Learning Task-Oriented Dexterous Grasping from Human Knowledge

Hui Li, Yinlong Zhang, Yanan Li, and Hongsheng He*

***Abstract*—Industrial automation requires robot dexterity to automate many processes such as product assembling, packaging, and material handling. The existing robotic systems lack the capability to determining proper grasp strategies in the context of object affordances and task designations. In this paper, a framework of task-oriented dexterous grasping is proposed to learn grasp knowledge from human experience and to deploy the grasp strategies while adapting to grasp context. Grasp topology is defined and grasp strategies are learned from an established dataset for task-oriented dexterous manipulation. To adapt to various grasp context, a reinforcement-learning based grasping policy was implemented to deploy different task-oriented strategies. The performance of the system was evaluated in a simulated grasping environment by using an AR10 anthropomorphic hand installed in a Sawyer robotic arm. The proposed framework achieved a hit rate of 100% for grasp strategies and an overall top-3 match rate of 95.6%. The success rate of grasping was 85.6% during 2700 grasping experiments for manipulation tasks given in natural-language instructions.**

***Index Terms*—Dexterous Grasping, Task-Oriented Grasping, Grasp Topology, Reinforcement Learning**

I. Introduction

Robot dexterity plays an important role in accomplishing complex industrial tasks coping with a variety of objects in many applications, such as aerospace, automotive, manufacturing, warehousing, and distribution [1]–[4]. Industrial robots are consistent and precise, but they are currently confined to structured production lines or simple and repetitive tasks. Robotic dexterity will enable complex tasks in unstructured environments.

Dexterous manipulation is one demanding function in many industrial applications. The existing robotic systems, however, cannot determine proper grasp strategies in this context. Studies have shown that dexterous manipulation is related to object affordances [5]–[7], task designations [8], spatial relationships between objects [9], [10], and scene understanding [11]. Object affordances may include physical features of an object such as dimensions, shape, and mass; task designations could include action, force, constrain, etc; and in this work, fixed spatial relationships and scenes are used. Though there could be many possible ways to pick up an object, research has shown that human grasp topology trends to cluster over a large set of objects. Grasp taxonomy has been defined to simplify grasping choices. For example, a taxonomy of 16 grasping types was created according to grasp strategies used by machinists in manufacturing environments [12]. Another grasp taxonomy with 31 different grasp topology has also been proposed along with the Yale human grasping dataset [13]. Except for grasping topology, little work has been done in achieving task-oriented grasping or leveraging human knowledge in grasping strategies.

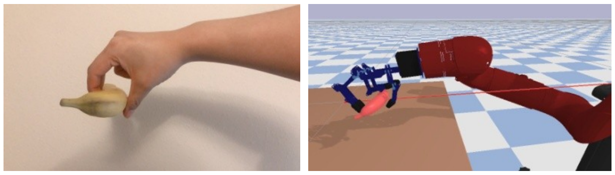

Figure 1. Robot accomplishes task with grasp strategy learned from human knowledge.

In this paper, we propose a task-oriented dexterous grasping approach, which can generate and deploy a proper grasping strategy based on task designation and object affordances. To represent the relation between object affordances, task designations, and human grasping strategies, a task-oriented object grasping dataset was developed, which enables learning grasping strategies of human experience (Fig. 1). We designed a deep learning network to discover the human knowledge of grasping from the dataset. We also implemented a reinforcement learning mechanism to deploy the selected grasping strategies. The workflow of the proposed system can be found in Fig. 2. The learning-based grasping balances grasping flexibility and dexterity by adapting to various grasp contexts. The main contributions of the paper include

- ◇ We achieved the learning of human knowledge in grasping from the developed database; and
- ◇ We demonstrated the feasibility and effectiveness of dexterous grasping by using reinforcement learning while following learned grasping strategies.

Hui Li and Hongsheng He are with the Department of Electrical Engineering and Computer Science, Wichita State University, USA.

Yinlong Zhang is with State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China; zhangyinlong@sia.cn.

Yanan Li is with the Department of Engineering and Design, University of Sussex, UK. His work is funded by UK EPSRC Grant EP/T006951/1.

This work was supported by the John A. See Innovation Award of Wichita State University.

*Correspondence should be addressed to Hongsheng He, hongsheng.he@wichita.edu.

