Learning Abstract Representations of Agent-Environment Interactions

Tanmay Shankar
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, United States
tanmay.shankar@gmail.com

Jean Oh
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, United States
jeanoh@cmu.edu

Abstract: In learning from demonstration, tasks are generally well defined and known to a learner, *e.g.*, opening a door. In this paper, we present an unsupervised approach for learning abstract representations of common manipulation tasks such that, given a demonstration of an unknown task, a high-level task strategy can still be transferred from a teacher to a learner. Building on prior work of learning temporal abstractions of agent behaviors, *e.g.*, trajectories of an arm, we propose to extend these to temporal abstractions that represent how an agent interacts with the objects in their environment, *e.g.*, a robot pushing a door. We show promising results that suggest such interaction abstractions could provide a unified basis for representing human and robot task strategies presented at https://sites.google.com/view/interaction-abstractions potentially facilitating new ways of learning from demonstration.

Keywords: Temporal abstractions, representation learning

1 Introduction

Consider an amateur cook learning a new dish by watching a chef demonstrate how to make a similar dish on YouTube; even amateurs can achieve excellent results doing so. Humans owe this aptitude for 'learning by imitation' to reasoning abstractly about human behaviours–including their own–and the task at hand. People ignore irrelevancies (ex. differences in kitchens), focus on patterns environmental change needed (ex. steps of recipes), and skill-sequences that effect these environmental changes (ex. techniques of chefs). In doing so, people abstractly represent task strategies; *i.e.*, the sequence of skills and patterns of environmental changes needed to accomplish the task. The prospect of equipping *robots* with these abilities is enticing. Such abstract representations of task strategies would allow us to view human and robot task strategies from a unified perspective. This in turn would enable robots to learn to solve various tasks by watching human demonstrators solve similar tasks in different environments from their own.

The robot learning community has made efforts towards realizing various parts of this vision. Sivakumar et al. [1], Arunachalam et al. [2], Ye et al. [3], Qin et al. [4], Wu et al. [5] engineer mappings between human and robot state to facilitate imitation of human videos. Shankar et al. [6], Smith et al. [7] have sought to learn such correspondences between humans and robots without manual specification, resorting to representation alignment machinery such as Zhu et al. [8], or unsupervised domain adaptation machinery [9, 6]. While successful in their own right, these approaches seek to transfer individual states or actions across domains, and lack a higher level understanding of the behaviors at hand. The community has attempted to introduce such higher level understanding in the form of abstractions. Gelada et al. [10], Li et al. [11], Zhang et al. [12], Hansen-Estruch et al. [13] works on learning *state* abstractions that facilitate ignoring irrelevant components of environments, and making analogies across various environment and task instances [13]. Sutton et al. [14], Eysenbach et al. [15], Sharma et al. [16], Shankar et al. [17], Shankar and Gupta [18], Krishnan et al. [19], Fox et al. [20] learn *temporal* abstractions of agent behavior, that facilitate reasoning over longer term behaviors.

Despite making significant advances in building high-level understanding of agent behavior, these works a significant pitfall. These works either learn abstractions over agent behaviors [15, 16, 17, 18, 19, 20], or over changes in environment state [13], but do not consider both together. While Sutton et al. [14], Sharma et al. [16] maintain notions of pre-conditions and effects, these notions are often hand-crafted into the abstractions, or are simplistic conditions on state. Further, approaches that *learn* temporal abstractions of behaviors are often unaware of the patterns of environmental and object state change they induce (i.e., their effects). Conversely, most environmental state abstractions are unaware of the behaviors that caused them. As a result, traditional approaches that use such abstractions to solve tasks typically need to perform a search for an appropriate sequence of abstractions to solve the task - a difficult problem that requires highly engineered heuristics to solve.

In this paper, we argue that it is therefore important to understand interactions between agents (humans and robots alike) *and* their environments *together*. To do so, we build temporal abstractions of interactions. These interaction abstractions differ from their behavioral and environmental counterparts in that they maintain *explicit* notions of an agent and environment (or objects in it). By doing so, we hope to equip agents with a *high-level* understanding of the effects of their behaviors, and therefore an understanding of which of their behaviors are needed to affect desired environmental changes. By then *aligning* understandings of effects of human and robot behaviors, we hope to transfer strategies of solving tasks from human demonstrators to robots. We do so by first applying a robot skill learning framework [18] to learn temporal abstractions of environmental state. We then combine this with the original robot skills, to build a framework for interaction abstractions. We show promising initial results that these interaction abstractions would aid transfer between human and robot task strategies.

