Few-shot Multitask Adaptation using Gradient-based Task Affinity

Jerry Chan
Computer Science
Stanford University
jerryyc@stanford.edu

John So
Computer Science
Stanford University
johnso@stanford.edu

Abstract

Robotics tasks share similar structure, but prior works in multi-task learning often are ran at scale with large data collection, where data scaling laws have shown that performance improves with increasing amounts of data. Recent work has shown that in lower demonstration regimes, multi-task training in policy learning fails to improve task performance. Motivated by this, we aim to leverage the inherent shared structure between robotic manipulation tasks for few-shot task adaptation in lower data regimes.

We present **ATA**, an efficient pipeline to improve few-shot task adaptation without collecting more data. Our key insight is that the structure of the tasks, such as proprioceptive motion, manipulated objects of interest, and task language description, are somewhat captured by the loss across a spatial and semantic trajectory. We leverage this information using a gradient-based lookahead method (TAG [7]) to quantify the difference between a target task and a set of pre-training tasks, and use it as a heuristic of task similarity. Lastly, we filter the pre-training tasks from the scores, and re-train on the filtered dataset. Notably, we do not leverage spatial inductive biases to enable few-shot policy learning; **ATA** is general and makes no assumptions about the underlying tasks.

We conduct experiments on the *Object* suite of LIBERO [16], a generalization benchmark, under a transfer learning setting – we first assume access to a large pre-training task dataset, and a target task dataset. We demonstrate that when given the same amount of demonstrations, multi-task training and training only on the target task perform about equally. When considering adapting to a target task with few (10) demonstrations, we show that **ATA** with fewer total demonstrations attains better performance than both training on only the target task, and training with all available demonstrations. Lastly, we additionally analyze the effectiveness of **ATA**'s heuristic and provide interpretations.

In conclusion, we demonstrate that in a low-data regime, it is important to consider task similarity in order to maximize the performance from the dataset. We believe that our work and task similarity is orthogional to, and can be combined with, approaches in data scaling and pre-training. We hope that our work inspires future work considering the task composition of policy learning datasets to maximize performance.

Our code may be found at `https://github.com/johnrso/libero_330`.

1 Introduction

Robotics currently is entering an era of foundation models, where powerful, large scale, pre-trained LLMs and VLMs enable semantic generalization, and at times emergent behaviors unseen in the training data. These large models are often trained on large multi-task embodied datasets, consisting of thousands of hours of data collected over many months [1, 3, 2]. However, at times it may be unfeasible to collect many demonstrations. Concurrent work aims at simplifying and scaling the demonstration collection process [29, 24, 17, 6], but such methods are only applicable in certain domains. For real-world applications, policies must be robust enough to function even in a few-shot setting.

Additionally, while it is understood that synergistic multi-task learning has the potential for improved performance or emergent behaviors in policy learning [2, 28, 10], the effectiveness of multi-task pre-training is unclear under lower data regimes; multi-task learning is impacted by factors such as the dataset balance of tasks, and the *affinities* between tasks. It is arguable that many of the successes of large-data multi-task schemes primarily benefit from scaling laws [12, 2, 5], and less so from the multi-task nature of the problem. Notably, a recent study [16] on various lifelong approaches has demonstrated that in a low data regime, such approaches perform worse overall on forward transfer than naively training on all of them sequentially, motivating the need to look into better lifelong adaptation schemes to exploit these synergies.

In this paper, we aim to address the challenge of data-efficient adaptation for robotic manipulation for downstream tasks. Towards this direction, we propose ***Affinity-based Task Adaptation*** (**ATA**), a method to efficiently determine similar tasks to a target task, and leverage the data for few-shot task adaptation. At the core of our method, we analyze the task affinity between a pre-training task and a target task by investigating the gradient of networks trained on pre-trained tasks with respect to the target task. We then use the information to adjust the task mixture by pruning tasks with respect to a similarity heuristic, and train a final policy on the filtered dataset. Our method is conceptually simple and does not make assumptions about the problem structure, making it a general method for policy learning. We believe that this task pruning is a novel contribution which will accelerate new task learning, especially in low-data regimes.

We summarize our contributions below:

- We propose Affinity-based Task Adaptation, a flexible and efficient method for few-shot adaptation in policy learning.
- We conduct experiments on a limited-demonstration robotics generalization benchmark, and demonstrate that, contrary to popular belief, training on a filtered dataset outperforms simply training on all available data.
- We demonstrate that our method approximates optimal task groupings, and that it is important to optimally filter the dataset.

