Learning to use different tools for objects rearrangement from demonstration

Yunchu Zhang*, Hao Zhu*, Zhaoyuan Fang,
Katerina Fragkiadaki, Christopher G. Atkeson

***Abstract*— Object rearrangement and cleaning tasks in complex scenes require the ability to utilize different tools. It is important to correctly switch between and deploy suitable tools. Previous works focus on either mastering manipulation tasks with a single tool, or learning task-oriented grasping for single tools. In this work, we propose an end-to-end learning framework that jointly learns to choose different tools and deploy tool-conditioned policies with a limited amount of human demonstrations. We evaluate our method on parallel gripper and suction cup picking and placing, brush sweeping, and household rearrangement tasks, generalizing to different configurations, novel objects, and cluttered scenes in the real world.**

I. INTRODUCTION

Humans use a lot of tools in daily life to complete different tasks (e.g open wine bottle with a wine-opener, cutting wood with a saw). Without various tools, we fumble at many tasks and even completely fail at some of them. Two key problems when learning to leverage tools are: 1) With multiple tools available, which one should be selected to best solve the task at hand? 2) What actions should be taken at each instant given the tool-task combination (how should the tool be used)? We argue that the ability to choose and deploy tools is crucial for successful use of tools both for humans and robots. In this paper, we explore how to enable robots to take full advantage of multiple tools. We propose a framework that jointly learns how to select the most suitable tool and how to best utilize the selected tool, from limited human demonstrations.

We build on Transporter Networks [13], an architecture that leverages alignment cues of deep features to predict action parameters from visual input. The architecture is general and is capable of performing various tasks like pushing piles of small objects and 6DoF pick-and-place. However, a separate model needs to be trained for each tool and an additional tool selection model has to be trained in order to leverage multiple tools. In this work, we build upon the framework of [13] to achieve the selection and use of multiple tools with a single model. Through extensive experiments, we demonstrate the effectiveness and the generalizability of the proposed framework.

Our main contributions are:

1) A generalizable framework for reasoning about the best tool to utilize and learning tool-conditioned polices from a small number of human demos.
2) A low-cost self-assemble suction gripper, and parallel gripper's tool adaptor for different tools including suction gripper, brush, which allows the robot to easily and steadily switch between different tools.
3) An efficient pipeline for collecting human demonstrations with minimal annotations.

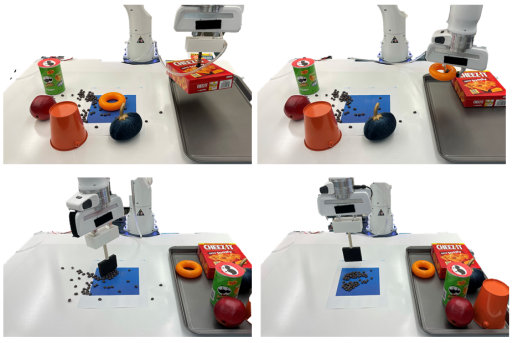

Fig. 1: **Rearrangement with different tools**

II. RELATED WORK

A. Tool Manipulation

Tool manipulation [2], [17] has been an essential problem for understanding intelligence in cognitive science studies. To enable robots to understand and manipulate tools, seminal works focus on predicting affordance and functional regions [3] for task specific grasping with self-supervision from physical interactions in a simulator [12]. [1] shows that manipulation of tools can be performed through learning and planning. However, these works only focus on how to utilize a single tool's different parts, in other words, how the tool should be grasped. In contrast, our work takes advantage of multiple tools and demonstrates generalization capabilities, through jointly learning a task-aware tool selection model and a manipulation policy from limited amount of human demonstrations.

Previous work [3], [9] learn to understand the synergy between grasping and manipulation for tool use. While most of these methods whether learn latent representations using end-to-end deep neural networks, or uses keypoint representations to provide structured and condense understanding of tool objects. However, our method does not rely on specific representations and motion primitives, instead, it only uses a general heatmap to predict the actions, and generalizes better to novel objects during test time.

*Authors with equal contribution, listed in alphabetical order.
The authors are with Carnegie Mellon University.