2 Approach

2.1 Background – Learning Behavioral Abstractions

We first describe an important building block of our work–a behavioral abstraction framework. We consider behavioral abstractions, or skills, are a representation of an agent acting consistently for a temporally extended period. Examples of such skills include a person stirring (a pot), or a person flipping an object such as a pancake, etc. In this paper, we specifically consider the behavioral abstraction framework of Shankar and Gupta [18]; though any such framework could be used [15, 21, 19, 17, 22]. Shankar and Gupta [18] learn behavioral abstractions, or skills, of agents from demonstrations in an unsupervised manner. Their method first represents robot skills as continuous latent variables z^r (subscript r depicts of the robot), and introduce a Temporal Variational Inference (TVI) to infer these skills or latent variables. Consider an agent trajectory $\tau^r = \{s_1^r, a_1^r, ...s_{n-1}^r, a_{n-1}^r, s_n^r\}$, where s_t^r is the state of the agent, a_t^r is the agent's action at time t, and n is the length of the trajectory. TVI trains a variational encoder $q^r(z|\tau^r)$ that takes as input a agent trajectory τ^r and outputs a sequence of skill encodings $z^r = \{z_1^r, z_2^r, ...z_k^r\}$, where $k < n$ is a learnt number of skills executed. TVI also trains a latent conditioned policy $\pi(a|s, z^r)$ that takes as input robot state s, and the chosen skill encoding z, and predicts the low-level action a that the agent should execute. TVI trains q^r and π to reconstruct the actions observed in the trajectory τ^r. We direct the reader to [18] for a thorough description of their skill learning approach.

2.2 Building Temporal Abstractions over Environment State

In order to build temporal abstractions of interactions between agents and environment state, one also needs temporal abstractions of environment state. These abstractions exemplify patterns of motion that objects often exhibit, *e.g.*, a bottle cap rotating and moving up away from the bottle, or a kettle being tilted downwards to pour from it. One can imagine how such abstractions are useful –

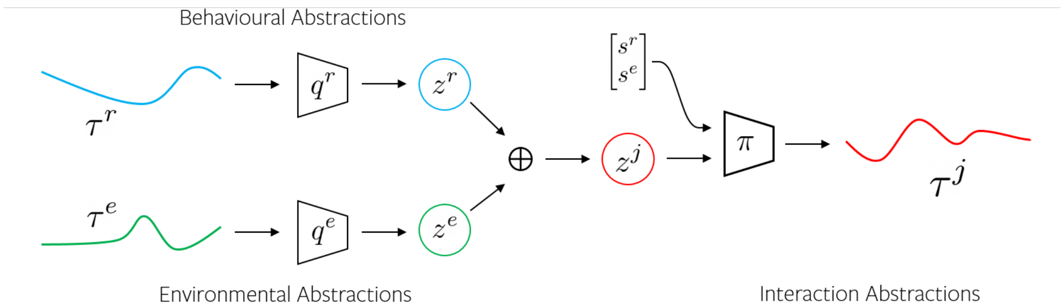

Figure 1: Overview of our approach. We learn a robot skill encoder q^r and an environmental abstraction encoder q^e, that learn representations of behavioral and environmental abstractions z^r and z^e respectively. We then combine z^r and z^e into a joint interaction abstraction z^j. We can then condition robot policy π on interaction abstraction z^j, in addition to state inputs s^r and s^e.

they could be used to specify steps that need to happen in a kitchen recipe, or more generally, describe changes in environmental state that need to occur. In this section, we describe how we adapt the behavioral abstractions of [18] to learn temporal abstractions of environment state. Consider a corresponding trajectory $\tau^e = \{s^e_1, a^e_1, ...s^e_{n-1}, a^e_{n-1}, s^e_n\}$ of *environment* state s^e over time. Here, a^e_t represents the change in environmental state at a given timestep t, rather than a notion of agent action. We can construct an equivalent *environmental* variational encoder $q^e(z|\tau^e)$, that predicts an equivalent sequence of latent encodings $z^e = \{z^e_1, z^e_2, ..., z^e_k\}$, that represent temporally abstract changes in environmental state. We can train an equivalent environment state "policy" π^e, conditioned on the desired environmental abstraction z^e. As in TVI, we train q^e and π^e to reconstruct the change in environmental state a^e_t at every timestep. We present preliminary results showing the efficacy of these environmental state abstractions below.

