2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Robotic Object Insertion with a Soft Wrist through Sim-to-Real Privileged Training

Yuni Fuchioka[1*], Cristian C. Beltran-Hernandez[1], Hai Nguyen[1*], and Masashi Hamaya[1]

***Abstract*— This study addresses contact-rich object insertion tasks under unstructured environments using a robot with a soft wrist, enabling safe contact interactions. For the unstructured environments, we assume that there are uncertainties in object grasp and hole pose and that the soft wrist pose cannot be directly measured. Recent methods employ learning approaches and force/torque sensors for contact localization; however, they require data collection in the real world. This study proposes a sim-to-real approach using a privileged training strategy. This method has two steps. 1) The teacher policy is trained to complete the task with sensor inputs and ground truth privileged information such as the peg pose, and then 2) the student encoder is trained with data produced from teacher policy rollouts to estimate the privileged information from sensor history. We performed sim-to-real experiments under grasp and hole pose uncertainties. This resulted in 100%, 95%, and 80% success rates for circular peg insertion with 0°, +5°, and -5° peg misalignments, respectively, and start positions randomly shifted ± 10 mm from a default position. Also, we tested the proposed method with a square peg that was never seen during training. Additional simulation evaluations revealed that using the privileged strategy improved success rates compared to training with only simulated sensor data. Our results demonstrate the advantage of using sim-to-real privileged training for soft robots, which has the potential to alleviate human engineering efforts for robotic assembly.**

I. INTRODUCTION

This study addresses the problem of controlling contact-rich manipulation, focusing on object insertion tasks under grasp and hole pose uncertainties. Enabling robots to deal with such uncertainty will significantly alleviate engineering efforts in performing elaborate calibration procedures and developing jigs, as is necessary for industrial assembly.

In the presence of such uncertainties, compliance to handle environmental contacts becomes necessary for object insertion. Higher compliance is desirable for more severe uncertainty. Many approaches adopt force control with rigid robots; however, they may struggle to achieve high compliance due to the limited bandwidth of the servo controller [1]. To address this, physically soft robots incorporating passive compliance in their mechanical design have been successfully applied to object insertion tasks [1], [2], [3]. This study employs a soft wrist [4], which provides large six Degree of Freedom (DOF) deformations (Fig. 1).

Despite its advantages, soft robots' structural compliance presents challenges for control, exhibiting complex behavior due to its nonlinear dynamics [5], and limitations in partial observation of the passive DOFs. Past studies often use motion capture systems to obtain the robot pose for full observation, but this requires external sensor calibration and suffers from occlusion issues [6]. Therefore, it is desirable for an end-use deployment to not rely on such systems.

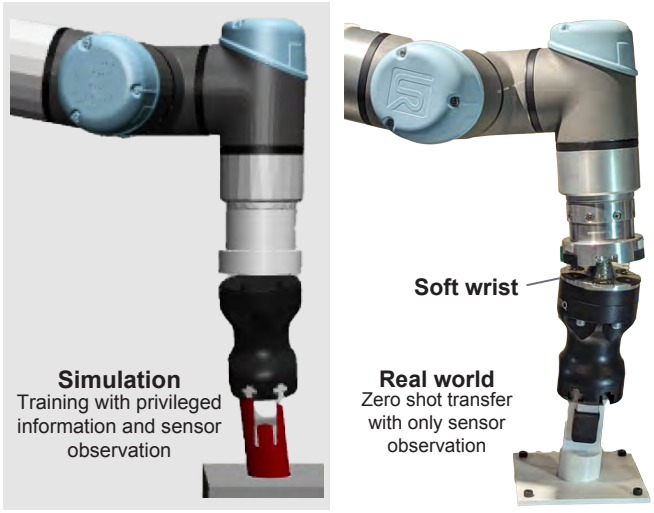

Fig. 1. We propose a sim-to-real approach for object insertion for a robot with a soft wrist through a privileged training strategy.

To address these challenges, recent approaches propose reinforcement learning (RL) and object localization techniques using force/torque (FT) sensors for insertion tasks [3], [7]. They exploit sequential FT information to explicitly or implicitly encode the relative pose between the inserted object and the hole. However, these methods require time-consuming and potentially unsafe real-world data collection, with heuristics or human demonstrations necessary.

This study proposes a sim-to-real approach to reduce the burden of real world data collection. We employ privileged reinforcement learning [8], which decomposes the learning process into two stages. First, a teacher agent with access to privileged knowledge, including the peg pose and alignment state, learns a policy to complete the task. Then, a student agent with no privileged information imitates the teacher by estimating this information from data obtainable on the robot, including arm pose and wrist FT sensor readings. We develop a simulator suitable for sim-to-real transfer based on a previous study [7] and compensate for remaining sim-to-real gaps through the judicious use of domain randomization [9].

To evaluate our proposed system, we performed real robot and simulation experiments. The real robot experiments demonstrate successful zero-shot transfer under hole and grasp pose uncertainty. The simulation experiments reveal that privileged training improves performance compared to a

[1]OMRON SINIC X Corporation, Tokyo, Japan. *Work done as an intern.

similar method without this input, as well as the effectiveness of domain randomization and including peg alignment states in the privileged information.

Contribution: we propose a sim-to-real transfer approach of contact-rich manipulation with a soft wrist using a privileged learning framework. Codes for our simulator and student-teacher training are available in the following link[1].

