YCB@Home: A Large Scale Synthetic Object Detection Dataset for RoboCup @Home League

Julio Ojalvo, Christopher Duarte, Shengxin Luo, Katarzyna Pasternak, and Ubbo Visser

Department of Computer Science, University of Miami, 1365 Memorial Dr, Coral Gables, FL 33146, USA
{jojalvo|cduarte|tluo|kwp|visser}@cs.miami.edu

Abstract. The RoboCup@Home league requires domestic service robots to perform complex tasks in dynamic, human-centric environments, where robust object detection remains a critical challenge. While existing datasets focus on general object recognition, they often lack the specificity and contextual relevance needed to evaluate performance under real-world RoboCup@Home conditions. To address this gap, we present YCB-@Home, a carefully curated validation dataset designed to benchmark object detection models in household settings. The dataset consists of 20,000 RGB-D images featuring 78 common household items, annotated with bounding boxes and captured under varying lighting, occlusion, and viewpoint conditions. YCB@Home, unlike generic object detection datasets, emphasizes task-specific scenarios reflective of RoboCup@Home challenges. We evaluate the dataset using the most commonly deployed state-of-the-art object detection model (YOLO11), analyzing performance gaps in occluded and context-dependent object recognition. Our results demonstrate that YCB@Home provides a more representative benchmark for RoboCup@Home than existing datasets, revealing key limitations in current systems and guiding future improvements for reliable, context-aware robotics. The dataset can be found at the following link: https://github.com/Julio-Ojalvo/YCB-Home.git

Keywords: Dataset · Simulation · Isaac Sim · Computer Vision · Object Detection · Service Robots · Domestic Robotics · Human Support Robot

1 Introduction

Simulators play a pivotal role in advancing research, development, and testing of robotic systems [1, 16, 23, 15] in the realm of service robotics. They serve as invaluable tools for exploring various aspects of robot perception, navigation, and interaction with the environment and users [18, 10, 3]. Among the key challenges in robotics is enabling robust perception, where vision-based object detection is critical for tasks such as manipulation, navigation, and human-robot interaction.

A robotics simulator must provide realistic sensory inputs to effectively support the development of perception systems [18, 21]. Vision sensors, in particular,

are essential for enabling robots to interpret their surroundings accurately. However, collecting and annotating real-world training data for object detection is often time-consuming, expensive, and limited in variability [5, 9]. Synthetic data generation offers a promising alternative, allowing for scalable, diverse, and precisely labeled datasets that can be used to train and evaluate object detection models [2, 24]. As a small RoboCup team competing in RoboCup@Home, where objects to be used in tasks are only revealed days prior to the competition, this is a vital aspect of our data generation method.

NVIDIA Isaac Sim is a powerful simulation platform that leverages high-fidelity, ray-traced rendering to generate photorealistic scenes [11]. Its capabilities in simulating realistic lighting, textures, and sensor noise make it well-suited for generating synthetic training data that can help bridge the sim-to-real gap in vision-based perception [1, 14]. By programmatically controlling scene composition, object placement, and environmental conditions, Isaac Sim enables the creation of large-scale, annotated datasets tailored to specific robotic applications [4, 25].

Through benchmarking and analysis using synthetic data generated in Isaac Sim, we demonstrate how simulation can enhance object detection robustness for real-world home environments. By releasing YCB@Home as an open-source validation benchmark, we aim to support RoboCup@Home teams in evaluating and refining their perception systems. We hope this contribution fosters collaboration within the robotics community, accelerating progress toward reliable and adaptable domestic service robots.

2 Related Work

The use of synthetic data for training and benchmarking robotic vision systems has gained significant traction due to the scalability and diversity it offers compared to real-world data collection. NVIDIA's Isaac Sim, a high-fidelity robotics simulation platform built on Omniverse, has emerged as a powerful tool for generating synthetic datasets. Isaac Sim enables physically accurate simulations with domain randomization, which improves the sim-to-real transferability of learned models [20]. Isaac Sim employs NVIDIA RTX graphics cards to achieve physically accurate ray-traced illumination. Given our existing inventory of NVIDIA RTX hardware, this platform was selected over alternative simulation environments to maximize compatibility and performance.

