2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Decomposing the Generalization Gap in Imitation Learning for Visual Robotic Manipulation

Annie Xie[1*], Lisa Lee[2*], Ted Xiao[2], Chelsea Finn[1,2]

***Abstract*— What makes generalization hard for imitation learning in visual robotic manipulation? This question is difficult to approach at face value, but the environment from the perspective of a robot can often be decomposed into enumerable *factors of variation*, such as the lighting conditions or the placement of the camera. Empirically, generalization to some of these factors have presented a greater obstacle than others, but existing work sheds little light on precisely how much each factor contributes to the generalization gap. Towards an answer to this question, we study imitation learning policies in simulation and on a real robot language-conditioned manipulation task to quantify the difficulty of generalization to different (sets of) factors. We design a simulated benchmark of 19 tasks with 11 factors of variation to facilitate more controlled evaluations of generalization. From our study, we determine an ordering of factors based on generalization difficulty, that is consistent across simulation and our real robot setup.[1]**

I. INTRODUCTION

Robotic policies often fail to generalize to new environments, even after training on similar contexts and conditions. In robotic manipulation, data augmentation techniques [1], [2], [3], [4], [5] and representations pre-trained on large datasets [6], [7], [8], [9], [10], [11], [12] improve performance but a gap still remains. Simultaneously, there has also been a focus on the collection and curation of reusable robotic datasets [13], [14], [15], [16], [17], but there lacks a consensus on how much more data, and what *kind* of data, is needed for good generalization. These efforts could be made significantly more productive with a better understanding of which dimensions existing models struggle with. Hence, this work seeks to answer the question: *What are the factors that contribute most to the difficulty of generalization to new environments in vision-based robotic manipulation?*

To approach this question, we characterize environmental variations as a combination of independent factors, namely the background, lighting condition, distractor objects, table texture, object texture, table position, and camera position. This decomposition allows us to quantify how much each factor contributes to the generalization gap, which we analyze in the imitation learning setting (see Fig. 6 for a summary of our real robot evaluations). While vision models are robust to many of these factors already [18], [19], [20], robotic policies are less mature, due to the smaller and less varied datasets they train on. In robot learning, data collection is largely an *active* process, in which robotics researchers design and control the environment the robot interacts with. As a result, naturally occurring variations, such as different backgrounds, are missing in many robotics datasets. Finally, robotics tasks require dynamic, multi-step decisions, unlike computer vision tasks such as image classification. These differences motivate our formal study of these environment factors in the context of robotic manipulation.

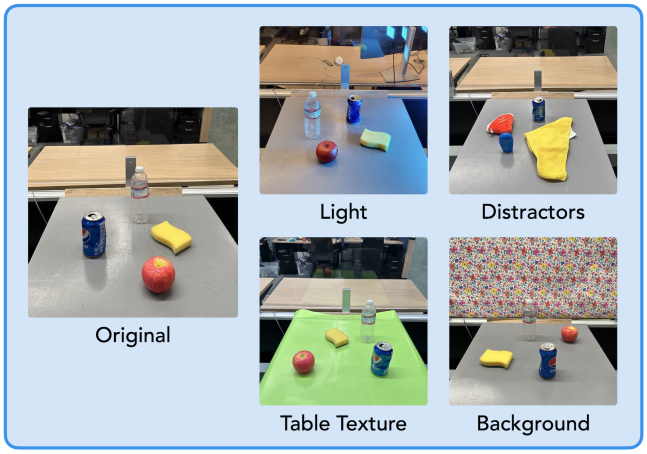


Fig. 1: Examples of our real robot environment. We systematically vary environment factors, including the lighting condition, distractor objects, table texture, background, and camera pose.

In our study, we evaluate a real robot manipulator on over 20 test scenarios featuring new lighting conditions, distractor objects, backgrounds, table textures, and camera positions. We also design a suite of 19 simulated tasks, equipped with 11 customizable environment factors, which we call *Factor World*, to supplement our study. With over 100 configurations for each factor, *Factor World* is a rich benchmark for evaluating generalization, which we hope will facilitate more fine-grained evaluations of new models, reveal potential areas of improvement, and inform future model design. Our study reveals the following insights:

- *Most pairs of factors do not have a compounding effect on generalization performance.* For example, generalizing to the combination of new table textures and new distractor objects is no harder than new table textures alone, which is the harder of two factors to generalize to. This result implies that we can study and address environment factors individually.
- *Random crop augmentation improves generalization even along non-spatial factors.* We find that random crop augmentation is a lightweight way to improve generalization to spatial factors such as camera positions, but also to non-spatial factors such as distractor objects and table textures.

