GenAug: Retargeting behaviors to unseen situations via **Gen**erative **Aug**mentation

Zoey Chen[1], Sho Kiami[1], Abhishek Gupta*[1], Vikash Kumar*[2]
[1]University of Washington [2] Meta AI
{qiuyuc, shokiami, abhgupta}@cs.washington.edu vikashplus@meta.com

genaug.github.io

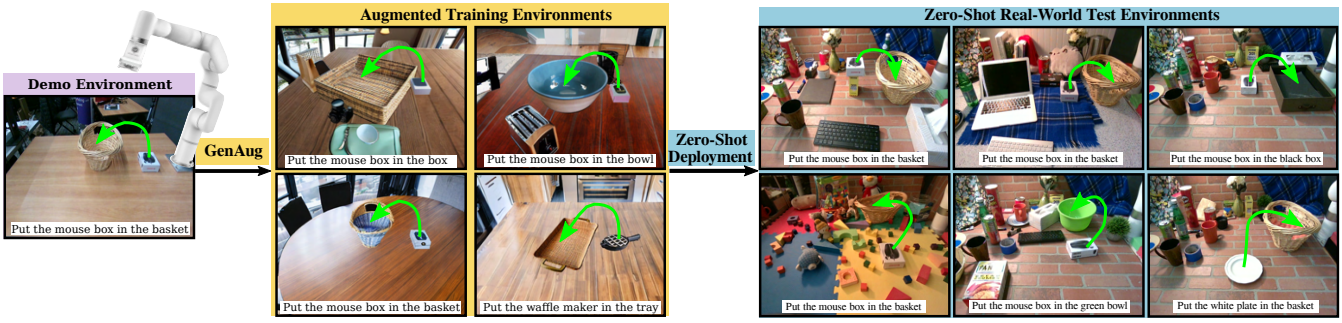

Fig. 1: An illustration of the problem setting for our proposed system *GenAug*. *GenAug* takes a small set of image-action demonstration data on a robotics problem like tabletop pick-and-place and generates a diverse set of augmented image observations to supplement the real-world demonstration dataset. These augmented observations add semantically meaningful visual diversity in objects, distractors, and backgrounds while maintaining functional invariance of the actions. Training on this augmented training set leads to significant improvements in policy generalization, without requiring additional data collection.

***Abstract*—Robot learning methods have the potential for widespread generalization across tasks, environments, and objects. However, these methods require large diverse datasets that are expensive to collect in real-world robotics settings. For robot learning to generalize, we must be able to leverage sources of data or priors beyond the robot's own experience. In this work, we posit that image-text generative models, which are pre-trained on large corpora of web-scraped data, can serve as such a data source. We show that despite these generative models being trained on largely non-robotics data, they can serve as effective ways to impart priors into the process of robot learning in a way that enables widespread generalization. In particular, we show how pre-trained generative models can serve as effective tools for semantically meaningful data augmentation. By leveraging these pre-trained models for generating appropriate "semantic" data augmentations, we propose a system *GenAug* that is able to significantly improve policy generalization. We apply *GenAug* to tabletop manipulation tasks, showing the ability to re-target behavior to novel scenarios, while only requiring marginal amounts of real-world data. We demonstrate the efficacy of this system on a number of object manipulation problems in the real world, showing a 40% improvement in generalization to novel scenes and objects.**

I. Introduction

While robot learning has often focused on the search for optimal behaviors [1, 2] or plans [3], the power of learning methods in robotics comes from the potential for *generalization*. While techniques such as motion planning or trajectory optimization are effective ways to solve the policy search problem in highly controlled situations such as warehouses or factories, they may fail to generalize to novel scenarios without significant environment modeling and replanning [4]. On the other hand, techniques such as imitation learning and reinforcement learning have the *potential* for widespread generalization without significant environment modeling and replanning, especially when combined with deep neural network function approximators [5, 6, 7].

Let us consider this question of learning from human demonstrations [8]. While imitation learning methods circumvent the challenges of exploration, these methods often impose a heavy burden on data collection by human supervisors. Human demonstrations are often collected by expensive techniques such as on-robot teleoperation or kinesthetic teaching, which limit the amount of real-world data that can be collected. Beyond the sheer quantity of data, the rigidity of most robotics setups makes it non-trivial to collect *diverse* data in a wide variety of scenarios. As a result, many robotics datasets involve a single setup with just a few hours of robot data. This is in stark contrast to the datasets that are common in vision and language problems [9, 10], both in terms of quantity *and* the diversity of the data. Given how important large-scale data has been for generalization in these domains, robot learning is likely to benefit from access to a similar scale of data.

Data augmentation can provide a way to improve model generalization, but these techniques typically perform augmentation in low-level visual space, performing operations such as color jitter, Gaussian blurring, and cropping, among others. While this may help with generalization to low-level changes

in scene appearance, they are unable to handle large semantic differences in the scene such as distractors, background changes, or object appearance changes. In this work, we aim to provide *semantic* data augmentation to enable broad robot generalization, by leveraging pre-trained generative models. While on-robot data can be limited, the data that pre-trained generative models are exposed to is significantly larger and more diverse [9, 10]. Our work aims to leverage these generative models as a source of data augmentation for real-world robot learning. The key idea of our work builds on a simple intuition: demonstrations for solving one task in one environment should still largely be applicable to the same task in new environments despite the visual changes in scenes, background, and object appearances. The small amount of on-robot experience provides demonstrations of the desired behavior, while a generative model can be used to generate widely varying visual scenes, with diverse backgrounds and object appearances under which the same behavior will still be valid. Furthermore, since such generative models are trained on realistic data, the generated scenes are visually realistic and extremely diverse. This allows us to cheaply generate a large quantity of *semantically augmented* data from a small number of demonstrations, providing a learning agent access to significantly more diverse scenes than the purely on-robot demonstration data. As we show empirically, this can lead to widely improved generalization, with minimal additional burden on human data collection.

We present GenAug, a semantic data augmentation framework that leverages pre-trained text-to-image generative models for real-world robot learning via imitation. Given a dataset of image-action examples provided on a real robot system, GenAug automatically generates "augmented" RGBD images for entirely different and realistic environments, which display the visual realism and complexity of scenes that a robot might encounter in the real world. In particular, GenAug leverages language prompts with a generative model to change object textures and shapes, adding new distractors and background scenes in a way that is physically consistent with the original scene, for table-top manipulation tasks with a real robot. We show that training on this *semantically augmented* dataset significantly improves the generalization capabilities of imitation learning methods on entirely unseen real-world environments, with only 10 real-world demonstrations collected in a single, simple environment.

In summary, our key contributions are:

1) We present a general framework for leveraging generative models for data augmentation in robot learning.
2) We show how this framework can be instantiated in the context of tabletop manipulation tasks in the real world, building on the framework of CLIPort introduced in [11].
3) We demonstrate that GenAug policies can show widespread real-world generalization for tabletop manipulation, even when they are only provided with a few demonstrations in a simple training environment.
4) We provide a number of ablations and visualizations to understand the impact of various design decisions on learned behavior.

II. GenAug: Generative Augmentation for Real-World Data Collection

In this section, we will describe the problem statement we consider in our semantic data augmentation technique - Generative Augmentation (*GenAug*), show how generative models can conceptually be used to inject semantic invariances into robot learning and instantiate a concrete version of this setup for learning policies for tabletop robotic manipulation tasks.

