Fast Skill Transfer Method for Peg-in-Hole Assembly Tasks Under Varied Visual Conditions

Kai Wu[1], *Member, IEEE*, Qi Chen[1], Huan Zhao[2], *Member, IEEE*, Mingfeng Wang[3], *Member, IEEE*

Abstract—**Deep Reinforcement Learning (DRL) has emerged as a transformative approach in robotic assembly, offering unparalleled adaptability and efficiency in automating complex tasks. However, existing DRL methods with weak generalization require retraining of policy when facing new assembly scenarios, which require a significant amount of interaction and may harm the robots or parts. This paper presents a fast skill transfer approach for submillimeter-level assembly tasks. The approach enables rapid adaptation to varying textures and lighting variations, which are commonly encountered in flexible manufacturing environments. The model parameters can be quickly adjusted to facilitate seamless adaptation. Specifically, a concise distance-based encoder model is proposed to extract the latent representation from the low dimensional seam-based image (SBI) and map the extracted feature to the distance space. Then, the fine-tuning strategy is used to align the features of new scenes with those in the source scenes. The transfer strategy necessitates only the retraining of the feature extraction model, obviating the need to retrain the underlying RL policy. Simulation and real-world experiments are conducted to evaluate the proposed method, and the transfer can be finished in a few minutes. The policy trained in the simulation can be transferred to the different real-world assembly scenes with the proposed method with an average success rate of 94.3%, highlighting its potential for practical applications.**

Index Terms*—*Assembly, Reinforcement Learning, Assembly skill learning, Skill transfer learning.

I. INTRODUCTION

The peg-in-hole insertion, as a typical assembly task in manufacturing, has been widely studied by applying DRL in recent years[1]. Although many studies focus on the force and torque information due to the inherently contact-rich nature of the peg-in-hole task [2], [3], there is a growing tendency to utilize visual information for its advantages in adapting to a wider initial error range and improving assembly speed [4]. Compared with traditional methods[5], DRL approaches are less dependent on precise system calibration and specific visual setups. However, in personalized products and flexible manufacturing, products often have similar shapes but different textures and various light sources for illumination. This poses challenges for sample efficiency and generalization in DRL-based assembly [6], requiring extensive retraining in the new scenes. Efforts have been made to enhance training efficiency, but the complex network structures and high-dimensional state observation spaces inherent to DRL often impede training speed. Additionally, high-performance computers for learning may not be available in practical applications, making it necessary to design an efficient and concise transfer method. Therefore, how to adapt robots' existing skills to new tasks expeditiously is significance [7].

In response to the demands of flexible manufacturing, this article focuses on how to adjust assembly skill parameters in a short period to adapt to different manufacturing scenarios, rather than relying on a highly generalization model to cover all scenarios. Considering that assembly involves hole-search and insertion stages, and visual variations primarily affect the former. This work proposes a DRL-based fast skill transfer method to learn peg-in-hole assembly skills in the search stage, with a focus on the transfer strategy for assembly tasks under different part textures and lighting conditions. Contributions of this paper are summarized as follows:

1) A visual domain adaption-based fast skill transfer method is proposed for assembly tasks under different visual conditions, by which the skill can be directly transferred with only a few minutes of contactless data collection without the need for retraining the RL policy.
2) A novel visual state representation for assembly tasks called seam-based image (SBI) is proposed, which is low dimensional and focuses on features closely related to the assembly process. A concise feature extract model is proposed to extract the distance information from the SBI.
3) A fully automatic two-stage transfer strategy is proposed to fine-tuning the model, mapping the extracted features from different scenes to the same physical distance space, thereby enhancing both efficiency and success rate.
4) Both simulation and real-world experiments of submillimeter-level insertion tasks are evaluated. By using the proposed skill transfer method, the RL policy trained in simulation can be successfully transferred to different real-world assembly scenes in a few minutes.

Manuscript received April 7, 2025; revised July 8, 2025; accepted September 22, 2025. This paper was recommended for publication by Editor Ashis Banerjee upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Fundamental Research Funds for the Central Universities (Grant No. 2024ZYGXZR107), the GJYC program of Guangzhou (Grant No. 2024D03J0005), National Key R&D Program of China (Grant No. 2024YFB4709200). *(Corresponding author: Kai Wu).*

Kai Wu, Qi Chen are with the Shien-Ming Wu School of Intelligent Engineering, South China University of Technology, Guangzhou 510640, China (e-mail: whphwk@scut.edu.cn; chenqi3532@163.com).

Huan Zhao is with Huazhong University of Science and Technology, State Key Laboratory of Digital Manufacturing Equipment and Technology, Wuhan, 430074,China. (e-mail: huanzhao@hust.edu.cn).

Mingfeng Wang is with Brunel University London, Department of Mechanical and Aerospace Engineering, London, UB83PH, UK. (e-mail: mingfeng.wang@brunel.ac.uk).

Digital Object Identifier (DOI): see top of this page.

Citation information: DOI: 10.1109/LRA.2025.XXXXXXX,

II. Related Work

In early research, DRL assembly methods were directly trained in real-world. Theses method were time-consuming and dangerous [8]. Recently, techniques were used to reduce the training time. This section mainly introduces the related works of efficient learning skills for RL-based assembly.

A. Accelerate Policy Learning with Task Information

Several techniques have been investigated to accelerate the DRL learning process. Firstly, the residual RL utilizes prior trajectories to reduce random exploration space. Johannine et al. [9] proposed a residual RL method in which the trained policy is fine-tuned on the prior trajectories from the PI controller, thus reducing the exploration scope and improving the learning efficiency. Secondly, the model accelerated technology establishes environmental dynamics models to generate virtual data for training. Zhao et al. [10] utilized a Gaussian process for dynamics modeling, and expanded the replay buffer with predicted information, thereby reducing the interaction frequency. Lastly, low-dimensional observation space reduces the complexity of input features. Zhao et al. [11] proposed a low-dimensional visual state representation based on pixel-wise linear and rectangular to enhance the training efficiency. G. Schoettler et al. [12] and Finn [13] reduced the dimensions by resizing images and autoencoders, which are also effective. Although studies have employed various techniques to accelerate training, policy was still learned from scratch and only utilized information from current tasks.

