

# Bridge Data: Boosting Generalization of Robotic Skills with Cross-Domain Datasets

Frederik Ebert<sup>\*1</sup>, Yanlai Yang<sup>\*1</sup>, Karl Schmeckpeper<sup>3</sup>, Bernadette Bucher<sup>3</sup>, Georgios Georgakis<sup>3</sup>, Kostas Daniilidis<sup>3</sup>, Chelsea Finn<sup>2</sup>, Sergej Levine<sup>1</sup>

**Abstract**—Robot learning holds the promise of learning policies that generalize broadly. However, such generalization requires sufficiently diverse datasets of the task of interest, which can be prohibitively expensive to collect. In other fields, such as computer vision, it is common to utilize shared, reusable datasets, such as ImageNet, to overcome this challenge, but this has proven difficult in robotics. In this paper, we ask: what would it take to enable practical data reuse in robotics for end-to-end skill learning? We hypothesize that the key is to use datasets with multiple tasks and multiple domains, such that a new user that wants to train their robot to perform a new task in a new domain can include this dataset in their training process and benefit from cross-task and cross-domain generalization. To evaluate this hypothesis, we collect a large multi-domain and multi-task dataset, with 7,200 demonstrations constituting 71 tasks across 10 environments, and empirically study how this data can improve the learning of new tasks in new environments. We find that jointly training with the proposed dataset and 50 demonstrations of a never-before-seen task in a new domain on average leads to a 2x improvement in success rate compared to using target domain data alone. We also find that data for only a few tasks in a new domain can bridge the domain gap and make it possible for a robot to perform a variety of prior tasks that were only seen in other domains. These results suggest that reusing diverse multi-task and multi-domain datasets, including our open-source dataset, may pave the way for broader robot generalization, eliminating the need to re-collect data for each new robot learning project.

## I. INTRODUCTION

Humans and animals can generalize a learned skill to a wide variety of contexts without needing to relearn the skill every time. Endowing robots with the same capability would be a significant advance toward making robots more applicable to a range of real-world settings. However, the prevailing paradigm of robot learning is to repeat data collection and policy training from scratch for every new task and environment. Learning policies in isolation not only increases the costs of data collection, but also limits the policy’s scope of generalization.

In other fields, such as computer vision [14] and natural language processing (NLP) [3], utilizing large, diverse datasets has shown considerable success in enabling generalization to new problems or domains with a small amount of data (e.g., via pretraining and finetuning). However, in robotics, datasets are usually collected with a specific robotic platform and domain in mind, typically by the same researcher who intends to use that dataset. What would it take to make datasets

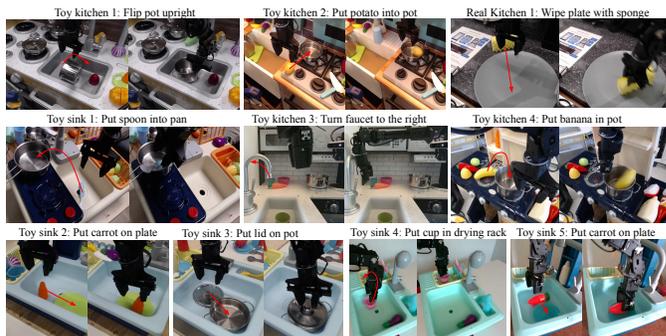


Fig. 1: Illustration of our bridge dataset. The dataset includes demonstrations in 10 environments (4 toy kitchens and 5 toy sinks and 1 real kitchen), collected using a WidowX250 robot controlled via an Oculus Quest2 VR device, and consists of 7200 demonstrations. The red arrows indicate the desired movement of the target object.

reusable in robotics in the same way as large supervised datasets are reused (e.g., ImageNet [2])? Each end-user of such a dataset might want their robot to learn a different task, which would be situated in a different domain (e.g., a different laboratory, home, etc.). It is currently an open question whether such reuse is feasible in robotics, and we posit that any such dataset would need to cover both multiple different tasks and multiple different domains. To this end, the aim of our paper is to investigate the degree to which such a multi-task and multi-domain dataset, which we refer to as a *bridge* dataset, can enable a new robot in a new domain (which was not seen in the bridge data) to more effectively generalize when learning a new task (which was also not seen in the bridge data), as well as to transfer tasks from the bridge data to the target domain. We also propose a new dataset that enables this goal in the context of kitchen-themed tasks with a low-cost robotic arm and is intended to be reused by other researchers.