B. Tools for Robot

There has been works that aim to advance the design of tools to extend the range of tasks a single robot can achieve. This mainly refers to changing a robot's tools so it adapts to different tasks and scenarios more easily. [4] uses an electro-mechanical actuator to help switch between different tools, providing robust and precise tool-switching performance. However, it operates in the domain of industrial robots and comes at a high cost. [7] designs low cost interfaces that act as adaptors between different tools and the original end-effector. The design side of our work shares the idea of [6], which designs a series of low cost mechanical tools for robots by grasping two opposite sides of the tool. In [6], the main contribution lies in the hardware and the tool switching and deployment is performed manually. In contrast, our method adaptively selects the tool to use conditioned on the given scene.

C. Learning from Demonstration

In the data-driven (deep) imitation learning context, many of those works train one policy in an end-to-end manner to directly map observations to a sequence of actions or single step action [10], [11], [16]. These methods do not show much generalization across novel objects and scenes.

III. PROBLEM STATEMENT

[8] proposals to choose discrete and continuous actions separately, and make the discrete choices before making any continuous choices. Alternatives include making all choices simultaneously (such as using a neural network to predict both discrete and continuous outputs) or making continuous choices first such as choosing where an object had to move from or go, and then selecting a distinct strategy or tool. In our cases the discrete choice would be selecting a tool T and it's corresponded control parameters. This choice is made on the basis of the current task state, in this case what the robot sees O. This leads to the selection policy:

$$f_{select}(O) \rightarrow T \tag{1}$$

We follow [13] and assume most rearrangement tasks can be treated as a sequence of displacements of the objects such as: picking an object up and placing it at another location, or pushing a group of small particles from the initial position to the goal area. And actions can be specified by an initial and final pose of the tool. $P_{initial}$ and P_{final} are continuous parameters, generated given the task state and the discrete choices already made.

$$f_{continuous}(O, T) \rightarrow P_{initial}, P_{final} \tag{2}$$

An action a consists of discrete variables (selections) and continuous variables (which we could call parameters). Overall, we have a policy that maps from task state to an action:

$$f(O_t) \rightarrow a = (T, P_{initial}, P_{final}) \in \mathbb{A} \tag{3}$$

$\mathbb{A}$ is the set of possible actions. Time steps can be indicated by a subscript t.

There are several reasons for the the approach of separating discrete and continuous action choices, and making the discrete choices first. First, making the discrete choices may be simpler than making the continuous choices since there are fewer options for discrete variables compared to continuous ones. Second, the continuous choices or parameters are often not defined until a discrete choice of tool, strategy, or behavioral primitive has been made. And the landscape of the continuous choices may be simplified and possibly made convex by particular discrete choices. Thus, we favor a research approach that attempts to push all the complexity of action generation on to the discrete choices, leaving continuous choices that can be made by greedy algorithms like gradient descent. An alternative research agenda is to make all action generation continuous. This can be done by using probabilistic policies where a continuous parameter determines the probability of making a particular discrete choice .

IV. METHODS

A. Demo Collection

The field of reinforcement learning has come to recognize that "seeding" a policy with human demonstrations can speed up learning, or sometimes just make learning possible at all. There are several challenges, however. It is tedious and sometimes costly to collect large numbers of human demonstrations, clean up the typically noisy data, and then map human behavior on to robot capabilities. Robots don't typically have hands as capable as humans, aren't as compliant, and don't have the limb and body degrees of freedom and range of motion of a human, in addition to more limited perceptual capabilities. Our approach is to force humans to act using a robot-like hand whose behavior can be more easily captured. This reduces the "human2robot" capability gap, focuses attention on the relevant human actions (they only involve the robot-like hand), and help us avoid the inclusion of sub-optimal actions using simple filters. We constrain human demonstrators' actions by asking them to use two fixed tools as shown in Figure 3. The first fixed tool – grabber is used to collect the parallel grasping and suction grasping demos. And the second fixed tool – brush with self designed adaptor is used to collect the sweeping demos.

We put in frame a screen that alternates between two colors (red and green) to indicate the occurrences of keyframes and the keyframe indicator is generated by human demonstrator during the demonstration. More specifically, when the tools reach the pick/place positions, the human demonstrator uses a remote to prompt the keyframe indicator screen to change color. The timesteps at which the indicator screen changes color are extracted as the keyframes. To obtain robust training signals from the extracted keyframes, we train detectors with heavy augmentations for the two tools by labelling a tiny portion of the keyframes (1,000 pictures). We then use the detectors to generate pseudo-labels for all the demonstrations.