B. Learning from Prior Policies

Some studies extract knowledge from similar tasks rather than solely relying on current tasks. Such as filled the replay buffer with prior knowledge or used a pre-trained policy. Jin at el. [7] used Muti-kernel Maximum Mean Discrepancy to select the knowledge from the source tasks that are similar to the new tasks, and added them into the replay buffer to speed up the policy training. Zang et al. [14] proposed a geometric-feature based pose transition model to pretrain the actor and critic network. Yasutomi et al. [15] designed a hole map and utilized it for offline training and minimal data extraction from the environment. In these works, although prior knowledge can accelerate the learning process, a small amount of interaction with the environment is still required to complete the final training of the policy.

C. Techniques for Direct Policy Transfer

Some research has also explored how to directly transfer RL policies to the new scenes, such as domain randomization (DR), image masking, and domain adaptation (DA). DR extensively randomizes parameters (e.g., lighting, texture [16], [17]) in the training process, thus improving the generalization performance. Image masking employs masks to unify textures and backgrounds, ensuring that images' input features remain consistent across various scenes. Ahn et al. [18] utilize mask images to transfer the image-based policy directly from the virtual to the real world. The image parts of peg, hole, and background are segmented and filled with different gray value. Xie et al. [19] designed pose estimators based on the mask images, and only the segmentation module needs to be retrained during transfer. DA aims to map the feature from different sources to the same space. For example, Shi et al. [20] utilize domain adaption to bridge the sim2real gap of images, where the Cycle-GAN is used to transfer the real-world images to the simulation style. Then, a Variational Autoencoder (VAE) was used to extract features for input into the RL policy. However, the methods mentioned above rely on complex network architectures (e.g., segmentation networks and GAN), which seriously impact the training speed. Meanwhile, these networks are partially affected by the environmental background. Uninterpretable features from inaccurate reconstruction outputs can decrease assembly success rates, especially in tasks with small clearances.

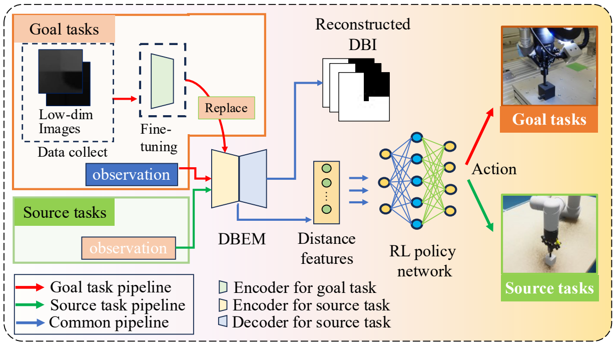

Fig. 1. Overview of the fast skill transfer framework.

Inspired by previous works, we present a technique for fast transfer of RL assembly skills. In conclusion, our research diverges notably from the previous works. Firstly, most studies use original images as input, relying on complex networks or attention mechanisms to filter out background information. The proposed SBI is extracted around the peg and only focuses on the task-related seam information with a small amount of background. This design enables even a simple network to process the information effectively. Secondly, most works require fine-tuning RL policies when using transfer methods, inevitably resulting in random or even dangerous contact actions to the environment. The proposed method maps image observations into physics - informed latent space, and achieves feature alignment for different tasks in an ingenious way, thereby ensuring input consistency of the RL strategy and avoiding retraining the policy. It is contactless during the transfer process. These characteristics make our method highly efficient and advantageous for rapid deployment in flexible manufacturing environments.

III. Methodology

A fast skill transfer framework is constructed for assembly tasks under different lighting and texture conditions (Fig. 1). The proposed method mainly includes two parts: 1) a feature extract model that introduces a novel low-dimensional visual representation and a concise distance-based encoder model (DBEM); 2) a two-stage transfer method, including the data collection and encoder-only fine-turning stage. The overall system will be introduced firstly, then the proposed method.

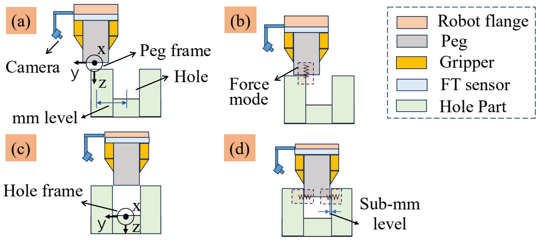

Fig. 2. Peg in hole assembly process. (a) Start from initial position with millimeter level distance. (b) Contact with the part surface, search the hole in x, y direction and apply a constant force in z direction. (c) Find the hole. (d) Insert the peg into the sub-millimeter level clearance hole.

A. Overview of the Peg-In-Hole Assembly Process

As shown in Fig. 2, the robot initiates at a distance of millimeters from the hole. Assuming the direction is pre-aligned, the control policy is designed to learn a refined positioning strategy from the images captured by the in-hand camera. It aims to search the hole and correct positioning errors through multi-step adjustments. The process can be considered as Markov decision process, and DRL algorithm is used to learn the assembly policy.

During the training phase, the RL-based assembly policy network is trained with the follows reward function:

$$r = r_1 + r_2 + r_3 - r_4 \tag{1}$$

$$r_1 = \begin{cases} -\alpha * \Delta d, & if \quad \Delta d > 0 \\ -\alpha * \Delta d, & if \quad \Delta d < 0 \ \& \ |\Delta d| < step_{\min} \\ 0, & else \end{cases} \tag{2}$$

$$r_2 = \begin{cases} -1, & if \quad d > d_{\max} \\ 0.01, & if \quad d < d_{\min} \end{cases} \tag{3}$$

where reward r_1 is used to encourage the robot to approach the hole while maintaining sufficiently small action to enable fine-grained exploration. $step_{min}$ is used to discourage large action, and set to 0.5mm. Δd is the difference in distance between step t and step t-1, i.e., Δd = d_t-d_{t-1}, where d_t, d_{t-1} are the relative distance(mm) between the peg and hole (calculate using Δx, Δy and Δz between the hole and peg frames) at steps t and t-1, respectively. α is a scale factor to limit the range of reward, set to 0.01. r_2 is defined to encourage the robot to stay in the position of the hole without moving away, r_3 is a sparse reward that only gets a positive value when the peg inserts the hole, it's set to 1. and r_4 is the step penalty that propels the robot to finish the task as quickly as possible, it is a fixed value set to 0.005.

