Proceedings of the 2022 IEEE
International Conference on Robotics and Biomimetics
December 5-9, 2022, Xishuangbanna, China

2022 IEEE International Conference on Robotics and Biomimetics (ROBIO) | 978-1-6654-8109-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ROBIO55434.2022.10012027

Learning to Manipulate Tools Using Deep Reinforcement Learning and Anchor Information

Junhang Wei[1], Shaowei Cui[2], Peng Hao[2], and Shuo Wang[3], *Member, IEEE*

Abstract— **Endowing robots with tool manipulation skills helps them accomplish challenging tasks. While robots manipulate tools to achieve goals, the alignment of tools and targets is a noise-sensitive and contact-rich task. However, it is difficult to access the accurate pose of the tool and the target. When there is unknown noise in the observations, reinforcement learning can't be sure to perform well. In this paper, we define the easier-to-obtain accurate task-related information as *anchor* information and introduce a tool manipulation method based on reinforcement learning and anchor information, which can perform well when the observations include unknown noise. To evaluate the method, we build a simulated environment ToolGym, which includes four different kinds of tools and different noise sampling functions for each tool. Finally, we compare our method with baseline methods to show the effectiveness of the proposed method.**

I. Introduction

The ability to manipulate tools plays an important role in natural intelligence [1], which has also attracted a lot of attention in recent years in robotics [2]. To complete a tool manipulation process, robots need to complete the following four steps in sequence [3]: 1) realize the desired targets of tasks, 2) select appropriate objects as tools, 3) determine how to grasp and manipulate the selected tools, 4) grasp the tools and manipulate them to complete tasks.

Most prior works [4], [5], [6] have focused on studying the first three steps and assumed the last step would be successfully executed when the grasping and manipulating poses are successfully predicted. The predicted actions in the third step can be parsed from different forms of the method output (e.g., affordance maps [5] and keypoints [6]). But, in practice, the predicted grasping and manipulating poses often contain unknown noise that is possibly generated by the inaccurate prediction model or the calibration error of sensors. For grasping actions that contain unknown noise, most prior works [2] leveraged additional grasping methods [7] to complete the grasping actions. But few works studied manipulating actions. The manipulating process often requires higher prediction or execution accuracy, especially when aligning tools and targets. The unknown noise in the predicted outputs will degrade the performance of the manipulation. In this paper, we focus on learning a tool manipulation policy that can perform well in the alignment process when the observations (i.e., the predicted manipulating poses) include noise.

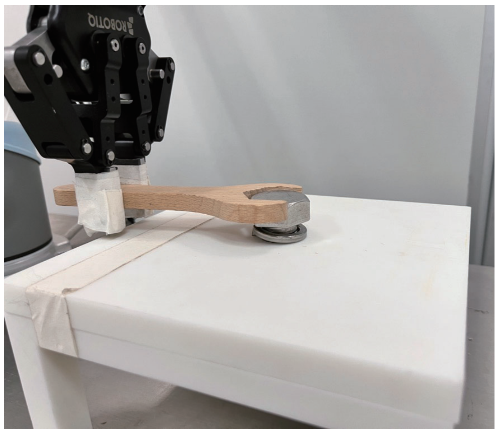


Fig. 1: An example of the robot manipulating a wrench to tighten a hex screw.

The alignment of tools and targets is critical for tool manipulation tasks, which is a noise-sensitive and contact-rich sub-task. Take wrenches and hex screws as an example (as shown in Fig. 1), the alignment processes can be thought of as a special kind of hole-on-peg task while the ends of the wrenches are holes and the heads of the screws are pegs. In screwdriver manipulation tasks, the alignment process is similar to that of the peg-in-hole task. Both of them are typical contact-rich tasks and are difficult to accomplish when the poses of holes and pegs (i.e., the tooltips and the targets) cannot be accurately acquired [8]. Although the observation of the tooltip contains unknown noise, the spatial relationship between the pose of the end-effector and the actual pose of the tooltip is fixed once the robot grasps the tool. In this paper, we define task-related information that can be easily obtained during task execution as *anchor* information (e.g., the accurate pose of the end-effector). It is possible to learn an efficient policy with the help of anchor

This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0103003, in part by the Beijing Natural Science Foundation under Grant 4222056, in part by the Beijing Nova Program under Grant Z211100002121152, and in part by the National Natural Science Foundation of China under U1913201. (Corresponding author: Shuo Wang.)