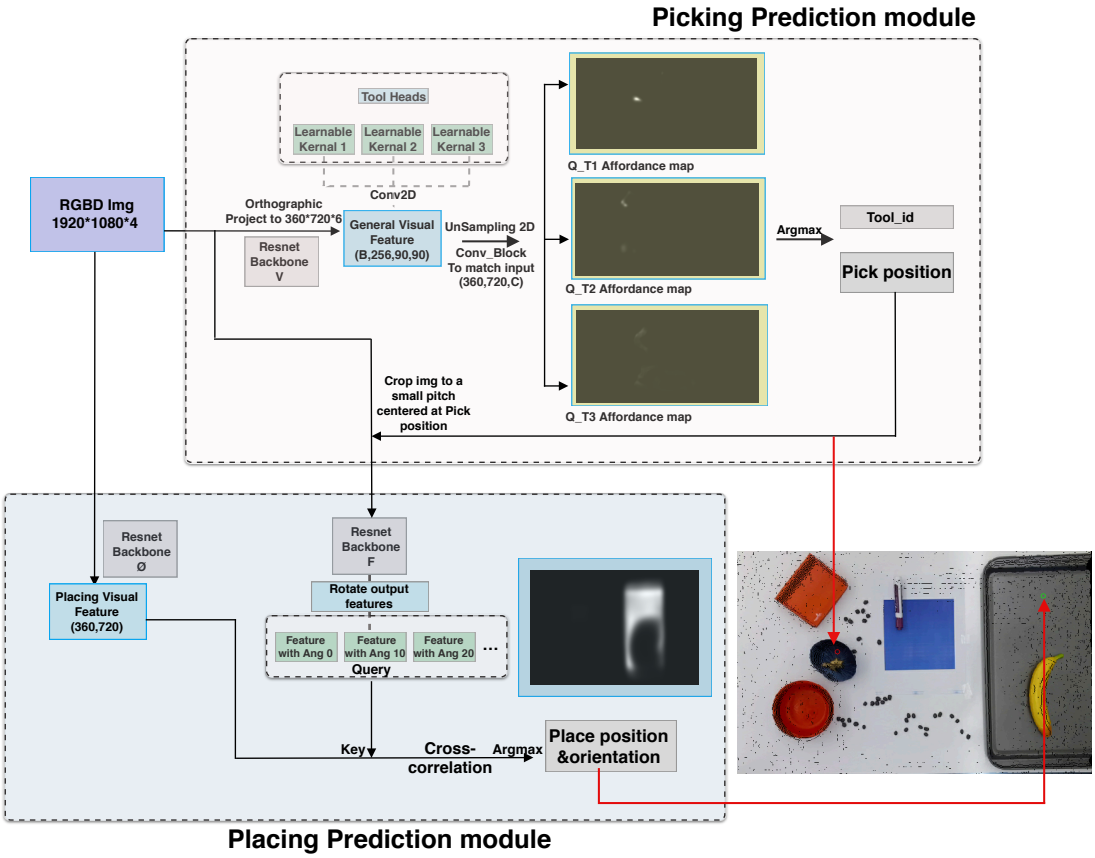

Fig. 2: **Network architecture**. Our method is composed of two modules: the picking prediction module and placing prediction module. The picking prediction module figures out which tool should be used and the picking location at the same time. The placing prediction module does cross-correlation on the rotated picking prediction module's output's features and placing feature to get the placing prediction.

For parsing pick and place demos, we use the detector's detection results for gripper as the pseudo-labels of the pick/place locations, and the pose of the Aruco tag (attached on the grabber) as pick/place rotation angles. For sweeping demos, we use the detector's 3 keypoints outputs to get both location and rotation. To get the tool affordance supervision, we design the collection process with pre-defined orders (e.g collect parallel grasping first then suction grasping and last for sweepings).

B. Representations and Algorithms

In order to solve a complex household rearrangement task as shown in Figure 2, the policy consists of two parts: the affordance-aware tool selection policy and the selection-conditioned continuous action policy. The affordance-aware tool selection module is in charge of figuring out which tool to deploy at each step and where to deploy it. In other words, it needs to be able to learn the affordance in the input image. For example, the robot could learn to first move the objects to clear the workspace for sweeping, instead of trying to sweep beans while the objects are still in the way. Given the predicted starting location, the second module chooses how the tool should act. We implement these policies as neural networks and train with gradient-based training algorithms.

