

AugInsert: Learning Robust Visual-Force Policies via Data Augmentation for Object Assembly Tasks

Ryan Diaz¹, Adam Imdieke¹, Vivek Veeriah², Karthik Desingh¹

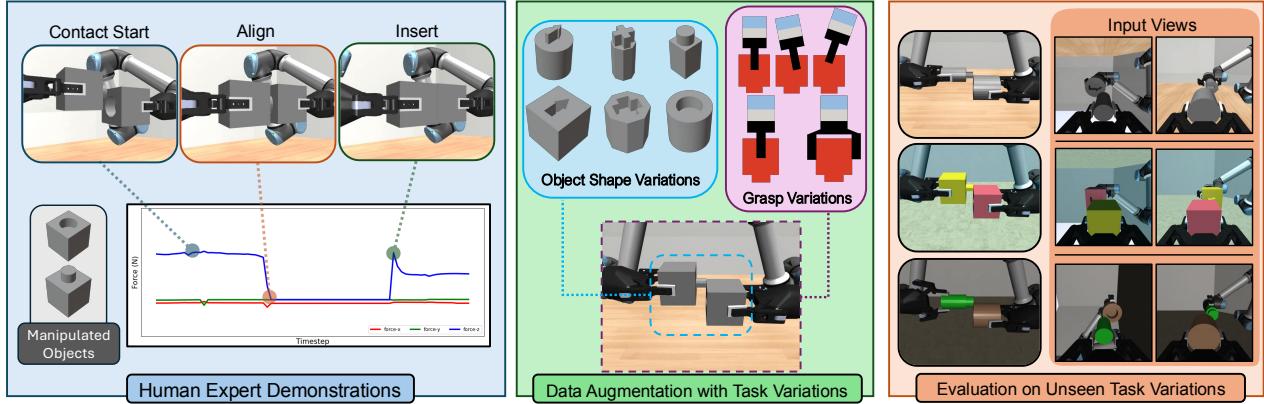


Fig. 1: AugInsert is a data collection and policy evaluation pipeline aimed towards analyzing the robustness of a multisensory (vision, force-torque, and proprioception) model with respect to different observation-level task variations in object shape, grasp pose, and visual environmental appearance. Our framework introduces task variations to a dataset of human-collected demonstrations through a system of online data augmentation.

Abstract—This paper primarily focuses on learning robust visual-force policies in the context of high-precision object assembly tasks. Specifically, we focus on the *contact phase* of the assembly task where both objects (peg and hole) have made contact and the objective lies in maneuvering the objects to complete the assembly. Moreover, we aim to learn contact-rich manipulation policies with multisensory inputs on limited expert data by expanding human demonstrations via online data augmentation. We develop a simulation environment with a dual-arm robot manipulator to evaluate the effect of augmented expert demonstration data. Our focus is on evaluating the robustness of our model with respect to certain task variations: *grasp pose*, *peg/hole shape*, *object body shape*, *scene appearance*, *camera pose*, and *force-torque/proprioception noise*. We show that our proposed data augmentation method helps in learning a multisensory manipulation policy that is robust to unseen instances of these variations, particularly physical variations such as *grasp pose*. Additionally, our ablative studies show the significant contribution of force-torque data to the robustness of our model. For additional experiments and qualitative results, we refer to the project webpage <https://bit.ly/47skWXH>.

I. INTRODUCTION

Peg-in-hole assembly tasks are representative of many everyday tasks involving contact-rich manipulation, where objects remain in contact throughout the process. Consider tasks such as capping a bottle, plugging in a cable, or inserting a K-cup pod into a coffee machine. These tasks typically involve multiple phases: a *pick-up phase* where the objects (e.g., cap and bottle) are grasped and picked up by the grippers; an *orienting phase*, where the objects

are maneuvered into a desired relative pose before contact; and a *contact phase*, where the objects are in contact, and appropriate forces are applied to complete the task (e.g., screwing the cap onto the bottle, fully inserting the plug). While the *contact phase* may seem trivial to perform for humans, it poses significant challenges for robots—especially those meant to operate in household contexts—to learn from data due to several factors: a) contact-rich manipulation tasks are difficult for humans to demonstrate in order to facilitate large scale data collection for learning policies from demonstration, b) these tasks require multisensory observations (visual and tactile) and robust encoding methods to extract meaningful representations for policy learning [1]–[4], and c) humans can easily generalize these tasks to novel scenarios (e.g. varying object shapes and geometries), but such generalization is highly challenging for robot learning models while maintaining the robustness required [5], [6].

In this paper, we address these challenges by proposing a multisensory observation encoding and policy learning framework, leveraging data augmentation on limited human demonstrations to train contact-rich manipulation policies for an assembly task that generalizes to unseen task variations. Our pipeline processes multiple camera views and force-torque readings from a dual-arm setup. We also make use of a multisensory data augmentation method via trajectory replay that can introduce both sensor-specific (camera and force-torque sensor) variations as well as physical factors such as manipulated object shape, peg and hole geometries, and grasp pose variations that affect the sensing modalities. In this way, we can expand small expert datasets to learn robust manipulation policies that can handle a wide variety of environmental conditions.

¹R. Diaz, A. Imdieke, and K. Desingh are with the Department of Computer Science and Engineering at the University of Minnesota, Twin Cities. {diaz0329, imdie022, kdesingh}@umn.edu

²V. Veeriah is with Google DeepMind. vveeriah@google.com

To analyze the robustness of our model with respect to specific observation-level task variations and understand the effect of our data augmentation method, we develop an experimental setup in the MuJoCo [7] simulation environment with a dual-arm robot that can manipulate objects with peg and hole geometries to complete the assembly task. Our experiments show that certain variations, such as *Grasp Pose* variations, cause large drops in success rate for our task and so should be included in training data through data augmentation in order to ensure robustness to these variations. Additionally, we conduct ablation studies to understand how each sensory modality in the multisensory setup affects the performance of the contact-rich assembly task. These studies also help identify the specific variations impacted by each modality. We observe that touch provides the most relevant information for the task and supports model robustness; visual input, on the other hand, has the least significant impact on generalization ability while also being susceptible to many of our task variations. We provide an extensive discussion of these results in the following sections.

The main contributions of our work are:

- A dual-arm object assembly task formulation and a simulation environment that allows for independent application of 6 types of task variations that involve 54 different types of peg and hole objects.
- A data augmentation pipeline integrated with the simulation environment that can generate new observations for training using a set of provided expert demonstrations through trajectory replay.
- An extensive set of experiments to understand the effect of data augmentation and evaluate the robustness of specific sensing modes against observation-level task variations.