At each timestep, the state representation S_t is calculated by the feature extract network and input to the RL policy network, then the policy network will output the action $a_t = [\Delta x, \Delta y, \Delta z]$ in TCP (Tool Center Point) frame to the robot control system. The hybrid force-position control is used to control the robot, which configures the system in position mode along X and Y directions and force control along the Z direction. For fast response, force signal is used to judge whether alignment is completed. Once the force in Z direction is smaller than the contact force threshold, it means the alignment is completed, X and Y directions will switch to compliance force mode, allowing for deeper insertion.

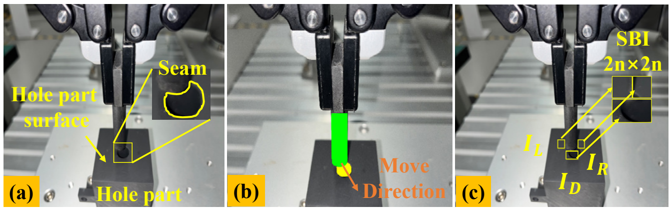

Fig. 3. Using seam features to reduce the image dimension. (a) Seam between peg and hole. (b) The assembly direction is the direction that fills the hole. (c) Composition of the SBI.

B. Feature Extract Model

In this section, an efficient model for feature extraction and fast transfer is proposed. Firstly, a low-dimensional assembly image is designed to exclude irrelevant background information and serve as input for the model. Then, the distance-based encoder model is designed to extract the task-related physics feature from the input image.

1) Seam-based Images: Previous studies have shown that assembly skills can be acquired even without direct visual observation of the target hole, provided that the image content is sufficiently distinctive [11], [21]. Nevertheless, it is critical to acknowledge that these distinctive elements may be task-irrelevant features embedded in the background. Background variations are often unavoidable in practical applications, and variations in background features can significantly undermine the robustness and reliability of the learned skills. In contrast, the pixels surrounding the hole are relatively more stable and are directly related to the task.

As shown in Fig. 3(a), a seam area is formed due to the different darkness reflections between the part surface and hole inner surface under lightening. Inspired by the human insertion of a peg into the hole, it always follows the rule of moving the peg to reduce the seam area, as illustrated in Fig. 3(b). Based on this rule, it's possible to reduce dimensionality on the source images and remove irrelevant backgrounds, and only reserve the seam pixels around the peg. These pixels provide sufficient information to ascertain the hole's direction. Therefore, the seam-based image (SBI) I_S is proposed and defined as follows:

$$I_S = \text{Concatenate}(I_L, I_R, I_D) \tag{4}$$

where I_L and I_R represent the image patches (size of n×n) on the left and right sides of the peg tip, respectively. I_D represents the image patch beneath the peg tip, with the size of 2n×n. By applying concatenate operation on the three patches, the seam-based image can be generated with the size of 2n×2n. Here, coordinates of I_L, I_R and I_D are determined through an automated localization algorithm (introduced in sect IV-C), which ensures a certain degree of repeatability.

Theoretically, using I_S as input to the network is efficient to the training process and reduces the burden of transfer, because the size of the image is reduced to a small patch, meaning that the parameters of the feature extract network can be reduced accordingly. Besides, I_S only contains the pixel around the hole, so that the network does not design additional attention modules to exclude background interference.

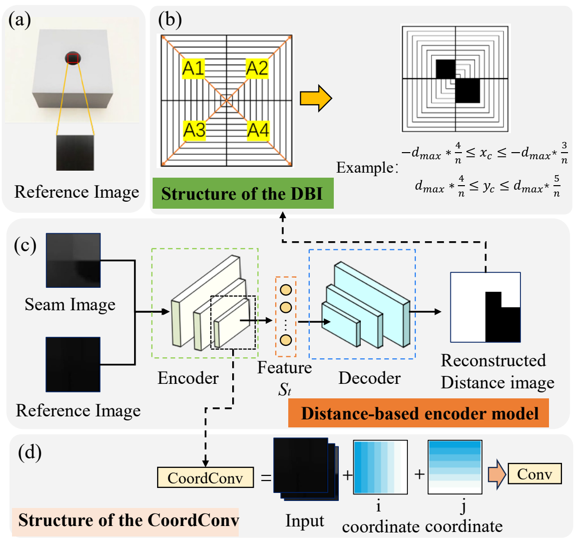

Fig. 4. Distance-based encoder model. (a) Sampling position of the reference image. (b) Structure of the DBI. (c) Distance-based encoder model. (d) Structure of the CoordConv module.

2) Distance-based Encoder Model: The encoder-decoder architecture has been widely used to extract meaningful features[13], [15], [19]. Here, a distance-based encoder model (DBEM) is proposed based on the encoder-decoder architecture, which is used to extract the distance-relevant latent representation from SBI. The latent representation is then used as the state representation input to the RL policy. As shown in Fig. 4(c), SBI and a reference image are stacked and input to the DBEM, where the reference image is generated from the grayscale values of the hole (Fig. 4(a)), it is used to guide the network to focus on areas with similar grayscale. The output of the decoder is a reconstructed image, consisting only of black and white pixels.

The design of the reconstructed object is crucial because it determines the meaning of the latent representation extracted by the encoder. Since our goal is to transfer the skill in different textures and lighting environments, it aims to extract features that are texture and lighting-independent, and ideally universal across different tasks. Therefore, the latent representation is designed to be solely related to the distance between the peg and hole, and the reconstructed object is set to a distance-based image (DBI).

