Generalization and Efficiency in Robot Manipulation via Semantic Augmentations and Action Chunking

Homanga Bharadhwaj[*,1,2] **Jay Vakil**[*,2] **Mohit Sharma**[*,1]

Abhinav Gupta[1] **Shubham Tulsiani**[1] **Vikash Kumar**[1]

[1] **The Robotics Institute, Carnegie Mellon University**

[2] **FAIR, Meta AI**

`hbharadh|mohits1@cs.cmu.edu`

Figure 1: Glimpse of a subset of the diverse manipulation capabilities of MT-ACT . We train a single agent capable of 12 manipulation skills across 30 tasks encompassing 6 activities. Videos of evaluations are on the website https://robopen.github.io/

Abstract: The grand aim of having a single robot that can manipulate arbitrary objects in diverse settings is at odds with the paucity of robotics datasets. Acquiring and growing such datasets is strenuous due to manual efforts, operational costs, and safety challenges. A path toward such an universal agent would require a structured framework capable of wide generalization but trained within a reasonable data budget. In this paper, we develop an efficient system (MT-ACT) for training universal agents capable of multi-task manipulation skills using (a) *semantic augmentations* that can rapidly multiply existing datasets and (b) *action representations* that can extract performant policies with small yet diverse multi-modal datasets without overfitting. In addition, reliable task conditioning and an

Preprint. Under review. * authors contributed equally

expressive policy architecture enables our agent to exhibit a diverse repertoire of skills in novel situations specified using language commands. Using merely 7500 demonstrations, we are able to train a single agent capable of 12 unique skills, and demonstrate its generalization over 30 tasks spread across common daily activities in diverse kitchen scenes. On average, MT-ACT outperforms prior methods by over 40% in unseen situations while being more sample efficient and being amenable to improved deployment performance through fine-tuning.

1 Introduction

Training a robot manipulator with multiple skills requires exposure to diverse experiences and the ability to acquire skills from a diverse data corpus. Collection of such a multi-skill data corpus in the real-world requires substantial effort and suffers from high operational cost and safety challenges. Given the expense, efficiency in robot learning paradigms is necessary for real world training and deployment. While there are recent efforts in scaling robotic datasets despite these challenges [1], efficiency seems to be overlooked in the attempts to scale.

With the acknowledgment that robot learning will generally benefit as the scale of the robotics dataset grows, the focus of our work is on investigating generalization in developing capable agents under a *given data budget*. We restrict ourselves to a dataset with 7,500 robot manipulation trajectories (an order of magnitude less than related works [1]) containing a diverse collection of manipulation skills across different tasks. As a robot under deployment in real environments like homes, hospitals, etc., will always find itself in unseen scenarios, we set out to develop the most capable agent with an emphasis on its ability to generalize to novel situations within this data budget.

At first sight, wide generalization with a data budget seems like wishful thinking - while it's possible to provide large representation capabilities to the agent's policy, scaling without data diversity will likely lead to overfitting and no generalization. Our insight is twofold: we ensure sufficient coverage of different skills in different scenarios in a dataset we collect through teleoperation; we enable generalization by augmenting the dataset with semantic variations in the objects [2, 3, 4] and training a language-conditioned manipulation policy with multi-task action-chunking transformers capable of handling the multi-modal data distribution. The architecture leverages the fact that robot movements are temporally correlated, by predicting action chunks [5] instead of per-step actions, leading to smoother behaviors and mitigating covariate shift commonly observed in the low data imitation learning regime.

Overall, we emphasize that the data efficiency lessons we present are *general* and will help in achieving generalizable agents independent of the available data budget. Building on these insights, we make the following contributions:

- We present an efficient method MT-ACT designed to recover **generalist agents on a data budget**. MT-ACT leverages data multiplication via semantic augmentations and action representations to drive efficiency gains in low-data settings.
- MT-ACT 's architecture can effectively ingest multi-modal trajectory data to recover a single policy that can perform a diverse set of tasks through language instructions. Through extensive real-world experiments, we show our agent is **capable of exhibiting 12 manipulation skills**.
- We perform extensive generalization studies to demonstrate that MT-ACT is 40 % more performant than alternatives, exhibits much **superior generalization to diverse novel scenarios**, and can be **improved during deployment through fine-tuning**.
- We meticulously recorded all the data collected during the course of the project which we are open-sourcing as part of `RoboSet`- one of the **largest open-source robotics datasets** on commodity hardware. It contains high-quality human teleOp trajectories spanning a balanced distribution of 12 skills across 38 tasks in diverse kitchen scenes.

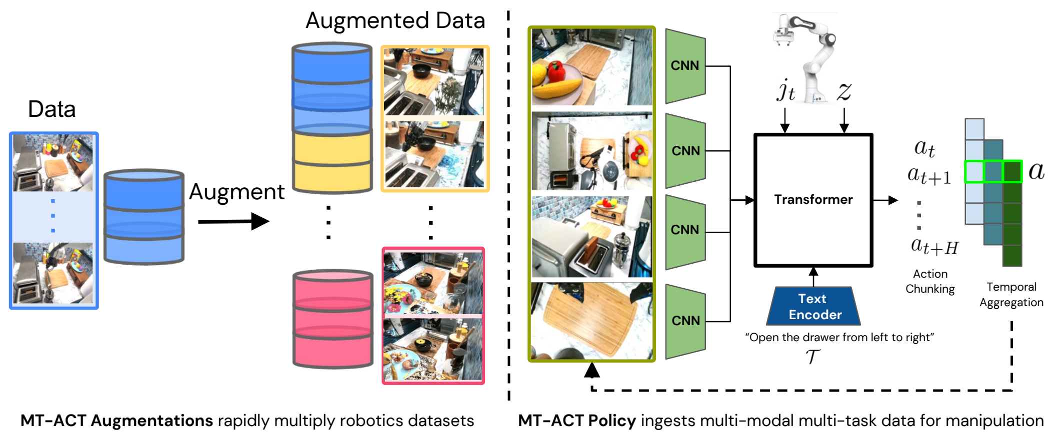


Figure 2: Summary of the overall framework MT-ACT showing the two main stages: semantic augmentation for multiplying data, and leveraging efficient action representations for ingesting multi-modal multi-task data into a single multi-skill multi-task policy.