The YCB Object Set [5] has become a standard benchmark in robotic manipulation and perception research due to its carefully selected household objects with varied shapes, textures, and physical properties. Prior works have leveraged the YCB objects for tasks such as object detection [8], 6D pose estimation [19], and grasp planning [13]. However, collecting large-scale real-world datasets with all 78 YCB objects is labor-intensive, motivating the use of synthetic data generation.

Several studies have explored synthetic dataset generation for robotics using the YCB objects. For instance, Denninger et al. [6] introduced BlenderProc, a

Blender-based pipeline for generating photorealistic synthetic data. Similarly, Tremblay et al. [20] demonstrated that training on synthetic data with domain randomization can achieve performance comparable to real-world data. More recently, Isaac Sim has been used to generate large-scale datasets for robotic grasping [7], benefiting from its GPU-accelerated rendering and physics simulation.

While synthetic datasets have been used to train object recognition models for RoboCup@Home [12], existing efforts typically focus on limited subsets of objects or generic household items. Vaz et al. [22] provide a concise dataset for RoboCup@Home, but their process differs greatly from ours. Their solution relies heavily on their ODUTF method to generate images of new objects. The primary focus of our method is to have a quick and portable way to generate data on a previously unseen object, given the structure of the RoboCup@Home competition. The use of a large hardware device to capture object images is a restraint we cannot afford to have for our competitions. Our work introduces a comprehensive synthetic dataset covering all 78 YCB objects, specifically designed to serve as both a training resource and - more critically - a rigorous validation benchmark for RoboCup@Home tasks. This dataset enables teams to quantitatively evaluate perception systems against the full spectrum of YCB objects encountered in competition scenarios.

3 Methodology

Fig. 1. Figure (a) illustrates the office environment in the dataset, (b) is the warehouse environment, (c) is the home environment, and (d) is the RoboCanes lab digital twin environment.

3.1 Scene Configuration

All 78 objects in the YCB dataset are represented across 20,000 images rendered in Isaac Sim. Four different environments, as seen in Figure 1, are used to generate the data to include variability surrounding the YCB objects. These environments were chosen out of their similarity either visually or semantically to the RoboCup@Home setting. Each environment has its own background colors and textures, lighting conditions, and varying degrees and types of clutter. We generate 5,000 images inside each of the environments for an even distribution across all so as not to overfit to any particular one. Additionally, the process to generate data within each environment is identical.

Anywhere from 5 to 15 objects of the 78 are randomly selected for each scene, and 22 different camera poses are used relative to the objects. The 5 to 15 objects range for each scene configuration was chosen to closely follow a configuration likely to be seen in RoboCup@Home competition. The 22 camera poses were chose to both simulate realistic visuals from competition and to show features on objects from a variety of distances and angles. The 22 camera poses include: four at just above object level facing the objects from cardinal directions to have up close images capturing features in detail, four elevated views replicating this arrangement but from a farther distance to include instances of objects with more abstract feature presentations, and 14 poses simulating a robotic approach trajectory. This configuration ensures comprehensive coverage from fixed perspectives while capturing dynamic viewpoint transitions. Shahinfar et. al [17] suggests to produce 150 to 500 images per object for an effective object detection dataset. With 78 objects, 250 images per object, and an average of 10 objects per image, that comes out to 3,900 total images. To account for all the variability, we produce slightly more than this recommended amount for each environment. Using this configuration, 20,000 unique images are created to represent a robust set of annotated images within the context of the RoboCup@Home competition.

3.2 Domain Randomization

The four major parameters randomized in this dataset are 1) object poses, 2) camera poses, 3) lighting conditions, and 4) the environment in which objects are captured. Our goal is to provide enough specificity to effectively be used for RoboCup@Home tasks but be robust enough within that context to account for any changes in lighting, object placement, occlusion, and competition venues. We structure our domain randomization parameters to achieve this goal.