2 Related Work

Data Efficient Robot Learning Previous works in data efficient learning, in both single-task and multi-task settings, often impose strong structural biases in observation or action spaces, such as physics [21] or geometry [20, 19, 9, 8], which limits the general application of these methods. Our work instead aims to make minimal assumptions about the structure of the task; our method is general, and compatible with any visuomotor framework.

Another popular paradigm is to leverage the semantic or geometric priors captured in large pretrained models (i.e. DINO, CLIP) for visual generalization [15, 2]. These priors enable large scale zero-shot or few-shot generalization to geometry and concepts. We do not consider internet-scale visual pre-training in our work, as we do not primarily consider zero-shot generalization. We believe that our paper is orthogonal, and may be combined with these methods to improve performance.

Multi-task Synergies and Affinity In policy learning, auxiliary losses are commonly used to encourage meaningful representations or leverage inductive priors [18, 25, 23]. We aim to adopt notions of task synergies not at the loss level, but at the dataset level; whereas these works learn different heads for different output modalities, we consider pruning tasks with the same input

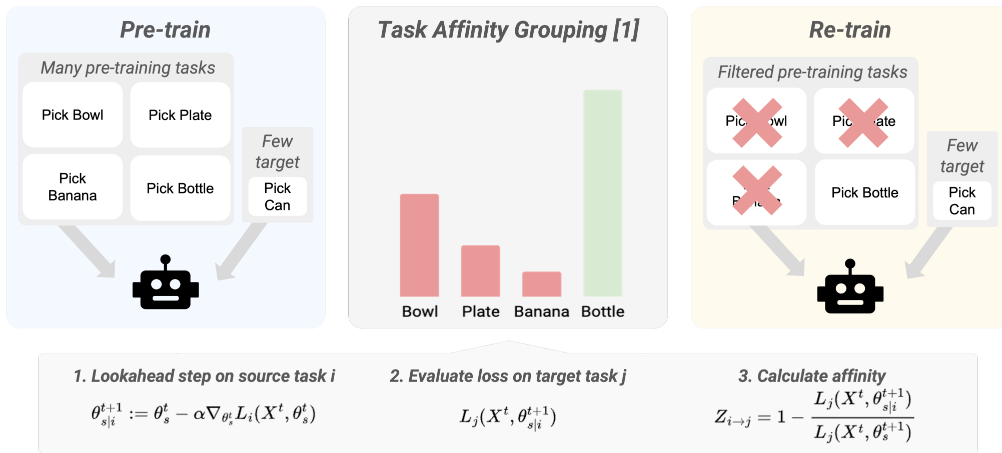


Figure 1: An overview of **ATA**. We first pre-train a model on the entire multi-task dataset. We then approximate task similarity using a method adopted from TAG [7]: for each pre-training task, we perform a *lookahead step* by performing gradient updates with respect to the pre-training task's dataset. We then approximate the task affinity as a comparison of the target task's loss pre- and post-lookahead. We then re-train the model on the filtered dataset.

and output space. Along these lines, several schemes aim to improve the forward transfer policy performance onto target tasks by leverage large amounts of pre-training interactions to bootstrap learning with a small set of target interactions [4, 22, 11].

Another line of work attempts to quantify the synergy between tasks using loss and gradient-based heuristics beyond simply training on all data. Regularization based approaches [14, 26] aim to restrict the network's gradient updates across tasks, which can be used during pre-training to find a good initialization, or to avoid negative transfer. Most related to our work are methods which leverage fully trained networks for heuristics, either by training individual task-specific networks [27] or a large multi-task network [7]. We aim to extend these methods to a policy learning setting.

3 Method

We present our method, Affinity-based Task Adaptation (**ATA**). Our hypothesis is that robotics tasks, even with vastly different goals or initial states, share *structure*. For example, even if manipulating different objects, a policy can learn general reaching motions from shared structure in proprioception information. Objects with similar appearances, or even similar grasping affordances, may also share most structure than those which don't. For example, we may expect grabbing a carton of milk and a carton of juice to be more similar compared to grabbing a banana.

Combining tasks without consideration for structure will harm the model's performance. A successful policy must learn diverse, potentially multi-modal behavior. This becomes challenging with the data imbalance and data coverage problems present in few-shot task adaptation. For example, the task may fail to learn to grasp milk cartons if there are only 10 demonstrations of grasping a milk carton from over 400 demonstrations. Thus, in order to maximize performance, it is important to only leverage tasks with most similar structure.

