Review on Peg-in-hole Insertion Technology Based on Reinforcement Learning

Liancheng Shen
Institute of Automation
Chinese Academy of Sciences
School of Artificial Intelligence
University of Chinese Academy of Sciences
Beijing, China
shenliancheng2021@ia.ac.cn

Jianhua Su*
Institute of Automation
Chinese Academy of Sciences
Beijing, China
jianhua.su@ia.ac.cn

Xiaodong Zhang
Institute of Spacecraft System Engineering
China Academy of Space Technology
Beijing, China
15810002976@139.com

***Abstract*—Peg-in-hole insertion is a critical process in industrial production. Traditional peg-in-hole insertion methods are based on planning the robot's motion trajectory through the analysis of contact models. However, due to the complexity of contact states, it's challenging to establish precise and reliable contact models, leading to poor generalization of these methods. Reinforcement learning is a technique that learns insertion strategies from environmental interactions, avoiding the tedious process of analytical modeling. Thus, it has become a trending direction in the robotics field in recent years. This article aims to survey the mainstream peg-in-hole insertion technologies based on reinforcement learning methods and discuss future research directions. First, we introduce the task requirements for peg-in-hole insertion. Subsequently, a preliminary framework of reinforcement learning algorithms for peg-in-hole insertion is presented. Discussions are then divided into two main categories: traditional reinforcement learning methods (including model-based and model-free methods) and reinforcement learning methods accelerated by prior knowledge (including residual reinforcement learning, reinforcement learning from demonstration, meta-reinforcement learning, and other acceleration techniques). Finally, this article explores several potential future research directions for peg-in-hole insertion technologies based on reinforcement learning.**

***Index Terms*—Robot Peg-in-hole Insertion; Reinforcement Learning; Meta-Reinforcement Learning**

I. Introduction

A. Background Introduction

Peg-in-hole insertion is an extremely vital step in industrial production. It constitutes the foundational operation in significant equipment fabrication processes such as large-scale aerospace component assembly [1] and micro-scale microsystem assembly [2].

High precision and high compliance are the main challenges faced by robotic peg-in-hole insertion. In terms of precision, the insertion accuracy ranges from $0.02mm$ to $0.2mm$ [3], posing substantial challenges for robot planning precision, underlying control precision, and structural accuracy. Regarding compliance, assembly requires the right amount of contact force—too little contact force might not complete the insertion, while excessive force might destroy the parts or even the robot, leaving residual stress on the assembly parts. To softly and smoothly insert the peg into the hole, researchers have proposed passive compliance and active compliance to enable the robot's "gentle" operations. Passive compliance is achieved through mechanical structures [3], but usually requires designing specific compliance mechanisms for specific objects, compromising flexibility. Active compliance uses force sensor feedback control to achieve assembly, providing good adaptability and becoming the primary method for robotic compliant operations.

The active compliance control strategies for peg-in-hole insertion can be categorized into two broad types [4]:

1) Contact model-based: Analyzing the mechanical model of the peg-in-hole contact process, dividing the assembly process into various phases based on contact states, and adopting different strategies for assembling during different phases [4], [5].
2) Contact model-free: Using machine learning methods, leveraging human demonstration data for learning [6], or learning directly from the environment [7].

Assembly strategies that rely on the contact model are split into contact state identification and compliant control phases. Initially, based on the sensor input data, analytical methods [3] or statistical methods [8] are employed to identify the peg-in-hole contact state, guiding the robot's compliant assembly actions. When the peg and hole transition to a new contact state, a new state recognition is undertaken, and the process is repeated until the robot completes the assembly. For instance, for cylindrical peg-in-hole insertion, the entire insertion process can be arbitrarily divided based on the number of contact points between the peg and hole, such as no contact, one-point contact, and two-point contact [3]. Robot arm motion is then guided based on varying contact states. However, when part geometric parameters change, a re-analysis of the peg-

This work was supported in part by Beijing Natural Science Foundation under grant number L201019, National Natural Science Foundation of China under grant number 62273343; supported by the key laboratory of spaceflight dynamics technology Foundation under grant number KGJ6142210210311, XTB6142210210302.

Corresponding authors: jianhua.su@ia.ac.cn

2023 China Automation Congress (CAC) | 979-8-3503-0375-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/CAC59555.2023.10450390

in-hole contact model is needed. Moreover, as the required accuracy for assembly increases and the shapes of assembly parts become more complex, establishing an accurate peg-in-hole contact model becomes challenging.

Peg-in-hole insertion methods that do not rely on contact models are divided into those based on imitation learning and those based on reinforcement learning. Imitation learning-based peg-in-hole insertion methods [6] do not function through pre-programming but directly imitate human assembly action demonstrations, allowing the robot to mimic human flexible assembly behaviors. It is worth noting that robots can imitate not just the demonstration trajectory, but also the compliant adjustments of force. Imitation learning can significantly enhance the learning efficiency of robot assembly strategies. However, there's limited research on uncertain dynamic scenarios (like non-fixed pegs or holes) and new operational scenarios for peg-in-hole insertion.