2.3 Building Interaction Abstractions from Behavioral and Environmental Abstractions

In order to develop a high-level understanding of how agents interact with their environments, we would also like to build temporal abstractions of *interactions*. Examples of such abstractions would include using a cutting skill to chop an onion, or using a flipping skill to turn over an omelette. To reiterate, the proposed interaction abstractions differ from behavioral and environmental counterpart in that interaction abstractions maintain *explicit* notions of an agent and environment (or objects in it), *e.g.*stirring a pot, or flipping a pancake. In contrast, prior behavioral abstractions [18, 19] only maintain *implicit* notions of objects, *e.g.*, stirring or flipping skills. As a result, we view the interaction abstractions from an active perspective; *i.e.*, that the explicit agent effects changes on its environment. In contrast, we view the environmental abstractions above from a passive perspective, in that the state of objects in an environment changes, agnostic of what agent effects these changes.

Given behavioral abstractions z^r and environmental abstractions z^e, we combine these abstractions to form interaction abstractions z^j. We explore several options of combining these abstractions; the simplest is to simply concatenate them, *i.e.*, $z^j = [z^r z^e]$. These interaction abstractions thus contain information of both the agent skill being executed, and its (desired) effect on the environmental state. We can inform the policy of this desired agent skill and desired pattern of environmental change by conditioning the policy π on z^j rather than on z^r alone, *i.e.*, $\pi = \pi(a|s, z^j)$.

We now describe the pipeline of how we construct interaction abstractions z^j. We first sample agent and environmental state trajectories τ^r & τ^e from the appropriate dataset. We then pass these trajectories through their corresponding encoders q^r and q^e, to retrieve latent encodings of these abstractions, $\{z^r_1, z^r_2, ..., z^r_k\}$ & $\{z^e_1, z^e_2, ..., z^e_k\}$ respectively. We concatenate these encodings to form latent encodings of the interaction, $\{z^j_1, z^j_2, ..., z^j_k\}$. We then condition the agent policy π on this desired interaction abstraction $\{z^j\}^k_{t=1}$. During inference, given a pattern of desired interactions, $\{z^j_1, z^j_2, ..., z^j_k\}$, we can condition the agent policy on these desired interactions $\pi(a|s, z^j)$, and query it at the current state s^r, s^e, for the next agent action to execute a^r. We depict this pictorially in fig. 1.