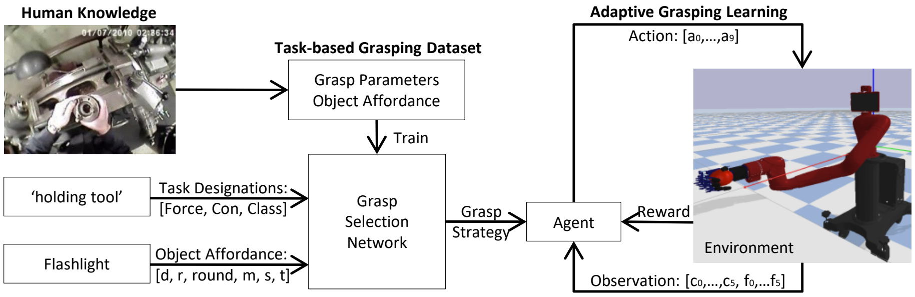

Figure 2. Workflow and structure of the task-oriented grasp strategy generation system. Details of object affordances and task designation can be found in Table I while action and observation are specified in Section 3.

Table I
FEATURES IN THE PROPOSED TASK-ORIENTED OBJECT GRASPING DATASET

	Parameters	Description
Grasp	Grasp type	Grasp choices defined in [14]
	OppoType	Opposition type of a grasp: {Pad, Palm, Side}
	PIP	Power, intermediate or precision
Task	Task names	High-level task description
	Force	Type of forces required : {weight, interaction}
	Constraint	Constraints of the task
	Class	Function class of the task: {hold, feel, use}
Object	Dimensions	Dimensions of object in cm: {A, B, C}
	Rigidity	Rigidity of the object
	Roundness	Dimensions along which object is round
	Mass	Mass of the object in $gram$
	Shape	Basic shape class [15]
	Type	Object type, as defined by [8]
Duration	Duration	Length of the grasp instance

II. Learning Grasp Strategies

Grasping strategies are related to object affordances and task designations [8]. The problem of grasp selection considering grasp context is a challenging problem, due to many factors such as task representation, object property measurement, and the determination of a suitable grasp strategy. To address this challenge, we implemented task-oriented dexterous grasping through learning human knowledge in grasping. We developed and trained a deep learning neural network to predict proper grasp strategies.

A. Representing Human Grasping Knowledge

We established a dataset for task-oriented object grasping based on the Yale human grasping dataset [13]. The Yale human grasping dataset was recorded by two housekeepers and two machinists about their regular work activities including 27.7 hours of tagged video. A spreadsheet with 18210 entries was established based on the videos including task designations, object affordances, and labeled with grasp parameters. Task designations, object affordances, and grasp parameters are defined in Table I.

The grasping tasks in the Yale dataset are redundant and ambiguous in task descriptions. We, therefore, refined the task list to 74 unique tasks by combining the tasks with similar actions. The dataset includes 31 unique types of grasp topology some of which are identical or similar, and the total number of grasp types can be reduced to 15 [16]. The refined dataset includes 15 grasp topology as shown in Fig. 3, 255 unique objects, and 74 unique actions. Some examples of the dataset are shown in Fig. 4. In the figure, we can see that different tasks on the same object may result in different grasp selections. The dataset is available online[1].

Besides, the Yale human grasping dataset does not provide 3D models of objects. We collected 157 3D object models that are similar to the objects in the dataset (Fig. 5).

B. Learning Grasp Strategies

Task designations can be easily extracted from a task description using a natural language processing algorithm named SpaCy. A task is usually a verb while the target object is a noun. Cosine similarity S is computed between the extracted action a and each unique action A_k in the task-oriented object grasping dataset. The action A_k which maximized S will be selected as the action for this task, and other task designations of A_k would also be used for this task. Object affordances can be obtained using our previous work [17], [18].

The task-oriented object grasping dataset is further processed to train the model. The categorical features in the dataset are converted to indicator values which expand the number of features in the dataset to 127. The continuous features such as dimension and mass are normalized to improve the robustness of the network. In real-world scenarios, more than one grasp topology could be used for a certain task. We, therefore, designed a multi-class multi-label grasp selection network to predict grasp topology. The network includes three subnetworks: the grasp topology selection network $g(f)$, the OppoType selection network $o(f)$, and the PIP selection network $p(f)$ where f is the input feature.