The notion that multi-task data can speed up learning or improve generalization has been studied in many prior works [26, 13]. However, unlike this paper, the focus in these prior works, as we discuss in Section II, is not on enabling new users to quickly train generalizable skills in a new setting or domain, but rather to utilize multi-task learning to lower the data requirements of acquiring a pre-defined set of tasks. More closely related to our work, RoboNet [1] contains data from multiple robots and domains, but this data is collected using random motions, and does not provide examples of multiple different tasks that can be used for more complex task-directed manipulation. We discuss other datasets in Section II; but in summary, no existing dataset covers *both* multiple tasks *and* multiple domains in a way that is suitable to study our

\* denotes equal contribution

<sup>1</sup> University of California Berkeley, <sup>2</sup> Stanford University, <sup>3</sup> University of Pennsylvania

central hypothesis: can prior data be used to improve the generalization of *new* tasks in *new* domains? We will call this the *bridge data hypothesis*. We believe this is a critical requirement for effective data reuse in robotics, where different labs and researchers can all bootstrap from the same shared datasets. To study this, we collected a new multi-domain manipulation dataset with 7,200 demonstrations of 71 distinct and semantically meaningful tasks, themed around household tasks in kitchen environments. The data was collected across 10 distinct “toy” kitchens, as shown in Figure 1. This data is suitable for imitation learning, which is the focus of our work, though it could also be repurposed for offline RL and other algorithms in the future. We present our new dataset, and then use it to evaluate the bridge data hypothesis that is stated above, using three types of transfer scenarios: **(1)** When the user needs to train an existing task in a new domain, does the inclusion of bridge data boost performance? This roughly corresponds to a standard domain adaptation setting. **(2)** After the user has collected some data for a few tasks in a new domain, can their robot then perform other tasks that were *not* seen in the new domain, but are only present in the bridge data (i.e., can it “import” tasks from the bridge data)? **(3)** When the user collects some data in a new domain for a task that was *not* seen in the bridge data, can the performance and generalization of this task be boosted by including the bridge data in training? Scenario **(3)** directly evaluates our central hypothesis, while the other scenarios illustrate other potential uses for bridge data.

The main contributions of our work consist of an empirical evaluation of the bridge data hypothesis and a practical example of a bridge dataset with 7,200 demonstrations for 71 tasks in 10 environments, which we have released publicly on the project website<sup>1</sup>. To the best of our knowledge, our work is also the first to demonstrate transfer scenarios **(2)** and **(3)** above. This is significant, because **(2)** provides users with a low-cost way to “import” all of the skills in the bridge dataset into their own domain with just a small number of demonstrations in their domain, while **(3)** provides for a way to boost the performance of an entirely new skill with previously collected reusable bridge data. Our results suggest that accumulating and reusing diverse multi-task and multi-domain datasets, at least when all data is collected with the same type of robot, may make it possible for researchers to endow robots with generalizable skills using only a modest amount of in-domain data for their desired task.

## II. RELATED WORK

While most prior work on deep visuomotor learning trains a single task in a single domain [8, 4, 11, 25, 17, 21, 9, 23, 28], our goal is not to develop better learning methods, but rather to illustrate how generic multi-domain, multi-task datasets can be used with existing algorithms to boost the generalization of *new* tasks in *new* domains. Prior work on multi-task reinforcement learning [13] has shown that data from other

Dataset	# Tasks	# Trajec.	# Domains	suitable for BC/IL
DAML [25]	3	2.9k	1	✓
MIME [22]	22	8.2k	1	✓
RoboNet [1]	N/A	162k	7	✗
RoboTurk [18, 19]	3	2.1k	1	✓
Vis. Imit. Made [24]	2	2k	50	✓
<b>Ours</b>	<b>71</b>	<b>7.2k</b>	<b>10</b>	<b>✓</b>

Fig. 2: Comparison of our dataset and prior works. Our dataset has by far the most tasks, and is the only dataset with more than 2 tasks that has many domains. This is critical for evaluating the bridge data hypothesis.

tasks can boost generalization of new tasks, however this study is carried out in a *single* domain.

Existing robot learning datasets do not exhibit the right properties for boosting the generalization of new tasks in *new* domains or zero-shot transferring skills from the prior dataset to a target domain. We provide an overview of the most related datasets in Figure 2. Most existing robot datasets, such as MIME [22], DAML [25], RoboTurk [18, 19], and many others [20, 6, 16, 12, 5, 27, 13] only feature a single domain, making them difficult to use for boosting the generalization in *other* domains. Merging multiple existing datasets into one multi-domain dataset is difficult due to inconsistencies in data collection protocols, time discretization, robot morphologies, and sensors. Learning from multiple robots has been studied with RoboNet [1], which provides a dataset with 7 different robots in different domains. Here the data is generated with random motions which do not produce semantically meaningful tasks. This limits task complexity to pushing and basic grasping, and makes the data poorly suited for imitation learning.

