

PolyFit: A Peg-in-hole Assembly Framework for Unseen Polygon Shapes via Sim-to-real Adaptation

Geonhyup Lee*, Joosoon Lee*, Sangjun Noh, Minhwan Ko, Kangmin Kim, and Kyoobin Lee†

Abstract—The study addresses the foundational and challenging task of peg-in-hole assembly in robotics, where misalignments caused by sensor inaccuracies and mechanical errors often result in insertion failures or jamming. This research introduces PolyFit, representing a paradigm shift by transitioning from a reinforcement learning approach to a supervised learning methodology. PolyFit is a Force/Torque (F/T)-based supervised learning framework designed for 5-DoF peg-in-hole assembly. It utilizes F/T data for accurate extrinsic pose estimation and adjusts the peg pose to rectify misalignments. Extensive training in a simulated environment involves a dataset encompassing a diverse range of peg-hole shapes, extrinsic poses, and their corresponding contact F/T readings. The study proposes a sim-to-real adaptation method for real-world application, using a sim-real paired dataset to enable effective generalization to complex and unseen polygon shapes. Real-world evaluations demonstrate substantial success rates of 96.7% and 91.3%, highlighting the robustness and adaptability of the proposed method. Videos of data generation and experiments are available online at <https://sites.google.com/view/polyfit-peginhole>.

I. INTRODUCTION

Peg-in-hole assembly using robotics is a fundamental yet challenging task. Misalignment issues often occur between assembled parts due to sensor inaccuracies and mechanical errors, potentially resulting in system damages like insertion failures or jamming. Various methods based on compliance control and heuristic algorithms have been proposed to address misalignment in peg-in-hole assembly [1]–[4]. However, traditional methods require manual calibration for specific peg-hole configurations or elaborate modeling of contact dynamics. Such requirements complicate the development of assembly strategies for diverse component shapes and limit to generalize across various shapes.

Learning-based peg-in-hole methodologies have been proposed to address the aforementioned limitations [5]–[9]. One study guided a peg into a hole by visually minimizing the seam through vision-based reinforcement learning (RL), showcasing adaptability to unseen shapes [5]. However, vision-based methods are sensitive to external conditions such as occlusion, varying lighting conditions, and camera placement.

Meanwhile, methodologies based on Force/Torque (F/T) sensors operate through direct contact with objects and exhibit less sensitivity to environments challenging for vision sensors. Several studies have utilized contact F/T data

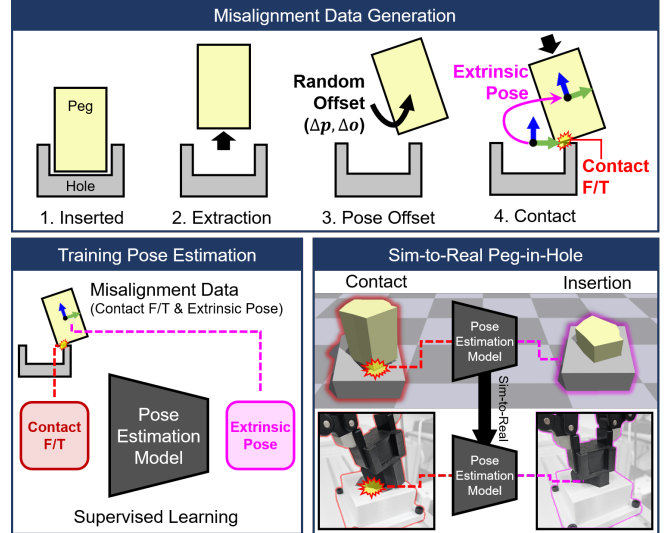


Fig. 1. Schematic illustration of addressing misalignment using the proposed PolyFit framework for a peg-in-hole task.

sampling in the real world for peg-in-hole tasks, employing learned contact dynamics and an insertion policy to transfer to new shapes and validate against untrained ones [6], [7]. Demonstrating adaptability to unseen objects, collecting real-world data can be challenging due to efficiency reduction and potential damage risk to the robot. In response to these challenges, contact-based sim-to-real adaptation has emerged as a promising strategy to use simulations for data collection [8], [9]. While simulation environments offer a safer and richer context for learning, defining and randomizing dynamic parameters present substantial obstacles to efficient sim-to-real adaptation.

The proposed PolyFit introduces a significant paradigm shift in peg-in-hole assembly approaches, moving from a RL methodology to a supervised learning-based approach. Addressing the limitations of traditional methods, PolyFit is an F/T-based supervised learning framework designed for 5-degrees-of-freedom (DoF) peg-in-hole assembly (Fig. 1). The framework undergoes extensive training in simulation and is adapted for real-world environments. The simulation dataset includes a diverse array of peg-hole shapes and extrinsic poses between the peg and hole, utilizing the concept of extrinsic poses introduced by Ma et al. [10]. This dataset also incorporates corresponding contact F/T readings. PolyFit utilizes this simulation dataset to estimate extrinsic poses using only F/T data, enabling it to adjust

*Equal Contribution

All authors are with the School of Integrated Technology (SIT), Gwangju Institute of Science and Technology (GIST), Cheomdan-gwagiro 123, Buk-gu, Gwangju 61005, Republic of Korea. † Corresponding author: Kyoobin Lee kyoobinlee@gist.ac.kr