2 Related Work

Frameworks for Scaling Robot Learning. Given the cost of supervision in robot learning, self-supervised learning [6, 7, 8] methods leveraging large unlabeled datasets have been a dominant paradigm in efforts towards building general-purpose agents. Large-scale simulations [9, 10, 11, 12] have also been leveraged with the hope of learning a general multi-task policy for diverse tasks [13, 14, 15, 16, 17, 18] first and then transferring it to the real world via sim2real[19]. However, many existing multi-task RL works focus on narrow domains in simulation [16, 20], and those in the real-world show limited generalization and task diversity [21, 3]. While other works [13, 14, 22] focus on multi-task settings in diverse scenarios, they restrict to evaluating trained policies mostly in simulation. By contrast, our work focuses on a large, diverse set of real-world manipulation tasks. Recently, many works have used imitation learning with large-scale real-world robot interaction datasets [23, 24, 25, 26, 27]. While early works collect limited real-world data [24, 27], more recent approaches [26, 1] collect much larger datasets. In fact, [1] gathers, possibly, the largest dataset ($\approx 130K$ demonstrations) and shows impressive generalization with skills learned using this data. Our work is similar in spirit, *i.e.*, we focus on real-world manipulation tasks and aim to learn a multi-task policy using *limited* real-world demonstrations. However, unlike [26], we avoid toy environment setups and focus on realistic real-world kitchen setups with clutter and multiple feasible tasks in a scene. Additionally, our agents exhibit more skills than [1] while being trained on 7.5k trajectories, as opposed to 135k in [1]. Importantly, we collect our data with commodity hardware (see Figure 6) and are making it readily available to robotic researchers worldwide.

Alternate Data Sources in Robotics. Recent successes of large-scale self-supervised approaches within both language and vision communities have showcased the advantage of large-scale data. Many recent works propose using pre-trained visual representations trained primarily on non-robot datasets [28, 29], for learning control policies [30, 31, 32, 33, 34]. Most of these works focus on single-task settings [30, 31, 35, 36], or in simulated robot environments [32, 33]. Given the inherently large cost of collecting real-world robotics datasets, many works have focused on using alternate data sources such as language [37, 38, 39, 40], human videos [41, 42, 43, 44, 45, 46], and generative augmentations [47, 48, 3, 2, 4]. Our work is most similar to the latter set of works, some of which use diffusion models to generate augmentations for data collected in the real world. However, unlike some prior works [3, 2] we do not manually provide segmentation masks [3] or object meshes [2] for generating augmentation data. Overall, our work is most similar to [4] which adapts a pre-trained open-world object detection model [49] for generating segmentations that are used with text-guided diffusion models to generate augmentations. While our approach is broadly similar to [4], we do not require any further fine-tuning of a separate module for open-vocabulary

Table 1: Open-source real-world manipulation dataset landscape: `RoboSet` is of the largest open-source robotics datasets. It contains high-quality human teleOp trajectories spanning a balanced distribution of 12 skills across 38 tasks in diverse kitchen scenes.

	Trajectories	**Tasks**	**Skills**	**Scenes**	**Source**
`RoboSet` **(MT-ACT)**	7,500	30	12	10	TeleOp
`RoboSet` **(full)**	30,050	38	12	10	TeleOp
BridgeData [26]	33,200	72	8	10	TeleOp
BC-Z [51]	25,000	100	9	N/A	TeleOp
RoboTurk [24]	2,100	N/A	3	1	TeleOp
Amazon Pick-Place [52]	100,000	N/A	1	1	Heuristics
RoboNet [23]	162,000	N/A	2	7	Heuristics
BAIR Pushing [53]	N/A	N/A	1	1	Heuristics

segmentation and language grounding. More importantly, we further investigate scaling laws with respect to semantic data augmentations.

3 Augmented Multi-Task Action Chunking Transformer

To learn generalizable manipulation policies, it is essential for robots to be exposed to rich and diverse experiences, encompassing a wide range of skills and contextual variations. However, collecting such extensive and diverse manipulation data poses practical challenges. Therefore, in order to address this limitation and learn effective manipulation policies with limited data, our focus is twofold. First, we design efficient structured augmentations that generate semantically diverse data from limited samples. These augmentations modify either the object being interacted with or the scene background, thereby creating multiple demonstrations with distinct semantic contexts, at no extra robot or human cost. This approach incorporates real-world semantic priors into the multi-task manipulation agents, accounting for out-of-distribution scenarios during deployment. Second, we explore techniques to learn robust skills from limited skill data, adapting design choices from previous limited single-task settings to the context of large-scale generalization in multi-task multi-scene manipulation tasks with diverse skills. To model the diverse multi-modal multi-task augmented datasets, we employ a Conditional Variational Autoencoder (CVAE) [50] to identify action distribution modes. This enables us to fit a high-capacity Transformer conditioned on the CVAE encodings, effectively capturing the variations and dependencies in the augmented dataset.

3.1 Dataset

For training a generally capable multi-task agent that can handle diverse skills in widely varying scenes, while being robust to distractors we need datasets with sufficient coverage and diversity. In order to ensure diverse behaviors, we try to realize sufficient coverage over different core skills. To achieve this we instantiate each task in different kitchen scenes, visually illustrated in Appendix A. Instead of having a random set of several tasks, we structure groups of tasks as belonging to a part of an activity, such that they can be executed in sequence to obtain a meaningful outcome, such as cleaning a kitchen. In addition, we collect all our trajectories with four cameras views to ensure robustness to occlusion and scene variations.

The dataset we used for this project consists of 7,500 trajectories (from `RoboSet` Table 1) involving 12 skills. Figure 3 shows a distribution of skills over all demonstrations. While the commonly used pick-place skills cover 40% of the dataset, we also include contact rich skills such as (*Wipe*, *Cap*) as well as skills involving articulated objects (*Flap-Open*, *Flap-Close*). We collect the overall dataset across 4 different kitchen scene instantiations with various everyday objects in the scenes. Thus, each skills uses many variations of certain target objects. Figure 6 (Right) provides a glimpse of some of the objects used. Finally, we define each task with a language command, and each

