THE COLOSSEUM: A Benchmark for Evaluating Generalization for Robotic Manipulation

Wilbert Pumacay*
Department of Computer Science,
Universidad Católica San Pablo

Ishika Singh*
Department of Computer Science,
University of Southern California

Jiafei Duan*
University of Washington

Ranjay Krishna
University of Washington
Allen Institute for Artifical Intelligence

Jesse Thomason
Department of Computer Science,
University of Southern California

Dieter Fox
University of Washington
NVIDIA

*equal contribution
robot-colosseum.github.io

***Abstract*—To realize effective large-scale, real-world robotic applications, we must evaluate how well our robot policies adapt to changes in environmental conditions. Unfortunately, a majority of studies evaluate robot performance in environments closely resembling or even identical to the training setup. We present THE COLOSSEUM, a novel simulation benchmark, with 20 diverse manipulation tasks, that enables systematical evaluation of models across 14 axes of environmental perturbations. These perturbations include changes in color, texture, and size of objects, table-tops, and backgrounds; we also vary lighting, distractors, physical properties perturbations and camera pose. Using THE COLOSSEUM, we compare 5 state-of-the-art manipulation models to reveal that their success rate degrades between 30-50% across these perturbation factors. When multiple perturbations are applied in unison, the success rate degrades $\geq$75%. We identify that changing the number of distractor objects, target object color, or lighting conditions are the perturbations that reduce model performance the most. To verify the ecological validity of our results, we show that our results in simulation are correlated ($\bar{R}^2 = 0.614$) to similar perturbations in real-world experiments. We open source code for others to use THE COLOSSEUM, and also release code to 3D print the objects used to replicate the real-world perturbations. Ultimately, we hope that THE COLOSSEUM will serve as a benchmark to identify modeling decisions that systematically improve generalization for manipulation.**

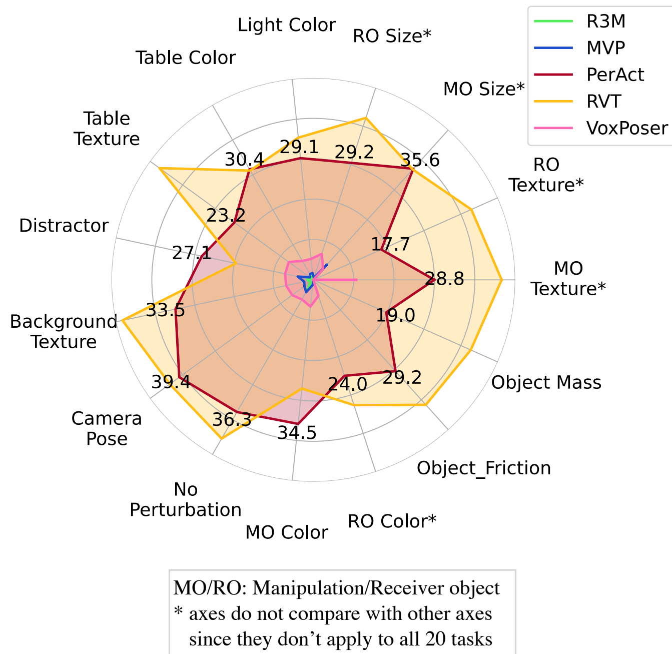

Fig. 1: **Evaluating generalization with THE COLOSSEUM.** Task-averaged success rate for 5 SotA robotic manipulation policies over 14 perturbation factors and 20 robotic manipulation tasks. Changes in RGB input space affects all models due to end-to-end RGB-based training. Image-based models are also affected by camera pose change, while models without in-the-wild pretraining suffer in the presence of distractors.

I. INTRODUCTION

The promise of robotics requires ubiquity. For effective real-world deployment, robots must operate in a variety of environments. When asked to turn on a stove, a robot should be able to turn the stove's knob, regardless of the size of the knob, irrespective of the kitchen's backdrop, invariant to the kitchen counter's texture, during the day, or even under a dim evening light. Unfortunately, a majority of studies evaluate robot performance in environments closely resembling or even identical to the training setup [60, 21, 4, 10].

Naturally, generalization to environmental conditions has been a large focus in recent literature. Both Reinforcement Learning (RL) [63, 44, 40] and Behavior Cloning (BC) [71, 60, 43, 21] struggle with generalization if not trained on sufficiently representative data. In response, robotics researchers have recently released large-scale diverse behavior cloning datasets, with trajectories collected either in simulation [17, 19] or in the real world [48]. With these datasets, different techniques—including data augmentation [35, 67, 60], pretraining on large vision and robot datasets for BC [43, 53, 5], and incorporating 3D priors [60, 59, 20, 61]—claim to improve generalization for manipulation tasks. Although these tech-

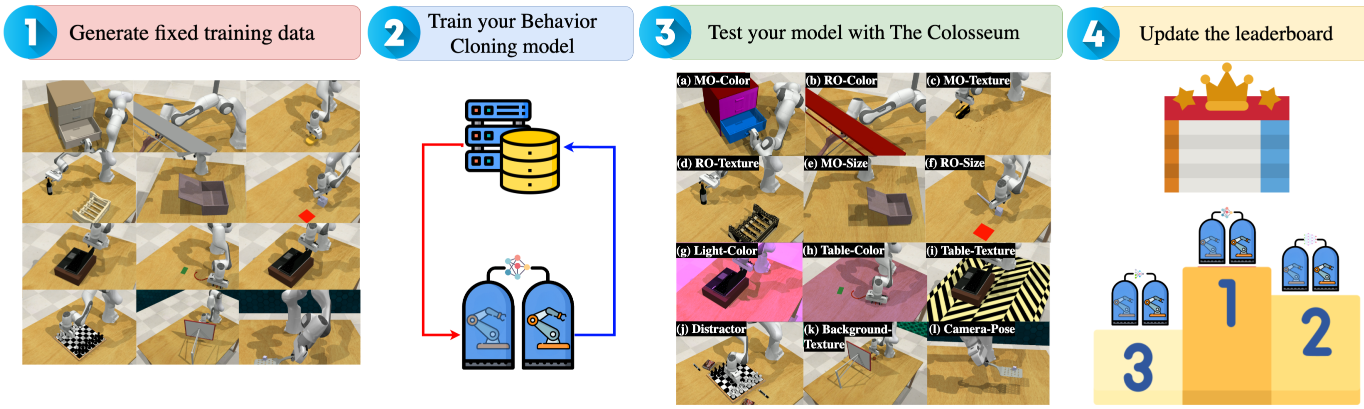

Fig. 2: **THE COLOSSEUM Challenge.** This challenge is designed to enhance generalization of Behavior Cloning (BC) models in robotic manipulation tasks. It involves four key phases: 1) Participants generate a standard training dataset from 20 tasks with 100 demonstrations each, without `perturbation_factors`. 2) Participants train their BC models using this standardized dataset. 3) The models are restricted to evaluate over a fixed 25 episodes across 14 different `perturbation_factors`. 4) Models are ranked on a leaderboard based on the percentage change in their performance across these factors. We've shown that simulation aligns with real-world evaluation, so participants can expect similar generalization when participating in the simulation benchmark.

niques showcase improvements, the evaluation benchmarks are not designed to stress-test the policies against systematic perturbations to the environment.

We introduce THE COLOSSEUM, a comprehensive benchmark aimed at systematically evaluating the generalization of robot manipulation to environmental perturbations. THE COLOSSEUM introduces perturbations across 20 different tasks from the RLBench [31] framework, spanning 14 dimensions of perturbations. These perturbations include object color, object texture, object size, table color, table texture, the presence of distractor objects, changes to the camera pose, and changes to physical properties like friction and mass. THE COLOSSEUM also includes a parallel real world evaluation with task setups and objects reproducible via open-sourced 3D printing models.

We evaluate four state of the art BC models using THE COLOSSEUM and draw insights into answers for critical research questions on generalization for BC policies. Considering 3D versus 2D reasoning methods, we find that 3D-based BC models demonstrate superiority over 2D-based methods in terms of overall task performance when using a fixed set of training data while also achieving better robustness to environmental perturbations. Among 2D and 3D models, distractors, color-related and lighting perturbations have the most significant impact on task success. Conversely, perturbations on object size had less impact in both settings. Finally, we establish a strong correlation between falling task success under perturbations in simulations and those observed in real-world scenarios for the same tasks, suggesting that THE COLOSSEUM evaluations in simulation give reliable insight into real world generalization at a fraction of the setup cost. THE COLOSSEUM challenge and leaderboard (Figure 2) will provide as a unified platform to develop, evaluate, and compare future robotic manipulation methods that stand the test of robustness and generalization.

