2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA) | 978-1-6654-4337-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICMLA52953.2021.00091

Use of Embedding Spaces for Transferring Robot Skills in Diverse Sensor Settings

Kazushi Ninomiya
Artificial Intelligence Laboratory
University of Tsukuba
Tsukuba, Japan
ninomiya@ai.iit.tsukuba.ac.jp

Masakazu Hirokawa
Artificial Intelligence Laboratory
University of Tsukuba
Tsukuba, Japan
hirokawa_m@ieee.org

Kenji Suzuki
Artificial Intelligence Laboratory
University of Tsukuba
Tsukuba, Japan
kenji@ieee.org

***Abstract*—We propose a method for learning transferable skills among various state-action spaces for reinforcement learning using embedding spaces. Most existing approaches for robot skill transfer assume similar hardware settings between robots. However, in practical applications, identical hardware settings among different robots is not necessarily guaranteed. This study is an attempt to abstract and represent the characteristics of state-action space caused by differences in hardware, such as sensor settings and actuators, using an embedding space. Using this method, the skills become transferable between different hardware settings without requiring heuristic tuning by experts. In this study, we implement the proposed method on a mobile robot in a two-dimensional plane and demonstrate its effectiveness by performing a navigation task in a simulation environment.**

***Index Terms*—Reinforcement Learning, Transfer Learning, Embedding Space**

I. Introduction

In recent years, deep reinforcement learning has been actively investigated in the field of autonomous robot control as it can acquire more policies of an action value function than heuristic programming by humans without requiring prior knowledge of the surrounding environment [1]. However, unlike software-based applications such as Atari games, applications of deep reinforcement learning on robots in real-world robots remain challenging [2]. Some methods have been proposed to learn simple tasks by a robot in a reasonable time frame [3]; however, a complex task still requires a considerable amount of time to learn [4]. Moreover, a significant number of trials and errors may damage the hardware. Several approaches have been proposed to solve this problem. For instance, imitation learning allows robots to learn policies from expert data without trial and error [5], *Sim-to-Real* enables learning results to be transferred from simulations to actual machines [6] [7]. However, both approaches require a considerable amount of expert data and simulations for each robot type, and they are difficult to adapt to a robot of a different system configuration. The most relevant approach to this study is that by Hausman et al., who proposed a method to embed latent variables in an embedding space by learning the policies for solving source tasks as skill [8]. Subsequently, the target task is solved by combining the embedded skills in the embedding space.

Most of these methods assume that the robot configuration and surrounding environment are consistent between the source and target tasks, as well as aim to solve difficult tasks through knowledge transfer. Hence, when the robot is replaced with another robot of a different hardware configuration, all the source tasks must be relearned by the new robot.

The aim of this study is to propose a method that enables knowledge transfer among robots of different system configurations. According to the taxonomy of Lazaric's survey [10], this study is categorized under "transfer across tasks with different domains." Hence, we extend Hausman's embedding space method to embed knowledge of different tasks together with the characteristics of state-action space in which the tasks were learned. Using this method, after performing a few rounds of training to characterize the state-action space, a robot can perform an unlearned task using the skills learned by different robots.

To verify the proposed method, the transfer of navigation skills in a two-dimensional plane among mobile robots of different sensor settings was addressed via a simulation environment. The results show that the skills were successfully transferred to a robot of a novel sensor setting through the embedding space, and that the robot successfully performed multiple unlearned tasks at zero shot after learning a practice task.

II. Related Work

Transfer learning with respect to reinforcement learning has been recognized and investigated as an important research direction [9] [10] [11]. In particular, knowledge transfer aims to solve more complex target tasks than source tasks in the same environment. However, few studies have focused on knowledge transfer between different environmental settings. Taylor et al. demonstrated the skill transfer between different robots by mapping the states and actions on the source tasks to those of the target task. However, the mapping function must be handcrafted by human experts [12]. Ammar et al. used manifold alignment to obtain a state mapping between the source and target tasks; subsequently, the mapping was used to transfer the optimal state trajectory set of the source task to the target task such that the target policy can use the knowledge, similar to learning from demonstration [13].

978-1-6654-4337-1/21/$31.00 ©2021 IEEE
DOI 10.1109/ICMLA52953.2021.00091

TABLE I
THE SYMBOLS USED HERE IN.