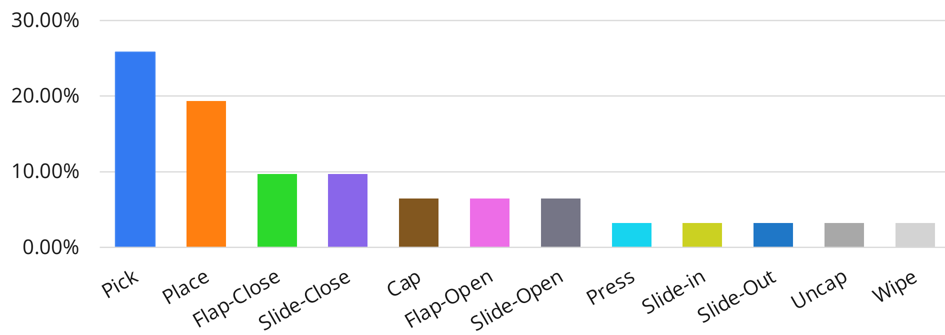
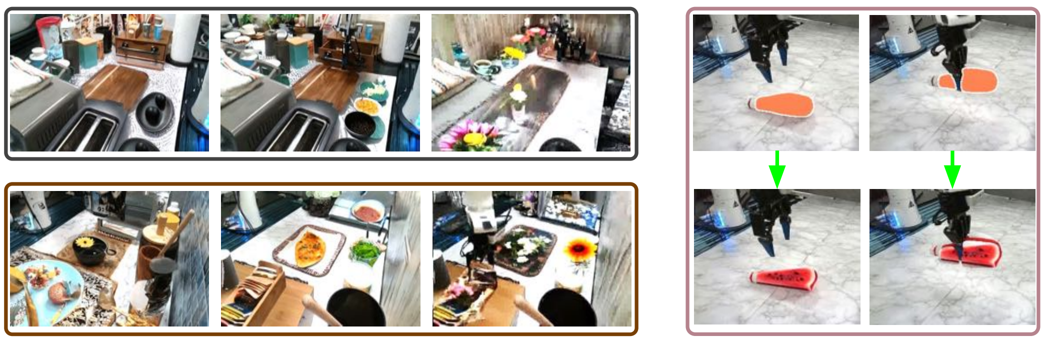


Figure 3: Skill distribution in terms of % of trajectories with a certain skill in the dataset. We note the wide coverage across 12 different skills, enabling diverse manipulation behaviors.



(a) Augmenting Background (b) Augmenting Interaction Object

Figure 4: Illustration of the data augmentations that we develop to rapidly multiply limited robot datasets with diverse semantic scene variations. In (a) we show the scene around the robot and the interaction object changing. In (b), we show the interaction object itself changing while preserving the rest of the scene.

trajectory consists of 4 different camera views - three static cameras and one wrist camera, and robot proprioception (Figure 6 Left).

We also compare our dataset with existing *open-source* robot manipulation datasets in Table 1. Compared to prior open-source datasets our dataset includes a larger number of skill and scene variations and is the largest dataset with commodity hardware in real-world setup. Finally, despite our data diversity, our dataset is still much smaller in size in comparison to other recent papers that use *proprietary* robot datasets, e.g. RT1 which has 135K trajectories [1].

3.2 Data Augmentation

Generally useful robot manipulation systems will need to be able to deal with out-of-distribution scenarios (e.g. different homes and offices). Since the dataset we collect cannot have the diversity of scenes and objects to the extent we would need for generalization (due to physical access constraints), we seek to multiply the dataset by augmenting scenes with diverse variations offline while preserving the manipulation behavior in each trajectory. Based on recent advances in segmentation and inpainting models [54, 55], we can distill semantic real-world priors from internet data, to modify a scene in a structured manner.

Given an initial robot dataset consisting of different trajectories, we augment each trajectory per frame, to produce new trajectories that preserve the *robot behavior* in the trajectories which makes several semantic variations in the scene, per frame. To enable this, we inpaint a part of an image specified by a mask to a different image introducing objects in the masked region informed by a text prompt. As opposed to [3, 2, 4] needing manual masks, object templates, etc., our approach is fully automatic. We use the SegmentAnything model [54] to automatically detect objects in the scene to be augmented. We apply augmentations separately to the object and the rest of the environment respecting the object and robot boundaries. See Appendix A.1 for additional details.

3.3 MT-ACT Architecture

In order to develop generalizable robot manipulation policies within real-world constraints of collecting limited data, we need to design efficient policy architectures. In scenarios that have sufficient coverage within the training data, we want the policies to stay close to nominal behaviors (efficient imitation), and also want to generalize to new contexts that are unseen during training (efficient task conditioning). In addition, we want the policies to exhibit temporally correlated behaviors that accomplish a task with minimal compounding errors, and incorporate action-chunking [5] to achieve this.

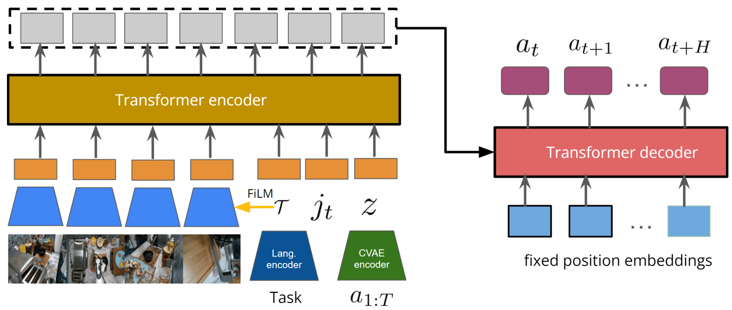


Figure 5: Policy architecture for MT-ACT . We use a CVAE that learns latent encodings for action sequences to implicitly identify different *modes* in the data. A transformer takes as input a latent code, language embedding of the task, and embeddings from four camera views, to output an action sequence.

The policy architecture for MT-ACT is designed to be a Transformer model of sufficient capacity that can handle multi-modal multi-task robot datasets. In order to capture multi-modal data, following prior works [5] we incorporate a CVAE that encodes action sequences into latent *style* embeddings z. The decoder of the CVAE is the Transformer policy that conditions on latents z. This formulation of expressing the policy as a generative model helps in effectively fitting to the multi-modal teleop data, without ignoring regions of a trajectory crucial for precision, which are also likely to be more stochastic. In order to model multi-task data, we incorporate a pre-trained language encoder that learns an embedding $\mathcal{T}$ of a particular task description. To mitigate issues of compounding error, at each time-step, we predict actions H steps in the future and execute them through temporal-smoothing of overlapping actions predicted for a particular time-step [5]. For improving robustness to scene variations, we provide the policy with four different views of the workspace through four cameras.

The transformer encoder takes as input four observation views at the current time-step, $o_t^{1:4}$, the current joint pose of the robot j_t, the style embedding from the CVAE z, and the language embedding $\mathcal{T}$. We use a FiLM based conditioning [56, 1], in order to ensure that the image tokens are able to reliably focus on the language instruction, such that the policy doesn't get confused about the task when multiple tasks are possible in a scene. The encoded tokens go to the decoder of the Transformer policy with fixed position embeddings, which finally outputs the next action chunk (H actions) from the current time-step. For execution, we average over all overlapping actions predicted for a current time-step (As $H > 1$, the action chunks overlap), and execute the resulting averaged action.