◇ The grasp type selection network $g(f)$ includes one input layer with 131 neurons, three hidden layers with 2^{11}, 2^{8}, and 2^{6} neurons, and an output layer with 15

[1] https://github.com/hhelium

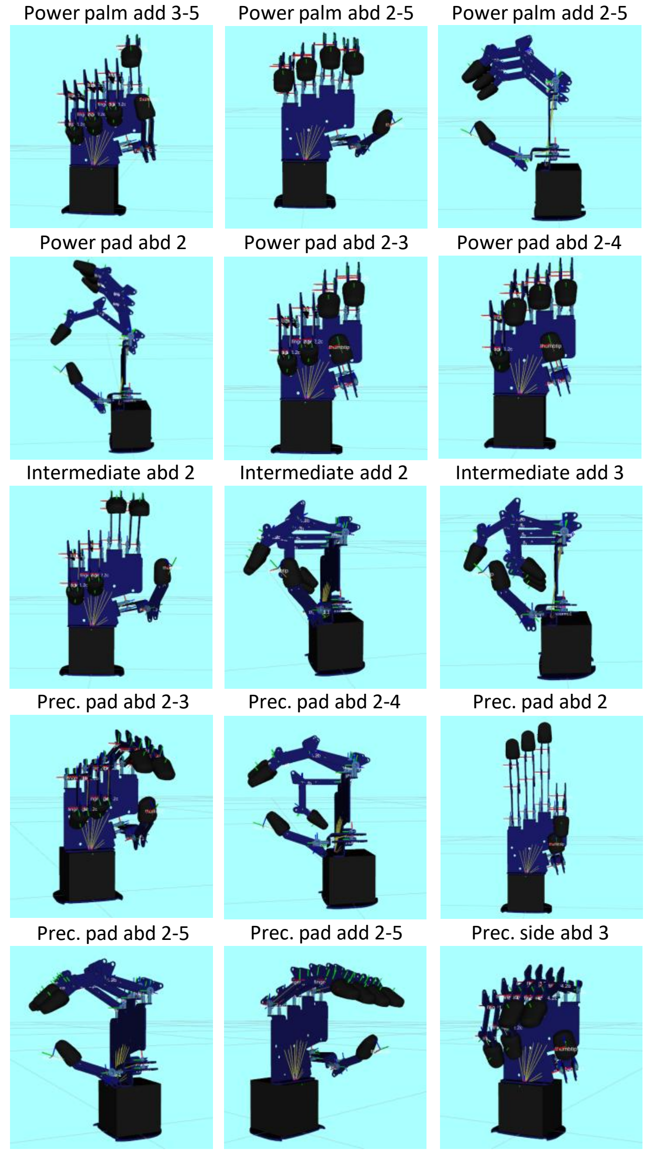
Power palm add 3-5
Power palm abd 2-5
Power palm abd 2-5
Power pad abd 2
Power pad abd 2-3
Power pad abd 2-4
Intermediate abd 2
Intermediate add 2
Intermediate add 3
Prec. pad abd 2-3
Prec. pad abd 2-4
Prec. pad abd 2
Prec. pad abd 2-5
Prec. pad add 2-5
Prec. side abd 3

Figure 3. Defined grasp topology demonstrated in simulation environment.

neurons. All units in the input layers and hidden layers are activated by ReLU while the output layer uses a sigmoid activation function. The output of $g(f)$ is a set of probability distributions over 15 grasp types.

◇ The OppoType selection network $o(f)$ and the PIP selection network $p(f)$ have the same structure. They share the input layer and the first two layers of hidden layers with $g(f)$ and their output layer contains three neurons with a softmax activation function.

The task-oriented object grasping dataset was divided into two parts. The training and validation part contains 90% of grasping samples, and the test data includes the rest 10% (650). The algorithm was trained with 4-fold cross-validation. The hyper-parameters include batch size 64, 5000 training epochs, and learning rate 0.001 using an Adam optimizer. We achieved training accuracy of 85% for $o(f)$ and $p(f)$, and a 100% hit rate for $g(f)$.

III. Adaptive Grasp Deployment

The predicted grasp strategies need to be deployed to grasp objects. Since the grasp size may vary depending on the shape, size, or even rigidity of the object, we trained a deep reinforcement learning network to deploy adaptive grasps for different objects. The traditional grasping task requires complex calculation on path and grasp points, and it is not adaptive enough for task-oriented grasping. In this paper, we achieved adaptive grasping by designing and training a reinforcement learning algorithm that allows the robot to learn to grasp by itself and respond to unforeseen environments. A Proximal Policy Optimization (PPO) reinforcement learning algorithm [19] was implemented using actor-critic policy [20] to teach the robot to grasp and move a target object adaptively. PPO updates its policy using previous experience

$$\theta_{new} = \arg\max_{\theta} E[\min(r_t(\theta)A_t,\, clip(r_t(\theta),\, 1{-}\varepsilon,\, 1{+}\varepsilon)A_t)] \tag{1}$$

where θ is policy parameter, θ_{new} is updated policy parameter, E indicates empirical expectation over time steps, and r_t is the ratio between the new policy and the old policy. The term A_t is an estimator of the advantage function at time t, ε is a small number usually 0.1 or 0.2, and $clip$ will clip A_t while r_t is not between $1-\varepsilon$ and $1+\varepsilon$.