Some prior works have also used datasets collected by humans without a robot, across multiple domains. For example, Young et al. [24] presents results on data across many more domains than our bridge data, collected via a hand-held gripper, but only presents two grasping tasks.

## III. BRIDGE DATASETS

In this section, we describe the basic principles behind bridge datasets and how they can be used to boost generalization. Then, we present a description of the specific bridge dataset that we collected using teleoperation of a low-cost robotic arm for a range of kitchen-themed manipulation tasks. We use the term *bridge dataset* to refer to a large and diverse dataset of robotic behaviors collected in a range of settings (e.g., different viewpoints, lighting conditions, objects, and scenes), for a range of *different* tasks, so as to make it possible to “bridge” gaps in the generalization that arise when the user provides a small to medium amount of data in their specific target domain. We define the term “target domain” to refer to the environment where the robot must perform the desired task. This target domain is distinct from *any* of the settings seen in the bridge dataset: the intent is for the same large bridge dataset to be used by all users for whichever target domain they require.

### A. Boosting Generalization via Bridge Datasets

We consider three types of generalization in our experiments, though other modes may also be feasible:

<sup>1</sup><https://sites.google.com/view/bridgedata>

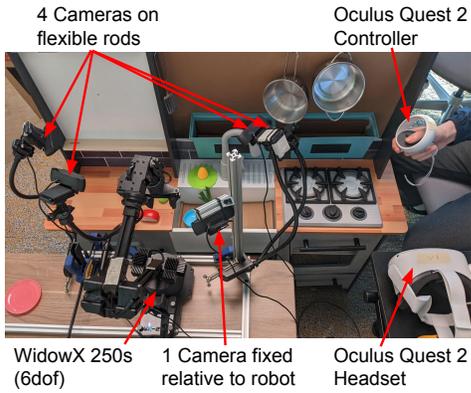


Fig. 3: Demonstration data collection setup using VR Headset. The scene is captured by 5 cameras simultaneously. While one of the cameras is fixed, the others are on

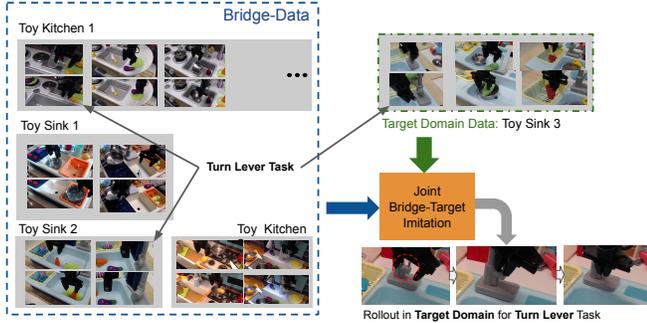


Fig. 4: **Scenario (1): transfer with matching behaviors.** In this setting, bridge data is used to improve the performance and generalization of tasks in the target domain for which the user has collected some amount of data. These tasks must also be present in the bridge data. In this example, the user demonstrates the “turn lever,” “squash into pot,” and “flip cup” tasks in the target domain, and these tasks are also present in several domains in the bridge data. After including the bridge data in training, the performance and generalization of these tasks in the target is significantly higher.

**(1) Transfer with matching behaviors**, where the user collects some small amount of data in their target domain for tasks that are also present in the bridge data (e.g., around 50 demos per task), and uses the bridge data to boost the performance and generalization of these tasks. We illustrate this scenario in Figure 4. This scenario is the most conventional, and resembles domain adaptation in computer vision, but it is also the most limiting, since it requires the user’s desired tasks to be present in the bridge data. However, as we will show, bridge data can enable very significant performance and generalization boosts in this setting.

**(2) Zero-shot transfer with target support**, where the user utilizes data from a few tasks in their target domain to “import” other tasks that are present in the bridge data *without* additionally collecting new demonstrations for them in the target domain. For example, the bridge data contains the tasks of putting a sweet potato into a pot or a pan, the user provides data in their domain for putting brushes in pans, and the robot is then able to *both* put brushes as well as put sweet potatoes in pans. We illustrate this scenario in Figure 5. This scenario increases the repertoires of skills that are available in the user’s target environment, simply by including the bridge data, thus eliminating the need to recollect data for every task in every target environment.

