TRANS-AM: Transfer Learning by Aggregating Dynamics Models for Soft Robotic Assembly

Kazutoshi Tanaka[1], Ryo Yonetani[1], Masashi Hamaya[1], Robert Lee[1], Felix von Drigalski[1], and Yoshihisa Ijiri[1]

***Abstract*— Practical industrial assembly scenarios often require robotic agents to adapt their skills to unseen tasks quickly. While transfer reinforcement learning (RL) could enable such quick adaptation, much prior work has to collect many samples from source environments to learn target tasks in a model-free fashion, which still lacks sample efficiency on a practical level. In this work, we develop a novel transfer RL method named TRANSfer learning by Aggregating dynamics Models (TRANS-AM). TRANS-AM is based on model-based RL (MBRL) for its high-level sample efficiency, and only requires dynamics models to be collected from source environments. Specifically, it learns to aggregate source dynamics models adaptively in an MBRL loop to better fit the state-transition dynamics of target environments and execute optimal actions there. As a case study to show the effectiveness of this proposed approach, we address a challenging contact-rich peg-in-hole task with variable hole orientations using a soft robot. Our evaluations with both simulation and real-robot experiments demonstrate that TRANS-AM enables the soft robot to accomplish target tasks with fewer episodes compared when learning the tasks from scratch.**

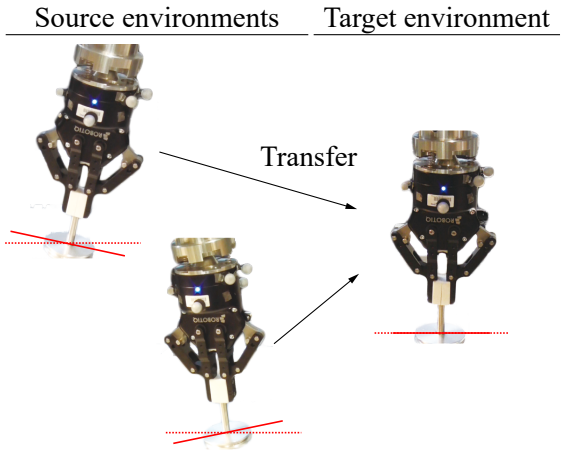

Fig. 1. **Motivating example**. We address a problem of transferring robotic peg-in-hole skills between different hole orientations, which is formulated as transfer RL between different state-transition dynamics.

I. Introduction

Industrial robots at factory lines perform a variety of contact-rich manipulation tasks, such as stacking a piece of items onto a base and inserting a peg into a hole. In practical scenarios, it is also critical for such robots to get adapted to a new environmental setup quickly, *e.g.*, when launching a new factory line where robots are installed in different ways or accommodating the robots in existing lines to a new workpiece with different physical characteristics. While reinforcement learning (RL) could allow robots to acquire manipulation skills automatically [1], quick adaptation of learned skills to new environmental setups is still challenging as such skills are often specific to the environmental configurations where the learning is performed [2], [3].

The scenarios above can be viewed as a problem of transfer RL that aims to quickly learn a new target task by leveraging information acquired from source environments [4]. Particularly in this work, we consider the problem setting where 1) *source and target environments are the same task but different in their state-transition dynamics, 2) exact dynamics of the environments are unknown, and 3) the communication with the source environments is prohibited when learning in the target environment*. As a motivating example, Figure 1 shows a series of robotic peg-in-hole tasks with variable hole orientations, which often appear in practical situations (*e.g.*, [5]). While the peg is in contact with the hole, even a small difference in hole orientations has a great influence on robot's state transitions (*i.e.*, the same action commands result in different poses or positions of the peg), which requires a unique strategy for peg insertion. Moreover, a full specification of state-transition dynamics parameters is not typically available due to unknown, inaccurate, or time-varying robot dynamics, such as link mass, joint dumping, friction, and inertia [2]. Finally, poor communication channels among distributed factory lines can prevent each robot from interacting with other robots to collect information on a large scale [3].

While much work has been done to enable inter-dynamics transfer [2], [4], [6]–[8], most of them require to collect thousands of samples from source environments to take into account their state-transition dynamics. Other relevant work adopted a meta-learning approach for quickly learning target tasks under unknown dynamics [9]–[11]. However, they typically assume robots to be trained on a diverse set of source tasks in advance. Either way, these approaches are not easily applicable to our problem setting as long as they

[1]Kazutoshi Tanaka, Ryo Yonetani, Masashi Hamaya, Robert Lee, Felix von Drigalski, and Yoshihisa Ijiri are with OMRON SINIC X Corporation, Hongo 5-24-5, Bunkyo-ku, Tokyo, Japan { kazutoshi.tanaka, ryo.yonetani, f.drigalski, masashi.hamaya, robert.lee, yoshihisa.ijiri }@sinicx.com

*This work was supported by JSPS KAKENHI Grant Numbers JP19K14936.

require frequent access to source environments.

On the other hand, recent studies have proposed to utilize policies acquired from source environments [3], [12]. These approaches do not require source samples and can work even when source state-transition dynamics are unknown. Nevertheless, they are based on model-free RL and necessitate many interactions at target environments. We argue that this is inefficient and undesirable, especially for contact-rich manipulation tasks at factories, as conducting many trials for sample collection not only leads to long delays until deployment but can also damage the robots and workpieces.

