on Learning Representations*, 2022. 7

[27] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021. 1

[28] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15262–15271, 2021. 1

[29] IDEA-Research. Grounding sam. https://github.com/IDEA-Research/Grounded-Segment-Anything, 2023. 5, 7

[30] Sho Inayoshi, Keita Otani, Antonio Tejero-de Pablos, and Tatsuya Harada. Bounding-box channels for visual relationship detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 682–697. Springer, 2020. 18

[31] Krishna Murthy Jatavallabhula, Soroush Saryazdi, Ganesh Iyer, and Liam Paull. gradslam: Automagically differentiable slam. *arXiv preprint arXiv:1910.10672*, 2019. 7

[32] Oğuzhan Fatih Kar, Teresa Yeo, Andrei Atanov, and Amir Zamir. 3d common corruptions and data augmentation, 2022. 5

[33] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1, 18

[34] 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

[35] 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. *Int J Comput Vis*, 123:32–73, 2017. 1, 8

[36] Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, et al. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. *arXiv preprint arXiv:2108.03272*, 2021. 2, 3

[37] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, Mona Anvari, Minjune Hwang, Manasi Sharma, Arman Aydin, Dhruva Bansal, Samuel Hunter, Kyu-Young Kim, Alan Lou, Caleb R Matthews, Ivan Villa-Renteria, Jerry Huayang Tang, Claire Tang, Fei Xia, Silvio Savarese, Hyowon Gweon, Karen Liu, Jiajun Wu, and Li Fei-Fei. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Proceedings of The 6th Conference on Robot Learning*, pages 80–93. PMLR, 2023. 1, 2, 3, 4

[38] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975, 2022. 5, 7

[39] Wenbin Li, Sajad Saeedi, John McCormac, Ronald Clark, Dimos Tzoumanikas, Qing Ye, Yuzhong Huang, Rui Tang, and Stefan Leutenegger. Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. *arXiv preprint arXiv:1809.00716*, 2018. 1, 3

[40] Zhengqin Li, Ting-Wei Yu, Shen Sang, Sarah Wang, Meng Song, Yuhan Liu, Yu-Ying Yeh, Rui Zhu, Nitesh Gundavarapu, Jia Shi, et al. Openrooms: An end-to-end open framework for photorealistic indoor scene datasets. *arXiv preprint arXiv:2007.12868*, 2020. 1, 3

[41] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7083–7093, 2019. 1

[42] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 1, 7

[43] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 1

[44] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded

pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 5, 7, 15, 16

[45] Khaled Mamou. Volumetric approximate convex decomposition. In *Game Engine Gems 3*, chapter 12, pages 141–158. A K Peters / CRC Press, 2016. 4

[46] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *International Conference on Learning Representations*, 2018. 8

[47] Roberto Martin-Martin, Mihir Patel, Hamid Rezatofighi, Abhijeet Shenoi, JunYoung Gwak, Eric Frankel, Amir Sadeghian, and Silvio Savarese. Jrdb: A dataset and benchmark of egocentric robot visual perception of humans in built environments. *IEEE transactions on pattern analysis and machine intelligence*, 2021. 1

[48] Toki Migimatsu and Jeannette Bohg. Grounding predicates through actions. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 3498–3504, 2022. 8, 18

[49] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 18

[50] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 1, 2, 7

[51] Luigi Piccinelli, Christos Sakaridis, and Fisher Yu. idisc: Internal discretization for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21477–21487, 2023. 7

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 8, 18

[53] Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021. 1, 3

[54] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021. 7

[55] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10912–10922, 2021. 1, 3

[56] Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D'Arpino, Shyamal Buch, Sanjana Srivastava, Lyne Tchapmi, et al. igibson 1.0: A simulation environment for interactive tasks in large realistic scenes. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7520–7527. IEEE, 2021. 1, 2, 3

[57] Gunnar A. Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Actor and observer: Joint modeling of first and third-person videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[58] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[59] Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Elliott Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, Karen Liu, et al. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In *Conference on Robot Learning*, pages 477–490. PMLR, 2022. 2

[60] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in Neural Information Processing Systems*, 34:251–266, 2021. 2, 3

[61] Unity Technologies. Unity synthhomes: A synthetic home interior dataset generator. https://github.com/Unity-Technologies/SynthHomes, 2022. 3