**(3) Boosting generalization of new tasks**, where the user provides a small amount of data (50 demonstrations in prac-

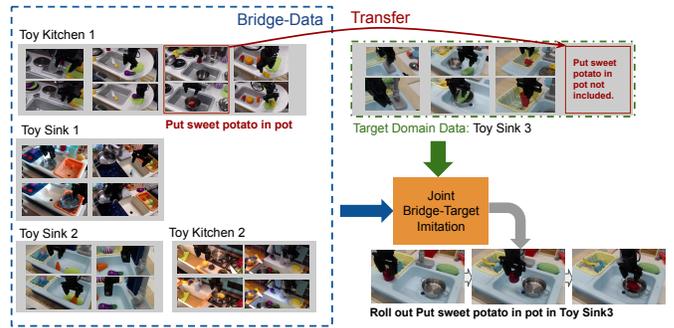


Fig. 5: **Scenario (2): zero-shot transfer with target support.** In this setting, the goal is to “import” a task from the bridge data that was *not* seen in the target domain. The user provides a few tasks in the target domain that are used to connect to the bridge data, and then asks the robot to perform a task that they did not provide, but which was seen in the bridge data. In this case, the “put sweet potato in pot” task is present in the toy kitchen 1 domain in the bridge data, but is *not* demonstrated by the user in the target domain. After training with user-provided data for other tasks, the robot is able to

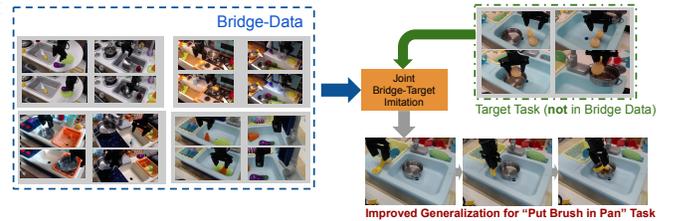


Fig. 6: **Scenario (3): boosting generalization of new tasks.** The user provides some data for a *new* task that was not seen in the bridge data, and the bridge data is included in training to boost performance and generalization for this new task.

tion) for a new task that is not present in the bridge data, and then utilizes the bridge data to boost generalization and performance of this task. This scenario, illustrated in Figure 6, most directly reflects our primary goals, since it uses the bridge data without requiring *either* the domains or tasks to match, leveraging the diversity of the data and structural similarity to boost performance and generalization of entirely new tasks.

To enable this kind of generalization boosting, we conjecture that the key features that bridge datasets must have are: (i) a sufficient variety of settings, so as to provide for good generalization; (ii) shared structure between bridge data domains and target domains (i.e., it is unreasonable to expect generalization for a construction robot using bridge data of kitchen tasks); (iii) a sufficient range of tasks that breaks unwanted correlations between tasks and domains. Analogously to how the ImageNet dataset [2] provides broad coverage that makes it possible to boost generalization for a range of computer vision tasks, the broader a bridge dataset is, the more likely target tasks receive a generalization boost in a particular target domain.

### B. A Bridge Dataset of Large-Scale Kitchen Tasks

We instantiate a bridge dataset based on the principles above as follows:

**Robotic system overview.** Since our dataset is likely the most useful for users with the same or similar type of robot, we chose to use a low-cost and widely available robot, a 6-dof WidowX250s (US\$2900), which many other users of our dataset are likely to be able to obtain. The total cost of the setup is less than US\$3600 (excluding the computer). To collect demonstrations, we use an Oculus Quest headset, where

we put the headset on a table as illustrated in Figure 3 next to the robot and track the user’s handset while applying the user’s motions to the robot end-effector via inverse kinematics. We capture images from 3 to 5 cameras concurrently, using standard webcams as well as Intel RealSense depth cameras. **Data collection protocol.** Our proposed bridge dataset, illustrated in Figure 1 consists of a total of 7200 demonstrations for 71 different tasks, collected in 10 different environments, focusing on the theme of household kitchen tasks. Each task has between 50 and 300 demonstrations. We opted to use kitchen and sink ”play sets” for children, since they are smaller than real-world kitchens and therefore ideal for small-scale robots, and they are comparatively robust and low-cost, while still providing settings that resemble typical household scenes. During data collection we randomize the kitchen position (translations of 0-20cm) and the camera positions (translations of 0-10cm and rotations of 0-30 degrees) for all cameras on flexible rods every 25 trajectories. The positions of distractor objects (i.e. objects not needed for a task) are randomized at least every 5 trajectories. All environments except toy sink 4, toy sink 5, and kitchen 3 were collected at Institution 1 and use Logitech C920 webcams, the three remaining environments were collected at Institution 2 and use Intel RealSense RGB-D cameras. The trajectories collected at Institution 2 randomize all camera positions once every 50 trajectories. Instructions for how users can reproduce our setup and collect data in new environments can be found on the project website.<sup>2</sup>