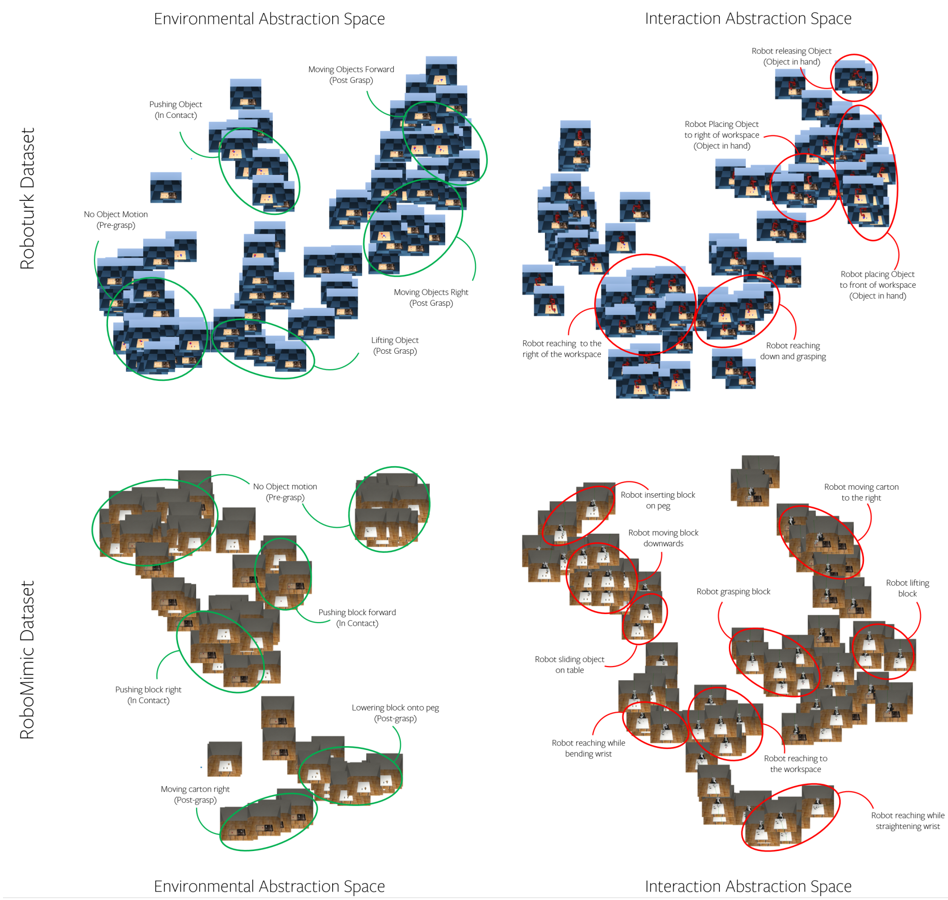

Figure 2: Depiction of preliminary results on both the environmental abstractions (left column), and the interaction abstractions (right column), across Roboturk dataset (top row) and Robomimic dataset (bottom row). These embedding spaces depict abstract representation spaces that model both agent behaviors and the their effects on the environment. In all four cases, note the clustering of similar motions into similar parts of their respective latent spaces. For dynamic versions of these spaces, view https://sites.google.com/view/interaction-abstractions.

During training, we can update encoders q^r & q^e, and policy π, to optimize a reconstruction loss on the state changes observed in trajectories τ^r & τ^e. We specifically optimize the likelihood of observed state changes, as in TVI.

3 Experiments

We would like to answer the two following questions with respect to our proposed abstractions. Firstly, can the learnt environmental and interaction abstractions accurately model interactions between agents and their environments? Second, do the learnt abstractions spaces show promise in facilitating downstream task strategy transfer from humans to robots? In this section, we present preliminary results towards answering these questions.

3.1 Datasets and Experimental Setup

We present results on the Roboturk dataset [23] and the RoboMimic dataset [24], datasets of tele-operated demonstrations collected on the Sawyer robot and the Franka Panda respectively. In addition we also present results on the GRAB dataset [25], which consists of *humans* manipulating various objects themselves. These datasets consist of their respective agents interacting with a variety of different objects, such as milk cartons, cereal, bread, cans, differently shaped pegs, tools, *etc.* Collectively, these datasets span a variety of interactions, including lifting, moving, releasing,