[62] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013. 1

[63] Yiran Wang, Min Shi, Jiaqi Li, Zihao Huang, Zhiguo Cao, Jianming Zhang, Ke Xian, and Guosheng Lin. Neural video depth stabilizer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9466–9476, 2023. 7

[64] Zian Wang, Wenzheng Chen, David Acuna, Jan Kautz, and Sanja Fidler. Neural light field estimation for street scenes with differentiable virtual object insertion. In *European Conference on Computer Vision*, pages 380–397. Springer, 2022. 3

[65] Xinyue Wei, Minghua Liu, Zhan Ling, and Hao Su. Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search. *ACM Transactions on Graphics (TOG)*, 41(4):1–18, 2022. 4

[66] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018. 1, 3

[67] 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

[68] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2955–2966, 2023. 5, 7

[69] Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, et al. Habitat-matterport 3d semantics dataset. *arXiv preprint arXiv:2210.05633*, 2022. 3

[70] 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

[71] Amir R. Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018. 3

[72] Hao Zhang, Feng Li, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, and Lei Zhang. A simple framework for open-vocabulary segmentation and detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1020–1031, 2023. 5, 7

[73] Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, et al. Recognize anything: A strong image tagging model. *arXiv preprint arXiv:2306.03514*, 2023. 5, 7

[74] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022. 7

[75] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 2023. 1

BEHAVIOR Vision Suite: Customizable Dataset Generation via Simulation

Supplementary Material

A. Extended BEHAVIOR-1K Assets

Fig. 7 shows more examples of Extended BEHAVIOR-1K Assets (main paper Sec.3.1): (a-f) examples for different main categories. (g) examples of improved collision mesh quality. (h) examples of articulation objects. (i) examples of different light sources, (j) examples of fillable volumes for containers.

B. Customizable Dataset Generator

We prepare a **video** in the supplementary material to show an example visualization of the Customizable Dataset Generator: specifically, we show the scene instance augmentation by scene object (furniture) randomization and inserting additional everyday objects.

C. Experiments

C.1. Parametric Model Evaluation

Details of Dataset Generation Process. We synthesized the evaluation videos for each axis (Object articulation, Lighting, Visibility, Zoom, Pitch) according to the following pipeline.

As shown in the main paper Sec. 4.1, each video includes a target object with changes focused on a single parameter under examination. First, we sample one of the scene instances and randomly choose a target object in the scene. For the object articulation axis, we only sample objects with movable parts, such as cabinets, microwaves, refrigerators, etc. Next, we sample a random camera angle and distance with the target object placed in the center. Then, for all except pitch, we keep this camera pose and perform an axis-specific manipulation to generate a video with the desired variation:

- **Object articulation**: We linearly interpolate the joint angle from being closed to fully open, utilizing the joint maximum range annotations provided in BVS assets. We record the image with the joint in each intermediate state. For objects with multiple movable parts, e.g., a cabinet with three drawers, we randomly sample a subset of joints to manipulate and keep the rest closed.
- **Lighting**: We linearly increase the intensity of all indoor light sources in the scene simultaneously.
- **Visibility**: There are three key components in the visibility (occlusion) setting: camera, target object, and occluding object. We first set the camera centering on the target object, then we place an occluding object (relatively large object, e.g., cabinet) in the line between the camera and the target object, fully occluding the target in view. Then, we fix the distance between the camera and the target object and move the camera around the target object until the target is fully visible. The visibility score (number of visible pixels/number of total pixels) of each frame is calculated by rendering the video again and removing the occluding cabinet. Although the object orientation in camera view might slightly change since the camera is not static, we implemented the following practices to eliminate the effect of this factor. First, we set the camera relatively far from the target but occluding objects close to the camera, allowing minimal camera pose change needed to capture the "fully occluded to fully visible" process. In addition, the initial object pose is randomized, so when we average evaluation performance, the effect of this factor shall largely cancel out.
- **Zoom**: With the camera pose fixed, we change its focal length to model the zooming effect. We strategically changed the focal length such that the resulting video illustrates an approximately linear zooming behavior. We always make the target object at the center of the view, and it remains mostly unaffected by distortion, even under extreme focal length.
- **Pitch**: We linearly change the camera pitch angle while keeping the original camera distance as well as the yaw angle unmodified.