#### IV. USING BRIDGE DATA IN IMITATION LEARNING

As a proof-of-concept to illustrate the utility of bridge datasets for boosting generalization in robot learning, we will present experimental results for an imitation-based approach that utilizes this data, although the data could also be used with a variety of other robotic learning algorithms such as offline RL and model-based planning.

**Incorporating bridge data.** While a variety of transfer learning methods have been proposed in the literature for combining datasets from distinct domains, we found that a simple joint training approach is effective for deriving considerable benefit from bridge data. For each of the scenarios outlined in Section III-A, we take the user-provided demonstrations in the target domain and combine them with the entire bridge dataset for training. Since the sizes of these datasets are significantly different, we rebalance the datasets by weighting each datapoint, as discussed at the end of this section. Imitation learning then proceeds normally, simply training the policy with supervised learning on the combined dataset using the architecture described in the following paragraph. It is also possible to incorporate bridge data in other ways, for example by pretraining and finetuning. We found pretraining to be significantly less effective than joint training in our experiments, a finding that is consistent with prior works [16], but we emphasize that bridge datasets can be combined with target domain data in a variety of ways. **Policy architecture.** We use

task-conditioned behavioral cloning (BC) with an additional task-id input to the policy, which is used to distinguish tasks during training and testing. In some cases, a task cannot be uniquely determined by only observing the input image, and a one-hot vector representing the task will solve this issue. The images are first fed into a 34-layer ResNet [10] and the resulting feature maps are passed through a spatial softmax [7, 15], which extracts a set of spatial positions of the relevant features. The spatial features are then concatenated with the one-hot task-id vector, and are fed into 3 layers of fully-connected networks by which the final action prediction is produced. During training, for a batch of training data containing tuples of task ids, images, and ground-truth actions, the network is trained by minimizing the standard  $\ell_2$ -error between the ground-truth actions and the predicted actions given by the policy provided the task id and the image observation as the input.

**Training details.** Since the amount of target domain data is usually significantly less than the amount of bridge data, we rebalance the two datasets during training. In the matching behaviors and zero-shot transfer with target support scenarios, the ratio between the number of trajectories in the bridge and target data is roughly 10:1, and we rebalance the data such that 70% of the dataset is bridge data and 30% is target domain data. In the ”boosting generalization of new tasks” scenario the imbalance is more severe, roughly 60:1, and so we rebalance such that 90% of the dataset is bridge data and 10% is target domain data. Lower rebalancing ratios of bridge data and target domain data tend to produce overfitting when the amount of target domain data is as low as 50 demonstrations.

#### V. EXPERIMENTAL RESULTS

Our experimental evaluation aims to study how well bridge data can facilitate generalization in scenarios (1), (2), and (3), as outlined in Section III-A. We utilize the bridge dataset described in Section III-B. We evaluate generalization on a set of new target domains with limited target domain data for each of the generalization scenarios, and compare the performance of learned policies with and without bridge data. Videos of the experiments are included in the supplementary materials and on the project webpage, which we encourage the reader to view to get a clearer sense for the diversity of the tasks: <https://sites.google.com/view/bridgedata>

**Quantitative metrics.** All quantitative evaluations use 10 trials per task, varying object positions and distractors on every trial and varying the position of the robot relative to the environment every 5 trials. This ensures that all test configurations are unique and different from any condition seen in training, providing a measurement of generalization performance for the policy. When the experiments in toy kitchen 1-3 and toy sink 1-3 were conducted, the bridge dataset only comprised 4700 trajectories. Other experiments use the full dataset with 7200 trajectories total.