Given few demonstrations of a target task and a large pre-training dataset, **ATA**, aims to efficiently leverage structure from pre-training tasks in order to improve downstream task adaptation. We detail **ATA** in the following sections.

3.1 Task Specification

The input of the model is a set of pre-training tasks and a target task. Specifically, we define a task X using a language description ℓ, and a set of trajectories τ_X. Each trajectory is a sequence of timesteps consisting of an RGB observation o_t, proprioceptive information (joint states) p_t, and the corresponding action a_t. In our work, tasks share the same input and output space; each task is defined specifically by its language description. Qualitatively, tasks differ in terms of initial state and goal distributions. For example, tasks may involve manipulating different objects between varying spatial locations.

3.2 Affinity-based Task Adaptation

In order to maximize performance on the target task, **ATA** consists of three phases.

1. **Pre-training**: We pre-train a randomly initialized model on all available data.
2. **Grouping and Filtering** We measure task affinity, and select the best performing tasks for re-training. All other pre-training tasks are discarded.
3. **Filtered Dataset Re-training** We then re-train the model from scratch using only the target data, and the filtered pre-training data.

The pipeline is visualized in Figure 1. Note that in step 3, re-training from scratch can be expensive as more tasks are introduced. An alternative to be explored in future work would be simply fine-tuning the pre-trained model on the set of filtered and target tasks.

Trajectory-based Task Affinity At the core of our method is determining task affinity. Determining optimal task groupings is an NP-hard problem; thus, we aim to find an *efficient* heuristic of task affinity. Task specification is a difficult problem, especially in robotics. Language specifications lack fine-grained, trajectory level understanding of motion; image-based specifications, while rich in context, likely require more data to generalize. Thus, we are motivated to use a heuristic which is both general and rich in phyiscal context. Our key insight is that similar tasks likely contain more similar gradients across the entire trajectory. When considering the loss across demonstrations, we observe on pre-training data that the loss tends to spike at certain time intervals. These intervals correspond to the task-specific observations and actions (such as the visual representation of object or committing to motion towards the object). Meanwhile, the loss is lower for aspects such as motion towards the object, which is shared across different tasks. We provide an example in Section 3.

Task Affinity using TAG [7] Motivated by this observation, we use a gradient-based heuristic to measure task affinity. To measure task affinity, we adopt the TAG metric [7]. TAG is an efficient method of measuring task similarity, which can then be used downstream for pruning and grouping. Given trained parameters of a model θ_s, the TAG metric calculates an *affinity score* between two tasks, τ_i and τ_j. In order to do so, TAG calculates a *lookahead step* by performing n gradient steps:

$$\theta_s^{t+1}|i = \theta_s^t - \alpha \nabla_{\theta_s^t} L_i(X_i, \theta_s^t) \tag{1}$$

Algorithm 1: Affinity-based Task Adaptation

Data: Pre-training Tasks X_{pre}, Target Tasks X_{target}, number of trajectories k, number of gradient steps n, number of filtered tasks t
Result: Target task policy π_{target}
affinities = [];
θ_s^{iter} = Fit($X_{\text{pre}} \cup X_{\text{target}}$) ; `// Pre-train on all tasks`
for *Task j in X_{target}* **do**
  `// Sample` k `trajectories and lookahead for` n `steps per task`
  for *Task i in X_{pre}* **do**
    for *iter in range(n)* **do**
      $\theta_{s|i}^{\text{iter}} = \theta_s^{\text{iter-1}} - \alpha \nabla_{\theta_s^{\text{iter-1}}} L_i(X_i[:k], \theta_s^{\text{iter-1}})$
    affinities += [1 - $\frac{L(X_j, \theta_{s|i}^{\text{n}})}{L(X_j, \theta_s^{\text{iter}})}$]
Sort X_{pre} by descending affinities ; `// Filter and retrain on top` t
π_{target} = Fit($X_{\text{pre}}[:t] \cup X_{\text{target}}$);
return π_{target};

Given the lookahead model for task i, we calculate the transfer affinity of task i onto task j using a ratio of their losses. Let $L_i(X_i, \theta)$ denote the loss of parameters θ onto the task i. The affinity score is denoted by the following:

$$Z_{i \to j} = 1 - \frac{L_i(X_i, \theta_s^{t+n}|j)}{L_i(X_i, \theta_s^t)} \tag{2}$$

The second term is the ratio of task j's loss before and after training on task i. Thus, if $Z_{i \to j}$ is positive, training on task i reduces task j's loss, and is considered as positive forward transfer. Likewise, a negative $Z_{i \to j}$ indicates negative forward transfer.