*Equal contribution [1]Stanford University [2]Google DeepMind
[1]Videos & code are available at: https://sites.google.com/stanford.edu/gengap-icra

979-8-3503-8457-4/24/$31.00 ©2024 IEEE

- *Visual diversity from out-of-domain data dramatically improves generalization.* In our experiments, we find training on data from other tasks and domains like opening a fridge and operating a cereal dispenser can improve performance on picking an object from a table.

II. RELATED WORK

Datasets and benchmarks. Existing robotics datasets exhibit rich diversity along multiple dimensions, including objects [21], [16], [17], [22], domains [16], [4], [17], and tasks [13], [14], [15]. However, collecting high-quality and diverse data *at scale* is still an unsolved challenge, which motivates the question of how new data should be collected given its current cost. The goal of this study is to systematically understand the challenges of generalization to new objects and domains[2] and, through our findings, inform future data collection strategies. Simulation can also be a useful tool for understanding the scaling relationship between data diversity and policy performance, as diversity in simulation comes at a much lower cost [23], [24], [25], [26]. Many existing benchmarks aim to study exactly this [27], [28], [29], [30]; these benchmarks evaluate the generalization performance of control policies to new tasks [27], [28] and environments [29], [30]. *KitchenShift* [30] is the most related to our contribution *Factor World*, benchmarking robustness to shifts like lighting, camera view, and texture. However, *Factor World* contains a more complete set of factors (11 versus 7 in *KitchenShift*) with many more configurations of each factor (over 100 versus fewer than 10 in *KitchenShift*).

Pretrained representations and data augmentation. Because robotics datasets are generally collected in fewer and less varied environments, prior work has leveraged the diversity found in large-scale datasets from other domains like static images from ImageNet [31], videos of humans from Ego4D [32], and natural language [6], [9], [8], [10], [12]. While these datasets do not feature a single robot, pretraining representations on them can lead to highly efficient robotic policies with only a few episodes of robot data [8], [12], [33]. A simpler yet effective way to improve generalization is to apply image data augmentation techniques typically used in computer vision tasks [34]. Augmentations like random shifts, color jitter, and rotations have been found beneficial in many image-based robotic settings [1], [35], [2], [36], [3], [4]. While pretrained representations and data augmentations have demonstrated impressive empirical gains in many settings, we seek to understand when and why they help, through our factor decomposition of robotic environments.

Generalization studies. Several prior works have studied the robustness of robotic policies to different environmental shifts, such as harsher lighting, new backgrounds, and new distractor objects [37], [30], [22], [38]. Many interesting observations have emerged from them, such as how mild lighting changes have little impact on performance [37] and how new backgrounds (in their case, new kitchen countertops) have a bigger impact than new distractor objects [22]. However, these findings are often qualitative or lack specificity. For example, the performance on a new kitchen countertop could be attributed to either the appearance or the height of the new counter. A goal of our study is to formalize these prior observations through systematic evaluations and to extend them with a more comprehensive and fine-grained set of environmental shifts.

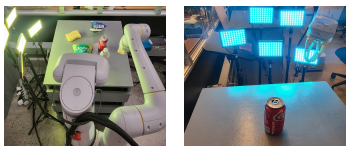
[2]We define an environment as the domain and its objects.

III. ENVIRONMENT FACTORS

To draw more robust conclusions, our study is conducted across three different domains: on a real robot and in the *Factor World* and *KitchenShift* [30] simulators. On the real robot and in *KitchenShift*, we use pre-existing datasets to understand how models trained on them are impacted by factored variations. Importantly, the environmental factors and distributions over them are *not* designed by us, and thus are representative of experimental setups studied in robotics research. To augment these domains, we also design *Factor World* which allows easier control over individual factors and generation of datasets with specific factor distributions.

A. Real Robot Manipulation

In our real robot evaluations, we study: lighting condition, distractor objects, background, table texture, and camera pose. In addition to selecting factors that are specific and controllable, we also take inspiration from prior work, which has studied robustness to many of these shifts [37], [30], [22], thus signifying their relevance in real-world scenarios. Our experiments are conducted with mobile manipulators. The robot has a right-side arm with seven DoFs, gripper with two fingers, mobile base, and head with integrated cameras. The environment, visualized in Fig. 1, consists of a cabinet top that serves as the robot workspace and an acrylic wall that separates the workspace and office background. To control the lighting condition in our evaluations, we use several bright LED light sources with different colored filters to create colored hues and new shadows (see Fig. 2). We introduce new table textures and backgrounds by covering the cabinet top and acrylic wall, respectively, with patterned paper. We also shift the camera pose by changing the robot's head orientation Due to the practical challenges of studying factors like the table position and height, we reserve them for our simulated experiments.