Reinforcement learning, with its capability to learn skills from interactions, is considered one of the significant methodologies for robot skill learning [9]. Moreover, reinforcement learning enhances the modeling capability for intricate problems when combined with deep neural networks. Peg-in-hole insertion methods based on reinforcement learning are categorized into model-based and model-free. Model-based methods converge quickly, but they have issues related to algorithmic stability and learning safety [10]. Model-free methods can better adapt to changing environments, but their data utilization efficiency is not high [11]. To overcome the shortcomings of the aforementioned methods, researchers have proposed combining model-free reinforcement learning with domain prior knowledge to improve the robot's learning efficiency. This article will also introduce several mainstream methods combining model-free reinforcement learning with domain prior knowledge in the third section.

B. Purpose of this Paper

Numerous articles on peg-in-hole insertion have been published to date, but there are fewer reviews on peg-in-hole insertion strategies based on reinforcement learning. This paper investigates and summarizes articles in the domain of robotic peg-in-hole insertion that utilize reinforcement learning algorithms to train insertion strategies, categorizing them into traditional reinforcement learning and reinforcement learning accelerated by prior knowledge. Furthermore, it aims to analyze the current conditions for the application of prior knowledge, understand the distinct characteristics of various prior knowledge, and explore feasible solutions applied to robotic peg-in-hole insertion by leveraging their respective advantages. The primary objectives of this paper are:

1) To trace and summarize the progress of peg-in-hole robotic insertion technologies based on reinforcement learning and analyze the current developmental trends in this research domain.
2) To initially categorize the peg-in-hole robotic insertion techniques based on reinforcement learning and provide a brief analysis of the primary methods within each category.
3) To present our insights on some of the existing challenges in current methodologies and discuss open problems that may be researched in the future.

The remaining sections of this paper are organized as follows: The second section analyzes the mathematical expression of peg-in-hole insertion and introduces some classical reinforcement learning methods used for peg-in-hole insertion. The third section presents several improved reinforcement learning methods integrated with domain prior knowledge, elaborating on their characteristics. The fourth section summarizes the potential future research directions.

II. Peg-in-hole Insertion Strategies Based on Traditional Reinforcement Learning

A. Mathematical Definition

The robotic peg-in-hole insertion process exhibits certain Markovian characteristics, and thus it's typically described using Markov processes [12]. A Markov process is represented by $< S, A, P, R, \gamma >$, where these symbols respectively denote state space, action space, transition probability, reward function, and discount factor.

1) State Space: In the peg-in-hole insertion process, the state space input information for robots includes visual information, force/torque information, and joint/position information. In practical applications, multi-modal hybrid inputs are often employed [12]. Different types of sensors have their respective advantages and disadvantages. For example, while visual information is cost-effective, it's primarily used for scenarios with simple contact states and less stringent insertion precision requirements. Force/torque information can detect contact forces well, effectively simulating human insertion processes; however, high-precision force sensors are pricey and easily affected by noise. Joint/position information can also be input directly to the robot's controller, but it requires complex data preprocessing. Thus, effectively integrating diverse input information to realize faster, more compliant, and precise insertion remains an open issue in robotics.

2) Action Space: The robot's action space can be divided into two categories. The first type of action space consists of the robot's end-effector pose set. Based on the desired end-effector pose and combined with inverse kinematics, the angles of each robot joint are calculated to complete action control. In practical applications, this type of action can be equivalently described as translational motion in different directions and rotational motion around different axes by fixed steps. For example, defining translation component δ and rotation component α, the robot's action space can be represented as $a_t = [\delta_x, \delta_y, \delta_z, \alpha_x, \alpha_y, \alpha_z]$ [13]. Additionally, the robot's end-effector speed can be controlled, indirectly achieving the purpose of controlling its pose [14]. The second type of action space consists of the robot's joint angles set, often used for complex multi-degree-of-freedom robots, such as dexterous hands [15]. The first type of action space doesn't

require learning the robot's dynamic model and has better transferability, thus being more widely used.

3) Transition Probability: The state transition probability represents the likelihood of the robot transitioning from its current state to another state after taking a certain action. It can be described as $P_{sa} = P(s_{t+1} = s'|s_t = s, a_t = a)$. It's worth noting that, in peg-in-hole insertion, the state transition probability is generally unknown.

4) Reward Function: Reward functions are primarily categorized into sparse rewards and dense rewards. Sparse rewards refer to giving a positive reward when the robot successfully inserts, with all other scenarios resulting in a reward of 0 [16]. Sparse rewards are simple to deploy, but it's challenging for the robot to learn insertion strategies directly from rewards. Hence, some scholars proposed dense reward methods, where each robot action is assigned a reward signal. Dense rewards often accompany many human-designed phases and hyperparameter settings, and selecting hyperparameters is very tedious and time-consuming. Current research focuses on how robots can learn quickly with sparse rewards.