A. Problem Formulation

In this work, we consider robotic decision-making problems, specifically in robotic manipulation. For the sake of exposition, let us consider prediction problems where an agent is provided access to sensory observations $o \in \mathcal{O}$ (e.g. camera observations) and must predict the most appropriate action $a \in \mathcal{A}$ (e.g. where to move the robot arm for picking up an object). The goal is to learn a predictive model $f_\theta : \mathcal{O} \rightarrow \Delta A$ (where ΔA denotes the simplex over actions) that predicts a distribution over actions such that the action $a \sim f_\theta(.|o)$ is able to successfully accomplish a task when executed in the environment. In this work, we will restrict our consideration to supervised learning methods for learning $f_\theta(.|o)$. We will assume that a human expert provides a dataset of optimal data $\mathcal{D} = \{(o_0, a_0), (o_1, a_1), \ldots, (o_N, a_N)\}$, and learn policies with standard maximum likelihood techniques [12, 11]:

$$\max_\theta \mathbb{E}_{(o,a)\sim\mathcal{D}} \left[\log f_\theta(a|o)\right] \tag{1}$$

This training process is limited to the training dataset $\mathcal{D}$ that is actually collected by the human supervisor. Since this might be quite limited, data augmentation methods apply augmentation functions $q : \mathcal{O} \times \mathcal{A} \times \mathcal{Z} \rightarrow \mathcal{O} \times \mathcal{A}$ which generate augmented data $(o', a') = q(o, a, z); z \sim p(z)$, where different noise vectors z generate different augmentations. This could include augmentations like Gaussian noise, cropping, and color jitter amongst others [13, 14, 15, 16]. Given an augmentation function, an augmented dataset can be generated $\mathcal{D}_{\text{aug}} = \mathcal{D} \cup \{(o', a')_i\}_{i=1}^{M}$, where $M \gg N$, and then used for maximum likelihood training of $f_\theta(a|o)$. Typically these augmentation functions q are not learned but instead hand-specified by an algorithm designer, with no real semantic meaning. Instead, they impose invariances to the corresponding disturbances such as color variations, rotations and so on to help combat overfitting. Next, we describe how generative models can be leveraged for semantic data augmentation.

B. Leveraging Generative Models for Data Augmentation

While data augmentation methods typically hand-define augmentation functions $(o', a') = q(o, a, z); z \sim p(z)$, the generated data (o', a') may not be particularly relevant to the distribution of real-world data that might be encountered during evaluation. In this case, it is not clear if generating

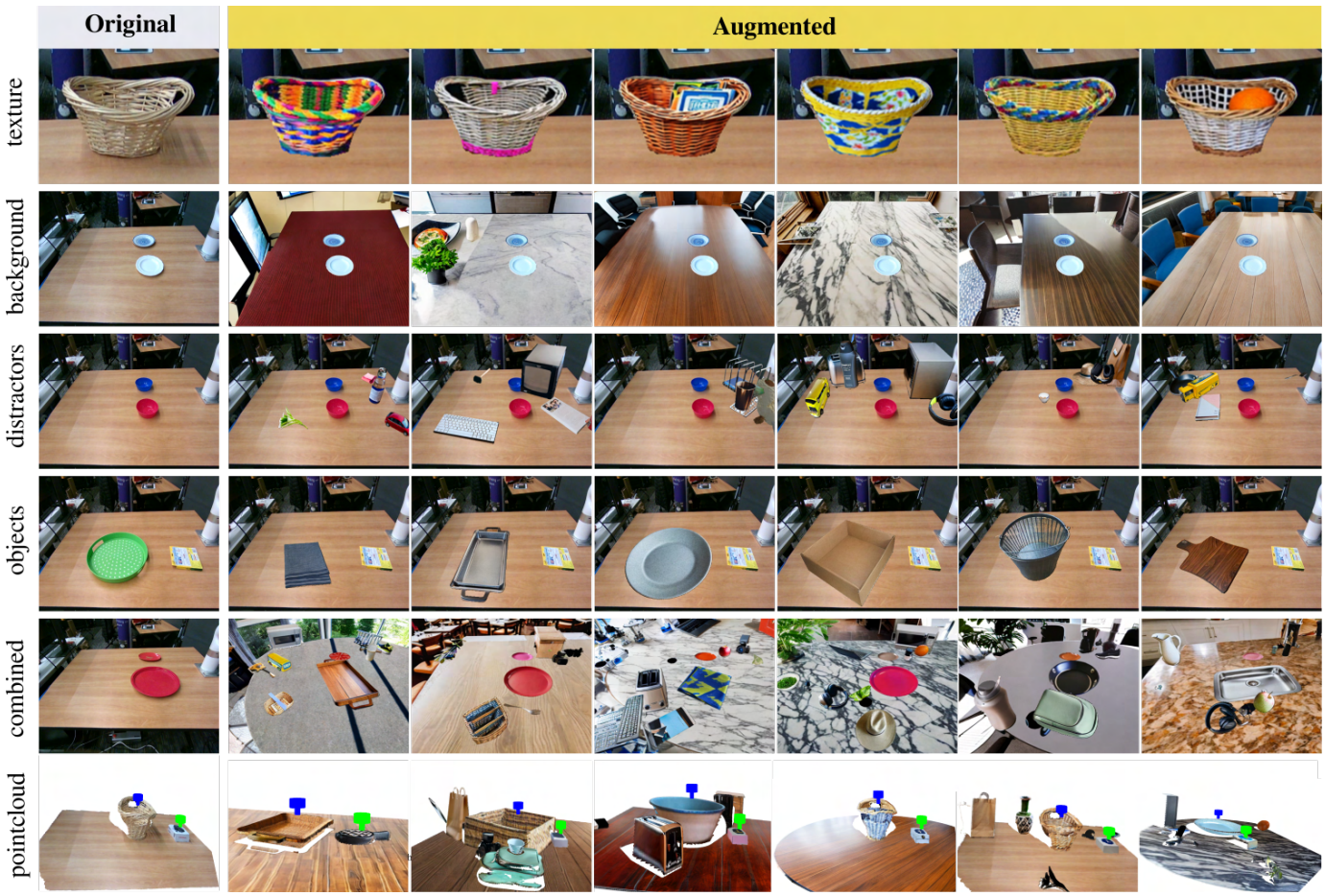

Fig. 3: *GenAug* provides the ability to augment the scene by changing the object texture (first row), changing the background (second row), adding distractors (third row) and changing object categories (fourth row)

a large augmented dataset $\mathcal{D}_{\text{aug}}$ will actually help learned predictors f generalize in real-world settings. The key insight in *GenAug* is that pretrained generative models are trained on the distribution $p_{\text{real}}(o)$ of real images (including real scenes that a robot might find itself in). This lends them the ability to generate (or modify) the training set observations o in a way that corresponds to the distribution of real-world scenes instead of a heuristic approach such as described in [16]. We will use this ability to perform targeted data augmentation for improved generalization of the learned predictor f_θ.

Let us formalize these generative models as $g : \mathcal{T} \times \mathcal{O} \times \mathcal{Z} \rightarrow \mathcal{O}$, mapping from a text description, an image, and a noise vector to a modified image $o' = g(o, t, z); z \sim p(z)$. This includes commonly used text-to-image inpainting models such as Make-A-Video [17], DALL-E 2 [18], Stable Diffusion [19] and Imagen [20]. It is important to note that since these generative models are simply generating images, they are not able to appropriately generate novel actions a, simply novel observations o. This suggests that data generated by these generative models will be able to impose *semantic invariance* on the learned model f_θ, i.e ensure that an equivalence group $\{o, g(t_1, o, z_1), g(t_2, o, z_2), \ldots, g(t_M, o, z_M)\}$ all map to the same action a. To leverage a pretrained text-image generative model for semantic data augmentation, we can simply generate a large set of semantically equivalent observation-action pairs $\{(o, a), (g(t_1, o, z_1), a), (g(t_2, o, z_2), a), \ldots, (g(t_M, o, z_M), a)\}$ for every observation $(o, a) \in \mathcal{D}$ using the generative model g. Note that the generative models cannot simply generate arbitrary observations $g(t, o, z)$, but only observations that retain semantic equivalence, i.e the ground truth actions for the generated augmentations $\{o, g(t_1, o, z_1), g(t_2, o, z_2), \ldots, g(t_M, o, z_M)\}$ are all a. We discuss how to actually instantiate this semantic equivalence in the following section. *GenAug* allows us to generate a large dataset of *realistic* data augmentations, that ensures robustness to various realistic scenes that may be encountered at test time, while still being able to perform the designated task. Unlike typical data augmentation with the hand-defined shifts described above, the generated augmented observations $\{g(t_1, o, z_1), g(t_2, o, z_2), \ldots, g(t_M, o, z_M)\}$ have high likelihood under the distribution of real images $p_{\text{real}}(o)$ that a robot may encounter on deployment. This ensures that the model generalizes to a wide variety of novel scenes, making it significantly more practical to deploy in real world scenarios, since it will be robust to changes in objects, distractors, backgrounds and other characteristics of an environment. It is important to note the limitations

of doing this type of augmentation - it will not be able to generate novel actions a, but instead generate invariances to realistic observational disturbances $o' = g(t, o, z)$ that are generated by the text-image generative model. If not performed carefully, these augmentations can also possibly invalidate the original action a due to factors such as physical inconsistencies or collisions. Next, we discuss a concrete instantiation of this framework in the context of tabletop robotic manipulation.