II. RELATED WORK

A. Soft robots for object insertion

A variety of modalities have been employed in applying softness mechanisms for robotic object insertion, ranging from springs and elastomers [1], [4], [10], [11], [12], compliant fingers driven by tendons and springs [3], fin-ray effect grippers [2], and pneumatic soft grippers [13]. Learning approaches incorporating force and tactile inputs have been effective in reconciling the partial observability of soft robot control. Azulay et al. combined reinforcement learning and heuristic force-based control in the technique that they called Haptic Glance [3]. Royo-Miquel et al. used an approximate linear model to estimate the soft robot's pose via self supervised learning on tactile sensory inputs in the frequency domain [14]. Nguyen et al. employed reinforcement learning, relying on a recurrent structure and geometrical symmetries to accelerate learning [7]. These methods demonstrate robust insertion, but require heuristics and data collection in the real world. In contrast, we use a sim-to-real approach for object insertion tasks on a robot with a soft wrist. Although [13] also used sim-to-real learning with a soft gripper robot, it did not model the compliant dynamics of the gripper in simulation. Conversely, we develop a simulation environment explicitly modeling the compliant dynamics, leveraging MuJoCo's ability to model passive joints actuated by spring-damper systems [15].

B. Proprioception in soft robotic sim-to-real transfer

Many studies have successfully achieved sim-to-real transfer for soft robot control [16]. For example, accurate position tracking has been demonstrated by obtaining dynamics using simulation with Finite Element Method (FEM) [17], [18], [19], learning inverse kinematics [20], or through vision systems [21], [22]. Agile maneuvers of a pneumatic soft robot have also been demonstrated, but the robot pose was obtained from a motion capture system [23]. Graule et al. used an approximate model using segmented rigid components for in-hand manipulation with a pneumatic hand [24]. An Extended Kalman Filter was used for joint angle estimation for a tendon-driven hand [25]. Unlike these studies, we propose a student-teacher training approach to simultaneously obtain proprioceptive state estimation and control policies.

C. Privileged training for robot control

Privileged training was initially proposed for autonomous driving [26] but has since been used for various robot applications. This includes quadrupedal locomotion [8], [27], aerial robots [28], cloth manipulation [29], push manipulation [30], and in-hand manipulation [31], [32]. Our study is the first attempt to use privileged training for a soft robot's object insertion task. In contrast to previous studies using privileged RL primarily to predict information relating to interaction states and parameters external to the robot, such as ground contact, terrain height, or grasped object properties [8], [27], [31], our use of it can be interpreted as a method of producing an estimation module for states internal to a soft robot which cannot easily be observed directly without additional sensors.

[1] https://omron-sinicx.github.io/soft-robot-sim-to-real/

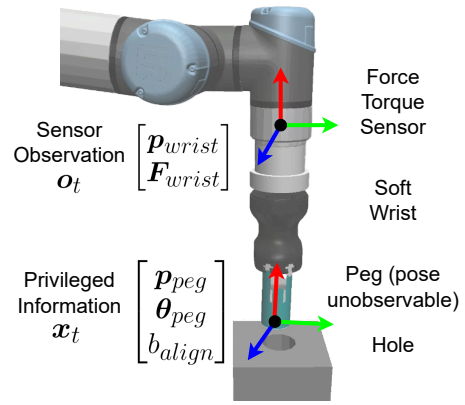

Fig. 2. The problem setting considered in this study, showing variable definitions for quantities observable from sensors, versus privileged information only accessible in simulation.

III. PRIVILEGED TRAINING

A. Problem Statement

The problem setting in this study is illustrated in Fig. 2. The Robotiq Hand-E gripper is attached to the Universal Robotics UR5e robot arm through the soft wrist connection introduced in [4] consisting of three springs. The gripper grasps peg-like objects, and the objective is to insert the grasped object into a hole. The controller has access to sensory data on the robot, including the wrist pose and readings from a 6-axis FT sensor attached to the end effector above the soft wrist.

The passive compliance in the soft wrist enables safe contact interactions suitable for this peg-in-hole task [4]. However, it also introduces a challenge for control since the gripper pose cannot be observed directly without external sensing hardware such as cameras or motion capture devices. Moreover, we assume that the in-grasp pose of the peg is not precisely known, and there is uncertainty about the hole location relative to some known nominal location. We introduce these sources of uncertainties and opt not to use external sensing hardware in light of the motivation of achieving industrial assembly tasks without expensive calibration effort and hardware necessary to remove such uncertainties. In this setting, the goal is to predict the pose of the peg and carry out the insertion task with the limited sensor data available on the robot.