As shown in Fig. 4(b), DBI is designed to indicate distance information between the peg and hole. And it is divided into four areas, namely A1, A2, A3, and A4. A1 and A2 are used to represent the current distance between the hole and the peg in the X direction, where A1 will be filled with black when in the negative direction and A2 will be filled when in the positive direction. A3 and A4 represent the distance in the Y direction in the same way. Each area is then divided into n grids. The farther away, the larger the black area.

As for the network, CoordConv[22] is used to replace the Convolutional module, as shown in Fig. 4(d). Two new channels (i, j coordinate) are added to the input data, and it excels at reconstructing such images with white and black blocks. The encoder is composed of 3 CoordConv layers, and the decoder is composed of 4 Coor-Deconvolutional layers.

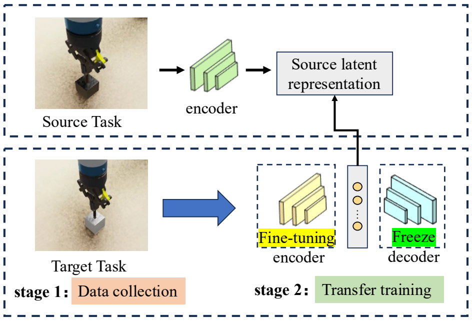

Fig. 5. Two-stage transfer strategy, mapping the target latent representation to the source latent representation.

Finally, mean square error (MSE) is used for the loss function to train the network:

$$L = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{w \times h}\sum_{j=1}^{w}\sum_{k=1}^{h}\| f_i(j,k) - X_i(j,k) \|^2\right) \tag{5}$$

where f is the distance-based image generated by the distance, X is the reconstructed image, w and h are the width and height of the image, n is the number of samples in each epoch.

C. Two-Stage Transfer Strategy

When facing new scenes, the texture of the assembly part and the environment lighting may be different. Denote the source task domain as $D_S \equiv (S_S, A_S, P_S, R_S)$ and target task domain as $D_T \equiv (S_T, A_T, P_T, R_T)$, and assume that D_S and D_T share the action space ($A_S \approx A_T$), transitions ($P_S \approx P_T$), and reward functions ($R_S \approx R_T$), but with different state representations ($S_S \neq S_T$). In order to directly use the source RL policy in the target domain, what needs to be done is to convert S_T into S_S (as in (6)), then the output action of the RL policy can be similar (as in (7)):

$$f(S_T) \approx S_S \tag{6}$$

$$Action = \pi_\theta(S_S) \approx \pi_\theta(f(S_T)) \tag{7}$$

A two-stage transfer strategy is proposed, including data collection in the new environment and fine-tuning training, as shown in Fig. 5.

1) Data collection stage: it is necessary to ensure that the movement range covers the actual workspace as much as possible, which can be achieved through offline programming. During the movement, images are recorded, and SBI and DBI can be generated. At the same time, take an image of the interior of the hole directly above it as a reference image.

2) Transfer training stage: The encoder only fine-tuning strategy is utilized in this stage. The purpose is to fine-tune the feature extract model parameters, and a similar output can be obtained by input images from different environments. As mentioned in above, the reconstructed image of DBEM is generated from distance information and not related to visual changes such as texture and lighting. Thus, the different tasks have the same reconstructed image as long as they are located at the same position. However, the neural networks have a certain degree of randomness in the training process. If the fine-tuned operation is applied to the encoder and decoder at the same time, it's possible to get the same reconstruction result but with a different latent representation. Therefore, the decoder parameters are frozen, and only the encoder will perform parameter fine-tuning during the training process,

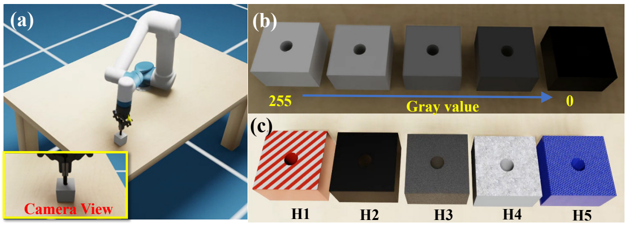

Fig. 6. (a) Virtual assembly environment in Isaac sim. (b) Training environments. (c) Testing environments.

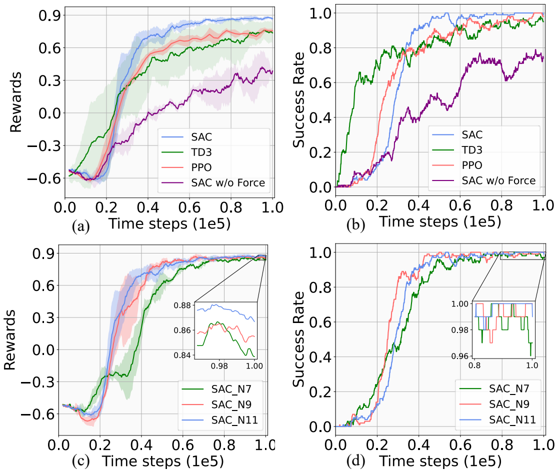

Fig. 7. Learning curves of different RL algorithms and SBI size. (a) and (b) are trained with SAC, SAC w/o Force, PPO and TD3. (c) and (d) are trained with SAC, and SBI size set to 7, 9 and 11.

making the different image input mapping to the same distance space as the source task.

IV. Experiment

To verify the performance of the proposed transfer method, circle peg insertion tasks were conducted in both simulation and real-world environments. A computer with an RTX 4090 GPU and an Intel i9 CPU was used to fine-tuning the model. A laptop with an RTX 1060 GPU and Intel i7 CPU was used to control the robot in the real-world experiment. In this section, the details of model pre-training in source tasks were first introduced, and then the transfer experiments were conducted in simulation and real-world scenarios separately.