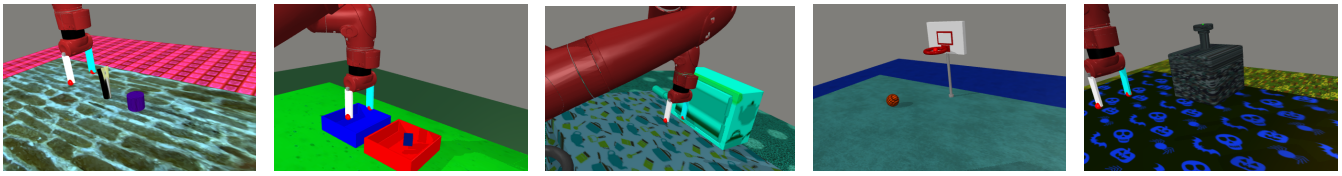


(a) Light setup 1 (b) Light setup 2

Fig. 2: Evaluation setup for lighting.

B. Simulation

Factor World. We implement the environmental shifts on top of *Meta World* [27]. While *Meta World* is rich in diversity of control behaviors, it lacks diversity in the environment, placing the same table at the same position against the same background. Hence, we implement 11 different factors of variation, visualized in Fig. 3 and fully enumerated on the supplementary website. These include lighting; texture, size,



(a) Pick Place (b) Bin Picking (c) Door (Open, Lock) (d) Basketball (e) Button (Top, Side, Wall)

Fig. 3: *Factor World*, a suite of 19 visually diverse robotic manipulation tasks. Each task can be configured with multiple factors of variation such as lighting; texture, size, shape, and initial position of objects; texture of background (table, floor); position of the camera and table relative to the robot; and distractor objects.

shape, and initial position of objects; texture of the table and background; the camera pose and table position relative to the robot; the initial arm pose; and distractor objects. In our study, we exclude the object size and shape, as an expert policy that can handle any object is more difficult to design, and the initial arm pose, as this can usually be fixed whereas the same control cannot be exercised over the other factors, which are inherent to the environment.

Textures (table, floor, objects) are sampled from 162 texture images (81 for train, 81 for eval) and continuous RGB values in $[0, 1]^3$, which modifies the texture image. Distractor objects are sampled from 170 object meshes (100 for train, 70 for eval) in Google's Scanned Objects Dataset [39], [40]. For lighting, we sample continuous ambient and diffuse values in $[0.2, 0.8]$. Changes in camera and table positions are sampled from $[-0.025, 0.025]$ meters. While fixing the initial position of an object across trials is feasible with a simulator, it is generally difficult to precisely replace an object to its original position in physical setups. Thus, we randomize the initial position of the object (between $[-0.1, 0.1]$ meters) in each episode in the experiments.

KitchenShift [30]. In addition to *Factor World*, we examine a second simulated environment, *KitchenShift*. *KitchenShift* modifies Franka Kitchen with variations to the lighting, camera view, and textures (object, counter, and floor). There are 4 lighting settings, 10 camera positions, 4 counter textures, 7 floor textures, 4 microwave models, 6 cabinet textures, and 8 kettle models. The microwave and cabinets represent distractors, while the kettle models are different object textures. The table position is fixed in *KitchenShift*.

IV. STUDY DESIGN

We seek to understand how each factor in Sec. III contributes to the difficulty of generalization. In our pursuit of an answer, we aim to replicate, to the best of our ability, the scenarios that robotics practitioners are likely to encounter in the real world. We therefore start by selecting a set of tasks commonly studied in the robotics literature and the data collection procedure (Sec. IV-A). Then, we describe the algorithms studied and our evaluation protocol (Sec. IV-B).

A. Control Tasks and Datasets

Real robot. We study the language-conditioned manipulation problem from [22], specifically, focusing on the "pick" skill for which the most data is available. The goal is to pick up the object specified in the language instruction. For example, when given the instruction "pick pepsi can", the robot should pick up the pepsi can among the distractor objects from the countertop (Fig. 1). We select six objects for our evaluation; all "pick" tasks can be found on the website. The observation consists of 300×300 RGB image observations from the last six time-steps and the language instruction, while the action controls movements of the arm (xyz-position, roll, pitch, yaw, opening of the gripper) and movements of the base (xy-position, yaw). The actions are discretized along each of the 10 dimensions into 256 uniform bins. The real robot manipulation dataset consists of over 115K human-collected demonstrations, collected across 13 skills, with over 100 objects, three tables, and three locations. The dataset is collected with a fixed camera orientation but randomized initial base position in each episode.

Simulation. While *Factor World* supports 19 manipulation tasks, our study focuses on 3 tasks commonly studied in robotics: pick-place (Fig. 3a), bin-picking (Fig. 3b), and door-open (Fig. 3c). In pick-place, the agent must move a block to the goal among a distractor object placed in the scene. In bin-picking, the agent must move a block from the right-side bin to the left-side bin. In door-open, the agent must pull on the door handle. We use scripted expert policies from the *Meta World* benchmark, which compute expert actions given the object poses, to collect demonstrations in each simulated task. The agent is given 84×84 RBG image observations, the robot's end-effector position from the last two time-steps, and the distance between the robot's fingers from the last two time-steps. The actions are the desired change in the 3D-position of the end-effector and whether to open or close the gripper. In *KitchenShift*, we study the kettle task, which requires moving the kettle from the bottom to the top burner. See [30] for environment details.