Compared to TAG as proposed, we adopt TAG for the specific learning problem of few shot adaptation. Additionally, we assume that tasks have a unified input/output space, as opposed to learning a separate predictor for each task. The authors calculate TAG as a running mean across training. We instead adopt TAG at the end of pre-training. We additionally modify TAG with our key insights in consideration. Notably, for each task, we sample a set of trajectories from the demonstration dataset.

Using the affinity score Z obtained from TAG, we then filter the top k tasks and re-train the model. The entire pipeline for **ATA** is visualized in Algorithm 1.

4 Experiment Setup

For our experiments, we adopt a Behavior Cloning (BC) scheme. Specifically, we train a transformer-based policy [13, 16] which takes in temporal sequences of proprioception, RGB image, and BERT task embedding, and outputs an action distribution parameterized as a Gaussian Mixture Model. We optimize the model end-to-end using negative log likelihood. Please refer to LIBERO [16] for architecture-specific details.

Design Decisions For TAG, we specifically sample 2 trajectories for each task, and perform 5 gradient updates for the lookahead step as design decisions. We attempt to analyze the effect of different hyperparameters of TAG in our experiments. For the re-training step, we first pre-train on the top $k = 3$ tasks, and finetune on the target task. We leave more sophisticated methods of re-training the network to future work.

4.1 Benchmark

We conducted our experiments on LIBERO-Object [16], a benchmark containing 10 object pick-place tasks. The LIBERO benchmark focuses on generalization across different axes; LIBERO-Object focuses on object level generalization, with similar goal states for all trajectories and tasks. This makes LIBERO-Object ideal for demonstrating that **ATA** can improve target task performance by reasoning about visual appearance and motion.

Each task has 50 expert demonstrations and is associated with a short text instruction (i.e. "Pick up milk and put it in the basket"). The initial state of the demonstrations and in the simulation are generated randomly with unique target object positions and random non-target objects, so the agent has to learn to condition on text prompt and recognize different objects. Figure 2 shows an illustration of the benchmark.

4.2 Pre-training, Adaptation, and Evaluation

For the pre-training step, we break down all the demonstration trajectories from the given set of pre-training tasks into small sequences to form our pre-training dataset. The model is trained on those sequences in random order for 50 epochs. We decided not to include any evaluation during the pre-training as simulations are time-consuming and we didn't observe significant overfitting during the pre-training phase. However, we did experiment with different numbers of epochs and found model usually achieves the best policy with 35 to 50 epochs of training in terms of success rate on pre-train tasks.

For the adaptation step, we followed the assumption of limited target task demonstrations and randomly selected 10 demonstrations from the target task as our target dataset. The model was trained

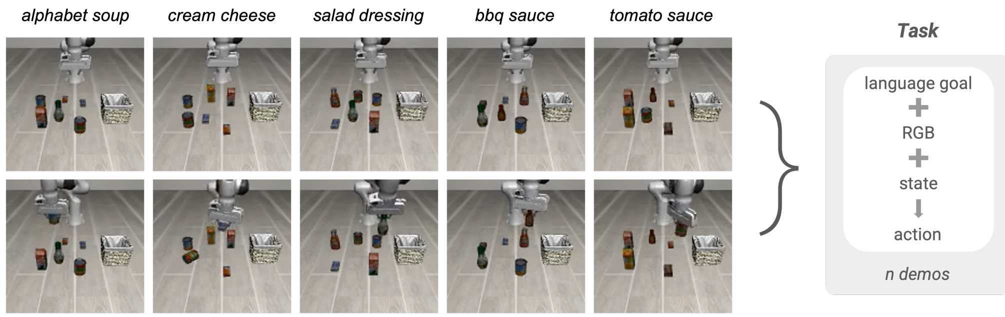


Figure 2: An example of tasks in LIBERO-Object. Tasks are required to differentiate between different objects on the same pick and place skill, where the initial state is randomized. These tasks share much structure, in terms of the motion and proprioception information. The benchmark contains 50 demonstrations per task, and 10 tasks.

for 5 epochs. With much fewer demonstrations, it's easy to overfit the model to the target dataset. Therefore, we conducted evaluation at the end of every epoch with 20 simulation trials and selected the checkpoint with the highest success rate on the 20 trials as the final model.