Since we focus on adaptive grasp deployment, fixed grasp paths and object locations are used in this work. Each joint in the hand is controlled by a scaler $a_i \in [0, 1]$. The joint will fully extend when $a_i = 0$ and completely close while $a_i = 1$. The AR10 hand has 10 joints, and the action function is $\mathrm{a} = [a_0,\, a_1,\, a_2,\, a_3,\, a_4,\, a_5,\, a_6,\, a_7,\, a_8,\, a_9]$. The algorithm will control the hand to grasp the object using the action function based on the configuration of the selected grasp topology. In this way, the robot is able to adjust the grasp size while keeping the configuration of the selected grasp topology.

We modeled the hand into six parts: palm, index finger, middle finger, ring finger, little finger and thumb. The observation of the algorithm is a 12 dimensions vector $\mathrm{O_b} = [c_0,\, c_1,\, c_2,\, c_3,\, c_4,\, c_5,\, f_0,\, f_1,\, f_2,\, f_3,\, f_4,\, f_5]$ where c_i is the closest distance between each part of the hand and the target object, and f_i indicates norm force provided by each part of the hand.

A steady grasp prefers more contact points, also the contact force should not be too large or it would either break the object or waste energy. To achieve optimized grasps, we define the reward function as

$$R = \begin{cases} -1000 & n = 0 \\ \beta_1 n - \beta_2 f & n > 0 \end{cases} \tag{2}$$

where n is the number of contact points on the target object, f is the total force, and β is the scaler to control the weight of n and f. If a grasp is a success, $R = R + 1000$. In this

Object	A	B	C	Rigidity	Roundness	Mass	Shape	Type	Duration	Task	Task_group	OppoType	PIP	Grasp_group
aerosol can cap	6	6	5	squeezable	c	16	equant	cubic cylinder	5	arranging object	arranging	Pad	Precision	Precision pad abd 2-3
aerosol can cap	6	6	5	squeezable	c	16	equant	cubic cylinder	4	securing cap	securing	Pad	Power	Power pad abd 2-3
air hose	15	3.25	1.75	rigid	a	231.5	bladed	ellipse A	11.9	blowing away chips	blowing	Palm	Power	Power palm abd 2-5
air hose	15	3.25	1.75	rigid	a	231.5	bladed	ellipse A	2.1	cleaning off part	cleaning	Palm	Power	Power palm abd 2-5
flashlight	15	2	2	rigid	a	215	prolate	cylinder	20.4	moving object	moving	Side	Intermediate	Intermediate add 3
flashlight	15	2	2	rigid	a	215	prolate	cylinder	1.64	holding tool	holding	Pad	Precision	Precision pad abd 2-3
hammer	15	2	2	rigid	a	400	prolate	cylinder	0.36	holding tool	holding	Side	Intermediate	Intermediate add 3
hammer	15	2	2	rigid	a	400	prolate	cylinder	2.04	pounding hammer	pounding	Palm	Power	Power palm add 2-5
metal Block	12.5	8.5	1	rigid	non-round	135	oblate	short prism	1.6	moving object	moving	Palm	Power	Power palm abd 2-5
paper towel roll	15	9	9	squeezable	a	200	prolate	cylinder	2	arranging object	arranging	Pad	Precision	Precision pad abd 2-5
part on machine	15	9	9	rigid	a	1000	prolate	cylinder	2	holding part	holding	Palm	Power	Power palm abd 2-5
remote	15	6	2.5	rigid	non-round	165	bladed	irregular	1.16	arranging object	arranging	Pad	Power	Power pad abd 2
remote	15	6	2.5	rigid	non-round	165	bladed	irregular	1.8	moving object	moving	Pad	Precision	Precision pad abd 2-5
rod	15	1.25	1.25	rigid	a	300	prolate	cylinder	3	adjusting rod	adjusting	Pad	Precision	Precision pad abd 2
rod	15	1.25	1.25	rigid	a	300	prolate	cylinder	3.44	holding part	holding	Palm	Power	Power palm add 2-5
ruler	15	2.25	0.25	rigid	non-round	32.5	bladed	irregular	17.04	holding tool	holding	Side	Intermediate	Intermediate add 3
ruler	15	2.25	0.25	rigid	non-round	32.5	bladed	irregular	0.4	pinching tool	pinching	Pad	Precision	Precision pad abd 2
screwdriver	15	3.5	3.5	rigid	a	100	prolate	cylinder	7.17	turning handle	turning	Pad	Precision	Precision pad abd 2-3
screwdriver	15	3.5	3.5	rigid	a	100	prolate	cylinder	1.88	holding tool	holding	Side	Intermediate	Intermediate add 3
small knob	4.5	2.5	2.5	rigid	a	150	prolate	cylinder	0.04	opening cabinet	opening	Side	Intermediate	Intermediate add 2
small knob	4.5	2.5	2.5	rigid	a	150	prolate	cylinder	10	turning knob	turning	Pad	Precision	Precision pad abd 2-3
sponge	10	6	2.5	squeezable	non-round	17.5	bladed	irregular	1	wiping counter	wiping	Pad	Precision	Precision pad abd 2
sponge	10	6	2.5	squeezable	non-round	17.5	bladed	irregular	3.9	moving object	moving	Palm	Power	Power palm add 2-5
spray bottle	15	6	3.5	squeezable	non-round	400	bladed	irregular	2	picking up	picking up	Pad	Precision	Precision pad abd 2-4
spray bottle	15	6	3.5	squeezable	non-round	400	bladed	irregular	25	holding	holding	Palm	Power	Power palm abd 2-5
thin rod	11.5	0.65	0.65	rigid	a	105	prolate	cylinder	20.36	applying gel to part	applying	Side	Intermediate	Intermediate add 3
thin rod	11.5	0.65	0.65	rigid	a	105	prolate	cylinder	8.76	turning tool	turning	Pad	Precision	Precision pad abd 2-3
wrench	15	2	0.75	rigid	non-round	250	bladed	irregular	2	holding tool	holding	Pad	Precision	Precision pad abd 2-3
wrench	15	2	0.75	rigid	non-round	250	bladed	irregular	10	cranking wrench	cranking	Palm	Power	Power palm add 2-5