Based on the above background, we develop a new transfer RL method called *TRANSfer learning by Aggregating dynamics Models (TRANS-AM)*. To achieve high sample efficiency, our method adopts a model-based RL (MBRL) approach that allows agents to select optimal actions by learning a model of state-transition dynamics of a given environment. Our key idea is then to collect and utilize a collection of dynamics models acquired from source environments (*i.e.*, source dynamics models) for learning a state-transition dynamics of the target environment, rather than collecting source samples or policies that existing model-free transfer RL used. As illustrated in Figure 2, each source dynamics model approximates unknown state-transition dynamics as a black-box function, and is used in TRANS-AM as follows:

- TRANS-AM learns to adaptively aggregate outputs from multiple source dynamics models in an MBRL loop. Doing so allows it to flexibly interpolate or extrapolate source models to approximate target state-transition dynamics without knowing exact dynamics parameters of each environment.
- Moreover, TRANS-AM learns an *auxiliary model* jointly with adaptive aggregation to enable the prediction of residuals around aggregated outputs. As with the auxiliary policies used in model-free RL [3], [12], our auxiliary model can ensure the expressiveness of the resulting target dynamics model — a necessary trait particularly when there is a substantial gap between the source and target state-transition dynamics.

As a practical scenario, we address a robotic assembly, more specifically a challenging peg-in-hole task with variable hole orientations using a robotic arm with a soft wrist connecting the end of the arm to the gripper with springs [13]. Although recent work has revealed the effectiveness of soft robots for contact-rich manipulation, learning their complex dynamics in MBRL is still challenging and has been done independently for every single task [14]. Through extensive simulation and real-robot experiments, we confirmed that the proposed TRANS-AM enabled soft robots to accomplish a target task in a shorter time by utilizing dynamics models acquired in source environments, compared to when conducting MBRL in the target environment from scratch.

II. Related Work

A. Learning for robotic assembly

Our work is aimed at accomplishing robotic manipulation tasks for practical industrial assembly scenarios, where

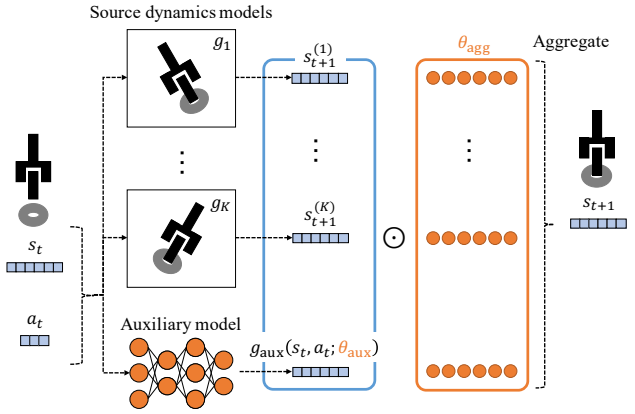

Fig. 2. **TRANS-AM framework**. Our model-based transfer RL approach adaptively aggregates dynamics models acquired from source environments to approximate target state-transition dynamics. Trainable parameters (θ_{aux} and θ_{agg}) are highlighted in orange.

the efficiency of doing so in unseen environmental setups is a critical factor. Much work has proposed RL-based frameworks to learn optimal assembly strategies [15]. In particular, we are interested in model-based approaches (*i.e.*, MBRL), as they are sample efficient and shown effective for a variety of assembly tasks [1], [16]–[19]. Recent work has leveraged deep neural networks to deal with complex dynamics [20]–[22]. However, they tend to overfit on small samples compared to when using simpler models [21], such as ones based on Gaussian process [23], [24].

A promising direction to make learning sample efficient is to leverage knowledge acquired from other (source) tasks relevant to the target one. This motivation is often found in the literature of transfer RL. As summarized in [4], transfer RL methods can be categorized based on how source and target environments are different (*e.g.*, reward functions, state-transition dynamics, or state/action spaces) and what are transferred (*e.g.*, policies, Q functions, or dynamics models). Among much work done so far, attempts on transfer between different state-transition dynamics are relatively limited [2], [3], [6]–[8], [12]. More critically, they are all model-free and not applicable to MBRL used in our proposed approach.

Another relevant domain is meta learning, which has recently been studied actively to enable robotic agents to quickly adapt unknown tasks [9], [10], [25]–[27]. As discussed in the previous section, meta-RL approaches are not always useful for practical industrial assembly scenarios where robots in factories are not necessarily kept accessible to a variety of different environments. On the other hand, our proposed approach can work by receiving dynamics models from several other relevant environments only once.

B. Soft robotics

As a case study to show the effectiveness of our approach, we are interested in using *soft robots* [28] for industrial assembly. The physical softness allows a robot to compensate for positional error when being controlled, by deforming their body while contacting objects. Such soft robots have

been realized in a variety of forms, with compliant grippers [29], [30], wrists [31], [32], and arms [33], [34].

Nevertheless, the complexity of soft robotic bodies makes it hard to design controllers manually [35] and required data-driven approaches [36]–[38]. Most recently, MBRL has been used to learn soft-robotic control tasks [14], [39]. Our study extends this line of research and proposes a transfer RL method that works effectively for sample efficiency even when state-transition dynamics are complex and unknown due to the softness of robotic arms.