**Scenario (1): transfer with matching behaviors.** Figure 7 (left) shows results for the transfer learning with matching behaviors scenario, where the user provides some data for a

<sup>2</sup><https://sites.google.com/view/bridgedata>

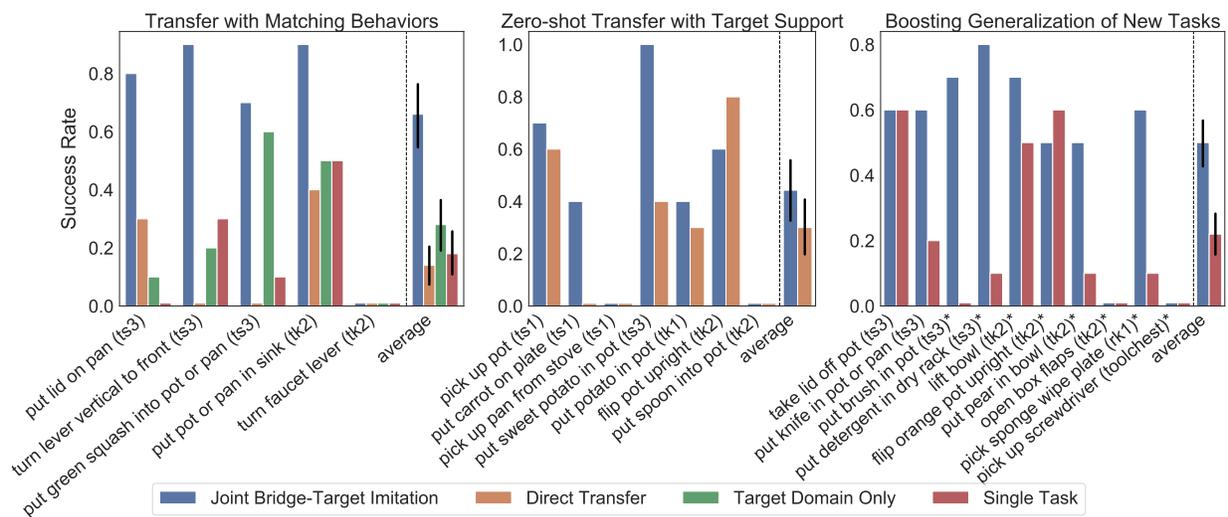


Fig. 7: Comparisons of joint training with bridge data (blue) and other approaches for each type of scenario. The black vertical lines on the average success rate bar denote the standard error of the mean across different tasks for that scenario. **Left:** Performing joint training on bridge and target data leads to improved performance, here the task is included both in the bridge and target dataset. **Middle:** Using target domain data from other tasks helps transferring tasks from the bridge dataset to the target domain. **Right:** Joint training with the bridge data and a target task that is *not* contained in the bridge dataset enables significant generalization improvement compared to only training on the target task alone. Tasks with an asterisk (\*) uses objects that are not part of the bridge dataset.



Fig. 8: Examples of successful trajectories performed by the policy jointly trained with prior data and target domain data. Left: put pot in sink (scenario 1); middle: put carrot on plate (scenario 2); Right: wipe plate with sponge (scenario 3).

set of tasks in the target domain (which are also present in the bridge data), and we evaluate whether including bridge data during training improves performance and generalization. For comparison, we include the performance of the policy when trained *only* on the target domain data, without bridge data (Target Domain Only), a baseline that uses only the bridge data without any target domain data (Direct Transfer), as well as baseline that trains a single-task policy on data in the target domain only (Single Task). The Toy Kitchen 2 (tk2) target domain has 6 tasks, and Toy Sink 3 (ts3) has 10 tasks, each with 50 demonstrations.

As can be seen in the results, jointly training with the bridge data leads to significant gains in performance (66% success averaged over tasks) compared to the direct transfer (14% success), target domain only (28% success) and the single task (18% success) baseline. This is not surprising, since this scenario directly augments the training set with additional data of the *same* tasks, but it still provides a validation of the value of including bridge data in training (for a qualitative example see Figure 8, left).

**Scenario (2): zero-shot transfer with target support.** In the next experiment, we evaluate tasks in the target domain for which the user did *not* provide any data. Instead, the user only collected data for *other* tasks in the target domain. This experiment evaluates whether bridge data can be used to “import” tasks into the target domain. We provide a qualitative example for this scenario in Figure 8 middle, which shows an experiment where we transfer the “put carrot on plate” task into the Toy Sink 1 target domain using the bridge data and target domain data consisting of 10 *other* tasks. Due to space

constraints, We provide a visualization of these other tasks on the project webpage.