After each video was collected, we filtered out the videos where the target object was not properly visible. The detailed statistics for each axis is shown in main Table 2.

Metric Details. Each generated video contains exactly one target object (main paper Figure 3 magenta). We use different open-vocabulary object detection and segmentation models to detect or segment the target object. These models act as indicators of performance in challenging environments, such as those with limited lighting or long-distance zoom. Therefore, we compute the Average Precision (AP) metric using the target object as the sole ground truth, considering only predictions that classify the target object class. However, it is plausible to encounter scenarios where multiple objects of the same category as the target object exist. For instance, in a video, there might be several chairs, and the target object is one of those chairs. Models might have correctly detected all chairs, but since only one is the ground truth, all rest will be marked as incorrect. To counter such an undesired situation, we employ a simple heuristic to filter predictions for the non-target object: For non-target objects sharing the same category, we calculate the IoU with each prediction and exclude those with an IoU exceeding a predefined threshold of 0.3. This means treat-

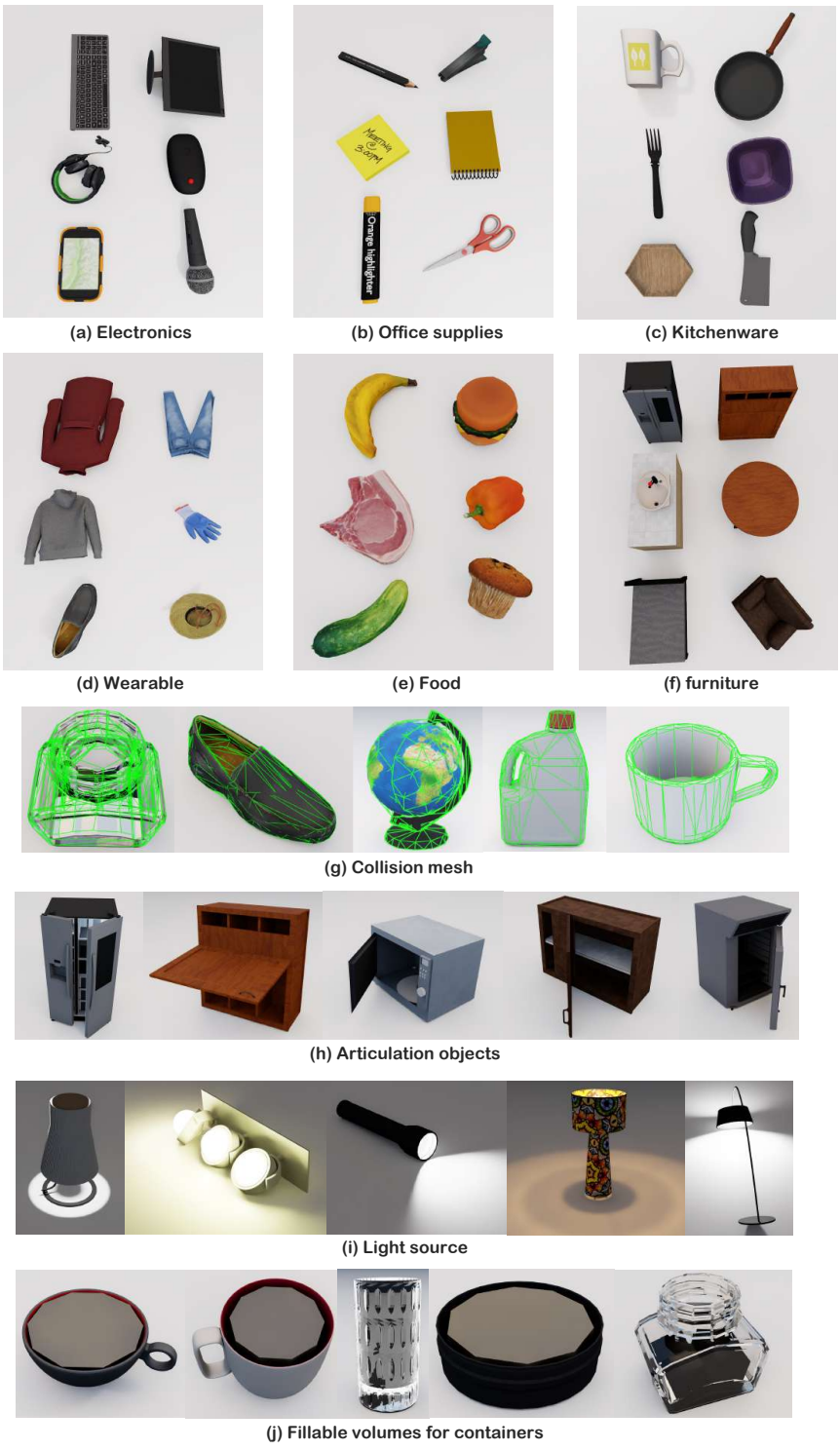


Figure 7. More examples of Extended BEHAVIOR-1K Assets: (a-f) examples for different main categories. (g) examples of improved collision mesh quality. (h) examples of articulation objects. (i) examples of different light sources, (j) examples of fillable volumes for containers.

ing these predictions as valid for non-target objects rather than false positives. This threshold is chosen empirically based on a few selected cases where objects of the same category are densely packed together. We believe this choice generalizes well to less crowded scenes and ensures the reliability of our evaluation process.

Failure Case Analysis on Five Evaluation Parameters. In Fig. 8, we present failure case examples of Grounding DINO [44] across five evaluation axes: Object articulation, Lighting, Visibility, Zoom, and Pitch. Each row in our presentation represents one axis and comprises four example groups. In each group, there are two images: the left image illustrates the ground truth, highlighting the target object in magenta, while the right image shows the Grounding DINO's prediction for the target object, as indicated at the top of the first image in each group. The example groups are arranged such that, from left to right, the intensity along the respective axis increases (e.g., progressing from zoomed in to zoomed out), the intensity value (0-1) is shown on top of each prediction. We find and highlight some interesting findings for each parametric evaluation in Fig. 8 and detailed below. For the full axis, we prepare a **video** in supplementary to show qualitative examples about running different detection models on the same video.