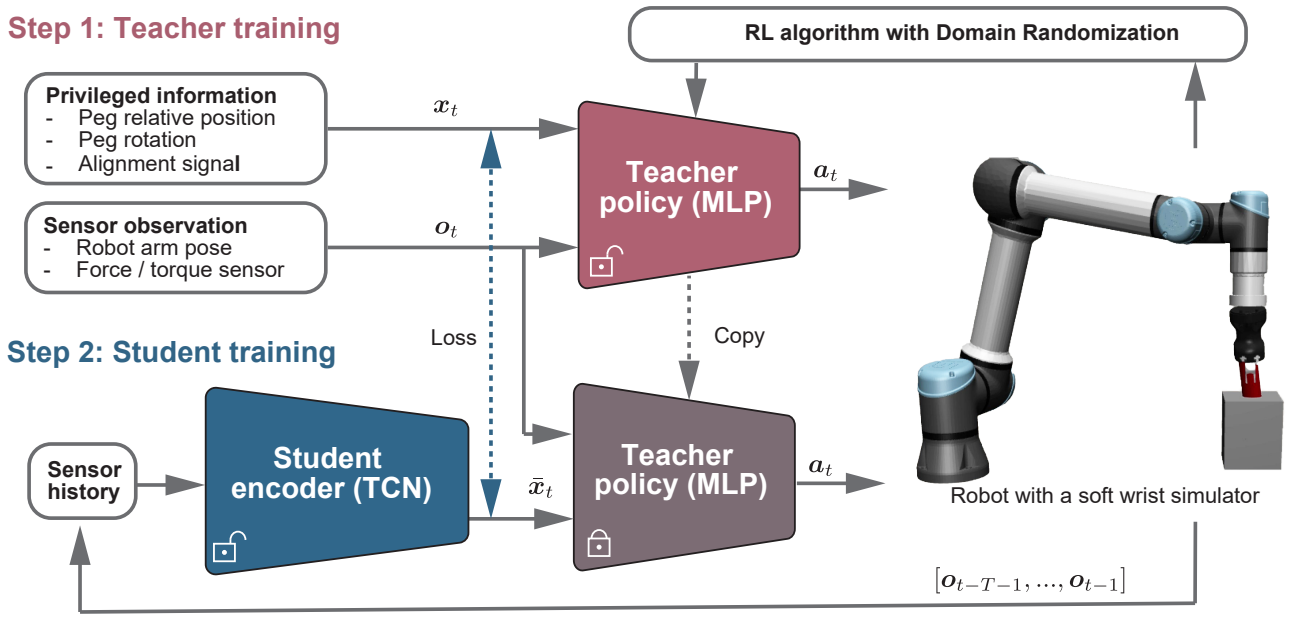

Fig. 3. Overview of the proposed framework. The privileged training has two phases: 1) teacher training: the teacher policy to control the robot is trained with sensor inputs and ground truth privileged information, and 2) student training: the student encoder is trained by running the learned teacher policy to estimate the privileged information from sensor history.

B. Overview

The proposed system is shown in Fig. 3. The inability to directly observe certain system states is solved through privileged RL, which is a sim-to-real model-free RL method based on [26]. This approach has two training phases. In the first phase, the teacher policy network is trained using privileged information only accessible in simulation. In the second phase, a student network is trained to enable imitation of the teacher's motions with access only to simulated sensor data. The result can then be transferred directly to real hardware since the necessity to rely on privileged information is removed through the second training phase.

C. Privileged Teacher Training

In the first phase of teacher training, a policy is trained to complete the peg-in-hole insertion task with access to simulator information not accessible on the real robot. We do this by formulating the problem as a Markov Decision Process (MDP) and solve it using Proximal Policy Optimization (PPO) [33] with simple 2-layer 256×256 Multi-Layer Perceptron (MLP) actor and critic networks. The definitions of each component of the MDP are outlined below.

1) State: We assume that the environment is fully observable during teacher training. In addition to the end effector pose and wrenches obtained through simulated sensors, the teacher policy has privileged access to the pose of the peg and its alignment state within the hole. The alignment state can be interpreted as a subtask switching condition facilitating more robust task completion. In practice, we remove the rotational components of the pose and wrench data, resulting in an $\mathbb{R}^6$ vector of wrist position $\boldsymbol{p}_{wrist}$ and force $\boldsymbol{F}_{wrist}$ as the simulated sensor observation $\boldsymbol{o}_t$. This was because commanding translational motions was found to be sufficient given the rotational degrees of freedom provided passively to the peg through the flexible wrist, and we also observed that torque readings have negligible magnitude compared to forces given the mechanical compliance structure.

For the privileged information $\boldsymbol{x}_t \in \mathbb{R}^{10}$, we use the position $\boldsymbol{p}_{peg}$, orientation $\boldsymbol{\theta}_{peg}$, and binary alignment state b_{align} of the peg, which returns true when the pose error between the peg and hole is smaller than a threshold. The peg orientation is represented as an $\mathbb{R}^6$ vector consisting of the first two columns of the $SO(3)$ rotation matrix, as this representation was shown to be effective for neural network regression in [34], which will be necessary for the student training phase. All observations are normalized with manually specified offset and scale factors to make values roughly lie within $[-1, 1]$ during an insertion motion.

2) Action: As described in the previous section, linear translations of the wrist were found to be sufficient to enable peg insertion without the need for rotations. In particular, we define actions for our MDP as end effector position changes relative to the current position, $\boldsymbol{a}_t = \Delta(\boldsymbol{p}_{wrist})_t$, with a maximum value of 3mm over a 50ms interval corresponding to the 20Hz policy frequency, for each of the three coordinate axes. These position commands are converted to linear trajectories through Robosuite's built-in interpolator function, so they can be tracked smoothly using joint torques calculated with the built-in implementation of Operational Space Control (OSC) [35] at the MuJoCo default simulation frequency of 500Hz. We found that the default OSC parameters in Robosuite were overly compliant for accurate pose tracking and were not reflective of motions on the physical robot, so we increased and randomized it as

discussed in Sec. IV-B.

3) Reward: The reward function used in this study is inspired by [36] and has the form

$$r = r_p - r_i - r_a + r_s, \tag{1}$$

where

(1.a) $r_p = (d_{t-1} - d_t)/0.001$ is the progress reward, where $d_t \in \mathbb{R}$ is the weighted peg distance at timestep t defined as $d_t = \sqrt{\boldsymbol{e}_t^T \boldsymbol{W} \boldsymbol{e}_t}$, with $\boldsymbol{e}_t \in \mathbb{R}^3$ representing the positional error between the peg tip and the hole, and $\boldsymbol{W} \in \mathbb{R}^{3\times3}$ being a diagonal weight matrix with elements $[1, 1, 10]$.

(1.b) $r_i = w_i(a_t)_z^2$ is the insertion reward, where $(a_t)_z$ is the action along the vertical axis, and w_i is 0.001 if the alignment condition defined below is satisfied, and 1 otherwise.