5) Discount Factor: The discount factor represents the emphasis a robot places on future rewards. In peg-in-hole insertion, since the objective is to successfully insert the peg into the hole, the discount factor is generally set to 1 or 0.99, signifying greater emphasis on future rewards.

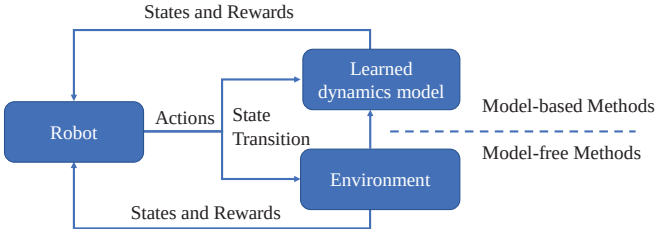

Fig. 1. Schematic diagram of traditional reinforcement learning

B. Model-free Methods

Researchers as early as 1992 discussed the integration of reinforcement learning and neural networks for peg-in-hole insertion problems [17]. However, at that time, due to the immature training methods of reinforcement learning and neural networks, the training time was long and the input vector dimension was low. Deep reinforcement learning, which is characterized by learning skills from interactions and has excellent modeling capabilities, has promoted its application in contact-rich problems such as peg-in-hole insertion.

Model-free methods can be divided into value-based methods and policy-based methods. Q-learning [11] is a representative algorithm for value-based methods. Gullapalli et al. [17] combined it with ϵ-greedy to accomplish the round peg-in-hole insertion task. DQN (Deep Q-network), using experience replay and a target network, has achieved outstanding performance in domains like gaming [17]. Experience replay ensures that training data in reinforcement learning meets the assumption of independent and identically distributed samples, which is crucial for neural network training. Using a target network can prevent the continuous movement of the target during neural network updates, making updates challenging. Therefore, many researchers have tried to introduce DQN into peg-in-hole insertion strategy learning. Inoue et al. [7] proposed replacing the regular MLP (Multilayer Perceptron) network with an LSTM (Long Short-Term Memory) network [18] to better estimate and compute Q-values. Experiments have shown that this method can complete assembly tasks that exceed the precision of the robot itself.

Value-based learning methods can only output discrete actions, making it challenging to achieve more refined continuous action learning. Policy-based learning methods can output continuous actions directly through the network, but the downside is that they have low learning efficiency, and policies easily get stuck in local optima (such as REINFORCE [19]). Therefore, researchers combined the two to propose reinforcement learning algorithms in the Actor-Critic framework, such as the DDPG (Deep Deterministic Policy Gradient) algorithm [20], which has a more stable training effect and has been successfully applied in peg-in-hole insertion.

Ren et al. [21] used DDPG to complete the peg-in-hole insertion task under continuous robot control. Xu et al. [13] proposed a force control model-based DDPG algorithm and a fuzzy reward mechanism to achieve multi-peg insertion. Hou et al. [22] proposed a Proportional-Derivative control framework-based KDDPG algorithm to achieve dual peg-in-hole insertion. In addition, algorithms like TD3 [23] and SAC [24] have also been applied to model-free reinforcement learning for peg-in-hole insertion [25]. However, compared to directly adjusting controller parameters, model-free reinforcement learning methods have a low sample utilization rate, and the controller's training convergence time is long. Therefore, how to better utilize models or prior knowledge becomes the key to improving performance.

C. Model-based Methods

The core idea of model-based methods is to first obtain a model of the environment and then use the model for action and policy learning. After a long period of development, classic model-based reinforcement learning algorithms such as Dyna [26], PILCO (Probabilistic inference for learning control) [27], and MPC (Model predictive control) [28] have emerged. In the field of peg-in-hole insertion, the GPS (Guided Policy Search) algorithm [10] is currently widely used. The GPS algorithm, for the established environmental dynamics model, minimizes the accumulated cost and the deviation of the policy network. It uses methods such as iLQR/iLQG [29] to obtain a guiding distribution. Then, training samples are extracted from the guiding distribution, and supervised learning is used to make the trajectory of the policy network approximate the trajectory of the guiding distribution.

Levine et al. [30] used this method to allow robots to learn a series of complex operations through a few minutes of interaction. Model-based methods are more efficient than traditional model-free learning methods, with a narrower ex-

ploration space and faster convergence. However, when the system rigidity is large and force or torque feedback is used, the system performance is affected due to the non-smooth system dynamics and a small trust region. To solve this problem, Luo et al. [31] proposed the MDGPS (Mirror Guided Policy Search). By explicitly incorporating force data into reinforcement learning, they achieved the assembly of tight gears. Luo et al. [32] also used this method to successfully insert a rigid peg into a deformable hole with a smaller diameter.

However, model-based methods need to learn the dynamics model of the target, and the quality of the model largely affects the control effect of the robot. Moreover, the learning rate for rigid systems using this method is relatively low, and its accuracy and robustness are not as good as model-free algorithms.