- **Articulation.** The model may have limited exposure to open states for articulation objects, which makes it less likely to predict a microwave with its door open correctly.
- **Lighting.** When the environment is dark, the model performance is negatively affected. However, when the lighting exceeds a certain threshold, in this case 0.5, the model becomes robust to increasing illumination.
- **Visibility.** The model's detection performance suffers when most of the target object is occluded. Surprisingly, a correct prediction can be made with only half of the object visible.
- **Zoom.** When the model is zoomed out, nearby objects become distorted, leading the model to identify irrelevant objects as the target mistakenly. This suggests that the model's recognition may rely partly on contours rather than solely on semantic information.
- **Pitch.** We find that, generally, the model can achieve better performance in a look-down angle compared to a look-up angle.

Segmentation Results on Five Axes. Figure 3 of the main paper shows the performance of open-vocabulary detection models on five axes. In Fig. 9, we show the performance of open-vocabulary segmentation models instead. The average performance for each axes corresponds to one angle in the radar plot (main Figure 4). Observations in main section 3.1 also apply to segmentation tasks.

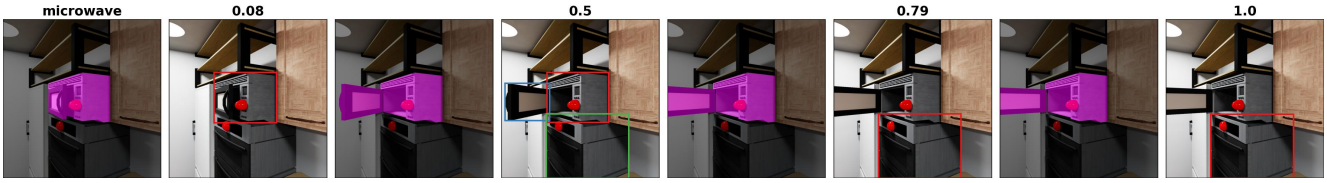
Real Experiment Setup and Results In order to evaluate the sim2real transfer capability of parametric evaluation results, we curated a set of real images to perform the same evaluation. In total, we manually collected 430 images of 15 to 22 objects from various categories. For each axis, we took photo of each target object with 5 levels of the corresponding distribution shift that matches the intensity level 0, 0.25, 0.5, 0.75, 1 in the synthetic data. For example, when collecting data for a microwave object for the articulation axis, we collect 5 images of the microwave being fully closed, 25% open, half open, 75% open, and fully open. An example object from the real dataset (and the comparison to the most similar counterpart in simulation) is shown in . For the zoom axis, due to the limited focal length range of our real cameras, we only covered intensity levels 0 to 0.3.