Description	Symbol
State at step i	s_i
Action at step i	a_i
Transition probability	$p(s_{i+1}\|s_i, a_i)$
Reward	$r_i(s_i\|a_i)$
Initial state distribution	$p_0(s_0)$
Weighting term	$\alpha = \alpha_1 + \alpha_2 + \alpha_3$
Entropy term	$H[\ \ ]$
Task ID	t
Sensory ID	e
Latent variable	z or z_t or z_e
Policy	$\pi_\theta(a_i\|s_i, z)$
Embedding network	$p_\phi(z\|t)$ or $p_\omega(z_e\|e)$ or $p_\rho(z_e\|e)$
Inference network	$q_\psi(z\|a_i, s_i^H)$
Policy network parameter	θ
Embedding network parameter	ϕ or ω or ρ
Inference network parameter	ψ

Their method learns the mapping between states, but the measurement formula for distances between states is provided in advance.

A similar study was conducted by Devin et al. [14]. They segregated the policy network into multiple modules based on tasks and robots (modularization) to address unlearned task–robot combinations with zero shots or an insignificant amount of learning (e.g., robot A can solve task B efficiently using knowledge obtained from other combinations: tasks A and B, B, and A). However, because the modules are associated with the task and robot, the modules must be relearned when a new robot is introduced.

III. EMBEDDING ROBOT SKILLS IN DIVERSE SENSOR SETTINGS

A. Preliminaries

The symbols used here in are shown in Table I.

B. Learning Robot Skills

We extend the method proposed by Hausman et al. to enable skill transfer between robots of different system configurations, such as sensor settings and tasks. First, we describe the method of embedding robot skills into the embedding space proposed in [8]. In their study, each task is represented by t, which is expressed as a one-hot vector. Subsequently, a latent variable z is introduced to embed the skills required to solve each task, and the probability distribution of sampling z from t is denoted as $p(z|t)$. In addition, they introduced a policy $\pi(a|s, t)$ as the conditional probability of states s and task t. The objective function is formulated as the earned reward plus entropy regularization.

$$\max_{\pi} \mathbb{E}_{\pi, p_0, t\, \in T} \Bigg[\sum_{t=0}^{\infty} \gamma^i \left(r_t(s_i, a_i) \right. \\ \left. + \alpha H[\pi(a_i|s_i, t]) \,|a_i \sim \pi(\cdot|s, t), s_{i+1} \sim p(s_{i+1}|a_i, s_i) \Bigg] \tag{1}$$

Equation (1) can be expanded to (2) by calculating the lower bound using variational estimation (see [8] for details).

$$\begin{aligned} & L(\theta, \phi, \psi) \\ & = \mathbb{E}_{\pi_\theta(a,z|s,t), t\in T} \Bigg[\sum_{i=0}^{\infty} \gamma^i \hat{r}(s_i, a_i, z, t) | s_{i+1} \\ & \sim p(s_{i+1}|a_i, s_i) \Bigg] + \alpha_1 \mathbb{E}_{t\in T} \left[H[p_\phi(z|t)] \right], \\ & \text{where } \hat{r}(s_i, a_i, z, t) \\ & = \left[r_t(s_i, a_i) + \alpha_2 \log q_\psi(z|a_i, s_i^H) + \alpha_3 H[\pi_\theta(a|s_i, z)] \right] \end{aligned} \tag{2}$$

Equation (2) is written in a form for achieving the following three points.

- A similar probability distribution between the embedding and inference networks.
- A policy that is not extremely decisive.
- A wide distribution of each skill in the embedding space.

The algorithm learns to become maximize (2) by simultaneously learning the policy $\pi_\theta(a_t|s_t, z)$, the embedding network $p_\phi(z|t)$, and the inference network $q_\psi(z|a_t, s_t^H)$, each implemented in a separate neural network. In this study, task t was sampled from the task set T in each learning episode. It has been shown experimentally in [8] that complex target tasks can be solved by additionally learning only the mapping function $f(z|s)$ from the current state in which the skill is to be used.

C. Embedding Robot Skills with Sensor Settings

To transfer skills between robots of different sensor settings, we modified their method, as shown in Fig.1. The flow during learning is as follows: First, tasks and sensor settings are separately embedded as latent variables (z_t, z_e) on the embedding space using embedding networks, and combined as z (Fig.1(a)). Subsequently, the action is predicted using policies as a probability distribution conditioned to the state and the z (Fig.1(b)). Finally, the inference networks predict z from the action and state history (Fig.1(c)), and the loss is calculated for using back propagation.

$$z = \begin{pmatrix} z_t \\ z_e \end{pmatrix} \sim p_\phi(z_t|t), p_\omega(z_e|e) \tag{3}$$

Here, z_t and z_e are the column vectors. To facilitate the transition, we embed tasks and sensor settings into the embedding space using separate embedding networks, $p_\phi(z_t|t)$ and $p_\omega(z_e|e)$, respectively, and concatenate z_t and z_e to form a latent variable z.