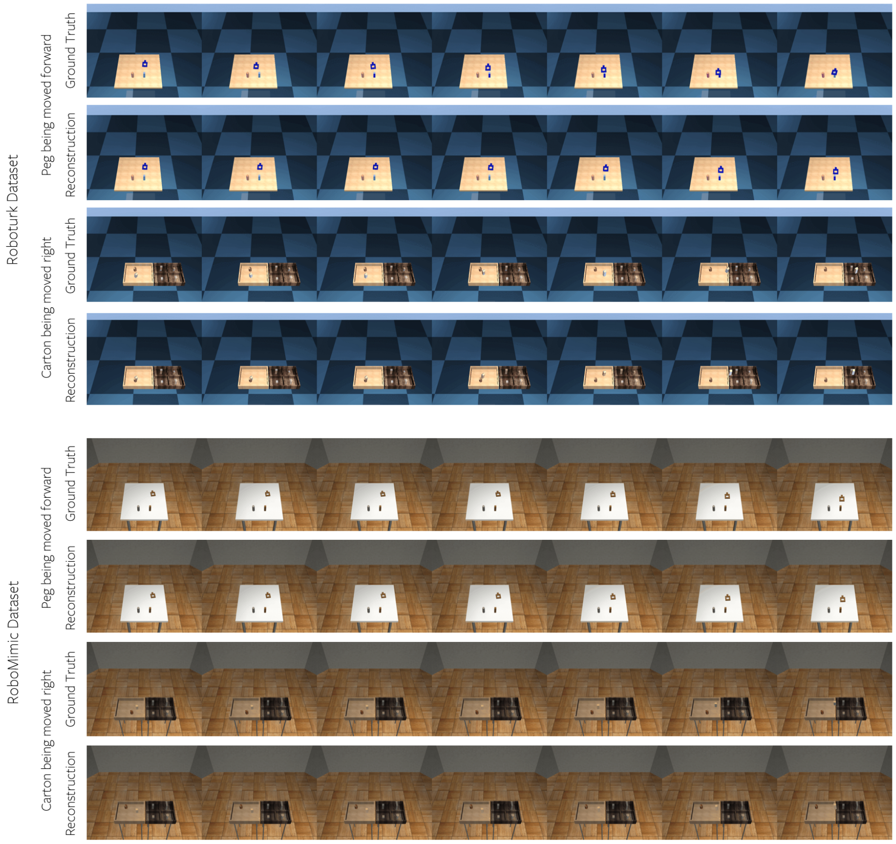

Figure 3: Qualitative results on environmental abstractions. We show two object state demonstrations for each dataset, (labelled ground truth), and their corresponding reconstructions via our learnt abstractions (labelled reconstructions). Note the accuracy of the reconstructions in capturing the object trajectory despite the low dimensional nature of the latent encoding.

pushing, rotating, reorienting, *etc.* We present both qualitative and quantitative results for both the environmental and interaction abstractions.

In each case, we have access to the joint-state of the agents, 8 joint angles for the robots, and a 24 dimensional joint state for the humans. We also make use of the object state in these trajectories, consisting of 6 object pose. In each case we assume the "actions" of the agents are joint state velocities.

3.2 Preliminary Qualitative Results

We first present qualitative results; in the form of visualizations of both individual abstractions and the representation spaces $\mathbb{Z}^e$ & $\mathbb{Z}^j$ of abstractions. We provide static versions of these visualizations in our main paper, and include dynamic visualizations on our project webpage, https://sites.google.com/view/interaction-abstractions.

3.2.1 Environmental Abstractions

We first present preliminary results on applying the behavioral abstraction framework of Shankar and Gupta [18] to learning environmental abstractions. To do so, we consider object state trajectories present in the various datasets, consisting of 6-D pose (position and orientation) of the object. We provide these object state trajectories as input to TVI [18]. TVI then provides us with a latent space $\mathbb{Z}^e$, that represents temporal abstractions of object state. Each z^e in this space $\mathbb{Z}^e$ represents a different pattern of environmental state. In this case, since we represent environment state as object state, each z^e may be thought of as a different pattern of object state.