We evaluated the SOTA detection methods with manually labeled bounding boxes. Fig. 10 shows that under different types of distribution shift, the performance of the SOTA methods varies on real data just as it varies in simulation.

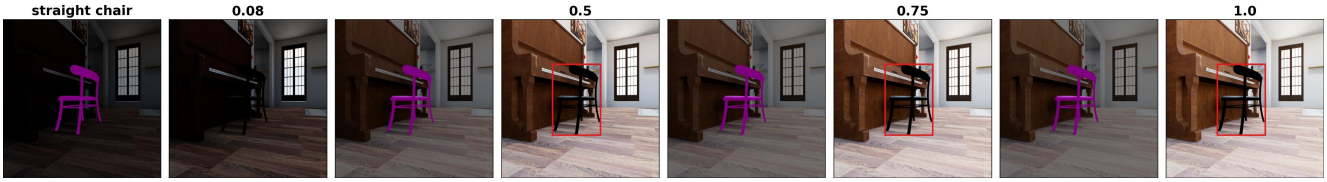
C.2. Holistic Scene Understanding (main paper Sec. 4.2)

Details of Generation Process. To generate a scene traversal video, we adhere to a standard process. Initially, we sample a scene instance and subsequently define a camera trajectory using the BVS toolkit. Following this, we render the traversal video, incorporating all required labels. This section will detail the specifics of the trajectory sampling procedure. In general, we want the sampled video to provide rich information (good coverage) about the scene, which can be broken down into two aspects. Firstly, we aim for the camera to physically cover the room. That means the sampled camera positions should enable visiting most open spaces in the scene, rather than just focusing on the largest open space. This guides our design for camera position sampling. Second, we want the actual video to capture as many objects in the scene as possible while still being realistic (i.e., facing the direction of movement while moving). This guides our design for camera orientation sampling. Next, we will establish the detailed steps. We will open-source all codes and generate a video dataset.

- To sample camera positions within the trajectory, a basic approach might involve randomly selecting traversable points within the scene. However, this brings the issue that points in larger rooms are more likely to be selected compared to those in smaller rooms. Our objective is to achieve a more uniform coverage across the entire scene, avoiding overconcentration in larger open areas. Thus, we used the **farthest point sampling** method to sample a set of key points that sparsely span the scene. Our focus is on ensuring the trajectory covers the main open spaces, without the necessity of navigating narrow spaces such as



(a) **Articulation.** The model may have limited exposure to open states for articulation objects, which makes it less likely to recognize a microwave with its door open correctly.



(b) **Lighting.** When the environment is dark, the model performance is negatively affected. However, when the lighting exceeds a certain threshold, in this case 0.5, the model becomes robust to increasing illumination.

(c) **Visibility.** The model's detection performance suffers when most of the target object is occluded. Surprisingly, a correct prediction can be made with only half of the object visible.

(d) **Zoom.** When the model is zoomed out, nearby objects become distorted, leading the model to identify irrelevant objects as the target mistakenly. This suggests that the model's recognition may rely partly on contours rather than solely on semantic information.

(e) **Pitch.** We find that, generally, the model can achieve better performance in a look-down angle compared to a look-up angle.

Figure 8. Error analysis for Grounding DINO [44]. Similar trends are also observed in other detection models. Each row in our presentation represents one axis and comprises four example groups. In each group, there are two images: the left image illustrates the ground truth, highlighting the target object in magenta, while the right image shows the Grounding DINO's predictions (colored differently) for the target object, as indicated at the top of the first image in each group. The example groups are arranged such that, from left to right, the intensity along the respective axis increases (e.g., progressing from zoomed in to zoomed out), and the intensity value (0-1) is shown on top of each prediction.

the gap between a cabinet and a wall. To achieve this, we perform the sampling on an **eroded** version of the traversal map. This technique effectively highlights the larger open areas in a scene while eliminating smaller gaps and corners.