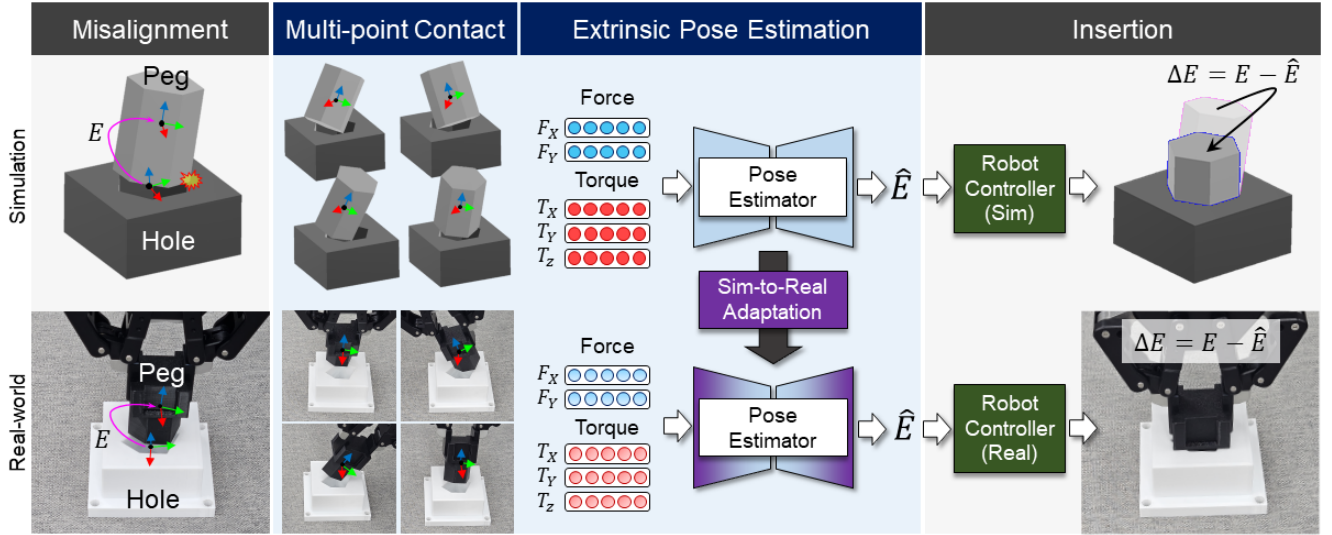


Fig. 2. Overview of the proposed PolyFit framework, a supervised learning-based strategy for peg-in-hole insertion.

the peg pose to correct misalignments. A multi-point contact strategy is implemented to enhance pose estimation accuracy, considering that identical F/T data can occur in different poses. PolyFit is trained on polygonal shapes with 4 to 6 vertices, and its robustness is validated through tests on more complex, unseen configurations with 7 to 10 vertices, demonstrating its ability to generalize to new configurations. Sim-to-real adaptation methods are introduced to extend the model to the real-world. The adaptation method utilizes a minimal set of sim-real paired data, capturing identical contact F/T and poses in simulated and real environments. This adaptability is validated through peg-in-hole experiments on unseen polygon shapes, confirming the model's precision and effective adaptability to real-world applications. An overview of the PolyFit model is presented in Fig. 2. Despite relative errors induced by perception issues or noise, PolyFit ensures successful task completion through iterative multi-point contact and F/T data acquisition.

The contributions of this work can be summarized as follows:

- 1) Development of a supervised learning framework for 5-DoF peg-in-hole task, leveraging only F/T data for extrinsic pose estimation and misalignment correction.
- 2) Introduction of a sim-to-real adaptation method for a pose estimation model, and validation of peg-in-hole assembly in the real-world, demonstrating generalizability across untrained polygon shapes.

II. RELATED WORK

A. Learning-based Method for Assembly

Vision-based Approach: Significant research has been conducted on vision-based methodologies for robotic peg-in-hole tasks [11]–[15]. Studies focusing on specialized reward mechanisms, such as sparse rewards and goal-oriented images, have been carried out using RL frameworks for complex industrial insertion problems [11], [12]. However,

these methods have been primarily tested on a limited number of pegs and holes. Therefore, approaches that utilize a “residual policy” have been explored to extend capabilities across a broader array of assembly components and enhance efficiency and flexibility in real connectors used in robotic assembly [13], [14]. However, these strategies encountered limitations in adapting to new configurations. Offline RL was employed to reduce the requisite exploration in traditional RL and facilitate the efficient learning of assembly policies across various configurations, with online fine-tuning concurrently explored for swift adaptation to new configurations [15]. However, this methodology encountered limitations owing to the dimensional constraints of the action space that only considered positional errors. As previously noted, vision-based methods face challenges due to sensor noise, occlusion, and variable lighting conditions. In response to these challenges, the proposed PolyFit framework employs F/T modalities to execute 5-DoF peg-in-hole tasks, thereby offering a robust performance that effectively mitigates the challenges encountered by vision sensors.

Force-based Approach: The limitations of vision-based methods have led researchers to explore alternative approaches, particularly those utilizing force-based methods. RL with F/T information has been employed to learn force controller action policies, resulting in a precise insertion strategy [8], [16]. However, this method faced challenges when dealing with unseen shapes. To overcome this limitation, subsequent research focused on shape generalization in assembly tasks [6], [17]. A method was developed using contact data sampling that efficiently learned a dynamic model using multi-pose F/T state representations, enabling quick adaptation to new peg-in-hole configurations with minimal trials [6]. Offline meta-RL was introduced to enhance learning across diverse components and accelerate adaptation through online fine-tuning [17]. Despite these advancements, significant challenges persist. The acquisition of real-world

data introduces concerns regarding efficiency and safety [6], [8], [17]. In response to these challenges, this research proposes a methodology for conducting peg-in-hole tasks based on F/T, bypassing the inefficiencies and safety concerns of real-world data acquisition by employing a dynamic simulator for data collection and learning. Consequently, the proposed method ensures robust performance across different component geometries and conditions without the necessity for extensive data collection in the real world.