III. TRANS-AM: Transfer Learning by Aggregating Dynamics Models

A. Preliminaries

1) Markov decision process: We formulate our problem as a standard RL problem [40]. Specifically, contact-rich robotic manipulation tasks are modeled using the Markov decision process (MDP) with tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, T, R \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ respectively denote the state and action spaces, $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is a function representing state-transition dynamics, and $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function. At each time step t, the agent in state $\boldsymbol{s}_t \in \mathcal{S}$ executes an action $\boldsymbol{a}_t \in \mathcal{A}$ to receive reward $R(\boldsymbol{s}_t, \boldsymbol{a}_t)$. The state is then transitioned to the next one, $\boldsymbol{s}_{t+1} = T(\boldsymbol{s}_t, \boldsymbol{a}_t)$. For contact-rich manipulation tasks, each state can involve positions and orientations of the end-effector or those of parts being grasped, whereas actions can be implemented as continuous commands to control the velocity of the arm tip. The reward function R quantifies how effective the current state is at accomplishing a given task. More specific configurations for our experiments are presented in Sections IV and V.

2) Model-based reinforcement learning: The goal of RL is to obtain the optimal controller that can execute a sequence of actions to maximize the return (*i.e.*, total reward accumulated over time). In this work, we adopt MBRL for its excellent sample efficiency. Specifically, we learn a model that approximates the state-transition dynamics T using samples collected from real dynamics, which we denote by $g(\boldsymbol{s}_t, \boldsymbol{a}_t; \theta)$ parameterized by θ. The learned model is then used to generate a distribution of state trajectories by applying sequences of actions, and select optimal actions based on the return computed for each generated trajectory.

3) Transfer learning between different dynamics: In this work, we focus on a transfer RL setting where the source and target environments have unknown different state-transition dynamics. Specifically, consider K relevant source environments that are each characterized by a distinct dynamics model $g_k : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ $(k = 1, \ldots, K)$ mapping a given state-action pair to different states. Crucially, similar to the setting of [3], [12], these models may or may *not* be parameterized and trainable (*e.g.*, learned neural networks or hand-engineered simulators with heuristic rules that approximate a real system) and *regarded as a fixed black-box function with no trainable parameters*. Then, for a new target environment with unknown state-transition dynamics, our goal in this work is to quickly learn a model $g^{\mathrm{target}}(\boldsymbol{s}_t, \boldsymbol{a}_t; \theta)$ by leveraging a collection of source dynamics models $G = \{g_k\}_{k=1}^K$ to further improve the sample efficiency of MBRL.

B. Adaptive aggregation of multiple dynamics models

As illustrated in Figure 2, the proposed TRANS-AM learns a model of target state-transition dynamics by: a) adaptively aggregating the outputs from source dynamics models while b) jointly learning an auxiliary model *to predict the residual around the aggregation results*. Formally, for a set of source dynamics models $G = \{g_k\}_{k=1}^K$, the output $\boldsymbol{s}_{t+1}^{(k)} = g_k(\boldsymbol{s}_t, \boldsymbol{a}_t)$ from a certain state-action pair is concatenated as follows[1]:

$$S_{t+1} = \left[(\boldsymbol{s}_{t+1}^{(1)})^\top, \ldots, (\boldsymbol{s}_{t+1}^{(K)})^\top \right] \in \mathbb{R}^{K \times D}, \tag{1}$$

where D is the dimension of state space $\mathcal{S}$. We denote the auxiliary model parameterized by θ_{aux} as $g_{\mathrm{aux}}(\theta_{\mathrm{aux}}) : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$. Any function that is fully differentiable and accepts the same state and action spaces can be used to represent g_{aux}. Then, for a new target environment, the source models and the auxiliary model are aggregated adaptively to represent a target dynamics model g^{target} as follows:

$$\begin{aligned} \boldsymbol{s}_{t+1} &\approx g^{\mathrm{target}}(\boldsymbol{s}_t, \boldsymbol{a}_t; \theta = (\theta_{\mathrm{agg}}, \theta_{\mathrm{aux}})) && (2) \\ &= \mathbb{1}^\top \left(\theta_{\mathrm{agg}} \odot \left[S_{t+1}, g_{\mathrm{aux}}(\boldsymbol{s}_t, \boldsymbol{a}_t; \theta_{\mathrm{aux}})^\top \right] \right), && (3) \end{aligned}$$

where $\theta_{\mathrm{agg}} \in \mathcal{R}^{(K+1) \times D}$ is a matrix of trainable aggregation weight parameters, $\odot$ denotes element-wise multiplication, and $\mathbb{1}$ is the all-ones column vector of size $K + 1$. Note that the matrix θ_{agg} is not particularly regularized. This enables the flexible aggregation of outputs from each model, which was shown in previous work on model-free transfer RL to be advantageous when accurate specifications of dynamics are not available for both of the source and target environments [3].

C. Learning algorithm

As indicated in Eq. (2), TRANS-AM characterizes a dynamics model of target environments using a set of parameters $\theta = (\theta_{\mathrm{agg}}, \theta_{\mathrm{aux}})$. These parameters can be trained in arbitrary MBRL algorithms that directly update the model via back-propagation. Inspired by [21], we adopt the model-predictive control (MPC) through the use of the cross-entropy method (CEM) [41] with trajectory sampling.

IV. Simulation Experiment

Keeping in mind that our original focus is a contact-rich manipulation task for industrial robots, we evaluate the effectiveness of the proposed TRANS-AM method on a challenging soft robot assembly task. While physical softness of robotic arms makes contact-rich manipulation easy and safe, it also comes with difficulty in learning dynamics for MBRL. To this end, we first conduct a systematic evaluation using an extensive simulation experiment on soft-robotic peg-in-hole tasks with variable hole orientations.

