2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

UniDoorManip: Learning Universal Door Manipulation Policy Over Large-scale and Diverse Door Manipulation Environments

Yu Li *1 Xiaojie Zhang *1 Ruihai Wu*2 Zilong Zhang 1 Yiran Geng 2 Hao Dong†2 Zhaofeng He†1
1Beijing University of Posts and Telecommunications 2School of CS, Peking University

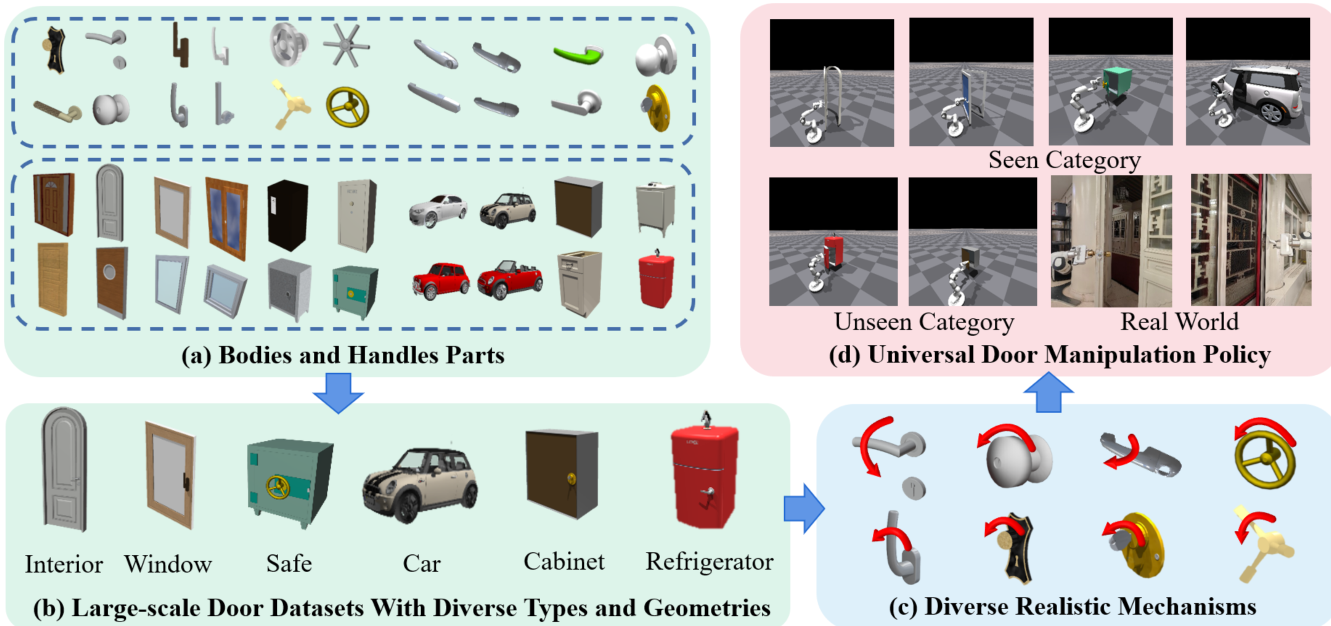


Fig. 1: **Our Proposed Environment, Dataset and Universal Manipulation Policy.** We build a novel door manipulation environment equipped with a large-scale door dataset covering 6 door categories with hundreds of door bodies and handles, and configure it with different realistic door manipulation mechanisms. To learn the universal door manipulation policy, we propose a novel framework which can generalize to unseen shapes and categories.

***Abstract*— Learning a universal manipulation policy encompassing doors with diverse categories, geometries and mechanisms, is crucial for future embodied agents to effectively work in complex and broad real-world scenarios. Due to the limited datasets and unrealistic simulation environments, previous studies fail to achieve good performance across various doors. In this work, we build a novel door manipulation environment reflecting different realistic door manipulation mechanisms, and further equip this environment with a large-scale door dataset covering 6 door categories with hundreds of door bodies and handles, making up thousands of different door instances. To learn a universal policy over diverse doors, we propose a novel framework disentangling the whole manipulation process into three stages, and integrating them through conditional training. Extensive experiments validate the effectiveness of our designs and demonstrate our framework's strong performance in simulation and real world. Code, data and videos are available on https://unidoormanip.github.io/.**

I. INTRODUCTION

To enable robots to exhibit human-like abilities in performing a wide range of tasks, it is crucial for them to acquire proficiency in manipulating articulated objects. Among these tasks, door manipulation holds significant importance due to the frequent need to open or close doors in various scenarios. While previous works have focused primarily on interior doors [1], [2], we aim to extend doors to a more general setting, *e.g.*, doors in windows, cars, safes, as illustrated in Fig.1 (b). In the above broad scenarios, the door manipulation task covers doors with diverse types, geometries and manipulation mechanisms, which poses a great challenge to learn a universal door manipulation policy.