Since there is no target domain data for these tasks, we cannot compare to a baseline that does not use bridge data at all, since such a baseline would have no data for these tasks. However, we do include the “direct transfer” baseline, which utilizes a policy trained only on the bridge data. Note that this comparison is non-trivial: it is not at all clear a priori that target domain data for *other* tasks should boost transfer performance of tasks that are only present in the bridge data. The results, shown in Figure 7 (middle), indicate that the jointly trained policy which obtains 44% success averaged over tasks indeed attains a very significant increase in performance over direct transfer (30% success), suggesting that the zero-shot transfer with target support scenario offers a viable way for users to “import” tasks from the bridge dataset into their domain.

**Scenario (3): boosting generalization of new tasks.** The last generalization scenario, which most directly evaluates the *bridge data hypothesis*, aims to study how well bridge data can boost the generalization of entirely new tasks in the target domain, which are not present in the bridge data. To study this question, we collected data for 10 different unique tasks in 4 different environments and excluded them from the bridge data to simulate a user collecting their own unique task in their new target environment. Figure 8 right illustrates one of these scenarios, where we collected 50 demonstrations for the “wipe place with sponge” task in the the real kitchen 1 target domain. *Neither* data from the target domain nor this task or this object are present in the bridge data. After jointly training with both bridge and target data

we obtain a significant generalization boost when running the policy in the target domain, compared to a policy trained on only the single-task target domain data. Direct transfer is impossible here, because the bridge data does not contain this task. The results are presented in Figure 7 (right), and show that training jointly with the bridge data leads to significant improvement on 6 out of 10 tasks across three evaluation environments, leading to 50% success averaged over tasks, whereas single task policies attain around 22% success – a  $2\times$  improvement in overall performance (the asterisks denote in which experiments the objects are *not* contained in the bridge data). The significant improvements obtained from including the bridge data suggest that bridge datasets can be a powerful vehicle for boosting generalization of new skills, and that a single shared bridge dataset can be utilized across a range of domains and applications. Of course, structural similarity between environments and tasks is important, and all of these evaluations use other toy kitchen or sink setups. We expect the applicability of a bridge dataset to increase as the breadth of domains and tasks in the dataset increases. **When does bridge data help?** In Figure 9 we provide a list of example scenarios where the bridge data helps and where it does not (the first 7 rows). More qualitative results, including videos of these tasks and additional discussion, are provided on the project [website](#) due to space constraints. Qualitatively, we observed that the tasks that most consistently benefit from the inclusion of bridge data contain objects that visually resemble those seen in the bridge data (e.g., there are gains for ‘put pear in bowl,’ where the pear resembles the vegetables in the bridge data, but no gains in ‘flip orange pot upright,’ since the orange pot looks very different from any container in the bridge data), contain behavior that physically is related to behavior seen in the bridge data (e.g., in ‘put detergent in dry rack,’ the bridge data helps since the motion resembles the pick-and-place motions in the prior data, whereas in ‘open box flaps’ bridge data does not help since the type of pushing motions involved in this task are very rare in the bridge data), and take place in domains that are visually and structurally related to those in the bridge data (e.g., the bridge data helps with ‘wipe plate with sponge’ in Figure 8 in a real kitchen, but does not help with ‘pick screwdriver from tool chest task,’ since the scene does not resemble the toy kitchens in the bridge data). Unfortunately, it is difficult to provide a more precise and formal treatment of when transfer learning succeeds in general, though we expect this would be an exciting direction for future research.

## VI. CONCLUSION

We show how a large, diverse bridge dataset can be leveraged to improve generalization in robotic learning. Our experiments demonstrate that including bridge data when training skills in a new domain can improve performance across a range of scenarios, both for tasks that are present in the bridge data and, perhaps surprisingly, entirely new tasks. This means that bridge data may provide a generic tool to improve generalization in a user’s target domain. In

addition, we showed that bridge data can also function as a tool to *import* tasks from the prior dataset to a target domain, thus increasing the repertoires of skills a user has at their disposal in a particular target domain. This suggests that a large, shared bridge dataset, like the one we have released, could be used by different robotics researchers to boost the generalization capabilities and the number of available skills of their imitation-trained policies.

Both our experimental evaluation and our technical approach do have a number of limitations. While we carefully set up our experiments to reflect a likely real-world usage scenario, where the target domain is distinct from the bridge data (i.e., to reflect what would happen if *someone else* were to use our bridge data for their robot in their lab), we still only evaluate in a few distinct settings, namely in 5 different environments at Institution 1.

However, our imitation learning results do illustrate the benefits of diverse bridge data, and we hope that by releasing our dataset to the community, we can take a step toward generalizing robotic learning and make it possible for anyone to train robotic policies that readily generalize to varied environments without repeatedly collecting large and exhaustive datasets.