4 Experimental Design

Through experiments, we want to understand the following research questions

- How does MT-ACT perform, quantitatively and qualitatively, on a large set of vision-based robotic manipulation tasks? How does it generalize to new tasks, objects, and environments?
- Does data augmentation improve robustness to noise/distractors?
- Does data augmentation improve policy generalization (i.e. scenes with new target objects)?
- Does the policy architecture of MT-ACT enable efficient learning with high performance?
- Does action chunking help with temporally consistent trajectories, achieving higher success?

To answer these research questions we instantiate our framework in the real world using commodity hardware and objects commonly used in everyday kitchens. Next, we outline the system and dataset used to investigate our questions and then describe the different generalization axes for evaluation.

Robot hardware. Figure 6 shows our robot environment that consists of a kitchen setup with everyday objects, a Franka Emika Panda arm with a two-finger Robotiq gripper fitted with Festo Adaptive Fingers, three fixed cameras (top, left, right), and a wrist camera mounted above the end-effector. The four camera views provide complementary perspectives of the workspace, and we utilize all of them for robust policy learning.

Data collection. Our robot manipulation dataset for the experiments consists of 7,500 trajectories, collected through tele-operation by a human operator, over a period of two months. We collect all the data in different kitchen-like environments with a Franka Emika [57]. The tele-operation stack is based on [58] and uses VR-controllers. The dataset comprises of diverse manipulation skills like opening/closing drawers, pouring, pushing, dragging, picking, placing, etc. across several everyday objects. Figure 3 shows the distribution of skills in the dataset. Additional details regarding the dataset, along with sample trajectories, and a link to the entire dataset are in the project website linked with the paper. We will publicly release this dataset, as part of the larger RoboSet described in subsection 3.1. To our knowledge, this is one of the largest open-source robot manipulation datasets with the most commonly used non-proprietary robot hardware (Franka Panda [57]) containing diverse behaviors beyond pick and place.

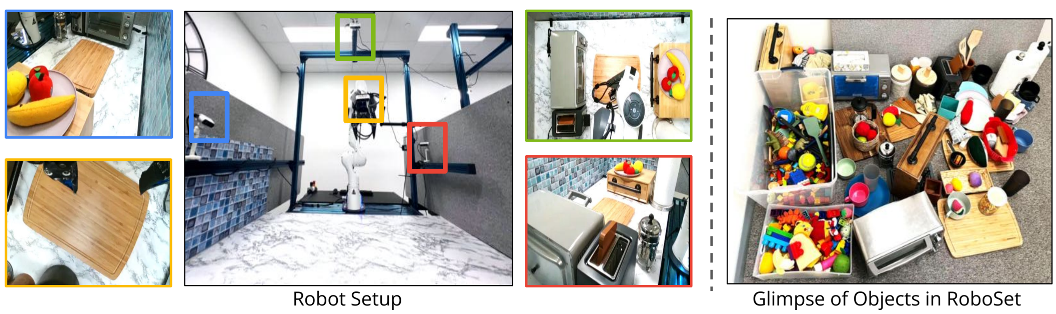


Figure 6: A zoomed-out view of the robot environment, showing all four cameras in the scene (in colored squares) and a glimpse of the diverse set of objects in the dataset, used for all our experiments.

Generalization Axes. Following prior work [1, 51, 14], we define each *task* to consist of a particular language command like 'pick a cube of butter from the drawer on the left' that defines an object to be interacted with (butter), a skill to be executed (pick), and some context (drawer on the left). We consider evaluations in terms of different levels of generalization, illustrated visually for a scene in Fig. 7: **L1(Effectiveness)**: Generalization of the agent to variations in object positions and orientations, and in lighting conditions. **L2 (Robustness):** New background, different distractor object variations, and unseen distractor objects introduced in the scene. **L3 (Generalization):** New tasks never seen before, including new object-skill combinations. **L4 (Strong Generalization):** New kitchen never seen before (see Figure 9 Left).

5 Experiments

Through detailed real-world robot manipulation experiments, we evaluate the proposed framework for sample efficiency, and generalization of the agent to diverse scenes.

Baselines. We compare multiple baselines that use visual policy learning for robotics. *Single Task Agents*: We compare ACT-based policies [5] trained for individual tasks, and evaluated on the respective tasks. These policies don't need to generalize across tasks and scene, and represent an approximate *oracle* performance per task. *Visual Imitation Learning (VIL)*: We compare with regular language-conditioned multi-task visual imitation learning. *CACTI* [3]: This baseline is a

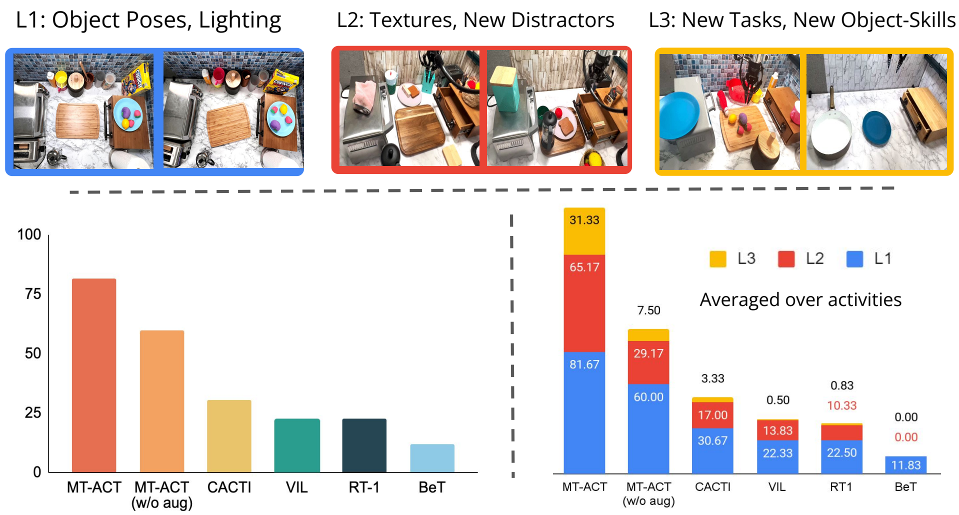


Figure 7: Visualization of different generalization axes, evaluating effectiveness with lighting variations and smaller scene changes such as object poses (L1), robustness to significant scene variations (L2), generalization to unseen tasks (L3). *Bottom-Left:* Results for commonly used L1-generalization. *Bottom-Right:* Results for different levels of generalization. See 9 for L4-generalization results.

prior framework for multi-task learning that also uses some scene augmentations for generalization. *RT1* [1]: We re-implement a baseline RT1-like agent. *BeT* [59]: We modify the Behavior Transformer architecture with language conditioning and train it in a multi-task manner.

Next, we present results and analysis from our large-scale real-world experiments that attempt to understand the research questions presented in section 4.