To overcome the inefficiency of model-free algorithms and the lack of stability and accuracy of model-based methods, Fan et al. [33] combined the two. First, they used GPS to obtain a simple controller and put the trajectories generated by the controller into the replay buffer. DDPG is then used to sample and learn from the trajectory replay buffer and the environment. In other words, in the initial search, model-based methods guide learning, helping to search for an optimal path. Model-free methods are used later for adjustments, thus considering both the efficiency and accuracy of the algorithm.

III. Combining Domain Prior Knowledge for Peg-in-hole Insertion

Traditional reinforcement learning methods can complete typical peg-in-hole insertion tasks. However, for complex-shaped peg-in-hole insertion problems or application scenarios that require rapid algorithm convergence, traditional reinforcement learning methods take longer to train because they do not consider the prior knowledge of peg-in-hole insertion. Researchers have proposed certain domain prior knowledge that can be leveraged. This section summarizes several representative improved algorithms based on prior knowledge.

A. Residual Reinforcement Learning (Residual RL)

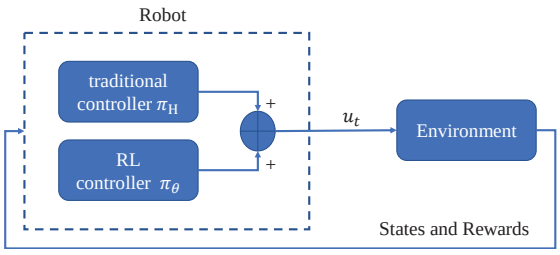

Fig. 2. Illustration of Residual Reinforcement Learning

Considering the characteristic of the assembly that the actual search space is relatively small, Johannink et al. [34] proposed that most robot actions can be controlled by traditional controllers, and reinforcement learning only needs to focus on learning complex contact actions. Therefore, the control signal u_t can be obtained from the traditional controller π_H and the reinforcement learning controller π_θ.

$$u_t = \pi_H(s_t) + \pi_\theta(s_t) \tag{1}$$

Using this method, Johannink et al. [34] achieved the task of inserting a block between two movable blocks. This scenario is challenging to model because the blocks can be moved, resulting in complex contact dynamics. Residual reinforcement learning can solve such problems, but the exact position of the block needs to be known. Schoettler et al. [25] used a simple p-controller combined with reinforcement learning to achieve plug-in assembly with only visual input and sparse rewards. Beltran et al. [35], [36] also used traditional controllers and combined residual actions with variable impedance learning to complete a soft insertion task for various peg-in-hole tasks using position-controlled robots.

In addition to designing the traditional controller π_H with modern control theory, more and more researchers are trying to train π_H from demonstration data using imitation learning. For example, researchers have explored combining DMP (Dynamic Motion Primitives) with reinforcement learning [37]. This method first uses DMP to extract the initial policy based on the demonstration trajectory, and then the residual policy trained by reinforcement learning corrects the trajectory generalized by DMP, finally achieving the assembly of round holes, gears, and network lines. The use of DMP effectively enhances the generalizability of the algorithm. Wang et al. [38] first used hierarchical imitation learning to learn nominal motion trajectories, and then combined it with the SAC algorithm to learn force control schemes, eventually achieving the assembly of a tightly coupled L-shaped object.

Residual reinforcement learning narrows the exploration space of reinforcement learning (most actions are performed by traditional controllers), reducing the training difficulty and sample demand of reinforcement learning. Also, since the search range is reduced, the training safety is significantly improved.

B. Reinforcement Learning from Demonstration (RLfD)

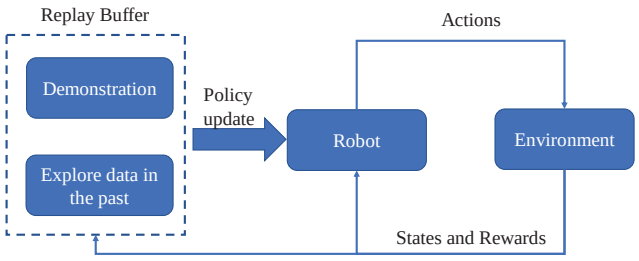

Fig. 3. Illustration of Reinforcement Learning from Demonstration

In long-term industrial practice, technicians have accumulated a series of script-based industrial robot assembly strategies and a large amount of training data from human demonstrations. One of the focuses is how to use these data

to assist the training of reinforcement learning and thereby accelerate the entire training process.

Reinforcement learning methods based on demonstrations are divided into two categories. One is the combination of action imitation learning and reinforcement learning, which has been described in Section III.A. The other type is directly using neural networks to perform end-to-end learning from demonstrations. The most direct method is to use Behavior Cloning [39], allowing the neural network to be initialized using demonstration data and then adjusted using reinforcement learning. This approach can avoid some abnormal actions caused by improperly set reward functions [40].

