Language-Informed Transfer Learning for Embodied Household Activities

Yuqian Jiang [1*], **Qiaozi Gao** [2], **Govind Thattai** [2], **Gaurav Sukhatme** [2,3]

[1] The University of Texas at Austin, [2] Amazon Alexa AI, [3] University of Southern California
jiangyuqian@utexas.edu, {qzgao, thattg}@amazon.com, gaurav@usc.edu

Abstract

For service robots to become general-purpose in everyday household environments, they need not only a large library of primitive skills, but also the ability to quickly learn novel tasks specified by users. Fine-tuning neural networks on a variety of downstream tasks has been successful in many vision and language domains, but research is still limited on transfer learning between diverse long-horizon tasks. We propose that, compared to reinforcement learning for a new household activity from scratch, home robots can benefit from transferring the value and policy networks trained for similar tasks. We evaluate this idea in the BEHAVIOR simulation benchmark which includes a large number of household activities and a set of action primitives. For easy mapping between state spaces of different tasks, we provide a text-based representation and leverage language models to produce a common embedding space. The results show that the selection of similar source activities can be informed by the semantic similarity of state and goal descriptions with the target task. We further analyze the results and discuss ways to overcome the problem of catastrophic forgetting.

Introduction

Domestic service robots have been envisioned to help in a variety of household activities. Imagine a single robot that can be versatile enough from tidying up the rooms to playing with kids. Such a robot not only requires the sensing, navigation, and manipulation capabilities, but also needs to intelligently combine these skills to perform each activity as requested by the users.

Since every home is different, a simple library of preprogrammed tasks will hardly serve the purpose. For example, when a user wants to clean the kitchen cupboard, the specific goal conditions they would like to achieve will depend on their personal preferences and constraints of the environment. Does the robot re-arrange the dishes in a certain pattern? Does the robot dust the outside of the cupboard? The reality is that there could be an infinite number of combinations of goals, and a robot will most likely have to learn to solve new goals after it is deployed in the individual homes.

In this paper, we study the problem of learning novel user-specified household activities for a service robot that is shipped with pre-trained policies for a set of standard activities. We propose to learn the new activity by transferring from the policy of a similar activity. Our hypothesis is that the transfer can be more efficient than learning the new activity from scratch if their initial state and goal conditions are similar. Intuitively, a robot should be able to learn *putting away cleaned dishes* efficiently if it has a good policy for *cleaning kitchen cupboard*. Further, we can measure activity similarities by leveraging language models to embed their state and goal descriptions.

We test our hypothesis using the BEHAVIOR benchmark (Srivastava et al. 2021). BEHAVIOR simulates a large number of household activities for an embodied AI to learn. We first present a reinforcement learning (RL) approach to solve a subset of activities from scratch. The approach leverages text descriptions of the agent's current state and goal to allow the policies to operate in a common state space. We then initialize the learner with each of the pretrained policies when training it on a new activity, and evaluate the hypothesis that the transfer performance corresponds to the semantic similarity between the activity text descriptions. We present some initial results to show the potential of this approach for enabling versatile and adaptive home robots.

Related Work

Transfer learning leverages the knowledge learned in a source domain to improve the performance of a learner on the target domain. Transfer learning in reinforcement learning has been studied to transfer knowledge between different Markov Decision Processes (MDPs) (Zhu, Lin, and Zhou 2021; Taylor and Stone 2009). While many approaches are evaluated in tasks with the same high-level goal and only different configurations in Mujoco, navigation, and Atari domains (Barreto et al. 2017; Schaul et al. 2015), a few recent transfer learning approaches have demonstrated positive transfer between distinct Atari games (Rusu et al. 2016; Fernando et al. 2017). Soemers et al. introduces an approach that transfers policy and value networks between distinct board games that have different action spaces (Soemers et al. 2021). Encouraged by these successes, we propose to transfer RL policies among distinct embodied household activities which require high-level long-horizon reasoning about a large variety of goal conditions. Further, this work proposes to use language models on activity descriptions to inform the

[*]Work completed during an internship with Amazon Alexa AI.

selection of source domains.

BEHAVIOR is a benchmark where embodied AI solutions are evaluated on household activities in a realistic physics simulation. The activities are selected from the American Time Use Survey to reflect the real distribution of household chores. There has been very little success using RL to solve BEHAVIOR in its original setting (Srivastava et al. 2021). In this paper, the method of providing the text-based, fully observable state representation is most similar to the work done by Shridhar et al. for the ALFRED benchmark (Shridhar et al. 2021).