For all methods, we take the final adapted model and evaluate the final model with an additional 50 simulation trials to reduce the variance. We report the success rate over the 50 trials as our performance metric.

4.3 Baselines

In this section, we outline the baseline approaches of selecting the pre-train task sets to establish points of comparison for the performance achieved by ATA. We compared ATA to the three naive approaches:

- Learning from Scratch (**LFS**): Directly train on the target task without any pre-training
- Multi-task pre-training (**Multi**): Pre-training with all the tasks except the target task.
- Random task grouping (**Random**): Pre-training on three randomly selected, non-target tasks.

5 Experiments

5.1 Exploratory Experiments

Multitask Training We started by evaluating the effectiveness of multitask training and pre-training. This experiment aims to see if the model's performance on the target task improves from multi-task pre-training on similar tasks under the scenario with sufficient data on the target task. All 50 demonstrations were available to the model, and we experimented with two training methods:

1. **LFS**: 50 epochs of training on the target task.
2. **Multi-task Complement Pre-training**: 50 epochs of multi-task pre-training on all the non-target tasks, followed by fine-tuning on the target task with 5 epochs.

Method (50 demos)	Avg. Success
LFS	**0.81**
Multitask Pre-training	0.70

Table 1: Comparison between training methods with 50 demonstrations per task. Notably, training with all tasks (500 total demonstrations) underperforms compared to training only on the 50 tasks for each demonstration.

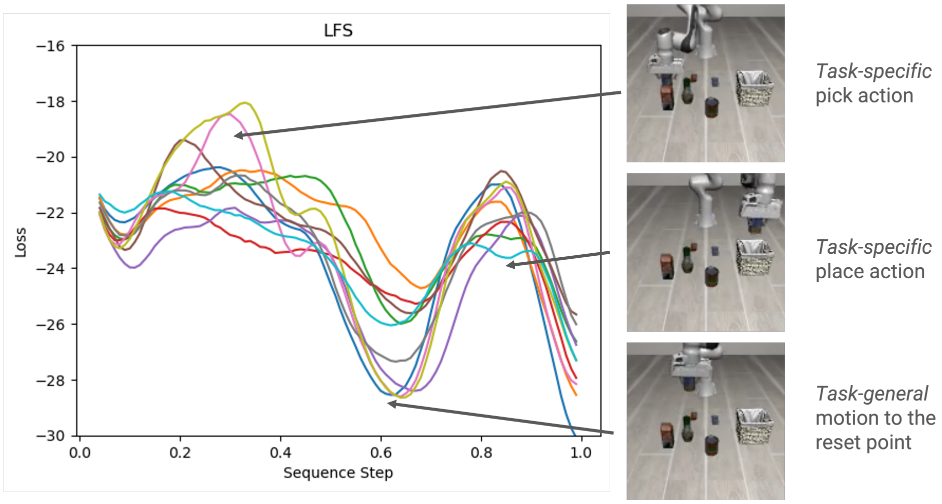


Figure 3: Visualization of loss across tasks in LIBERO-Object. We note that the loss spikes at areas of multimodality or task-specific actions, such as motion towards the object of interest.

We evaluated each method with 20 simulation trials on each of the 10 tasks and reported the average success rate in Table **??**. Contrary to the belief that more data leads to better performance, Multi-task Complement Pre-training underperforms compared to LFS, which has a magnitude less data. This observation underscores the necessity of a more curated pre-training method and pre-training task selection.

Loss Visualization To better understand the task dataset, we visualize the loss over demonstrations and compared the pattern across tasks. We applied the model trained with LFS and computed the loss on 50 demonstrations from the corresponding target task. The results are shown in Figure 3. Each line represents the loss of one task in the LIBERO Object benchmark. To better capture the pattern of various lengths of demonstrations, we normalized the time step by the length of the demonstration and averaged the loss at each normalized time step across demonstrations.

We observed that the loss of each task shares a similar pattern. The loss spikes when it's picking up the object, drops when the arm is moving the object toward the basket, and spikes again when it's about to place the object. As discussed in Section 3.2, this motivates reasoning about the task structure through the loss function of the converged network. By performing updates on an entire trajectory, we expect that similar tasks will decrease the spikes further compared to dissimilar tasks.