In Hauseman's approach, the task and sensor settings were combined into a single ID. This renders it difficult to transfer skills to a robot of different sensor settings, including for solving the same task. Therefore, in the proposed method, the embedded network is modularized for each knowledge group (task and sensor settings in this study) to facilitate the transfer of knowledge regarding tasks and the robot's sensor settings separately, in a manner similar to that presented in [14]. Their

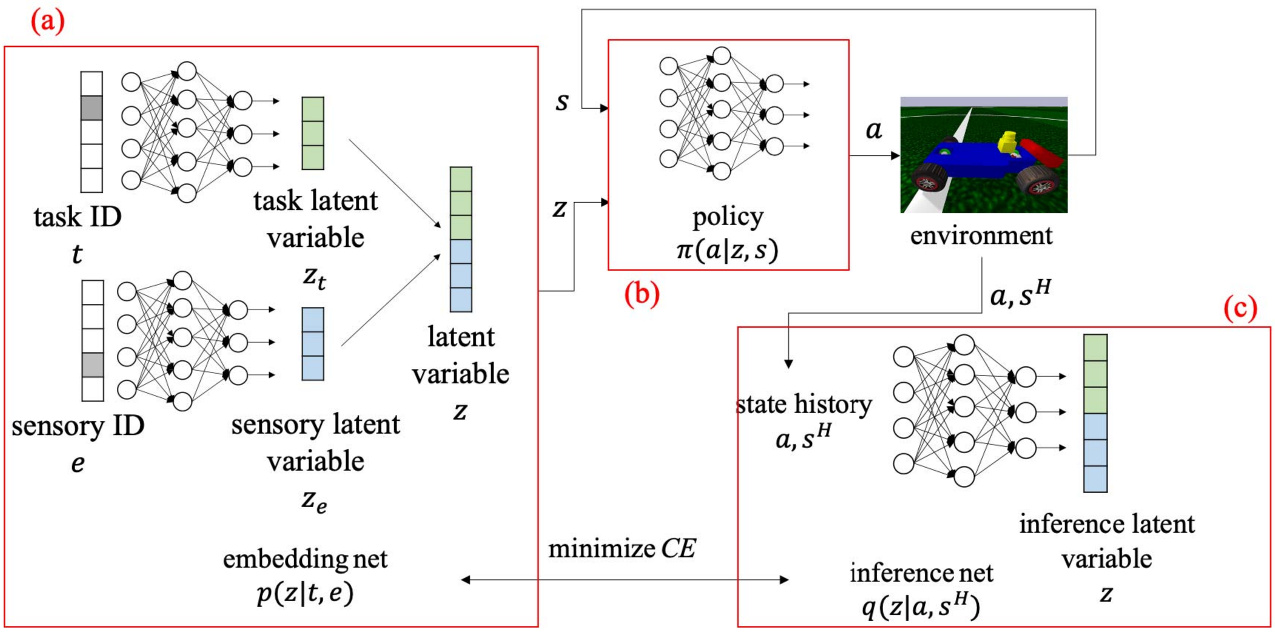

Fig. 1. Overview of the proposed method. (a) Tasks and sensor settings were separately embedded as latent variables (z_t, z_e) in embedding space using embedding networks, and then combined as z. (b) Polices for solving source tasks were learned as a probability distribution conditioned to the state and the latent variable z. (c) Inference networks as a probability distribution conditioned to the state history and the action were learned near embedding network.

modules are separated for each task and robot, and although they can be reused easily, the modules must be relearned completely in situations where the task or robot settings differ even slightly. As an advantage over their approach, the proposed method embeds tasks and sensor settings in a continuous embedding space such that even unlearned settings in the vicinity of the learned task and sensor (robot) settings will be interpolated because of the continuity of the embedding space.

To learn the embedding space, we performed pre-training on multiple tasks using robots of multiple sensor settings. In this regard, four networks were learned: the task embedding network $p_\omega(z_e|e)$, the policy $\pi_\theta(a_t|s_t, z)$, and the inference network $q_\psi(z|a_t, s_t^H)$. Because this learning must be performed by sampling the tasks and sensor settings from the candidates for each episode, we assume that it will be performed in a simulation environment.