C. Instantiating GenAug for Tabletop Robotic Manipulation

We scope our discussion of *GenAug* for this work to tabletop rearrangement tasks with a robot arm. This problem has a non-trivial degree of complexity when performed from purely visual input, especially in very cluttered and visually rich domains, and constitutes a significant body of work in robot learning [12, 11]. In particular, we consider tasks where the observation o is a top-down view of the scene, and the action is a spatial action map over the image, indicating where to pick and place objects with a suction-activated gripper. This builds on the transporter networks[12, 11] framework for visual imitation learning. In this section, we describe how to instantiate GenAug for semantic data augmentation for these table-top manipulation problems.

The important question to answer is — how do we use and prompt the generative model g to generate the appropriate equivalence set of semantically equivalent augmented states for an observation o - $\{o, g(t_1, o, z_1), g(t_2, o, z_2), \ldots, g(t_M, o, z_M)\}$, such that the same action a would apply across all of them? We leverage the fact that for a tabletop manipulation task involving picking and placing objects, the same actions are applicable across a wide range of visual appearances including objects being grasped, distractor objects, target receptacles, and backgrounds, as long as the approximate position of the object of interest and the target remain unchanged.

Given a pick-and-place task on a tabletop, we can perform data augmentations on the visual appearance of 1) the object being grasped or the target receptacle, 2) distractor objects 3) the background or table. We will next describe how we can use the text-to-image depth-guided image generation for generating *GenAug*'s augmentations and maximize visual diversity while preserving semantic invariances for each of these types of augmentations.

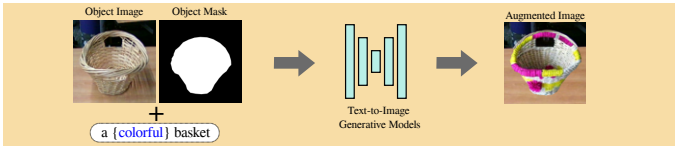

Fig. 4: *GenAug* takes the RGB image and the object mask and uses a depth-guided diffusion model to perform in-category data augmentation.

1) Generating Diverse Grasping Objects and Target Receptacles: Simply generating new scenes with an image generation model is unlikely to retain the semantic invariance that we desire for in *GenAug* since the images will be generated in an uncontrolled way with no regard for functionality. To appropriately retain semantic invariance, we propose a more controlled image generation scheme. In particular, we assume access to masks $\mathcal{M}(o)$ for every observation o, labeling the object of interest and the target receptacle. Note that this is only needed for the small number of demonstrations that are collected, not at inference time. To generate a diversity of visuals, we consider augmentation both "in-category" and "out-of-category", as described below:

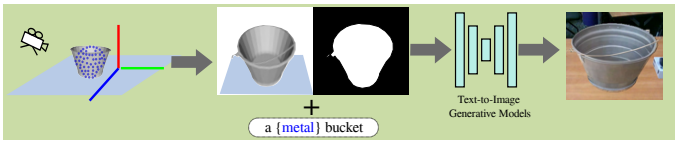

Fig. 5: *GenAug* randomly chooses a new object and place it at the center of the original object to perform cross-category data augmentation

In-Category Generation For in-category generation, GenAug takes the provided mask $\mathcal{M}(o)$ of the object to grasp (or the target receptacle) and the original RGB image, and applies a pretrained depth-guided text-to-image inpainting model [21] to generate novel visual appearances for objects from the same category. To encourage diverse visual appearances of the object, we leverage the fact that the generative model is guided by text and randomly generate novel text prompts involving color (e.g. red, green, yellow, etc) and material (glass, marble, wood, etc) to generate visually diverse objects. Since the object masks remain the same, the resulting positions and shapes are the same, thereby retaining semantic invariance.

Cross-Category Generation While in-category generation provides a degree of visual diversity, it often falls short at generating novel objects altogether and we must consider replacing object categories altogether. When replacing the category of the original object O_i (e.g. a basket) with a new object (e.g. a bucket), one potential technique involves using inpainting models to generate images directly over the masked object (or target receptable). However, naively applying this technique does not generate physically plausible images since it does not appropriately account for geometric consistency during image generation, which causes problems for robotic manipulation.

To allow the generated images to show physical plausibility and 3-D consistency, we leverage a dataset of object meshes to assist the generative model's generation process. In particular, we first render randomly scaled and sized object meshes from a different category without any visual detail to get the perspective correct using the same camera pose, followed by a process of visual generation with a depth-aware generative model, as described for an in-category generation. The object meshes are able to ensure 3-D consistency and physical plausibility, while the generative model allows for significant visual diversity. We note that since we are doing top-down grasping with a suction gripper, even cross-category generation ensures semantic invariance leaving the point of interaction with the object largely unchanged while boosting visual diversity.

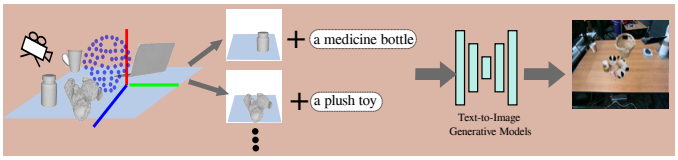

Fig. 6: *GenAug* places a distractor with collision check on the table and uses a depth-guided model to generate realistic-looking objects that are physically plausible.

2) Generating Distractors with Diverse Visual Appearances: While Section II-C1 discusses how to augment the appearance of the object to grasp and the target receptacle, real-world scenarios are often cluttered scenes with several irrelevant distractors. GenAug leverages the same techniques described in Section II-C1 to generate scenes with a diversity of visual distractors. Similar to cross-object augmentation, to add a new distractor D_i, we randomly choose a new object mesh from a family of object assets and render it on the table, followed by visual generation with a text-to-image generative model as described in Section II-C1. Importantly since the distractors must be generated in a way that retains semantic invariance, they must not be in collision with the object to grasp or the target receptacle. To ensure this, we compute collisions by checking for overlapping bounding boxes (in image space) between the generated distractor D_i and masks $\mathcal{M}(o)$ for the object to grasp and the target receptacle and remove this distractor if it is in collision. This ensures semantic invariance while being able to generate very cluttered and diverse scenes, as shown in Figure 6.

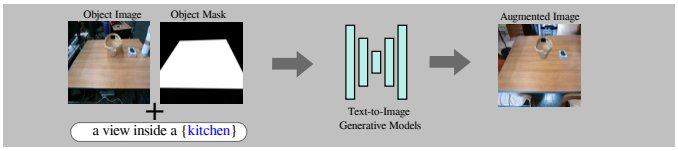

Fig. 7: *GenAug* takes the original RGB image and the mask of the Non-background regions to generate different styles of background scenes such as kitchen, living room, or restaurant.

3) Generating Diverse Backgrounds: To augment not just the appearances of objects, but also the background of the scene while ensuring semantic invariance, we can simply invert the process of generation in Section II-C1 and II-C2. In particular, we can simply hold the object, target receptacle, and distractor masks fixed while asking a text-guided generative model to generate a diverse range of backgrounds, as shown in Figure 7. Since the object masks are all held fixed, their positions remain invariant, while ensuring that the visual appearance of the background and table varies widely.