- Now we have a set of candidate key points sampled in the scene, but a view from many of them may provide similar information about the scene (for example, two nearby points in the same room). We don't need to visit both of them in the same trajectory. To ensure efficiency and avoid redundancy, we need to select a subset of these key points while still preserving a comprehensive view of the scene (such as not excluding all points from a specific room). Our selection process begins by assessing the unique information each key point provides, specifically the objects visible from that point. We place a virtual camera at each key point and rotate it 360 degrees, recording the angle at which the maximum number of objects in

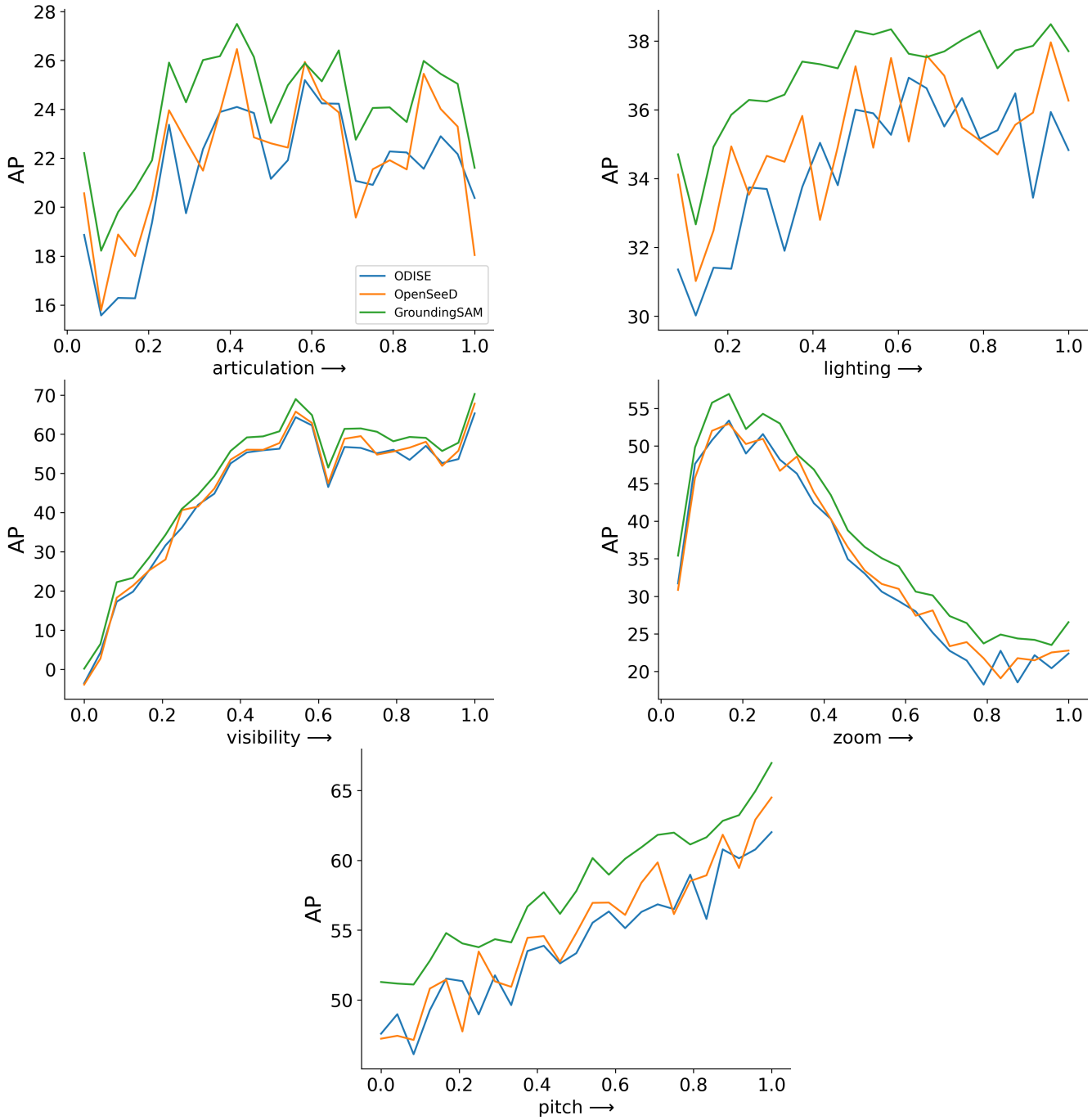


Figure 9. Similar to Figure 3 in the main (Section 4.1), we plot the Segmentation model results for each of the five axes. AP is calculated using target object only.