D. Transferring Skills to a New Robot

The robot skill is represented on the latent variable z by pre-training, and the policy is represented as a conditional probability of this latent variable. For a sensor setting that has already been trained, latent variables can be obtained by providing the corresponding one-hot vector to the embedded network. However, for a new sensor setting, additional latent variables corresponding to it must be trained. If we use the existing $p_\omega(z_e|e)$ for additional training, then catastrophic forgetting will occur; hence, a new embedding network $p_\rho(z_e|e)$ was introduced and trained for the new sensor setting. In this case, the parameters of the four networks learned during the pre-training were fixed. The task to be used for training is any one of the tasks learned during pre-training.

IV. Experiments

A. Tasks and Sensor Settings

The purpose of the experiment is to demonstrate the feasibility of the proposed method in realizing the following two aspects: 1) the characteristics of different sensor settings can be learned in an embedded space, and 2) the unlearned task can be accomplished without trial and error by transferring skills through embedded spaces. As an experimental environment, we considered a navigation task for a mobile robot in a two-dimensional plane. The mobile robot receives the amount of movement in the x- and y-directions as the control input and then outputs the updated xy-coordinates after moving. This output value contains a different amount of error for each sensor setting. The error is sampled from the following normal distribution:

$$N\left(\mu, \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}\right) \tag{4}$$

In Eq. 4, a different μ value is used for each sensor setting. To accomplish a task, the mobile robot begins from the coordinate $(0, 0)$, and the goal coordinate is different for each task ID. Tables II and III present a summary of the tasks and sensor settings. The parameters of the new task and sensor settings for additional learning are shown in Tables II and III, respectively.

B. Results and Discussions

Fig.2 shows the trajectory of the mobile robot after the pre-training. As shown, the policy to attain the goal was

TABLE II
TASK SETTINGS

task ID	goal coordinate (x, y)	pre-training	additional training
0	(3.5, 3.5)	Used	Used
1	(-3.5, 3.5)	Used	Unused
2	(-3.5, -3.5))	Used	Unused
3	(3.5, -3.5))	Used	Unused

TABLE III
SENSOR SETTINGS

sensory ID	error mean (x, y)	pre-training	additional training
0	(0, 0)	Used	Unused
1	(1, 0)	Used	Unused
2	(0, 1)	Used	Unused
3	(1, 1)	Used	Unused
4	(0.75, 0.75)	Unused	Used

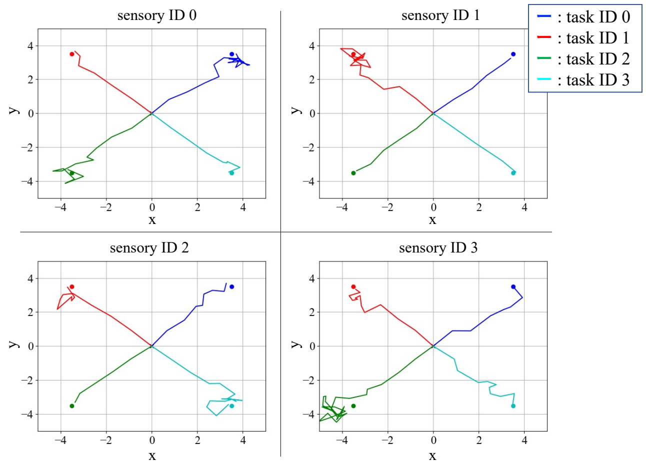

Fig. 2. Trajectories of the robot in 2D plane after pre-training with four tasks and four sensor settings. Lines shows a trajectory and points shows a goal coordinate. Initial position of the robot is $(0, 0)$. Task and sensor settings are shown in Tables II and III, respectively.

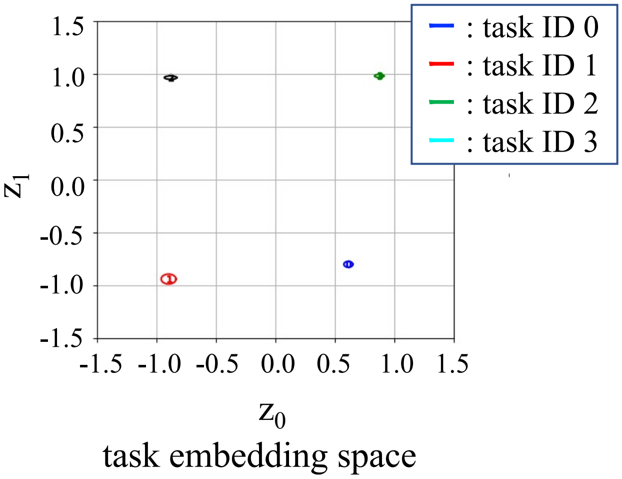

Fig. 3. Distribution of learned latent variables in 2D embedding space for task. Embedded net $p_t(z|t)$ and $p_e(z|e)$ represents the normal distributions, numbers in figure represent mean, and the ovals represent the range of standard deviations.