Approach

Our approach consists of two steps. In the first part, we introduce a text-based state representation for a RL agent to efficiently learn a set of diverse BEHAVIOR activities from scratch. The state representation is also in a common embedding space to allow easy knowledge transfer to other activities. In the second part, we introduce how these pre-trained policies are re-used for learning new activities, and test our hypothesis that the semantic similarity between activity descriptions can be used to predict transfer performances.

Learning Single Activities

We introduce a different RL formulation from the original one in the BEHAVIOR benchmark, in order to speed up learning these activities using standard RL algorithms.

Text-Based State and Goal Representation Given the low RL performance in the original setting of BEHAVIOR, we take a similar approach to ALFWORLD (Shridhar et al. 2021) by providing full observability of the logical state in the form of language. The simulator backbone of BEHAVIOR extracts logical predicates that describe the current states and relations of all objects in the world. We filter the logical predicates to the ones relevant to the activity, and use a template to generate text descriptions of the logical state. Similarly, the goal conditions are represented with text descriptions. Figure 1 shows the initial state for one instance of the *cleaning kitchen cupboard* activity. Figure 2 shows the goal definition of the *cleaning kitchen cupboard* activity. There are two goals: 1) dust every cabinet and 2) move all cups to one cabinet and all bowls to the other. For the example initial state, there are two ways to ground the second goal based on how the cups and bowls are assigned to cabinets, and each grounding leads to a distinct set of subgoals.

Action Primitives The action space includes a set of discrete action primitives implemented in BEHAVIOR: GRASP, TOGGLE ON, TOGGLE OFF, OPEN, CLOSE, PLACE INSIDE, PLACE ON TOP. Each action primitive takes a parameter that refers to an object. For example, PLACE INSIDE(cabinet_0) means the robot will put the object currently in its gripper into the cabinet.

Problem Formulation We formulate a BEHAVIOR activity as a Markov Decision Process denoted by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, R)$. $\mathcal{S}$ is the space that consists of tokenized state and goal descriptions. $\mathcal{A}$ is the space of action primitives, parameterized by the objects relevant to the activity. $\mathcal{P}(\cdot|s, a)$ is the unknown stochastic transition probabilities. $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function. Given the grounded subgoals of the activity, R is defined as follows: if a is not executable at s, $R(s, a, s') = -1$; otherwise, let $g(s)$ be the number of subgoals satisfied in the state s, $R(s, a, s') = \frac{g(s')-g(s)}{\text{total number of subgoals}} \cdot c$ where c is a large constant. The reward function penalizes choosing action primitives that are not executable, such as TOGGLE OFF(cup_0), and generously rewards achieving new subgoals. The objective is to learn a policy $\pi : \mathcal{S} \to \mathcal{A}$ that maximizes the expected total reward.

top_cabinet_47 is *dusty*. top_cabinet_47 is *next to* cup_1. bottom_cabinet_41 is *dusty*. bottom_cabinet_41 is *on top* cup_0. bottom_cabinet_41 is *next to* cup_0. bottom_cabinet_41 is *next to* bowl_1. countertop_26 is *under* bath_towel_0. countertop_26 is *in reach of* robot. countertop_26 is *in same room as* robot. bath_towel_0 is *on top* countertop_26. bath_towel_0 is *in reach of* robot. soap_0 is *on top* countertop_26. soap_0 is *in reach of* robot. bowl_0 is *on top* countertop_26. bowl_0 is *in reach of* robot. bowl_1 is *inside* bottom_cabinet_41. bowl_1 is *next to* bottom_cabinet_41. cup_0 is *inside* bottom_cabinet_41. cup_0 is *next to* bottom_cabinet_41. cup_1 is *inside* top_cabinet_47. cup_1 is *next to* top_cabinet_47. room_floor_kitchen_0 is *in reach of* robot. room_floor_kitchen_0 is *in field of view of* robot.

Figure 1: An example initial state of *cleaning kitchen cupboard*

For every cabinet, the following is NOT true:
the cabinet is dusty.
For at least one cabinet, for every bowl, the bowl is inside the cabinet, and the following is NOT true:
cup1 is inside the cabinet.
For at least one cabinet, for every cup, the cup is inside the cabinet, and the following is NOT true:
bowl1 is inside the cabinet.

Figure 2: An example goal definition of *cleaning kitchen cupboard*

Actor-Critic Policy The policy can be trained by policy gradient methods such as PPO (Schulman et al. 2017). Figure 3 shows the actor-critic architecture. We use a pre-trained DistilBert model (Sanh et al. 2020) to tokenize and encode the input text. The actor network outputs a tuple of the action primitive index and the object index.