a) Affordance-aware tool selection:: With the visual representations as input (see Sec. IV-C for details), we first use a ResNet-style [5] backbone to extract general visual features $\mathbf{v}$, which are tool-agnostic and contain entangled visual information relevant for all tasks. Then, for each tool T_i, a learnable tool-specific filter Φ_{T_i} implemented as a 1x1 convolution operation is applied to each spatial location to extract tool-conditioned affordance features $\mathbf{a}_{T_i}$ — information relevant only to that specific tool:

$$\mathbf{a}_{T_i} = \Phi_{T_i} \circledast \mathbf{v} \tag{4}$$

From $\mathbf{a}_{T_i}$, a heatmap predictor which consists of convolutions and upsampling operations estimates the heatmap $\mathbf{Q}_{T_i} \in \mathbb{R}^{1\times H\times W}$, with $\mathbf{Q}_{T_i}(x, y)$ whether the spatial location (x, y) suits the specific tool or not. Finally, we obtain the selected tool and the pick location simply by taking an argmax over the collection of heatmaps:

$$P_{initial} = \underset{(x,y,T_i)}{\arg\max}\, \mathbf{Q}_{T_i}(x, y) \tag{5}$$

Note that in the affordance-aware selection module, all the weights are shared except the tool-specific filters Φ_{T_i}'s. This

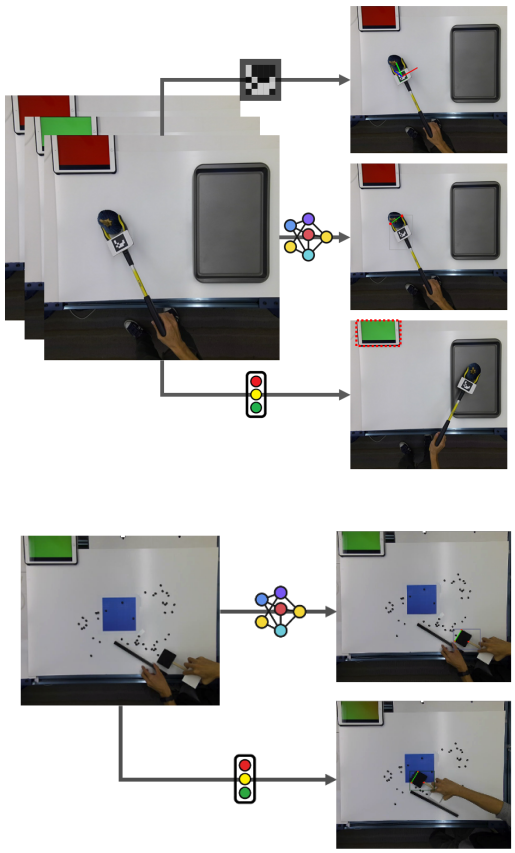

Fig. 3: **Dataset Collection**. Our demo data is collected by recording a grabber hand's actions manipulated by human experts. The attached ARTag is used to get the orientation of the grabber, and a pre-trained detector is used to get the accurate pick and place points in the workspace. Also, a ipad made signal lights is used to label the pick/place key frames we need. We use the pre-trained detector's output keypoints to get orientation and translation for pick and place points

formulation forces all the tool-specific information to be stored in Φ_{T_i}'s, while the rest of the network remains tool-agnostic and shared across all tools. In contrast to learning separate networks from tool-specific data, this inductive bias on the architecture allows the network to maximize information sharing and learn generalizable representations.

b) Generating the final position:: We keep the architecture of our pick-conditioned placing module similar as in [13], to learn the placing locations and rotations, we partially crop the o_t around initial points to get $P_{initial}$, then rotate it with 360/k angle to form k queries. Those queries and o_t are then feed into two different ResNet to get two embeddings then do cross correlation

$$P_{initial} = \arg\max_{(x,y)} Q_{init}((x,y)|o_t, T_t, P_{initial}) \quad (6)$$

$$P_{final} = \arg\max_{(x,y)} Q_{goal}((x,y)|o_t, T_t, P_{final}) \quad (7)$$

We do not use the prediction angle θ on the sweeping task, since given the initial and final points, the angle between positive axis and a vector pointing from start to end will indicate the optimal value, which can be defined as a primitive.