GenAug leverages these three forms of semantic augmentation - 1) visual object generation, 2) distractor generation and 3) background generation to augment robot learning data with a large amount of semantically invariant, yet visually diverse data. This data has a significant overlap with the types of environments that might be encountered by a system in the real world, and as we show empirically in Section IV, is able to improve the robustness and generalization of robot learning models significantly. Once data is generated with a combination of these three forms of augmentation, we can then simply run standard maximum likelihood techniques for learning manipulation from the augmented dataset. To enable GenAug to be an effective tool for robot learning, we next describe the setup we used for real-world experiments.

III. System Details

A. Hardware Setup

Since *GenAug* is instantiated in this work for tabletop manipulation, we use a robot arm equipped with a suction gripper for all our hardware experiments. In particular, we use the 6 DoF xArm5 with a vacuum gripper manipulator, and control it directly in end-effector space. As shown in Figure 8, we attached the xArm to the end of a large wooden table in a brightly lit room and set up the depth camera on a tripod on one side of the table so that it has a clear view of the robot and any objects on the table.

GenAug requires RGBD observations of the demonstration scenes, masks for the object of interest and the target receptacles, as well as a calibrated camera pose. In our real-world setup, we obtain RGBD images from an Intel RealSense Camera (D435i) and manually label the object masks for the collected demonstrations. While the input for *GenAug* is the camera observations from the RealSense camera, the input observation and the action for the predictive model f_θ operate on a top-down view, as described in [12].

In order to guide the robot to complete the tasks in the cluttered environment, we largely build on the architecture and training scheme of CLIPort [11], which combines the benefits of language-conditioned policy learning with transporter networks [12] for data-efficient imitation learning in tabletop settings. *GenAug* replaces the imitation learning dataset in CLIPort with a much larger augmented one, as described in Section III-C. Implementation details can be found in Appendix B.

B. Demonstration Collection

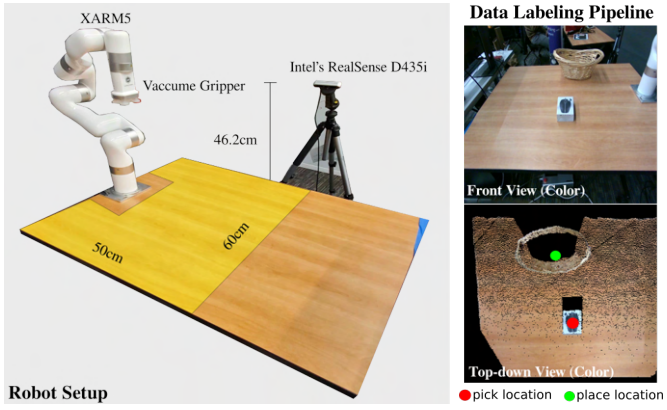

Fig. 8: An illustration of our robot experiment setup and data labeling pipeline. A user clicks locations on a top-down view image, to indicate pick (red) and place (green) locations in the robot space.

To collect demonstrations, we rely on humans to collect action labels for various pick-and-place tasks. We first project the front camera view to a 2D top-down image and height map of the scene. The user can manually click locations on

the top-down images to indicate the pick and place locations. These locations are then converted to end-effector positions in full Cartesian space, which is then provided to a low-level controller that uses inverse kinematics and position control. We save the demonstration if the robot can successfully complete the task. We collect 10 demonstrations per task and 10 tasks in total, all in one single environment as shown as "Demo Environment" in Figure 9. Details of each task can be found in Appendix A.

C. Augmentation Infrastructure

As described in Section III-A, *GenAug* requires object meshes to generate cross-category augmentations and distractors. To perform this augmentation, we use 40 object meshes from the GoogleScan dataset [22] and Free3D [23]. Of these, we choose 11 objects to augment the original object of interest and 12 objects to augment the target receptacle. Any of the remaining 38 objects are then randomly chosen as distractors. During augmentation, we randomly select which components (table, object texture, shape, distractors) to change to generate the augmented training dataset. For each demonstration, we apply *GenAug* 100 times resulting in 1000 augmented environments per task. The augmented data is then passed into Cliport [11] to learn a language-conditioned policy.

IV. Experiment

We evaluate the effectiveness of *GenAug* in both the real world and simulation. Our goal is to: (1) demonstrate *GenAug* is practical and effective for real-world robot learning, (2) compare *GenAug* with other baselines in end-to-end vision manipulation tasks, (3) investigate different design choices in *GenAug*. We will first show our results in a real-world setting, followed by an in-depth baseline study in simulation.

A. Real-world Experiment

1) Design of demo and test environment: To show the generalization capability of a model trained with *GenAug*, we collect demonstrations of 10 tasks in one single environment and create different styles of test environments such as "Playground", "Study Desk", "Kitchen Island", "Garage" and "Bathroom" as shown in Figure 9. For evaluation, we randomly add and rearrange objects from each test style and create unseen environments. Please see Appendix A for further details.

2) Result: We train CLIPort with augmented RGBD and text prompts for tasks collected in the real world and evaluate in various unseen environments. In particular, for each task, we randomly choose an environment style from Figure 9, randomly rearrange and add objects on the table to create 10 unseen environments, 10 scenes with unseen objects to pick, and 10 scenes with unseen objects to place. We observe that *GenAug* shows a significant generalization to unseen environments with an average of 85% success rate. On more challenging tasks of unseen objects to pick or place, *GenAug* is able to achieve 45% and 52% success rates, which are expected to improve with more demonstrations and more object meshes for augmentation. Results for each task can be found in Table I.

Fig. 9: Demonstration environment and examples of test environments used in our robot experiments.

3) Baselines for real-world experiments: To further show the effectiveness of *GenAug*, we compare our approach with CLIPort trained without *GenAug*, shown in Table II. To ensure both methods are tested with the same input observations, we evaluate the success rate by comparing the predicted pick and place affordances with ground truth locations. For each task, we evaluate both methods on 5 unseen environments, 5 unseen objects to pick, and 5 unseen objects to place. By averaging the success rates from Table II, we observe *GenAug* provides a significant improvement for zero-shot generalization. In particular, *GenAug* achieves 80% success rate on unseen environments compared to 38% without *GenAug*. On unseen objects to place, *GenAug* achieves 54% success rate compared to 8% without. Finally, *GenAug* achieves 46% success rate on unseen objects to pick compared to 10% without. We visualize and compare the differences in their predicted affordances in Appendix A.

B. Simulation

To further study in depth the effectiveness of *GenAug*, we conduct large-scale experiments with other baselines in simulation. In particular, we organize baseline methods as (1) in-domain augmentation methods and (2) learning from out-of-domain priors, as described below.

1) In-domain augmentation methods: (1) "No Augmentation" does not use any data augmentation techniques. (2) "Spatial Augmentation" randomly transforms the cropped object image features to learn rotation and translation equivariance, as introduced in TransporterNet [12]. (3)"Random Copy Paste" randomly queries objects and their segmented images from LVIS dataset [24], and places them in the original scene. This includes adding distractors around the pick or place objects or replacing them. Further visualization of this approach can be found in Appendix C. (4)"Random Background" does not modify the pick or place objects but replaces the table and background with images randomly selected from LVIS dataset. (5)"Random Distractors" randomly selects segmented images from LVIS dataset as distractors.