II. RELATED WORK

Prior works have made contributions towards benchmarking robot manipulation, developing robust models, and demonstrating generalization. THE COLOSSEUM builds on these efforts to create a systematic evaluation of multiple forms of test time generalization a trained policy may face.

A. *Robotic Manipulation Benchmarks*

Benchmarks in computer vision have significantly advanced the development of more generalized vision systems in recent decades by introducing numerous challenges and leaderboards [14, 34, 36], subsequently scaling into extensive foundational vision models [7, 3, 37]. Similarly, robotics datasets have demonstrated considerable diversity across various dimensions [12, 16, 18, 33, 68], particularly with the evolution of imitation learning and, more specifically, behavior cloning (BC). This progress has led to a proliferation of datasets and benchmarks aimed at assessing BC model task performance. Furthermore, most of these robotic benchmarks focus on evaluating model's capability to adapt to new tasks by changing the nature of the task [31, 73], it's functionalities[25, 38], or even environment [13, 51]. However, a gap remains in systematically evaluating and comparing different BC models on a large-scale, both in simulation and in real-world. Additionally, many of the benchmarks and datasets with perturbations have been specifically defined or curated as part of various BC or reinforcement learning works [72, 24, 68]. However, they do not provide an unified framework to evaluate all potential perturbations for generalization, primarily because they are not the main focus of these works.

Factor World [65] and KitchenShift [66] are similar efforts to THE COLOSSEUM. However, Factor World encompasses only 11 variation factors across 19 tasks, whereas KitchenShift contains 7 variation factors across 3 tasks. In contrast, The

Benchmark	Simulator	No. of perturbations	No. of tasks	Physical perturbation	Real-world reproducibility	No. of SoTA models
GROOT	LIBERO	3	3	×	✓	4
VLMBench	RLBench	8	8	×	×	3
KitchenShift	Isaac Sim	7	3	×	×	5
FactorWorld	MuJoCo	11	19	×	×	2
THE COLOSSEUM (ours)	RLBench	14*	20	✓	✓	5

TABLE I: **Generalization benchmarks comparison.** THE COLOSSEUM is the largest and most diverse benchmark for evaluating generalization in robotic manipulation policies, covering a wide range of variations and tasks. It is also the first to incorporate physical property perturbations. Similar to GROOT [72], we offer comprehensive instructions and 3D printed components for replicating real-world experiments. Additionally, we evaluated a variety of robot manipulation policies. One asterisk (*) indicates that, unlike the other benchmarks, the object's position is considered a default perturbation and was not counted as an additional perturbation.

COLOSSEUM boasts 14 factors of variation over 20 tasks. Beyond that, THE COLOSSEUM supports both 2D and 3D models, unlike Factor World which only evaluates on 2D visual-motor policies. Furthermore, we employed 3D printed objects to test with a Franka Panda robot arm to enable easy replication of our real-world experiments.

B. Robotic Manipulation Methods

There are myriad approaches that model robotic manipulation in simulation and real world in various different ways. Vanilla RL or BC [45, 42, 57, 49, 70] have been the popular choice since a long time, where a Multi-layer Perceptron (MLP) [50] or a Recurrent Neural Network (RNN) [56] type models use either low-dimensional object poses [62] or images [10] as the state input and predict continuous actions for the robot's controller in end-effector or joint space [58]. An emerging area of work takes the path of representation learning, either by pretraining a model with external knowledge [43, 53] or using pretrained representations from vision or language domains [59, 27, 4, 5]. Recently, training generalist models on large-scale real robot datasets, collected across several robotics research labs [48], to obtain a diverse set of skills in diverse environments and robots, have shown promising results in effectively scaling robust robot policies. Another line of work uses diffusion architecture to learn to generate robot trajectories in state or action space using the denoising process [2, 10, 8]. Some works have proposed distilling neural feature field representations for downstream BC [69] or RL [15]. Gervet et al. [20] learn with 3D feature field created from 2D pretrained image features and adaptive scene resolution to compute 3D action maps of high spatial resolution. Wang et al. [64] and Mandlekar et al. [39] propose scalable learning by watching humans or automatically generating large-scale datasets from a few human demonstrations. Another work uses large-scale TAMP generated data in simulation to learn a scalable multitask policy [11]. Recent robotic manipulation works have also proposed learning keypoint action prediction [32, 60, 61, 21] or action chucking [71] instead of predicting continuous control actions. We select a few recent SOTA methods, that have been applied to both simulation and real world, to evaluate on THE COLOSSEUM

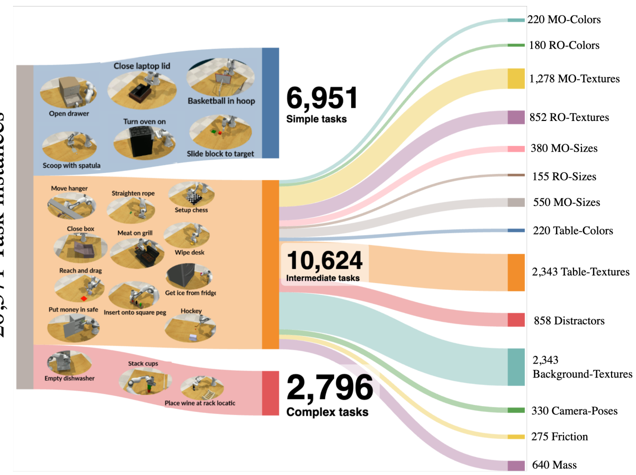

Fig. 3: **THE COLOSSEUM benchmark distribution.** This benchmark encompasses 14 `perturbation_factors` within 20 distinct RLBench tasks, categorized into three tiers (simple, intermediate, and complex) according to the number of way-points involved (task horizon). Collectively, THE COLOSSEUM presents 20,371 unique task perturbations instances.

including 2D and 3D learning methods, that operate with keypoint action prediction.

C. Generalization in Robotic Manipulation

Traditionally, enhancing generalization in imitation learning involves employing image or 3D data augmentation techniques, akin to those used in computer vision. These techniques encompass random shifts, color adjustments, and rotations [60, 35, 23, 67]. Sim-to-real transfer methods utilize advanced simulation environments for domain randomization and preliminary policy training before real-world application [46, 26, 41]. Recent developments in extensive vision and language models present a novel path to generalization: pretrained on vast image and language datasets, these models offer potentially more robust representations for robotic manipulation or can even directly inform actions [1, 27, 9, 5].

III. THE COLOSSEUM

THE COLOSSEUM is a comprehensive simulation benchmark, built by extending RLBench, consisting of 20 diverse robotic manipulation tasks, each enabled with 14 `perturbation_factors`.We base off of RLBench as it provides a variety of realistically useful tasks, with a scripted demonstration generation framework, broad variance in their task horizon (i.e. the number of controller steps required to complete the task) and primitive actions (such as pick, place, open, close, turn, and slide). We define `perturbation_factors` as scene properties, such as object color or lighting conditions. These properties can be changed to cause data distribution shifts at test-time such that the input distribution changes $p(x_{test}) \neq p(x_{train})$, but the conditional probability of action distribution remains the same $p(y_{test}|x_{test}) = p(y_{train}|x_{train})$, as the underlying task does not change. This form of distribution shift in Out-of-Distribution (OoD) generalization research is referred to as covariate shift [28].

With THE COLOSSEUM, we proposed over 20,371 unique task instances from a list of 20 tasks. The tasks are also categorized into three tiers of difficulties based on the task horizon, which inherently makes the tasks harder due to compounding error in BC [55]. The detailed breakdown of the number of unique instance per variation is also shown in Figure 3.

We describe our task selection strategy, `perturbation_factors` category and implementation in the following subsections. Thereafter, we describe the extension of THE COLOSSEUM in the real-world for 4 tasks replicated from the simulation. We finally propose the THE COLOSSEUM Challenge and explain the training and evaluation procedure expected for leaderboard participation compliance.