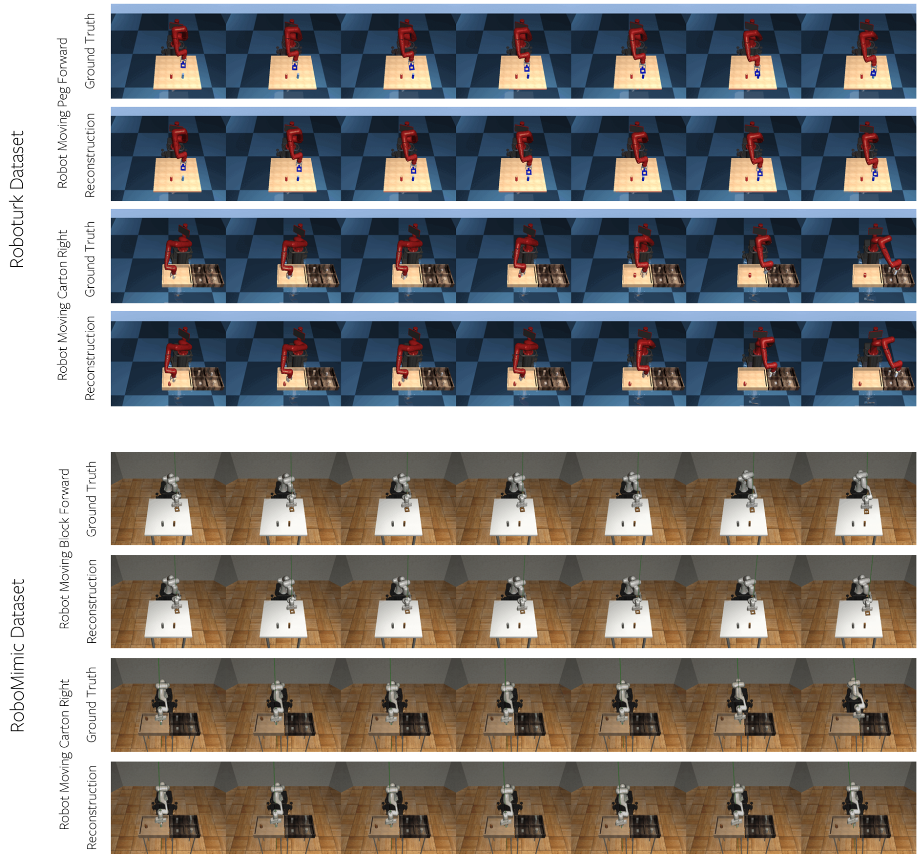

Figure 4: Qualitative results on interaction abstractions. We show two demonstrations for each dataset of the respective robot interacting with objects (labelled ground truth), and their corresponding reconstructions via our learnt abstractions (labelled reconstructions). We visualize the same tasks as in fig. 3, for reasons in section 3.4.2. Note how the robot skills executed *lead* to the environmental abstractions depicted in fig. 3. The reconstructed space is able to capture this, and therefore the overall pattern of motions of agent and environment.

We present a 2D visualization of this space on the left in fig. 2, produced via T-SNE [26]. Note the clustering of similar patterns of motion of objects into similar parts of the latent space. We manually annotate the various clusters of environmental abstractions present. Our environmental abstraction space is able to capture a variety of different object motions, including pushing, lifting, and moving objects.

In addition, we also visualize individual samples of abstractions present in the space $\mathbb{Z}^e$, in fig. 3 and https://sites.google.com/view/interaction-abstractions. We visualize ground truth trajectories (of a peg being moved forward, and a milk carton being moved to the right), and their respective reconstructions via their learnt abstraction encodings z^e. Notice the high fidelity reconstructions of the trajectories, and the similarity in trend of object motions in the ground truth and reconstructed trajectories. We note that while these results are to be expected from TVI [18], this validates the soundness of the learnt *environmental* state abstractions.

3.2.2 Interaction Abstractions

Having verified the soundness of environmental abstractions, we may now analyze how well they combine with prior behavioral abstractions (*i.e.*, robot skills) to form our proposed interaction abstractions. To do so, we feed in trajectories of agent and object state to their respective encoders q^r & q^e, retrieving their predicted latent encodings z^r & z^e respectively, concatenate these encodings

Table 1: Comparison of reconstruction error of environmental and interaction abstractions against baseline approaches for reconstructing environmental and joint agent environmental state respectively. Lower is better.

	Environmental State			Agent & Environment State			
Dataset	LSTM	VAE	EnvAbs (Ours)	LSTM	VAE	IntAbs Joint (Ours)	IntAbs Factored (Ours)
RoboTurk	1.40	1.83	**0.46**	1.78	1.93	0.72	**0.62**
RoboMimic	1.33	1.47	**0.38**	1.58	1.75	0.84	**0.58**
GRAB	1.16	1.38	**0.57**	2.30	2.24	0.93	**0.80**

to form a interaction abstraction encoding z^j, and finally reconstruct the joint agent-object trajectory τ^j from this encoding. As above, each z^j in the interaction abstraction space $\mathbb{Z}^j$ represents a different pattern of interaction between agent and object state.

As above, we present a 2D visualization of this latent interaction abstraction space on the right of fig. 2, also produced by T-SNE. Dynamic visualizations of this are available at https://sites.google.com/view/interaction-abstractions. As in the case of environmental abstractions, the latent space is clustered based on the nature of interactions taking place. In each cluster, the robot and object undergo common patterns of change of state. There is a single cluster that models no interactions between the agent and the robot (*i.e.*, the robot is reaching towards the object and has not made contact with it yet). There are other clusters modeling interactions with the object in-hand, such as moving or lifting the objects. There are additionally clusters where the agent is making or breaking contact with the object, such as the robot releasing the object cluster in the top right. We emphasize that our method is able to capture such clustering of interactions despite being trained without any supervision over the types of interactions, robot skills, or object motions.

In addition to this, we also present visualizations of individual interactions present in the space $\mathbb{Z}^j$ in fig. 4, and in https://sites.google.com/view/interaction-abstractions. For ease of comparison, we visualize the same trajectories as visualized in fig. 3, *i.e.*the *robot* moving a peg forward, and the robot placing the milk carton to the right. Note that in contrast with the above environmental abstractions, where we refer to these object moiton patterns from a passive perspective, here we think of the interaction abstractions as active abstractions, where the robot effects desired change in its environment. We observe in fig. 4 that the interaction abstractions in each case capture *both* the robot skill (or behavioral abstraction) that is executed by the robot, *as well as* the effect it has on environmental state. Particularly, we observe the robot executing skills that result in the environmental state changes observed in fig. 3. Together with our other qualitative results, this suggests our proposed interaction abstractions are capable of abstractly modelling both agent behavior, and their effects on environmental state.