2) Learning from out-of-domain priors: In addition, we investigate whether learning from a pretrained out-of-domain

TABLE I: Real-World Robot Experiments tested on 10 tasks. On average, *GenAug* achieves 85% success rate on unseen environment, 52% on unseen object to place, and 45% on unseen object to pick.

	bowl to Coaster	box to basket	red bowl to blue bowl	red plate to tray	box to tray	plate to box	white plate to blue plate	coaster to salt container	coaster to dust bin	mouse box to fruit box
Unseen Environment	0.8	0.9	1	1	1	0.9	0.9	1	0.5	0.5
Unseen Place	0.7	1	0.5	0.3	0.6	0.3	0.4	0.4	0.4	0.6
Unseen Pick	0.2	0.6	0.5	0.6	0.7	0.3	0.3	0.7	0	0.6

TABLE II: Evaluating with and without *GenAug* on unseen scenes collected in the real world across 10 tasks. On average, *GenAug* shows significant improvement in unseen environments and objects.

	box to tray			**box to basket**			**coaster to dust pan**			**plate to tray**			**bowl to coaster**		
	env	pick	place	env	pick	place	env	pick	place	env	pick	place	env	pick	place
No GenAug	0.8	0	0	0.2	0.2	0	0.8	0.4	0.4	0	0	0	0	0	0
GenAug	1	0.6	1	0.6	0.6	0.8	1	0.4	0.4	1	0.4	0.2	0.6	0.6	0.6
	plate to plate			**box to box**			**plate to box**			**coaster to salt**			**bowl to bowl**		
	env	pick	place	env	pick	place	env	pick	place	env	pick	place	env	pick	place
No GenAug	0	0	0.2	0.2	0	0	0.6	0.2	0	0.2	0	0.2	1	0.2	0
GenAug	1	0	0.6	0.8	0.4	0.4	1	0.8	0	1	0.4	0.4	1	0.4	1

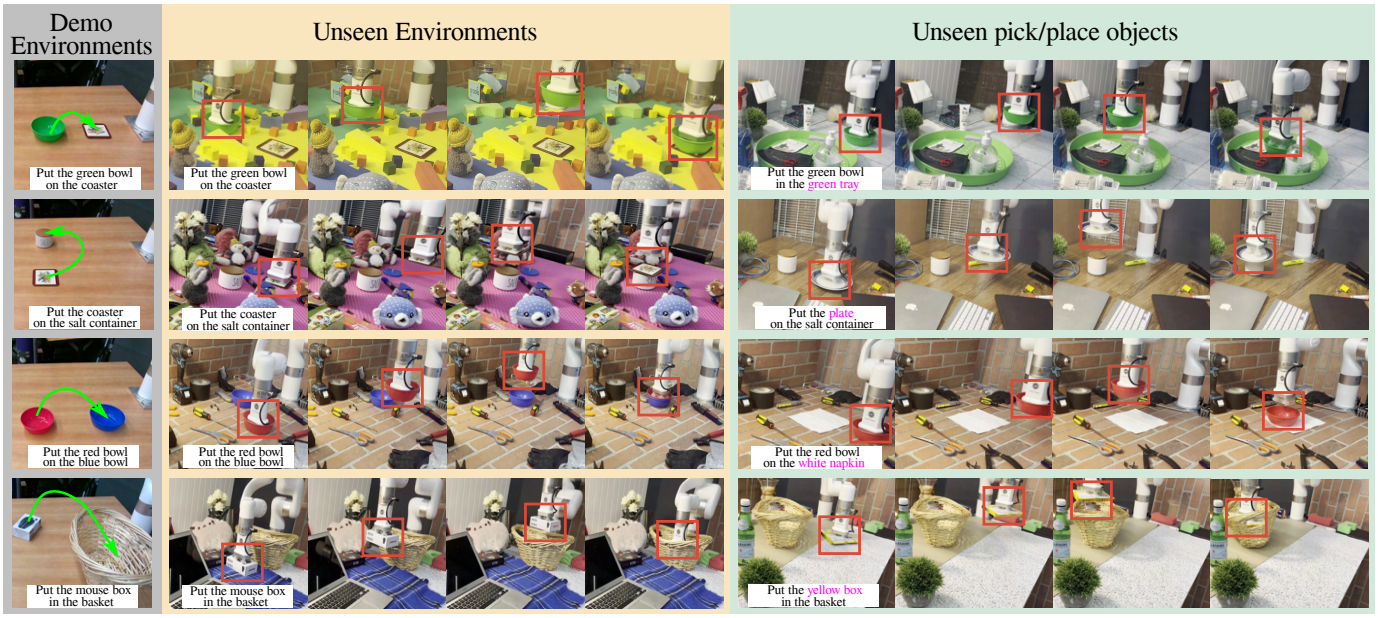

Fig. 10: Examples of real-world experiments. Given demonstrations in one simple environment, GenAug enables the robot to generalize unseen environments and objects.

visual representation can improve the zero-shot capability on challenging unseen environments. In particular, we initialize the network with pre-trained R3M [25] weights and finetune it on our dataset.

We use baselines described above with two imitation learning methods: TransportNet [12] and CLIPort [11]. Since all the baselines cannot update the depth of the augmented images, we only use RGB images instead of RGBD used in the original TransporterNet and CLIPort. For each baseline, we train 5 tasks in simulation and report their average success rate in Table III. We observe *GenAug* significantly outperforms other approaches in most of the tasks. One interesting observation is that randomly copying and pasting segmented images or replacing the background images can provide reasonable robustness in unseen environments but are not able to achieve similar performance as *GenAug* at unseen objects. This indicates generating physically plausible scenes that are semantically meaningful is important. Details of tasks in simulation experiments can be found in Appendix B.

C. Ablations

In this section, we aim to study different design choices in *GenAug*. In particular, our goal is to investigate (1) how the number of augmentations affects the generalization performance to unseen environments, (2) when *GenAug* will fail in real-world unseen environments. We further justify the choice of using depth-guided models and compare this with in-painting models in Appendix A.

1) Impact of the number of augmentations: Given the task "put the brown plate in the brown box" in simulation, we apply *GenAug* $0, 10, 50$ and 100 times and compare their success rate on 100 scenes of "unseen environment", 100 scenes of "unseen object to pick" and 100 scenes of "unseen object to place". As shown in Figure 11, the performance improves as the number of augmentations increases, which indicates the importance of using augmentation as a way to robustify the generalization capability.

TABLE III: Baseline experiments evaluated in simulation. We compare the average performance of *GenAug* with other methods on 5 pick-and-place tasks and observe *GenAug* provides a significant improvement at unseen environments and objects.

	Unseen Environment						Unseen to place						Unseen to pick					
	TransporterNet			CLIPort			TransporterNet			CLIPort			TransporterNet			CLIPort		
Method	1	10	100	1	10	100	1	10	100	1	10	100	1	10	100	1	10	100
No Augmentation	4.8	8.1	9.8	11.7	14.3	14.4	15.1	30.4	52.6	39.4	40.8	44.6	8.5	34.6	54.9	46.0	67.0	64.1
Spatial Augmentation	11.0	12.2	8.3	23.3	16.1	26.7	44.3	50.5	65.3	26.1	36.9	50.7	53.6	57.2	66.4	38.2	56.9	80.3
Random Copy Paste	**53.1**	67.0	73.5	38.2	39.8	64.3	55.1	65.4	**84.9**	39.7	55.9	73.9	48.3	67.0	76.1	52.5	65.0	81.0
Random Background	53.0	75.3	79.1	33.6	62.2	55.4	24.5	22.1	35.5	7.6	9.9	17.9	44.4	40.7	35.9	19.2	52.7	72.3
Random Distractors	10.1	9.7	13.7	15.4	36.2	35.8	28.2	60.7	66.0	27.5	51.8	54.3	42.5	47.4	62.3	31.0	64.0	69.1
R3M Finetune	4.1	6.0	4.8	22.2	16.8	20.9	43.5	40.6	41.9	30.9	43.5	57.5	45.6	45.7	41.1	46.7	50.7	72.7
GenAug	43.9	58.5	77.6	46.6	57.0	71.9	**69.1**	**76.5**	83.6	62.6	**83.9**	**86.3**	**75.3**	75.6	**87.2**	**61.5**	**77.7**	**83.1**
***GenAug* (w Depth)**	47.8	**83.8**	**91.2**	**47.2**	**60.9**	**73.4**	39.9	67.2	74.2	**64.8**	73.8	84.6	71.2	**83.4**	87.1	56.2	67.3	81.5

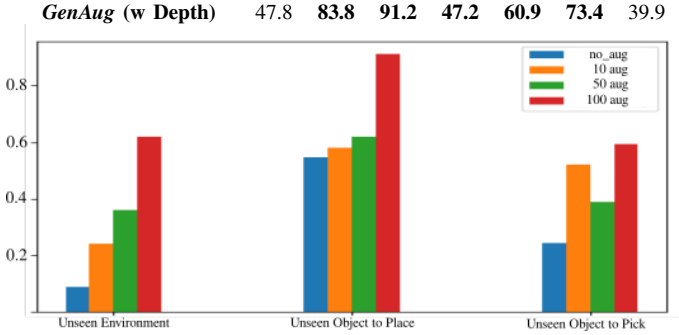

Fig. 11: Analysis of the number of augmentation on unseen scenes.