Inspired by DQfD (Deep q-learning from demonstrations) in the gaming field, Vecerik et al. [41] proposed DDPGfD (Deep deterministic policy gradient from demonstration), replacing the DQN in the algorithm with DDPG to adapt to continuous space control. They also used Prioritized Experience Replay [42] to give human demonstration data greater weight and added network weight regularization, N-step return accumulation, etc., to improve training results. Following this approach, Vecerik et al. [41] achieved dual peg-in-hole insertion under sparse rewards. However, the convergence speed of this method is slow. This is because when the robot explores an incorrect position, the correct approach is to pick up the workpiece and restore it to its original position for another assembly attempt, but it is challenging to learn this method with sparse rewards. To address this problem, Luo et al. [43] drew inspiration from DAGGER [44] and added an online correction module. When the robot explores an incorrect position, manual corrections are used. Experiments have shown that this method can achieve tight plug-in assembly with only about 10 corrections; it achieved a success rate of 99.8% in over 10,000 tests, reaching a level comparable to humans in some tasks. The algorithm's robustness also surpasses traditional force search methods.

Introducing expert demonstrations can prevent abnormal actions caused by improperly set reward functions. The design of a dense reward function requires a lot of engineering experience; otherwise, inappropriate reward signals may cause the robot to take dangerous actions. Introducing expert experience to design the reward function can allow robots to learn actions better under sparse rewards, avoiding the complicated reward function design process. Besides, expert demonstration data is generally optimal or suboptimal, so the expert experience can also help narrow the search range, meaning that the robot only needs to search for policies near the demonstration data. It is worth noting that there is a certain difference between this method and traditional imitation learning: traditional imitation learning focuses on encoding trajectories into various parameter models and then selecting or generating trajectories from the models. Although imitation learning can quickly reproduce demonstration actions, its generalization performance has certain limitations.

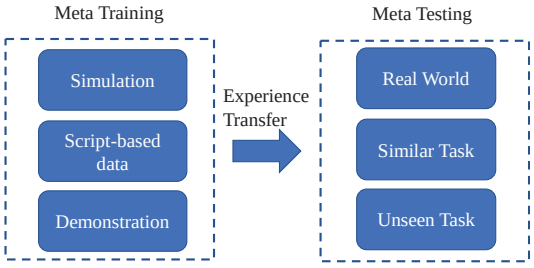

Fig. 4. Illustration of Meta Reinforcement Learning Process

C. Meta Reinforcement Learning (Meta RL)

Sim-to-real transfer is an essential means for robot skill learning. Sim-to-real techniques mainly include domain adaptation [45], [46], domain randomization [47], and progressive networks [48]. Domain randomization has been employed successfully in robot skill learning on several occasions [15], [49]. However, it has limitations, such as the need for precise model recognition, and a large amount of training required to transfer virtual skills to physical robots.

Meta-learning, also known as "learning to learn", aims to learn a more efficient learning algorithm, acquiring skills from past experiences to adapt to new tasks. Therefore, meta-learning offers a new solution for sim-to-real transfer. By transforming the goal of traditional meta-learning into a Markov decision process and utilizing the framework of reinforcement learning algorithms, meta-RL can be realized, aiming to quickly adapt to different new tasks and obtain optimal policies.

Meta-RL assumes that training tasks (old tasks) and new tasks come from the same distribution. This means that although there are differences between different tasks, there are commonalities that can be referenced. Let's denote the task as τ, then the task distribution is $p(\tau)$. In the meta-RL scenario, the goal is to use a small dataset D_{τ}^{tr} to let the neural network quickly adapt to the new task τ.

In the robotics domain, meta-RL is generally first conducted in a simulated environment and then transferred to physical robots. For instance, Schoettler et al. [16] used domain randomization in a simulation environment to train a network and then transferred it to physical robots. Ultimately, using less than 20 real-world experience data, the robot completed the plug-in assembly task.

Additionally, the industrial sector has a large amount of operation data and scripts, which can also be applied to the meta-RL process. For example, Zhao et al. [50] divided the training process into two phases: offline meta-pretraining and adapting to unknown tasks. In the offline meta-training phase, they used different tasks combined with different training data (teleoperation, online RL, script-based data) to assist in training the meta-network. During the adaptation phase, only a demonstration data segment is required for task inference. The final results show that for some scenarios very similar to the training tasks, the network can directly achieve a 100% assembly success rate. For some significantly different

scenarios, after half an hour of training, it can also achieve a 100% assembly success rate, further illustrating the robustness and deployability of the method. Some researchers also used model-based meta-RL methods combined with MPC control to complete plug-in assembly [51].

Methods based on meta-RL significantly reduce the exploration cost (most tasks are conducted in a simulated environment or offline), and can also discover commonalities between tasks, facilitating the expansion of robot operational skills.