The object poses are randomly positioned in a circular arrangement atop a chosen plane. Those planes are a) a coffee table in the RoboCanes lab environment, b) a kitchen counter in the home environment, c) a desk in the office environment, and d) a shelf in the warehouse environment. The objects are chosen at random and randomly rotated within this circular arrangement to create varying levels of clutter and occlusion in every scene. The camera poses were chosen to represent different angles of view and distances to the objects. Of the 22 total camera poses, 8 circle around the plane of objects at different heights,

and 14 are along the path of an HSR (Human Support Robot) approaching the table as it would in a RoboCup@Home task. Those camera poses are cycled through, with a new set of objects for each pose, until all 5,000 images are taken per environment.

The environments themselves provide a different background for the objects but also their own lighting conditions. The office and RoboCanes lab's main light sources are differently shaped rectangular light arrays on the ceilings of different heights. The home's light source is a disk-shaped light above the object plane, and the warehouse uses sunlight coming through a window at an angle. These lighting conditions provide different specularity and shadows to every scene to allow for more robustness to lighting conditions in the training dataset. The ray-tracing capabilities of Isaac Sim provides a realistic set of lighting conditions for our data. Examples of images produced with these conditions can be seen in Figure 2.

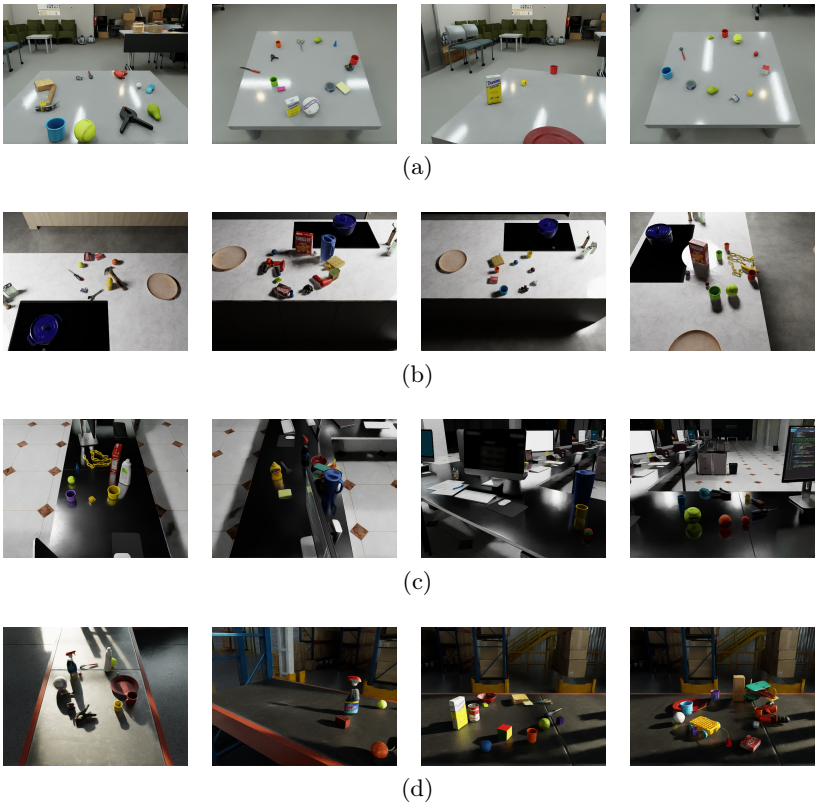

Fig. 2. Row (a) shows the images taken from 4 different camera poses of the 22 in the RoboCanes lab environment. Row (b) is the home, Row (c) is the office, and Row (d) is the warehouse.

3.3 Annotation Automation

Isaac Sim's replicator module is used to generate all of the annotations for the data. Toyota provides an implementation of the HSR in Isaac Sim, and the camera (ASUS Xtion RGBD) located on robot's head is used in our work. This implementation was also used to capture some of the camera poses present in our data. We do this to further simulate the RoboCup@Home setting. An Isaac Sim object detection writer from the replicator module is attached to the render product of this camera. The writer then produces four files for every frame of the camera. These are the png image file of the render product in that moment, an npy matrix file with the annotations of the objects represented as USDs, and two json files for the prim paths of the USDs and their label names. The data must then be converted into a format suitable for our training process.