A. Model Training in Source Tasks

Initially, the RL policy and feature extraction model were trained on the source task in simulation. The training process and its details were described as follows:

1) Pre-Training of the DBEM: A robotic assembly scene was established in Nvidia Issac Sim, as shown in Fig. 6(a). The DR technology was used to randomly set the initial position and the gray value of the hole part from 0 to 255 (see Fig. 6(b)) at each start of the epoch. Then, data was collected and the DBEM was trained and saved as the pre-train DBEM, which was subsequently used to train the RL policy in the source scene and fine-tuning in the new scenes.

2) Training of RL Policy: The environment for RL policy training was also configured with DR. The pre-train DBEM was used to extract features for the policy network. Then the RL algorithm was utilized to update the weight of the policy network. The maximum training timestep was set to 10^5. After convergence, the policy network was saved and used in the subsequent experiments.

3) Configuration of Training: The experiment was first conducted to evaluate whether the seam feature was sufficient for the robot to acquire assembly skills. Both state-of-the-art on-policy and off-policy DRL algorithms were used for validation. As shown in Fig. 7. (a) and (b), SAC, TD3, and PPO can all successfully learn the skill, which demonstrates the effectiveness of the seam feature and shows its ability to freely switch its base actor-critic training approaches. Besides, SAC outperformed the others and was thus selected to be used in the subsequent experiments. Additionally, it shows the training environment without force control. The peg was easily stuck on the surface or inside the hole, resulting in a decrease in the success rate. Fig. 7. (c) and (d) showed the performance of different sizes of SBI. It showed that n=11 has a slightly higher reward and success rate than the other two at the end of training. Thus, the SBI size in the simulation and real-world experiments was set to 22×22 (2n×2n).

B. Skill Transfer in Simulation

In this section, the transfer experiments in the simulation were conducted. The configuration was set as same as the training environments (peg and hole clearance with 0.9mm). However, the render lighting and the hole parts texture were different, as shown in Fig. 6(c).

1) Method Comparison: Comparative experiments were designed to evaluate the performance of our method with other methods. Cycle-GAN was chosen as a baseline. Because it is widely used in visuomotor skill transfer and has been applied in manipulation tasks like insertion[20], pouring[23], and grasp[24]. Variational auto-encoder (VAE) is a commonly used and powerful feature extraction model, and was selected to be compared. Besides, to evaluate whether DBI can be replaced as classification task, MLP was used to replace the decoder. All the methods are as follows:

a) **Y. Shi et al.** [20]: Cycle-GAN based method was used to transfer the assembly images in new scenes into source scenes. Then, VAE is used to extract features and input them into the RL policy network.

b) **DR:** the policy network trained in the DR environment was directly used to test in the transfer scenes.

c) **VAE-based approach:** Replaced the DBEM with VAE, used the proposed method to transfer the skill.

d) **VAE-FBED:** Replaced the DBEM with VAE. and fine-tuning both the encoder and decoder.

e) **DBEM-FBED:** Use DBEM and fine-tuning both the encoder and decoder.

f) **MLP-4：** Instead of using DBI as the label, an MLP classified the encoder features into four categories defined by the sign combinations of relative displacement along the x and y axes.

g) **MLP-**484. MLP was used to classify features into 484 categories, following the 22×22 DBI configuration.

h) **Our method.**

2) Result and Analysis: The test results in terms of training time (TT) and success rate (SR) are listed in Table I. The

Citation information: DOI: 10.1109/LRA.2025.XXXXXXX,

TABLE I
EXPERIMENTAL RESULTS WITH DIFFERENT METHODS IN SIMULATION

	Y. Shi et al [20]		DR		VAE		VAE-FBED	
Part	TT	SR	TT	SR	TT	SR	TT	SR
H1	871s	97%	\	7%	1896s	6%	2040s	20%
H2	4352s	33%	\	99%	1870s	5%	2041s	10%
H3	1450s	98%	\	51%	1888s	2%	2038s	3%
H4	1305s	99%	\	97%	1890s	18%	2041s	3%
H5	1305s	99%	\	52%	1885s	8%	2033s	5%
	DBEM-FBED		MLP-4		MLP-484		**Ours**	
Part	TT	SR	TT	SR	TT	SR	TT	SR
H1	1202s	23%	1212s	51%	1224s	5%	**42s**	**99%**
H2	1201s	22%	1226s	65%	1224s	15%	**21s**	**100%**
H3	1199s	41%	1222s	62%	1219s	7%	**107s**	**98%**
H4	1080s	97%	1213s	36%	1205s	18%	**41s**	**99%**
H5	1204s	36%	1209s	44%	1231s	8%	**21s**	**99%**

results were counted every 15 episodes, and the TT was recorded when the SR was greater than 95% (evaluate 100 times) or training above the max episodes (Specifically, max episodes for Cycle-GAN was set to 150, and 2000 for others.).

The results showed that DR has the advantage of zero-shot transfer and is adaptable to unseen hole parts with similar textures (e.g. H2 and H4). But some slight differences can lead to a decrease in SR (e.g. H3). And it's not adapted to larger differences (e.g. H1).

For DBEM-FBED, the success rate remained low even after extended training, despite the loss converging to a similar level as the proposed method. This indicates that without freezing the decoder, intermediate feature consistency with source tasks cannot be ensured. In both variants where DBEM was replaced by VAE, the SR was also low, highlighting VAE's limitation in preserving feature alignment with source representations. These results emphasize the necessity of integrating DBEM with the proposed transfer method.

For MLP-4, the 4-classification task reduces the granularity of distance features, achieving only a 70% success rate even in the source task. During transfer, the coarse features are hard to maintain consistency, leading to degraded performance. MLP-484 struggles with the high-dimensional classification task due to its simple architecture, resulting in low classification accuracy and similarly poor overall success rate.

Compared with the Cycle-GAN, our method achieved a high success rate in all cases. It is worth noting that Cycle-GAN also achieved a high success rate, except for H2. Because the color of the holes in H2 is very close to the peg, it is difficult for Cycle-GAN to accurately generate the position of the holes during reconstruction in such a situation.