II. RELATED WORK

A. Multisensory Contact-Rich Manipulation

Multisensory contact-rich manipulation in the form of peg-in-hole insertion using vision and force-torque data has been widely studied. Lee et al. [1] developed a self-supervised learning method to learn a multisensory representation using vision and force-torque. In their follow up work [8] they add a reconstruction and latent distance objective to their self-supervision framework to mitigate the effects of possible sensor corruption. More recently, Spector et al. [3], [4] developed a multiview and multisensory system for localizing and performing realistic insertion tasks, Chen et al. [2] used a transformer [9] encoder for vision and force-torque inputs to learn a higher-quality representation, and Kohler et al. [10] leveraged symmetry in the peg-in-hole task by using equivariant networks to improve sample efficiency. Although these works can achieve efficient peg-in-hole assembly, their generalization studies are limited in scope when evaluating robustness to both physical and sensory task variations.

B. Evaluating Generalization Abilities of Learned Policies

Generalization is difficult to define in robot manipulation policy learning as there are several factors in the robot's

environment that could vary from training phase to the evaluation phase. There have been recent efforts to perform in-depth analyses of the generalization abilities of visuomotor robotic policies by decomposing task environment variations into individual variation “factors” [5], [6], [11], [12]. We aim to bring this type of analysis to the multisensory domain by introducing a set of variation factors that perturb force-torque and proprioceptive inputs in addition to image inputs.

C. Extrapolating Human Trajectories for Imitation Learning

Imitation learning can be a powerful method for learning complex tasks in robotics, but it can be challenging to collect large enough demonstration datasets for learning effective policies. Recent works address this problem by extrapolating a small dataset of human demonstrations; Mandlekar et al. [13] generated new trajectories by decomposing task demonstrations into object-centric subtasks, and Jia et al. [14] used different point cloud projections as new observations to simulate new transitions within a demonstration. Focusing more on robotic assembly, Ankile et al. [15] annotated expert trajectories with “bottleneck” states off of which perturbations and their corresponding corrective actions could be automatically generated. These works somewhat resemble our trajectory replay method for online augmentation, but they are more suited for long-horizon tasks that do not necessarily focus on precision.

III. TASK SETUP

A. Assembly Task Definition

Our experimental setup consists of a dual-arm robot manipulator with a multisensory configuration, featuring two force-torque (F/T) sensors and two RGB cameras attached to its wrists. The robot is tasked with performing an insertion assembly, where one arm’s gripper holds a peg-shaped object and inserts it into a hole-shaped object held by the other arm’s gripper. Since our focus is on the contact phase of the assembly, the objects are already in contact at the start of the task.

The objective of the robot learning framework is to execute the assembly task without explicit information about the object geometries or peg-and-hole shapes, while maintaining robustness to various task variations. We take the behavior cloning approach where expert demonstrations are used to clone the contact-rich manipulation policy to perform the assembly.

B. Task Initialization

The task is initialized with the peg and hole offset within a range of [1.5cm, 3.0cm] along both the X and Y axes relative to the object coordinate frame (perpendicular to the direction of insertion). Our setup ensures that while the arm holding the peg moves, the other arm remains compliant, applying a constant force until the peg and hole are aligned. To define a successful task rollout, we consider position coordinates $\mathbf{p} = (x_p, y_p, z_p)$ for the peg object and $\mathbf{h} = (x_h, y_h, z_h)$ for the hole object in the global coordinate frame. We set thresholds $\mathbf{d} = (d_x, d_y, d_z)$ such that during a successful insertion, $|\mathbf{p}_i - \mathbf{h}_i| < \mathbf{d}_i$ for all axes $i \in \{x, y, z\}$.

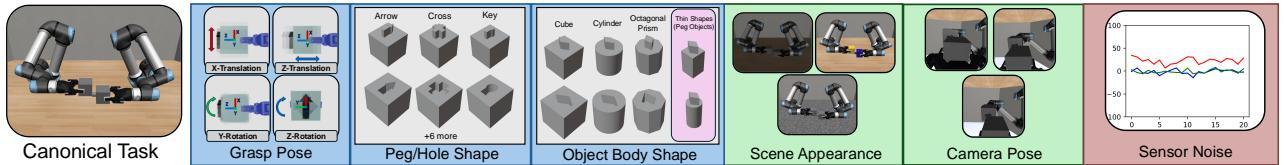


Fig. 2: A visualization of each of the task variations used in our environment setup. We differentiate between physical task variations (in blue) and sensor-based task variations that target vision (green) and force-torque/proprioception (red).

C. Task Variations

To evaluate the robustness of our trained models, we design a set of observation-level task variations which alter the distribution of incoming observations while preserving the underlying task. In total, there are six variations that are part of the experiments (see Figure 2 for sample visualizations of these variations):

- 1) **Peg and Hole Shape:** There are 9 possible peg and hole shapes, with each peg and hole sharing the same shape and allowing for a tolerance to ensure insertion compatibility.
- 2) **Object Body Shape:** There are 3 possible object body shapes, and the peg and hole in a given pair may not share the same shape. Additionally, we create thinner versions of the peg to introduce variability, resulting in 6 total peg and hole object pairs.
- 3) **Grasp Pose:** Our grasp pose variation follows the approach in [16], which provides a more detailed overview.
- 4) **Scene Appearance:** This category encompasses variations in lighting, floor texture, and object color.
- 5) **Camera Pose:** We vary the position and orientation of the wrist cameras between demonstrations, while keeping them constant within each demonstration.
- 6) **Sensor Noise:** We add zero-mean Gaussian noise to low-dimensional measurements, with standard deviations set to approximately 5% of the maximum measurement for force-torque and 4% of the maximum task-initialized offset for proprioception.

Canonical Task Setup: We define a “canonical” task setup which represents an environment without any task variations applied. For discrete task variations, we choose the key peg and hole shape, cube object body shape, and light-wood floor texture in our canonical setup.

IV. METHODOLOGY

A. Imitation Learning Framework

Observation and Action Spaces: In our task, the observation space is defined as a composition of four modality spaces, $\mathcal{O} = \mathcal{I}_{left} \times \mathcal{I}_{right} \times \mathcal{T} \times \mathcal{S}$. The image spaces, $\mathcal{I}_{left} \subseteq \mathbb{R}^{84 \times 84 \times 3}$ and $\mathcal{I}_{right} \subseteq \mathbb{R}^{84 \times 84 \times 3}$, represent 84×84 RGB wrist views from the left and right arms. The tactile space, $\mathcal{T} \subseteq \mathbb{R}^{32 \times 12}$, corresponds to a history of the last 32 force and torque readings from both arms (concatenated), while the robot state space, $\mathcal{S} \subseteq \mathbb{R}^{14}$, represents the end-effector positions and orientations (expressed as quaternions) for both arms (concatenated). Our action space, $\mathcal{A} = [0, 1]^3$,

consists of end-effector position deltas relative to the current pose.