The annotations are originally in the form of an npy file containing a matrix in which each row is an object in the scene. The first column contains an integer value representing the label, while the subsequent four columns specify the pixel coordinates of the bounding box, corresponding to the minimum x-coordinate, minimum y-coordinate, maximum x-coordinate, and maximum y-coordinate, respectively. To train using ultralytic's YOLO11 object detection model, the annotations are transformed to a txt file with the same name as its corresponding png, and the bounding box is defined by the center point x and y coordinates, followed by the width and height of the box. These numbers are also normalized form 0 to 1 rather than the raw pixel values. The prim path files are ignored, and the label name files are used to generate the yaml file defining file paths to the data as well as a dictionary mapping integer labels present in the txt annotations to string labels of those objects. This allows for more human readable annotations to ensure they are correct as well as to visualize the final output as can be seen in Figure 3.

4 Results and Analysis

To assess the effectiveness of YCB@Home as a task-specific benchmark, we trained a YOLO11 model exclusively on our synthetic dataset and evaluated its performance on two distinct test sets: a hand-annotated validation dataset replicating RoboCup@Home competition scenarios, and the YCB-Video (YCBV) dataset, a general-purpose benchmark for object detection in unstructured environments. The training started with the large YOLO11 weights pretrained on COCO and ran for 200 epochs with default values for all other training parameters.

The model achieved a mean Average Precision (mAP@0.5) of 75.3% on the hand annotated test set, with precision and recall scores of 78.7% and 73.7%, respectively. In contrast, performance dropped noticeably on YCBV, with significantly lower accuracy and detection rates. This substantial performance gap suggests a meaningful distributional mismatch between generic benchmarks and task-specific RoboCup@Home environments.

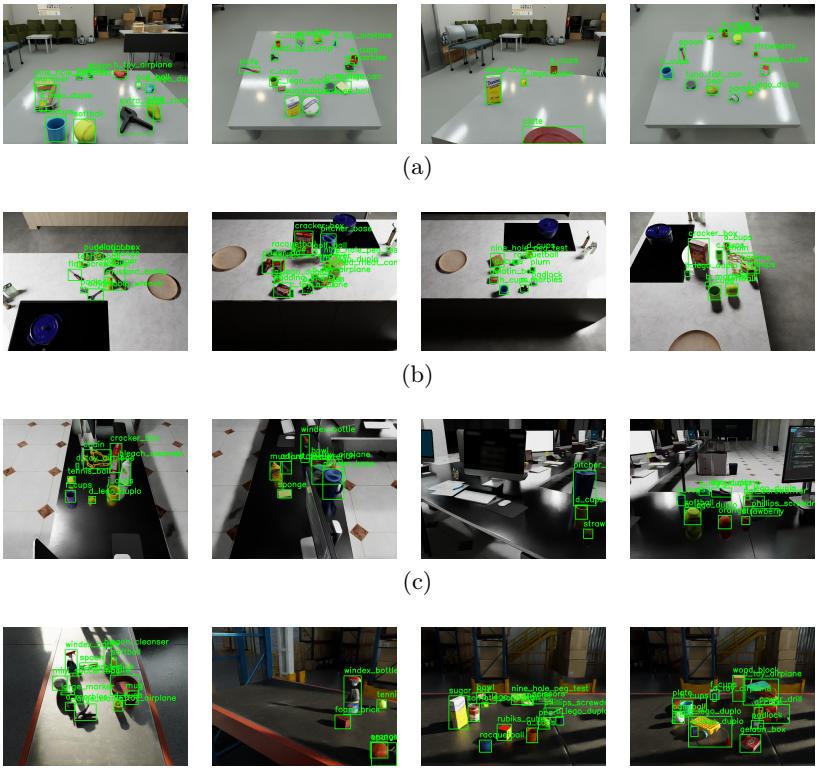

Fig. 3. Rows (a), (b), (c), and (d) are the RoboCanes lab, home, office, and warehouse environments respectively, as shown in Figure 2 but with the automated annotations drawn on.

Qualitatively, the model excelled in cluttered, contextually grounded scenes, accurately localizing occluded objects critical to domestic tasks. An example of this can be seen in Figure 4, where we can see the predictions on heavily cluttered scenes. We can also see examples of occluded objects in the yellow bowl labeled "j_cups" in the bottom left and top two images, as well as the banana in the bottom right image. The yellow bowl is mislabeled in one of the images as a softball, however, the bonding box around the bowl is accurate in all instances. This success reflects YCB@Home's design, which emphasizes domain randomization tailored to competition workflows.