D. Based on Other Prior Knowledge

There are also some approaches based on other prior knowledge, such as learning from the CAD models of plug-ins [52], using geometric motion planning to guide reinforcement learning, making learning faster and more reliable. The robot only needs a few minutes to learn operational skills and can still complete high-precision insertion tasks without accurate state estimation. Luck et al. [53] used a trajectory optimizer to guide the DDPG algorithm, making the learning algorithm and trajectory optimization promote each other. As a result, the robot can still complete peg-in-hole insertion operations with sparse rewards and pure image input. Jin et al. [54] used offline data to train the GVF (General Value Function) [55], achieving counterfactual predictions from visual inputs. Counterfactual prediction provides rich information representation, making robot learning more efficient. Combining counterfactual predictions with force feedback from online learning can achieve efficient skill learning with sparse rewards.

With the development of primitive learning in recent years [56], [57], a new perspective has been provided for the learning of complex robot actions. Primitive learning describes the entire task action as a collection of basic actions (primitives). An operation task might consist of hundreds of control commands, but at the same time, it can also be represented using a few primitives. The design of primitives comes from human prior knowledge, and its interpretability and robustness are significantly enhanced. The reduction in exploration space also shortens task complexity and learning time. In this direction, some researchers [14], [58] have used the method of action primitives for peg-in-hole insertion, achieving good results.

IV. Conclusion

This research surveyed various seminal works in the peg-in-hole insertion domain leveraging reinforcement learning algorithms and classified and analyzed them, primarily compared as shown in Table 1. Reinforcement learning-based methods can learn from the environment and address intricate peg-in-hole insertion problems that traditional contact models couldn't handle. Although recent advancements in reinforcement learning incorporating prior knowledge have made significant strides, there remains a substantial gap between current achievements and industrial deployment. To address the current challenges, we suggest several potential future research directions:

A. Industrial Deployment of Reinforcement Learning

While reinforcement learning-based peg-in-hole insertion has already demonstrated superior generalizability and robustness compared to traditional force search methods [43], research should extend beyond the laboratory. The widespread deployment and evaluation of reinforcement learning in industrial settings are crucial. Incremental learning and multi-stage task assembly, as previously mentioned, are critical for realizing industrial automation. This area may benefit from cross-research with findings from lifelong learning [60]–[62] and multi-task learning [63].

B. Combining Active and Passive Compliance

Considering potential issues like excessive rigidity during the peg-in-hole insertion process facilitated by reinforcement learning, combining active and passive compliance might be a valuable research direction. This combination might ensure safety while achieving superior assembly precision.

C. Sim-to-Real Transfer

In actual industrial production, given the time and safety costs, learning directly on physical robots can be challenging. The sim-to-real transfer offers a novel approach. As previously discussed, combining domain randomization with meta-learning has been proven to effectively facilitate sim-to-real transfers. However, current sim-to-real transfers often necessitate fine-tuning the network on the actual robot, which could compromise practicality and safety. The subsequent research direction might explore achieving favorable performance with zero-shot or few-shot sim-to-real transfers [15].

D. Offline Reinforcement Learning

A significant direction for reinforcement learning-based peg-in-hole insertion is further reducing training time. In addition to sim-to-real transfer, another promising approach is offline reinforcement learning followed by fine-tuning in the real training environment. Progress has already been made in this field [64]–[67], and it might be beneficial to incorporate these advancements into robotic assembly to expedite learning.

References

[1] Z. Qin, P. Wang, J. Sun, J. Lu, and H. Qiao, "Precise robotic assembly for large-scale objects based on automatic guidance and alignment," *IEEE transactions on instrumentation and measurement*, vol. 65, no. 6, pp. 1398–1411, 2016. Publisher: IEEE.

[2] R. Chang, C. Lin, and P. Lin, "Visual-based automation of peg-in-hole microassembly process," *Journal of Manufacturing Science and Engineering*, vol. 133, no. 4, 2011. Publisher: American Society of Mechanical Engineers Digital Collection.

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

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

[5] Y. Fei and X. Zhao, "An assembly process modeling and analysis for robotic multiple peg-in-hole," *Journal of Intelligent and Robotic Systems*, vol. 36, no. 2, pp. 175–189, 2003. Publisher: Springer.

[6] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009. Publisher: Elsevier.

TABLE I
COMPARISON OF VARIOUS REINFORCEMENT LEARNING METHODS FOR ROBOTIC PEG-IN-HOLE INSERTION TECHNOLOGY

Method	Type	Features	Representative Applications
Based on Traditional Reinforcement Learning	Model-free Methods	Simple algorithms, but low data efficiency.	[7], [13], [21]
	Model-based Methods	Fewer interactions with the environment, but lacks stability and safety.	[31]–[33]
Methods Accelerated with Domain Prior Knowledge	Residual RL	Policy is generated by superimposing traditional control and reinforcement learning, reducing the search space while enhancing safety.	[25], [34], [38]
	RLfD	Utilizes demonstration data for acceleration, avoiding the complexity of reward engineering, and narrowing down the exploration scope.	[41], [43], [59]
	Meta RL	Transfers past experiences to new tasks to reduce the number of online interactions, and facilitate deployment.	[16], [50], [51]
	Primitive Learning	Decomposes actions into several primitives, reducing the dimension of action space and search scope, and enhancing interpretability.	[14], [58]

[7] T. Inoue, G. De Magistris, A. Munawar, T. Yokoya, and R. Tachibana, "Deep reinforcement learning for high precision assembly tasks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 819–825, IEEE, 2017.