B. Algorithms and Evaluation Protocol

The real robot manipulation policy uses the RT-1 architecture [22], which tokenizes the images, text, and actions, attends over these tokens with a Transformer [41], and trains with a language-conditioned imitation learning objective. In simulation, we equip vanilla behavior cloning with several different methods for improving generalization. Specifically, we evaluate techniques for image data augmentation (random crops and random photometric distortions) and evaluate pre-trained representations (CLIP [7] and R3M [12]) for encoding image observations. More details on the implementation and training procedure can be found on the website.

Evaluation protocol. On the real robot task, we evaluate the policies on 2 new lighting conditions, 3 sets of new

distractor objects, 3 new table textures, 3 new backgrounds, and 2 new camera poses. For each factor of interest, we conduct 2 evaluation trials in each of the 6 tasks, and randomly shuffle the object and distractor positions between trials. We report the success rate averaged across the 12 trials. To evaluate the generalization behavior of the trained policies in *Factor World*, we shift the train environments by randomly sampling 100 new values for the factor of interest, creating 100 test environments. In *KitchenShift*, we evaluate on 1 lighting setting, 7 camera positions, 1 counter texture, 4 floor textures, 1 microwave model, 3 cabinet textures, and 5 kettle models. We report the average **generalization gap**, which is defined as $P_T - P_F$, where P_T is the success rate on the train environments and P_F is the new success rate under shifts to factor F.

V. EXPERIMENTAL RESULTS

Our experiments aim to answer the following questions:

- How much does each environment factor contribute to the generalization gap? (Sec. V-A)
- What effects do data augmentation, pretrained representations, and model architecture have on the generalization performance? (Sec. V-B)
- How do different data collection strategies, such as prioritizing visual diversity in the data, impact downstream generalization? (Sec. V-C)

We also study different image resolutions and control frequencies. The results of these ablations are on the website.

A. Impact of Environment Factors on Generalization

Individual factors. We begin our real robot evaluation by benchmarking the model's performance on the set of six training tasks, with and without shifts. Without shifts, the policy achieves an average success rate of 91.7%. Our results with shifts are presented in Fig. 5, as the set of green bars. We find that the new backgrounds have little impact on the performance (88.9%), while new distractor objects and new lighting conditions have a slight effect, decreasing success rate to 80.6% and 83.3% respectively. Finally, changing the table texture and camera orientation causes the biggest drop, to 52.8% and 45.8%, as the entire dataset uses a fixed head pose. Since we use the same patterned paper to introduce variations in backgrounds and table textures, we can directly compare these two factors, and conclude that new textures are harder to generalize to than new backgrounds.

Fig. 4a compares the generalization gap due to each individual factor on *Factor World*. We plot this as a function of the number of training environments represented in the dataset, where an environment is parameterized by the sampled value for each factor of variation. The success rates under individual factor shifts in *KitchenShift* are visualized in Fig. 6. **Consistent across simulated and real-world results, new backgrounds, distractors, and lighting are easier factors to generalize to, while new table textures and camera positions are harder.** In *Factor World*, new backgrounds are harder than distractors and lighting, in

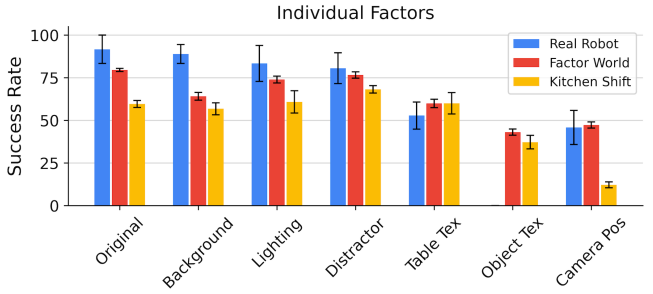
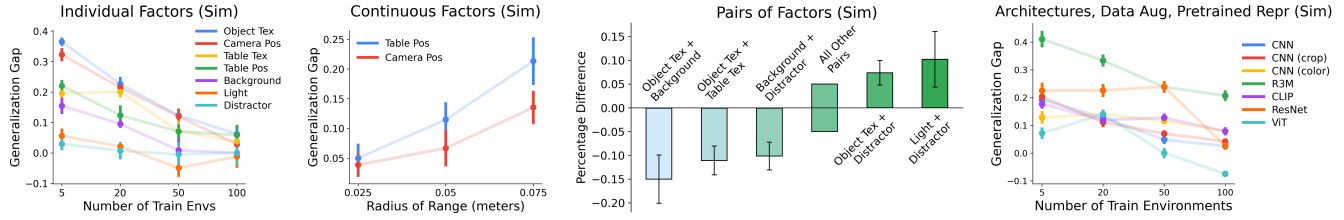