The performance disparity arises from fundamental differences in dataset construction. YCB@Home's domain randomization (e.g., venue lighting, camera poses simulating HSR navigation) bridges the sim-to-real gap, whereas YCBV's variability is much narrower and less structured. These results affirm that task-specific synthetic datasets are indispensable for evaluating perception systems in domain-constrained robotics applications.

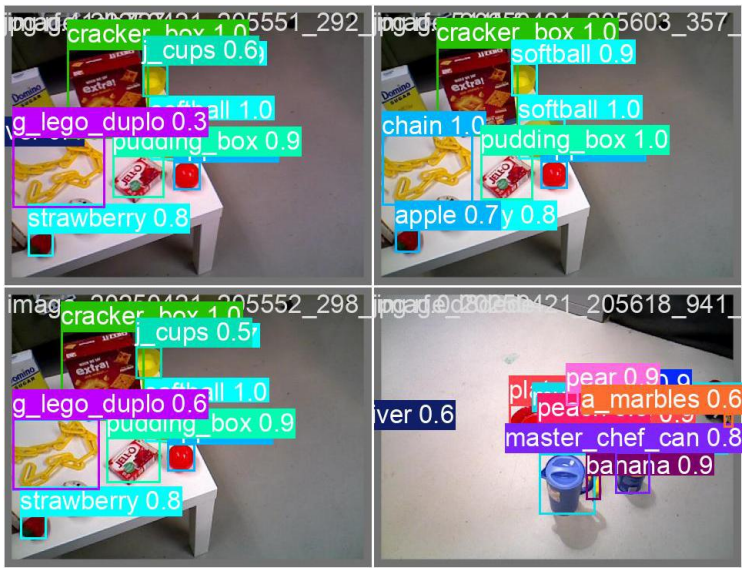

Fig. 4. Validation batch prediction demonstrating performance on cluttered scenes.

5 Conclusion

This work introduces YCB@Home, a synthetic dataset designed to benchmark or train object detection models in RoboCup@Home scenarios. Our experiments demonstrate that models trained on YCB@Home excel in task-specific validation but struggle with generic benchmarks like YCBV, exposing a critical distributional gap between domain-focused and general-purpose datasets. Our key contributions are:

- YCB@Home provides a rigorous validation tool for diagnosing model weaknesses and performance in context-aware object detection, directly addressing RoboCup@Home's unique challenges; and
- The dataset's design, prioritizing task relevant scene configurations, environmental randomization, and realistic partial occlusion, narrows the sim-to-real gap for domestic service robotics.

In our future work, we plan to explore hybrid training strategies combining YCB@Home with real-world data to enhance generalization without sacrificing task specificity. We also hope to extend the dataset to include more variability in its lighting conditions, background colors and textures, camera poses, camera noise and blur, and distractor objects. In Figure 4, we can see examples of objects that tend to be misclassified, such as the chain and the strawberry. In the future, we plan to address this by tailoring the distribution of object instances in the dataset to skew more heavily towards objects the models struggle to detect, as opposed to an even distribution across all classes. By addressing these directions,

we aim to advance the development of robust, context-aware vision systems for domestic robotics, ultimately improving their reliability in competition and real-world settings.