(1.c) $r_a = ||\boldsymbol{a}_t - \boldsymbol{a}_{t-1}||^2$ is the action smoothness reward designed to prevent the policy from producing vibrations and jerky motions that are not conducive to safe and successful sim-to-real transfer.

(1.d) r_s is the sparse success reward having value 1 if insertion is successful as defined by $||\boldsymbol{e}_t|| < 0.005$, and 0 otherwise.

4) Alignment: The binary alignment condition is defined as the state where translational peg position errors are within some threshold,

$$b_{align} = \sqrt{(e_t)_x^2 + (e_t)_y^2} < 0.007. \tag{2}$$

This is used both for defining weights in the insertion reward in Sec. III-C.3, and as a privileged observation in Sec. III-C.1. Modulating the vertical action weight according to peg alignment was determined in [36] to be crucial for the soft wrist peg-in-hole task, as the robot should drag the peg lightly on the flat surface adjacent to the hole until it senses through contact interactions and FT sensor readings that the hole is found and the peg is vertically aligned. Allowing excessive vertical force before alignment results in policies converging to local minima, where friction prevents the peg from moving toward the hole after contact is made.

5) Termination: Additionally to the rewards, early episode termination can be used to specify desired motions, to avoid undesired local minima and to prevent the agent from collecting data in state space regions irrelevant to the task [37]. In this study, we terminate the episode and add an additional -5 penalty to the reward whenever the weighted peg distance d_t becomes 1.2 times the value at the beginning of the episode, typically indicating a local minimum where the peg is placed at the correct height outside of the hole without attempting insertion. The same penalty is applied if the maximum allowable episode length of 200 timesteps, corresponding to 10 seconds, is reached without successful insertion. We additionally terminate the episode without penalty after successful insertion.

D. Student Training

After the RL-based teacher policy training has converged, the second phase consists of training a student encoder network to imitate teacher policy behaviors despite using only wrist position and force sensory inputs as per Sec. III-C.1. Specifically, we define a student encoder network to take a horizon $T = 20$ sensor history $[\boldsymbol{o}_{t-20}, \dots \boldsymbol{o}_{t-1}]$ corresponding to 1 second of sensor reading data as inputs, and outputs an estimate $\bar{\boldsymbol{x}}_t$ of the privileged data defined in Sec. III-C.1. This history buffer is initialized with zeros at the beginning of the episode. A Temporal Convolutional Network (TCN) [38] was used as the architecture similarly to [8]. As per [8], we train this using supervised learning with data collected through student rollouts to avoid distribution shift issues between the teacher and student [39]. The loss between predicted and ground truth values is given as:

$$\mathcal{L}(\bar{\boldsymbol{x}}, \boldsymbol{x}) = \mathcal{L}_{mse}(\bar{\boldsymbol{x}}_p, \boldsymbol{x}_p) + w\mathcal{L}_{bce}(\bar{b}_{align}, b_{align}), \tag{3}$$

where $\boldsymbol{x}_p$ is composed of the peg pose $\boldsymbol{p}_{peg}$ and orientation $\boldsymbol{\theta}_{peg}$, $\mathcal{L}_{mse}$ is the mean square error loss, $\mathcal{L}_{bce}$ is the binary cross entropy loss, and w is a weight set to 0.1. Unlike [8] and more similarly to [27], we do not retrain the policy and only train the encoder network in the second phase. Unlike both [8] and [27], however, we do not use a latent embedding of estimated variables and instead directly regress on physical quantities. In addition to providing outputs that are human-interpretable and simple to implement, we show through experiments that this was sufficient in our setting to provide accurate estimates.

IV. SIM-TO-REAL TRANSFER

A. Soft wrist simulator

In order to adapt Robosuite [40] to model our problem setting, we created custom MuJoCo XML files for the flexible wrist and gripper assemblies. We converted the Robotiq Hand-E URDF file provided in [41] to a MuJoCo XML file and rigidly attached peg objects to it under the assumption that no in-grasp slip occurs, and did not model finger actuation. We model the flexible wrist mechanism as a series of four springs, each providing one passive DOF along four axes parametrized by $k = [k_z, \kappa_x, \kappa_y, \kappa_z] = [1000, 0.5, 0.5, 5]$ and $b = [b_z, \beta_x, \beta_y, \beta_z] = [1, 0.005, 0.005, 1]$, where k, κ are stiffness constants and b, β are damping constants along linear and rotational axes respectively, and SI units used for all values. The simplification of not modeling the x-y translational DOFs of the physical 6 DOF flexible wrist was made under the observation that x-y forces applied on the peg tip primarily caused tilting along y-x axes respectively, with negligible translational motions.

Given that the spring parameter values were determined by making simulated wrist motions visually similar to the physical robot's, a more accurate system identification procedure could likely be performed through calibration involving curve fitting to motion capture data. However, we did not perform this given the project motivation described in Sec. III-A, and instead show that successful control can be achieved despite highly approximate setup procedures. Following [7], we decompose hole object geometries into convex meshes to enable contact simulation within MuJoCo.

B. Domain randomization

During teacher and student training, we apply domain randomization, a common technique in sim-to-real RL to enable robust transfer of policies to physical hardware [9]. In this study, we randomize the in-grasp angle of the peg within $\pm 5^{\circ}$ and the position of the hole within $\pm 10\mathrm{mm}$.