Transfer Learning

Since the aim of this work is not to achieve top performances on BEHAVIOR, but rather to explore the connection between transfer performance and activity similarity, we adopt a straightforward method to re-use pre-trained policies and compare the learning curves.

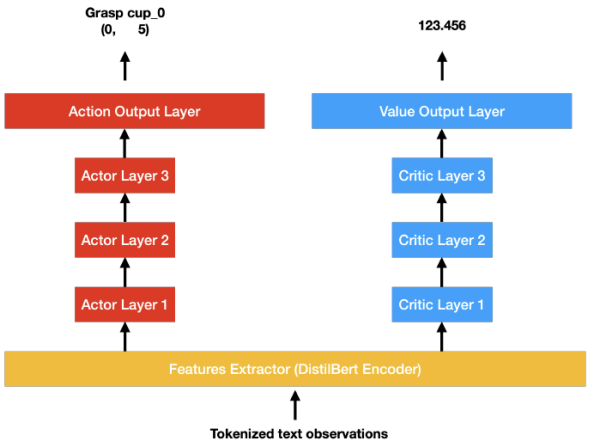

Figure 3: Actor-critic network architecture for learning one BEHAVIOR activity.

State and Action Mappings Since S is a space of tokenized state and goal descriptions, the state space is common for all activities. However, the action primitives are parameterized by the objects in the scene, so the action space can have different sizes. To re-use a policy for a new activity, we copy all the weights in the network (Figure 3) except for the actor output layer. Then we resize the actor output layer to match the new action space and randomly initialize it before training.

Semantic Similarity Given a new activity with an initial state and a set of goal conditions, the text-based state and goal representation constructed for the MDP formulation is also a unique description of this activity. We use the pre-trained SimCSE model (Gao, Yao, and Chen 2022) to embed activity descriptions, and compute the consine similarity between the embeddings of any pair of activities.

Transfer Metric We evaluate the transfer performance of each pair of activities by the transfer ratio (or transfer score) metric (Taylor and Stone 2009; Rusu et al. 2016). The transfer ratio measures the ratio of the total reward given to the transfer learner and the total reward given to the non-transfer learner after a certain number of training steps. It can be computed by the ratio of the area under the transfer learning curve over the area under the non-transfer learning curve.

Experiments

We choose to study 7 activities from BEHAVIOR: *storing food*, *cleaning kitchen cupboard*, *putting away Halloween decorations*, *collect misplaced items*, *putting away cleaned dishes*, *locking every window*, *cleaning microwave oven*.

The policies are trained with the PPO algorithm as implemented in the stable-baselines3 library (Raffin et al. 2021). An episode terminates when all the subgoals are achieved or the maximum number of steps (64) has been taken. The hyperparameter c in the reward function is set to 200. As a result, the highest total reward of an episode is 200, i.e. achieving all subgoals without any penalty. The lowest total reward is -64, i.e. always executing invalid actions.

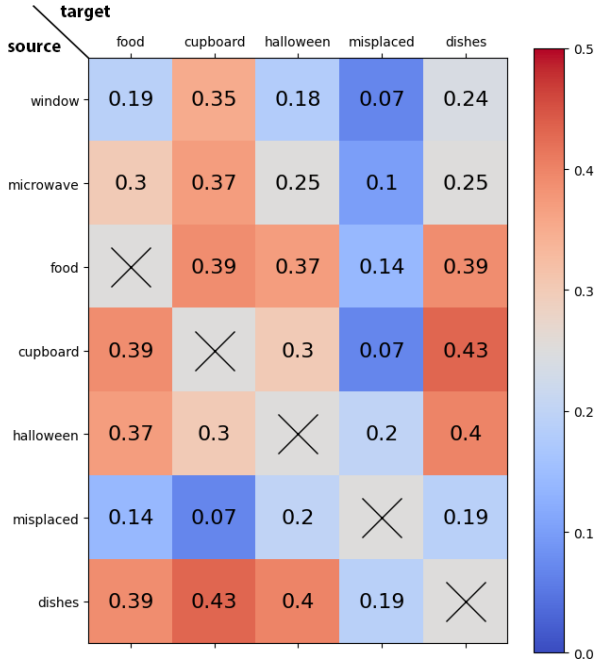

Figure 4: Semantic similarities between source and target activities.