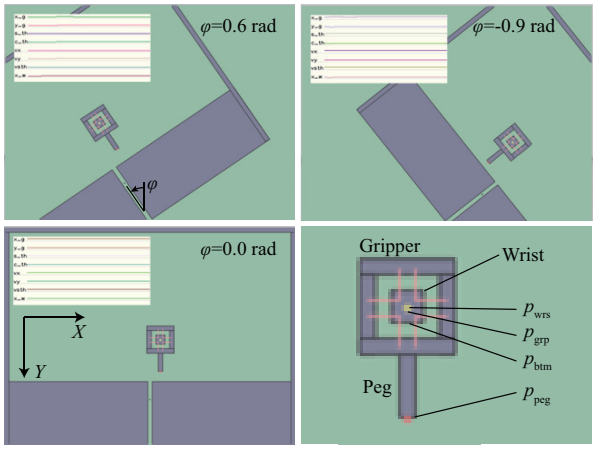

Fig. 3. **Soft-robotic peg-in-hole simulation**. Each pane shows a different environmental setting involving variations in hole orientations.

A. Simulation setup

We implemented a 2D simulation of soft-robotic peg-in-hole environments using the Box2D physics engine [42]. As shown in Figure 3, the simulated robotic arm consists of a wrist and a gripper with a peg attached, which are connected with eight springs to realize physical softness. The state of the arm is defined as $\boldsymbol{s}_t = [\boldsymbol{p}_{\mathrm{wrs}}^\top, \boldsymbol{p}_{\mathrm{btm}}^\top, \boldsymbol{p}_{\mathrm{grp}}^\top, \boldsymbol{p}_{\mathrm{peg}}^\top]^\top$, where $\boldsymbol{p}_{\mathrm{wrs}}$ and $\boldsymbol{p}_{\mathrm{grp}}$ are the centroid of the wrist and the gripper, respectively, and $\boldsymbol{p}_{\mathrm{btm}}$ and $\boldsymbol{p}_{\mathrm{peg}}$ are the bottom-center locations of the wrist and the peg, respectively. The action is described by $\boldsymbol{a}_t = [v_X, v_Y, \omega]^\top$ that are respectively the horizontal, vertical, and angular velocities of the wrist. We give a reward function R based on the distance between the peg and the hole on a surface, where the task was regarded as successful when the distance fell within a specific threshold. With this simulation environment, we created a collection of seven environment instances with distinct hole orientations $\phi \in \{-0.9, -0.6, -0.3, 0.0, 0.3, 0.6, 0.9\}$ (rad) and random initial arm positions. In this way, each environment instance was characterized by distinct state-transition dynamics because *the same action commands result in different next states while the peg is in contact with the hole.*

B. Implementation details

1) Source dynamics models: For each environment instance, we trained an ensemble of standard three-layer multilayer perception (MLP) models with batch normalization and ReLU activation (except the last layer activated linearly), and used it as a source dynamics model[2] g_k. The model was trained using an Adam optimizer with a learning rate of 0.001. As introduced in Section III, we adopted CEM-based MPC to determine optimal next actions via MBRL. The prediction horizon of MPC was set to three steps and the number of generated state trajectories was set to 300, from which we chose the 100 best action sequences to update the mean and standard deviation of the CEM distribution. The task horizon for CEM updates was set to 10.

2) Transfer learning setup: From the dynamics models each obtained under different hole orientation ϕ, we selected the following eight model combinations as the sources for transfer learning: $\phi = -0.6, \phi = 0.0, \phi = 0.6$ (*i.e.*, single source cases, $K = 1$), and $\phi \in \{-0.6, 0.6\}, \phi \in \{0.3, 0.3\}, \phi \in \{-0.6, -0.3\}$ ($K = 2$), $\phi = \{-0.6, 0.0, -0.6\}$ ($K = 3$), and $\phi = \{-0.6, -0.3, 0.3, 0.6\}$ ($K = 4$). These source models were regarded as a black-box function throughout learning target tasks. The auxiliary model $g_{\mathrm{aux}}(\theta_{\mathrm{aux}})$ had the identical architecture as that of the source models. We selected the target environment instances so as not to have the same hole orientations as those of source instances (*e.g.*, six cases for $K = 1$ and five for $K = 2$). The training was done for 20 episodes and repeated ten times with different random seeds. Note that we specify state-transition dynamics of source and target environments concretely in this way for making our evaluation systematic and reproducible; the specific values of ϕ were not when learning target tasks as they were inaccessible in practice.

3) Evaluation protocol: To evaluate the effectiveness of transfer via TRANS-AM, we calculated the mean and standard deviation for the number of the first episodes that a target task resulted in success. We compared TRANS-AM with a baseline method that learned the target task from scratch, where *earlier first successes mean effective transfer*. We also measured average success rates up to $e = 5, 10, 15, 20$-th episodes to show more detailed statistics about obtained results.

C. Results

1) Quantitative comparison: As summarized in Table I, we confirmed that TRANS-AM improved the average first success episode compared to the baseline that learned the task from scratch, especially when multiple source models were used for $K \geq 2$ cases. Moreover, the standard deviation of first success episodes was smaller using TRANS-AM, indicating its robustness against the choices of source and target environments. Figure 4 shows the mean and standard error of episodic returns. Through the use of source dynamics models, TRANS-AM received much higher returns from the first episode, which led to earlier success. Note that we did not confirm the monotonic improvement for the first-success episode criterion with respect to the number of source models K, presumably because the number of aggregation parameters also increased.