While the above randomization is designed to enable practical task generalization, we additionally randomize gains in the OSC controller as mentioned in Sec. III-C.2. Position controller gains are randomized uniformly on a logarithmic scale between $[10^3, 10^4]$ with the same values used for all 6 DOF. This was done to account for the discrepancy that the simulation and OSC formulation assume a torque-controlled robot, whereas the physical UR5e has mechanically stiff joints and precise position control implemented at a low level. While one could perform system identification to determine the simulated OSC parameters that best correspond to the physical robot motions, we instead opted for randomization, as we expect generally robust policies to emerge due to the high sensitivity of motions to gain parameters that the policy must learn to adapt to.

The initial pose of the robot is also randomized by adding Gaussian noise to all six joints with a standard deviation of 0.002 rad. This is with respect to a configuration where the peg nearly touches the flat surface adjacent to the hole. All randomizations are applied at the beginning of every episode, and unless otherwise noted, they are uniformly distributed.

C. Real robot controller

In order to execute trained networks to control the physical robot, the code in [42] was used, which provides Python interfaces to Universal Robot's ROS drivers [43], providing convenient interfacing with our PyTorch [44] based training pipeline. Important features include the ability to specify the final position and time of end effector trajectories and the ability to interrupt commands to ensure a 20Hz update frequency in case a path execution takes longer than the specified time. Given the mismatch of compliant OSC-based control in simulation and high precision position control on the physical robot as discussed in Sec. IV-B, ensuring that position displacement targets were tracked accurately and at the same frequency for both simulated and real robots was found to be crucial for sim-to-real transfer.

V. RESULTS

A. Overview

We perform a series of experiments on real and simulated robots to characterize the performance of the proposed method, as well as the effect of various design parameters on performance. In particular, we 1) show the overall sim-to-real performance and report success rates on the real robot, 2) demonstrate the advantage of using the two-stage privileged RL approach, 3) evaluate the importance of domain randomization, and 4) visually show the effectiveness of estimates produced by our student encoder.

For training, the teacher policy and student encoder were updated with 10,000 and 5,000 iterations, respectively, with each iteration consisting of 1,000 timesteps. The teacher and student training took approximately 24 and 12 hours, respectively, on a workstation with an AMD Ryzen Threadripper Pro 5975WX CPU and an RTX 4080 GPU.

A circular peg with 40 mm outer diameter and a hole with 42 mm inner diameter was used to train the policy for all experiments, with similarly dimensioned pegs used for tests involving other peg shapes. In the real robot experiments, we used 3D-printed pegs and holes.

TABLE I
NUMBER OF SUCCESSES OF REAL ROBOT EXPERIMENTS

	Shape			
	Circle		Square (Unseen)	
Misalignment	Best	Median	Best	Median
0 deg	20/20	20/20	14/20	9/20
+5 deg	19/20	16/20	9/20	2/20
-5 deg	16/20	18/20	11/20	17/20

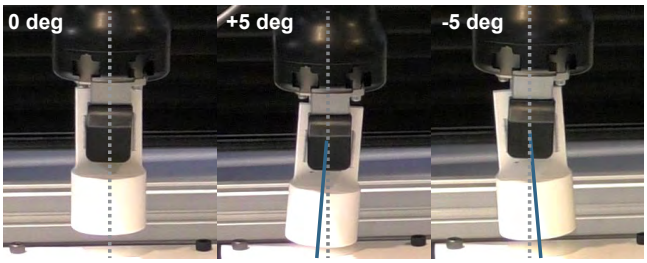

Fig. 4. Pegs' misalignments for the real-robot experiments.

B. Sim-to-Real Peg-in-Hole Insertion

Table I shows the number of successes of the real-robot experiments with circle and square pegs under uncertainty, where 0 and ± 5 ° misalignments were provided (Fig. 4), and the initial x and y positions were uniform-randomly shifted in a range ± 10 mm from a default position. We performed 20 trials on each condition. We used teacher and student networks trained on the circular peg for both insertion tasks, meaning that the robot never saw the square peg during training. The experiment was performed by selecting the best and median-performing teacher-student network pair as evaluated in simulation. This variation across random seeds is illustrated in Fig. 6 and elaborated in the next section.

The robot successfully completed circular peg insertion in most trials, despite misalignments. While the best performing network showed a 70 % success rate for the unseen square peg insertion without misalignment, the median-performing network showed a 45 % success rate. Both median and best performance decreased for the square peg in the presence of misalignment. However, the median policy actually outperformed the best policy in the -5 ° misalignment case, as the specific motion learned by the median policy was well suited to insertion with negative angle peg insertions.

The attached video, as well as the snapshots provided in Fig. 5, demonstrate zero-shot sim-to-real transfer of the contact-rich task of peg-in-hole insertion despite various

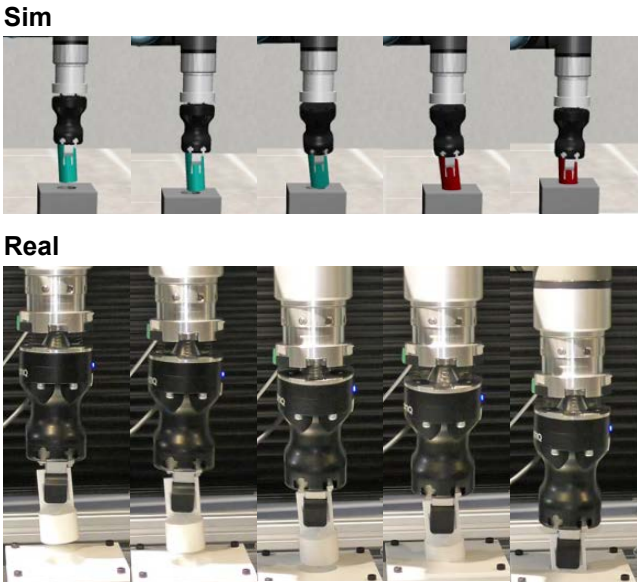

Fig. 5. Snapshots of successful insertion in simulation and real-world. For the simulation, the peg's color turning red indicates that the student encoder detected the alignment of the peg.