[1]Junhang Wei is with the School of Future Technology, University of Chinese Academy of Sciences, Beijing 100049, China, and also with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China weijunhang2018@ia.ac.cn

[2]Shaowei Cui and Peng Hao are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, {shaowei.cui, haopeng2017}@ia.ac.cn

[3] Shuo Wang is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, with the University of Chinese Academy of Sciences, Beijing 100049, China, and also with the Center for Excellence in Brain Science and Intelligence Technology Chinese Academy of Sciences, Shanghai 200031, China shuo.wang@ia.ac.cn

information.

In this paper, we leverage deep reinforcement learning (DRL) and anchor information to learn tool manipulation skills with imperfect observations. Reinforcement learning can learn policies from interactions with the environment, which can help robots handle many challenging situations that are hard to model. Neural networks are powerful function approximators that can directly learn from training data. DRL combines the advantages of both reinforcement learning and neural networks, and has been widely applied to address many challenging tasks [9]. Although the observation of the tooltip contains unknown noise, anchor information (e.g., the pose of the end-effector) can be used to assist the DRL model to learn the manipulation policy. Furthermore, to collect enough training data and evaluate our method, we design a simulated environment ToolGym that includes four different kinds of tools: the normal wrench, the pipe wrench, the ratchet wrench, and the screwdriver.

The outline of this paper is as follows. Section II provides related works. The details of the proposed method are given in Section III. The details of the simulation are given in Section IV. The experiments are presented in Section V. Section VI concludes this paper.

II. Related Work

A. Tool Manipulation

Learning to manipulate tools in robotics has been studied for decades [10], [11]. Toussaint *et al.* [12] formulated the tool manipulation problem as a constraint-satisfaction optimization problem and leveraged an optimization-based task-and-motion-planning method to solve the manipulation problem. Fang *et al.* [2] proposed a simulated self-supervised learning framework to learn the tool manipulation skill. The framework leverages reinforcement learning to control the robot to execute the manipulation. Do *et al.* [4] leveraged a segmentation model to predict the affordance label of each tool, and a planning method was utilized to generate the whole-body motions of the robot based on the affordance labels. Qin *et al.* [6] leveraged the keypoint representations to model the key primitive actions in tool manipulation skills and utilized other approaches to parse the predicted keypoints to generate appropriate robotic motions.

B. Addressing Imperfect Observations

To address the imperfect observations, abundant approaches have been proposed in recent years, and most of them have leveraged additional information to estimate the underlying states. Schoettler *et al.* [13] proposed an image-based reinforcement learning algorithm to deal with the imperfect state information and sparse reward signals for the industrial insertion task, where the visual inputs contain fully state information. Hao *et al.* [14] formulated the noise distributions as the task distributions of meta-reinforcement learning, and combined meta-reinforcement learning and residual reinforcement learning methods to handle the peg-in-hole tasks, where the meta-learning method leveraged context embeddings to represent the task information. Meng *et al.* [15] formulated the problem that the observation contains noise as a partially observable Markov decision process (POMDP) and leveraged the history information to estimate the underlying states, where the history can be thought of as *anchor* information in their experiments. In our case, considering the characteristics of the tool manipulation tasks, the spatial relationship between the end-effector and the tooltip is always the same for the specific tool once it has been grasped. Therefore, we can leverage both the pose of the tooltip with unknown noise and the actual pose of the end-effector to implicitly estimate the actual pose of the tooltip and learn an effective policy.

III. Method

A. Problem Statement

In DRL, each task has been formulated as a Markov decision process (MDP) that can be defined as a tuple $(\mathcal{S}, \mathcal{A}, P, r, \rho_0, \gamma)$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the transition probability distribution, $r : \mathcal{S} \to \mathbb{R}$ is the reward function, ρ_0 is the distribution of the initial state s_0, and $\gamma \in (0, 1)$ is the discount factor. Let π denote the policy and $R(\pi)$ denote the policy's expected discounted return. The goal of DRL is to learn an optimal policy π^* that can maximize the expected return $R(\pi)$. A partially observable Markov decision process (POMDP) [16] is a generalization of an MDP, but it assumes the state is partially observable (e.g., the observation contains unknown noise), and is defined as a tuple $(\mathcal{S}, \mathcal{A}, P, r, \rho_0, \gamma, \mathcal{O}, \Omega)$, where $\mathcal{O}$ denotes the additional observation space and Ω denotes the observation model. The performance of RL methods is not guaranteed in a POMDP problem.