B. Sim-to-Real Adaptation for Assembly

Adapting from simulation to real-world environments presents a fundamental challenge in robotics owing to inherent discrepancies between the two domains, including differences in physical dynamics and unmodeled factors. Various research initiatives aim to identify and mitigate these disparities, ensuring the effective transfer of learned behaviors from virtual to physical environments [18]–[21]. Domain-randomization techniques, such as visual randomization [5], [12], [22] and dynamic parameter randomization [8], [9], have demonstrated robustness across a variety of real-world conditions in assembly tasks. Furthermore, scaled robot force was output continuously as a strategy [23], and scripted actions in simulations and reality have been executed for system identification [24]. However, existing methods that were focused on adjusting dynamic parameters between simulations and reality often faced scalability and flexibility issues due to the complexity of these parameters. The proposed method counters these limitations by directly adapting contact dynamics through minimal-paired data, capturing the same contact F/T and pose in sim-real environments, and enabling swift, data-efficient adaptation and robust performance across real-world scenarios.

III. PROBLEM DESCRIPTION

This study aimed to develop a framework capable of accurately estimating and correcting extrinsic pose misalignments during peg-in-hole assembly tasks using F/T measurements within a supervised learning context.

While peg and gripper poses are known, the extrinsic pose between the peg and hole remains uncertain, leading to misalignments and subsequent contact during assembly. Force control is applied to stabilize F/T readings for the pose estimation model. Observing F/T in isolation can yield identical readings for different poses. Therefore, a series of m rotational actions, referred to as multi-point contact, are executed to acquire comprehensive contact pose information. All F/T values discussed in this paper are obtained from these multi-point contact operations. The model inputs include forces F_x, F_y and torques T_x, T_y, T_z , excluding the constant z-direction force F_z . In mathematical terms, F/T is articulated as $C \in \mathbb{R}^{5 \times k}$, encompassing force $F \in \mathbb{R}^{2 \times k}$ and torque $T \in \mathbb{R}^{3 \times k}$, where $k = m + 1$ and m signify the number of additional contact points derived from multi-point contact. The 5-DoF extrinsic pose of the initial contact state, denoted as $E \in \mathbb{R}^{5 \times 1}$, is estimated without considering the component p_z . It incorporates the relative position $p =$

$(p_x, p_y) \in \mathbb{R}^{2 \times 1}$ and orientation $o = (o_x, o_y, o_z) \in \mathbb{R}^{3 \times 1}$ of the peg with respect to the hole. The peg-in-hole procedure follows a structured framework that sequentially involves multi-point contact, pose estimation, and peg control based on an estimated extrinsic pose. This framework demonstrates applicability in simulated and real-world environments.

IV. METHODOLOGY

A. Simulation Data Generation

Simulation environment: A simulation environment was established within the Isaac Gym simulator [25], as depicted in Fig. 3. This environment focused predominantly on the peg-hole interaction by intentionally omitting robotic elements. A 6-DoF motion system simulated peg movements, and compliance parameters were adjusted to mirror real-world robotic controllers. It should be noted that these parameters can slightly vary between different real-world implementations. Contact F/Ts were recorded via a centrally located F/T sensor on the peg, with measurements averaged over 50 simulation steps for consistency. Signed Distance Function (SDF)-based collisions with a set resolution of 512 were employed to accommodate the minimal tolerance between the peg and hole [26]. Various peg and hole CAD models were developed for seven polygon configurations ranging from 4 vertices to 10 vertices, creating 20 unique shapes per configuration, totaling 140 models. The polygon geometry was defined by selecting several vertices n and then randomly assigning radius r and angle θ values, ensuring the total angle sum of 360° ($\sum_{i=1}^n \theta_i = 360^\circ$). Radius values $\{r_1, \dots, r_n\}$ ranged from 10 to 20 mm. Shapes with angles summing to 180° across three consecutive vertices were excluded to ensure diversity. Furthermore, a tolerance of 1 mm was adhered to for pegs and holes.

Peg-hole Misalignment Dataset: In this study, as illustrated in the misalignment data generation process in Fig. 1, we generated peg-in-hole misalignment data to train an extrinsic pose estimation model. This process involved extracting a peg from a hole, applying a random pose offset, and subsequently recording the contact F/T and the extrinsic

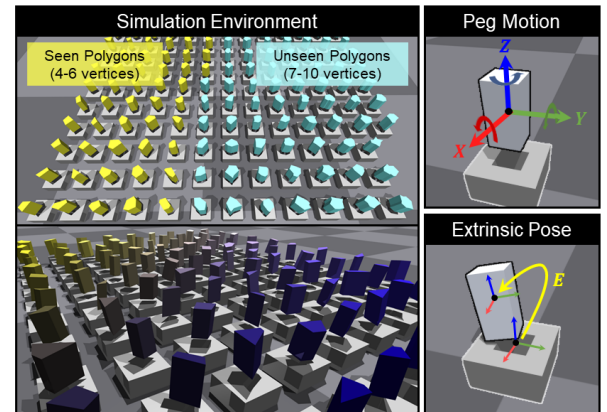


Fig. 3. Isaac Gym-based simulation environment for data generation.