A. Methodology for Task Selection

We curate the task list for THE COLOSSEUM from the default suite of 100 tasks in RLBench. This selection ensures the feasibility of generating waypoints to facilitate task execution after incorporating our `perturbation_factors`. Our methodology involved discerning overlays within certain tasks, leading to the inclusion of tasks that require a versatile spectrum of primitive actions. This spectrum encompasses tasks ranging from straightforward ones, requiring fewer than 100 steps (e.g., `open drawer`), to more intricate challenges like `empty dishwasher`, which may exceed 1000 steps. Figure 3 shows the complete list of tasks classified into zones of complexity based on the horizon of the tasks.

B. Perturbation Factors

We create 14 `perturbation_factors` and apply each of them to the above 20 tasks where compatible. We categorize them as follows:

a) Manipulation object (`MO`) perturbation: `MO` is a task-relevant object that is directly manipulated or interacted with by the robot. For instance, in `put wine in rack` task, the 'wine bottle' is the manipulation object. `MO` variations include `MO_Color`, `MO_Texture`, `MO_Size`.

b) Receiver object (`RO`) perturbation: `RO` is a task-relevant object that is not directly interacted with by the robot, for example, the 'rack' in `put wine in rack` task. `RO` variations include `RO_Color`, `RO_Texture`, `RO_Size`.

c) Background perturbation: Factors that do not relate to task-relevant objects, but are background characteristic of the scene. These variations include `Light_Color`, `Table_Color`, `Table_Texture`, `Distractor` objects, `Background_Texture` of the walls, and `Camera_Pose`.

d) Physical perturbation: Factors that affect physical properties of the objects involved in the task, such as, `Object_Friction` where the task involves sliding of an object, and `Object_Mass` where the gripper needs to adapt to force required for moving the object.

THE COLOSSEUM supports applying one or more `perturbation_factors` defined above in the same scene and study its effect at test-time.

C. Implementation of Perturbation Factors

Following RLBench, our implementation utilizes `PyRep` [30], a low-level API, to interact with the underlying `CoppeliaSim` [54] simulator. The `PyRep` API allows the control of simulator properties such as color, texture, scaling, and pose of the objects in the scene. We implement 14 `perturbation_factors` (shown in Figure 2: part 3) as an extension of the RLBench task benchmark. We expose the configuration of supported perturbations via configuration files written in `YAML`. Our implementation is easily extensible for other researchers to build on and edit the benchmark, for adding new `perturbation_factors` or new tasks, along with the ease of configuring any combination of `perturbation_factors` and their parameters, where compatible.

To implement texture or color perturbations, we randomly sample a texture or color from our curated set. We provide a set of 213 textures and 20 colors. These assets were used for implementing `MO_Color`, `MO_Texture`, `RO_Color`, `RO_Texture`, `Table_Color`, `Table_Texture`, and `Background_Texture`. To implement size perturbations (`MO_Size` and `RO_Size`), we sample a scaling factor from a continuous range, specified for each object that supports this factor. For instance, the range for `MO_Size` in the task `basketball_in_hoop` is $[0.75, 1.25]$, differs from that in task `hockey` $[0.95, 1.05]$, as this parameter is quite dependent on the conditions of the objects in the scene. We include the remaining task parameters in the Appendix. The waypoints get re-scaled with the object, and if not, we reposition them proportionally with respect to the object center, while ensuring that RLBench's scripted demonstration generation from waypoints remains functional. The

`Distractor` objects are sampled from a set of 78 object models taken from the YCB Object Dataset [6] and converted to `CoppeliaSim` compatible `.ttm` format. We utilize predefined object spawn boundaries to place these objects on table-top or on another object. To modify `Light Color`, we randomly sample RGB values from our specified range of $[0.0, 0.0, 0.0]$ to $[0.5, 0.5, 0.5]$, and apply it to all 3 directional lights surrounding the scene. We perturb `Camera Pose` for 3 cameras — front, left shoulder, and right shoulder — by changing their positions and orientations in Euler angles, sampled from ranges $[-0.1, -0.1, -0.1]$ to $[0.1, 0.1, 0.1]$ and $[-0.05, -0.05, -0.05]$ to $[0.05, 0.05, 0.05]$ respectively. `Object_Friction` is implemented by changing the friction coefficient of the object with a value sampled from the range $[0.75, 1.0]$. `Object_Mass` changes the mass of objects with a value sampled from a given range, where the ranges are task dependent (provided in Appendix). We provide our texture, color, and object model assets with the benchmark code, which can also be augmented easily for additional assets, as required.

Some `MO` and `RO` perturbations do not apply to all the tasks, due to two main reasons. `RO` perturbations do not apply to tasks when there is no `RO` object, for instance, `open drawer` task. Additionally, `PyRep` doesn't support application of surface texture or scaling for objects made up of compound shapes, such as the 'dishwasher' in `empty dishwasher` task.

D. Real-World Tasks and Perturbations

For the real-world extension of THE COLOSSEUM, we implement `perturbation_factors` akin to those in simulation. We create real-world mirrors for 4 RLBench tasks: `insert onto square peg`, `slide block to target`, `scoop with spatula`, and `setup chess`. To ensure replicability of our real-world benchmark extension, we created these four tasks with 3D-printed objects, identical to those in the RLBench tasks, with the various variant factors to support the perturbations. We open-source our 3D-printed object models, which are inexpensive to print, to facilitate reproduction of our real-world tasks and their `perturbation_factors`.

Our real-world experiments utilized a Franka Panda robot arm, replicating the setup in RLBench, for both collecting training data and evaluation on the `perturbation_factors`. To create size perturbation (`MO_Size` and `RO_Size`), we 3D printed identical manipulation objects in 2 additional sizes. For object color and texture perturbation (`MO_Color`, `RO_Color`, `MO_Texture` and `RO_Texture`) we 3D printed the objects in two alternate colors and textures. For `Table_Color`, `Table_Texture`, and `Background_Texture`, we use two different sets of table mats or wallpapers to mirror these `perturbation_factors` in the real-world. For `Camera_Pose`, we re-calibrate the front camera at two different spots that differs from the camera pose set during training data collection. Lastly, the `Light_Color` was simulated with a dynamically color-changing spotlight. Finally, to introduce `Distractor` objects, we incorporated additional random tabletop objects into the scene. We present our real-world setup in Figure 4.

E. THE COLOSSEUM *Challenge*

We propose THE COLOSSEUM Challenge to enable development of generalizable Behavior Cloning (BC) models for robotic manipulation tasks. As shown in Figure 2, the challenge involves four key phases: 1) Participants generate a standard training dataset for 20 tasks with 100 demonstrations each, without `perturbation_factors`. 2) Participants train their BC models using this standardized dataset. 3) The models should evaluate over a fixed 25 episodes set of each of the 14 `perturbation_factors`. 4) Models are ranked on a leaderboard based on the percentage change in their performance across these `perturbation_factors`. Demonstrations data for training and testing can be generated via scripted experts from RLBench. Each task is equipped to instantiate object pose-variated episodes ensuring an inexhaustible supply of task-specific demonstrations. Demonstrations are collected automatically through motion planners that navigate through manually defined waypoints.

As we show in our results, evaluation in the simulated benchmark aligns well with that on our reproducible real-world mirror. Therefore, participants can expect similar generalization in real-world when participating in THE COLOSSEUM Challenge in simulation. Participants can reproduce and evaluate on the real-world part of benchmark, however, submitting real-world results to the leaderboard will remain optional.

IV. EXPERIMENTS

In this section we define our problem formulation for baseline training, followed by describing the baselines methods and their respective training logistics. Thereafter, we elucidate THE COLOSSEUM's standard training and evaluation protocol. Finally, we describe our real-world setup and its training details.

A. Dataset and Problem Formulation

The problem is to learn action prediction from robot's observation and the language instruction. The training dataset of demonstration consists of N trajectories, $\tau_i = \{(o_j, a_j, p_j), l\}_{j=1}^{T}$ where o is the observation and a is a continuous robot arm action. An action a_j is the 6-DoF gripper pose and it's open or close state, an observation o_j is a set RGBD images from a given number of cameras, and robot arm's proprioception p_j is the arm's current pose, at time step j. Each trajectory is paired with a template-generated English language instruction l.

Following recent SotA [29, 32, 60, 21], we use keypoint-based action prediction instead of predicting continuous 7-DoF actions. The keypoint actions are discovered using intuitive heuristics, such as instances where the arm's joint velocities are close to zero, and whether the gripper's open state has changed.