Policy Learning: The goal is to learn a policy $\pi : \mathcal{O} \rightarrow \mathcal{A}$ that maps observations to actions, enabling task completion. In the imitation learning setting, an expert policy π^* is provided, where $a^* = \pi^*(o)$ represents the optimal action for an observation $o \in \mathcal{O}$. Our objective is to learn a policy π that closely resembles the behavior of π^* . There are several approaches to learning such a policy from demonstrations, with the simplest being *behavior cloning*. In behavior cloning, the expert provides a dataset of N demonstration trajectories $\mathcal{D} = \{\{(o_i, a_i^*)\}_{i=1}^{n_j}\}_{j=1}^N$, where n_j is the horizon for demonstration j . The policy π is then trained to replicate the expert actions from π^* for the corresponding observations using supervised learning. We train our observation encoder and policy network (shown in Figure 3) end-to-end using an L_2 loss between expert and predicted actions.

B. Data Collection with Human Experts

We collect a dataset of 50 human demonstrations in our simulation environment, built using the Robosuite framework [20] with MuJoCo [7] as the simulation engine. All demonstrations are performed in the canonical environment setup described in Section III-C. A human expert teleoperates the robot’s moving arm via keyboard inputs, with actions recorded as the difference between the end-effector positions in consecutive frames. The simulation automatically terminates and records the demonstration upon detecting a successful completion.

C. Multisensory Data Augmentation

A common technique to enhance a model’s robustness without requiring additional human effort in data collection is data augmentation, which involves transforming input data while preserving the original labels. This approach is most commonly used with image inputs, where transformations such as cropping, flipping, and color adjustments are applied to help the model learn invariance to these changes [3], [21], [22]. Image augmentation can also be performed at the semantic level using generative models [23]. However, applying this type of augmentation to contact-rich tasks poses challenges, as these tasks involve *physical* variations (e.g., object size, shape), which may introduce non-independent perturbations across the multisensory input that cannot be captured through conventional offline augmentation.

To address this, we employ an *online* data augmentation technique by replaying human-generated trajectories on task instances with identical initial offsets but with a subset of

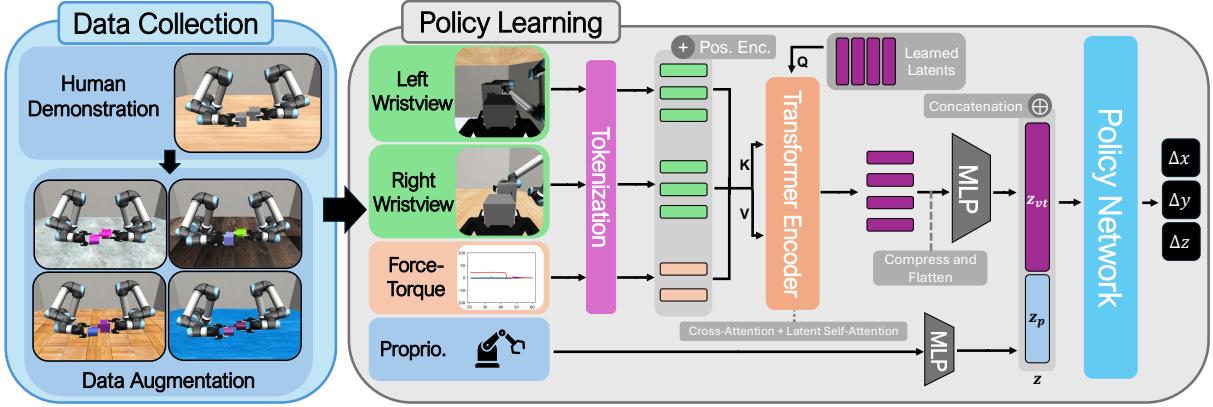


Fig. 3: An overview of our data collection and policy learning framework. We use BC-MLP [17] with a multilayer perceptron policy network to output actions. Image and force-torque observations are encoded with a visuotactile transformer [2] that includes a cross-attention step with a set of learned latent vectors (similar to Perceiver IO [18], [19]). More details on our network architecture can be found in our supplementary material and website.

task variations applied. Given the previously-defined dataset of expert demonstration trajectories \mathcal{D} , task variations \mathcal{V} (e.g. *Grasp Pose*, *Peg/Hole Shape*, etc.), and a function $f_{\mathcal{K}} : \mathcal{O} \rightarrow \mathcal{O}$ that returns an input observation with a subset $\mathcal{K} \subseteq \mathcal{V}$ of task variations applied, our online augmentation process takes an expert demonstration $d_j = \{(o_i, a_i^*)\}_{i=1}^{n_j} \in \mathcal{D}$ and outputs a set of new demonstrations $\Omega_{d_j} = \{\{(f_{\mathcal{K}}(o_i), a_i^*)\}_{i=1}^{n_j}\}_{t=1}^T$, where T is the number of augmentations per expert demonstration. The indexed functions $f_{\mathcal{K}}^1, \dots, f_{\mathcal{K}}^T$ indicate that although each application of task variations \mathcal{K} is different between each generated demonstration in Ω_{d_j} , the specific application of \mathcal{K} is consistent for each observation in a given augmented demonstration. After the augmentation process, we construct a new dataset $\hat{\mathcal{D}} = \mathcal{D} \cup \left(\bigcup_{n=1}^N \Omega_{d_n}\right)$ that can be used for training.

V. EXPERIMENTAL SETUP

We conduct experiments to evaluate the robustness of our model with respect to each of the 6 implemented task variations (see Section III-C), as well as a combination of all variations (denoted as *All Variations*).

A. Training and Evaluation Details

In all experiments, we train the models for 100 epochs and perform 50 rollouts on the same task variations used in the training dataset. The model checkpoint that achieves the highest success rate in these rollouts during training is selected for evaluation. This training process is conducted over 6 random seeds per model, and the performance is averaged across all seeds during evaluation. A rollout is considered successful if it results in a successful insertion, and it is deemed failed if the maximum horizon is exceeded without insertion. Additionally, a rollout fails if the force-torque measurement surpasses a predefined threshold, to prevent unsafe behavior that could damage the robot arms or objects.

B. Evaluation on Unseen Task Variation Instances

To ensure rigorous evaluation, we explicitly separate task variation instances encountered during training from those

used for evaluation within each variation category. For instance, when training with *Grasp Pose* variations, we include demonstrations involving x-axis translation and z-axis translation and rotation, while reserving y-axis rotation for evaluation. This approach guarantees that the model encounters unseen variations during evaluation, enabling us to assess its generalization to out-of-distribution inputs across all variation categories. A detailed overview of training versus evaluation instances for each variation category is provided in Table I.