3.3 Preliminary Quantitative Results

In addition to the above qualitative results, we also present preliminary *quantitative* results to further verify the ability of our approach to abstractly model interactions. We do so by measuring the reconstruction error of both the environmental and interaction abstractions, and comparing this against other baseline approaches of predicting trajectories of environmental and joint agent-environment state respectively. We present these results in table 1. Note that we do only compare approaches reconstructing environmental state amongst one another, and not against approaches reconstructing joint agent-environmental state, and vice versa.

We consider two variants of our interaction abstraction approaches. The first is a straightforward application of TVI [18] to reconstructing joint agent-environmental state, represented by joint latent encodings (rather than the factored encodings presented in section 2.3). We term this approach as

IntAbs-Joint. The second is the factored version of interaction abstractions present in section 2.3, termed as IntAbs-Factored. We compare these approaches against two baselines; an auto-regressive LSTM trained to reconstruct trajectories, and a VAE approach that uses a *single* latent variable (across all timesteps) to represent the trajectory.

In the case of environmental state, we observe that our environmental abstractions are more accurate predictors of trajectories than the flat baselines, across all 3 datasets. This trend is mirrored in the case of reconstructing joint agent-environment state, where both interaction abstractions also perform better than the baselines. Further, we observe that our factored interaction abstractions are also able to outperform the joint interaction abstractions, suggesting that our choice of modelling the agent abstractions and environmental abstractions with separate latent variables is appropriate.

3.4 Suitability for Transfer

We now seek to verify whether these abstractions also exhibit characteristics suitable to transfer human to robot strategies.

3.4.1 Factored Encodings for Transfer

We first consider suitability for transfer along the architectural axis. By choosing to adopt a *factored* encoding of the agent and environmental abstractions fig. 1 rather than a joint encoding, our approach possesses a compositional architecture. Consider transferring from a human to robot task strategies. After training interaction abstractions for both human and robot agents $\mathbb{Z}^{j,human}$ & $\mathbb{Z}^{j,robot}$ respectively, we believe we can then transfer between these human and robot interaction abstractions, by swapping out the particular agent encoder q^r for that of another agent $q^{r'}$. While this also requires the alignment of the environmental abstractions associated with both agents, this is a fairly straightforward problem to solve, using machinery such as Cycle-GAN Zhu et al. [27] *etc.*.

3.4.2 Unified Perspectives of Skills and Task Strategies

Our approach can represent similar interactions across different agents well, as exemplified by the similarity of representations of similar tasks across datasets in fig. 3 & fig. 4. By aligning representations of environmental abstractions across these agents (a relatively straightforward problem given machinery like Zhu et al. [8]), we could view such interactions from a unified perspective.

Consider two agents interacting in two domains D_1 & D_2. Given an alignment (or mapping) $f(z^e_{D_1}) \rightarrow z^e_{D_2}$ of environmental abstractions across domains D_1 and D_2, we may align the interaction abstractions $z^j_{D_1}$ & $z^j_{D_2}$ associated with $z^e_{D_1}$ & $z^e_{D_2}$ respectively using the same mapping; thus facilitating easy transfer of task strategies across these domains.

3.4.3 Visualization of task strategies

To further demonstrate the ability of our approach to represent task strategies, we also provide visualizations of task strategies in the latent representation space of abstractions. To do so, we first present a 2-D projection of the latent space $\mathbb{Z}^j$ (as in fig. 2). Given a representation $\{z^j_1, z^j_2, ..., z^j_k\}$ of a task strategy, we then visualize this sequence of z^j_t's overlaid in space $\mathbb{Z}^j$. We present these results at https://sites.google.com/view/interaction-abstractions due to space constraints.

4 Conclusion

In this work, we present a framework for learning abstract representations of interactions between agents and their respective environments. We use these interaction abstractions to represent task strategies adopted by agents. These interaction abstractions could also model forward and inverse dynamics between agents' skills and their effects on the objects in their environments, and transfer task strategies from human demonstrators to robots. We believe this could enable robots to adopt the task strategies of human demonstrators, therefore broadening their repertoire of tasks.

References

[1] A. Sivakumar, K. Shaw, and D. Pathak. Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube. *RSS*, 2022.