Number of Demonstrations With LFS achieving decent performance with 50 demonstrations, we tried to determine the number of target task demonstrations under which LFS is ineffective and can benefit from additional training data from another task. To mimic a low data setting, we train an LFS baseline with 10% (5), 20% (10), and 50% (25) demonstrations and measured the performance with 20 simulation trials. Results are shown in Table . As expected, performance drastically decreases with the decrease in the number of demonstrations. For experiments involving few-shot task adaptation,

Number of Demos	Avg. Success
LFS-50	0.81
LFS-25	0.66
LFS-10	0.15
LFS-5	0.01

Table 2: Comparison of LFS methods trained with varying numbers of demonstrations. With 10 demonstrations, the performance noticeably drops.

we use 10 demonstrations per target task as a design decision, as it does not demonstrate complete failure in completing the task.

5.2 Task Affinity Adaptation

We now compare **ATA** against the baseline methods under a few-shot task adaptation setting. We provide 10 demonstrations for the target task, and 50 demonstrations for each pre-training task. The results are visualized in Table 3.

We perform much better than **LFS**, which may partially be attributed to having more data. Notably, we perform better than **Multi**, despite having a much more narrow dataset. This supports our hypothesis that training on similar tasks results in increased performance. Additionally, **ATA** begins to approach the performance of training on 50 demonstrations for the target task (**LFS-50**). One interpretation of this is that as the data distribution becomes more narrow, fewer demonstrations are necessary to improve performance; 50 in-domain demonstrations (**LFS-50**) and 60 similar demonstrations (**ATA**) are comparable, but 460 potentially dissimilar demonstrations (**Multi**) is not. Lastly, **Random** underperforms compared to **Multi**. This highlights the importance of selecting positively similar tasks.

Few-shot Method	Avg. Success
LFS	0.145
Multi	0.625
Random	0.455
ATA	**0.775**

Table 3: Comparison between **ATA** and baseline methods in a few-shot task adaptation setting. Specifically, we provide 50 demonstrations per pre-training task and 10 target task demonstrations. **ATA** outperforms baselines.

5.3 Task Grouping Methods

To determine the effectiveness of our task similarity heuristic, we additionally ablate on the number of pre-training tasks to choose, including selecting the **Top 1** most similar task, **Top 3** most similar tasks, and **Top 5** most similar tasks. As a baseline, we also compare to **Random**.

As shown in Table 2, all affinity-based grouping methods outperform randomly selecting pre-training tasks by a significant margin, and are competitive with multi-task training on all of the data. The results show our method successfully ranked tasks based on their benefit to target task adaptation, and that selecting tasks with positive transfer is important with limited data. The effectiveness is further justified by the performance decline as we incorporate less similar tasks from the pre-training tasks.

Grouping Method	Avg. Success
Top 1	**0.775**
Top 3	0.740
Top 5	0.605
Random 3	0.455

Table 4: Comparison between pre-train task selection methods. **ATA** performs on par with multi-task training with varying number of tasks chosen. Notably, **Random** underperforms compared to **Multi**, highlighting the necessity of a strong affinity heuristic.

5.4 Visualizing TAG Similarity

To interpret our results, we visualize the affinity score between all tasks in Figure 4a. Some of the results of the affinity matrix are interpretable; for example, "cream cheese" corresponds well with "butter", as the objects are similar geometrically. However, there are also other results which point out flaws in our methodology. For example, ketchup has a negative affinity with itself; this is because datapoints from the trajectory are stochastically sampled. When performing gradient descent on the entire trajectory instead, all tasks have highest affinity with themselves.

Additionally, to compare the task affinity source directly with the effectiveness of pre-training on the task, we gathered the target task success rate of all target task and source task pairs. This serves as a proxy for the optimality of