Figure 4. Selected data samples in the task-object grasp dataset. The objects in the table are the same objects used in Fig. 9.

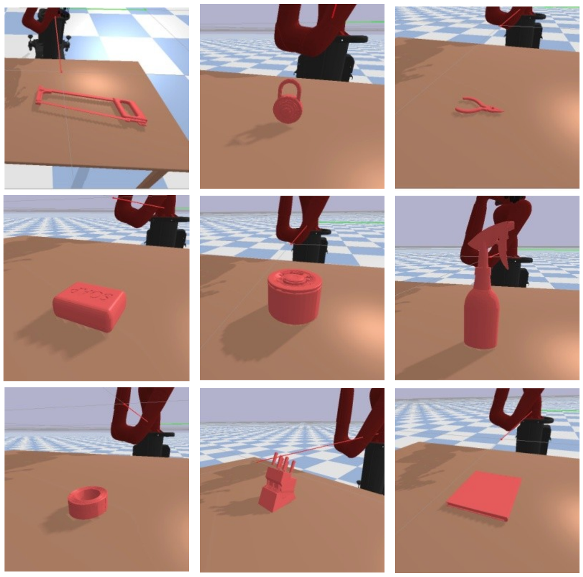

Figure 5. Samples of 3D objects in the 3D object dataset.

work, we define a grasp as a success when the hand is able to grasp the target object, move the object above a certain height, and hold it for a certain amount of time. This reward function would reward more contact points and punish large contact forces.

A total of 15 predefined grasp topology was developed to grasp and move an object in the simulation environment. A trial is terminated and reset whenever it achieves a success grasp or it runs out of time.

The training set contains 537 data samples. We also added a small noise (less than 5 mm) to the object position to improve the adaptivity of the algorithm and extended the size of the training set to 5370. The network is trained with learning rate $2.5e^{-4}$ and discount factor 0.9. The accuracy on the training objects achieved 88.2%.

IV. Experiments

We conducted experiments to evaluate the proposed algorithm and to demonstrate its performance in a simulation environment.

A. Experiment Setup

A simulation environment as shown in Fig. 6 was developed to evaluate the performance of the proposed algorithm. The environment includes a robot and a workspace. The robot contains an AR10 robotic hand and a Sawyer robotic arm, which are attached together to deploy manipulation tasks. A table is placed in front of the robot as a workspace and objects are placed at a fixed location on the table. The environment is developed with Pybullet using Python 3.5 in Ubuntu 16.04.

About 10% (650) of the data from the task-oriented object grasping dataset was selected and used for testing. A random noise that follows the standard deviation was added to the continuous features of the remained data, and the size of the training set was increased to 136627.