Since the observation of the pose of the tooltip contains unknown noise, the manipulation task can be thought of as a POMDP problem. However, after the tool is once grasped by the robot, the spatial relationship between the tool and the end-effector is invariant for each task. Therefore, the actual pose of the end-effector can be thought of as anchor information for the manipulation task, which can assist DRL to handle the unknown noise in the observation.

B. Efficient Off-Policy Reinforcement Learning

Off-policy reinforcement learning is one of the classes of reinforcement learning. One of its implementations often selects appropriate actions by estimating the expected discounted return $Q(s_t, a_t)$ after the agent takes the action a_t when the state is s_t. The values of Q can be recursively estimated by the Bellman equation:

$$Q(s_t, a_t) = \mathbb{E}_{s_{t+1}}[r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})] \qquad (1)$$

Off-policy data can be used to learn the Q-values, where the off-policy data can be represented as (s_t, a_t, r_t, s_{t+1}). Therefore, off-policy reinforcement learning is sample-efficient. In this paper, we leverage the soft actor-critic (SAC) method [17] to learn the basic tool manipulation skills, where SAC is sample-efficient, stable, and requires few hyperparameter

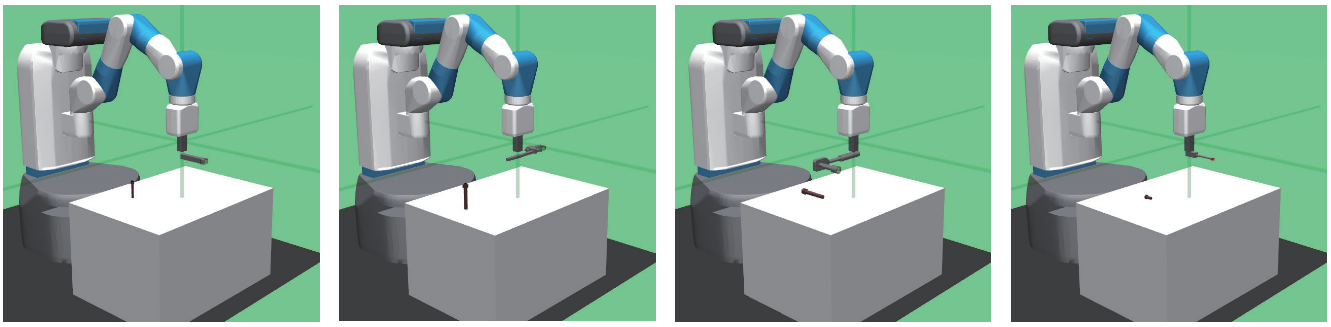

(a) Normal wrench (b) Pipe wrench (c) Ratchet wrench (d) Screwdriver

Fig. 2: An overview of the designed environment for four different kinds of tools.

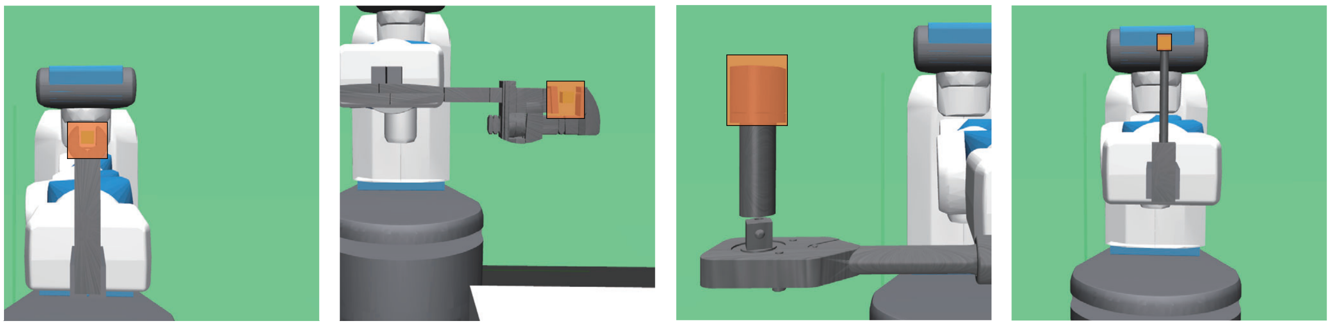

(a) Normal wrench (b) Pipe wrench (c) Ratchet wrench (d) Screwdriver

Fig. 3: An overview of the noise ranges for four different kinds of tools. After adding the noise to the observation, the received pose of the tooltip would be the pose of a point (i.e., the *fake* center point) belonging to the orange area.