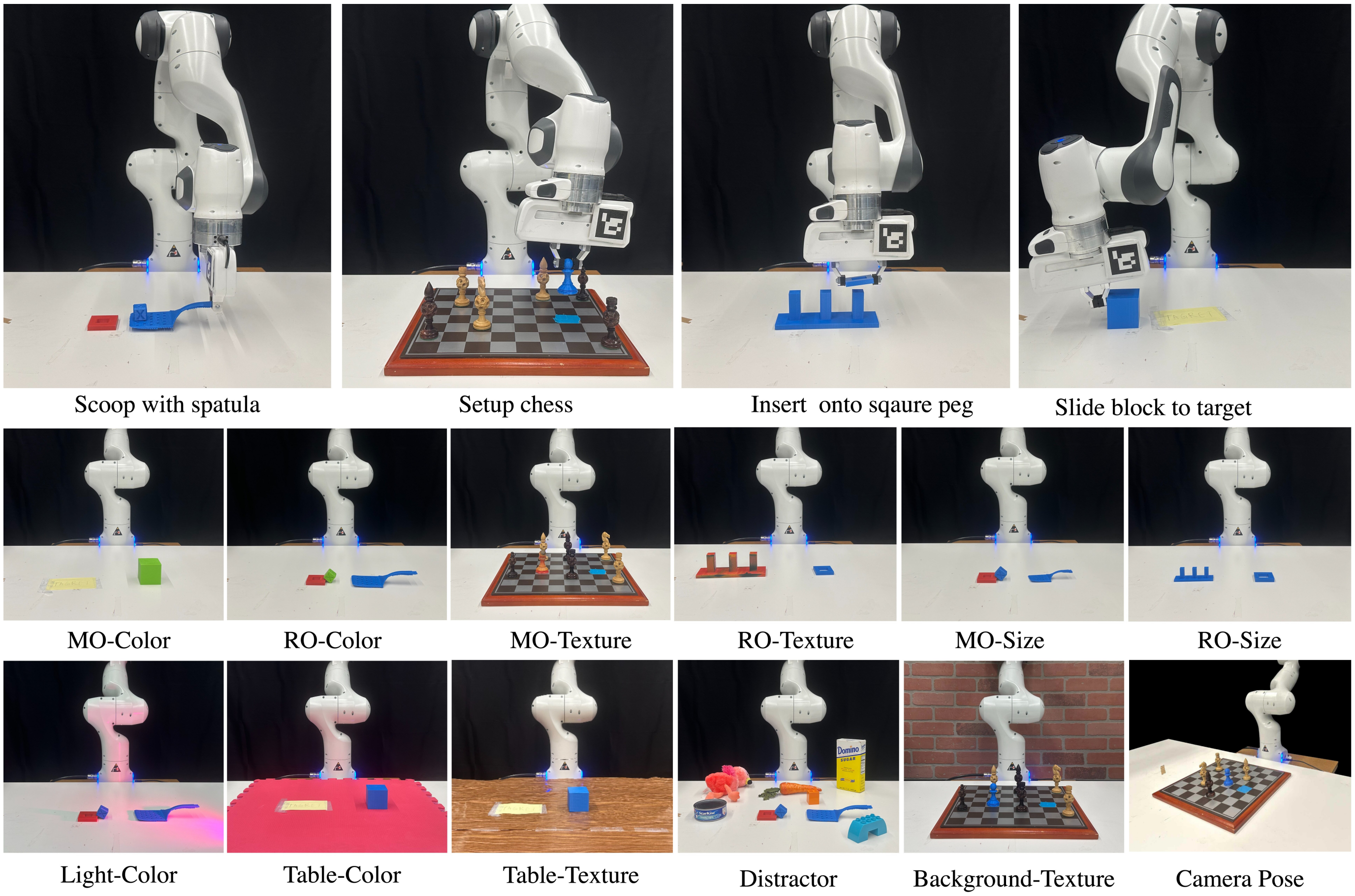

Fig. 4: **Real-World training tasks and their evaluation time perturbations.** A PerAct agent, trained using real-world demonstrations for the four tasks shown, was tested on real-world `perturbation_factors`. This evaluation involved perturbing factors similar to the procedural benchmark in the simulation.

B. *Baselines*

We study 5 SotA baselines, including one zero-shot open vocabulary model (`VoxPoser`), two 2D learning models (`R3M-MLP`, `MVP-MLP`) and two 3D learning models (`PerAct`, `RVT`). We choose these methods as baselines as they establish themselves as strong robot learning methods. They are also diverse in their approach, allowing us to study the effect of aspects such as, pretraining and 2D vs 3D based learning. For all baselines, the language is encoded using a frozen CLIP [52] model.

1) 2D learning models: `R3M-MLP` and `MVP-MLP` use pretrained visual encoders, pretrained on out of domain task-agnostic real-world images. MVP [53], a ViT-Base model with 86M parameters, learns representations with 4.5M in-the-wild images on task-agnostic real world data using a self-supervised masked reconstruction objective. R3M [43], a ResNet-50 model with 23M parameters, learns representation using egocentric human videos with captions [22] via video-language contrastive and temporal loss objectives. Both representations have been shown to be effective for downstream task adaptation using RL or BC, via an MLP action prediction head, both in simulated and real world settings. We adapt these pretrained encoders similarly by freezing the encoders and adding an MLP prediction head with $\sim$3M trainable parameters. We train both models with batch size 32 for 300k training iterations. The input to the model is 4 camera RGB views encoded by their respective pretrained train visual encoder (no depth, as per prior work's use case), encoded language instruction, and proprioception. Following the prior work [43, 53], we predict raw 7-DoF keypoint pose of the robot arm in continuous space.

2) 3D learning models: `PerAct` is a transformer-based robotic manipulation BC model that takes tokenized voxel grid and language instruction as the input, to predict discretized voxel grid translation point, discrete rotation in Euler angles, and gripper's binary (open/close) state. `PerAct` works with 3D voxel grid tokens, akin to visual patch tokens or language tokens in vision or language transformers. Following the original implementation, we use a voxel grid of size 100^3, corresponding to an actual volume of 1.0m^3. The patch tokens of size 5^3 are encoded via a 3D convolution layer with kernel-size and stride of 5, resulting in $20^3 = 8000$ voxel observation tokens. Actions are discretized via voxelized keypoint-based action prediction. The actions are then predicted as the next-best voxel that is closest to the center of the gripper fingers for the next translation pose. Rotation pose is discretized into bins of 5^o increments. The input to the model is the encoded language instruction, proprioception, and 4 camera RGBD views, which gets preprocessed into a voxel grid with voxel occupancy and RGB channels. The model has ∼33M trainable parameters. We train this model with batch size 16 for 300k training iterations.

`RVT` is a multi-view transformer-based robotic manipulation BC model that uses tokenized image patches and CLIP-encoded language instruction tokens as input to predict keypoint actions as translation heatmaps, discretized rotation in Euler angles, and gripper's binary state. `RVT` re-renders the captured RGBD views from new virtual camera views via constructing a 3D point cloud. This procedure decouples the camera images from the images fed to the model, as well as allows generating more viewpoints unrestricted by real-world constraints. The transformer attends over language instruction, re-rendered views, and robot's proprioception to predict actions. The model predicts heatmaps for each input view, which is then back-projected to a discretized set of 3D points densely populating the robot's workspace, out of which the point with the highest score is chosen as the next translation point. Rotation and gripper state prediction remain the same as `PerAct`. We train `RVT`, with ∼36M trainable parameters, on our 20 tasks with batch size 24 for 100k iterations, following its original configuration.

C. Zero-Shot manipulation model using Large Pretrained World Models

`VoxPoser`[27] is a formulation that aims to extract affordances and constraints using LLMs. Through code, it composes 3D value maps in observation space to guide robotic interactions. We utilized their RLBench implementation for VoxPoser, providing variation descriptions from each demonstration as input language text. We manually annotated all corresponding RLBench[31] objects with their respective object names mentioned in the variation descriptions. We conducted a zero-shot evaluation of `VoxPoser` using THE COLOSSEUM's evaluation protocol, without any training involved.

D. Training and Evaluation Protocol

We apply THE COLOSSEUM training and evaluation to each of the above models. We train with 100 demos per task without any of THE COLOSSEUM perturbations applied. However, we do apply the default RLBench task variations in the training, i.e., changing language instruction and task target — for instance, `open drawer`'s RLBench variations include `open bottom drawer, open middle drawer, open top drawer` — to maintain the original baseline training settings.