Prior works in the field have struggled to learn the universal manipulation policy due to the lack of diversity in terms of types, geometries, and manipulation mechanisms. Besides using push or pull to open the door[3], [4], [5], [6], [7], [8], DoorGym [1] presents an approach that automatically generates door bodies and handles with hard-code, which results in bad performance when faced with new unseen doors due to the limited geometric diversity. To facilitate the learning of a universal door manipulation policy, we build a comprehensive door manipulation environment that encompasses doors across diverse types, geometries, and manipulation mechanisms. Our efforts in constructing this environment have been focused on the following two key aspects. Firstly, recognizing the limited door types, geome-

*Equal contribution.
†Corresponding author.

tries and quantities in existing datasets [6], [9], [10], [1], we propose a large-scale door dataset with diverse categories and geometries. Our dataset consists of two door components: body and handle, providing users the flexibility to configure doors according to their specific requirements, as illustrated in Fig.1 (a). Tab.I shows that our dataset encompasses 6 distinct types of doors, comprising 328 door bodies and 204 handles, enabling the composition of thousands of door objects while ensuring their compatibility. Secondly, to mitigate the gap between simulation and the real world, we introduce more realistic settings in our door manipulation environment, especially on the design of diverse manipulation mechanisms as shown in Fig.1 (c).

To manipulate doors with latching mechanisms, previous works [1], [7] have explored the training of a single universal policy using reinforcement learning (RL) algorithms. However, directly training such a policy for the entire door manipulation process in an end-to-end manner poses great challenges. This is because door manipulation contains three separate but related stages: handle grasping, handle manipulation, and door opening. The inherent separation of these stages results in complicated manipulation mechanisms and vast exploration space, making it difficult for a single RL policy to learn. To tackle this challenge, we propose a novel framework that disentangles these three stages, each stage contains a specific manipulation process, making it easier to learn a generalizable manipulation policy. Besides, we train these policies in a conditional way to reveal their interrelations and dependencies.

In the three-stages manipulation, the first stage, handle grasping, only requires the grasp pose of the end-effector, while the latter two stages require action sequence generation. Hence we employ a policy specifically for predicting handle grasping action. Leveraging the inherent generalizability across diverse geometries provided by visual affordance [11], the policy predicts a point-level score map where a higher score indicates a greater chance of successful door manipulation, to propose the grasp action. After grasping the handle steadily, the goal of the following stages is to manipulate the handle to unlock the door and open the door. Although handle manipulation and door opening share the similarity of generating action sequences, their specific action types differ significantly. Hence we train separate policies for each stage in a close-loop manner to alleviate the burden of the model, enabling a universal capability for handle manipulation and door opening. Furthermore, to seamlessly integrate the three separate but related universal policies for each stage into a single universal policy for the entire manipulation process, we introduce a conditional training strategy that bridges the gap among policies.

Through extensive experiments in simulation, we validate the effectiveness of our design choices and demonstrate that our approach significantly outperforms previous methods. Additionally, we conduct real-world experiments to show the generalization capability of our approach in the real world.

In summary, our main contribution encompasses:

- We are the first to build a door manipulation environment with diverse realistic mechanisms and a large-scale dataset covering diverse doors, handles, and geometries.
- To achieve universal and realistic door manipulation, we propose a novel framework that disentangles the whole manipulation process into three stages with respective universal policies and integrates them into the whole universal policy leveraging conditional training.
- Extensive experiments validate the effectiveness of our designs and demonstrate our framework's strong performance in simulation and the real world.

II. Related Work

A. Door Manipulation Environment and Datasets

Building an environment that simulates the real world and transferring the policy in simulation to the real world have been the main approach for door manipulation in recent years. However, recent works covering door manipulation have mainly two drawbacks: 1) Unrealistic simulation. Besides using push or pull to open the door[3], [4], [5], [6], [7], [8], Robosuite [12] benchmarks opening doors with latching mechanism as standardized task but employs small doors with large handles. 2) Lack of diversity in the dataset. PartNet-Mobility [10] and AKB-48 [9] provide a diverse dataset for articulated objects including doors. Focusing on the cross-category diversity, they ignore the intra-category diversity of doors. To address these two problems, we build a door manipulation environment with diverse realistic manipulation mechanisms, and equip this environment with a large-scale door dataset covering diverse types, handles and geometries for universal manipulation policy learning.