successfully learned for each task and sensor setting. Figs.3 and 4 show that each task and each sensor setting can be embedded in different regions of the embedding space. For example, the goal coordinates of tasks ID 0 and 2 are diagonal to the latent variable in the embedding space.This is because the skills are embedded in a continuous space, and the similarity of skills is represented as distance in the space. Therefore, skills with higher similarity are embedded in the neighborhood, whereas those with lower similarities are embedded at a distant location. This continuity is beneficial for solving tasks with unlearned sensor settings by interpolating from the learned sensor settings based on similarity.

Next, Figs. 5 and 6 show the results of training based on a new sensor setting. As expected from the characteristics of the embedding space described above, the latent variables in the vicinity of sensor ID = 3, which is the most similar sensor setting to the embedding space shown in Fig. 6, were learned to correspond to the new sensor setting. Using this latent variable for the sensor setting, not only task ID 0, which has been learned, but also tasks that were not learned (ID: 1, 2, and 3) can be solved (Fig.5). In the proposed method, the embedding networks of the tasks and sensor settings were learned using separate modules such that each latent variable can be changed independently. In addition, we performed training using all combinations of the four tasks and sensor settings during pre-training, while considering that the network acquired the embedding space without relying on any particular combination. Because the policy is a conditional probability of a latent variable, learning proceeds and affects only the task difference for the latent variable of the task and the sensor setting difference for the latent variable of the sensor setting. Hence, the network can accommodate unlearned combinations.

V. Conclusions

The challenge of transferring learning between different robots is defined as the challenge in incorporating the differences in the state action space and knowledge transfer. In this study, we extended Hausman's approach to embed the characteristics of the state action space separately from the task. Furthermore, we demonstrated that the proposed method can be used to solve unlearned tasks with zero shots in a new sensor setting using the learned embedding space. One future research direction is to improve the learning method of the embedding space. Because learning the embedded space requires the sampling of different tasks and robot settings in parallel, the actual machine is difficult to learn, and a simulator is required. One possible direction of development is to store data for each environment with a single task and robot setting at the time of learning, and then use these data to learn the embedded space in an off-policy manner. Hence, if we can construct an embedding space that considers various

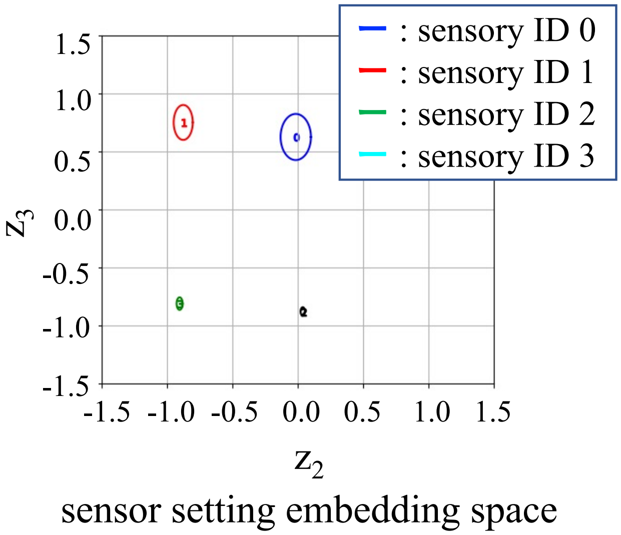

Fig. 4. Distribution of learned latent variables in 2D embedding space for sensor setting.

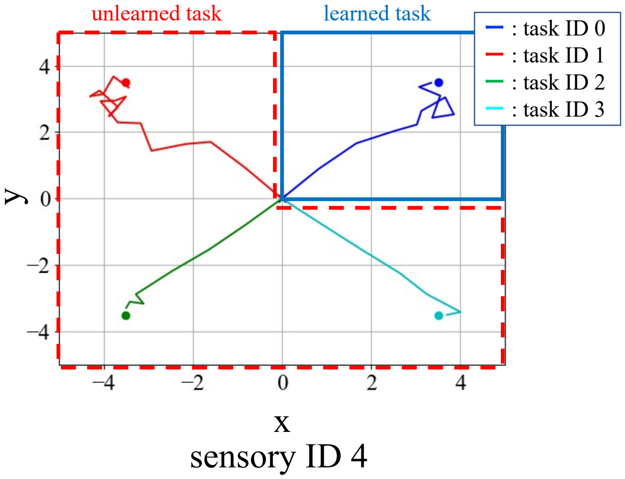

Fig. 5. Trajectories of the robot in sensory ID 4. Task ID 0 was used during additional training. Task IDs 1, 2, and 3 were used only evaluation.

environments and tasks from the data of each environment, then we can use it as a base model for new learning.