5.1 Multi-Task Real-World Results

Performance. Figure 7 (Left-Bottom) compares our proposed MT-ACT policies against commonly used imitation learning architectures. In this figure (Figure 7 Left-Bottom) we only plot results for *L1-generalization* since this is the standard setting most other imitation learning algorithms use. From the above figure we see that all approaches which only model next step actions (instead of sub-trajectories) perform quite poorly. Among these approaches we find that action-clustering based approaches (BeT [59]) for multi-task settings, perform significantly worse. We believe this happens because naive clustering in very diverse action distributions may not result in clusters that generalize across diverse skills. Additionally, since we are operating in the low data regime, we observe that RT1-like approaches that require a lot of data do not perform well in this setting. By contrast, our MT-ACT policy which uses action-checking to model sub-trajectories significantly outperforms all baselines. Overall, our further demonstrating the relative sample-efficiency of MT-ACT

Generalization and Robustness. Figure 7 (Bottom-Right) shows the results for all methods across multiple levels of generalization (**L1**, **L2**, and **L3**). Recall that these levels of generalization include diverse table backgrounds, distractors (**L2**) and novel skill-object combinations (**L3**). From Figure 7 (Bottom-Right) we see that by virtue of semantic augmentations and action representations, MT-ACT significantly outperforms all the baselines we consider. More interestingly, we see that semantic augmentations have less effect for L1-generalization ($\approx 30\%$ relative), they provide a *much more* significant improvement for both L2-generalization ($\approx 100\%$ relative) and L3-generalization ($\approx 400\%$ relative). Since semantic augmentations affect both scenes (backgrounds and distractor objects) as well as target objects being manipulated they provide useful support for the policy to achieve increasing levels of generalization.

Additionally, in Figure 8 we also separately report generalization results for each activity separately. From Figure 8 we see that each our proposed semantic augmen-

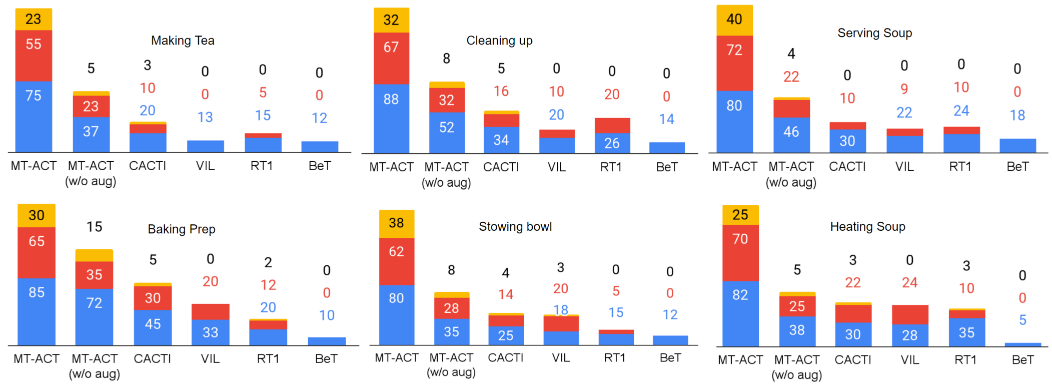


Figure 8: On the left we show results for MT-ACT , its ablated variant without semantic augmentations, and baselines, for different activities, with L1, L2, L3 levels of generalization. On the right, we show results for different levels of generalization averaged over different activities. We can see that MT-ACT achieves over 30-40% success rate than baselines in L2, and L3 generalization levels.

tations positively affect each activity's performance. Interestingly, we find that for some of the harder activities (Making-Tea, Stowing-Bowl, Heating Soup) the relative improvement in performance due to semantic augmentations is much larger. Overall, our results show that traditional visual imitation learning (without any augmentations) i.e. VIL and RT1 trained on our relatively small dataset, completely fail for L3 and L2, indicating a lack of generalization to unseen scenarios, due to data paucity. Finally, we also test our policy on a completely new kitchen with novel objects, arrangements, distractors, i.e., L4 generalization. Figure 9 (Left) visualizes this new kitchen environment. We evaluate all methods on this new kitchen for 3 tasks. Figure 9 (Right) shows the results for each method on this new environment. Specifically, we obtain an average success rate of 25% for MT-ACT (and 0 for all the other baselines). We see that MT-ACT without semantic augmentations also fails completely on this new environment thus indicating the benefits of our approach for zero-shot adaptation.

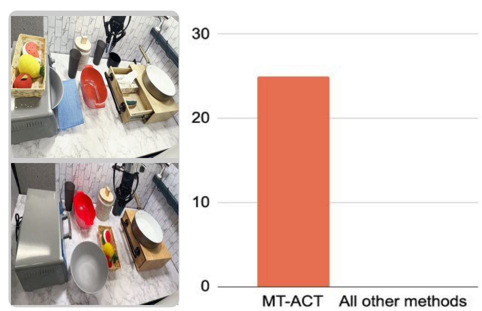


Figure 9: Only MT-ACT policies can perform tasks in a completely new kitchen environment (L4).

5.2 Ablation

We ablate the different design choices we make in our proposed architecture.

Task Specification using FiLM conditioning. For language conditioned multi-task policy, as described in section 3.3, we use a FiLM based conditioning [56] for the language embedding of task descriptions [60]. Here, we compare this design choice with a simple concatenation-based conditioning of the language embeddings with image tokens for the policy. In Fig. 10 we show results for this ablation study averaged over all activities, and we observe a 5-10% drop in performance of the version of MT-ACT without FiLM conditioning.

Chunk Size for Action Representations. Here we train variants of MT-ACT with different chunk sizes 10, 20, 40. In Fig. 10, we see that a chunk size of 20 performs the best, with a 0-5% drop in performance with chunk size 10. In addition, large chunk size 40 performs significantly worse with more than 20% drop in performance indicating the inability of the policy to correct errors as the chunks grow in size.

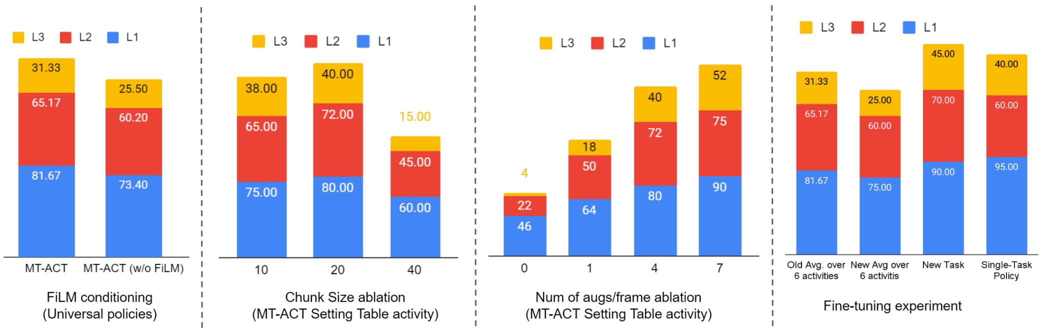


Figure 10: We shows results of the different ablation studies and analysis in section 5.2, showing the benefits of FiLM conditioning, the effect of varying chunk sizes in the predictions, the number of augmentations per frame for multiplying the dataset, and the feasibility of fine-tuning MT-ACT for improved deployment.