Task Variation	Train Instances	Eval Instances
<i>Grasp Pose</i> [16]	XT, ZT, ZR	XT, ZT, ZR, YR
<i>Peg/Hole Shape</i>	key, circle, cross	arrow, u, pentagon, line, hexagon, diamond
<i>Object Body Shape</i>	<i>Peg/Hole Objects</i> : cube, cylinder	Hole Object : cube, cylinder, octagonal prism, Peg Object : thin cube, cylinder, and octagonal prism
<i>Scene Appearance</i>	6 floor textures, object color	14 unseen floor textures, object color, lighting

TABLE I: Instances for task variations during training (if included the training set) and evaluation. Task variations not in this table are the same both in training and evaluation.

VI. EXPERIMENTS AND RESULTS

In this section, we address a series of questions related to generalization through experimental evaluations.

A. Which task variations are most difficult to generalize to?

In determining the difficulty of generalizing to each of our task variations, we train a model exclusively on human-collected demonstrations without any task variations applied and report the success rate for each task variation during evaluation. These results can be found in Figure 4.

Takeaways: We observe that *Grasp Pose* variations the hardest challenge to generalize to out of all of the individual

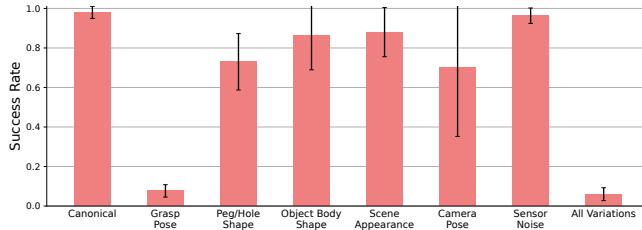


Fig. 4: Success rates on each task variation for a model trained exclusively on non-augmented human demonstration data. Error bars represent one standard deviation from the mean. The model suffers the largest success rate drop compared to the canonical environment when evaluated on *Grasp Pose* variations, and subsequently does poorly when evaluated on *All Variations*.

task variations, as we see a drop from a mean success rate of **0.987** on *Canonical* rollouts with no task variations to **0.087** when *Grasp Pose* variations are applied. We hypothesize that the large negative impact on performance comes from the significant perturbation that grasp variations apply to all sensing modalities, unlike other variations such as *Scene Appearance* and *Sensor Noise* which only target specific modalities and thus have a smaller negative impact for the overall model.

B. Which task variations included in the training set produce the largest impact on robustness?

To evaluate the effect of introducing task variations to the training dataset, we evaluate models trained on datasets augmented with different subsets of our task variations. These datasets contain the original collected human demonstrations as well as 6 augmentations per demonstration, with each augmentation containing a composition of all the task variations included in the dataset (which we refer to as the “training set variations” for that specific dataset). We evaluate these models both on instances of their training set variations that were unseen during training (as discussed in Section V-B) as well as all variations not included in the training set (which we refer to as the “evaluation set variations” for that specific dataset). In Figure 5 we report % success rate change from the canonical environment success rate averaged over 6 seeds for each variation, defined as

$$\% \text{ success rate change} = \frac{\text{task var. success} - \text{canon. success}}{\text{canon. success}}$$

Takeaways: We observe that the additions of *Peg/Hole Shape*, *Object Body Shape*, and *Grasp Pose* (considered the Base variations for this task) to the training set greatly reduce the generalization gap on unseen instances of these variations during evaluation. Curiously, we also observe a reduced generalization gap for a dataset with the Base training set variations on the evaluation set variations of *Scene Appearance* and *Camera Pose*, even though these variations had not been explicitly included in the dataset. This may be due to the similarity between the effects of applying grasp variations and perturbing the camera pose, as both alter the view of the object held by the gripper and the opposing object. Additionally, the resulting visual

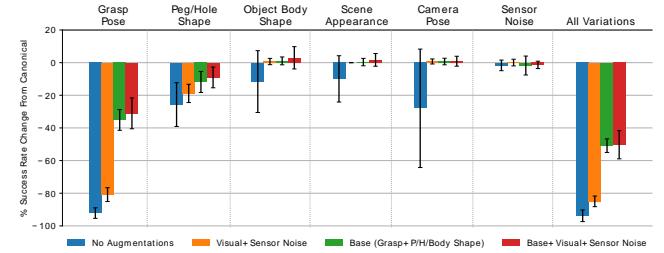


Fig. 5: % success rate changes on each task variation for models trained on different subsets of task variations. Error bars represent one standard deviation from the mean. The addition of grasp variations to the training set greatly improves generalization to unseen grasp variations, while visual variations and sensor noise do not have any significant effect on generalization ability on physical task variations.

variations may have contributed to improving the model’s robustness to changes in scene appearance. Explicitly adding both visual variations (*Scene Appearance* and *Camera Pose*) and *Sensor Noise* to the training set does not further enhance generalization to their respective variations during evaluation.

C. Can increasing the number of augmentations per demonstration improve robustness?

Building off of our investigation into determining the ideal training set variations, we also seek to analyze the effect of adding more augmentations per demonstration involving these variations. Aligning with the previous experiment, we choose *Grasp Pose*, *Peg/Hole Shape*, and *Object Body Shape* as our training set variations, and train models on datasets with different numbers of augmentations per human demonstration. Success rates on each task variation are reported in Figure 6.

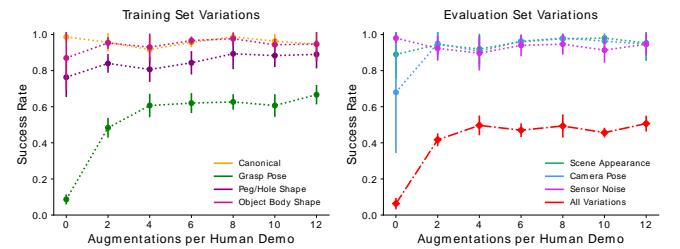


Fig. 6: Success rates on each task variation for models trained on a base set of training variations with different number of augmentations for a human demonstration. *All Variations* represents a composition of both training set and evaluation set variations. Error bars represent one standard deviation from the mean. Success rate on *Grasp Pose* evaluations increases the most with an increasing number of augmentations, while the other task variations maintain stable success rates.

Takeaways: The most significant improvement in success rate as the number of augmentations increases seems to be the performance on *Grasp Pose* evaluations, with a more subtle upward trend in the other training set variations. Since the task variations of *Peg/Hole Shape* and *Object Body Shape* are discrete variations with a small subset of all possible shapes being included in the training set, they would benefit less from having more augmentations as the dataset would start to contain redundant instances of these variations. *Grasp*

Pose variations, on the other hand, are continuous and so would benefit more from a larger sample of grasps. For the evaluation set variations (aside from *All Variations*), the success rate remains stable, suggesting that the model is not overfitting to the training set variations even when the dataset has more samples biased towards those perturbations.