2) Failure cases: In addition, we also analyze failure cases and visualize them in Appendix A. We observe failure cases usually happen when the object and background share similar colors or the table is cluttered with too many objects.

V. Related Work

*a) **Image space Augmentation**:* In the absence of diverse data, a promising direction is finding ways to inject structure directly into learned models for widespread generalization. The most widely used technique is various forms of data augmentation [26], such as cropping, shifting, noise injection and rotation. These methods have been used in many robot learning approaches and provide a significant improvement in data efficiency [13, 14, 15, 27]. For example, [28] investigate different augmentation modes in Meta-learning settings. In addition, several methods attempt to enforce geometric invariance through architectural innovations such as [29] and [30]. While these methods can provide a local notion of robustness and invariance to perceptual noise, they do not provide generalization to novel object shapes or scenes. More recently out-of-domain models have started making their way into robot learning. For example - [31] uses large text-image models like Dall-e [18] to generate favorable image goals for robots, and CACTI [32] shows adding distractor objects using in-painting models [19] improves multi-task policies. These approaches while helpful in task specification, provide limited benefit for robots to generalize to entirely unseen situations. In contrast, *GenAug* induces semantic changes to the observations thereby helping acquire behavior invariance to new scenes.

*b) **Leveraging Simulation Data**:* A natural solution to this problem is to leverage simulation data [33]. However, while this method is efficient at generating a large *quantity* of data, it can be challenging to create diverse content (objects meshes, physics, layouts, and visual appearances), and often the resulting simulations exhibit a significant gap from reality. To address this issue, domain randomization [34, 35] varies multiple simulation parameters in order to boost the effectiveness of trained policies in the real world. Randomization of lighting and camera parameters during training can allow a policy to be invariant to the effects of lighting and visual perturbations in the real world. Physics parameters of the scene can also be randomized to transfer policies trained in simulation to the real world [36]. While effective, challenges in creating a simulation with visual and physical realism for every task and behavior severely restrict the applicability of these methods to isolated known tasks and limited diversity. Furthermore, they require the user to define augmentation parameters and their ranges which can be very nontrivial for complex tasks [37, 38].

VI. Limitations

Action Assumption: Despite showing promising visual diversity, *GenAug* does not augment action labels and reason about physics parameters such as material, friction, or deformation, thus it assumes the same action still works on the augmented scenes. For the augmented cluttered scenes, *GenAug* assumes the same action trajectory is not colliding with the augmented objects.

Augmentation and Speed: *GenAug* cannot guarantee visual consistency for frame augmentation in a video. *GenAug* usually takes about 30 seconds to complete all the augmentations for one scene, which might not be practical for some robot learning approaches such as on-policy RL.

VII. Conclusion

We present *GenAug*, a novel system for augmenting real-world robot data. *GenAug* leverages a data augmentation approach that bootstraps a small number of human demonstrations into a large dataset with diverse and novel objects. We demonstrate that *GenAug* is able to train a robot that generalizes to entirely unseen environments and objects, with 40% improvement over training without *GenAug*. For future work, we hope to investigate *GenAug* in other domains of robot learning such as Behavior Cloning and Reinforcement learning, and extend our table-top tasks into more challenging manipulation problems. Moreover, investigating whether a combination of language models and vision-language models can yield impressive scene generations would be a promising direction in the future.

VIII. Acknowledgement

We thank Aaron Walsman for helping with transporting materials from Home Depot for creating robot environments. We thank Mohit Shridhar for the discussions about CLIPort training. We thank all members from the WEIRD lab, the RSE lab at the University of Washington, as well as Kay Ke, Yunchu Zhang, and Abhay Deshpande for many discussions, support, late-night food ordering, and snack sharing. We are thankful to Boling Yang and Henri Fung for their patience in letting us book all the slots for using the machine before the deadline. We are also grateful to Selest Nashef for his help to keep our robot experiments safe and organized. Part of this work was done while Zoey Chen was an Intern at Meta AI. The work was also funded in part by Amazon Science Hub.

References

[1] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *CoRR*, abs/1504.00702, 2015. URL http://arxiv.org/abs/1504.00702.

[2] Anusha Nagabandi, Kurt Konoglie, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *CoRL*, 2019.

[3] Ahmed Hussain Qureshi, Mayur J. Bency, and Michael C. Yip. Motion planning networks. *CoRR*, abs/1806.05767, 2018. URL http://arxiv.org/abs/1806.05767.

[4] Adam Fishman, Adithyavairavan Murali, Clemens Eppner, Bryan Peele, Byron Boots, and Dieter Fox. Motion policy networks. *CoRR*, abs/2210.12209, 2022. doi: 10.48550/arXiv.2210.12209. URL https://doi.org/10.48550/arXiv.2210.12209.

[5] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *CoRR*, abs/1806.10293, 2018. URL http://arxiv.org/abs/1806.10293.

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael S. Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-1: robotics transformer for real-world control at scale. *CoRR*, abs/2212.06817, 2022. doi: 10.48550/arXiv.2212.06817. URL https://doi.org/10.48550/arXiv.2212.06817.

[7] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. In Nancy M. Amato, Siddhartha S. Srinivasa, Nora Ayanian, and Scott Kuindersma, editors, *Robotics: Science and Systems XIII, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, July 12-16, 2017*, 2017. doi: 10.15607/RSS.2017.XIII.058. URL http://www.roboticsproceedings.org/rss13/p58.html.

[8] Dean Pomerleau. ALVINN: an autonomous land vehicle in a neural network. In David S. Touretzky, editor, *Advances in Neural Information Processing Systems 1, [NIPS Conference, Denver, Colorado, USA, 1988]*, pages 305–313. Morgan Kaufmann, 1988. URL http://papers.nips.cc/paper/95-alvinn-an-autonomous-land-vehicle-in-a-neural-network.

[9] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: an open large-scale dataset for training next generation image-text models. *CoRR*, abs/2210.08402, 2022. doi: 10.48550/arXiv.2210.08402. URL https://doi.org/10.48550/arXiv.2210.08402.

[10] 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 Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society, 2009. doi: 10.1109/CVPR.2009.5206848. URL https://doi.org/10.1109/CVPR.2009.5206848.

[11] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.

[12] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. *arXiv preprint arXiv:2010.14406*, 2020.

[13] Gregory Benton, Marc Finzi, Pavel Izmailov, and Andrew G Wilson. Learning invariances in neural networks from training data. *Advances in Neural Information Processing Systems*, 33:17605–17616, 2020.

[14] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[15] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal*

of big data, 6(1):1–48, 2019.

[16] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[17] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

[18] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[19] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[20] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

[21] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[22] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. *arXiv preprint arXiv:2204.11918*, 2022.

[23] free3d. free3d. https://free3d.com/.

[24] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[25] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[26] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *J. Big Data*, 6:60, 2019. doi: 10.1186/s40537-019-0197-0. URL https://doi.org/10.1186/s40537-019-0197-0.

[27] Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

[28] Afia Zafar, Muhammad Aamir, Nazri Mohd Nawi, Ali Arshad, Saman Riaz, Abdulrahman Alruban, Ashit Kumar Dutta, and Sultan Almotairi. A comparison of pooling methods for convolutional neural networks. *Applied Sciences*, 12(17):8643, 2022.

[29] Dian Wang, Mingxi Jia, Xupeng Zhu, Robin Walters, and Robert Platt. On-robot policy learning with o(2)-equivariant SAC. *CoRR*, abs/2203.04923, 2022. doi: 10.48550/arXiv.2203.04923. URL https://doi.org/10.48550/arXiv.2203.04923.

[30] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector neurons: A general framework for so (3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12200–12209, 2021.