2) Ablation study: To investigate the contributions of each technical component of TRANS-AM in more detail, we evaluated the following two degraded methods for $K = 2$: a) the **w/o aux** approach that just aggregated two source models without applying an auxiliary model g_{aux}; and b) the **w/o adaptive agg.** approach that simply averaged source model outputs instead of learning their weights. The results in Table II clearly show that both the adaptive aggregation and auxiliary model approaches contribute significantly to

[1]Here, we denote row-wise stacking by $[a, b, c, \dots]$.

[2]Note that any configurations of dynamics models, including non-trainable simulators and non-differential models, can be used as long as they performed reasonably well in carrying out given tasks.

TABLE I

RESULTS OF SIMULATION EXPERIMENTS.

	First success	$e = 5$	10	15	20
Baseline	7.98 ± 0.54	0.24	0.54	0.79	0.87
TRANS-AM ($K = 1$)	6.67 ± 0.31	0.31	0.66	0.80	0.86
($K = 2$)	5.98 ± 0.32	0.41	0.69	0.84	0.89
($K = 3$)	**5.76 ± 0.65**	0.41	0.70	0.77	0.85
($K = 4$)	6.27 ± 0.72	**0.48**	**0.71**	**0.87**	**0.96**

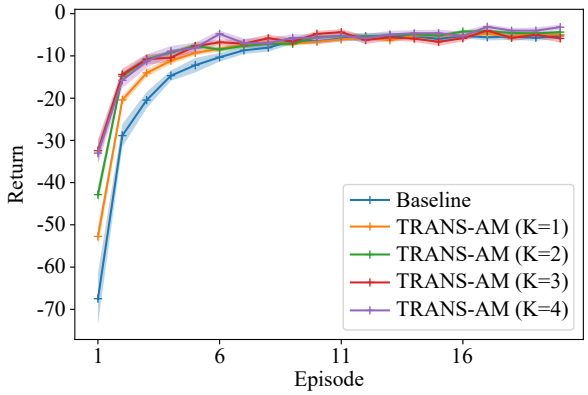

Fig. 4. **Averaged episodic returns for the simulation experiment.** Episodic returns are computed and averaged over multiple sessions.

the success of TRANS-AM. Specifically, even though both degraded methods outperformed the baseline in terms of achieving the first success episode, their overall success rates, particularly that of **w/o aux**, were quite limited.

3) Choice of source dynamics models: Figure 5 shows the learned weights for different source/target configurations. Overall, source models with dynamics (*i.e.*, hole orientations) similar to those of the target instances receive higher weights, while otherwise increased weights are given to the auxiliary model to alleviate the mismatch between source and target dynamics. For example, the bottom row of the figure shows that the weight assigned to the auxiliary model was much higher than those for source models, when the target hole orientation was $\phi = 0.9$ while source orientations were $\phi = -0.3$ and $\phi = 0.3$.

V. REAL-ROBOT EXPERIMENT

To further investigate the practical impact of TRANS-AM, we conduct a soft-robotic peg-in-hole task with variable hole orientations in the real-world environment.

A. Robotic system setup

We used the UR5 robot arm (Universal Robots; see Figure 6) as a base robotic system. We equipped the robot arm with a compliant wrist described in [13], a gripper (2F-85, ROBOTIQ), and force-torque sensor (FT300, ROBOTIQ). The pose of the gripper was measured using six motion capture cameras (FLEX13, OptiTrack). A 10-mm diameter peg made of stainless steel was fixed to the gripper and inserted to a hole with the depth of 10 mm on the surface. The tolerance between the peg and the hole was H7/h7.

TABLE II

ABLATION STUDY.

	First success	$e = 5$	10	15	20
TRANS-AM ($K = 2$)	5.98 ± 0.32	**0.41**	**0.69**	**0.84**	**0.89**
w/o aux	**5.34 ± 0.45**	0.36	0.48	0.59	0.64
w/o adaptive agg.	7.19 ± 0.35	0.33	0.61	0.78	0.87

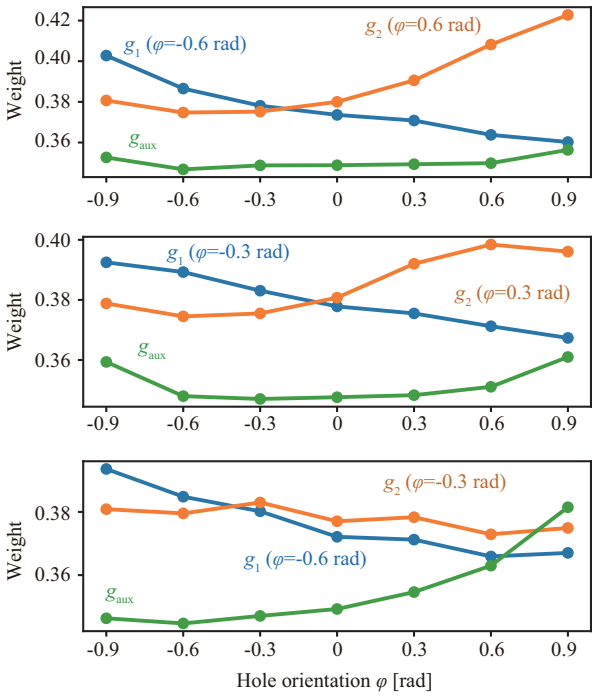

Fig. 5. **Learned aggregation weights** ($K = 2$). Horizontal axes indicate hole orientations of target environment instances. Blue and orange plots correspond to source dynamics models to be aggregated; green plots shows the auxiliary dynamics model.

With this system, we designed three different peg-in-hole tasks with different hole orientations: $\phi \in \{-10°, 0°, 10°\}$ (see also Figure 2). As with the previous simulation experiment, these tasks are characterized by different state-transition dynamics due to ϕ.

