A Hybrid 2-Stage Method for Robotic Planar Pushing

ZiYan Gao, Armagan Elibol, and Nak Young Chong

***Abstract*— Robotic manipulation has been applied to a particular setup and a limited number of known objects. In order to cope with these limitations, robots need to be capable of manipulating novel objects. In this work, we proposed a computationally efficient 2-stage framework for planar pushing, allowing a robot to push novel objects to a specified pose with a minimum number of steps. We developed three modules: Coarse Action Predictor (CAP), Forward Dynamic Estimator (FDE), and Physical Property Estimator (PPE). CAP predicts a mixture of Gaussian distribution of actions. FPE learns the causality between action and successive object state. PPE based on Recurrent Neural Network predicts the physical center of mass (PCOM) from the visual center of mass (VCOM) and robot-object interaction. Our preliminary experiments show the promising results to meet the required capability of pushing novel objects.**

I. INTRODUCTION

Inspired by the recent work done by [1] and [2], in this work, we proposed a hybrid 2-stage robotic pushing framework which can reduce the computational time significantly. In the first stage as shown in Fig.1, a greedy planer is used to minimize the relative position error. In the second stage shown in Fig. 2, the relative orientation error is paid more attention. Also instead of random sampling in action space, we sample actions from a mixed Gaussian distribution given by a probabilistic model (CAP). Then we developed a forward model FDE to predict the future effort caused by these action candidates. FDE is a simple yet efficient plane neural network that utilizes PCOMs, VCOMs of the current object masks and executed action to predict the future state of the object. PPE is used to estimate PCOM throughout two stages. During the evaluation, we used a greedy planner to minimize the position error, and then used the second stage models to adjust the orientation. Firstly, we generate a sub-goal image by using the same method mentioned in [1]. Then hundreds of action candidates are sampled by CAP based on the current and sub-goal image and PCOM predicted by PPE in the previous time step. Afterward, action candidates were evaluated by FDE, and the best one, which has a minimum mean square error with the required change, was selected. Finally, action, current, and sub-goal images are sent to PPE to estimate the next PCOM. We performed extensive simulations to evaluate our approach quantitatively. The results show that the proposed framework is effective at pushing tasks with relatively small computation effort.

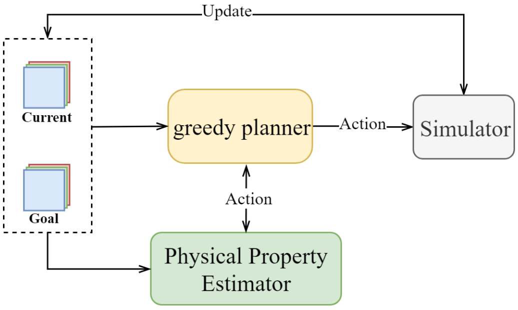

Fig. 1. First stage with greedy planner utilizing VCOMs of current object and target object in the image plane to minimize relative position error. PPE, a recurrent neural network, receives successive images, VCOM and action to update its own internal cell state and predicting PCOM.

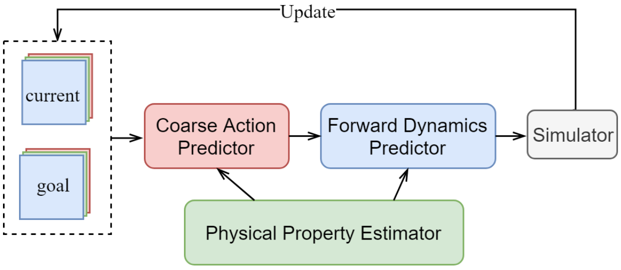

Fig. 2. Second stage with 3 modules to adjust both relative orientation error and relative position error. PPE offers PCOM information to CAP and FDE for better prediction.

II. PROPOSED METHOD

An action is defined by

$$a = \{x_s, y_s, x_e, y_e\}, \tag{1}$$

where X_s and Y_s represent the initial horizontal and vertical coordinates, while X_e and Y_e represent the final ones, respectively. The illustration of the proposed CAP is given in Fig. 3. The inputs are two mask images (M_t, M_{t+1}) and the current PCOM which is estimated by PPE. We used residual blocks trained on ImageNet as a feature extraction module. The mixture distribution is computed by the following equation:

$$P(a_t|M_t, M_{t+1}, VCOM_t) = \sum_{c=1}^{C} \alpha_c \mathcal{D}(\mu, \sigma), \tag{2}$$

where c denotes the index of the corresponding mixture component. There are up to C mixture components, α_c depends on the input and the sum of all α_c is one. $\mathcal{D}$ denotes the distribution to be mixed. In this work, we used Gaussian distribution determined by μ and σ.