D. How much does each sensory modality contribute to model robustness?

In an effort to investigate the significance of each of the modalities in our system—vision (wristview cameras), touch (force-torque), and proprioception—we conduct an ablation study with models that have one or more input modalities missing. We evaluate each model when trained on a dataset with no variations (i.e. no augmentations) and a dataset with 6 augmentations per demonstration on a training variation set of *Grasp Pose*, *Peg/Hole Shape*, and *Object Body Shape* to analyze how each modality combination reacts when task variations are introduced during training. Figure 7 shows reported % success rate change from the canonical environment success rate averaged over 6 seeds for each variation.

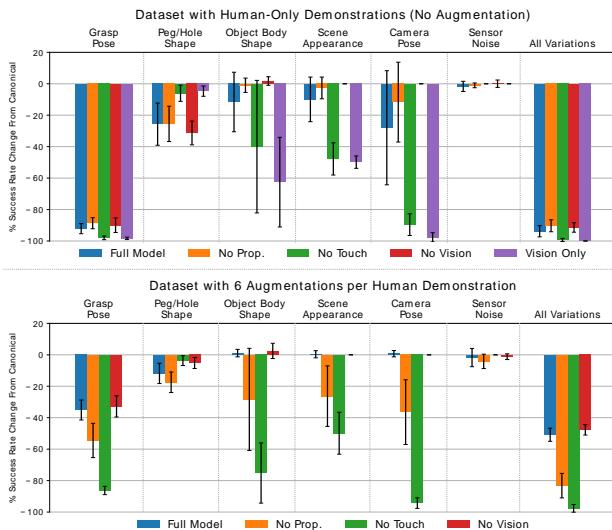


Fig. 7: % success rate changes on each task variation for models with different modality input combinations trained on no task variations (top) or a subset of task variations (bottom). The *Vision Only* model is omitted from the bottom plot due to training instability. Error bars represent one standard deviation from the mean. The removal of force-torque input sees the largest % success rate drop for many of the task variations out of each of the individual modalities, while the removal of vision has little impact on % success rate change compared to the full model.

Takeaways: We observe that out of the individual modalities, *No Touch* has the highest success rate drop for many of the variations (*Peg/Hole Shape* and *Sensor Noise* being the only exceptions) for both human-only and augmented demonstrations. On the other hand, *No Vision* has comparable (or sometimes even improved) % success rate changes to *Full Model*, suggesting its reduced significance in our overall framework compared to the other modalities. Since our task begins immediately in a contact state that is maintained

throughout a majority of the task’s duration, it follows that force-torque provides the most valuable information about the task state. Proprioception may also give important task state information (as evidenced in the % success rate change for the *No Prop.* model), as the position of the two end-effectors relative to each other is highly correlated to the position of the peg and hole relative to each other, which is essential knowledge in completing the insertion task. Thus, visual observations provide the least relevant information for our task while still being susceptible to many of the task variations. However, visual input may still be essential in task contexts outside of the one studied here, especially in situations with little to no force-torque feedback (such as aligning the peg and hole objects to be in the same orientation before contact as was studied in our previous work [16]).

E. Additional Experiments

We perform a set of additional experiments to gain more insight into our system, the full results of which can be found in our supplementary material and website.

Attention Visualization: We plot attention weights in the cross-attention step of the visuotactile encoder averaged over the learned latent vectors and find that tokens from the tactile input take up a much larger proportion of attention than the visual tokens throughout the task, despite there being twice as many visual tokens as tactile ones. This provides further evidence of the importance of tactile over visual information first discussed in Section VI-D.

Real World Experiments: We construct a real-world setup of our insertion task and conduct a smaller-scale robustness experiment to compare with our results gathered in simulation.

VII. CONCLUSION

We present a pipeline for data collection, augmentation, policy training, and evaluation to learn robust policies for an object assembly task across diverse observation-level task variations. Our experiments reveal that grasp variations pose the greatest challenge for generalization, and incorporating them through data augmentation significantly improves performance on unseen variations. Additionally, we demonstrate that force-torque input is critical for robustness to task variations, while the removal of RGB input has minimal impact.

While we have demonstrated the ability of our system to learn the underlying task, we acknowledge that the behavior cloning setup used is highly susceptible to covariate shift and cannot recover from erroneous actions. We plan to extend our generalization studies using more advanced imitation learning frameworks, such as Diffusion Policy [24] and ACT [25], and compare their performance with our BC-MLP setup. Moreover, the constrained task initialization and action space in our setup highlight the need to explore more complex, longer-horizon tasks with broader action spaces, and to assess how observation-level task variations affect policies in these contexts.