Fig. 6: Success rates on different shifts across 3 domains. Object texture is not evaluated on the robot.

contrast to the real robot results, where they were the easiest. This may be because the real robot dataset contains a significant amount of background diversity, relative to the lighting and distractor factors, as described in Sec. IV-A. In *Factor World*, we additionally study object textures and table positions, including the height of the table. New object textures are about as hard to overcome as camera positions, and new table positions are as hard as table textures. Fortunately, the generalization gap closes significantly for *all* factors, from a maximum gap of 0.4 to less than 0.1, when increasing the number of training environments from 5 to 100. Notably, table (counter) textures are easier in *KitchenShift* compared to *Factor World* and the real robot. This is likely because while the texture of the counter changes, the texture of the stovetop, on which the kettle lies, does not.

Pairs of factors. Next, we evaluate with respect to pairs of factors to understand how they interact, i.e., whether generalization to new pairs is harder (or easier) than generalizing to one of them. On the real robot, we study the factors with the most diversity in the training dataset: table texture + distractors and table texture + background. Introducing new background textures or new distractors on top of a new table texture does not make it any harder than the new table texture alone (see green bars in Fig. 5). The success rate with new table texture + new background is 55.6% and with new table texture + new distractors is 50.0%, comparable to the evaluation with only new table textures, which is 52.8%.

In *Factor World*, we evaluate all 21 pairs of the 7 factors, and report a different metric: the success rate gap, normalized by the harder of the two factors. Concretely, this metric is defined as $(P_{A+B} - \min(P_A, P_B)) \,/\, \min(P_A, P_B)$, where P_A is the success rate under shifts to factor A, P_B is the success rate under shifts to factor B, and P_{A+B} is the success rate under shifts to both. **Most pairs of factors do not have a compounding effect on generalization performance.** For 16 of 21 pairs, the relative percentage difference in the success rate lies between -6% and 6%. In other words, generalizing to the combination of two factors is not significantly harder or easier than individual factors. In Fig. 4c, we visualize the performance difference for the remaining 5 factor pairs that lie outside of this $(-6\%, 6\%)$ range (see website for results with all factor pairs). Interestingly, the following factors combine synergistically, making it easier to generalize to compared to the (harder of the) individual factors: object texture + distractor and light + distractor. This result suggests these factors can be studied independently of



(a) (b) (c) (d)

Fig. 4: (a) Generalization gap under shifts to individual factor in *Factor World*. (b) Generalization gap versus the radius of the range that camera and table positions are sampled from, in *Factor World*. (c) Performance on pairs of factors, reported as the percentage difference relative to the harder factor of the pair, in *Factor World*. All results are averaged across the 3 simulated tasks with 5 seeds for each task. Error bars represent standard error of the mean. (d) Generalization gap with data augmentations, pretrained representations, and different architectures in *Factor World*. Lower is better. Results are averaged across the 7 factors, 3 tasks, and 5 seeds for each task.

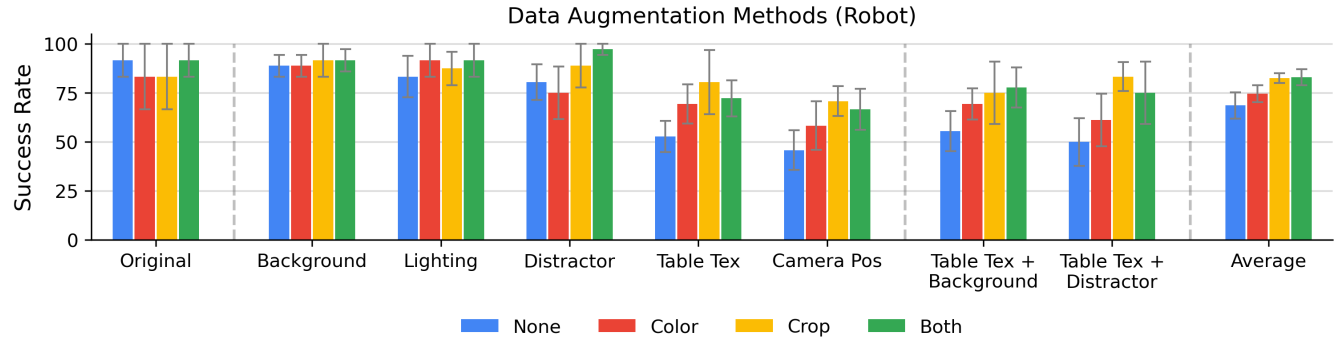


Fig. 5: Performance of real-robot policies trained without data augmentation (blue), with random photometric distortions (red), with random crops (yellow), and with both (green). The results discussed in Sec. V-A are with "Both". "Original" is the success rate on train environments, "Background" is the success rate when we perturb the background, "Distractors" is where we replace the distractors with new ones, etc. Error bars represent standard error of the mean. We also provide the average over all 7 (sets of) factors on the far right.

one another, and improvements with respect to one factor may carry over to multiple factor shifts.