References

1. Acosta, B., Yang, W., Posa, M.: Validating Robotics Simulators on Real World Impacts. IEEE Robotics and Automation Letters **7**(3), 6471–6478 (2022)
2. Anderson, P., Shrivastava, A., Truong, J., Majumdar, A., Parikh, D., Batra, D., Lee, S.: Sim-to-Real Transfer for Vision-and-Language Navigation. In: Conference on Robot Learning. pp. 671–681. PMLR (2021)
3. Belo, J.P.R., Romero, R.A.F.: A Social Human-Robot Interaction Simulator for Reinforcement Learning Systems. In: 2021 20th International Conference on Advanced Robotics (ICAR). pp. 350–355 (2021)
4. Blanco-Mulero, D., Barbany, O., Alcan, G., Colomé, A., Torras, C., Kyrki, V.: Benchmarking the Sim-to-Real Gap in Cloth Manipulation. IEEE Robotics and Automation Letters **9**(3), 2981–2988 (2024)
5. Calli, B., Singh, A., Bruce, J., Walsman, A., Konolige, K., Srinivasa, S., Abbeel, P., Dollar, A.M.: Yale-CMU-Berkeley Dataset for Robotic Manipulation Research. The International Journal of Robotics Research **36**(3), 261–268 (2017)
6. Denninger, M., Sundermeyer, M., Winkelbauer, D., Zidan, Y., Olefir, D., Elbadrawy, M., Lodhi, A., Katam, H.: Blenderproc. arXiv preprint arXiv:1911.01911 (2019)
7. Eppner, C., Mousavian, A., Fox, D.: Acronym: A large-scale grasp dataset based on simulation. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 6222–6227. IEEE (2021)
8. Grenzdörffer, T., Günther, M., Hertzberg, J.: Ycb-m: A multi-camera rgb-d dataset for object recognition and 6dof pose estimation. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 3650–3656. IEEE (2020)
9. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
10. Liu, Q., Li, Y., Liu, L.: A 3D Simulation Environment and Navigation Approach for Robot Navigation via Deep Reinforcement Learning in Dense Pedestrian Environment. In: 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE). pp. 1514–1519 (2020)
11. Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., State, G.: Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning (2021)
12. Massouh, N., Brigato, L., Iocchi, L.: Robocup@ home-objects: benchmarking object recognition for home robots. In: RoboCup 2019: Robot World Cup XXIII 23. pp. 397–407. Springer (2019)
13. Morrison, D., Corke, P., Leitner, J.: Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. arxiv 2018. arXiv preprint arXiv:1804.05172
14. Narang, Y., Sundaralingam, B., Macklin, M., Mousavian, A., Fox, D.: Sim-to-Real for Robotic Tactile Sensing via Physics-Based Simulation and Learned Latent Projections. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 6444–6451 (2021)

15. Qin, L., Peng, H., Huang, X., Liu, M., Huang, W.: Modeling and Simulation of Dynamics in Soft Robotics: a Review of Numerical Approaches. Current Robotics Reports **5**(1), 1–13 (Mar 2024)
16. Reckhaus, M., Hochgeschwender, N., Paulus, J., Shakhimardanov, A., Kraetzschmar, G.K.: An Overview about Simulation and Emulation in Robotics. Proceedings of SIMPAR pp. 365–374 (2010)
17. Shahinfar, S., Meek, P., Falzon, G.: "how many images do i need?" understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring. Ecological Informatics **57**, 101085 (2020)
18. Skinner, J., Garg, S., Sünderhauf, N., Corke, P., Upcroft, B., Milford, M.: High-Fidelity Simulation for Evaluating Robotic Vision Performance. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2737–2744 (2016)
19. Tekin, B., Sinha, S.N., Fua, P.: Real-time seamless single shot 6d object pose prediction. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 292–301 (2018)
20. Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S., Birchfield, S.: Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 969–977 (2018)
21. Trentsios, P., Wolf, M., Gerhard, D.: Overcoming the Sim-to-Real Gap in Autonomous Robots. Procedia CIRP **109**, 287–292 (2022), 32nd CIRP Design Conference (CIRP Design 2022) - Design in a changing world
22. Vaz, M.d.O., Rohrich, R.F., Fabro, J.A., De Oliveira, A.S.: A concise dataset for intelligent behaviors in domestic tasks. IEEE Access (2025)
23. Zagal, J.C., Ruiz-del Solar, J.: Combining Simulation and Reality in Evolutionary Robotics. Journal of Intelligent and Robotic Systems **50**(1), 19–39 (Sep 2007)
24. Zhao, W., Queralta, J.P., Westerlund, T.: Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey. In: 2020 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 737–744 (2020)
25. Zhu, Y., Wong, J., Mandlekar, A., Martín-Martín, R., Joshi, A., Nasiriany, S., Zhu, Y.: robosuite: A Modular Simulation Framework and Benchmark for Robot Learning (2022)