REFERENCES

- [1] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, “Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks,” in *2019 International conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 8943–8950.
- [2] Y. Chen, M. Van der Merwe, A. Sipos, and N. Fazeli, “Visuo-tactile transformers for manipulation,” in *Conference on Robot Learning*. PMLR, 2023, pp. 2026–2040.
- [3] O. Spector and D. Di Castro, “Insertionnet-a scalable solution for insertion,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5509–5516, 2021.
- [4] O. Spector, V. Tchuiiev, and D. Di Castro, “Insertionnet 2.0: Minimal contact multi-step insertion using multimodal multiview sensory input,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6330–6336.
- [5] A. Xie, L. Lee, T. Xiao, and C. Finn, “Decomposing the generalization gap in imitation learning for visual robotic manipulation,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 3153–3160.
- [6] W. Pumacay, I. Singh, J. Duan, R. Krishna, J. Thomason, and D. Fox, “The colosseum: A benchmark for evaluating generalization for robotic manipulation,” *arXiv preprint arXiv:2402.08191*, 2024.
- [7] E. Todorov, T. Erez, and Y. Tassa, “MuJoCo: A physics engine for model-based control,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033.
- [8] M. A. Lee, M. Tan, Y. Zhu, and J. Bohg, “Detect, reject, correct: Crossmodal compensation of corrupted sensors,” in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 909–916.
- [9] A. Vaswani, “Attention is all you need,” *Advances in Neural Information Processing Systems*, 2017.
- [10] C. Kohler, A. S. Srikanth, E. Arora, and R. Platt, “Symmetric models for visual force policy learning,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 3101–3107.
- [11] 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. [Online]. Available: <https://openreview.net/forum?id=DdgIKo8hBq0>
- [12] J. Gao, A. Xie, T. Xiao, C. Finn, and D. Sadigh, “Efficient data collection for robotic manipulation via compositional generalization,” *arXiv preprint arXiv:2403.05110*, 2024.
- [13] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox, “Mimicgen: A data generation system for scalable robot learning using human demonstrations,” in *Conference on Robot Learning*. PMLR, 2023, pp. 1820–1864.
- [14] M. Jia, D. Wang, G. Su, D. Klee, X. Zhu, R. Walters, and R. Platt, “Seil: simulation-augmented equivariant imitation learning,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1845–1851.
- [15] L. Ankile, A. Simeonov, I. Shenfeld, and P. Agrawal, “Juicer: Data-efficient imitation learning for robotic assembly,” *arXiv preprint arXiv:2404.03729*, 2024.
- [16] C. Ku, C. Winge, R. Diaz, W. Yuan, and K. Desingh, “Evaluating robustness of visual representations for object assembly task requiring spatio-geometrical reasoning,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 831–837.
- [17] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, “What matters in learning from offline human demonstrations for robot manipulation,” in *Conference on Robot Learning*. PMLR, 2022, pp. 1678–1690.
- [18] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira, “Perceiver: General perception with iterative attention,” in *International conference on machine learning*. PMLR, 2021, pp. 4651–4664.
- [19] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer *et al.*, “Perceiver io: A general architecture for structured inputs & outputs,” in *International Conference on Learning Representations*, 2022.
- [20] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu, “robosuite: A modular simulation framework and benchmark for robot learning,” *arXiv preprint arXiv:2009.12293*, 2020.
- [21] L. Perez, “The effectiveness of data augmentation in image classification using deep learning,” *arXiv preprint arXiv:1712.04621*, 2017.
- [22] X. Zhu, D. Wang, O. Biza, G. Su, R. Walters, and R. Platt, “Sample efficient grasp learning using equivariant models,” in *Robotics: Science and Systems*, 2022.
- [23] Z. Chen, S. Kiami, A. Gupta, and V. Kumar, “Genaug: Retargeting behaviors to unseen situations via generative augmentation,” *arXiv preprint arXiv:2302.06671*, 2023.
- [24] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” *The International Journal of Robotics Research*, 2024.
- [25] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, “Learning fine-grained bimanual manipulation with low-cost hardware,” *arXiv preprint arXiv:2304.13705*, 2023.
- [26] M. Shridhar, L. Manuelli, and D. Fox, “Perceiver-actor: A multi-task transformer for robotic manipulation,” in *Conference on Robot Learning*. PMLR, 2023, pp. 785–799.
- [27] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *International Conference on Learning Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=YicbFdNTTy>

APPENDIX

A. Task Variation Details

In this section, we provide more explicit details on the formulation and implementation of our task variations in simulation.

- 1) **Peg and Hole Shape:** The 9 possible peg and hole shapes are key, cross, diamond, line, circle, pentagon, hexagon, key, and u. The intrusions are built with a 5mm uniform tolerance on all sides.
- 2) **Object Body Shape:** The 3 possible object body shapes are cube (7.6cm on all sides), cylinder (7.6cm height and diameter), and octagonal prism (7.6cm height and diameter). The thin versions of these objects for the peg shape have 60% the width of the full-sized shapes.
- 3) **Grasp Pose:** X-axis translation variations lie within a distance of [-1.7cm, 1.7cm] from the center of the object face (these offsets are scaled down to 60% for thin object body shapes). Z-axis translation variations lie within a distance of [0.0cm, 1.4cm] from the center of the object face. Y-axis rotation variations range from [-10.0°, 10.0°], and Z-axis rotation variations are sampled from {0°, 90°, 270°} (180° was left out due to simulation instability).
- 4) **Scene Appearance:** Lighting variations involve both lighting color and the status of the light (either on or off). Object colors are sampled uniformly from all possible colors in the Robosuite [20] simulation, and floor textures were sampled from a discrete set of 20 possible textures (6 for training and 14 for evaluation).
- 5) **Camera Pose:** Camera position is perturbed in the range of [-4cm, 4cm] on the x, y, and z axes, and camera rotation is perturbed by sampling a random axis-angle rotation (with the axis sampled from a normalized spherical Gaussian and the angle sampled from [0°, 10°]) and applying it to the camera’s base orientation.
- 6) **Sensor Noise:** For force-torque readings, zero-mean Gaussian noise is applied to forces and torques separately (with a standard deviation of 5N for force and 0.15N·m for torque). For proprioceptive readings, zero-mean Gaussian noise is applied to end-effector position and orientation separately (with a standard deviation of 0.1cm for position and 0.57° for orientation).

B. Model Architecture

In this section, we provide further details for our observation encoder and policy network architectures, an extended version of which can be found in Figure 8.

To encode our observations, we draw upon the success of visuotactile transformer encoders [2] and utilize a similar attention-based mechanism for RGB and tactile modality fusion. Rather than performing self-attention directly with the input tokens, we found that introducing a cross-attention step similar to the PerceiverIO [18], [19] architecture seemed to work best for our task. The PerceiverIO module has been used in other robotic manipulation contexts [26] to encode high-dimensional data while still maintaining a high-quality representation.

We tokenize our inputs by computing linear projections of both visual patches [27] for RGB inputs and individual readings per timestep for the force-torque input, and then add modality-specific position encodings. We then cross-attend these input tokens with a set of 8 learned latent vectors that then travel through a series of self-attention layers before ultimately being compressed and projected (as in [2]) to an output latent embedding $z_{vt} \in \mathbb{R}^{288}$. We encode proprioception with a multilayer perceptron to get an output embedding $z_p \in \mathbb{R}^{32}$ and concatenate z_{vt} and z_p to get $z \in \mathbb{R}^{320}$, which is used as input to the policy network. The policy network is then a multilayer perceptron that outputs 3-dimensional actions $a \in \mathcal{A}$ that represent end-effector position deltas for one of the arms.

C. Visuotactile Attention Visualization

To gain further insight into the information being learned by our model, we visualize the attention weights in the latent vector cross-attention step of the transformer visuotactile encoder. For each modality, we plot attention weights as proportions of total attention to tokens in that specific modality averaged over the 8 learned latent vectors. Figure 9 shows an example visualization; more examples can be found on our website.

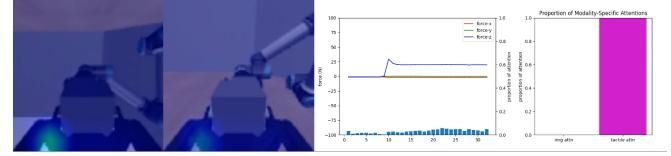


Fig. 9: A visualization of attention weights for vision and tactile tokens in the latent cross-attention step of the visuotactile transformer encoder. These weights are visualized as heatmaps overlayed on left and right wristview images for visual attention, and bars for each timestep under the force reading for tactile attention. We also plot the proportion of total attention for each modality (visual and tactile) during the course of a rollout.