With the low dimensional input and lightweight network architecture, our method only took about 0.7s with about 6000 images/episodes. The result showed that the fastest transfer can be completed in 21 seconds in the fastest cases (H2, H5). However, more time was needed for the Cycle-GAN due to the complex GAN architecture, which took about 30s/episode. The fastest transfer is finished in about 15 minutes (H1), which was trained with about 30 episodes.

In addition, the input of the Cycle-GAN was resized from the original images to 64×64, which will lose pixel accuracy and make the image blurry. If the original image is used as input, the training time will become longer. Instead, our method reduces the images through cropping operations, preserving the precision information in the images.

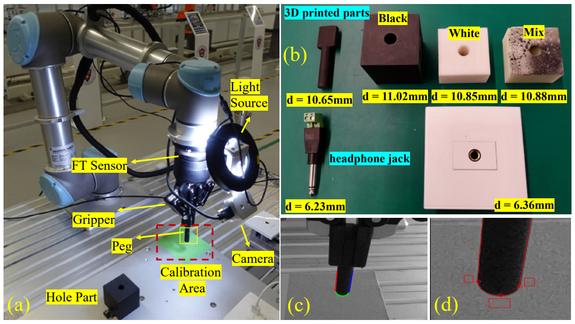

Fig. 8. (a) Hardware and Calibration area with green background. (b) Type and size of peg and hole. (c) Automatically detect and segment the peg edge. (d) extract the SBI based on the edge detect result.

C. Skill Transfer in Real-World

1) Experiment Setup: Three different 3D printed hole parts and a headphone jack (Hp jack) were tested, as shown in Fig.8 (b). A camera was mounted on the end-flange of UR5 robot without carefully calibration. The Ft Sensor (robotiq FT300s) was used to measure the contact force, and a 2-finger gripper was used to grip the peg (Fig. 8(a)). The robot control rate was 125Hz, RL policy output rate was 20Hz, and the max servo speed was limited to 5mm/s.

For each test, the initial x and y of the peg and hole ranged from -5mm to 5mm, and a 40-times test for each task was conducted. It is considered successful when the insert depth reaches the requirement (1cm of 3D-printed part and 2.3cm of headphone jack) and fails when the time exceeds 25s.

A green-background calibration area was designed to automatically extract the peg tip before each task using an edge detection algorithm. The algorithm identified the peg's edge and segmented it into two straight lines and one arc based on curvature changes (Fig. 8(c)). The intersection points between the straight lines and the arc, along with the arc's lowest point, were used as sampling coordinates to generate the corresponding SBI area automatically (Fig. 8(d)). Although a green background and geometric rules were used to extract coordinates. However, the extraction algorithm can be replaced as needed, as long as repeatability can be ensured.

A pre-programmed trajectory was used to capture comprehensive images of new parts. The robot first captured a pixel patch above the hole and resized it to match the SBI image using interpolation, creating a reference image. It then moved to the insertion point and followed spiral paths at five height levels to collect data. The process took about 5 minutes and produced around 6,000 images for SBI generation.

2) Result and Discussion (different texture): The average success rate (ASR), average execute time (AET), maximum initial distance (MAXID), maximum execute time (MAXET), minimum execute time (MINET), and training time (TT) were recorded.

As shown in Table II, all the parts achieved a success rate higher than 97.5% with our method, and the average finish time was about 5s. This indicates that the proposed method is equally effective. Even when applied on sim2real transfer, it

TABLE II
SUCCESS RATE AND FINISH TIME IN REAL-WORLD EXPERIMENT

Ours						
Part	ASR	AET	MAX ID	MAX ET	MINET	TT
Black	**97.5%**	**5.10 s**	6.7 mm	9.8s	4.0s	**31s**
White	97.5%	5.26 s	6.3 mm	9.0s	3.3s	**166s**
Mix	**100%**	**5.31s**	6.2 mm	19.9s	3.1s	**158s**
Hp jack	**97.5%**	**4.92s**	6.9mm	10.7s	3.2s	**62s**
Y. Shi et al [20]						
Part	ASR	AET	MAX ID	MAX ET	MINET	TT
Black	97.5%	6.14 s	6.7 mm	22.8 s	3.6 s	1893 s
White	**100%**	**3.72 s**	6.2 mm	5.7 s	2.8 s	2161 s
Mix	80%	6.9 s	6.3 mm	22.3 s	3.5 s	4048 s
Hp jack	7.5%	4.1s	2.3 mm	5.2 s	3.8 s	4050 s

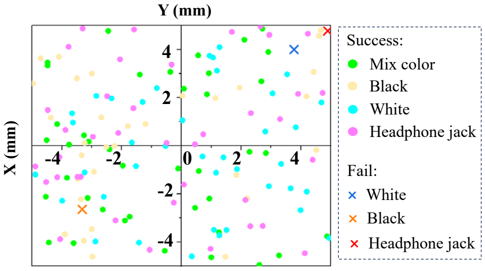

Fig. 9. Assembly results of three different textured parts. The figure shows the assembly situation at different initial positions.

can be successfully transferred in a few minutes or even tens of seconds. As for the method in [20], it should be noted that it can also achieve a high success rate when facing the white and black parts because these parts have a simple texture.

However, it's sensitive to the shape of the peg and hole. When the tip of the headphone jack has different shapes, the Cycle-GAN was hard to transfer the actual image to the simulation. On the other hand, our method only samples the pixels around the peg, so it's not affected by the shape of the hole part and only focuses on the texture differences between the hole and the hole surfaces.

Fig. 9. shows the overall testing results. The failed cases were randomly distributed in various positions. The reason for the failure was mainly due to the grasp stability. Although the gripper kept closing during the 40 tests, and re-calibration was conducted before each epoch. The pose of the peg may undergo slight changes during hole searching due to contact forces, leading to shifts in SBI sampling coordinates and discrepancies between the extracted features and those used during transfer training. Although the peg remains near the hole during execution, such discrepancies, combined with the sub-millimeter clearance, can prevent the task from being successfully completed.