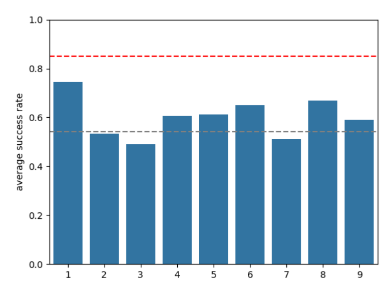
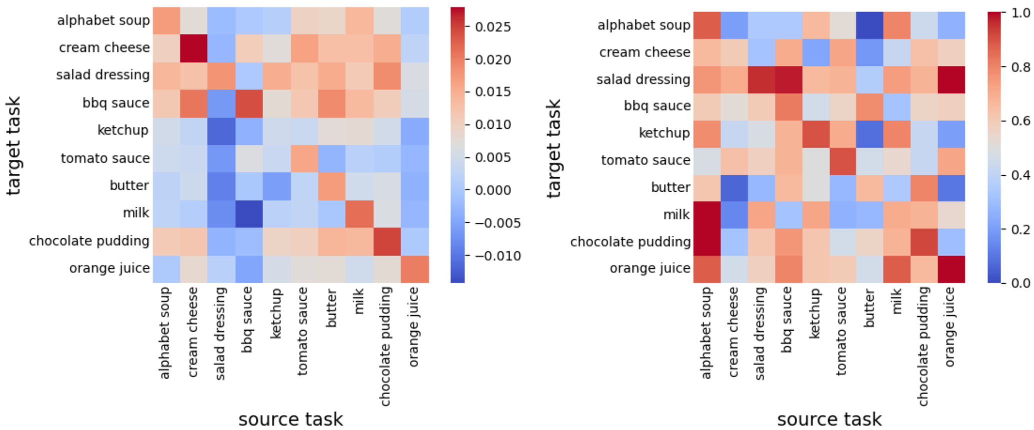


Figure 5: Average success rate vs. task affinity score ranking. We observe that the highest affinity score corresponds to the highest success rate. Red denotes highest success rate. Gray denotes success rate with random task pairings.



(a) Task Affinity Score

(b) Success Rate

Figure 4: Comparison between task affinity score and one-to-one transfer learning success rate. We use the success rate as a proxy for true task affinity, but the metrics are not aligned well. This may be a result of stochasticity in the training and evaluation process.

each pairing of single pre-training task to single taget task. For each pair, we pre-trained a model on the source task with 50 demonstrations and fine-tuned the model on the target task with 10 demonstrations. Each model is then evaluated on the target task for 50 trials. The success rates are reported as a heatmap in Figure 4b. We expected a correlation between the task affinity scores and the success rates but didn't observe any. There's high stochasticity in both TAG and the model training, fine-tuning, and evaluation process; more experiments are needed for a definitive conclusion.

With the success rate of all target and source task pairs, we plot out the average success rate of choosing the K most similar tasks using the task affinity score in Figure 5. Each bar represents the average success rate of choosing the K most similar task as pre-train tasks. Notice that we ruled out the pairs with the same task as the source and target. The red dashed line marks the best possible average success rate when we pick the best source task for each target task. The gray dashed line marks the expected average success rate if we choose the pre-train task at random. According to the figure, pre-training on the task with the highest task affinity score results in the highest average success rate, approaching the highest possible score. However, we didn't observe a clear pattern from the rest of the plot. The overall ambiguous results mark space for improvement in using task affinity scores as a proxy of the effectiveness of cross-task transfer learning.

6 Conclusion

In this paper, we presented **ATA**, a method to accelerate few-shot task adaptation by reasoning about task similarity. By using a gradient-based similarity approach, we reason about task structure at a more granular level than text and at a higher level than raw images. We demonstrate through experiments that our method does indeed perform better than naive pre-training in a limited data regime. Looking forward, there are several design decisions that are worth investigating, such as hyperparameters for TAG (i.e. the number of trajectories used, the number of gradient updates, etc.) which we aimed to investigate. Most importantly, we want to emphasize that the scope of our work primarily emphasizes the lower data regimes; while it is possible to investigate scaling with few demonstrations, it is unseen if our method scales with large datasets, at which point neural scaling laws [12] have demonstrated significant improvements. In spite of this, we hope that our work raises questions about how data is used even with larger scale datasets, and how large-scale data approaches can gain even further performance by imposing structural biases on the same, existing datasets.

Contributions

We copy the work split from our project proposal with updated numbers. Both members worked together on tasks and split the work about evenly; we discuss specific contributions below.

Tasks	Jerry	John
Project Proposal	0.5	0.5
Simulation Environment Setup	0.5	0.5
Implement Task Adaptation and Evaluation	0.5	0.5
Design and Implement Task Similarity Analysis	0.5	0.5
Design and Implement Task Adaptation Pipeline	0.5	0.5
Design and Execute Experiment	0.5	0.5
Final Report	0.5	0.5

Jerry Chan Jerry worked on implementing the experiment pipeline on top of the LIBERO codebase, running and designing experiences, and visualizing the results. He contributed to the Experiment Setup and the Experiment sections of the paper.