Training from Scratch To obtain a policy for each activity, we train for 512 episodes and take the top performing policy out of 3 runs. Table 1 shows the mean reward per episode achieved at the end of training by the top policy for each activity. Note that there is a wide gap between how well these activities are solved by our policies. The policies for *locking every window* and *cleaning microwave oven* are near optimal, whereas the policy for *cleaning kitchen cupboard* never manages to achieve all subgoals during training. This difference is due to the solution length and the stochasticity of executing the action primitives. Some activities require executing more than 10 actions in the correct order, and some actions (e.g. grasp) have a low success rate in producing the desired effects. The uncertain action effects reflect the challenge for real robots, since the task-level policy should know how to recover when there are failures during execution.

Since it's much faster to learn *window* and *microwave* than the other activities, they are only used as source tasks but not target tasks in the transfer experiments below.

Semantic Similarity Figure 4 summarizes the semantic similarity in a matrix. Each row is a source activity and each column is a target activity. A high number (or warm color) means the descriptions of the two activities are close in the embedding space, whereas a low number (or cool color) indicates that the embeddings are distant. It may not be intuitive why some activities are more similar than others based on their abbreviated names. For example, *storing food*, *cleaning kitchen cupboard*, *putting away dishes*, *putting away Halloween decorations* all involve moving ob-

food	cupboard	halloween	misplaced	dishes	window	microwave
-8.5	-34.5	1.1	4.0	-7.0	196.0	189.0

Table 1: Mean reward per episode achieved at the end of training.

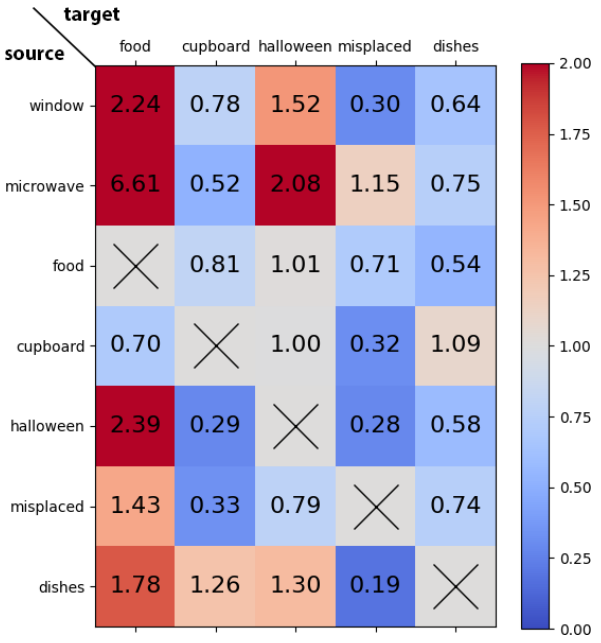

Figure 5: Transfer ratios of the first 80 episodes.

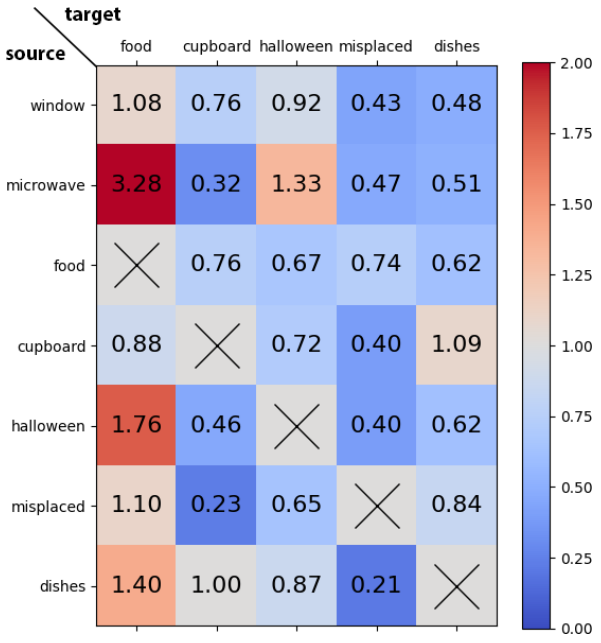

Figure 6: Transfer ratios of the first 160 episodes.

jects into cabinets, so their similarity scores are high when taking into account the full descriptions.