Fig. 10. shows the assembly process. The tasks start from random initial positions, including situations where the hole is occluded in the camera view. It can be seen that the RL policy did not direct the peg immediately towards the hole. Instead, the peg was maneuvered in a continuous trajectory around the hole's vicinity. Once aligned with the hole, the peg rapidly moved down under the influence of force control in Z direction, while the system detect the contact force smaller than 5N, it switched to the compliance control mode to complete the insertion. This strategic approach of persistent exploration near the hole substantially enhanced the system's adaptability to positional inaccuracies.

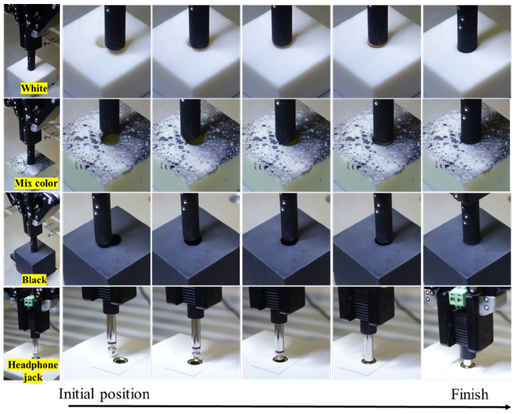

Fig. 10 UR5 executes the assembly operation successfully.

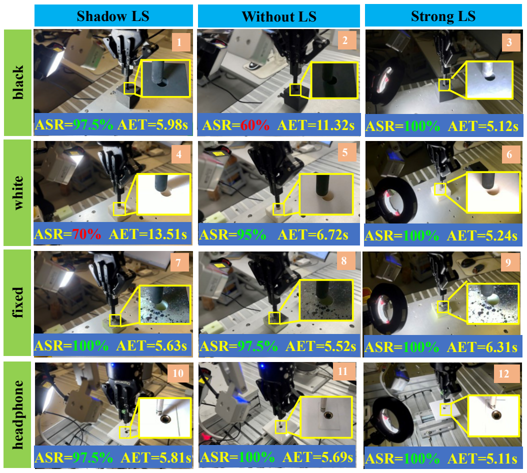

Fig.11 Experiment results in different lighting environments. Including Shadow lighting source (LS), Strong LS, and Without LS environments.

3) Result and Discussion (different lighting): As shown in Fig.11, our method performs well under most lighting conditions, with the success rate achieving over 95%. However, the success rates for Task 2 and 4 fall below 0.7, primarily because the shadowed regions coincide with the SBI areas, where the low contrast between the shadows and the holes leads to weak feature variation. This results in missing information and prevents the encoder from fully capturing distance features. We attempted to increase the encoder depth from 3 to 5 layers to enhance encoding capacity, but only improved the success rate from around 0.3 to 0.6, which was still unable to achieve satisfactory results.

Finally, we investigated and compared several recent studies of learning-based assembly approaches, with a particular focus on the success rate for round peg insertion, as summarized in Table III. The "Training environments" column indicates whether the RL policy was trained in simulation or real world. The success rate was obtained by testing the policy in a real-world environment. Notably, our assembly skill was learned in simulation and can maintain a high average success rate in real-world applications with the proposed methods. Additionally, its persistent searching behavior near the hole makes it less sensitive to clearance, and if the control frequency and precision can be improved, smaller clearances

TABLE III
COMPARISON OF THE SUCCESS RATE WITH OTHER WORKS ON THE ROUND PEG-IN-HOLE ASSEMBLY

Methods	Training environments	Clearance	Success rate
Zang et al. [14]	Sim	0.1mm	78%
Yasutomi et al. [15]	Real	0.2mm	93.9%
Kozlovsky et al. [25]	Sim	0.4mm	84%
Arik et al. [26]	Sim	1mm	84.29%
Cristian C et al. [27]	Sim+Real	0.2mm	80%
Cai et al. [28]	Real	0.80mm	89%
Ours	**Sim**	**0.13~0.37mm**	**94.3%**

could theoretically be handled.

V. CONCLUSION

This work proposes a visual domain adaption-based fast skill transfer method for peg-in-hole assembly in various visual conditions. The transfer speed is significantly improved by utilizing the proposed low dimensional SBI and transfer learning in the distance latent spaces. The proposed DBEM maps the latent representation to the physics distance spaces by reconstructing the DBI. The fine-tuning transfer learning method makes the features extracted from the target tasks similar to the source tasks, thus eliminating the need to retrain the RL policy. The results demonstrate that our method achieves favorable outcomes in both simulated and real-world applications. It requires only a brief training period, consisting of a 5 mins data collection phase followed by a 3 mins transfer learning process. This efficiency holds promise for rapid deployment in flexible manufacturing industries.

Our method also has certain limitations. When the contrast between the inside and outside of the hole is insufficient, or the shadow falls into the SBI area, the encoder may fail to extract reliable features, leading to transfer failures. Orientation skill transfer is also not considered in the current stage. Future work will focus on optimizing network architecture to enhance the feature extraction ability. And incorporating multi-modal information will be studied to cope with assembly tasks with different shapes of parts in 6D space.

REFERENCES

[1] J. Zhang *et al.*, “Dexterous hand towards intelligent manufacturing: A review of technologies, trends, and potential applications,” *Robotics and Computer-Integrated Manufacturing*, vol. 95, p. 103021, Oct. 2025.

[2] L. Xie *et al.*, “Learning Active Force–Torque Based Policy for Sub-mm Localization of Unseen Holes,” *IEEE Transactions on Industrial Informatics*, vol. 20, no. 4, pp. 6726–6738, Apr. 2024.

[3] X. Ma and D. Xu, “Automated robotic assembly of shaft sleeve based on reinforcement learning,” *Int J Adv Manuf Technol*, vol. 132, no. 3, pp. 1453–1463, May 2024.