Takeaways: Despite our model taking in twice as many visual tokens (72 tokens, 36 per view) as tactile ones (32 tokens), we observe that tactile attention takes up almost the entire proportion of attention across the input (as seen in the right-most plot of Figure 9). This finding provides further evidence to the importance of tactile information over visual information as discussed in Section VI-D, where we found that removing visual information from our input had little impact on the robustness of our model. Furthermore, we observe that the visual attention is mostly focused on semantically insignificant parts of the view, such as the gripper at the bottom of the view, suggesting that the model is not receiving much useful visual information.

D. Comparing Data Augmentation Methods

In an effort to evaluate the validity of the online augmentation method for increasing the robustness of our model, we construct a dataset of human-generated trajectories with an extended set of visual variations and sensor noise, attempting to emulate a baseline data augmentation method that applies augmentations independently to each sensory modality offline during training. More specifically, we generate a dataset

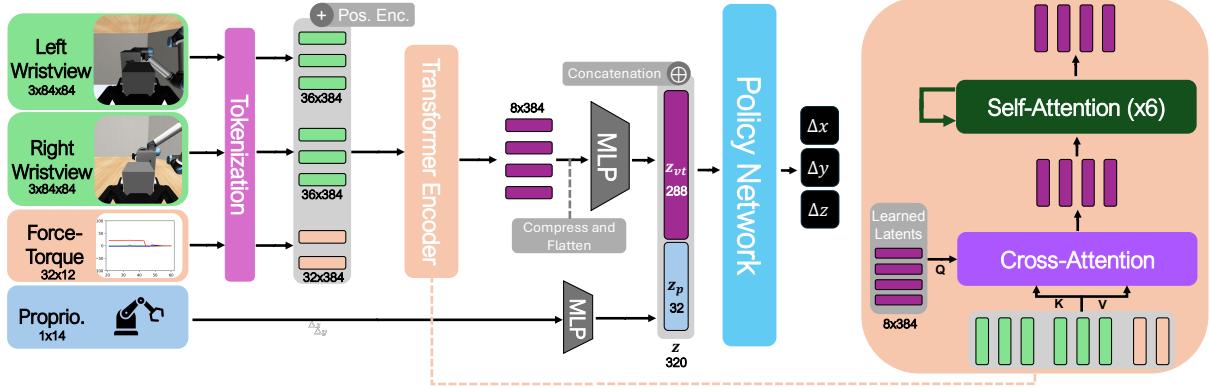


Fig. 8: An extended view of our network architecture with added component dimensions and a visualization of our visuotactile transformer encoder.

with training set variations of *Scene Appearance* (including object color, floor texture, and lighting), *Camera Pose*, and *Sensor Noise* with 12 augmentations per demonstration, but rather than keep applied variations consistent through each augmented rollout, we apply new instances of *Scene Appearance* and *Camera Pose* variations in each step of the demonstration. We also multiply the force and torque history reading by a random constant (from 0.1 to 2.0) independently determined each frame, following a similar data augmentation strategy used in [3]. We denote this dataset as *Expanded Visual+Noise*.

We report % success rate change from the *Canonical* environment success rate on models trained on the *Expanded Visual+Noise* dataset and compare it with the training set models from our original experiment (namely *Visual+Sensor Noise* that does not apply new variation instances per frame and *Base* that includes *Grasp Pose* variations); results can be found in Figure 10.

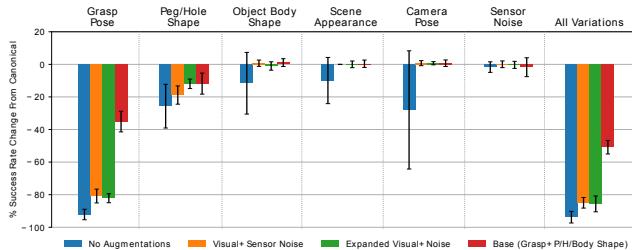


Fig. 10: % success rate changes on each task variation for models trained on different subsets of task variations, including an additional *Extended Visual+Noise* dataset. Error bars represent one standard deviation from the mean. Our extended dataset of *Scene Appearance*, *Camera Angle*, and *Sensor Noise* variations does not lead to any significant improvements on robustness to any of the task variations over the original *Visual+Sensor Noise* dataset, particularly on *Grasp Pose* variations, where the inclusion of *Grasp Pose* in the training dataset via online augmentation still provides the largest boost in generalization ability.

Takeaways: We observe that our dataset with an expanded set of augmentations independently applied to each sensory modality does not necessarily improve robustness on most task variations (save for *Peg/Hole Shape*) compared to the original *Visual+Sensor Noise* dataset that was less

extensive in terms of data augmentation. Most crucially, we do not see a significant improvement on *Grasp Pose* variations, validating the effect of non-independent multisensory data augmentation via trajectory replay. Thus, we have shown that even extensive independent augmentation of our multisensory input may not be enough to deal with certain task variations involved in our contact-rich task.

E. Success Rates for Canonical Environment

For full transparency for our experiments that involve reporting the % success rate change from the *Canonical* environment, we explicitly report the success rates in the no-variation *Canonical* environment, which the % success rate change is based off of, for each trained model. Average success rates over 6 training seeds are reported in Figures 11 and 12. It is worth noting that the average % success rate change across the 6 training seeds was calculated by determining the % success rate change for each individual seed and then calculating the average over those values, rather than calculating the average success rate across the 6 seeds first and then determining the difference of those averages.

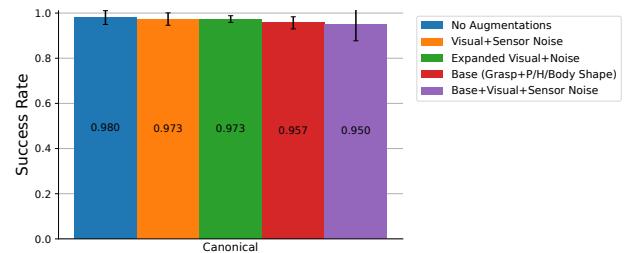


Fig. 11: Success rates on the *Canonical* environment for models trained on different training set variations. Error bars represent one standard deviation from the mean. These success rates correspond to the results reported in Figures 5 and 10.