For consistent evaluation, we generate training and test data once and use the last checkpoint for each of the above trained baselines, and evaluate on each of the `perturbation_factors`. We fix each task to the default RLBench task variations (for instance, `open bottom drawer` in the above example) in order to closely evaluate the effect of our applied perturbations. However, we do not fix the object pose variations across our test episodes. We refer to the default RLBench variation without any `perturbation_factors` applied as `No Perturbation` test set. In addition, we also analyze all perturbations activated together (`All Perturbations`). That makes THE COLOSSEUM test sets 235-strong, with 25 episodes per test set. A test episode is successful if the model completes the task fully. We report the average success rate for each test set, further averaged across tasks, referred to as task-averaged success rate hereon. We include per task performance in the Appendix. We report results with one training seed and one evaluation seed over the benchmark per baseline model.

E. Real-World Setup

In our real-robot experiments, we employed a Franka Panda manipulator equipped with a parallel gripper for data collection and evaluation. For perception, a front-facing Kinect-2 RGB-D camera was utilized. We collected real-world demonstrations using an HTC Vive controller, gathering 5 demonstrations for each of the 4 tasks. To facilitate comparative analysis, we trained a `PerAct` model [60] with ∼33M parameters on these real-world demonstrations for 200k iterations with a batch size of 1. In a similar vein, we trained another multi-task instance of `PerAct` in the simulated environment, focusing on the same four tasks. This simulation model, also comprising ∼33M parameters, was trained over 50k iterations with a batch size of 4. For consistency in evaluation between simulation and the real-world, we evaluated both models on all `perturbation_factors` and `No Perturbation` test sets, each with 10 episodes, for 3 separate runs.

V. RESULTS

We report our results as task-averaged success rate of different baselines on THE COLOSSEUM and draw insights based on which `perturbation_factors` affect which kind of baseline. We also perform an upper bound training ablation when training and testing with `All Perturbations` enabled. Lastly, we report our simulation and real-world benchmark alignment analysis based on model success rates on the `perturbation_factors`.

A. Performance of different baselines on THE COLOSSEUM

We report absolute task-averaged success rates in Figure 1 for all baselines and `perturbation_factors`. All polar axes are comparable, except `MO_Texture`, `MO_size`, `RO_Color`, `RO_Texture`, and `RO_Size` axes that have different task averages, since they don't apply to all 20 tasks (more details reported in the Appendix). We also report the above results as percentage change with respect to `No Perturbation`, averaged across 20 tasks, along with qualitative failure cases associated with each of the `perturbation_factors` in Figure 5. For factors that were not applicable or infeasible in simulation on some tasks, we compare their averages only with corresponding task's `No`

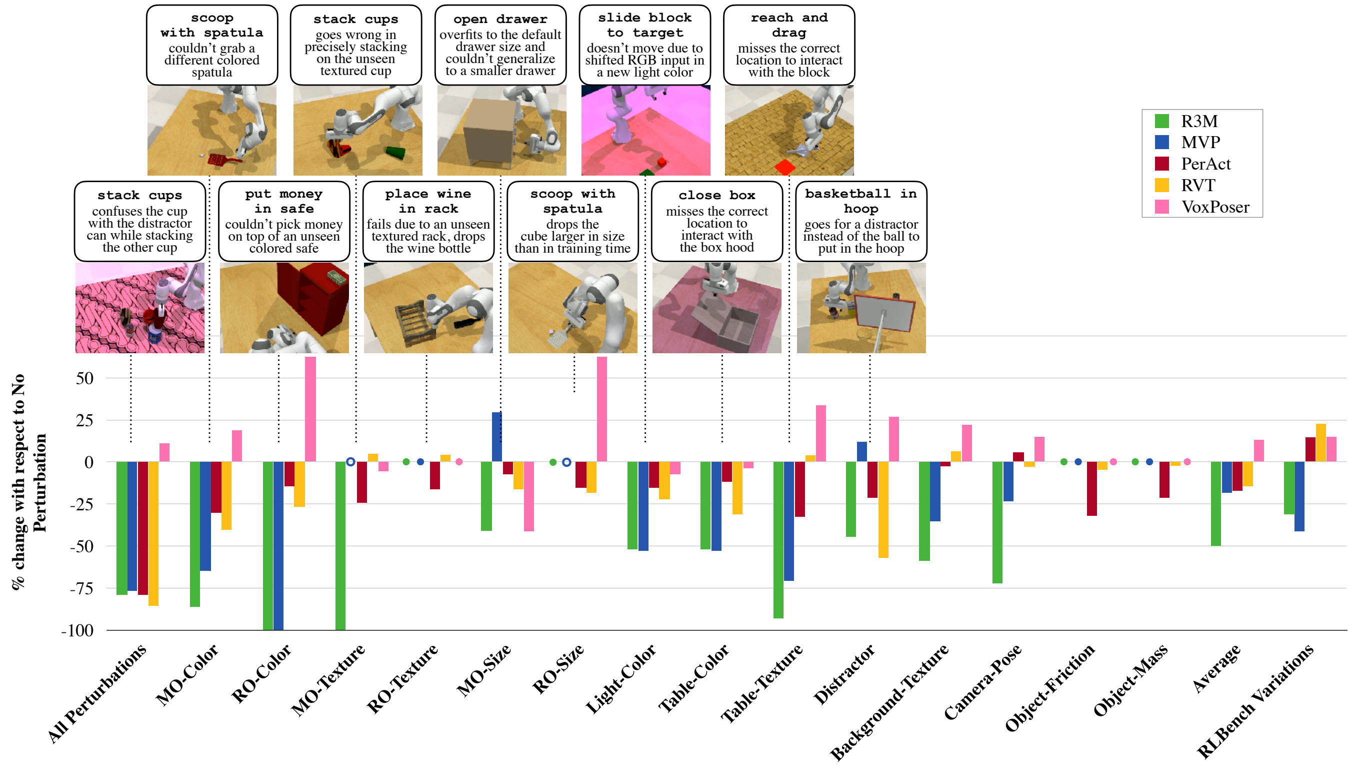

Fig. 5: **Task-averaged success rate % change for 4 baseline models on `perturbation_factors`, compared to `No Perturbation` test set**. We report the evaluation with `All Perturbations` enabled, followed by each individual factor, average of all individual factors, and on RLBench variations (that is sampled from the same distribution as the training set). The images on top show failure examples for each factor with captions explaining the failure. • indicates undefined value when the corresponding `No Perturbation` task averages are also 0. ∘ indicates 0% change with respect to `No Perturbation` task average.

`Perturbation` case on which that perturbation was applied and evaluated. Applying `All Perturbations` in the same scene influences all the models significantly, leading to ≥75% decrease in performance. What `perturbation_factors` are the most affecting?

For 2D-models (`R3M-MLP` and `MVP-MLP`), we observe that object and light color, texture, and camera pose are the most affecting factors. Since these models are trained end-to-end with RGB inputs, and the color or texture related perturbations shift the input space, thereby affecting the output space as well. Moreover, training with specific `Camera_Poses` when using RGB as input also affects the performance when camera poses are perturbed. We observe that `MVP-MLP` does better or is not affected in presence of `Distractors`, which may be due to the real world pretraining on cluttered scenes. This result indicates value in pretraining on real-world data.

For zero-shot manipulation models using Large Pretrained World Models, we observed that the system demonstrates robust generalization capabilities across various conditions, particularly excelling in tasks where it is predisposed to succeed. Specifically, for the two tasks in which `VoxPoser` excels, it maintains consistent performance across all variants. For example, in the task `slide_block_to_target`, the performance difference between the `No Perturbation` scenario and the average of all perturbations is a mere 3.21% relative to the `No Perturbation` performance. This aligns with our expectation that leveraging large pretrained world models enables the recognition of significant changes in environments and object perturbations.