Task	Tar. Env	Joint Train	Single Task	Potential reason for no gain with bridge data
turn faucet lever (1)	tk2	0%	0%	The faucet in tk2 has very different appearance from the other faucets
Pickup pan from stove (2)	ts1	0%	N/A	Not enough target domain data and prior data for this task
Put spoon into pot (2)	tk2	0%	N/A	Not enough target domain data and prior data for this task
flip orange pot upright (3)	tk2	50%	60%	There is no orange pot in prior dataset, only metal pots in prior dataset
open box flaps (3)	tk2	10%	10%	Boxes and pushing motions do not occur in prior data
take lid off pot (3)	ts3	60%	60%	Only 100 demos involving lids in prior dataset.
pick up screw driver (3)	toolchest	0%	0%	The toolchest and screwdrivers are visually very different from prior data
put pot or pan in sink (1)	tk2	90%	50%	
put carrot on plate (2)	ts1	40%	N/A	
Wipe plate w/ sponge (3)	tk1	70%	N/A	
put pear in bowl (3)	tk2	50%	10%	
put brush in pot (3)	ts3	90%	0%	
put detergent dry rack (3)	ts3	80%	10%	
lift bowl (3)	tk2	70%	50%	

Fig. 9: Comparison of scenarios where usage of the bridge data helps performance and where it does not. Scenarios where usage of bridge data does not help are marked in red font. The type of transfer setting is denoted by the number in brackets after the task description.

## REFERENCES

- [1] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. *arXiv preprint arXiv:1910.11215*, 2019.
- [2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, 2009.
- [3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [4] Yan Duan, Marcin Andrychowicz, Bradly C Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. *arXiv preprint arXiv:1703.07326*, 2017.

- [5] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.
- [6] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017.
- [7] Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Deep spatial autoencoders for visuomotor learning. In *International Conference on Robotics and Automation (ICRA)*, 2016.
- [8] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *Conference on Robot Learning*, pages 357–368. PMLR, 2017.
- [9] Ali Ghadirzadeh, Xi Chen, Wenjie Yin, Zhengrong Yi, Marten Bjorkman, and Danica Kragic. Human-centered collaborative robots with deep reinforcement learning. *IEEE Robotics and Automation Letters*, 2020.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016.
- [11] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *arXiv preprint arXiv:1606.03476*, 2016.
- [12] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, pages 651–673. PMLR, 2018.
- [13] Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv preprint arXiv:2104.08212*, 2021.
- [14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.
- [15] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.
- [16] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018.
- [17] YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *International Conference on Robotics and Automation (ICRA)*, 2018.
- [18] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, Silvio Savarese, and Li Fei-Fei. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, 2018.
- [19] Ajay Mandlekar, Jonathan Booher, Max Spero, Albert Tung, Anchit Gupta, Yuke Zhu, Animesh Garg, Silvio Savarese, and Li Fei-Fei. Scaling robot supervision to hundreds of hours with roboturk: Robotic manipulation dataset through human reasoning and dexterity. *arXiv:1911.04052*, 2019.
- [20] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *international conference on robotics and automation (ICRA)*. IEEE, 2016.
- [21] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1134–1141. IEEE, 2018.
- [22] Pratyusha Sharma, Lekha Mohan, Lerrel Pinto, and Abhinav Gupta. Multiple interactions made easy (mime): Large scale demonstrations data for imitation. In *Conference on robot learning*, pages 906–915. PMLR, 2018.
- [23] Stephen Tian, Suraj Nair, Frederik Ebert, Sudeep Dasari, Benjamin Eysenbach, Chelsea Finn, and Sergey Levine. Model-based visual planning with self-supervised functional distances. *arXiv preprint arXiv:2012.15373*, 2020.
- [24] Sarah Young, Dhiraj Gandhi, Shubham Tulsiani, Abhinav Gupta, Pieter Abbeel, and Lerrel Pinto. Visual imitation made easy. *arXiv e-prints*, pages arXiv–2008, 2020.
- [25] Tianhe Yu, Chelsea Finn, Annie Xie, Sudeep Dasari, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot imitation from observing humans via domain-adaptive meta-learning. *arXiv preprint arXiv:1802.01557*, 2018.
- [26] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *arXiv preprint arXiv:2001.06782*, 2020.
- [27] Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. *IEEE Transactions on Robotics*, 36(4):1307–1319, 2020.
- [28] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5628–5635. IEEE, 2018.