[2] S. P. Arunachalam, S. Silwal, B. Evans, and L. Pinto. Dexterous imitation made easy: A learning-based framework for efficient dexterous manipulation. *arXiv preprint arXiv:2203.13251*, 2022.

[3] J. Ye, J. Wang, B. Huang, Y. Qin, and X. Wang. Learning continuous grasping function with a dexterous hand from human demonstrations, 2022.

[4] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang. Dexmv: Imitation learning for dexterous manipulation from human videos, 2022.

[5] Y.-H. Wu, J. Wang, and X. Wang. Learning generalizable dexterous manipulation from human grasp affordance, 2022.

[6] T. Shankar, Y. Lin, A. Rajeswaran, V. Kumar, S. Anderson, and J. Oh. Translating robot skills: Learning unsupervised skill correspondences across robots. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 19626–19644. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/shankar22a.html.

[7] L. M. Smith, N. Dhawan, M. Zhang, P. Abbeel, and S. Levine. AVID: learning multi-stage tasks via pixel-level translation of human videos. *CoRR*, abs/1912.04443, 2019. URL http://arxiv.org/abs/1912.04443.

[8] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[9] C. Zhou, X. Ma, D. Wang, and G. Neubig. Density matching for bilingual word embedding. In *Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Minneapolis, USA, June 2019. URL https://arxiv.org/abs/1904.02343.

[10] C. Gelada, S. Kumar, J. Buckman, O. Nachum, and M. G. Bellemare. DeepMDP: Learning continuous latent space models for representation learning. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2170–2179. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/gelada19a.html.

[11] L. Li, T. J. Walsh, and M. L. Littman. Towards a unified theory of state abstraction for mdps. In *International Symposium on Artificial Intelligence and Mathematics, AI&Math 2006, Fort Lauderdale, Florida, USA, January 4-6, 2006*, 2006. URL http://anytime.cs.umass.edu/aimath06/proceedings/P21.pdf.

[12] Q. Zhang, T. Xiao, A. A. Efros, L. Pinto, and X. Wang. Learning cross-domain correspondence for control with dynamics cycle-consistency. 2021. URL https://openreview.net/forum?id=QIRlze3I6hX.

[13] P. Hansen-Estruch, A. Zhang, A. Nair, P. Yin, and S. Levine. Bisimulation makes analogies in goal-conditioned reinforcement learning. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 8407–8426. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/hansen-estruch22a.html.

[14] R. S. Sutton, D. Precup, and S. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 1999.

[15] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine. Diversity is all you need: Learning skills without a reward function. 2019. URL https://openreview.net/forum?id=SJx63jRqFm.

[16] A. Sharma, S. Gu, S. Levine, V. Kumar, and K. Hausman. Dynamics-aware unsupervised discovery of skills. 2020. URL https://openreview.net/forum?id=HJgLZR4KvH.

[17] T. Shankar, S. Tulsiani, L. Pinto, and A. Gupta. Discovering motor programs by recomposing demonstrations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgHY0NYwr.

[18] T. Shankar and A. Gupta. Learning robot skills with temporal variational inference. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8624–8633. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/shankar20b.html.

[19] S. Krishnan, R. Fox, I. Stoica, and K. Goldberg. Ddco: Discovery of deep continuous options for robot learning from demonstrations. *arXiv preprint arXiv:1710.05421*, 2017.

[20] R. Fox, S. Krishnan, I. Stoica, and K. Goldberg. Multi-level discovery of deep options. *arXiv preprint arXiv:1703.08294*, 2017.

[21] S. Krishnan, A. Garg, S. Patil, C. Lea, G. Hager, P. Abbeel, and K. Goldberg. Transition state clustering: Unsupervised surgical trajectory segmentation for robot learning. In *RR*. 2018.

[22] T. Kipf, Y. Li, H. Dai, V. Zambaldi, A. Sanchez-Gonzalez, E. Grefenstette, P. Kohli, and P. Battaglia. Compile: Compositional imitation learning and execution. In *ICML*, 2019.

[23] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, S. Savarese, and L. Fei-Fei. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, 2018.

[24] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *arXiv preprint arXiv:2108.03298*, 2021.

[25] O. Taheri, N. Ghorbani, M. J. Black, and D. Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *European Conference on Computer Vision (ECCV)*, 2020. URL https://grab.is.tue.mpg.de.

[26] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-sne. 2008.

[27] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017. URL http://arxiv.org/abs/1703.10593.