For 3D-models (`RVT` and `PerAct`), we observe that the most affecting factors are color-related including object, table and light colors as well as presence of `Distractors`, while other factors cause a smaller performance decrease. Since `RVT` and `PerAct` are both trained end-to-end with RGB images or voxel grid with RGB channels, the color perturbations remain challenging for these models as well. These models lack any real-world pretraining, thus, the presence of `Distractors` puts the scene out of distribution, significantly affecting their performance. **We observe that these model are robust to changes in `Camera_Pose`, because they do not directly learn on captured view.** They instead preprocess the input RGBD views into a voxel grid or re-rendered novel views. For these models, while each factor doesn't lead to a very significant effect on their performance, all factors combined in one scene (`All Perturbations`)

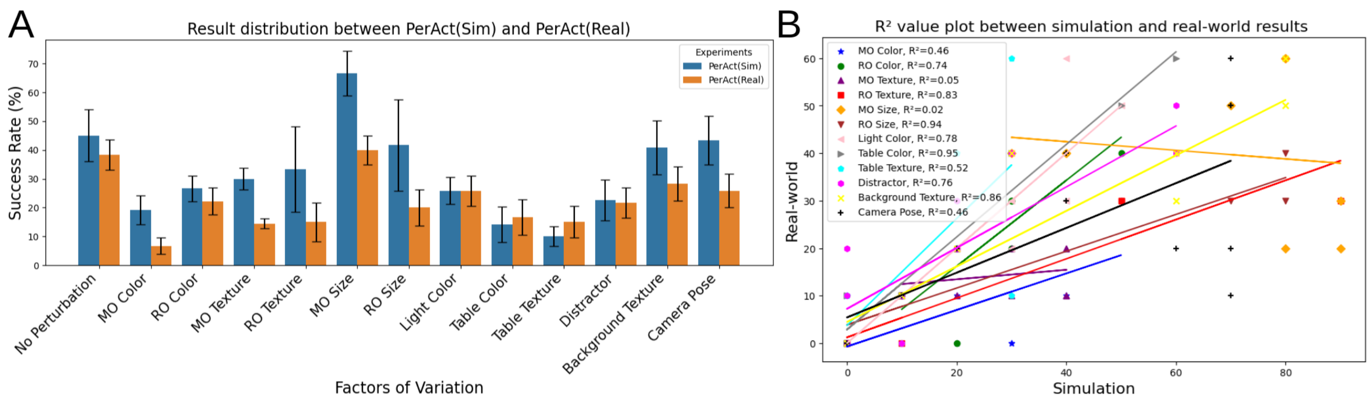

Fig. 6: **Real robot results and alignment analysis on `perturbation_factors` across 4 tasks.** A) The plot illustrates the empirical performance of two models, one trained using real-world data and the other with simulated data, each tested across 10 episodes and 3 runs in their respective environments. Additionally, it displays a uniform distribution of standard deviation between the models, highlighting which `perturbation_factors` align more strongly with empirical success rates. B) To examine the correlations between simulation and real-world results for each perturbation independently, we have plotted a scatter chart. This chart includes data points from one run for each task. We calculated the R^2 value for each task and illustrated their respective best-fit lines on the chart.

cause a significant decrease. On physical perturbations `RVT` performs better than `PerAct`, perhaps because modelling in `RVT` is more robust for keypoint prediction than `PerAct` under these perturbations. Physical perturbation results for other models are inconlcusive as they cannot perform the tasks that support these perturbations.

We observe that 3D baselines are better performing generally (Figure 1), and on average much more robust to environment perturbations as compared to 2D baselines (Figure 5). We also observe that `RVT`, trained only with RGB views, generally gets more affected with `perturbation_factors` as compared to `PerAct`, trained with complete 3D scene, notably in the case of `Distractors`. This result indicates value in learning with 3D scenes as input, for the resultant model is more robust to such environmental perturbations, as it might be learning 3D features of the objects instead of just their 2.5-dimensional projections.

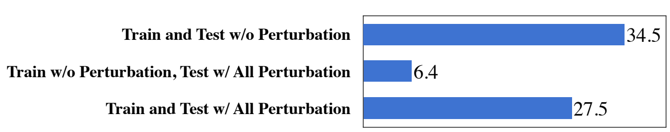

Fig. 7: **Task-averaged success rate ablation for `All Perturbations`** when training and testing with or without all perturbations enabled.

B. Training on `All Perturbations` ablation

We report results on training `PerAct` with 100 demos (with batch size 16 for 300k iterations) in `All Perturbations` setting in Figure 7. Zero-shot evaluation of a `PerAct` model trained on RLBench variations data with no `perturbation_factors` enabled achieves task-averaged success rate of 6.4% (28.1% lower than `No Perturbations` task-averaged success rate). When we train the model with with `All Perturbations` enabled, the task-averaged success rate increases by 21.1% only. However, the model should be able to perform the same tasks under any environmental setting for being practically deployable. This result indicates that THE COLOSSEUM's `perturbation_factors` not only study systematic perturbations added to the environment, but also increase the difficulty of the tasks itself, even with ground truth perturbed scenes available for training. From this ablation with `All Perturbations` as proxy for an extreme case of factor compounding, which is results in 28.1% lower in success rates with respect to `No Perturbations`, hence suggesting that compounding perturbation factors has some degree of compounding effect toward models' performance.

C. Real-world alignment analysis for THE COLOSSEUM

We first observed only a marginal difference in success rate of 6.67% during evaluation on the `No Perturbations` tasks between `PerAct` trained in simulation and that in real-world settings. This served as a crucial sanity check for tasks performance between the two models before advancing forward to evaluating them across the 14 `perturbation_factors`. We observed that for factors such as `MO_Texture`, `Light_Color`, `Table_Color`, `Table_Texture`, and `Distractor`, the discrepancies in performance between both models on each individual factor were marginal, remaining under 5%.

The observed variances in other factors may stem from differences in waypoint annotations, physical robot interactions, and training data's visual distinctions. To further investigate the correlation between simulation and reality, we used the success rate performance of each individual

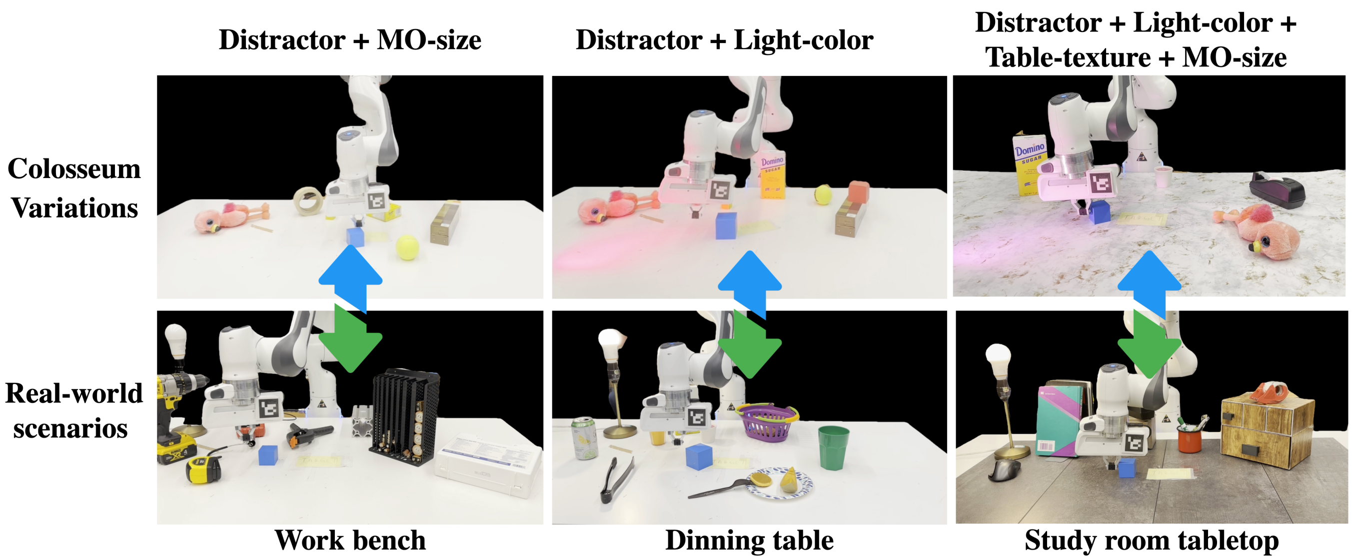

Fig. 8: **Example rollouts using THE COLOSSEUM perturbations in tandem with comparable real-world scenarios.** We combine various THE COLOSSEUM variations to form three combinations of compounding perturbation scenarios, each paired with a curated real-world scene (*workbench, dining table, study room table*) featuring the same set of perturbations that are naturally present in the scene, similar to the approach used by [9]. We evaluate these combinations using `PerAct` model trained on `No Perturbation`, and aim to establish a correlation between our simplified compounding perturbation scenarios and realistic real-world scenes.

run for each task as a data point to calculate the coefficient of determination, or R-square values. Our results indicated that for factors like `MO_Color`, `Table_Texture`, and `Camera_Pose`, there was a moderate level of correlation ($0.46 \leq R^2 \leq 0.52$). Conversely, factors such as `Background_Texture`, `Distractor`, `Table_Color`, `Light_Color`, `RO_Color`, `RO_Texture`, and `RO_Size` has a R^2 value between 0.74 and 0.94 with `Table_Color` being the most significant as illustrated in Fig 6B. These results suggest that for at least 7 out of 14 `perturbation_factors` there is a strong correlation between the performances of the two models, **thereby indicating a clear alignment between evaluation done on THE COLOSSEUM in simulation and in the real-world**.