John So John primarily worked with setting up the task similarity metric and task adaptation pipeline, and experimented with the hyperparameters of the method. He also did literature review. John also wrote parts of the paper, including Introduction, Related Works, Method, and some analysis in Experiments.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can, not as i say: Grounding language in robotic affordances, 2022.

[2] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023.

[3] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023.

[4] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. 2019.

[5] Open X-Embodiment Collaboration, Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, Antonin Raffin, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Brian Ichter, Cewu Lu, Charles Xu, Chelsea Finn, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Chuer Pan, Chuyuan Fu, Coline Devin, Danny Driess, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Federico Ceola, Fei Xia, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Giulio Schiavi, Hao Su, Hao-Shu Fang, Haochen Shi, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jaehyung Kim, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jiajun Wu, Jialin Wu, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jitendra Malik, Jonathan Tompson, Jonathan Yang, Joseph J. Lim, João Silvério, Junhyek Han, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Zhang, Keyvan Majd, Krishan Rana, Krishnan Srinivasan, Lawrence Yunliang Chen, Lerrel Pinto, Liam Tan, Lionel Ott, Lisa Lee, Masayoshi Tomizuka, Maximilian Du, Michael Ahn, Mingtong Zhang, Mingyu Ding, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Pannag R Sanketi, Paul Wohlhart, Peng Xu, Pierre Sermanet, Priya Sundaresan, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Sherry Moore, Shikhar Bahl, Shivin Dass, Shuran Song, Sichun Xu, Siddhant Haldar, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan

Welker, Stephen Tian, Sudeep Dasari, Suneel Belkhale, Takayuki Osa, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xuanlin Li, Yao Lu, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yueh hua Wu, Yujin Tang, Yuke Zhu, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zhuo Xu, and Zichen Jeff Cui. Open X-Embodiment: Robotic learning datasets and RT-X models. `https://arxiv.org/abs/2310.08864`, 2023.

[6] Hongjie Fang, Hao-Shu Fang, Yiming Wang, Jieji Ren, Jingjing Chen, Ruo Zhang, Weiming Wang, and Cewu Lu. Low-cost exoskeletons for learning whole-arm manipulation in the wild, 2023.

[7] Christopher Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning, 2021.

[8] Stephen James and Andrew J. Davison. Q-attention: Enabling efficient learning for vision-based robotic manipulation, 2022.

[9] Stephen James, Kentaro Wada, Tristan Laidlow, and Andrew J. Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation, 2022.

[10] Zhenyu Jiang, Yifeng Zhu, Maxwell Svetlik, Kuan Fang, and Yuke Zhu. Synergies between affordance and geometry: 6-dof grasp detection via implicit representations, 2021.

[11] Ryan Julian, Benjamin Swanson, Gaurav S. Sukhatme, Sergey Levine, Chelsea Finn, and Karol Hausman. Never stop learning: The effectiveness of fine-tuning in robotic reinforcement learning. 2020.

[12] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

[13] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision, 2021.

[14] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, mar 2017.

[15] Xingyu Lin, John So, Sashwat Mahalingam, Fangchen Liu, and Pieter Abbeel. Spawnnet: Learning generalizable visuomotor skills from pre-trained networks, 2023.

[16] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

[17] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations, 2023.

[18] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017.

[19] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.

[20] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[21] Stone Tao, Xiaochen Li, Tongzhou Mu, Zhiao Huang, Yuzhe Qin, and Hao Su. Abstract-to-executable trajectory translation for one-shot task generalization, 2023.

[22] Homer Walke, Jonathan Yang, Albert Yu, Aviral Kumar, Jedrzej Orbik, Avi Singh, and Sergey Levine. Don't start from scratch: Leveraging prior data to automate robotic reinforcement learning. 2022.

[23] Philipp Wu, Arjun Majumdar, Kevin Stone, Yixin Lin, Igor Mordatch, Pieter Abbeel, and Aravind Rajeswaran. Masked trajectory models for prediction, representation, and control, 2023.

[24] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators, 2023.

[25] Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. 2019.

[26] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. 2020.

[27] Amir R. Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[28] Andy Zeng, Shuran Song, Stefan Welker, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning, 2018.

[29] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023.