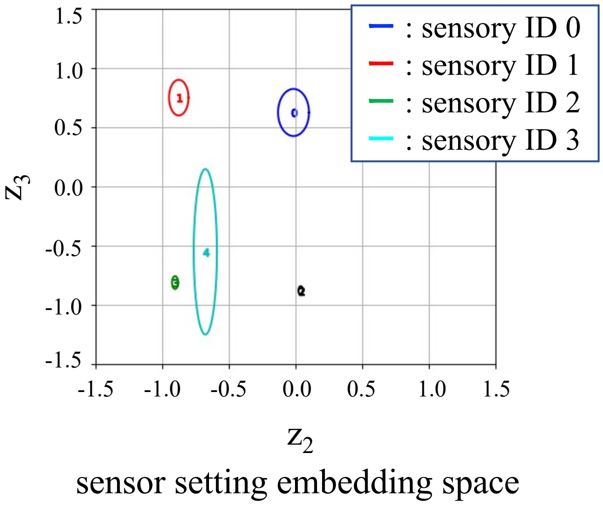

Fig. 6. Embedding space for sensor setting. Only task ID 0 was learned in this sensor setting.

REFERENCES

[1] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep Reinforcement Learning: A Brief Survey. IEEE Signal Processing Magazine. Volume: 34, Issue: 6, Nov. 2017.

[2] Hai Nguyen, and Hung Manh La. Review of Deep Reinforcement Learning for Robot Manipulation. IEEE International Conference on Robotic Computing (IRC), 2019.

[3] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. arXiv preprint arXiv:1801.01290, 2018.

[4] Dmitry Kalashnikov,et al. QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation. Conference on Robot Learning (CoRL), 2018.

[5] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. IEEE International Conference on Robotics and Automation (ICRA), 2018. 3803–3810.

[6] Kanishka Rao, Chris Harris, Alex Irpan, Sergey Levine, Julian Ibarz, and Mohi Khansari. Rl-cyclegan: Reinforcement learning aware simulation-to-real. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[7] Pin-Chu Yang, Kazuma Sasaki, Kanata Suzuki, Kei Kase, Shigeki Sugano, and Tetsuya Ogata. Repeatable Folding Task by Humanoid Robot Worker Using Deep Learning. IEEE Robot. Autom. Lett., vol. 2, no. 2, pp. 397–403, Apr. 2017.

[8] Karol Hausman, Jost Tobias Springenberg, Ziyu Wang, Nicolas Heess, and Martin Riedmiller. Learning an Embedding Space for Transferable Robot Skills. International Conference on Learning Representations (ICLR), 2018.

[9] Matthew E. Taylor, and Peter Stone. Transfer Learning for Reinforcement Learning Domains:A Survey. Journal of Machine Learning Research (JMLR), 2009, 10, 1633–1685.

[10] Alessandro Lazaric. Transfer in Reinforcement Learning: A Framework and a Survey, pp.143–173. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[11] Felipe Leno Da Silva, and Anna Helena Reali Costa. A Survey on Transfer Learning for Multiagent Reinforcement Learning Systems. Journal of Artificial Intelligence Research, 2019, 64, 645–703.

[12] Matthew Taylor, Peter Stone, and Yaxin Liu. Transfer Learning via Inter-Task Mappings for Temporal Difference Learning. Journal of Machine Learning Research, 2007, 8(1):2125–2167.

[13] Haitham Bou Ammar, Eric Eaton, Paul Ruvolo, and Matthew Taylor. Unsupervised Cross-Domain Transfer in Policy Gradient Reinforcement Learning via Manifold Alignment. AAAI Conference on Artificial Intelligence, 2015.

[14] Coline Devin, Abhishek Gupta, Trevor Darrell, Pieter Abbeel, and Sergey Levine. Learning Modular Neural Network Policies for Multi-Task and Multi-Robot Transfer. IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 2169–2176.