Continuous factors. The camera position and table position factors are continuous, unlike the other factors which are discrete, hence the generalization gap with respect to these factors will depend on the range that we train and evaluate on. We aim to understand how much more difficult training and generalizing to a wider range of values is, by studying the gap with the following range radii: 0.025, 0.050, ad 0.075 meters. For both camera-position and table-position factors, as we linearly increase the radius, the generalization gap roughly doubles (see Fig. 4b). This pattern suggests: (1) performance can be dramatically improved by keeping the camera and table position as constant as possible, and (2) generalizing to wider ranges may require significantly more diversity, i.e., examples of camera and table positions in the training dataset. However, in Sec. V-B, we see that existing methods can address the latter issue to some degree.

B. Augmentations, Pretrained Representations, Architectures

Data augmentation. We study 2 forms of augmentation: (1) random crops and (2) random photometric distortions. The photometric distortion randomly adjusts the brightness, saturation, hue, and contrast of the image, and applies random cutout and random Gaussian noise. Fig. 5 and Fig. 4d show the results for the real robot and *Factor World* respectively. On the robot, **crop augmentation improves generalization along multiple environment factors, most significantly to new camera positions and new table textures**. While the improvement on a spatial factor like camera position is intuitive, we find the improvement on a non-spatial factor like table texture surprising. More in line with our expectations, the photometric distortion augmentation improves the performance on texture-based factors like table texture in the real robot environment and object, table and background in the simulated environment (see the website for *Factor World* results by factor).

Pretrained representations. On the real robot, we evaluate the RT-2 policy [42], which finetunes PaLI-55B on a robot dataset. RT-2 has been shown to generalize better to new objects, instructions, and, most relevant to our work, *environments*. Importantly, the new "environments" that [42] evaluate include a kitchen and desk, which present new objects and workstation heights, among many other factors. Hence, we are interested in evaluating RT-2 along factored environment variations. As shown in Fig. 7, the generalization performance of RT-2 (green) does not improve upon RT-1 (yellow). Interestingly, the success rate of RT-2 on all factors is similar (at 75%), except on camera positions (54%).

We also study (1) R3M [12] and (2) CLIP [7] in *Factor World*. While these representations are trained on real, non-robotics datasets, policies trained on top of them have been shown to perform well in (simulated and real) robotics environments from a small amount of data. However, while they achieve good performance on training environments, they struggle to generalize to new but similar environments,

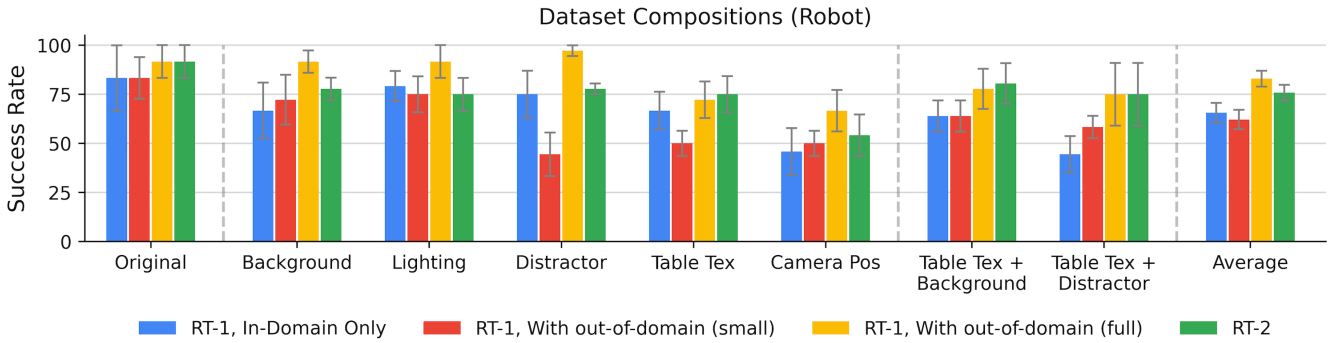


Fig. 7: Performance of RT-1 policies trained with in-domain data only (blue), a small version of the in- and out-of-domain dataset (red), and the full version of the in- and out-of-domain dataset (yellow). The RT-2 policy is pretrained and co-finetuned on Internet-scale data (green). Error bars represent standard error of the mean. We also provide the average over all 7 (sets of) factors on the far right.

leaving a large generalization gap across many factors (see Fig. 4d). Though, we find that CLIP does improve upon a trained-from-scratch CNN with new object textures.