FDE is a plane neural network which has three layers of size 64, 64, and 3. The input of FDE is an 11-dimensional

Authors are with the School of Information Science, Japan Advanced Institute of Science and Technology, Nomi, Japan {s1920013,aelibol,nakyoung}@jaist.ac.jp

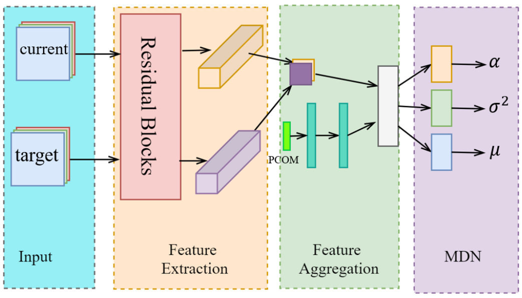

Fig. 3. Coarse action predictor overview

feature vector, which consists of the end position of action, contact point, the norm vector at the contact point, VCOM and PCOM of the current object state and area of the shape. The output is $[x_{t1}, y_{t1}, o_{t1}]$, where x_{t1}, y_{t1} are the change of object position and o_{t1} is the change of object orientation. PPE is developed by a recurrent neural network that has 2 LSTM layers of size 64 each and 2 plane layers of size 32 and 2. PPE is running both in the first and second stage; in the first stage, PPE updates its LSTM ceil and hidden state from robot action and the differences successive object masks. Meanwhile, it helps greedy planar translate object efficiently by offering PCOM. In the second stage, PPE keeps updating its state and broadcasts PCOM to CAP and FPE.

III. Experiment and Result

The dataset is collected in simulation. A cylinder of radius 0.5 cm and length 20 cm is attached to the end link of the UR10 robot. The dataset contains more than 57,000 sequences of interactions between the robot and different objects. The lengths of these sequences are ranging from 3 to 8. There were nine objects(illustrated in Fig.4) of different shapes and sizes considered in the experiment. PCOM is aligned to the object randomly within the bounding box of the objects. Actions are selected randomly but only guarantee that each action changes the object pose. Each action pushes 2.5 centimeters forward. In order to improve the generalization of the model, we randomly change the size and the ratio of the height, width, and length of the object. To re-use residual network layers, we tiled mask images to 3 channels. In the training phase of CAP, we used the negative log-likelihood function to minimize training error.

$$L = -log(P(a|M_t, M_{t+1}, PCOM_t)) \tag{3}$$

We used mean square error to minimize the training error of FDE. For PPE, we used weighted mean square error. The intuition behind this loss function is that the accuracy of estimating PCOM in the current step should be higher than the previous step.

$$L = \frac{1}{T}\sum_{t=1}^{T}\alpha_t(\widehat{pcom}_t - pcom_t)^2, \alpha_t = \frac{t}{\sum_{t=1}^{T}} \tag{4}$$

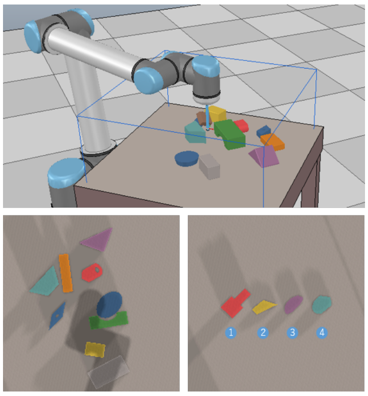

Fig. 4. Simulation Environment: nine objects in data collection phase (left lower) and four novel objects (right lower) in evaluation phase

ids	Rotation Effort				Translation Effort				acc
	mean	std	max	min	mean	std	max	min	
1	1.19	0.69	6.57	0.54	0.56	0.47	3.00	0.22	0.97
2	1.45	1.46	15.13	0.62	0.70	1.02	6.90	0.21	0.93
3	1.25	0.79	7.16	0.37	0.73	0.87	4.59	0.22	1.0
4	1.11	0.75	5.06	0.49	0.59	0.63	3.58	0.15	0.99

For training all modules, we used Adam optimizer and set the size of mini-batch 64. The learning rate was set to 0.001 for CAP and PPE and 0.0001 for FDE. In the whole training phase, the parameters of residual blocks were fixed. For evaluating our method, we used four novel objects. For each object, both target pose and initial pose were randomly initialized. Compared with [1], where the relative orientation errors were initially set within $[-90, 90]$ and the target positions of the object were fixed, our evaluation is more challenging. We repeated 100 times for each novel object. In the first stage, the object will be pushed to the goal region,and the total number of steps needed in this stage is not involved in calculations. The second stage will be executed at most 26 steps. The goal region is $\pm 5cm$ for re-positioning and $\pm 10°$ for re-orienting, which is the same as [1]. We proposed 2 metrics to evaluate predicted actions: **rotation effort** and **translation effort**. Rotation effort depicts how many steps are needed to rotate the object for 10 degrees, and translation effort means how many steps are needed to translate object for 1 cm. The result is given in the table.

IV. Conclusion

In this study, we proposed a hybrid 2-stage framework for robot planar pushing. It was evaluated quantitatively through extensive simulations. The result shows that our method can manipulate novel objects with unknown physical properties. Compared with previous work, our method is efficient in sampling and can handle a more challenging task. In the future, this method will be evaluated further, with more sophisticated FDE and PPE.

References

[1] Juekun Li, Wee Sun Lee, and David Hsu. Push-Net: Deep Planar Pushing for Objects with Unknown Physical Properties. 2018.

[2] Zhenjia Xu, Jiajun Wu, Andy Zeng, Joshua Tenenbaum, and Shuran Song. DensePhysNet: Learning Dense Physical Object Representations Via Multi-Step Dynamic Interactions. 2019.