adjustments. The SAC method can leverage a maximum entropy objective to learn stochastic policies. Since part of the observations contains unknown noise in our case, stochastic policies can encourage exploration to avoid sticking to the local minima and improve the robustness of the model to handle the changing noise [17].

IV. Simulation: ToolGym

In order to collect training data and evaluate the proposed method, we design a simulated environment based on the physical engine MuJoCo [18]. The environment is built upon the publicly available code at *gym* [19]. The environment contains three different wrenches: normal wrench (as shown in Fig. 2a), pipe wrench (as shown in Fig. 2b), ratchet wrench (as shown in Fig. 2c), and screwdriver (as shown in Fig. 2d), where the end of the socket wrench is a complete hole and manipulating it is supposed to be the most challenging task. Since we focus on studying the alignment problem in tool manipulation, all tools are fixed on the end-effector of the robot. The target pose of the screw in the environment is randomly generated within the workspace of the robot.

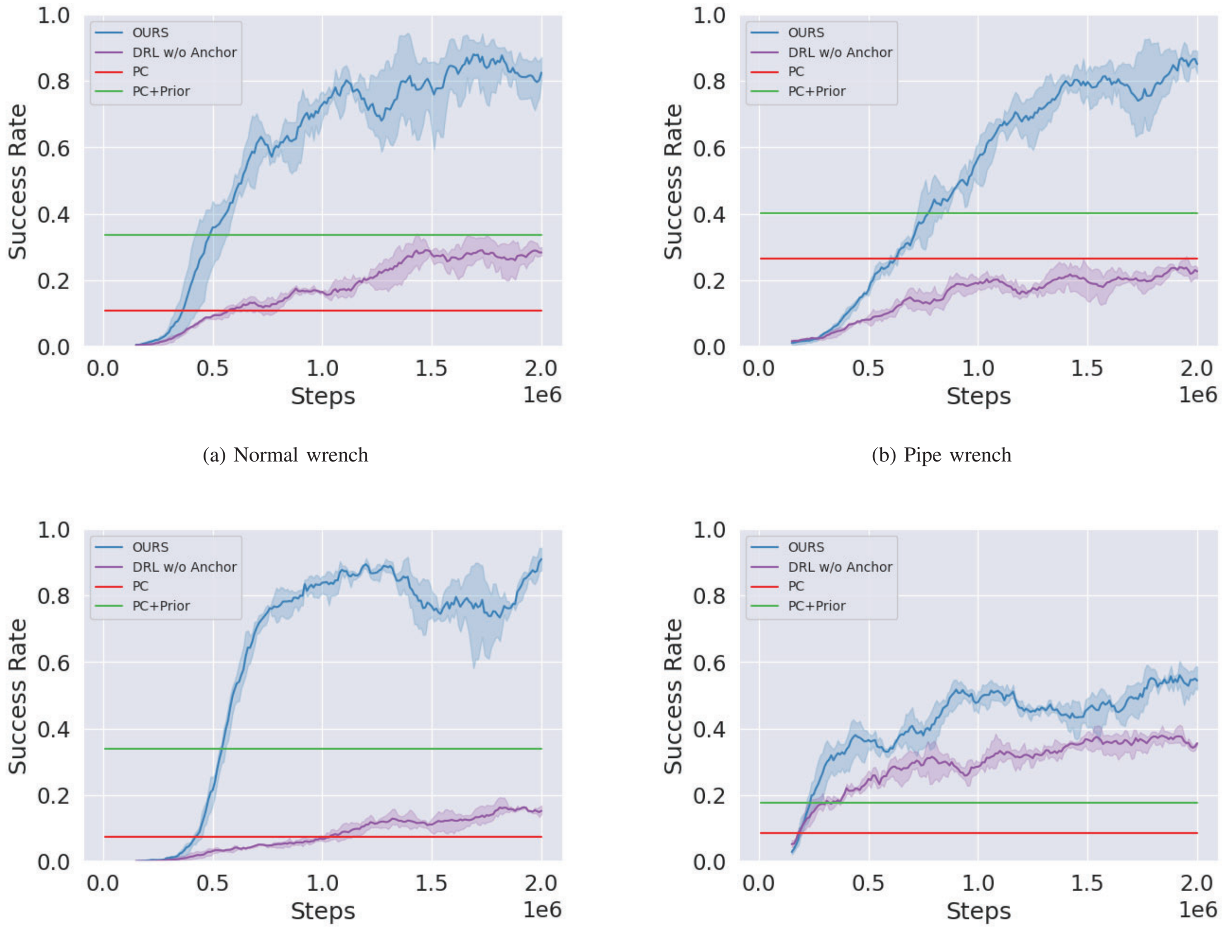
The observation in the environments includes three different vectors: the pose of the end-effector of the robot p_e, the pose of the tooltip p_t, and the pose of the target screw p_s. The correct values of p_e and p_s can be easily accessed in both the real world and the simulation. The pose of the tooltip p_t is obtained by predictions [4], [6] such that it could be inaccurate. Therefore, in the simulation, additional noise is added to the p_t to simulate real situations. Fig. 3 demonstrates the range of the offset of the real center point of the tooltip.