Transfer Ratios Figure 5 presents the transfer ratio matrix after 80 episodes (or about 5000 steps). A ratio above 1 indicates positive transfer, i.e. the transfer learner receives higher total reward during training. Comparing with the similarity score matrix, we can make two observations. First, a high-quality source policy can lead to positive transfer, even if the activity is not similar. The activities *storing food* and *putting away Halloween decorations* (two difficult tasks) are not similar to *locking every window* or *cleaning microwave oven* (two easy tasks), but we see high transfer ratios in the first two rows of their columns. Second, for each target activity, higher semantic similarity has a higher chance of positive transfer. *Cleaning kitchen cupboard* and *putting away cleaned dishes* have a high semantic similarity (0.43). The only positive transfer to *cupboard* was from *dishes* and vice versa. On the other hand, *collecting misplaced items* is semantically very different from all other activities, and gets some of the worst transfer ratios.

Catastrophic Forgetting While there are clear signs that re-using policies can jump start learning a new activity, the benefits of transfer quickly disappear as catastrophic forgetting takes place. Figure 6 shows the transfer ratios after 160 episodes (or about 10,000 steps). The general observations in Figure 5 still hold, but the ratios are getting lower and there are fewer cases of positive transfer.

For future studies, one of the ideas to transfer knowledge without suffering from the conflicting goals is by decoupling the task-independent knowledge from the task-dependent knowledge. In the case of household activities, there is a lot of shared knowledge across activities, especially the preconditions and effects of actions. For example, TOGGLE OFF(cup_0) is an invalid action in any activity. To this end, successor features (Barreto et al. 2017) and universal value function approximation (Schaul et al. 2015) are both methods to learn representations that decouple the dynamics from the rewards so they will generalize over different goals. Meanwhile, there are neural representations designed to avoid catastrophic forgetting. Progressive neural nets (Rusu et al. 2016) add a new column of network while preserving the weights learned in previous tasks.

Conclusion

We propose that home robots can efficiently learn novel household tasks from similar but distinct activities, and present our analysis in the BEHAVIOR benchmark. Our experiments show encouraging results: activity similarity measured by language embeddings can be used as a predictor for transfer performance, and a high-quality source policy of an easy but different activity can sometimes lead to a jumpstart. We also observe the problem of catastrophic forgetting and suggest future research in this direction.

References

Barreto, A.; Dabney, W.; Munos, R.; Hunt, J. J.; Schaul, T.; van Hasselt, H. P.; and Silver, D. 2017. Successor Features for Transfer in Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Fernando, C.; Banarse, D.; Blundell, C.; Zwols, Y.; Ha, D.; Rusu, A. A.; Pritzel, A.; and Wierstra, D. 2017. Pathnet: Evolution Channels Gradient Descent in Super Neural Networks. *arXiv preprint arXiv:1701.08734*.

Gao, T.; Yao, X.; and Chen, D. 2022. SimCSE: Simple Contrastive Learning of Sentence Embeddings. arXiv:2104.08821.

Raffin, A.; Hill, A.; Gleave, A.; Kanervisto, A.; Ernestus, M.; and Dormann, N. 2021. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research*, 22(268): 1–8.

Rusu, A. A.; Rabinowitz, N. C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; and Hadsell, R. 2016. Progressive Neural Networks. arXiv:1606.04671.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2020. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. arXiv:1910.01108.

Schaul, T.; Horgan, D.; Gregor, K.; and Silver, D. 2015. Universal value function approximators. In *International conference on machine learning*, 1312–1320. PMLR.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*.

Shridhar, M.; Yuan, X.; Côté, M.-A.; Bisk, Y.; Trischler, A.; and Hausknecht, M. 2021. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. arXiv:2010.03768.

Soemers, D. J. N. J.; Mella, V.; Piette, E.; Stephenson, M.; Browne, C.; and Teytaud, O. 2021. Transfer of Fully Convolutional Policy-Value Networks Between Games and Game Variants. arXiv:2102.12375.

Srivastava, S.; Li, C.; Lingelbach, M.; Martín-Martín, R.; Xia, F.; Vainio, K.; Lian, Z.; Gokmen, C.; Buch, S.; Liu, C. K.; Savarese, S.; Gweon, H.; Wu, J.; and Fei-Fei, L. 2021. BEHAVIOR: Benchmark for Everyday Household Activities in Virtual, Interactive, and Ecological Environments. arXiv:2108.03332.

Taylor, M. E.; and Stone, P. 2009. Transfer Learning for Reinforcement Learning Domains: A Survey. *Journal of Machine Learning Research*, 10(7).

Zhu, Z.; Lin, K.; and Zhou, J. 2021. Transfer Learning in Deep Reinforcement Learning: A Survey. *arXiv:2009.07888 [cs, stat]*.