[31] Ivan Kapelyukh, Vitalis Vosylius, and Edward Johns. Dall-e-bot: Introducing web-scale diffusion models to robotics. *arXiv preprint arXiv:2210.02438*, 2022.

[32] Zhao Mandi, Homanga Bharadhwaj, Vincent Moens, Shuran Song, Aravind Rajeswaran, and Vikash Kumar. Cacti: A framework for scalable multi-task multi-scene visual imitation learning. *arXiv preprint arXiv:2212.05711*, 2022.

[33] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied AI using procedural generation. *CoRR*, abs/2206.06994, 2022. doi: 10.48550/arXiv.2206.06994. URL https://doi.org/10.48550/arXiv.2206.06994.

[34] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[35] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. 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*, pages 969–977, 2018.

[36] Josh Tobin, Lukas Biewald, Rocky Duan, Marcin Andrychowicz, Ankur Handa, Vikash Kumar, Bob McGrew, Alex Ray, Jonas Schneider, Peter Welinder, et al. Domain randomization and generative models for robotic grasping. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3482–3489. IEEE, 2018.

[37] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[38] Ankur Handa, Arthur Allshire, Viktor Makoviychuk, Aleksei Petrenko, Ritvik Singh, Jingzhou Liu, Denys Makoviichuk, Karl Van Wyk, Alexander Zhurkevich, Balakumar Sundaralingam, et al. Dextreme: Transfer of agile in-hand manipulation from simulation to reality. *arXiv preprint arXiv:2210.13702*, 2022.

APPENDIX A
REAL-WORLD EXPERIMENTS

A. Real-World Tasks

We collect 10 pick-and-place tasks in the real world in one single environment, as shown in Figure 15. All tasks are collected with only 10 demonstrations in one single environment. For each demonstration, we randomly place objects within the work zone of the table. To augment the demonstration, we apply GenAug 100 times per demonstration, resulting in 1000 augmented demonstrations for each task.

B. Baselines in the real-world

We use the ground truth of pick and place locations to evaluate the performance of affordance prediction trained with GenAug and without, as shown in Figure 12

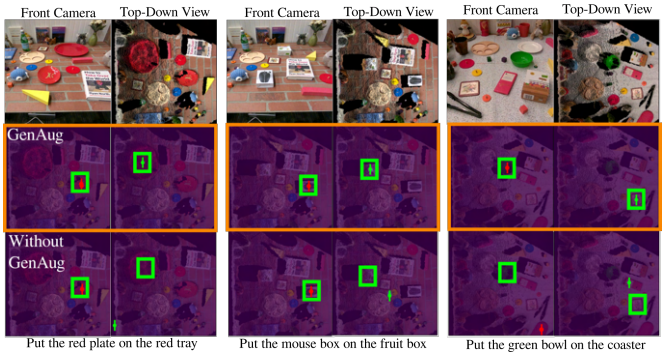

Fig. 12: Comparison between training with and without GenAug by comparing pick and place affordances predicted by two models. GenAug significantly improves generalization over unseen environments and objects compared to training without GenAug. pick affordances are highlighted in red, and place affordances are highlighted in green. Ground truth locations are represented in green boxes

C. Failure modes

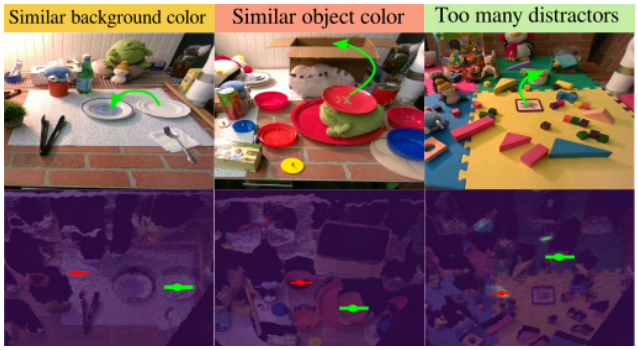

Fig. 13: Failure cases observed in the real-world setting

We observe failure cases usually occur when the background color is similar to the pick or place object. Or one of a few distractors has a very bright color or similar color. We expect this can be improved by increasing the number of augmentations in the training set, such that the training data can have higher coverage of possible combinations of the scenes.

D. depth-guided diffusion model vs inpainting

We further justify the choice of using a depth-guided diffusion model, as shown in Figure 14. Directly using the inpainting model often does not result in reasonable visual augmentation. Instead, GenAug uses a depth-guided diffusion model together with predefined 3D meshes, resulting in realistic new objects and scenes.

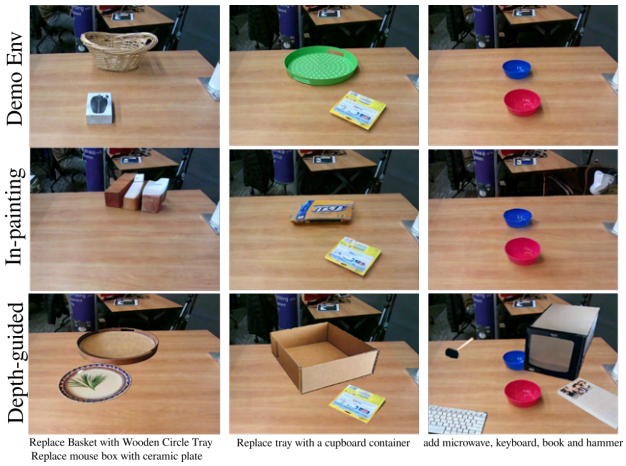

Fig. 14: Comparison between depth-guided diffusion model with access to predefined 3D meshes and inpainting models.

E. Real-World Unseen Environments

In this section, we visualize examples of unseen test scenes that are used for evaluation for all 10 tasks, as shown in Figure 22

F. Real-World Evaluation

We visualize the demonstration environments collected in the real world as well as their corresponding unseen test environments and objects. Please see our website for better visualization https://genaug.github.io.

We visualize pick and place affordances predicted by CLIPort trained with GenAug in Figure 19

APPENDIX B
SIMULATION EXPERIMENTS

A. Tasks

We perform a large-scale evaluation in simulation. In particular, we collect 1, 10, 100 demonstrations for 5 tasks: "Pack the brown round plate", "Pack the straw hat", "Pack the green and white striped towel", "Pack the grey soccer shoe" and "Pack the porcelain cup", and report the average success rate across all tasks. Following evaluation metrics defined in CLIPort [11] and TransporterNet [12], the success rate is defined as the total volume of the pick object inside the place object, divided by the total volume of the pick object in the scene.

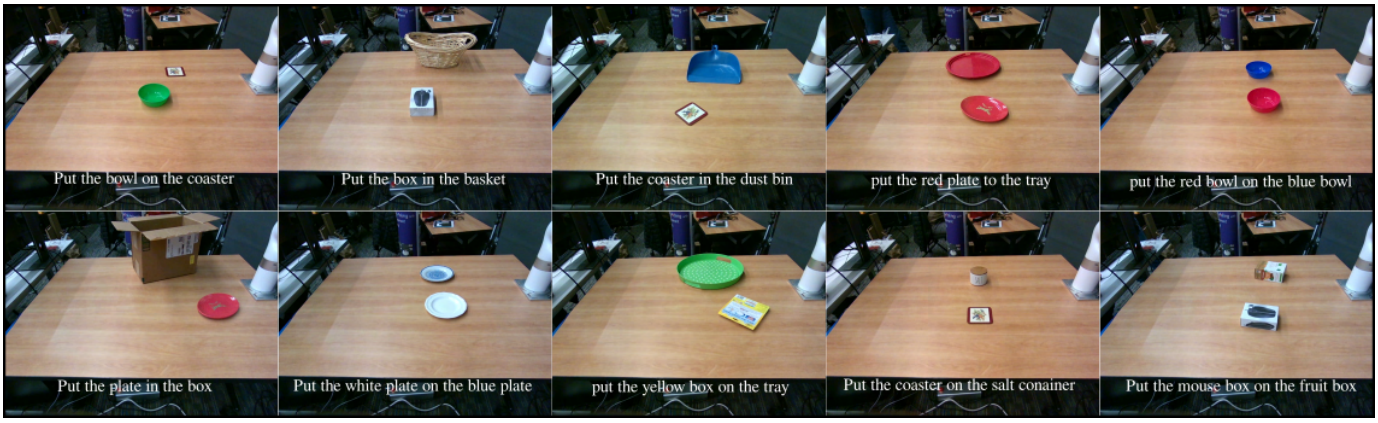

Fig. 15: Tasks used in the real-world experiments.