$$P_{initial} = \arg\max_{(x,y)} Q_{initial}((x,y)|o_t, T_t) \quad (8)$$

C. Visual Representations

We use top down view RGB-D images from a single camera as our inputs. The images are preprocessed by first unprojecting to point clouds in 3D space in the camera coordinates and then transforming to the robot base frame. Following [14], we render the points within the workspace through an orthographic projection, which not only keeps the appearance undistorted but also retains spatial meaning for each pixel. In addition, we use simple data augmentations (random transformations on current orthographic projected images) as in [13] for a fair comparison. Our loss functions are defined as follow:

$$L_{initial} = \sum_{O} CE(label_{init}, Q_{init}((x,y)|o_t, T_t, P_{initial})). \quad (9)$$

$$L_{final} = \sum_{O} CE(label_{final}, Q_{final}((x,y)|o_t, T_t, P_{final})). \quad (10)$$

where CE denotes softmax cross entropy loss.

D. Hardware Setup

The hardware setup is shown in Fig 4. We use a 7-DoF Franka Emika robot arm equipped with a parallel-jaw gripper. To enable the robot to automatically switch between tools, we built a tool shelf mounted at the back of the workspace, which includes a vacuum-suction end-effector and a brush end-effector. A control PC is connected to the robot to control the primitive behaviors (robot position and trajectory control during pick and place, and pushing) by generating high-frequency control commands [15]. A Raspberry Pi 4B+ controls the suction gripper. Finally, a Linux system PC is connected to both the control PC and the Raspberry Pi for overall control and synchronization. We obtain RGB-D images (resolution 1280x720) using one low cost Azure Kinect depth camera mounted above the robot, providing a bird's eye view.

V. EXPERIMENTS

Our experiments aim to answer the following questions: (1) Does the proposed affordance based tool-selection part correctly select which tools to use? (2) Does the tool-conditioned policy achieve almost the same performance as task-specific learning policy given limited demonstrations? (3) Whether our method could works well in complex environment with generalization ability to random positions, novel objects, cluttered and unseen scenes? Qualitative results are included in our supplementary video.

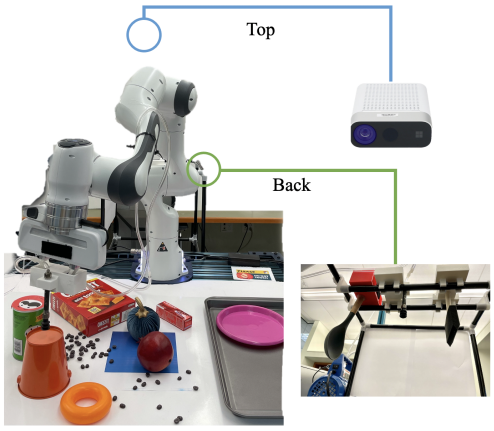

Fig. 4: **Hardware Setup.** We set up one Azure Kinect camera on top of the robot workspace to get RGB-D streams. And we put a tool shelf at the back of Franka arm for placing different tools. For each tools, we attached one 3D printed tool adaptor to help robot pick up tools easily and steadily.

Fig. 5: **The objects used in our experiments.**

A. *Parallel gripper and suction cup P&P (pick-and-place)*

On these two tasks, the robot require to pick up the objects and place them on the plate. 3 shows the objects we use in training and testing. For parallel gripper and suction cup P&P, we test our method and all baselines in three settings with increasing difficulties. First, we test single-object P&P with objects in the training set but with randomized object and container locations. We then replace the training objects with test objects, testing single-object P&P while still randomizing object and container locations. Finally, we select random subsets of test objects and form cluttered scenes with them. testing multi-object P&P. We use the success rate (the ratio of successful P&P attempts to the total number of attempts) as the metric. For each combination of method and setting, we collect results from 100 test trials.

For these tasks, we compare our method with Transporter [13], since it shows strong performance by learning from limited number of demonstrations. However, it requires to train a new model for each single task. As shown in Table I, our method outperform Transporter on parallel gripper for both 20 demos and 50 demos. This suggests that our architecture retains a strong performance, when a single policy is trained. In Table II, our method is even better than baseline since with the tool/task head added, the model could focus on learning to recognize flatten surface instead of learning to map certain shape/color/contour pattern of the object to actions, which will have better generalization ability to novel objects at novel scenes.