pose at the point of contact. A random offset sampling strategy introduced random offsets of position and orientation, Δp and Δo , in the peg-hole system to create diverse pose configurations, enabling comprehensive data collection for both extrinsic pose E and corresponding contact F/T C . Our random offset strategy introduced position and orientation offsets, Δp and Δo , within $\pm 10\text{mm}$ and $\pm 5^\circ$, respectively. Subsequently, the PolyFit framework employed a multi-point contact strategy, as outlined in Section III, involving a sequence of m additional contacts. In this study, we set m to 4. Each defined by a controlled $\pm 10^\circ$ rotation in the x and y orientations relative to the contact direction. This method enhanced the contact F/T data captured in the matrix C , consolidating measurements from both the initial C_0 and subsequent $\{C_1, \dots, C_m\}$ contacts. Paired with the extrinsic pose E of the initial contact. The dataset included 448 million data points across 140 distinct shapes, integrating measurements from $m + 1$ contact points, which consist of C and E collected over 20 timesteps. For training purposes, an single average value derived from these timesteps is used. Each shape includes 3,200 misalignments, with 1,200 allocated for training and 1,000 for both validation and testing. The data generation process was conducted on an RTX 3090 Ti GPU, taking about 3 min per shape.

B. Extrinsic Pose Estimation Network

The architecture for an extrinsic pose estimation network suitable for robotic operations utilized a lightweight design with a multi-layer perceptron (MLP), as shown in Fig. 4. Inputs comprising F/T data C , concatenated with values from multi-contact points, were processed through distinct three-layered MLP encoders, generating separate feature sets for F/T. These feature sets were then concatenated and passed through a fusion module consisting of a three-layered MLP to produce a unified feature. The network's outputs, the extrinsic pose \hat{E} at the misalignment, were estimated by two separate three-layered MLP heads. Implemented using PyTorch, the network employed the Mean Absolute Error (MAE) as the loss function and the Adam optimizer with an initial learning rate of 0.001. A cosine annealing scheduler was used in 100-epoch cycles for learning rate management during the 300-epoch training. The training, with a batch size of 256, was executed on an RTX 3090 and completed in approximately 30 min.

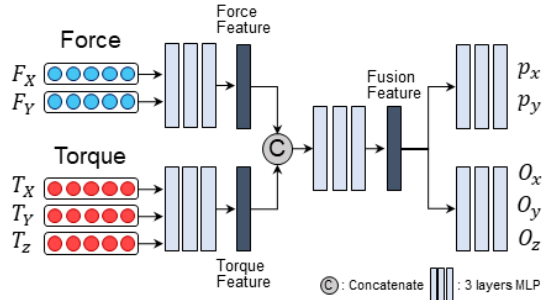


Fig. 4. Architecture of the extrinsic pose estimation network.

C. Sim-to-Real Adaptation for Extrinsic Pose Estimation Network

Sim-to-real adaptation is crucial for addressing the performance decline of models trained solely in simulations when applied to real-world tasks. These models often underperform in real scenarios, despite being well-trained in simulations. This issue arises mainly due to the differences in patterns between simulated and real-world data. Considering the significant effort and resources required to amass a diverse set of real-world training data, an approach was developed to align real-world contact F/T and feature extraction capacity of extrinsic pose estimation models with those from simulations. To address this, the proposed approach utilized sim-real paired data capturing contact F/T data with identical shapes and poses in simulated and real environments, as illustrated in Fig 5. Proposed sim-to-real adaptation method include data-level adaptation (DLA), feature-level adaptation (FLA), and real pose fine-tuning (RPF).

For DLA, an MLP-based conversion model trained on a sim-real paired dataset transformed real-world contact F/T to the matched F/T simulation data. The transformation model consisted of two three-layer MLP encoders that used real-world force (F^R) and torque (T^R) as inputs. The features extracted from each encoder were concatenated and passed through a three-layer MLP for fusion. The fused feature was then utilized to output simulation force and torque through separate three-layer MLP generation heads. The model was trained through the MAE loss with the simulation F/T data (F^S and T^S) as the ground truth for sim-to-real adaptation, as illustrated in Fig 5. In the FLA, the simulation model was assumed to have the ability to extract essential features for pose estimation that were robust to shapes and poses. The feature extraction capability was transferred to the real-world model using knowledge distillation through cosine similarity loss. The targets of knowledge distillation were force, torque, and fusion features, as shown in Fig.4. Finally in the RPF phase, the model was fine-tuned for the downstream task using only real-world contact F/T and pose data extracted from sim-real paired datasets.

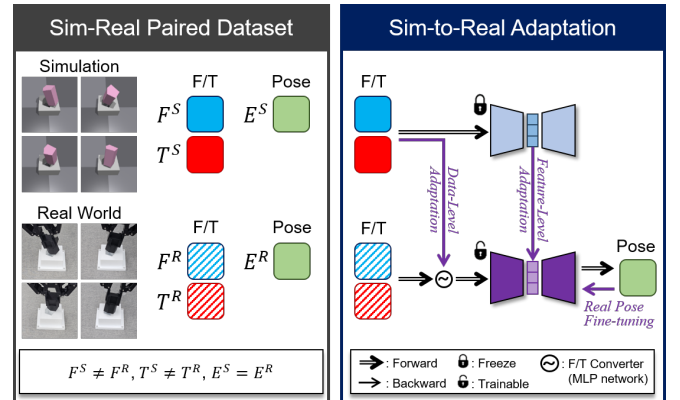


Fig. 5. Conceptual illustrations of the sim-real paired dataset and sim-to-real adaptation method. The sim-real paired dataset are utilized in the sim-to-real adaptation, represented with the same color and shape.