The action in the environments includes four dimensions (i.e., $a = (x, y, z, \theta)$), where (x, y, z) denotes the movement of the position of the end-effector and θ denotes the rotation of the end-effector along the axis that is perpendicular to the ground.

V. Experiments

In this section, we evaluate our method in all four environments (as shown in Fig. 2) and compare it with three baseline methods.

- **Position control (PC):** It controls the robot to move to the target pose based on the difference between the target pose and the observed pose that contains the noise of the tooltip.
- **Position control with prior knowledge (PC+Prior):** It is similar to *PC*, but it leverages the prior knowledge to improve the performance. Considering the characteristics of tool manipulation, the method first controls the robot to move the tool to the back of the target pose and then controls the robot to move forward to the target pose.
- **DRL without anchor information (DRL w/o Anchor):** It is similar to the proposed method, but the pose



(a) Normal wrench

(b) Pipe wrench

(c) Ratchet wrench

(d) Screwdriver

Fig. 4: Average testing success rates of different algorithms.

of the end-effector in the observations of the policy is replaced by a zero vector with the same dimensions.

A. Implementation Details

We implement the SAC algorithm based on the publicly available library *tianshou* [20]. We build all the neural networks in the SAC algorithm with the same architecture that includes two fully-connected layers, and each layer has 256 nodes. The learning rates of the actor and the critics are 0.001.

Noise function: For each tool, the noise range $(x_{min}, y_{min}, x_{max}, y_{max})$ is determined by the size of the tooltip, whose visualization results are shown in Fig. 3. The noise is also applied to the orientation, where the noise range is $[-\frac{\pi}{16}, \frac{\pi}{16}]$. In practice, the added noise $\eta = (\eta_x, \eta_y, \eta_\theta)$ is sampled from the uniform distributions, which means:

$$\eta_x \sim \mathcal{U}_{[x_{min}, x_{max}]} \tag{2}$$

$$\eta_y \sim \mathcal{U}_{[y_{min}, y_{max}]} \tag{3}$$

$$\eta_\theta \sim \mathcal{U}_{[-\frac{\pi}{16}, \frac{\pi}{16}]} \tag{4}$$

The noise is resampled whenever the environment is reset.

Reward function: In this paper, the robot receives a dense reward based on the distance from the real center point of the tooltip $p_r^t = (n_r^t, o_r^t)$ to the target screw head $p_s = (n_s, o_s)$, where n denotes the position and o denotes the orientation. The reward function is:

$$r^t = -\lambda \|n_r^t - n_s\|_2 - \rho \|o_r^t - o_s\|_2 \tag{5}$$

where $\lambda = 5$ and $\rho = 4$ in this paper.