Model architectures. In addition to the CNN, we also evaluate policies trained with a ResNet [43] and a Vision Transformer (ViT) [44] encoder. Both encoders succeed under more training environments (see Fig. 4d). However, with fewer train environments, the ViT encoder tends to outperform the CNN variants, while the ResNet encoder performs the worst of the three. We also find a similar ordering of factors across architectures (see the website for results by factor), with one main exception: ResNets generalize to camera positions better relative to other factors.

C. Investigating Different Strategies for Data Collection

Augmenting visual diversity with out-of-domain data. As described in Sec. IV-A, our real robot dataset includes demonstrations collected from other domains and tasks like opening a fridge and operating a cereal dispenser. Only 35.2% of the 115K demonstrations are collected in the same domain as our evaluations. While the remaining demonstrations are out of domain and focus on other skills such as drawer manipulation, they add visual diversity, such as new objects and new backgrounds, and demonstrate robotic manipulation behavior, unlike the data that R3M and CLIP pretrain on. We consider the dataset with only in-domain data, which we refer to as In-domain only. In Fig. 7, we compare In-domain only (blue) to the full dataset, which we refer to as With out-of-domain (full) (yellow). While the performance on the original six training tasks is comparable, the success rate of the In-domain only policy drops significantly across the different environment shifts, and the With out-of-domain (full) policy is more successful across the board. **Unlike representations pretrained on non-robotics datasets (Sec. V-B), out-of-domain robotics data can improve in-domain generalization.**

Prioritizing visual diversity with out-of-domain data. We also consider a uniformly subsampled version of the With out-of-domain (full) dataset, which we refer to as With out-of-domain (small). With out-of-domain (small) has the same number of demonstrations as In-domain only, allowing us to directly compare whether the in-domain data or out-of-domain data is more valuable. We emphasize that With out-of-domain (small) has significantly fewer in-domain demonstrations of the "pick" skill than In-domain only. Intuitively, one would expect the in-domain data to be more useful. However, in Fig. 7, we see that the With out-of-domain (small) policy (red) performs comparably with the In-domain only policy (blue) across most of the factors. The main exception is scenarios with new distractors, where the In-domain only policy has a 75.0% success rate while the With out-of-domain (small) policy is successful in 44.4% of the trials. Thus, if a particular application demands good generalization to distractors or table textures over other factors, in-domain data should be prioritized. However, if we only consider the average performance over all factors, collecting out-of-domain data does not harm performance.

VI. DISCUSSION

Summary. In this work, we studied the impact of different environmental variations on generalization performance. We determined an ordering of the environment factors in terms of generalization difficulty, that is consistent across simulation and our real robot setup, and quantified the impact of different solutions like data augmentation. Notably, many of the solutions studied were developed for computer vision tasks like image classification. While some of them transferred well to the robotic imitation learning setting, it may be fruitful to develop algorithms that prioritize this setting and its unique considerations, including the sequential nature of predictions and the often continuous, multi-dimensional action space in robotic setups. We hope this work encourages researchers to develop solutions that target the specific challenges in robotic generalization identified by our work.

Limitations. There are limitations to our study, which focuses on a few, but representative, robotic tasks and environment factors in the imitation setting. Our real-robot experiments required conducting a total number of 1440 evaluations over all factor values, tasks, and methods, and it is challenging to increase the scope of the study because of the number of experiments required. Fortunately, future work can utilize our simulated benchmark *Factor World* to study additional tasks, additional factors, and generalization in the reinforcement learning setting.

REFERENCES

[1] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas, "Reinforcement learning with augmented data," *Advances in neural information processing systems*, vol. 33, pp. 19 884–19 895, 2020.

[2] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto, "Mastering visual continuous control: Improved data-augmented reinforcement learning," *arXiv preprint arXiv:2107.09645*, 2021.

[3] N. Hansen and X. Wang, "Generalization in reinforcement learning by soft data augmentation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 611–13 617.

[4] S. Young, D. Gandhi, S. Tulsiani, A. Gupta, P. Abbeel, and L. Pinto, "Visual imitation made easy," in *Conference on Robot Learning*. PMLR, 2021, pp. 1992–2005.

[5] C. Graf, D. B. Adrian, J. Weil, M. Gabriel, P. Schillinger, M. Spies, H. Neumann, and A. Kupcsik, "Learning dense visual descriptors using image augmentations for robot manipulation tasks," *arXiv preprint arXiv:2209.05213*, 2022.

[6] L. Yen-Chen, A. Zeng, S. Song, P. Isola, and T.-Y. Lin, "Learning to see before learning to act: Visual pre-training for manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7286–7293.