V. EXPERIMENTS

A. Experimental Environment Setup

The Isaac Gym simulator was used as the simulation environment for evaluating the peg-in-hole task, with position control based on extrinsic pose estimation ensuring consistency in data generation and task execution. For real-world validation of the proposed approach, a 6-DoF UR5e universal robot equipped with a built-in F/T sensor at the wrist was used to capture contact data during the peg-in-hole task execution, as depicted in Fig. 6. The task utilized a 3D-printed peg with an integrated groove, manipulated securely with a Robotiq 2F-85 parallel gripper, to ensure reliable grasping. In this real-world context, compliance control was implemented for the peg-in-hole task following extrinsic pose estimation that utilized forward dynamic compliance control (FDCC) [35]. The robot system operated at a control frequency of 120 Hz and the F/T sensor sampled data at a rate of 500 Hz. Communication between the control architecture and hardware was facilitated through the ROS (robot operating system) framework. To overcome inaccuracies in pose estimation when transitioning from simulation to real-world applications, we utilized a method that decomposes the estimated extrinsic pose into directional (\vec{u}) and magnitude (m) components. This approach allows for determining \vec{u} and m needed to manage minor pose errors. These components are used to guide the robot’s motion adaptively through compliance control, enabling continuous adjustments to the robot’s trajectory based on real-time calculations of travel distance.

B. Simulation Evaluation

Extrinsic Pose Estimation Evaluation: The model, trained on polygons with 4-6 vertices, was tested on both seen and unseen polygons with 7-10 vertices, using MAE for pose error evaluation. The test set consisted of 20 polygons for each n-vertices. For seen polygons, average positional and orientational errors were 1.51 mm and 0.23°, respectively, as listed in Table I. Despite these averages exceeding the 1 mm peg-hole tolerance, successful assembly is achievable within individual trial tolerances, as detailed following section. Compared to seen polygons, unseen polygons showed slightly higher errors at 1.85 mm for position and 0.44° for orientation. These results indicate the model’s effective generalization across different polygonal shapes, maintaining consistent performance for shapes not included in the training.

Extrinsic Pose Estimation Network Feature Analysis: To ensure consistent performance on unseen shapes, the deep learning model must be capable of extracting features analogous to those from seen shapes. To verify this ability, the fusion features in the extrinsic pose estimation model were extracted from the test dataset and visualized using t-SNE as shown in Fig. 7(a). The t-SNE distribution was mapped in red-green-blue (RGB) according to the orientation XYZ of the peg in the upper row and colored in red-green (RG) according to the position XY in the lower row. The

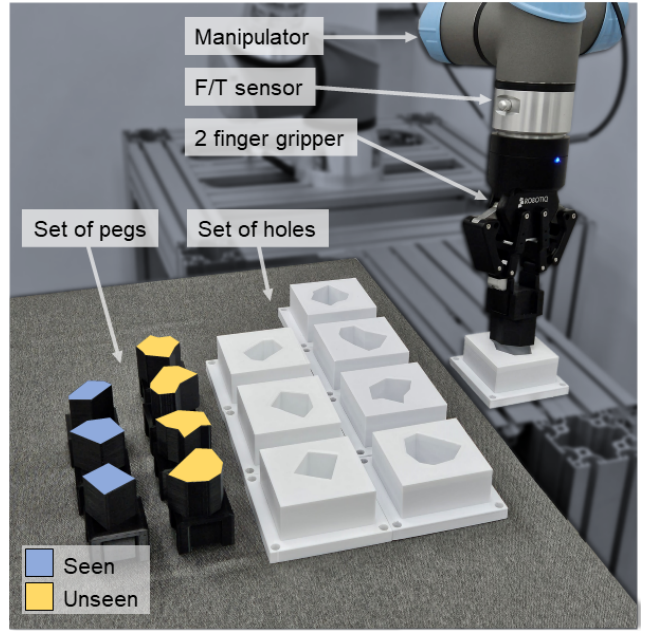


Fig. 6. Setup of the real-world peg-in-hole experiment featuring a UR5e robot with an integrated F/T sensor and a 3D-printed peg-hole sets.

TABLE I
EXTRINSIC POSE ESTIMATION AND PEG-IN-HOLE RESULTS, FEATURING MAE METRICS: POSITION MAE (PMAE) AND ORIENTATION MAE (OMAE), SUCCESS RATE (SR) AND AVERAGE NUMBER OF TRIALS (AT).

	Vertex	PMAE [mm]	OMAE [°]	SR [%]	AT
Seen Polygons	4	1.63	0.22	96.03	2.31
	5	1.51	0.23	97.50	2.36
	6	1.41	0.24	98.48	2.16
	Avg	1.51	0.23	97.33	2.27
Unseen Polygons	7	1.85	0.44	96.73	2.47
	8	1.82	0.43	95.90	2.30
	9	1.82	0.45	95.98	2.39
	10	1.91	0.47	96.63	2.46
	Avg	1.85	0.44	96.30	2.40

features of the seen and unseen polygons were expressed together and drawn as circles and X markers, respectively.

In the model’s feature analysis, orientation distributions, represented in RGB in Fig. 7(a)’s upper row, display a globally continuous gradation, indicating that similar orientations yield comparable feature values. Contrarily, position distributions, shown in the lower row, present a more discrete coloration, yet closer inspection reveals a local continuity in the gradient. This suggests that orientation predominantly influences the initial phase of position estimation, with subsequent feature extraction being locally modulated based on position. In addition, the distribution of the circle and X markers in the enlarged area in Fig. 7(a) indicates the feature distributions of the seen and unseen polygons. The two markers have similar colors in adjacent areas and are evenly mixed without distinction between markers. This demonstrates that the feature extraction capability was robust to shapes, even those not seen during the training phase.