[4] J. Jiang, Z. Huang, Z. Bi, X. Ma, and G. Yu, “State-of-the-Art control strategies for robotic PiH assembly,” *Robotics and Computer-Integrated Manufacturing*, vol. 65, p. 101894, Oct. 2020.

[5] A. Stemmer, G. Schreiber, K. Arbter, and A. Albu-Schaffer, “Robust Assembly of Complex Shaped Planar Parts Using Vision and Force,” in *IEEE MFI 2006*, Sep. 2006, pp. 493–500.

[6] J. Cui and J. Trinkle, “Toward next-generation learned robot manipulation,” *Science robotics*, vol. 6, no. 54, p. eabd9461, 2021.

[7] L. Jin, Y. Men, R. Song, F. Li, Y. Li, and X. Tian, “Robot Skill Generalization: Feature-Selected Adaptation Transfer for Peg-in-Hole Assembly,” *IEEE Transactions on Industrial Electronics*, pp. 1–10, 2023.

[8] F. Li, Q. Jiang, S. Zhang, M. Wei, and R. Song, “Robot skill acquisition in assembly process using deep reinforcement learning,” *Neurocomputing*, vol. 345, pp. 92–102, Jun. 2019.

[9] T. Johannink *et al.*, “Residual Reinforcement Learning for Robot Control,” in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6023–6029.

[10] X. Zhao, H. Zhao, P. Chen, and H. Ding, “Model accelerated reinforcement learning for high precision robotic assembly,” *Int J Intell Robot Appl*, vol. 4, no. 2, pp. 202–216, Jun. 2020.

[11] J. Zhao, Z. Wang, L. Zhao, and H. Liu, “A Learning-Based Two-Stage Method for Submillimeter Insertion Tasks With Only Visual Inputs,” *IEEE Transactions on Industrial Electronics*, pp. 1–10, 2023.

[12] G. Schoettler *et al.*, “Deep Reinforcement Learning for Industrial Insertion Tasks with Visual Inputs and Natural Reward Signals,” May 2019, Accessed: Aug. 30, 2023. [Online]. Available: https://openreview.net/forum?id=ryg5E-gy3E

[13] C. Finn, *et al*, “Deep Spatial Autoencoders for Visuomotor Learning,” Mar. 01, 2016, *arXiv*: arXiv:1509.06113. Accessed: Aug. 31, 2023. [Online]. Available: http://arxiv.org/abs/1509.06113

[14] Y. Zang, *et al*, “Geometric-Feature Representation Based Pre-Training Method for Reinforcement Learning of Peg-in-Hole Tasks,” *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3478–3485, Jun. 2023.

[15] A. Y. Yasutomi, *et al*, “Visual Spatial Attention and Proprioceptive Data-Driven Reinforcement Learning for Robust Peg-in-Hole Task Under Variable Conditions,” *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1834–1841, Mar. 2023.

[16] J. C. Triyonoputro, W. Wan, and K. Harada, “Quickly Inserting Pegs into Uncertain Holes using Multi-view Images and Deep Network Trained on Synthetic Data,” in *IROS 2019*, 2019, pp. 5792–5799.

[17] W. Chen, C. Zeng, H. Liang, F. Sun, and J. Zhang, “Multimodality Driven Impedance-Based Sim2Real Transfer Learning for Robotic Multiple Peg-in-Hole Assembly,” *IEEE Transactions on Cybernetics*, pp. 1–14, 2023.

[18] K.-H. Ahn, M. Na, and J.-B. Song, “Robotic assembly strategy via reinforcement learning based on force and visual information,” *Robotics and Autonomous Systems*, vol. 164, p. 104399, Jun. 2023.

[19] L. Xie *et al.*, “Learning to Fill the Seam by Vision: Sub-millimeter Peg-in-hole on Unseen Shapes in Real World,” in *ICRA 2022*, May 2022, pp. 2982–2988.

[20] Y. Shi *et al.*, “A Sim-to-Real Learning Based Framework for Contact-Rich Assembly by Utilizing CycleGAN and Force Control,” *IEEE Transactions on Cognitive and Developmental Systems*, pp. 1–1, 2023.

[21] J. Li, *et al.*, “Using Goal-Conditioned Reinforcement Learning With Deep Imitation to Control Robot Arm in Flexible Flat Cable Assembly Task,” *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 4, pp. 6217–6228, Oct. 2024.

[22] R. Liu *et al.*, “An Intriguing Failing of Convolutional Neural Networks and the CoordConv Solution,” Dec. 03, 2018, *arXiv*: arXiv:1807.03247. Accessed: Jan. 05, 2024. [Online]. Available: http://arxiv.org/abs/1807.03247

[23] D. Zhang, W. Fan, J. Lloyd, C. Yang, and N. F. Lepora, “One-Shot Domain-Adaptive Imitation Learning via Progressive Learning Applied to Robotic Pouring,” *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 1, pp. 541–554, Jan. 2024.

[24] D. Liu, Y. Chen, and Z. Wu, “Digital Twin (DT)-CycleGAN: Enabling Zero-Shot Sim-to-Real Transfer of Visual Grasping Models,” *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2421–2428, May 2023.

[25] S. Kozlovsky, E. Newman, and M. Zacksenhouse, “Reinforcement Learning of Impedance Policies for Peg-in-Hole Tasks: Role of Asymmetric Matrices,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10898–10905, Oct. 2022.

[26] A. Lämmle *et al.*, “Simulation-based Learning of the Peg-in-Hole Process Using Robot-Skills,” in *IROS 2022*, Oct. 2022, pp. 9340–9346.

[27] 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, Art. no. 19, Jan. 2020

[28] Y. Cai, J. Song, X. Gong, and T. Zhang, “Robotic Peg-in-hole Assembly Based on Generative Adversarial Imitation Learning with Hindsight Transformation,” in *ICMA 2024*, Aug. 2024, pp. 1–7.