Not all of the 157 3D object models were used to train/test the reinforcement learning network. Some of the 3D objects are either too large for the workspace or too small to be grasped by the robotic hand due to physical limitations of the hand. A number of 103 3D objects were used in this work, where 86 objects were used for training the adaptive grasp

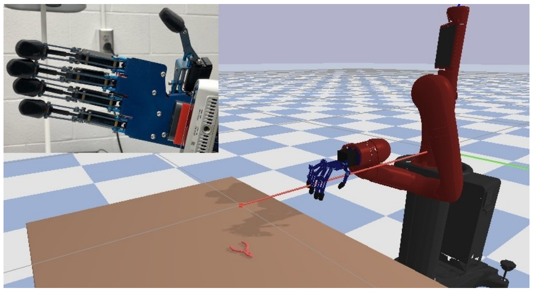

Figure 6. A simulation environment that allows complex grasping tasks.

deployment network and 17 objects were used for testing the algorithm.

B. Grasp Strategy Learning

The proposed grasp selection network was tested with 650 data samples that are completely new to the algorithm. The result of the grasp selection network achieved a hit rate of 100% and a top-3 match rate of 98.6%. Since the grasp selection network is a multi-label model, we defined the hit rate as the probability that the top-1 predict label is in the target label set. The top-3 match rate indicates how often the top-3 predictions are in the targets. The ROC curve of each class can be found in Fig. 7. From the figure, we can see that the AUCs for all classes are close to 1, demonstrating the performance of the grasp selection network. The testing accuracy was a little bit higher than the training accuracy. Two reasons may lead to this result. Firstly, the training set has been augmented, so that it is large enough to cover most object-task-grasp combinations. Secondly, there were only 650 testing samples in this experiment, and the accuracy may decrease as the number of testing samples increases.

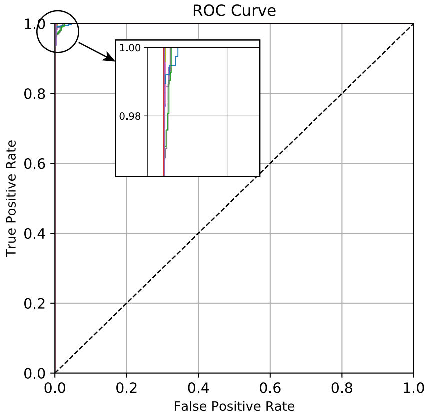

Figure 7. ROC curves for all 15 classes of the grasp selection network.

The accuracy of the OppoType selection network and the PIP selection network was 85.9% and 86.3% (Fig. 8) which were lower than the accuracy of the grasp selection network. Since the OppoType and the PIP selections are not designed as multi-label models, there is only one target label for each data sample. This leads to lower accuracy on the OppoType and PIP prediction. Although each grasp type is directly related to an OppoType and a PIP, the experiment results show that the proposed network is able to predict all grasp parameters with high accuracy.

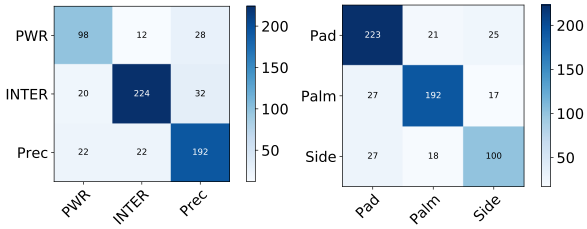

Figure 8. Confusion matrices of the PIP and OppoType selection network. PWR, INTER and Prec are short for Power, Intermediate and Precision.

C. Simulated Grasping of Arbitrary Objects

Since the AR10 hand only has 10 DOFs and can not perform some of the predefined grasp topology, a total of nine predefined grasp topology was employed in the simulations. Each grasp topology was tested with three new objects, and each object was grasped 100 times.

A total of 17 objects and 2700 grasping experiments were performed to evaluate the adaptive grasping learning system as shown in Fig. 9, and the corresponding results can be found in Table II. In the experiment, if the robotic hand successfully grasped a target object, picked it up for 20 cm, and held the object for 20 time steps (0.2 seconds) without dropping it, the grasp is defined as a success. A number of 2312 successes were achieved, and the final success rate was 85.6%.