[8] Z. Jakovljevic, P. B. Petrovic, and J. Hodolic, "Contact states recognition in robotic part mating based on support vector machines," *The International Journal of Advanced Manufacturing Technology*, vol. 59, no. 1, pp. 377–395, 2012. Publisher: Springer.

[9] N.-J. Liu, T. Lu, Y.-H. Cai, and S. Wang, "A review of robot manipulation skills learning methods," *Acta Automatica Sinica*, vol. 45, no. 3, pp. 458–470, 2019.

[10] S. Levine and V. Koltun, "Learning complex neural network policies with trajectory optimization," in *International Conference on Machine Learning*, pp. 829–837, PMLR, 2014.

[11] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. 2018.

[12] L. Stan, A. F. Nicolescu, and C. Pupăză, "REINFORCEMENT LEARNING FOR ASSEMBLY ROBOTS: A REVIEW," vol. 15, p. 13, 2020.

[13] J. Xu, Z. Hou, W. Wang, B. Xu, K. Zhang, and K. Chen, "Feedback deep deterministic policy gradient with fuzzy reward for robotic multiple peg-in-hole assembly tasks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1658–1667, 2018. Publisher: IEEE.

[14] X. Zhang, S. Jin, C. Wang, X. Zhu, and M. Tomizuka, "Learning insertion primitives with discrete-continuous hybrid action space for robotic assembly tasks," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 9881–9887, IEEE, 2022.

[15] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, and others, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020. Publisher: SAGE Publications Sage UK: London, England.

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

[17] V. Gullapalli, R. A. Grupen, and A. G. Barto, "Learning reactive admittance control.," in *ICRA*, pp. 1475–1480, Citeseer, 1992.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997. Publisher: MIT Press.

[19] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3, pp. 229–256, 1992. Publisher: Springer.

[20] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International conference on machine learning*, pp. 387–395, PMLR, 2014.

[21] T. Ren, Y. Dong, D. Wu, and K. Chen, "Learning-based variable compliance control for robotic assembly," *Journal of Mechanisms and Robotics*, vol. 10, no. 6, p. 061008, 2018. Publisher: American Society of Mechanical Engineers.

[22] Z. Hou, H. Dong, K. Zhang, Q. Gao, K. Chen, and J. Xu, "Knowledge-driven deep deterministic policy gradient for robotic multiple peg-in-hole assembly tasks," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 256–261, IEEE, 2018.

[23] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*, pp. 1587–1596, PMLR, 2018.

[24] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, pp. 1861–1870, PMLR, 2018.

[25] 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)*, pp. 5548–5555, IEEE, 2020.

[26] B. Peng, X. Li, J. Gao, J. Liu, K.-F. Wong, and S.-Y. Su, "Deep dyna-q: Integrating planning for task-completion dialogue policy learning," *arXiv preprint arXiv:1801.06176*, 2018.

[27] M. Deisenroth and C. E. Rasmussen, "PILCO: A model-based and data-efficient approach to policy search," in *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pp. 465–472, Citeseer, 2011.

[28] A. Afram and F. Janabi-Sharifi, "Theory and applications of HVAC control systems–A review of model predictive control (MPC)," *Building and Environment*, vol. 72, pp. 343–355, 2014. Publisher: Elsevier.

[29] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4906–4913, IEEE, 2012.

[30] S. Levine, N. Wagener, and P. Abbeel, "Learning contact-rich manipulation skills with guided policy search," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 156–163, 2015.

[31] 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)*, pp. 3080–3087, IEEE, 2019.

[32] J. Luo, E. Solowjow, C. Wen, J. A. Ojea, and A. M. Agogino, "Deep reinforcement learning for robotic assembly of mixed deformable and rigid objects," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2062–2069, IEEE, 2018.

[33] Y. Fan, J. Luo, and M. Tomizuka, "A learning framework for high precision industrial assembly," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 811–817, IEEE, 2019.

[34] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine, "Residual reinforcement learning for robot control," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6023–6029, IEEE, 2019.

[35] C. C. Beltran-Hernandez, D. Petit, I. G. Ramirez-Alpizar, T. Nishi, S. Kikuchi, T. Matsubara, and K. Harada, "Learning force control for

contact-rich manipulation tasks with rigid position-controlled robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5709–5716, 2020. Publisher: IEEE.

[36] 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. Publisher: MDPI.

[37] A. Wan, J. Xu, H. Chen, S. Zhang, and K. Chen, "Optimal path planning and control of assembly robots for hard-measuring easy-deformation assemblies," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 4, pp. 1600–1609, 2017. Publisher: IEEE.