Num of aug per frame. We ablate for the number of augmentations per frame, to see if more augmentations help MT-ACT in learning a more performant policy. From Fig. 10, we see that the number of augmentations per frame is strongly correlated with overall performance gains. Thanks to the real-world semantic priors injected via data augmentation, the gains are more notable for L2 and L3 levels where out-of-domain generalization is required from the policy.

Robustness analysis. We perform several robustness analyses of the universal MT-ACT agent, by manually perturbing the scene during evaluation, and also introduce system failures like blocking the views from one, two, or three cameras. On average, we find that the policy is robust to these strong active variations, and can recover and solve the specified task in about 70% of the 20 evaluations we run for this analysis. Videos showing these robustness results are in the supplementary.

Plasticity. Here, we evaluate the feasibility of adding additional capabilities to the universal MT-ACT agent, without requiring significant re-training. We take the trained agent (on 30 tasks) and fine-tune on $(1/10)^{\text{th}}$ of the original data combined with data for a new held-out task (placing toast in toaster oven). The new task has 50 trajectories, multiplied with 4 augmentations per frame, for a total of 250 trajectories. In Fig. 10 on the right, we see that the fine-tuned agent is able to learn this new task, without significantly deteriorating in performance on the previous 6 activities. Also, it achieves slightly better L2, L3 performance than a single-task policy trained only on augmented data of the new task, indicating efficient data re-use.

6 Discussion and Limitations

We developed a framework for sample-efficient and generalizable multi-task robot manipulation in the real world. Our framework is based on rapidly multiplying a small robotics dataset through semantic scene augmentations, and training a multi-task language-conditioned policy that can ingest the diverse multi-modal data obtained through augmentations. We combine and adapt several design choices like action chunking and temporal aggregation proposed in the context of single-task policies, and show that they yield significant boosts in performance even in the multi-task settings we consider. Finally, we release one of the largest robot manipulation datasets to date involving over 12 skills in kitchen environments which we hope will facilitate further research in developing robot manipulation systems with diverse real-world generalization. An important limitation of our work is that all the tasks consist of individual skills, and an interesting direction for future work would be to develop approaches for composing skills automatically for solving long-horizon tasks. Another limitation is that we do not explore the axes of language generalization, and use language embeddings from pre-trained encoders as is, without any modifications. Future work could investigate better language conditioning that is more flexibly adaptable to changes in task descriptions.

References

[1] 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.

[2] Z. Chen, S. Kiami, A. Gupta, and V. Kumar. Genaug: Retargeting behaviors to unseen situations via generative augmentation. *arXiv preprint arXiv:2302.06671*, 2023.

[3] Z. Mandi, H. Bharadhwaj, V. Moens, S. Song, A. Rajeswaran, and V. Kumar. Cacti: A framework for scalable multi-task multi-scene visual imitation learning. *arXiv preprint arXiv:2212.05711*, 2022.

[4] T. Yu, T. Xiao, A. Stone, J. Tompson, A. Brohan, S. Wang, J. Singh, C. Tan, J. Peralta, B. Ichter, et al. Scaling robot learning with semantically imagined experience. *arXiv preprint arXiv:2302.11550*, 2023.

[5] 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.

[6] L. Pinto and A. Gupta. Learning to push by grasping: Using multiple tasks for effective learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2161–2168. IEEE, 2017.

[7] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet. Learning latent plans from play. In *Conference on robot learning*, pages 1113–1132. PMLR, 2020.

[8] L. Berscheid, T. Rühr, and T. Kröger. Improving data efficiency of self-supervised learning for robotic grasping. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2125–2131. IEEE, 2019.

[9] 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*, pages 1094–1100. PMLR, 2020.

[10] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[11] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, et al. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 2023.

[12] 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.

[13] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[14] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan. Vima: General robot manipulation with multimodal prompts. *arXiv preprint arXiv:2210.03094*, 2022.

[15] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[16] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. *arXiv preprint arXiv:1802.01561*, 2018.

[17] S. Sodhani, A. Zhang, and J. Pineau. Multi-task reinforcement learning with context-based representations. In *International Conference on Machine Learning*, pages 9767–9779. PMLR, 2021.

[18] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019.

[19] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[20] H. F. Song, A. Abdolmaleki, J. T. Springenberg, A. Clark, H. Soyer, J. W. Rae, S. Noury, A. Ahuja, S. Liu, D. Tirumala, et al. V-mpo: On-policy maximum a posteriori policy optimization for discrete and continuous control. *arXiv preprint arXiv:1909.12238*, 2019.

[21] A. Gupta, J. Yu, T. Z. Zhao, V. Kumar, A. Rovinsky, K. Xu, T. Devlin, and S. Levine. Reset-free reinforcement learning via multi-task learning: Learning dexterous manipulation behaviors without human intervention. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6664–6671. IEEE, 2021.

[22] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020.

[23] 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. In *Conference on Robot Learning*, pages 885–897. PMLR, 2020.

[24] 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*, pages 879–893. PMLR, 2018.

[25] 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*, pages 1678–1690. PMLR, 2022.

[26] 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.

[27] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[28] 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*, pages 18995–19012, 2022.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[30] 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.

[31] 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.

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

[33] A. Majumdar, K. Yadav, S. Arnaud, Y. J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, P. Abbeel, J. Malik, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *arXiv preprint arXiv:2303.18240*, 2023.

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

[35] M. Sharma, C. Fantacci, Y. Zhou, S. Koppula, N. Heess, J. Scholz, and Y. Aytar. Lossless adaptation of pretrained vision models for robotic manipulation. In *The Eleventh International Conference on Learning Representations*.

[36] N. Hansen, Z. Yuan, Y. Ze, T. Mu, A. Rajeswaran, H. Su, H. Xu, and X. Wang. On pre-training for visuo-motor control: Revisiting a learning-from-scratch baseline. *arXiv preprint arXiv:2212.05749*, 2022.

[37] S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, and N. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, pages 1507–1514, 2011.

[38] C. Lynch and P. Sermanet. Language conditioned imitation learning over unstructured data. *arXiv preprint arXiv:2005.07648*, 2020.