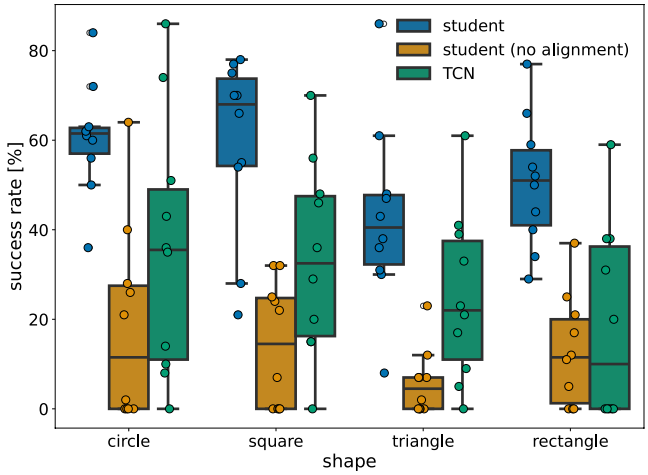

Fig. 6. Success rates between the proposed method (teacher-student) and the baseline (TCN) in simulation. The scatter points show the success rates of each seed, with box plots providing summary statistics. Our proposed method, including the alignment state, showed a higher success rate than the baseline. The policy generalizes to shapes unseen during training, since only circular pegs were used for training.

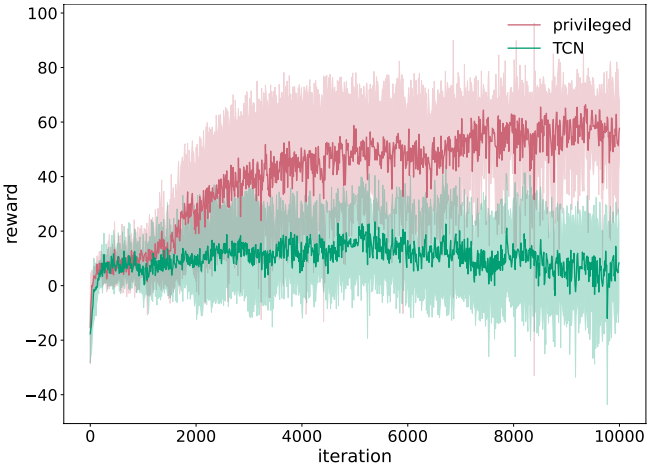

Fig. 7. The learning curve of the MLP policy with privileged information and TCN policy without it. The privileged information helped the learning process.

sources of uncertainties and limited sensor data. In Fig. 5, the color of the peg additionally provides visualization of the alignment state described in Sec. III-C.4.

C. Privileged Training

Fig. 6 shows the success rate of simulated peg-in-hole experiments to evaluate the effectiveness of the two-stage privileged training setup. "student" denotes our proposed method, whereas "TCN" shows a similar baseline used in [8] where a single RL training stage is used only on simulated sensor data without privileged data, with a TCN architecture used for both actor and critic networks for a fair comparison with the two-stage approach. Additionally, "no alignment" shows the case where peg alignment was removed from the privileged data in our proposed method. We trained these methods with 10 random seeds and evaluated them for 100 trials by uniformly randomizing the dynamics parameters, hole, and grasp poses in the same range as the training. Results are shown for four types of peg shapes, using policies trained only for the circular peg.

The results show the clear advantage of using two-stage privileged training over direct policy training with a TCN, as well as the importance of including alignment along with peg pose in the privileged data. The generalization across unseen peg shapes shows the extrapolation capabilities of RL-based controller synthesis. Despite the unexpected result that the circular peg's median success rate is outperformed by the square peg, which was unseen during training, the success rate of the square peg has a larger variance. In Fig. 6, it can be observed that the simulated success rates are generally lower than that of the physical experiment results provided by Table I. This is likely because the simulated training environment is harsher than the physical testing environment, since the joint controller gains are randomized during training in an effort to make sim-to-real robust, even though the joint stiffnesses are likely fixed on the physical robot.

In addition to evaluating success rates on converged policies, Fig. 7 shows learning curves for the privileged teacher training compared against the unprivileged TCN policy baseline. The thick line and shaded regions show the mean and standard deviation of training progress across 10 experiments, each with different random seeds. This plot shows that the privileged data helps RL training despite the potential architectural advantage that the TCN policy may have over the far smaller MLP used for the privileged policy.

D. Domain Randomization

Next, we perform a sim-to-sim evaluation of domain randomization. For this, we train the privileged teacher on a simulated environment with one of the randomizations

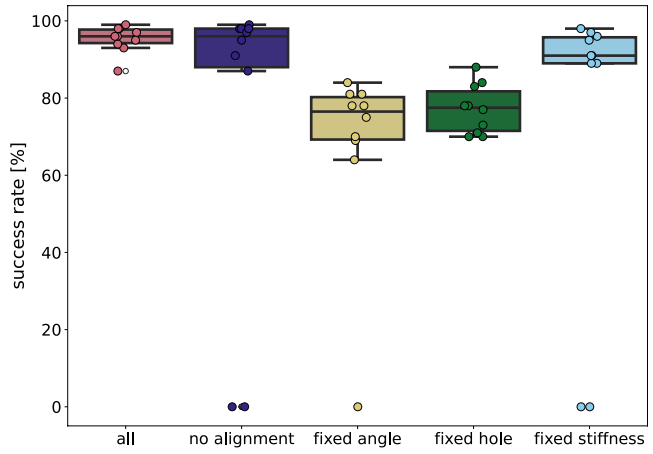

Fig. 8. Success rates of the teacher policy in simulation, removing the alignment state from the privileged information and turning the randomization off during training. The success rate is degraded without the hole and grasp pose randomization.