B. Comparative Results

For the DRL-based algorithms, we repeatedly train the policies three times. Each time, the policy is tested on five isolated environments, with each environment containing 20 task instances. In other words, each trained policy is tested on 100 task instances with 100 noise values. For trained policies, we compute the moving average success rate and the average success rate of all policies. We evaluate the other baseline methods 300 times and compute their average success rate. The final average success rates in the testing environments are shown in Fig. 4.

In general, the proposed method outperforms the baselines in all four tasks since the baseline method cannot leverage

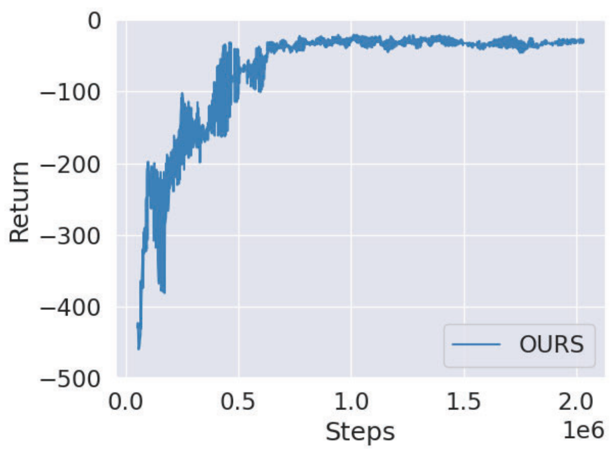
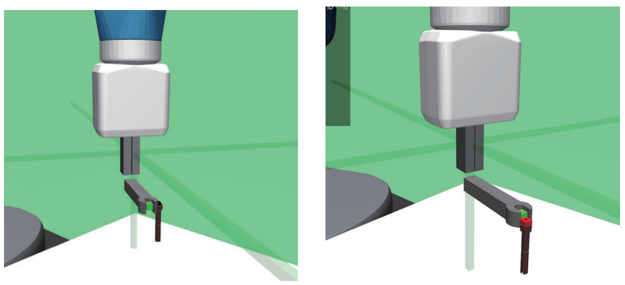


Fig. 5: Training returns for manipulating pipe wrench.



(a) Failure situation 1: the tool is stuck because of the unexpected contact

(b) Failure situation 2: the tool arrives at the error target pose because of the incorrect pose observed (i.e., the pose of the red cuboid)

Fig. 6: Examples of the failure situations when leveraging control method to manipulate tools and the observation contains noise.

correct information to handle the observation with noise. The experimental results demonstrate that without the help of anchor information, the tasks become POMDP problems and the performance of the DRL model is greatly degraded. For the proposed method, as the training progresses, the encountered noise situations during the testing phase are more likely to have already been experienced during training, so the model is more and more powerful in estimating the actual pose of the tooltip based on the observed information. Take the pipe wrench as an example, Fig. 5 demonstrates that the algorithm converges after about 7×10^5 steps, but the performance in the testing environment is still improving (as shown in Fig. 4b). For the DRL w/o Anchor, based on the MDP assumption, the policy can only perceive the current observations with unknown noise. Thus, it lacks enough task-related information to estimate underlying states.

Furthermore, the experimental results demonstrate that the traditional position control method can improve performance with the help of prior knowledge. For the traditional method, if the observation contains unknown noise, it will possibly fail in two situations (as shown in Fig. 6). For the PC+Prior method, we append a predefined workflow into the PC method based on prior knowledge about the tool manipulation tasks. It will improve the first failure situation (as shown in Fig. 6a). Because of the workflow, in the last step, the tooltip is always facing the target pose, making it more possible to complete the task when the actual pose of the tooltip is on the line connecting the incorrect pose and the target pose. The second failure situation will always exist (as shown in Fig. 6b). Because these methods cannot obtain the actual pose of the tooltip.

VI. Conclusion

In this paper, we study learning tool manipulation skills in robotics with unknown noise existing in observations. Observations including unknown noise are quite normal in the real world, which makes it difficult to estimate the true pose of the target points. We leverage the deep reinforcement learning method to address this problem with the help of anchor information. In this paper, we focus on the wrench and screwdriver manipulation that is close to the challenging task *hole-on-peg* and *peg-in-hole*. Both of them are typical kinds of contact-rich tasks. MuJoCo is a powerful physical engine to simulate contact among different objects. We design a simulated environment based on the MuJoCo engine to train tool manipulation skills and quickly collect data, which includes four different kinds of tools: the normal wrench, the pipe wrench, the ratchet wrench, and the screwdriver.