Based on the results presented in Figure 6A, we also observed that for real-world experiments, `MO_Color` exhibited a substantial decline in task success, with an 82.6% drop, whereas `MO_Size` demonstrated no performance reduction, instead enhancing performance by 4.34%. Further examination of individual episodes in the real-world experiments, we observed that perturbations in `Light_Color` could significantly alter an object's visual appearance by casting differently colored light, consequently impacting the BC model's success rate. Additionally, the `MO_Color` perturbation frequently impeded the robot's ability to accurately predict the 6D pose for grasping the manipulation object. This finding is consistent with results from simulation and underscores a critical aspect: BC models like `PerAct`, which construct their 3D encoders from the ground up without leveraging pretrained 3D features, struggle to generalize across a wide range of object visuals. This limitation highlights the challenge in developing robust BC models capable of adapting to diverse visual environments.

D. THE COLOSSEUM *perturbations grounded in the real-world*

To validate that the perturbations in THE COLOSSEUM accurately mimic naturally occurring environmental or object variations in real-world scenarios, we carried out an ablation study. This study compared three realistic scenes—workbench, dining table, and study room tabletop—with corresponding perturbation combinations derived from THE COLOSSEUM as shown in Figure 8. Utilizing a multitask instance of `PerAct` trained with `No Perturbation`. We evaluated all the pairs of scenarios for the task of `slide_block_to_target`. Over five trials with ten episodes each, we observed a significant correlation. Notably, the combination of [`Distractor` + `MO_Size`] resulted in an $R^2 = 0.75$, while the combination of [`Light_Color` + `Table_Texture` + `Distractor` + `MO_Size`] achieved an $R^2 = 0.83$. For further details on the results and methodology, please refer to Supplementary Section IV.E.

VI. LIMITATIONS AND FUTURE WORK

Currently, our leaderboard baselines only include 4 methods, which are all BC methods. In future, we plan to include RL-based methods [15]. In addition, we also plan to include several other baseline methods, such as, those based on diffusion [10], 3D feature feature fields [20, 69], large-scale robotics pretraining [47], action tokenization [5], and action chucking [71]. Expanding The Colosseum with these methods will unify comparing effectiveness of robot learning methods in a single leaderboard, while also providing a good starter framework for researcher to develop new methods along or beyond included baselines.

In our real-world experiments, a key limitation lies in precisely replicating the pose, orientation, and execution of tasks both in the collected training data and during evaluation. Additionally, due to resource constraints, each perturbed factor in the real-world setup was limited to only two alternate variations. As a result, the real-world findings primarily represent a comparative performance distribution between the simulation and real-world scenarios. Looking ahead, we aim to expand the number of real-world tasks, ensuring they closely mirror their counterparts in simulation. This expansion is intended to enhance the reproducibility of simulated tasks, thereby broadening benchmarking scope.

VII. Conclusion

We introduced The Colosseum, a comprehensive benchmark designed to assess the generalization capabilities of Behavior Cloning (BC) models in robotic manipulation. The Colosseum systematically perturbs the task environments of the robot's workspace along an exhaustive list of axes — including object appearance and size, lighting, physical properties of objects, background, table-top appearance, and camera pose — both in simulation and real-world. Through empirical studies conducted with SotA BC methods on The Colosseum, we identified which perturbation factors most significantly impact model's success rates on tasks they are trained to execute. Additionally, we demonstrated a close alignment between The Colosseum in simulated and real-world. To enhance reproducibility and facilitate future model evaluations in both simulated and real-world, we will open-source our resources along with the 3D printed assets. The Colosseum offers a platform for future research to develop and quantitatively evaluate robotic manipulation models before scaling via a unified leaderboard.

VIII. Acknowledgement

Jiafei Duan is supported by the Agency for Science, Technology and Research (A*STAR) National Science Fellowship. Wilbert Pumacay is supported by grant 234-2015-FONDECYT from Cienciactiva of the National Council for Science, Technology and Technological Innovation (CONCYTEC-PERU). We would also like to thank Yi Li and the members of the UW RSE Lab for their feedback on the paper.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. URL https://say-can.github.io/.

[2] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B. Tenenbaum, Tommi S. Jaakkola, and Pulkit Agrawal. Is Conditional Generative Modeling all you need for Decision Making? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sP1fo2K9DFG.

[3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. URL https://arxiv.org/pdf/2308.12966.pdf.

[4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil 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 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. In *arXiv preprint arXiv:2212.06817*, 2022. URL https://arxiv.org/abs/2212.06817.

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. URL https://arxiv.org/abs/2307.15818.

[6] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. The YCB object and Model set: Towards common benchmarks for manipulation research. In *2015 International Conference on Advanced Robotics (ICAR)*, pages 510–517, 2015. doi: 10.1109/ICAR.2015.7251504. URL https://ieeexplore.ieee.org/document/7251504.

[7] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023. URL https://arxiv.org/abs/2310.09478.

[8] Lili Chen, Shikhar Bahl, and Deepak Pathak. PlayFusion: Skill Acquisition via Diffusion from Language-Annotated Play. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 2012–2029. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/

chen23c.html.

[9] Zoey Chen, Sho Kiami, Abhishek Gupta, and Vikash Kumar. Genaug: Retargeting behaviors to unseen situations via generative augmentation. *arXiv preprint arXiv:2302.06671*, 2023.

[10] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023. URL https://arxiv.org/abs/2303.04137.

[11] Murtaza Dalal, Ajay Mandlekar, Caelan Garrett, Ankur Handa, Ruslan Salakhutdinov, and Dieter Fox. Imitating Task and Motion Planning with Visuomotor Transformers. 2023. URL https://arxiv.org/abs/2305.16309.

[12] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. RoboNet: Large-Scale Multi-Robot Learning. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 885–897. PMLR, 30 Oct–01 Nov 2020. URL https://proceedings.mlr.press/v100/dasari20a.html.

[13] 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. URL https://arxiv.org/abs/2206.06994.

[14] 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. URL https://ieeexplore.ieee.org/document/5206848.

[15] Danny Driess, Ingmar Schubert, Pete Florence, Yunzhu Li, and Marc Toussaint. Reinforcement Learning with Neural Radiance Fields. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2206.01634.

[16] Jiafei Duan, Samson Yu, Hui Li Tan, and Cheston Tan. Actionet: An interactive end-to-end platform for task-based data collection and augmentation in 3d environment. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1566–1570. IEEE, 2020.

[17] Jiafei Duan, Yi Ru Wang, Mohit Shridhar, Dieter Fox, and Ranjay Krishna. AR2-D2: Training a Robot Without a Robot. *arXiv preprint arXiv:2306.13818*, 2023. URL https://arxiv.org/abs/2306.13818.

[18] Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021. URL https://arxiv.org/abs/2109.13396.

[19] Kiana Ehsani, Tanmay Gupta, Rose Hendrix, Jordi Salvador, Luca Weihs, Kuo-Hao Zeng, Kunal Pratap Singh, Yejin Kim, Winson Han, Alvaro Herrasti, et al. Imitating Shortest Paths in Simulation Enables Effective Navigation and Manipulation in the Real World. *arXiv preprint arXiv:2312.02976*, 2023. URL https://arxiv.org/abs/2312.02976.

[20] Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3D: 3D Feature Field Transformers for Multi-Task Robotic Manipulation. In *Conference on Robot Learning*, pages 3949–3965. PMLR, 2023. URL https://arxiv.org/abs/2306.17817.

[21] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. *arXiv:2306.14896*, 2023.

[22] Kristen Grauman, Andrew Westbury, and et al. Ego4D: Around the World in 3,000 Hours of Egocentric Video. In *IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2022. URL https://ieeexplore.ieee.org/document/9879279.