[38] Y. Wang, C. C. Beltran-Hernandez, W. Wan, and K. Harada, "Robotic imitation of human assembly skills using hybrid trajectory and force learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11278–11284, IEEE, 2021.

[39] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," *Advances in neural information processing systems*, vol. 1, 1988.

[40] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," *arXiv preprint arXiv:1709.10087*, 2017.

[41] M. Vecerik, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, and M. Riedmiller, "Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards," *arXiv preprint arXiv:1707.08817*, 2017.

[42] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized Experience Replay," in *ICLR (Poster)*, 2016.

[43] J. Luo, O. Sushkov, R. Pevceviciute, W. Lian, C. Su, M. Vecerik, N. Ye, S. Schaal, and J. Scholz, "Robust multi-modal policies for industrial assembly via reinforcement learning and demonstrations: A large-scale study," *arXiv preprint arXiv:2103.11512*, 2021.

[44] A. Attia and S. Dayan, "Global overview of imitation learning," *arXiv preprint arXiv:1801.06503*, 2018.

[45] E. Tzeng, C. Devin, J. Hoffman, C. Finn, X. Peng, S. Levine, K. Saenko, and T. Darrell, "Towards adapting deep visuomotor representations from simulated to real environments," *arXiv preprint arXiv:1511.07111*, vol. 2, no. 3, 2015.

[46] A. Gupta, C. Devin, Y. Liu, P. Abbeel, and S. Levine, "Learning invariant feature spaces to transfer skills with reinforcement learning," *arXiv preprint arXiv:1703.02949*, 2017.

[47] 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)*, pp. 23–30, IEEE, 2017.

[48] A. A. Rusu, M. Večerík, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell, "Sim-to-real robot learning from pixels with progressive nets," in *Conference on Robot Learning*, pp. 262–270, PMLR, 2017.

[49] 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)*, pp. 3803–3810, IEEE, 2018.

[50] 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," *arXiv preprint arXiv:2110.04276*, 2021.

[51] D. Liu, X. Zhang, Y. Du, D. Gao, M. Wang, and M. Cong, "Industrial Insert Robotic Assembly Based on Model-based Meta-Reinforcement Learning," in *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1508–1512, IEEE, 2021.

[52] G. Thomas, M. Chien, A. Tamar, J. A. Ojea, and P. Abbeel, "Learning robotic assembly from cad," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3524–3531, IEEE, 2018.

[53] K. S. Luck, M. Vecerik, S. Stepputtis, H. B. Amor, and J. Scholz, "Improved exploration through latent trajectory optimization in deep deterministic policy gradient," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3704–3711, IEEE, 2019.

[54] J. Jin, D. Graves, C. Haigh, J. Luo, and M. Jagersand, "Offline learning of counterfactual perception as prediction for real-world robotic reinforcement learning," *arXiv preprint arXiv:2011.05857*, 2020.

[55] R. S. Sutton, J. Modayil, M. Delp, T. Degris, P. M. Pilarski, A. White, and D. Precup, "Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction," in *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pp. 761–768, 2011.

[56] C. J. Bester, S. D. James, and G. D. Konidaris, "Multi-pass q-networks for deep reinforcement learning with parameterised action spaces," *arXiv preprint arXiv:1905.04388*, 2019.

[57] N. Vuong, H. Pham, and Q.-C. Pham, "Learning sequences of manipulation primitives for robotic assembly," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4086–4092, IEEE, 2021.

[58] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International journal of robotics research*, vol. 37, no. 4-5, pp. 421–436, 2018. Publisher: SAGE Publications Sage UK: London, England.

[59] Y. Wu, M. Mozifian, and F. Shkurti, "Shaping rewards for reinforcement learning with imperfect demonstrations using generative models," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6628–6634, IEEE, 2021.

[60] C.-Y. Hung, C.-H. Tu, C.-E. Wu, C.-H. Chen, Y.-M. Chan, and C.-S. Chen, "Compacting, picking and growing for unforgetting continual learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[61] S. Kolouri, N. A. Ketz, A. Soltoggio, and P. K. Pilly, "Sliced cramer synaptic consolidation for preserving deeply learned representations," in *International Conference on Learning Representations*, 2019.

[62] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 532–547, 2018.

[63] M. Crawshaw, "Multi-task learning with deep neural networks: A survey," *arXiv preprint arXiv:2009.09796*, 2020.

[64] A. Singh, A. Yu, J. Yang, J. Zhang, A. Kumar, and S. Levine, "Cog: Connecting new skills to past experience with offline reinforcement learning," *arXiv preprint arXiv:2010.14500*, 2020.

[65] A. Nair, M. Dalal, A. Gupta, and S. Levine, "Accelerating online reinforcement learning with offline datasets," *arXiv preprint arXiv:2006.09359*, 2020.

[66] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1179–1191, 2020.

[67] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.