B. *Brush Sweeping*

In this task, robot need to use the brush to push all the beans to the target area. There are 40 beans on the table. The color can be black, grey or blue. The shape of the target area is square or round, and the area is 225 square centimeters. For this task, we compare the policy learned from our method against 3 baselines:

- Circular sweeping: a heuristic method that sweeps towards the target area, with starting positions initialized along a 3/4 circle large enough to cover all objects;
- Horizontal + vertical sweeping: a heuristic method that first sweeps all objects horizontally toward the middle (from both sides) and sweeps vertically to bring all objects into the target area;
- Human demonstrations: human testers were asked to sweep greedily only in straight lines, which serves as an upper bound of the performance. The demonstrations we collected are also used for training.

All models are trained with 750 human-demonstrated pushes. We use a simple OpenCV HSV filter to detect the state of the current trial and measure success. This allows us to measure success at different success thresholds. We report the number of pushes performed (the lower the better) before 95% and 100% of the beans are pushed into the target area. The maximum number of attempts is set to 30. Therefore if the robot can't sweep all the beans to the target area within 30 attempts, the number of pushes is 30.

We test our method and all baselines in 3 settings with different settings. First, we test the model with the same beans during training, but in randomized layouts. Second, we test the model on new beans (grey and blue), also in random layouts. Finally, we change the shape of target area from square to round, which is also unseen from the demonstrations. For each method under each setting, we test the model with 20 random scenes. The performance of all methods are shown in Table III.

C. *Household Rearrangement*

Household rearrangement task requires robot to move different objects with a suitable tool. In our settings, the robot need to use one of the three tools: parallel gripper, suction cup and brush to rearrange the objects on the table. During training, we use 41 demos from suction, 56 demos from parallel grasping, 87 demos for mixing parallel grasping and suction grasping, and 246 sweeping demos. To

	50 Gripper Demos			Mixed Demos		
	Random Positions	+Novel Objects	+Cluttered Scenes	Random Positions	+Novel Objects	+Cluttered Scenes
Transporter [13]	0.88	0.84	0.85	-	-	-
Ours	**0.93**	**0.91**	**0.90**	**0.95 (0.95)**	**0.90 (0.86)**	**0.85(0.75)**

TABLE I: SUCCESSFUL RATES OF TASK ON PARALLEL GRIPPER. Mixed demos denote demos contains objects on the left phrase of 5. Numbers outside parentheses denote success rate by using all possible tools for this task. Numbers in parentheses denote task success rate when it is restricted to use parallel gripper.

	20 Suction Demos			Mixed Demos		
	Random Positions	+Novel Objects	+Cluttered Scenes	Random Positions	+Novel Objects	+Cluttered Scenes
Transporter [13]	**0.95**	0.70	0.65	-	-	-
Ours	0.92	**0.89**	**0.83**	**0.93 (0.93)**	**0.79 (0.68)**	**0.77 (0.63)**

TABLE II: SUCCESSFUL RATES OF TASK ON SUCTION CUP. Mixed demos denote demos contains objects on the left phrase of 5. Numbers outside parentheses denote success rate by using all possible tools for this task. Numbers in parentheses denote task success rate when it is restricted to use suction gripper.

	Random Layouts		Novel Objects		Novel Goal	
	95% completed	100% completed	95% completed	100% completed	95% completed	100% completed
Human Demonstrations	8.5 $\pm$ 0.9	10.2 $\pm$ 0.9	8.4 $\pm$ 0.8	10.3 $\pm$ 0.9	8.2 $\pm$ 0.9	10.3 $\pm$ 0.9
Circular	17.4 $\pm$ 0.7	18.5 $\pm$ 0.5	17.5 $\pm$ 0.7	18.5 $\pm$ 0.6	17.2 $\pm$ 0.7	18.4 $\pm$ 0.7
Horizontal + Vertical	19.2 $\pm$ 0.6	19.6 $\pm$ 0.5	19.1 $\pm$ 0.6	19.4 $\pm$ 0.6	19.2 $\pm$ 0.5	19.5 $\pm$ 0.6
Ours	**11.8 $\pm$ 1.3**	**13.7 $\pm$ 1.2**	**11.9 $\pm$ 1.3**	**13.5 $\pm$ 1.3**	**11.7 $\pm$ 1.3**	**13.7 $\pm$ 1.2**

TABLE III: TIMES OF PUSHES OF TASK ON BRUSH

	Pick and Place		Sweep		
	Tool Selection	Execution	Tool Selection	95% completed	100% completed
Ours	**0.94**	**0.91**	**0.98**	**11.8 $\pm$ 1.2**	**13.7 $\pm$ 1.2**