Fig. 16: Augmented dataset for demonstrations collected in simulation.

B. Augmented Dataset in Simulation

Given demonstrations from a task collected in simulation, we apply GenAug 100 times for each demonstration. We visualize examples of the augmented dataset in Figure 16.

We also observe diverse visual augmentation on the same object template, as shown in Figure 17. Given different text prompts, GenAug is able to generate different and realistic textures.

Fig. 17: Diversity of the appearance of the generated objects

APPENDIX C
VISUALIZATION OF BASELINE DATA AUGMENTATION

We visualize some examples of randomly copying and pasting segmented images from LVIS dataset [24], as shown in Figure 18.

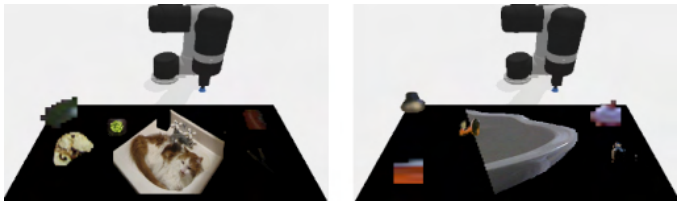

Fig. 18: Examples of random copy and paste baseline. We extracted queried segmented images from LVIS dataset and paste them directly on the original demonstration image. This usually leads to low-quality and incomplete image generation.

We observe this baseline often results in unrealistic, low-quality image generation, which is not usually matching observations during test time in both real-world and simulation.

APPENDIX D
VISUALIZATIONS FOR REAL-WORLD EXPERIMENTS

In this section, we visualize more examples of applying GenAug on demonstrations collected in the real world, as shown in Figure 20. We train CLIPort [11] with such a dataset and evaluate unseen environments and objects for 10 tasks. We further show affordance predictions in Figure 19 and Figure 21.

Fig. 19: Prediction of pick and place locations on various tasks with GenAug

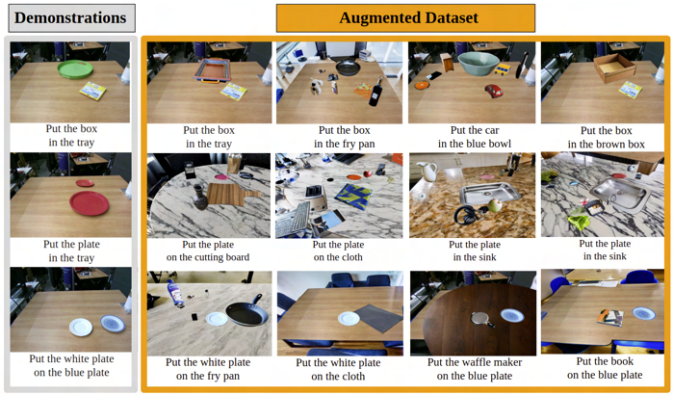

Fig. 20: Examples of augmented dataset given observations of demonstrations collected in a simple environment.

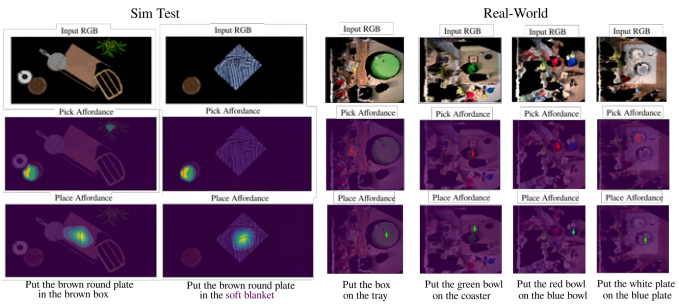

Fig. 21: Pick and Place affordance predicted by CLIPort that trained on GenAug on unseen environments and objects in simulation and the real world.

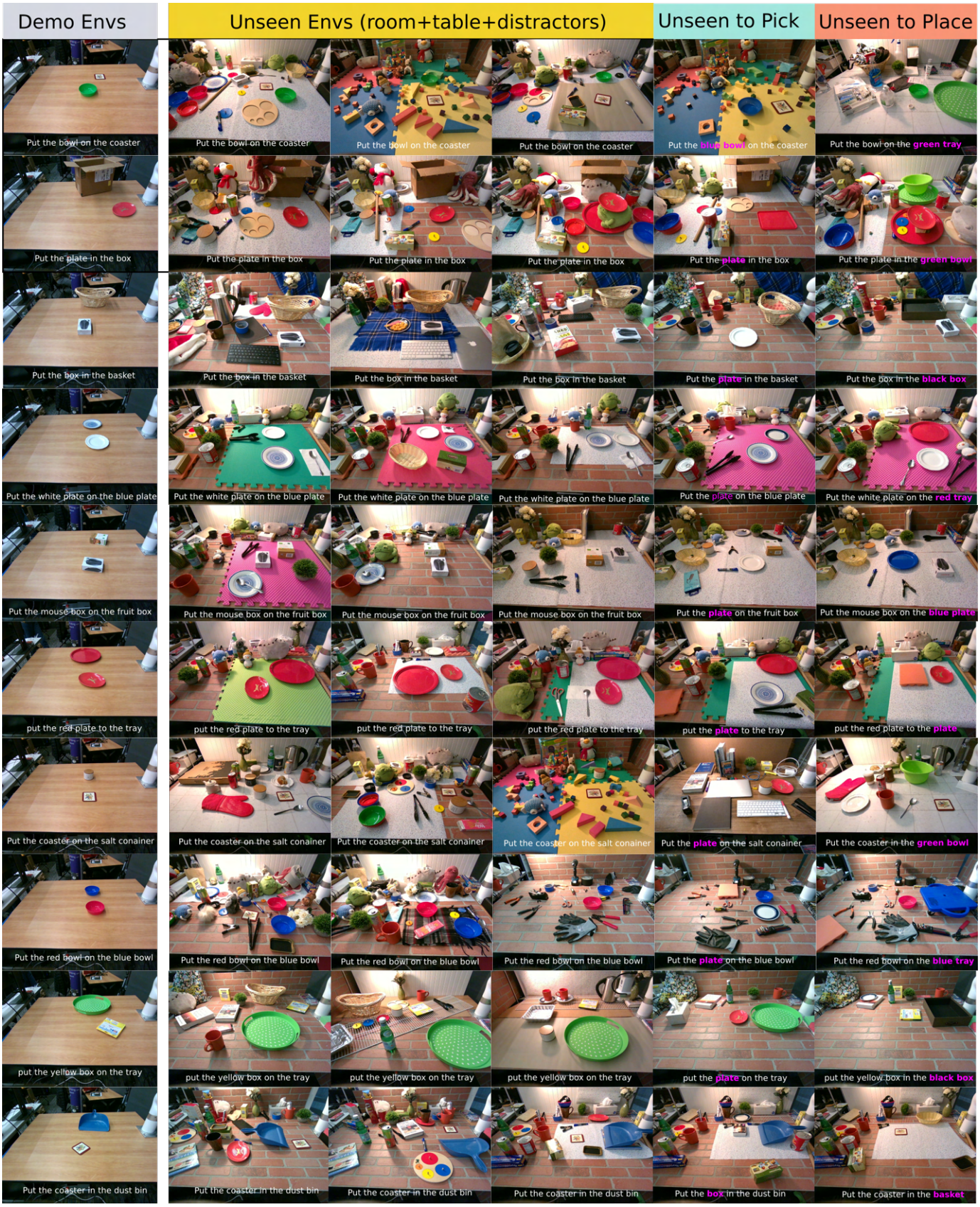

Fig. 22: Unseen test set in the real-world experiments.