the scene are visible. We note both the visible objects and their corresponding angles for each key point. Next, we randomize the order of the key points and go over each point sequentially to select the points that offer additional information. If a key point reveals an object not visible from **all** previously selected points, we retain it. Otherwise, we discard it. This method results in a smaller, more efficient subset of key points — referred to as waypoints in the subsequent step — which still allows a comprehensive observation of most objects in the scene.

- Once we have determined the set of waypoints, our next step is to devise a trajectory that connects them in the

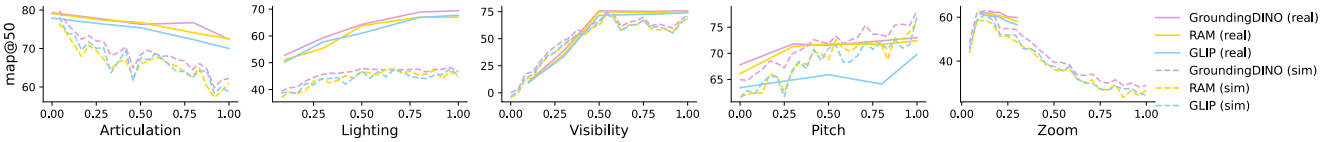


Figure 10. We also observe a comparable pattern in the parametric evaluation conducted on real data as observed in synthetic data (main Figure 3).

shortest possible path. To achieve this, we frame the task as a traveling salesman problem, treating the waypoints as nodes to be visited on the scene traversal graph. In this step, to ensure the camera maintains a safe distance from walls and furniture, we slightly erode the scene traversal graph. This adjustment prevents the camera from getting too close to these obstacles, ensuring smoother navigation through the scene.

- After establishing the sequence of positions in the previous step, we focus on determining the camera's orientation at each position. Our goal is to mimic the behavior of a real agent exploring the scene. Therefore, while in motion, the camera faces the direction it is moving towards. Additionally, at each waypoint, the camera 'makes a stop' and turns to an optimal angle. This angle, predetermined in an earlier step, allows the camera to capture the most objects from that viewpoint. This approach serves two purposes: firstly, to ensure that the trajectory includes keyframes or angles for optimal object visibility, and secondly, to simulate the natural behavior of an agent pausing to observe the surroundings.

C.3. Object States and Relations Prediction (main paper Sec. 4.3)

Details of Generation Process. For binary object relationship prediction, we synthesized 12.5k images, each annotated with one or more of the following five labels: `open`, `close`, `ontop`, `inside`, `under`. For instance, an image might depict a toy inside an open cabinet, thus making both "inside" and "open" labels applicable.

The image sampling process is as follows: Firstly, we select a scene and a piece of furniture to serve as the primary object in the relation (e.g., a table for placing items on top). Subsequently, we determine a plausible relationship related to this base object, with annotations provided via BVS. For example, an item might be positioned `ontop` or `under` a table, but not `inside` it. Following this, we select a random object to place in the scene. This object is then integrated into the scene, employing the physical state sampling function from BVS to ensure its placement aligns with the predetermined relationship. For instance, we might sample a cupcake and place it at a random location on top of the table. Lastly, we sample a random camera pose, ensuring the placed object is centered in the frame. We then filter out any instances where the objects of interest are not adequately visible. This procedure is repeated iteratively to compile our final dataset.

For unary object state prediction, we generated 500 images that either consist of a `filled` or empty (not filled) container, similarly 500 images for `folded`. The sampling process for unary states is simpler – we randomly sample a scene, then place a random container/cloth object in the scene, by 50% chance sample a filled or folded state for the target object, then also sample a random camera pose with the target object in the center.