TABLE IV: Performance on Rearrangement Task

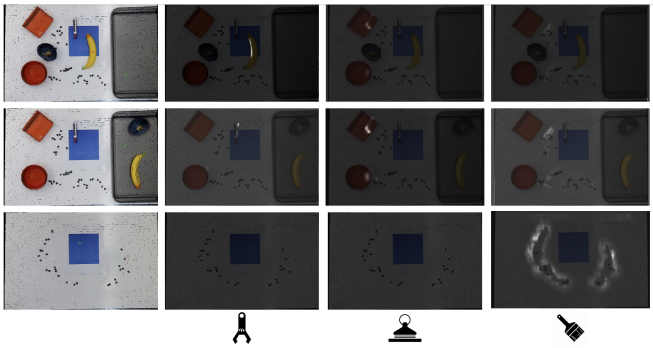

Fig. 6: **Visualizations of heatmaps on different tools**

be more clear, each demo means single step action a_t as we defined III. And there are 10 random objects and 40 black beans on the table. The robot need to move all the objects on the plate, and push all the beans to the blue square area. As we could see from table IV, our method could not only select correct tool with a successful rate greater than 90%, but could jointly learn a good representation and policy as shown in Figure 6. The visualizations for initial and final heatmaps show our model could discrete tool clearly and could generalize to novel object and random unseen scenes.

VI. CONCLUSIONS

We explored how robots can learn to select and use tools. The robot learned from a limited number of human demonstrations and can work well in different settings: different positions, novel objects, different viewpoints, and even cluttered scenes, with improved generalizability and success rate compared with all baselines. This success shows that our design choices, including a tool-selection network and tool-conditioned polices, together with our hardware for switching tools, are promising for further research.

REFERENCES

[1] H. Dang and P. K. Allen, "Semantic grasping: Planning robotic grasps functionally suitable for an object manipulation task," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 1311–1317.

[2] A. Edsinger and C. C. Kemp, "Toward robot learning of tool manipulation from human demonstration," *Citeseer, Tech. Rep.*, 2007.

[3] 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.

[4] D. Gyimothy and A. Toth, "Experimental evaluation of a novel automatic service robot tool changer," in *2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2011, pp. 1046–1051.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[6] Z. Hu, W. Wan, and K. Harada, "Designing a mechanical tool for robots with two-finger parallel grippers," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2981–2988, 2019.

[7] S. McKinley, A. Garg, S. Sen, D. V. Gealy, J. P. McKinley, Y. Jen, M. Guo, D. Boyd, and K. Goldberg, "An interchangeable surgical instrument system with application to supervised automation of multilateral tumor resection," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, 2016, pp. 821–826.

[8] J. K. Mogali, "Heuristics for routing and scheduling of spatio-temporal type problems in industrial environments," Ph.D. dissertation, Carnegie Mellon University, 2021.

[9] 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.

[10] N. Ratliff, J. A. Bagnell, and S. S. Srinivasa, "Imitation learning for locomotion and manipulation," in *2007 7th IEEE-RAS International Conference on Humanoid Robots*, 2007, pp. 392–397.

[11] M. Sieb, Z. Xian, A. Huang, O. Kroemer, and K. Fragkiadaki, "Graph-structured visual imitation," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, L. P. Kaelbling, D. Kragic, and K. Sugiura, Eds., vol. 100. PMLR, 30 Oct–01 Nov 2020, pp. 979–989. [Online]. Available: https://proceedings.mlr.press/v100/sieb20a.html

[12] D. Turpin, L. Wang, S. Tsogkas, S. Dickinson, and A. Garg, "Gift: Generalizable interaction-aware functional tool affordances without labels," *arXiv preprint arXiv:2106.14973*, 2021.

[13] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, and J. Lee, "Transporter networks: Rearranging the visual world for robotic manipulation," *Conference on Robot Learning (CoRL)*, 2020.

[14] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4238–4245.

[15] K. Zhang, M. Sharma, J. Liang, and O. Kroemer, "A modular robotic arm control stack for research: Franka-interface and frankapy," *CoRR*, vol. abs/2011.02398, 2020. [Online]. Available: https://arxiv.org/abs/2011.02398

[16] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5628–5635.

[17] X. Zhang and C. A. Nelson, "Kinematic analysis and optimization of a novel robot for surgical tool manipulation," *Journal of Medical Devices*, vol. 2, no. 2, 2008.