Table II
RESULTS FOR SIMULATED GRASPING IN FIG. 9

Grasp Topology	Grasping Results		
Power pad abd 2	94	93	85
Power pad abd 2-4	91	82	75
Power palm abd 2-5	89	83	78
Power palm add 2-5	82	98	77
Prec. pad abd 2	85	84	93
Prec. pad abd 2-3	95	74	96
Prec. pad abd 2-4	91	75	78
Prec. pad abd 2-5	89	98	76
Prec. pad add 2-5	82	78	91

Three reasons may lead to these results. The first one is that the testing data was completely new to the algorithm, and some features might not be covered by the training set. Secondly, the robot hand is not dexterous enough for certain objects. For example, the success rates on the wrench were low for most of the grasp topology, since the wrench was very thin and it was hard for the robot to grasp due to physical limitations. Finally, the 3D object models are not perfect and

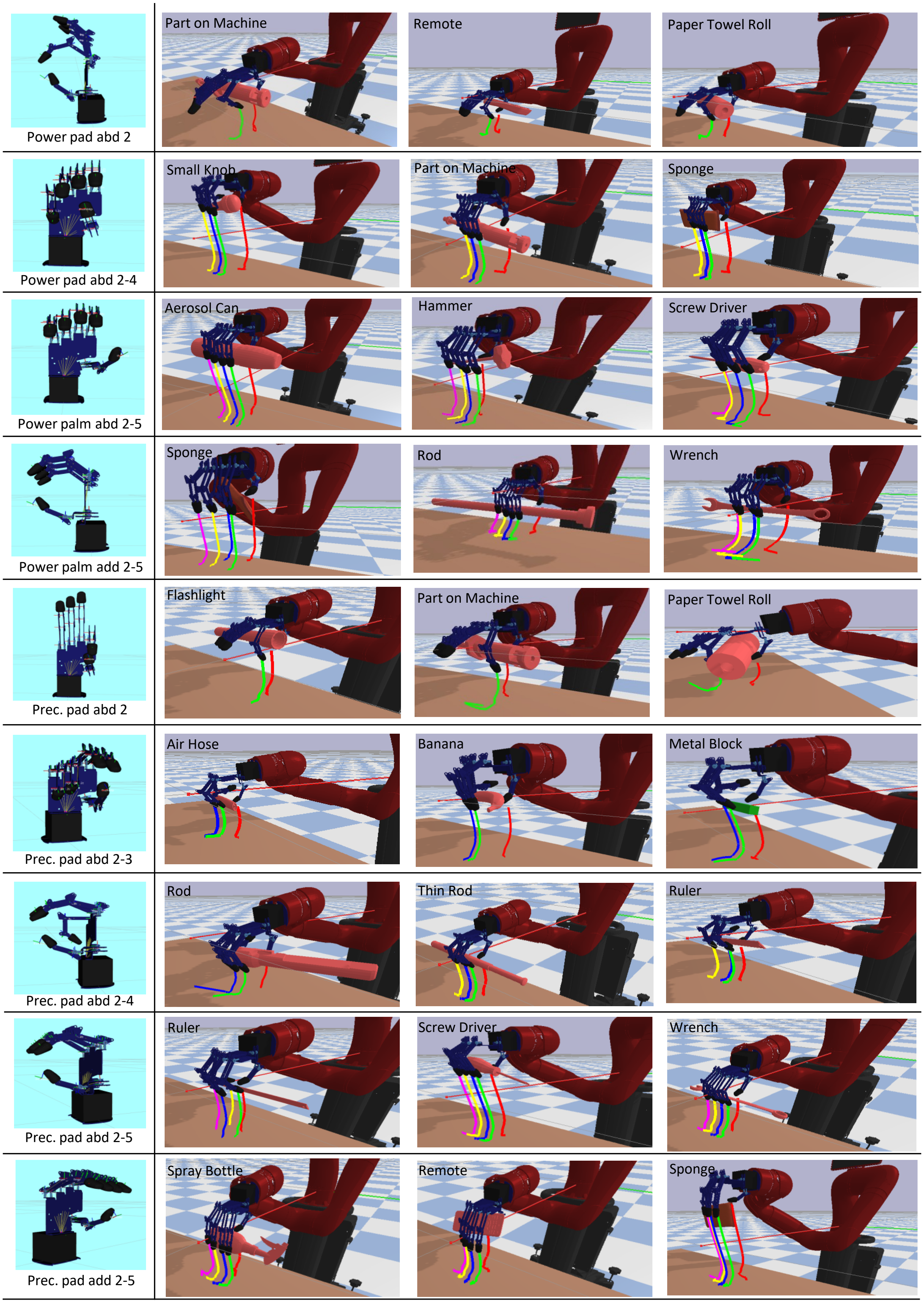

Figure 9. Grasping simulations on 17 arbitrary objects. Colored lines indicate trajectories of fingertips while grasping.

some of the models are not accurate enough. For example, for the banana shown in Fig. 9, we can see that the shape collision model of the banana is not the same as its visual model.