[39] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. *Advances in Neural Information Processing Systems*, 33:13139–13150, 2020.

[40] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning*, pages 287–318. PMLR, 2023.

[41] A. Nguyen, D. Kanoulas, L. Muratore, D. G. Caldwell, and N. G. Tsagarakis. Translating videos to commands for robotic manipulation with deep recurrent neural networks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3782–3788. IEEE, 2018.

[42] H. Bharadhwaj, A. Gupta, and S. Tulsiani. Visual affordance prediction for guiding robot exploration. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3029–3036, 2023.

[43] L. Shao, T. Migimatsu, Q. Zhang, K. Yang, and J. Bohg. Concept2robot: Learning manipulation concepts from instructions and human demonstrations. *The International Journal of Robotics Research*, 40(12-14):1419–1434, 2021.

[44] Y. Zhou, Y. Aytar, and K. Bousmalis. Manipulator-independent representations for visual imitation. *arXiv preprint arXiv:2103.09016*, 2021.

[45] H. Bharadhwaj, A. Gupta, S. Tulsiani, and V. Kumar. Zero-shot robot manipulation from passive human videos. *arXiv preprint arXiv:2302.02011*, 2023.

[46] S. Bahl, A. Gupta, and D. Pathak. Human-to-robot imitation in the wild. *arXiv preprint arXiv:2207.09450*, 2022.

[47] K. Rao, C. Harris, A. Irpan, S. Levine, J. Ibarz, and M. Khansari. Rl-cyclegan: Reinforcement learning aware simulation-to-real. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11157–11166, 2020.

[48] I. Kapelyukh, V. Vosylius, and E. Johns. Dall-e-bot: Introducing web-scale diffusion models to robotics. *arXiv preprint arXiv:2210.02438*, 2022.

[49] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, et al. Simple open-vocabulary object detection with vision transformers. *arXiv preprint arXiv:2205.06230*, 2022.

[50] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[51] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[52] C. Mitash, F. Wang, S. Lu, V. Terhuja, T. Garaas, F. Polido, and M. Nambi. Armbench: An object-centric benchmark dataset for robotic manipulation. *arXiv preprint arXiv:2303.16382*, 2023.

[53] F. Ebert, C. Finn, A. X. Lee, and S. Levine. Self-supervised visual planning with temporal skip connections. *CoRL*, 12:16, 2017.

[54] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[55] J. Yang, M. Gao, Z. Li, S. Gao, F. Wang, and F. Zheng. Track anything: Segment anything meets videos. *arXiv preprint arXiv:2304.11968*, 2023.

[56] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[57] S. Haddadin, S. Parusel, L. Johannsmeier, S. Golz, S. Gabl, F. Walch, M. Sabaghian, C. Jähne, L. Hausperger, and S. Haddadin. The franka emika robot: A reference platform for robotics research and education. *IEEE Robotics & Automation Magazine*, 29(2):46–64, 2022.

[58] V. Kumar and E. Todorov. Mujoco haptix: A virtual reality system for hand manipulation. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pages 657–663. IEEE, 2015.

[59] N. M. M. Shafiullah, Z. J. Cui, A. Altanzaya, and L. Pinto. Behavior transformers: Cloning k modes with one stone. *arXiv preprint arXiv:2206.11251*, 2022.

[60] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

A Dataset details

We used 7,500 human teleoperated demonstrations from the *RoboSet* dataset. The full dataset is much more diverse and consists of 9,500 tele-operated demonstrations and 11,500 kinesthetic demonstrations in various table-top settings. The subset that we used from the dataset consisted of RGB and depth frames from four camera views (right, left, top, and wrist) as shown in figure 6, Franka joint positions and velocities, end-effector/gripper position and velocities, controls applied to the Franka joints and end-effector/gripper, and the time-steps (40 steps).

The data was collected using an Oculus Quest 2 controller on a kitchen table-top setup at 5Hz and saved in HDF5 format. Rollouts from the data are shown in figure 11.

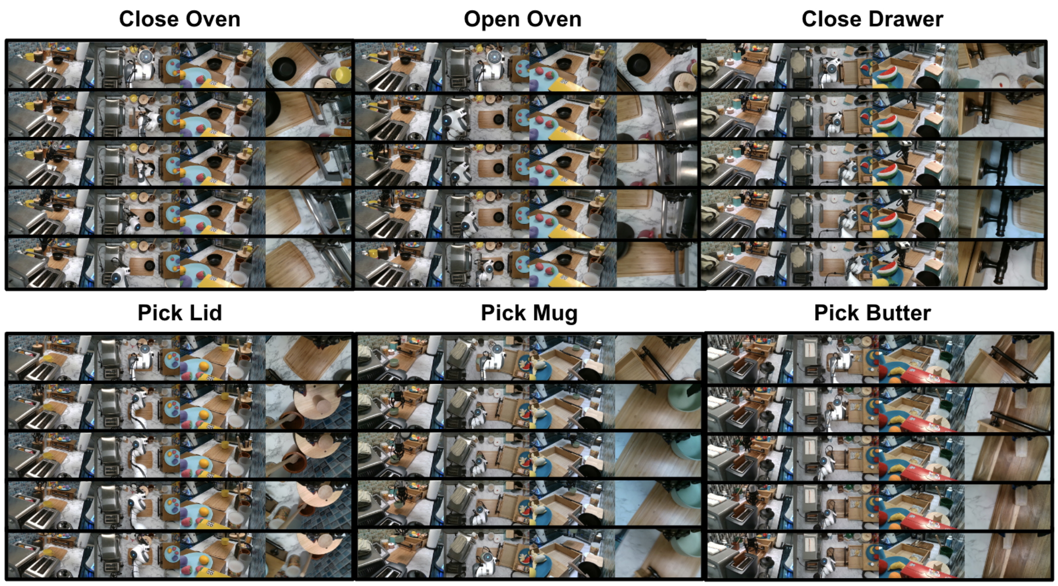


Figure 11: Rollouts of some tasks (visualizing four views horizontally, and five timesteps vertically) in the RoboSet, used for training.

A.1 Details on Semantic Augmentations

We enable two different types of scene augmentations for multiplying data, for enabling generalization to different scenes with novel distractors, and to scenes with different objects for interaction:

- **Augmenting interaction object:** Given the joint angle of the robot in a frame of a trajectory, we use forward kinematics to recover the robot mask as well as the end-effector position of the robot. We use the end-effector location to prompt SegmentAnything [54] for obtaining a mask of the object being interacted with. We then inpaint the region of the object being interacted with, based on a text prompt, and keep it consistent across time by tracking with TrackAnything [55].
- **Augmenting background:** We use SegmentAnything [54] to randomly choose a set of objects in the background that do not overlap with the robot mask, and the mask of the object being interacted with, and inpaint the scene based on the resulting overall mask over all the objects identified by SegmentAnything.