[23] Nicklas Hansen and Xiaolong Wang. Generalization in reinforcement learning by soft data augmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13611–13617. IEEE, 2021. URL https://ieeexplore.ieee.org/document/9561103.

[24] Nicklas Hansen and Xiaolong Wang. Generalization in reinforcement learning by soft data augmentation. In *International Conference on Robotics and Automation*, 2021.

[25] Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. FurnitureBench: Reproducible Real-World Benchmark for Long-Horizon Complex Manipulation. In *Robotics: Science and Systems*, 2023. URL https://arxiv.org/abs/2305.12821.

[26] Daniel Ho, Kanishka Rao, Zhuo Xu, Eric Jang, Mohi Khansari, and Yunfei Bai. Retinagan: An object-aware approach to sim-to-real transfer. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10920–10926. IEEE, 2021. URL https://arxiv.org/abs/2011.03148.

[27] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. URL https://arxiv.org/abs/2307.05973.

[28] Dieuwke Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, et al. A taxonomy and review of generalization research in NLP. *Nature Machine Intelligence*, 5(10):1161–1174, 2023. URL https://www.nature.com/articles/s42256-023-00729-y.

[29] Stephen James and Andrew J Davison. Q-attention: Enabling efficient learning for vision-based robotic manipulation. *IEEE Robotics and Automation Letters*, 7 (2):1612–1619, 2022. URL https://ieeexplore.ieee.org/document/9878928.

[30] Stephen James, Marc Freese, and Andrew J. Davison. PyRep: Bringing V-REP to Deep Robot Learning. *arXiv preprint arXiv:1906.11176*, 2019. URL https://arxiv.org/abs/1906.11176.

[31] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. URL https://ieeexplore.ieee.org/document/9001253.

[32] Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13739–13748, 2022. URL https://arxiv.org/abs/2106.12534.

[33] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 651–673. PMLR, 29–31 Oct 2018. URL https://proceedings.mlr.press/v87/kalashnikov18a.html.

[34] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017. URL https://link.springer.com/article/10.1007/s11263-016-0981-7.

[35] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement Learning with Augmented Data. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19884–19895. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e615c82aba461681ade82da2da38004a-Paper.pdf.

[36] 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. URL https://link.springer.com/chapter/10.1007/978-3-319-10602-1_48.

[37] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. In *NeurIPS*, 2023. URL https://arxiv.org/abs/2304.08485.

[38] Jianlan Luo, Charles Xu, Fangchen Liu, Liam Tan, Zipeng Lin, Jeffrey Wu, Pieter Abbeel, and Sergey Levine. FMB: a Functional Manipulation Benchmark for Generalizable Robotic Learning. *arXiv preprint arXiv:2401.08553*, 2024. URL https://arxiv.org/abs/2401.08553.

[39] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. MimicGen: A Data Generation System for Scalable Robot Learning using Human Demonstrations. In *7th Annual Conference on Robot Learning*, 2023. URL https://proceedings.mlr.press/v229/mandlekar23a.html.

[40] Bhairav Mehta, Manfred Diaz, Florian Golemo, Christopher J. Pal, and Liam Paull. Active Domain Randomization. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 1162–1176. PMLR, 30 Oct–01 Nov 2020. URL https://proceedings.mlr.press/v100/mehta20a.html.

[41] Marius Memmel, Andrew Wagenmaker, Chuning Zhu, Dieter Fox, and Abhishek Gupta. Asid: Active exploration for system identification and reconstruction in robotic manipulation. In *The Twelfth International Conference on Learning Representations*, 2023.

[42] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. URL https://arxiv.org/abs/1312.5602.

[43] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A Universal Visual Representation for Robot Manipulation. In *6th Annual Conference on Robot Learning*, 2022. URL https://openreview.net/forum?id=tGbpgz6yOrI.

[44] Mitsuhiko Nakamoto, Yuexiang Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. *arXiv preprint arXiv:2303.05479*, 2023. URL https://arxiv.org/abs/2303.05479.

[45] Andrew Y Ng, Adam Coates, Mark Diel, Varun Ganapathi, Jamie Schulte, Ben Tse, Eric Berger, and Eric Liang. Autonomous inverted helicopter flight via reinforcement learning. In *Experimental robotics IX: The 9th international symposium on experimental robotics*, pages 363–372. Springer, 2006. URL https://link.springer.com/chapter/10.1007/11552246_35.

[46] Phuong DH Nguyen, Tobias Fischer, Hyung Jin Chang, Ugo Pattacini, Giorgio Metta, and Yiannis Demiris. Transferring visuomotor learning from simulation to the real world for robotics manipulation tasks. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6667–6674. IEEE, 2018. URL https://ieeexplore.ieee.org/document/8594519.

[47] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An Open-Source Generalist Robot Policy.

2023.
[48] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. URL https://arxiv.org/abs/2310.08864.
[49] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988. URL https://dl.acm.org/doi/10.5555/2969735.2969771.
[50] Marius-Constantin Popescu, Valentina E Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8(7):579–588, 2009. URL https://dl.acm.org/doi/10.5555/1639537.1639542.
[51] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*, 2023. URL https://arxiv.org/abs/2310.13724.
[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. URL http://proceedings.mlr.press/v139/radford21a.
[53] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-World Robot Learning with Masked Visual Pre-training. In *6th Annual Conference on Robot Learning*, 2022. URL https://openreview.net/forum?id=KWCZfuqshd.
[54] E. Rohmer, S. P. N. Singh, and M. Freese. CoppeliaSim (formerly V-REP): a Versatile and Scalable Robot Simulation Framework. In *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013. URL https://ieeexplore.ieee.org/document/6696520.
[55] Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/ross11a.html.
[56] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986. URL https://www.nature.com/articles/323533a0.
[57] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.
[58] Asad Ali Shahid, Loris Roveda, Dario Piga, and Francesco Braghin. Learning continuous control actions for robotic grasping with reinforcement learning. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 4066–4072. IEEE, 2020. URL https://ieeexplore.ieee.org/document/9282951.
[59] 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. URL https://proceedings.mlr.press/v164/shridhar22a.html.
[60] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022. URL https://proceedings.mlr.press/v205/shridhar23a.html.
[61] Priya Sundaresan, Suneel Belkhale, Dorsa Sadigh, and Jeannette Bohg. KITE: Keypoint-Conditioned Policies for Semantic Manipulation. *arXiv:2306.16605*, 2023. URL https://arxiv.org/abs/2306.16605.
[62] Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13438–13444. IEEE, 2021. URL https://arxiv.org/abs/2103.14127.
[63] 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. URL https://arxiv.org/abs/1703.06907.
[64] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023. URL https://arxiv.org/abs/2302.12422.
[65] Annie Xie, Lisa Lee, Ted Xiao, and Chelsea Finn. Decomposing the Generalization Gap in Imitation Learning for Visual Robotic Manipulation, 2023. URL https://arxiv.org/abs/2307.03659.
[66] Eliot Xing, Abhinav Gupta, Sam Powers, and Victoria Dean. Kitchenshift: Evaluating zero-shot generalization of imitation-based policy learning under domain shifts. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021. URL https://nips.cc/virtual/2021/35477.
[67] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021. URL https://iclr.cc/virtual/2022/poster/6275.
[68] Zhecheng Yuan, Sizhe Yang, Pu Hua, Can Chang, Kaizhe Hu, and Huazhe Xu. Rl-vigen: A reinforcement learning benchmark for visual generalization. *Advances in Neural*

Information Processing Systems, 36, 2024.

[69] Yanjie Ze, Ge Yan, Yueh-Hua Wu, Annabella Macaluso, Yuying Ge, Jianglong Ye, Nicklas Hansen, Li Erran Li, and Xiaolong Wang. Multi-Task Real Robot Learning with Generalizable Neural Feature Fields. *CoRL*, 2023. URL https://proceedings.mlr.press/v229/ze23a.html.

[70] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5628–5635. IEEE, 2018. URL https://arxiv.org/abs/1710.04615.

[71] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023. URL https://arxiv.org/abs/2304.13705.

[72] Yifeng Zhu, Zhenyu Jiang, Peter Stone, and Yuke Zhu. Learning generalizable manipulation policies with object-centric 3d representations. *arXiv preprint arXiv:2310.14386*, 2023.

[73] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. Robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020. URL https://arxiv.org/abs/2009.12293.