F. Real-World Experimental Setup

Our real-world task setup, shown in Figure 13, is built to mirror our simulation setup as closely as possible. We designate one arm to be compliant, applying a constant

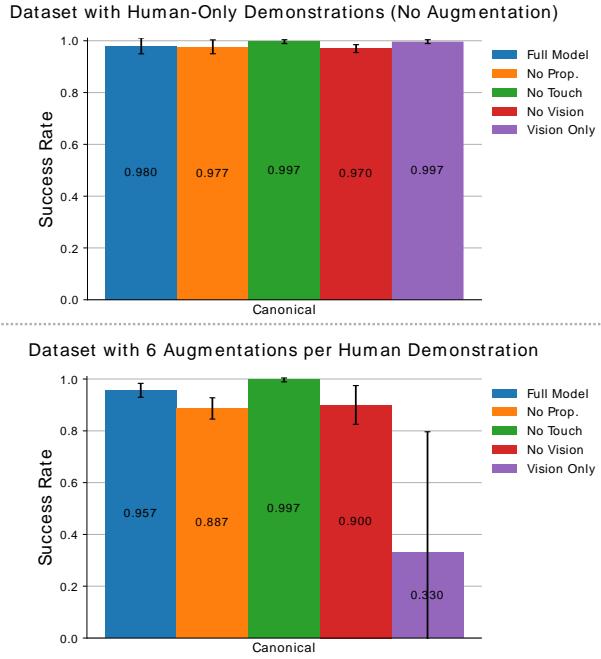


Fig. 12: Success rates on the *Canonical* environment for models using different modality combinations as input, trained on both human-only demonstration data as well as demonstration data augmented with *Grasp Pose*, *Peg/Hole Shape*, and *Object Body Shape* variations (denoted as the Base training set variations). Error bars represent one standard deviation from the mean. These success rates correspond to the results reported in Figure 7. We especially note the performance instability of the *No Vision* model trained on the Base training set variations, providing context for its omission in Figure 7.

amount of force while the other arm moves according to the actions given to it by the policy. In contrast to policies trained in simulation, our real-world policies predict 2-dimensional delta actions in the axes perpendicular to the axis of insertion (rather than 3-dimensional actions that include the axis along the direction of insertion), in order to prevent potentially unsafe interactions that may occur as a result of a premature insertion attempt. Once the peg and hole are aligned, the compliant arm automatically moves its held object forward to complete the insertion.

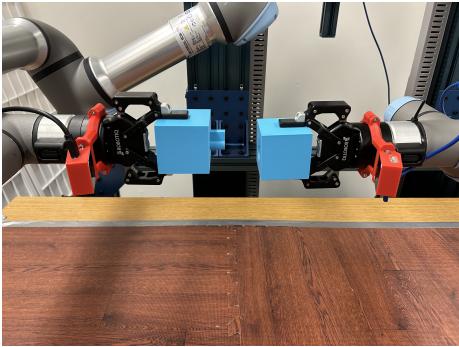


Fig. 13: Our real-world experimental setup. Peg and hole object models are 3D-printed directly from the model files used in our simulation environment.

We train our real-world models with the same hyperparameters as those in simulation, although we only initiate

1 training seed per model (rather than 6). Additionally, we evaluate each model at the end of the entire training process, rather than performing training set rollouts during the training process to determine the best checkpoint. Success rates are determined by the number of successful trials over 20 rollouts. Successes and failures follow the same general criteria as in simulation, though a human manually annotates successes and failures per trial.

G. Real-World Experiments and Results

Task Variation Difficulty: As a real-world analog to our experiments in Section VI-A, we evaluate a real-world policy trained on a dataset of human-generated demonstrations with no applied task variations on real-world versions of a subset of our task variations. Reported success rates over 20 rollouts can be found in Figure 14.

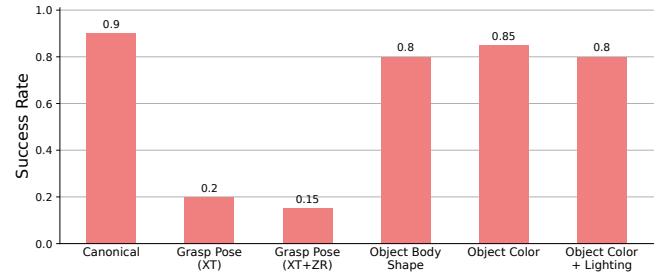


Fig. 14: Success rates on real-world task variations for a model trained on real-world non-augmented human demonstration data. The real-world model struggles the most with *Grasp Pose* (both translation-only and translation with rotation) variations, aligning with our observations in simulation.

Takeaways: Like in simulation, we observe that *Grasp Pose* variations seem to be the most difficult to generalize to, while the model is able to handle the mostly unisensory perturbations of *Object Body Shape* and *Scene Appearance* (object color and lighting). We also notice that our model struggles with *Grasp Pose* even when rotational grasp variations are removed; we hypothesize that this may be because a translational offset disrupts the desired behavior of lining up end-effector positions (given from proprioceptive input) in order to line up the peg and hole (i.e. solving the task can no longer be done by just matching the end-effector positions of the two arms). From these results, We believe that including *Grasp Pose* variations into the training dataset (as was done in simulation through online augmentation) may also improve performance in the real world.

Modality Input Ablation Study: We conduct a reduced real-world analog to the ablation study in Section VI-D. We train real-world policies on a dataset of only human demonstrations and evaluate them on a smaller subset of our real-world task variations. Reported success rates over 20 rollouts can be found in Figure 15.

Takeaways: Like in simulation, we observe that the removal of force-torque data as input (the *No Touch* model) leads to a significant drop in success rate for all task variations compared to the *Full Model*, including the no-variations *Canonical* environment. We also see a small drop in performance for the *No Vision* model, somewhat aligning with

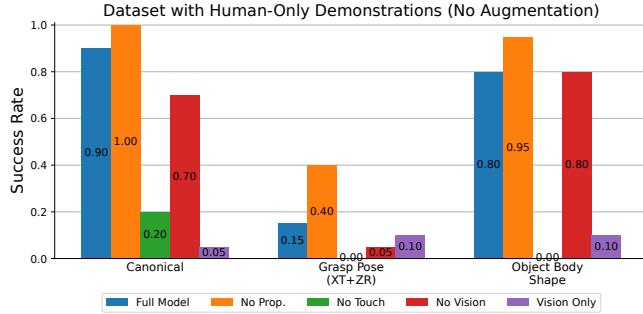


Fig. 15: Success rates on real-world task variations for models with different modality input combinations trained on no variations. The No Touch model sees the lowest success rates on all task variations, while the No Prop. model surprisingly sees an increase in performance over the Full Model for all task variations.

our findings in simulation of the insignificance of visual input for our task. Surprisingly, we see performance increases in all task variations for the No Prop. model. We hypothesize that the small ranges of possible end-effector poses in our training dataset due to the high precision required for our task may cause our models to not learn much useful information from the proprioceptive embedding, though this observation may of course also be the result of a low sample size of trained models. Averaging the performance of models trained over multiple seeds (as was done in simulation), which was not able to be performed due to time constraints, may give us some more robust results.