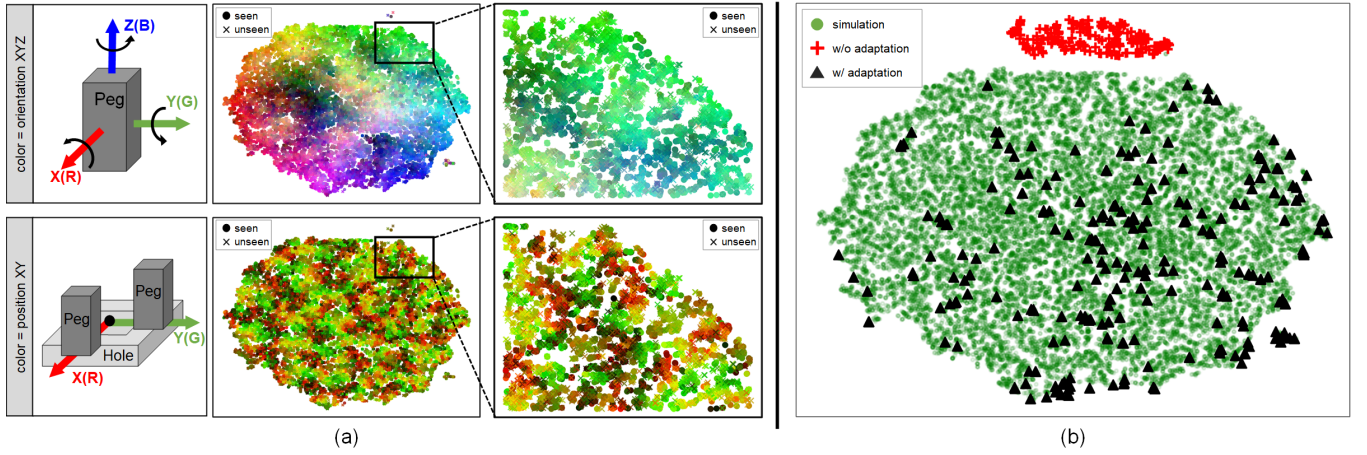


Fig. 7. t-SNE visualization of feature. (a) t-SNE visualization of simulation data features. Circles and X symbols indicate features of seen and unseen polygons, respectively. The upper images are colored with min-max normalized orientations. Euler X, Y, and Z are mapped to red, green, and blue, respectively. The red and green colors in the lower images are mapped to positions X and Y of the peg, respectively. All t-SNE distributions in the images are the same but colored differently. (b) Visualization of simulation features, where the green circle, red cross, and black triangle represent the sim feature, without sim-to-real, and with sim-to-real, respectively. The t-SNE distribution is the same as the distribution in (a).

TABLE II

EXAMPLE OF PEG-IN-HOLE EVALUATION IN SIMULATION, SHOWING MAXIMUM ABSOLUTE POSITION ERROR (MA-P) AND MAXIMUM ABSOLUTE ORIENTATION ERROR (MA-O). THE BOTTOM ROW VISUALIZES THE PEG INSERTION AT THE ESTIMATED POSES.

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5
MA-P [mm]	1.46	1.67	0.91	-	-
MA-O [°]	1.63	0.47	0.49	-	-
S/F	Fail	Fail	Success	-	-

Result of Trials 1 (Fail)	Result of Trial 3 (Success)

Peg-in-hole Evaluation in Simulation: Initial misalignments were randomly generated in the simulated peg-in-hole evaluation, within the same variation range as used during data generation, initiating a multi-point contact strategy for pose estimation and subsequent peg manipulation according to the inferred poses. Each of the 200 test scenarios allowed for a maximum of 5 trials. Successful peg insertion within these trials was recorded as a success, with performance evaluated based on the overall success rate. The evaluation resulted in a 97.3% success rate for seen polygons and 96.3% for unseen polygons, demonstrating robust performance and shape generalization, as presented in Table I. Moreover, the system completed tasks in an average of 2.27 and 2.40 trials for seen and unseen polygons, respectively, showcasing consistent performance across both familiar and novel polygon configurations.

Considering a tolerance of 1 mm, the relationship between the performances of pose estimation and peg-in-hole is shown in Table II. The table illustrates the maximum values of absolute errors for position XY and orientation (MA-P, -O) and peg-in-hole results for each trial when evaluating a 5-vertices shape. The initial two trials failed as the pose estimation error exceeded the tolerance. However, in the third trial, the error was reduced to within the tolerance, indicating the success of the peg-in-hole task. As the trials progressed, with a decrease in pose error, the probability of successful insertion increased. Consequently, the proposed framework demonstrated elevated peg-in-hole success rates based on the average trials of 2.27 and 2.40 for seen and unseen polygons.

C. Real-world Adaptation

Sim-Real Paired Dataset: In the real-world setting, polygons from the simulation dataset were 3D printed and utilized to generate a paired sim-real dataset. Similar to the extrinsic pose estimation model training set, only polygons with 4 to 6 vertices were used for sim-to-real adaptation. Misalignment data for peg-hole interactions in 40 random poses for each 4 to 6 vertices polygon were gathered in both simulation and real-world scenarios. Additionally, data from 15 poses, involving both seen and unseen polygons, were collected

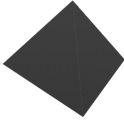






TABLE III

POSE ESTIMATION ERRORS OF SIM-TO-REAL ADAPTATION METHODS. BOLD AND UNDERLINE INDICATE THE FIRST AND SECOND BEST.