V. Conclusions

We presented a grasp learning algorithm that is able to learn grasp topology from human knowledge and deploy the grasp adaptively. A multi-output multi-label neural network was developed to predict grasp topology based on task designations and object affordances, and the top-1 accuracy for this network achieved 100%. A Proximal Policy Optimization reinforcement learning algorithm was implemented and trained for adaptive grasp learning. A total of 2700 grasping experiments were performed on 17 different objects and the overall accuracy achieved 85.6%. The performance of the system was demonstrated by the experiment results. Compared to deciding grasp topology for a task, choosing a proper grasp location based on a specific grasp pose, and approaching the hand to a suitable position and orientation to that location are even more challenging. Our further work will focus on those two problems, and the final goal is to develop a fully automated dexterous grasping system for both social and manufacturing environments.

References

[1] K. Zhou, G. Ebenhofer, C. Eitzinger, U. Zimmermann, C. Walter, J. Saenz, L. P. Castaño, M. A. F. Hernández, and J. N. Oriol, "Mobile manipulator is coming to aerospace manufacturing industry," in *2014 IEEE International Symposium on Robotic and Sensors Environments (ROSE) Proceedings*. IEEE, 2014, pp. 94–99.

[2] P. Božek, "Robot path optimization for spot welding applications in automotive industry," *Tehnicki Vjesnik–Technical Gazette*, vol. 20, no. 5, pp. 913–917, 2013.

[3] S. Mitsi, K.-D. Bouzakis, G. Mansour, D. Sagris, and G. Maliaris, "Off-line programming of an industrial robot for manufacturing," *The International Journal of Advanced Manufacturing Technology*, vol. 26, no. 3, pp. 262–267, 2005.

[4] S. Torres-Mendez and A. Khajepour, "Design optimization of a warehousing cable-based robot," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 46360. American Society of Mechanical Engineers, 2014, p. V05AT08A091.

[5] H. Li, Y. Yihun, and H. He, "Magichand: In-hand perception of object characteristics for dexterous manipulation," in *International Conference on Social Robotics*. Springer, 2018, pp. 523–532.

[6] A. B. Rao, H. Li, and H. He, "Object recall from natural-language descriptions for autonomous robotic grasping," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2019, pp. 1368–1373.

[7] F. Yan, D. M. Tran, and H. He, "Robotic understanding of object semantics by referringto a dictionary," *International Journal of Social Robotics*, vol. 12, no. 6, pp. 1251–1263, 2020.

[8] T. Feix, I. M. Bullock, and A. M. Dollar, "Analysis of human grasping behavior: Correlating tasks, objects and grasps," *IEEE transactions on haptics*, vol. 7, no. 4, pp. 430–441, 2014.

[9] F. Yan, D. Wang, and H. He, "Robotic understanding of spatial relationships using neural-logic learning."

[10] D. Wilson, F. Yan, K. Sinha, and H. He, "Robotic understanding of scene contents and spatial constraints," in *International Conference on Social Robotics*. Springer, 2018, pp. 93–102.

[11] F. Yan, S. Nannapaneni, and H. He, "Robotic scene understanding by using a dictionary," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2019, pp. 895–900.

[12] M. R. Cutkosky *et al.*, "On grasp choice, grasp models, and the design of hands for manufacturing tasks." *IEEE Transactions on robotics and automation*, vol. 5, no. 3, pp. 269–279, 1989.

[13] I. M. Bullock, T. Feix, and A. M. Dollar, "The yale human grasping dataset: Grasp, object, and task data in household and machine shop environments," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 251–255, 2015.

[14] T. Feix, J. Romero, H.-B. Schmiedmayer, A. M. Dollar, and D. Kragic, "The grasp taxonomy of human grasp types," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 66–77, 2016.

[15] T. Zingg, "Beitrag zur schotteranalyse," Ph.D. dissertation, ETH Zurich, 1935.

[16] T. Feix, R. Pawlik, H.-B. Schmiedmayer, J. Romero, and D. Kragic, "A comprehensive grasp taxonomy," in *Robotics, science and systems: workshop on understanding the human hand for advancing robotic manipulation*, vol. 2, no. 2.3. Seattle, WA, USA;, 2009, pp. 2–3.

[17] B. Rao, "Learning robotic grasping strategy based on natural-language object descriptions," 2018.

[18] H. Li, J. Tan, and H. He, "Magichand: Context-aware dexterous grasping using an anthropomorphic robotic hand," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9895–9901.

[19] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[20] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in neural information processing systems*, 2000, pp. 1008–1014.