[7] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.

[8] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, "Simple but effective: Clip embeddings for embodied ai," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 829–14 838.

[9] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 894–906.

[10] R. Shah and V. Kumar, "Rrl: Resnet as representation for reinforcement learning," *arXiv preprint arXiv:2107.03380*, 2021.

[11] S. Parisi, A. Rajeswaran, S. Purushwalkam, and A. Gupta, "The unsurprising effectiveness of pre-trained vision models for control," *arXiv preprint arXiv:2203.03580*, 2022.

[12] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," *arXiv preprint arXiv:2203.12601*, 2022.

[13] P. Sharma, L. Mohan, L. Pinto, and A. Gupta, "Multiple interactions made easy (mime): Large scale demonstrations data for imitation," in *Conference on robot learning*. PMLR, 2018, pp. 906–915.

[14] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, *et al.*, "Roboturk: A crowdsourcing platform for robotic skill learning through imitation," in *Conference on Robot Learning*. PMLR, 2018, pp. 879–893.

[15] A. Mandlekar, J. Booher, M. Spero, A. Tung, A. Gupta, Y. Zhu, A. Garg, S. Savarese, and L. Fei-Fei, "Scaling robot supervision to hundreds of hours with roboturk: Robotic manipulation dataset through human reasoning and dexterity," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1048–1055.

[16] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn, "Robonet: Large-scale multi-robot learning," *arXiv preprint arXiv:1910.11215*, 2019.

[17] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine, "Bridge data: Boosting generalization of robotic skills with cross-domain datasets," *arXiv preprint arXiv:2109.13396*, 2021.

[18] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *arXiv preprint arXiv:1903.12261*, 2019.

[19] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, *et al.*, "The many faces of robustness: A critical analysis of out-of-distribution generalization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8340–8349.

[20] R. Geirhos, K. Narayanappa, B. Mitzkus, T. Thieringer, M. Bethge, F. A. Wichmann, and W. Brendel, "Partial success in closing the gap between human and machine vision," *Advances in Neural Information Processing Systems*, vol. 34, pp. 23 885–23 899, 2021.

[21] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, *et al.*, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Conference on Robot Learning*. PMLR, 2018, pp. 651–673.

[22] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[23] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," *arXiv preprint arXiv:1804.10332*, 2018.

[24] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3803–3810.

[25] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox, "Closing the sim-to-real loop: Adapting simulation randomization with real world experience," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8973–8979.

[26] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 627–12 637.

[27] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on robot learning*. PMLR, 2020, pp. 1094–1100.

[28] K. Cobbe, C. Hesse, J. Hilton, and J. Schulman, "Leveraging procedural generation to benchmark reinforcement learning," in *International conference on machine learning*. PMLR, 2020, pp. 2048–2056.

[29] A. Stone, O. Ramirez, K. Konolige, and R. Jonschkowski, "The distracting control suite–a challenging benchmark for reinforcement learning from pixels," *arXiv preprint arXiv:2101.02722*, 2021.

[30] E. Xing, A. Gupta, S. Powers, and V. 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.

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[32] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, *et al.*, "Ego4d: Around the world in 3,000 hours of egocentric video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 995–19 012.

[33] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang, "Vip: Towards universal visual reward and representation via value-implicit pre-training," *arXiv preprint arXiv:2210.00030*, 2022.

[34] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[35] I. Kostrikov, D. Yarats, and R. Fergus, "Image augmentation is all you need: Regularizing deep reinforcement learning from pixels," *arXiv preprint arXiv:2004.13649*, 2020.

[36] N. Hansen, H. Su, and X. Wang, "Stabilizing deep q-learning with convnets and vision transformers under data augmentation," *Advances in neural information processing systems*, vol. 34, pp. 3680–3693, 2021.

[37] R. Julian, B. Swanson, G. S. Sukhatme, S. Levine, C. Finn, and K. Hausman, "Never stop learning: The effectiveness of fine-tuning in robotic reinforcement learning," *arXiv preprint arXiv:2004.10190*, 2020.

[38] G. Zhou, V. Dean, M. K. Srirama, A. Rajeswaran, J. Pari, K. B. Hatch, A. Jain, T. Yu, P. Abbeel, L. Pinto, *et al.*, "Train offline, test online: A real robot learning benchmark," in *Deep Reinforcement Learning Workshop NeurIPS 2022*.

[39] K. Zakka, "Scanned Objects MuJoCo Models," 7 2022. [Online]. Available: https://github.com/kevinzakka/mujoco_scanned_objects

[40] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke, "Google scanned objects: A high-quality dataset of 3d scanned household items," 2022. [Online]. Available: https://arxiv.org/abs/2204.11918

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[42] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[44] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.