Note that our augmentation approaches are all automatic and do not require any manual effort in specifying masks or object meshes etc. This is in contrast to prior works that require manual specification of a fixed mask per trajectory [3], and those that require templates of object textures and meshes [2]. In addition, unlike [4], we do not require training any further modules for identifying objects through open-vocabulary detection that relies on language grounding.

Figure 12: Qualitative results of rollouts for L2 and L3 levels of generalization, showing tasks *open drawer* and *pick a slab of butter*. For L2 we introduce different distractors in the scene, and change the background tiles. For L3, in addition to changes in L2 we introduce different task objects, for example by replacing a slab of butter with a piece of watermelon, or a banana.

B Train and Evaluation Details

In this section we present training and evaluation details both for our methods and the baselines.

B.1 Robot Environment and Evaluation Details

The robot environments for evaluation consist of table-top kitchen setups with diverse real objects in the scene. There are 4 cameras providing complementary views of the workspace. The robot is a Franka Emika Panda arm operated with joint position control, with an action space dimension of 8 (7 joint positions, 1 dimension for end-effector open/close). The robot arm has a two-finger gripper, and a wrist camera. The robot is operated at a frequency of 5Hz.

B.2 Hyper-parameters for MT-ACT and baselines

Here we provide hyper-parameter details of the policy architecture. We train all policies for 2000 epochs. For the overall MT-ACT agent, trained on the augmented dataset, this takes about 48 hours on a single 2080Ti GPU with a batch size of 8.

For our baseline implementations we did a hyperparameter search for relevant parameters. For each baseline implementation, [1] we adapt them from their officially released code. Specficially, for RT1 [1] we use `https://github.com/google-research/robotics_transformer` for reference. On the other hand, for [59] we use `https://github.com/notmahi/bet`. To provide language conditioning for both baselines we use similar FiLM [56] implementation as our approach.

For hyper-parameters we use 3 different discrete action sizes – 64, 256 and 512, we vary the learning rates from $(1e-3, 1e-4)$. We use the AdamW optimizer with a weight decay range in $(1e-2, 1e-3, 1e-4)$. Our RT-1 transformer uses 6 layers with 8 parallel attention heads and each head with size 64. Each transformer uses a feedforward layer with intermediate sie of 1024. On the other hand for [59] we experiment with 3 different action cluster sizes – 64, 256 and 512. We use a similar transformer implementation for BET as RT-1. Finally, for real-world evaluation we use the hyper-parameters with lowest validation loss.

C Additional Results

In this section we present some additional results. First, we present results and discuss how well does our multi-task policy perform when compared to single task policies. Figure 13 compares single-task policy performance against two sets of multi-task policies for the *Heat Soup* activity. For the first multi-task Single-Activity policy (MT Single-Activity) we only train it across all tasks within the same activity. For the latter multi-task universal multi-activity policy (MT-Universal) we train it across all tasks in all activities. From Figure 13 we see that for most tasks *MT Single-Activity*

Table 2: Hyper-parameters for MT-ACT

Name	Value
learning rate	1e-5
batch size	8
feedforward size	3200
Attention heads	8
chunk size	20
dropout	0.1
Transformer encoder layers	4
Transformer decoder layers	7
Language Embedding size	384

Table 3: Hyper-parameters for RT-1 [1]

Name	Value
learning rate	1e-4
discrete action tokens	256
batch size	64
feedforward size	1024
Attention heads	8
dropout	0.1
Transformer layers	6
Language Embedding size	384

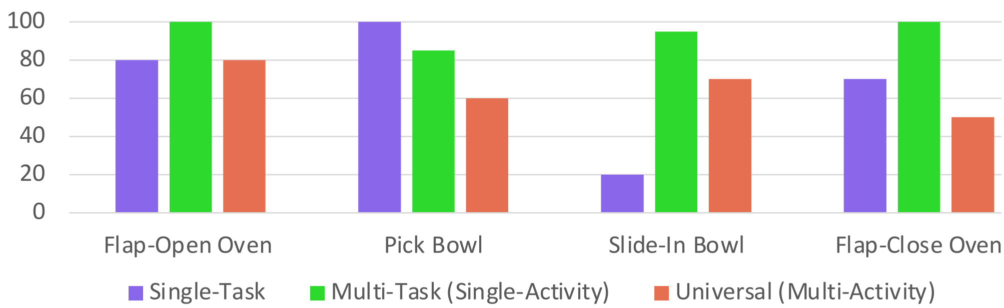


Figure 13: Single-Task vs Multi-Task comparison for Heat Soup activity. Multi-Task (Single Activity) represents a multi-task policy trained on only 4 tasks in Heat-Soup activity.

is able to outperform single task policies. Additionally, single-task policies are able to perform well on most tasks ($\approx 80\%$) except the more challenging constrained manipulation tasks (slide-in-bowl) ($\approx 20\%$). Finally, we also observe that MT-Single-Activity can outperform MT-Universal for most tasks. This happens because the universal agent is trained to perform a much larger variety of tasks. Given the very large variety of skills (Figure 3), such multi-task training can result in some negative transfer compared to training on a narrowly defined skills. We believe these reduced multi-task results present useful avenues for future research. Finally, in Table 4 we show results for all tasks in all activities using our single universal policy. From the below table, we see that the universal policy is able to perform well on most tasks except the more challenging tasks such as grasping small deformable objects (Pick Tea: 40%, Pick Lid: 50%).

Heat Soup	Success	**Serve Soup**	Success	**Baking Prep**	Success
Flap-Open Oven	80%	Flap-Open Oven	90%	Slide-Open Drawer	70%
Pick Bowl	60%	Pick Bowl	50%	Pick Butter	70%
Slide-In Bowl	70%	Slide-Out Bowl	80%	place Butter	90%
Flap-Close Oven	50%	Flap-Close Oven	80%	Slide-Close Drawer	90%
Making Tea	Success	**Cleaning Up**	Success	**Stow Bowl**	Success
Uncap Lid	80%	Pick Lid	70%	Slide-Open Drawer	70%
Place Lid	90%	Cap Lid	100%	Pick Bowl	70%
Pick Tea	40%	Slide-Close Drawer	90%	Place Bowl	80%
Place Tea	60%	Flap-Close Oven	80%	Slide-Close Drawer	80%
Pick Lid	50%	Pick Towel	90%		
Cap Lid	70%	Wipe Counter	90%		

Table 4: Results for each of the tasks using the learned **universal policy**.