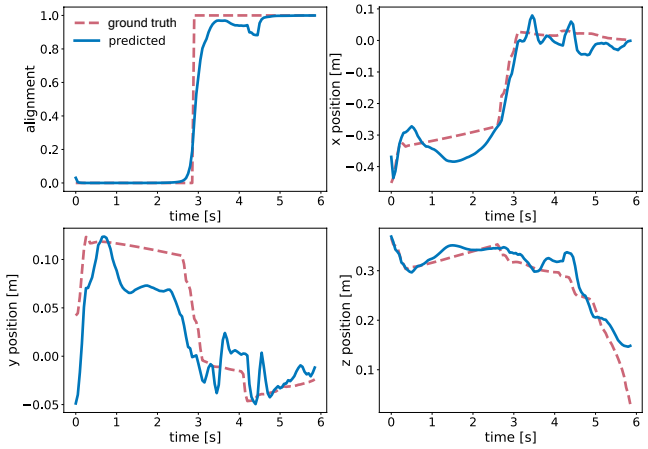

Fig. 9. Student encoder predictions for the alignment state, x, y, and z positions. The encoder was successful in detecting the peg's alignment, which is defined in Sec. III-C.4.

turned off and measure success rates for the same privileged policy tested on the randomized environment used in our proposed method to determine how well the policy adapts to variations unseen during training (Fig. 8). The number of evaluation trials was the same as the previous section V-C. "all" has all randomizations turned on, representing the case where the policy is tested on the same domain that it is trained on, "fixed angle," "fixed hole", and "fixed stiffness" respectively represent cases where the in-grasp peg angle, hole position deviation, and OSC gain parameter randomizations are turned off during training. The results show that, especially for uncertainties in grasp and hole pose, domain randomization during training helps significantly to adapt to uncertainties that may exist during deployment.

Although unrelated to domain randomization, the "no alignment" column in Fig. 8 shows the privileged teacher policy performance with the peg alignment state removed from the privileged information. The fact that the performance is not degraded significantly compared to the "all" case, despite the large performance degradation observed for the student performance in Fig. 6, suggests that including the peg alignment state becomes advantageous only after the second phase of privileged training. A possible explanation is that the student encoder network can better predict the alignment state compared to other quantities, providing useful inputs to the teacher policy when privileged data is estimated imperfectly by the student encoder, as opposed to when ground truth information is provided.

E. Student Network Estimation

It was hypothesized in the previous section that the alignment state can be better predicted by the student encoder compared to other physical quantities. This is confirmed by Fig. 9, which shows the physical quantities predicted by the student encoder compared to ground truth values, as evaluated in simulation. These plots illustrate the generally high quality of the estimation. The ability to produce these plots directly without having to train a separate decoder network purely for visualization, as was done in [8], is useful for debugging purposes and demonstrates the advantage of performing regression directly on physical quantities rather than on latent embeddings, as discussed in Sec. III-D.

VI. CONCLUSION AND FUTURE WORK

In this study, we present a method for controlling peg-in-hole insertion for an industrial manipulator robot with a soft wrist and uncertainties in physical setup, relying only on commonly available sensors without additional sensing hardware or calibration procedures. We do this through zero-shot sim-to-real transfer of neural networks trained through a privileged learning approach, leveraging the fact that training happens purely in simulation to use privileged data unavailable on the physical system.

A common failure mode was the peg "missing" the hole and never "finding" it through contact, as the policy only attempts insertion once it detects a hard impact between the peg and the inner wall of the hole through the release of energy stored in the spring from compression.

We expect that further reduction of the sim-to-real gap may be achieved through further randomization of key physical parameters such as spring stiffness and damping or surface friction, or through inclusion of randomization parameters in the privileged information. Additionally, rewards can be added to encourage faster task completion to prevent this behavior. Further future work could consider practical insertion scenarios with tighter tolerance parts, or electrical connectors.

ACKNOWLEDGMENTS

We wish to thank the members of the ETH Zürich Robotic Systems Lab for their valuable discussions. This study is supported by JST ACT-X, Grant Number JPMJAX22AC.

REFERENCES

[1] Q. Zhang, Z. Hu, W. Wan, and K. Harada, "Compliant peg-in-hole assembly using a very soft wrist," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 17–24, 2023.

[2] R. M. Hartisch and K. Haninger, "High-speed electrical connector assembly by structured compliance in a finray-effect gripper," *IEEE/ASME Transactions on Mechatronics*, pp. 1–10, 2023.

[3] O. Azulay, M. Monastirsky, and A. Sintov, "Haptic-based and SE(3)-aware object insertion using compliant hands," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 208–215, 2022.

[4] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 8752–8757.

[5] D. Kim, S.-H. Kim, T. Kim, B. B. Kang, M. Lee, W. Park, S. Ku, D. Kim, J. Kwon, H. Lee *et al.*, "Review of machine learning methods in soft robotics," *Plos One*, vol. 16, no. 2, p. e0246102, 2021.

[6] O. Yasa, Y. Toshimitsu, M. Y. Michelis, L. S. Jones, M. Filippi, T. Buchner, and R. K. Katzschmann, "An overview of soft robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 6, pp. 1–29, 2023.

[7] H. Nguyen, T. Kozuno, C. C. Beltran-Hernandez, and M. Hamaya, "Symmetry-aware reinforcement learning for robotic assembly under partial observability with a soft wrist," in *IEEE International Conference on Robotics and Automation*, 2024, pp. 9369–9375.