Details of Architecture Design and Hyperparameters. Our model architecture is adapted from [30, 48]. Given an image input with two (or one) bounding boxes, the model predicts the binary spatial (or unary) relationship between objects corresponding to boxes (Fig. 11). First, the model utilizes a Segment Anything image encoder [33] to extract hidden features. Subsequently, RoIAlign [21] is applied to the extracted features using the two (or one) bounding boxes. In the binary case, where the objective is to predict the spatial relationship between the two objects, RoIAlign effectively captures spatial information from the representation

Additional features are incorporated into the aligned representations to enhance semantic information. Unlike [48], which relies on word2vec vectors [49] and category names for the objects (which may not always be readily available in the world), we opt to use the Segment Anything extracted feature, from the cropped image encompassing the union of the two bounding boxes, as the additional feature. This approach preserves both spatial and semantic information.

The concatenated features are then fed into a trainable CNN to predict seven-way logits. To prevent overfitting, we freeze the Segment Anything image encoder, ensuring that the only learnable parameters are those of the randomly initialized CNN. Under a 0.3 learning rate with linear scheduling, the model is trained on 13.5k synthetic images only but can achieve strong performance in the real test set (Table 4 in the main paper).

Lastly, we discuss the details of **zero-shot CLIP** [52] baseline, which is used to mimic scenarios where synthetic datasets are not accessible. In this scenario, we have to rely on CLIP's zero-shot capacity. Specifically, akin to our ar-

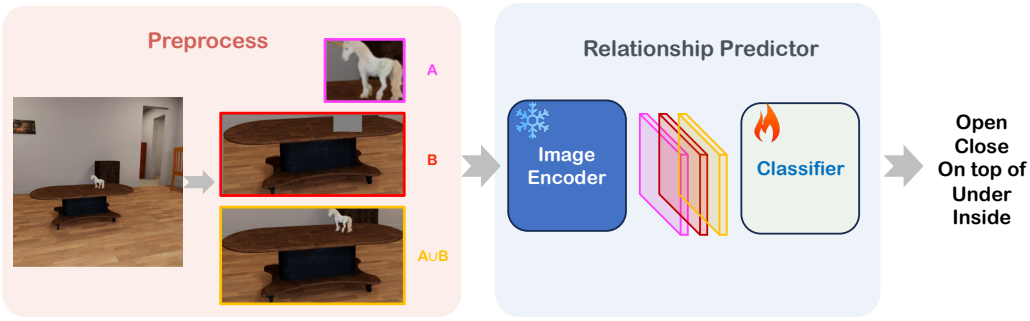


Figure 11. Relationship prediction model architecture used in Sec 3.3.

chitecture, the image is cropped to maximize the semantic information. Then the image embedding of the cropped images is compared with the label text embeddings from all verbalized prompts. A verbalized prompt can take the form of `<A> on top of <B>` where `<A>` and `<B>` are the placeholder for the actual object category name. Empirically we find including the category name in the prompt outperforms using predicate only (e.g., only `on top of`). We emphasize that having prior knowledge of category names in advance renders this task more straightforward and less fair to our approach where we have no assumption on access to any category name.

Shown in main Tables 4 and 5, BVS is capable of generating high-quality synthetic training data by demand [15, 18]. Models trained on such synthetic training data are able to capture the essence of predicate prediction and bridge the sim-to-real gap.

Folded and Filled Prediction BVS also supports nuanced unary object predicate such as `folded` and `filled`. Models training on synthesized photo-realistic images can transfer well to real images. We train two linear probes on top of the EVA02 [12] encoded representation to predict `folded` and `filled`, respectively. We manually collect 50 real test images for each of the two predicates and observe that linear probes can achieve 86% and 93% real test accuracy for `folded` and `filled`, respectively.

D. Demo Videos

We have provided `video-demo.zip`, which consists of demo videos of our generated videos and model prediction. Specifically, `BVS-highlight.mp4` shows all visualization content. `customizable-dataset-generator.mp4` shows scene instance augmentation. `predicate_prediction.mp4` consists of predicate prediction model prediction on our collected real videos (main Sec. 4.3). `compare-detection-{axis}.mp4` consists of three detection model predictions on five parametric evaluation axes (main Sec. 4.1).