B. Environment setup

To conduct MBRL, actions of the robot were given by the 3D velocity of the arm tip in the Cartesian coordinate, *i.e.*, $\boldsymbol{a} = [v_X, v_Y, v_Z]^\top$, while keeping the orientation of the wrist constant. Upon receiving action commands at 5 Hz in this form, we computed the angular velocity of robot joints using MoveIt [43] to execute the actions. The state was defined as

$$\boldsymbol{s} = [\boldsymbol{p}_{\mathrm{grp}}^\top, \boldsymbol{\phi}_{\mathrm{grp}}^\top, \boldsymbol{p}_{\mathrm{arm}}^\top, \boldsymbol{f}^\top]^\top, \tag{4}$$

where $\boldsymbol{p}_{\mathrm{grp}}$ and $\boldsymbol{p}_{\mathrm{arm}}$ are relative positions of the gripper and the arm tip with respect to the hole location, $\boldsymbol{f} = [f_X, f_Y, f_Z]^\top$ is a force measurement obtained by the force-torque sensor. These quantities are normalized by a predefined constant to approximately match their value range. $\boldsymbol{\phi}_{\mathrm{grp}}$ is a vector representing gripper orientations as follows:

$$\boldsymbol{\phi}_{\mathrm{grp}} = [\sin\alpha, \cos\alpha, \sin\beta, \cos\beta, \sin\gamma, \cos\gamma]^\top, \tag{5}$$

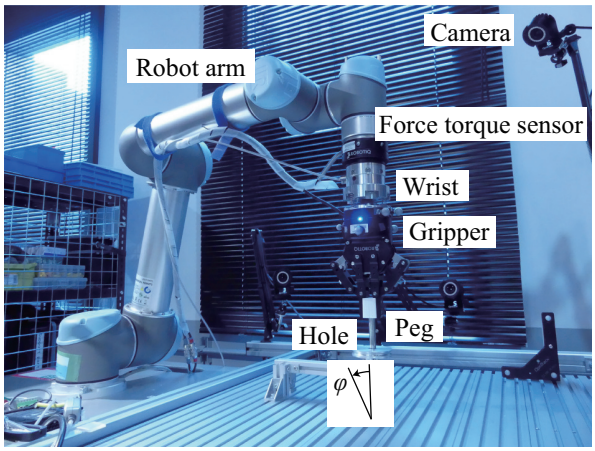

Fig. 6. **UR5 robot setup**. The robot is equipped with a compliant wrist [13] for physically soft control.

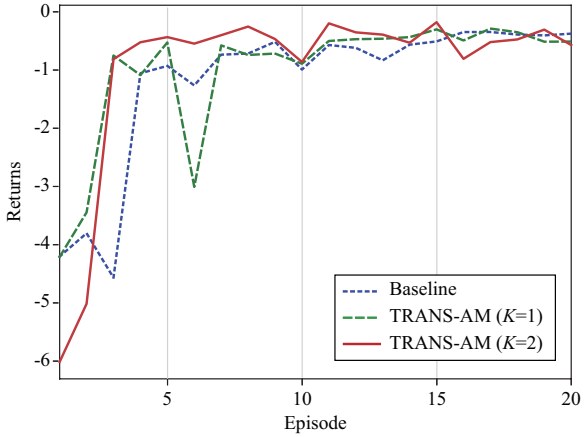

Fig. 7. **Averaged episodic returns for the real-robot experiment.** Episodic returns are computed and averaged over multiple sessions.

TABLE III
RESULTS OF REAL-ROBOT EXPERIMENTS.

	First Success	$e=5$	10	15	20
Baseline	9.25 ± 5.49	0.2	0.4	0.6	0.8
TRANS-AM $(K=1)$	8.20 ± 5.34	**0.4**	**0.6**	0.8	**1.0**
$(K=2)$	$\mathbf{7.20 \pm 3.37}$	**0.4**	**0.6**	**1.0**	**1.0**

where α, β, and γ are the roll, pitch, and yaw angle of the gripper, respectively. Finally, the reward was given by the gripper position $\boldsymbol{p}_{\mathrm{grp}}$ and the force measurement $\boldsymbol{f}$:

$$r = -||\boldsymbol{p}_{\mathrm{grp}}||^2 - ||\boldsymbol{f}||^2, \tag{6}$$

which reaches zero when the peg comes to rest in the hole.

C. Learning and evaluation setup

Following the previous simulation experiment, we trained ensembles of MLPs as the dynamics models of source environment instances and used them for MBRL with CEM-based MPC. Learning rate of the Adam optimizer was set to 0.01, while the other hyper-parameters were the same as those in the previous experiment. We conducted TRANS-AM for 1) transferring from $\phi = -10^\circ$ to $\phi = 0^\circ$ (*i.e.*, $K = 1$) and from $\phi \in \{-10^\circ, 10^\circ\}$ to $\phi = 0^\circ$ (*i.e.*, $K = 2$).

We continued learning for 20 episodes, where actions were selected randomly for exploration in the first two episodes. Each episode consisted of no longer than 100 timesteps and was completed once the peg was inserted into the hole successfully. Moreover, we also terminated episodes if actions with too strong force were about to be executed or if the grip moved too far away from the hole for safety reasons. We conducted the above learning sessions five times with different random seeds and computed an average of the first successful episodes as well as mean success rates up to $e = 5, 10, 15, 20$-th episodes.