[8] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science Robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[9] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 3803–3810.

[10] T. Goto, K. Takeyasu, and T. Inoyama, "Control algorithm for precision insert operation robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 10, no. 1, pp. 19–25, 1980.

[11] D. E. Whitney, "Quasi-static assembly of compliantly supported rigid parts," *Journal of Dynamic Systems, Measurement, and Control*, vol. 104, no. 1, pp. 65–77, 1982.

[12] 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 *IEEE-RAS International Conference on Humanoid Robotics*, 2017, pp. 67–74.

[13] S. Brahmbhatt, A. Deka, A. Spielberg, and M. Müller, "Zero-shot transfer of haptics-based object insertion policies," in *IEEE International Conference on Robotics and Automation*, 2023, pp. 3940–3947.

[14] J. Royo-Miquel, M. Hamaya, C. C. Beltran-Hernandez, and K. Tanaka, "Learning robotic assembly by leveraging physical softness and tactile sensing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2023, pp. 6469–6476.

[15] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033.

[16] M. Bächer, E. Knoop, and C. Schumacher, "Design and control of soft robots using differentiable simulation," *Current Robotics Reports*, vol. 2, no. 2, pp. 211–221, 2021.

[17] T. Du, J. Hughes, S. Wah, W. Matusik, and D. Rus, "Underwater soft robot modeling and control with differentiable simulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4994–5001, 2021.

[18] M. Dubied, M. Y. Michelis, A. Spielberg, and R. K. Katzschmann, "Sim-to-real for soft robots using differentiable fem: Recipes for meshing, damping, and actuation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5015–5022, 2022.

[19] J. Z. Zhang, Y. Zhang, P. Ma, E. Nava, T. Du, P. Arm, W. Matusik, and R. K. Katzschmann, "Sim2real for soft robotic fish via differentiable simulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 12 598–12 605.

[20] G. Fang, Y. Tian, Z.-X. Yang, J. M. Geraedts, and C. C. Wang, "Efficient jacobian-based inverse kinematics with sim-to-real transfer of soft robots by learning," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 6, pp. 5296–5306, 2022.

[21] C. Shentu, E. Li, C. Chen, P. T. Dewi, D. B. Lindell, and J. Burgner-Kahrs, "MoSS: Monocular shape sensing for continuum robots," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1524–1531, 2024.

[22] U. Yoo, H. Zhao, A. Altamirano, W. Yuan, and C. Feng, "Toward zero-shot sim-to-real transfer learning for pneumatic soft robot 3d proprioceptive sensing," in *IEEE International Conference on Robotics and Automation*, 2023, pp. 544–551.

[23] R. Jitosho, T. G. W. Lum, A. Okamura, and K. Liu, "Reinforcement learning enables real-time planning and control of agile maneuvers for soft robot arms," in *Conference on Robot Learning*, 2023, pp. 1131–1153.

[24] M. A. Graule, T. P. McCarthy, C. B. Teeple, J. Werfel, and R. J. Wood, "Somogym: A toolkit for developing and evaluating controllers and reinforcement learning algorithms for soft robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4071–4078, 2022.

[25] Y. Toshimitsu, B. Forrai, B. G. Cangan, U. Steger, M. Knecht, S. Weirich, and R. K. Katzschmann, "Getting the ball rolling: Learning a dexterous policy for a biomimetic tendon-driven hand with rolling contact joints," in *IEEE-RAS International Conference on Humanoid Robots*, 2023, pp. 1–7.

[26] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Conference on Robot Learning*, 2020, pp. 66–75.

[27] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "RMA: Rapid motor adaptation for legged robots," in *Robotics: Science and Systems*, 2021.

[28] A. Loquercio, E. Kaufmann, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza, "Learning high-speed flight in the wild," *Science Robotics*, vol. 6, no. 59, p. eabg5810, 2021.

[29] X. Lin, Y. Wang, Z. Huang, and D. Held, "Learning visible connectivity dynamics for cloth smoothing," in *Conference on Robot Learning*, 2022, pp. 256–266.

[30] M. Kim, J. Han, J. Kim, and B. Kim, "Pre-and post-contact policy decomposition for non-prehensile manipulation with zero-shot sim-to-real transfer," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2023, pp. 10 644–10 651.

[31] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik, "General in-hand object rotation with vision and touch," in *Conference on Robot Learning*, 2023, pp. 2549–2564.

[32] L. Röstel, J. Pitz, L. Sievers, and B. Bäuml, "Estimator-coupled reinforcement learning for robust purely tactile in-hand manipulation," in *IEEE-RAS International Conference on Humanoid Robots*, 2023, pp. 1–8.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[34] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5745–5753.

[35] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[36] M. Hamaya, K. Tanaka, Y. Shibata, F. Von Drigalski, C. Nakashima, and Y. Ijiri, "Robotic learning from advisory and adversarial interactions using a soft wrist," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3878–3885, 2021.

[37] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–14, 2018.

[38] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[39] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 627–635.

[40] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu, "robosuite: A modular simulation framework and benchmark for robot learning," in *arXiv preprint arXiv:2009.12293*, 2020.

[41] C. C. Beltran-Hernandez, "robotiq," https://github.com/cambel/robotiq, 2024.

[42] ——, "ur3," https://github.com/cambel/ur3, 2024.

[43] Universal Robots, "Universal_robots_ros_driver," https://github.com/UniversalRobots/Universal_Robots_ROS_Driver, 2024.

[44] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *Conference on Neural Information Processing Systems Workshop*, 2017.