Method	Seen Polygons		Unseen Polygons	
	PMAE [mm]	OMAE [°]	PMAE [mm]	OMAE [°]
w/o Adaptation	4.98	6.89	5.00	9.14
DLA	2.76	1.16	3.39	1.93
FLA	4.94	2.12	4.98	2.19
RPF	<u>1.67</u>	0.57	4.06	1.43
DLA + FLA	2.56	1.07	<u>3.19</u>	1.85
Ours	1.21	<u>0.68</u>	2.48	<u>1.64</u>

TABLE IV

REAL-WORLD PEG-IN-HOLE EVALUATION RESULTS ON SEEN (4 TO 6 VERTICES) AND UNSEEN (7 TO 10 VERTICES) POLYGONS.

	Success rate of seen polygons [%]			Success rate of s unseen polygons [%]				Average Trials
								
Method	4-vertices	5-vertices	6-vertices	7-vertices	8-vertices	9-vertices	10-vertices	
Spiral Search [3]	15.0 (3/20)	20.0 (4/20)	20.0 (4/20)	25.0 (5/20)	20.0 (4/20)	20.0 (4/20)	15.0 (3/20)	-
PolyFit (w/o AD)	30.0 (6/20)	40.0 (8/20)	45.0 (9/20)	35.0 (7/20)	30.0 (6/20)	20.0 (4/20)	30.0 (6/20)	3.11
PolyFit (w/ AD)	100.0 (20/20)	100.0 (20/20)	90.0 (18/20)	95.0 (19/20)	95.0 (19/20)	85.0 (17/20)	90.0 (18/20)	2.49

for evaluating pose estimation in real-world scenarios. The process of collecting real-world data for each shape took approximately 10 min.

Extrinsic Pose Estimation in Real World: For qualitatively verifying DLA, real-world contact F/T was converted to their corresponding simulation counterparts using an MLP-based network. Torque data examples from the real-world, simulation, and converted with DLA, are illustrated in Fig. 8 with red, green, and blue lines. In both seen and unseen polygons, the converted torque data exhibited a similar pattern to that observed in the simulation. In FLA, three types of features are extracted and visualized with t-SNE in Fig. 7(b): the features of simulation data are shown as green circles; the features of real-world data, extracted from the model without adaptation, are shown as red crosses; and the real-world features extracted with sim-to-real adaptation through knowledge distillation are shown as black triangles. The red cross indicates a separate distribution from the green circle and black triangle. However, the black triangles shifted the distribution among the green circles, suggesting that real-world features aligned with simulation features after knowledge distillation.

The extrinsic pose estimation performance was evaluated using a collected real-world dataset after sim-to-real adaptation with PMAE and OMAE as shown in Table III. Significant pose errors in position and orientation, occurred when applying a pose estimation model to a real-world

data without any adaptation shown as *w/o adaptation*. Each individual adaptation method, when reported separately, exhibited an improved pose estimation performance compared to the *w/o adaptation*. The RPF method demonstrated commendable performance on seen shapes with 1.67mm and 0.57°. However, it showed a lack of generalization on unseen polygons, where the position error significantly increased. When used alone, the FLA methodology did not significantly improve performance in terms of position, but when applied together with the DLA, it demonstrated enhanced performance including unseen polygons.

The proposed comprehensive methodology that incorporated all data achieved the best performance in position estimation and achieved the second-best performance in orientation with a slight margin. This indicates that aligning both the input data and feature extraction capability can prevent performance degradation during sim-to-real adaptation. The results were consistent for both seen and unseen polygons, demonstrating the generalization capability has been achieved in sim-to-real transfer with effectively collected sim-real paired dataset.

Peg-in-hole Evaluation in Real World: The peg-in-hole tasks were evaluated in a real-world environment using the proposed sim-to-real adapted pose estimation model. Each shape was evaluated 20 times using the same protocol as that in the simulation. Polygons of 4 to 6 vertices were classified as seen shapes and utilized for domain adaptation, while those with 7 to 10 vertices were designated as unseen, as shown in Fig. 6. Comparative benchmarks encompassed three methods: a standard spiral-search method [3], PolyFit without adaptation that was solely dependent on linear scaling between simulated and real data, and PolyFit with sim-to-real adaptation. The results for the adapted model demonstrated a substantial enhancement in performance, evidenced by success rates of 96.7% and 91.3% for seen and unseen polygons, respectively, as listed in Table IV. The success rates achieved by PolyFit with adaptation markedly exceeded those obtained without adaptation, 38.3% for seen shapes and 28.8% for unseen shapes. Additionally, these figures represent a substantial improvement over the success rates of 18.3% for seen shapes and 20.0% for unseen shapes achieved by the spiral search method. This notable difference emphasized the capability of the proposed pose estimation

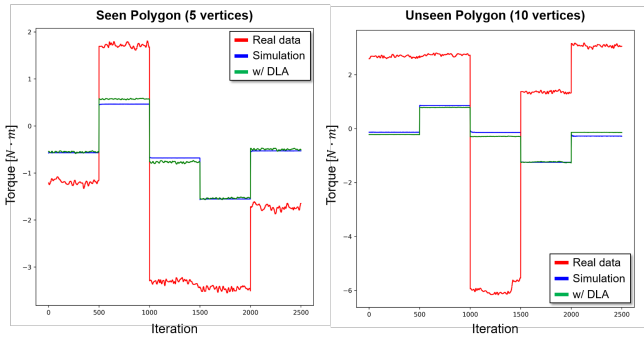


Fig. 8. Examples of data-level adaptation (DLA) on seen and unseen polygons. Red: real data, Blue: simulation data, Green: data distribution after applying the proposed DLA method to align real data with simulation.

model to accommodate unseen polygon shapes and validated the successful transition and adaptability of the simulation-trained model to real-world challenges.