D. Results

Table III shows quantitative results. Using source dynamics models in TRANS-AM improved the average and standard deviation of first success episodes compared to the baseline method that learned a target task from scratch (from 9.25 ± 5.49 to 7.20 ± 3.37 for $K = 2$), demonstrating the effectiveness of the proposed approach. Moreover, TRANS-AM allowed the robot to achieve 100% success rate at the 15-th episode in $K = 2$ and at the 20-th episode in $K = 1$. Figure 7 reports average episodic returns, showing that TRANS-AM got higher returns in earlier episodes. Finally, Figure 8 shows some snapshots of a successful session.

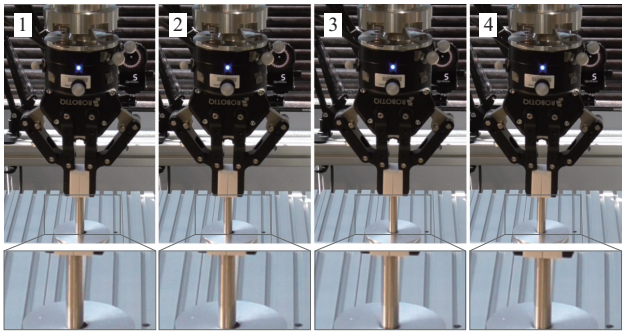

Fig. 8. **Snapshots of successful peg-in-hole actions**. These snapshots approximately correspond to the moments where the peg was 1) contacted, 2) slided, 3) aligned, and 4) inserted.

VI. CONCLUSION

We presented TRANS-AM, a new approach to model-based transfer reinforcement learning for different state-transition dynamics. The key idea is learning to adaptively aggregate dynamics models obtained in multiple source environments to approximate the target state-transition dynamics. We confirmed the effectiveness of the proposed method on a challenging soft-robotic peg-in-hole task in both simulation and real-robot environments.

Effective transfer between different state-transition dynamics would become critical not only when adapting acquired skills to unseen environmental setups but also when leveraging simulations in sim2real [8], [44] or when learning tasks in non-stationary environments via continual learning [45], [46]. Future work will seek to extend model-based transfer RL via TRANS-AM to work on such challenging scenarios.

REFERENCES

[1] S. Levine and V. Koltun, "Guided policy search," in *International Conference on Machine Learning*, 2013, pp. 1–9.

[2] T. Chen, A. Murali, and A. Gupta, "Hardware conditioned policies for multi-robot transfer learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 9333–9344.

[3] M. Barekatain, R. Yonetani, and M. Hamaya, "Multipolar: Multi-source policy aggregation for transfer reinforcement learning between diverse environmental dynamics," in *International Joint Conference on Artificial Intelligence*, 2020.

[4] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *Journal of Machine Learning Research*, vol. 10, no. 7, 2009.

[5] F. Von Drigalski, C. Schlette, M. Rudorfer, N. Correll, J. C. Triyonoputro, W. Wan, T. Tsuji, and T. Watanabe, "Robots assembling machines: learning from the world robot summit 2018 assembly challenge," *Advanced Robotics*, vol. 34, no. 7-8, pp. 408–421, 2020.

[6] J. Song, Y. Gao, H. Wang, and B. An, "Measuring the distance between finite markov decision processes," in *International Conference on Autonomous Agents and Multiagent Systems*, 2016.

[7] H. Wang, S. Dong, and L. Shao, "Measuring structural similarities in finite mdps," in *International Joint Conference on Artificial Intelligence*, 2019.

[8] W. Yu, C. K. Liu, and G. Turk, "Policy transfer with strategy optimization," in *International Conference on Learning Representations*, 2019.

[9] J. Vanschoren, "Meta-learning: A survey," *arXiv preprint arXiv:1810.03548*, 2018.

[10] A. Nagabandi, I. Clavera, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn, "Learning to adapt in dynamic, real-world environments through meta-reinforcement learning," in *International Conference on Learning Representations*, 2019.

[11] A. Rajeswaran, S. Ghotra, B. Ravindran, and S. Levine, "Epopt: Learning robust neural network policies using model ensembles," *arXiv preprint arXiv:1610.01283*, 2016.

[12] J. Rajendran, A. S. Lakshminarayanan, M. M. Khapra, P. Prasanna, and B. Ravindran, "Attend, adapt and transfer: Attentive deep architecture for adaptive transfer from multiple sources in the same domain," in *International Conference on Learning Representations*, 2017.

[13] F. Von Drigalski, K. Tanaka, M. Hamaya, R. Lee, C. Nakashima, Y. Shibata, and Y. Ijiri, "A compact, cable-driven, activatable soft wrist with six degrees of freedom for assembly tasks," in *International Conference on Intelligent Robots and Systems*, 2020.

[14] M. Hamaya, R. Lee, K. Tanaka, F. Von Drigalski, C. Nakashima, Y. Shibata, and Y. Ijiri, "Learning robotic assembly tasks with lower dimensional systems by leveraging physical softness and environmental constraints," in *International Conference on Robotics and Automation*, 2020, pp. 7747–7753.

[15] J. Xu, Z. Hou, Z. Liu, and H. Qiao, "Compare contact model-based control and contact model-free learning: A survey of robotic peg-in-hole assembly strategies," *arXiv preprint arXiv:1904.05240*, 2019.

[16] A. S. Polydoros and L. Nalpantidis, "Survey of model-based reinforcement learning: Applications on robotics," *Journal of Intelligent & Robotic Systems*, vol. 86, no. 2, pp. 153–173, 2017.