Although the proposed method outperforms the baseline methods in all four tasks, it does not perform as well in the screwdriver manipulation task as in the other tasks. Because the physical size of the end of the screwdriver and the slot of the screw is small, which makes this task more noise-sensitive than the others. Especially when the noise sampling function for orientation is the same for all tasks, the noise greatly influences the screwdriver task. A possible direction for future work is to leverage more environmental information or prior knowledge to (implicitly) estimate the actual pose of the tooltip. In this paper, we have focused on the manipulating process in the tool manipulation tasks and assumed the tool is fixed on the end-effector. However, in practice, the influence of the grasping point of the tool is also required to be considered. At each grasp, the spatial relationship between the pose of the end-effector and the pose of the actual tooltip may be different. Therefore, an interesting direction for future work is to leverage meta-learning methods to quickly adapt to the spatial relationship changes brought by the novel grasp point.

References

[1] E. Reynaud, M. Lesourd, J. Navarro, and F. Osiurak, “On the neurocognitive origins of human tool use: A critical review of neuroimaging data,” *Neuroscience & Biobehavioral Reviews*, vol. 64, pp. 421–437, 2016.

[2] K. Fang, Y. Zhu, A. Garg, A. Kurenkov, V. Mehta, L. Fei-Fei, and S. Savarese, “Learning task-oriented grasping for tool manipulation from simulated self-supervision,” *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 202–216, 2020.

[3] S. Brown and C. Sammut, *Tool Use and Learning in Robots*. Boston, MA: Springer US, 2012, pp. 3327–3330.

[4] T.-T. Do, A. Nguyen, and I. Reid, “Affordancenet: An end-to-end deep learning approach for object affordance detection,” in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 5882–5889.

[5] F.-J. Chu, R. Xu, and P. A. Vela, “Learning affordance segmentation for real-world robotic manipulation via synthetic images,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1140–1147, 2019.

[6] Z. Qin, K. Fang, Y. Zhu, L. Fei-Fei, and S. Savarese, “Keto: Learning keypoint representations for tool manipulation,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7278–7285.

[7] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, “Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics,” *arXiv preprint arXiv:1703.09312*, 2017.

[8] G. Schoettler, A. Nair, J. Luo, S. Bahl, J. Aparicio 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)*, 2020, pp. 5548–5555.

[9] S. Gupta, G. Singal, and D. Garg, “Deep reinforcement learning techniques in diversified domains: a survey,” *Archives of Computational Methods in Engineering*, vol. 28, no. 7, pp. 4715–4754, 2021.

[10] C. Lovchik and M. A. Diftler, “The robonaut hand: A dexterous robot hand for space,” in *Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C)*, vol. 2. IEEE, 1999, pp. 907–912.

[11] R. Holladay, T. Lozano-Pérez, and A. Rodriguez, “Force-and-motion constrained planning for tool use,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7409–7416.

[12] M. A. Toussaint, K. R. Allen, K. A. Smith, and J. B. Tenenbaum, “Differentiable physics and stable modes for tool-use and manipulation planning,” 2018.

[13] G. Schoettler, A. Nair, J. Luo, S. Bahl, J. A. Ojea, E. Solowjow, and S. Levine, “Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5548–5555.

[14] P. Hao, T. Lu, S. Cui, J. Wei, Y. Cai, and S. Wang, “Meta-residual policy learning: Zero-trial robot skill adaptation via knowledge fusion,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3656–3663, 2022.

[15] L. Meng, R. Gorbet, and D. Kulić, “Memory-based deep reinforcement learning for pomdps,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5619–5626.

[16] M. Igl, L. Zintgraf, T. A. Le, F. Wood, and S. Whiteson, “Deep variational reinforcement learning for pomdps,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 2117–2126.

[17] 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*. PMLR, 2018, pp. 1861–1870.

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

[19] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “Openai gym,” *arXiv preprint arXiv:1606.01540*, 2016.

[20] J. Weng, H. Chen, D. Yan, K. You, A. Duburcq, M. Zhang, Y. Su, H. Su, and J. Zhu, “Tianshou: A highly modularized deep reinforcement learning library,” *arXiv preprint arXiv:2107.14171*, 2021.