VI. CONCLUSIONS

This study introduces an F/T-based framework capable of executing 5-DoF peg-in-hole assembly, effectively addressing key challenges in robotic assembly tasks. The extrinsic pose estimation model exhibits noteworthy performance on both seen and unseen polygonal shapes, achieved through extensive simulation-based training. The introduction of sim-to-real adaptation, leveraging a sim-real paired dataset, proves instrumental in maintaining high-performance levels with unseen polygons. This underscores the framework's versatility across diverse geometries and operational environments. The findings of this research have the potential to mitigate inefficiencies and safety concerns associated with real-world data acquisition in the future. By utilizing a dynamic simulator for learning and data collection, the framework ensures robust performance across various component geometries and conditions, eliminating the need for extensive real-world data collection. Future work will focus on incorporating a closed-loop framework to enhance robustness and extend the application to more practical assemblies, such as connectors and cables.

ACKNOWLEDGMENTS

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. (23ZR1100, A Study of Hyper-Connected Thinking Internet Technology by autonomous connecting, controlling and evolving ways). And also supported by Artificial intelligence industrial convergence cluster development project funded by the Ministry of Science and ICT(MSIT, Korea)&Gwangju Metropolitan City.

REFERENCES

- [1] H. Bruyninckx, S. Dutre, and J. De Schutter, "Peg-on-hole: a model based solution to peg and hole alignment," in *Proceedings of 1995 IEEE International Conference on Robotics and Automation(ICRA)*, vol. 2. IEEE, 1995, pp. 1919–1924.
- [2] T. Tang, H.-C. Lin, Y. Zhao, W. Chen, and M. Tomizuka, "Autonomous alignment of peg and hole by force/torque measurement for robotic assembly," in *2016 IEEE international conference on automation science and engineering (CASE)*. IEEE, 2016, pp. 162–167.
- [3] S. R. Chhatpar and M. S. Branicky, "Search strategies for peg-in-hole assemblies with position uncertainty," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 3. IEEE, 2001, pp. 1465–1470.
- [4] H. Park, J. Park, D.-H. Lee, J.-H. Park, and J.-H. Bae, "Compliant peg-in-hole assembly using partial spiral force trajectory with tilted peg posture," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4447–4454, 2020.
- [5] L. Xie, H. Yu, Y. Zhao, H. Zhang, Z. Zhou, M. Wang, Y. Wang, and R. Xiong, "Learning to fill the seam by vision: Sub-millimeter peg-in-hole on unseen shapes in real world," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2982–2988.
- [6] J. Ding, C. Wang, and C. Lu, "Transferable force-torque dynamics model for peg-in-hole task," *arXiv preprint arXiv:1912.00260*, 2019.
- [7] 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.
- [8] 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 *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3080–3087.
- [9] M. Hebecker, J. Lambrecht, and M. Schmitz, "Towards real-world force-sensitive robotic assembly through deep reinforcement learning in simulations," in *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2021, pp. 1045–1051.
- [10] D. Ma, S. Dong, and A. Rodriguez, "Extrinsic contact sensing with relative-motion tracking from distributed tactile measurements," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 11 262–11 268.
- [11] G. Schoettler, A. Nair, J. Luo, S. Bahl, J. A. Ojea, E. Solowjow, and S. Levine, "Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5548–5555.
- [12] Y. Shi, C. Yuan, A. Tsitos, L. Cong, H. Hadjar, Z. Chen, and J. Zhang, "A sim-to-real learning based framework for contact-rich assembly by utilizing cyclegan and force control," *IEEE Transactions on Cognitive and Developmental Systems*, 2023.
- [13] O. Spector and D. Di Castro, "Insertionnet-a scalable solution for insertion," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5509–5516, 2021.
- [14] O. Spector, V. Tchuiev, and D. Di Castro, "Insertionnet 2.0: Minimal contact multi-step insertion using multimodal multiview sensory input," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6330–6336.
- [15] A. Nair, B. Zhu, G. Narayanan, E. Solowjow, and S. Levine, "Learning on the job: Self-rewarding offline-to-online finetuning for industrial insertion of novel connectors from vision," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7154–7161.
- [16] C. C. Beltran-Hernandez, D. Petit, I. G. Ramirez-Alpizar, and K. Harada, "Variable compliance control for robotic peg-in-hole assembly: A deep-reinforcement-learning approach," *Applied Sciences*, vol. 10, no. 19, p. 6923, 2020.
- [17] T. Z. Zhao, J. Luo, O. Sushkov, R. Pevceviciute, N. Heess, J. Scholz, S. Schaal, and S. Levine, "Offline meta-reinforcement learning for industrial insertion," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6386–6393.
- [18] 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 *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.
- [19] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3803–3810.
- [20] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox, "Closing the sim-to-real loop: Adapting simulation randomization with real world experience," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8973–8979.
- [21] Y. Du, O. Watkins, T. Darrell, P. Abbeel, and D. Pathak, "Auto-tuned sim-to-real transfer," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1290–1296.
- [22] E. Valassakis, N. Di Palo, and E. Johns, "Coarse-to-fine for sim-to-real: Sub-millimetre precision across wide task spaces," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5989–5996.
- [23] S. Scherzinger, A. Roennau, and R. Dillmann, "Contact skill imitation learning for robot-independent assembly programming," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4309–4316.
- [24] M. Kaspar, J. D. M. Osorio, and J. Bock, "Sim2real transfer for reinforcement learning without dynamics randomization," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4383–4388.
- [25] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.
- [26] M. Macklin, K. Erleben, M. Müller, N. Chentanez, S. Jeschke, and Z. Corse, "Local optimization for robust signed distance field collision," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 3, no. 1, pp. 1–17, 2020.