[17] G. Thomas, M. Chien, A. Tamar, J. A. Ojea, and P. Abbeel, "Learning robotic assembly from cad," in *International Conference on Robotics and Automation*, 2018, pp. 1–9.

[18] A. Wang, T. Kurutach, K. Liu, P. Abbeel, and A. Tamar, "Learning robotic manipulation through visual planning and acting," *arXiv preprint arXiv:1905.04411*, 2019.

[19] J. Luo, E. Solowjow, C. Wen, J. A. Ojea, A. M. Agogino, A. Tamar, and P. Abbeel, "Reinforcement learning on variable impedance controller for high-precision robotic assembly," in *International Conference on Robotics and Automation*, 2019, pp. 3080–3087.

[20] Y. Gal, R. McAllister, and C. E. Rasmussen, "Improving pilco with bayesian neural network dynamics models," in *Data-Efficient Machine Learning Workshop*, vol. 4, 2016, p. 34.

[21] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," in *Advances in Neural Information Processing Systems*, 2018, pp. 4754–4765.

[22] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *International Conference on Robotics and Automation*, 2018, pp. 7559–7566.

[23] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *International Conference on Machine Learning*, 2011, pp. 465–472.

[24] R. Grande, T. Walsh, and J. How, "Sample efficient reinforcement learning with gaussian processes," in *International Conference on Machine Learning*, 2014, pp. 1332–1340.

[25] S. Sæmundsson, K. Hofmann, and M. Deisenroth, "Meta reinforcement learning with latent variable gaussian processes," in *Conference on Uncertainty in Artificial Intelligence*, vol. 34, 2018, pp. 642–652.

[26] I. Clavera, J. Rothfuss, J. Schulman, Y. Fujita, T. Asfour, and P. Abbeel, "Model-based reinforcement learning via meta-policy optimization," in *Conference on Robot Learning*, 2018.

[27] G. Schoettler, A. Nair, J. A. Ojea, S. Levine, and E. Solowjow, "Meta-reinforcement learning for robotic industrial insertion tasks," in *International Conference on Intelligent Robots and Systems*, 2020, pp. 9728–9735.

[28] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.

[29] F. Ilievski, A. D. Mazzeo, R. F. Shepherd, X. Chen, and G. M. Whitesides, "Soft robotics for chemists," *Angewandte Chemie International Edition*, vol. 50, no. 8, pp. 1890–1895, 2011.

[30] K. Suzumori, S. Iikura, and H. Tanaka, "Applying a flexible microactuator to robotic mechanisms," *Control Systems Magazine*, vol. 12, no. 1, pp. 21–27, 1992.

[31] Y. Xu and R. P. Paul, "A robot compliant wrist system for automated assembly," in *International Conference on Robotics and Automation*, 1990, pp. 1750–1755.

[32] T. Nishimura, Y. Suzuki, T. Tsuji, and T. Watanabe, "Peg-in-hole under state uncertainties via a passive wrist joint with push-activate-rotation function," in *International Conference on Humanoid Robots*, 2017, pp. 67–74.

[33] C. Laschi, M. Cianchetti, B. Mazzolai, L. Margheri, M. Follador, and P. Dario, "Soft robot arm inspired by the octopus," *Advanced Robotics*, vol. 26, no. 7, pp. 709–727, 2012.

[34] W. McMahan, V. Chitrakaran, M. Csencsits, D. Dawson, I. D. Walker, B. A. Jones, M. Pritts, D. Dienno, M. Grissom, and C. D. Rahn, "Field trials and testing of the octarm continuum manipulator," in *International Conference on Robotics and Automation*, 2006, pp. 2336–2341.

[35] S. Kim, C. Laschi, and B. Trimmer, "Soft robotics: a bioinspired evolution in robotics," *Trends in Biotechnology*, vol. 31, no. 5, pp. 287–294, 2013.

[36] D. Braganza, D. M. Dawson, I. D. Walker, and N. Nath, "A neural network controller for continuum robots," *Transactions on Robotics*, vol. 23, no. 6, pp. 1270–1277, 2007.

[37] A. Gupta, C. Eppner, S. Levine, and P. Abbeel, "Learning dexterous manipulation for a soft robotic hand from human demonstrations," in *International Conference on Intelligent Robots and Systems*, 2016, pp. 3786–3793.

[38] T. George Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control strategies for soft robotic manipulators: A survey," *Soft Robotics*, vol. 5, no. 2, pp. 149–163, 2018.

[39] Z. Q. Tang, H. L. Heung, K. Y. Tong, and Z. Li, "A probabilistic model-based online learning optimal control algorithm for soft pneumatic actuators," *Robotics and Automation Letters*, vol. 5, no. 2, pp. 1437–1444, 2020.

[40] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. MIT Press, 1998.

[41] Z. I. Botev, D. P. Kroese, R. Y. Rubinstein, and P. L'Ecuyer, "The cross-entropy method for optimization," in *Handbook of Statistics*, 2013, vol. 31, pp. 35–59.

[42] "Box2d," https://box2d.org/.

[43] "Moveit," https://moveit.ros.org/.

[44] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *International Conference on Intelligent Robots and Systems*, 2017, pp. 23–30.

[45] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, "Continual learning for robotics," *arXiv preprint arXiv:1907.00182*, 2019.

[46] R. Julian, B. Swanson, G. S. Sukhatme, S. Levine, C. Finn, and K. Hausman, "Never stop learning: The effectiveness of fine-tuning in robotic reinforcement learning," *arXiv preprint arXiv:2004.10190*, 2020.