B. 3D Articulated Object Manipulation

Extensive investigations have been conducted in the field of articulated objects, encompassing various facets such as dynamic structure recontruction [13], [14], [15], [16], [17], [18], 6d pose estimation [19], [20], [21], [22], [6], robot manipulation [23], [4], [8], [7], [24], [25], [26], [27], [28], and point-level visual affordance [3], [5], [29]. Among these areas of study, particular attention has been dedicated to the manipulation of articulated objects, with a specific emphasis on generating optimal action sequences. Previous works [1], [7] have explored the training of a single universal policy using state-based [1] or visual-based RL [7], [26], which suffer from vast exploration space and complicated manipulation mechanisms for the door manipulation task. Otherwise, we propose a novel framework that disentangles the whole manipulation process into three stages with respective universal policies, and integrate them into the whole universal policy leveraging conditional training.

III. Large-scale Diverse Door Environment

An environment based on large-scale diverse door datasets and realistic simulation is a necessity for the training of a universal manipulation policy. Considering the lack of diversity and realness in the current simulation environments [1], [12], [3], [4], [5], [6], [7], [8], we propose a novel environment with large-scale diverse door dataset (section III-A) and realistic simulations (section III-B) based on IsaacGym [30].

TABLE I: **Statistic Comparisons** between previous dataset and ours. For category, **Int.**, **Win.**, **Car.**, **Saf.**, **Cab.**, **Ref.** respectively denote doors from Interior, Window, Car, Safe, Cabinet, Refrigerator. For asset number, **B**, **H**, **CO** indicate numbers of body, handle and composited object assets with the two parts.

Datasets	Int.			Win.			Car.			Saf.			Cab.			Ref.		
	B	**H**	**CO**	**B**	**H**	**CO**	**B**	**H**	**CO**	**B**	**H**	**CO**	**B**	**H**	**CO**	**B**	**H**	**CO**
AKB-48 [9]	-	9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PartNet-M [10]	26	22	26	3	1	3	-	-	-	30	14	30	155	-	-	4	-	-
GAPartNet [6]	14	11	14	-	-	-	-	-	-	29	1	29	133	-	-	4	-	-
DoorGym [1]	-	20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Ours	**57**	**96**	**5472**	**18**	**37**	**666**	**22**	**15**	**330**	**61**	**39**	**2379**	**160**	**8**	**1280**	**10**	**9**	**90**

TABLE II: **Comparison between Our Environment and Others**. For Simplicity, **P**, **A**, **G** and **D** respectively denote PartNet-M, AKB-48, GAPartNet and DoorGym in Tab.I.

Env.	Data.	Mob.	Latch.	Part.	Occ.
GAPartNet [6]	P + A				
W2A [3], [5], [29]	P			✔	
RLAfford [26]	P	✔			
PartManip [7]	G	✔		✔	
DoorGym [1]	D		✔	✔	
Env-aware Afford [31]	P			✔	✔
Ours	**Ours**	✔	✔	✔	✔

A. *Large-Scale Diverse Door Dataset*

In recent years, several works have proposed their datasets for door manipulation as illustrated in Tab.I. DoorGym [1] claims a large-scale and scalable dataset specifically for door manipulation. Due to the hard-coded generation and the same templates, the doors they construct lack the diversity of geometry, which leads to bad performance on unseen doors. Limited by the environment, the door boards don't have a corresponding urdf or object entity, which can not be transferred to other environments. PartNet-Mobility [10] and AKB-48 [9] provide a diverse dataset for articulated objects including doors. Focusing on the cross-category diversity, they ignore the intra-category diversity of doors, which are not suitable for our goal. Hence we introduce a large-scale diverse dataset specifically for door manipulation.

Compared with the current dataset providing the whole door without part composition over diverse configuration, we construct our dataset in the formulation of two door parts, named *body* and *handle*. Given the composition method we provide, users can configure their personalized door objects with diverse settings according to their needs. In total, we create 328 bodies and 204 handles which can be composed of thousands of door objects, as shown in Tab.I.

To boost diversity, we make efforts at both inter-category and intra-category levels. For category level, we select 6 representative categories that cover most of the door-opening scenes we encounter in the real world, including **Interior**, **Window**, **Car**, **Safe**, **Cabinet** and **Refrigerator**. For the intra-category level, we collect hundreds of door bodies and handles with irregular geometries and diverse mechanisms as illustrated in Fig.1 (a). Tab.I provides a detailed statistic comparison between previous datasets and ours. Notice that we do the calculation for all objects based on whether their manipulation mechanisms are similar to doors.

B. *Diverse and Realistic Manipulation Mechanisms*

To closely emulate the real world, we specifically configure with diverse manipulation mechanisms, as demonstrated in Fig. 1 (c). To simulate the latching mechanism, we apply a force $\mathscr{F}_{door}(\theta)$ in the reverse direction of the opening to the doors until the handle joint angle θ_h have reached the opening threshold $thre_{door}$. Furthermore, we also add a resilient force to both the handle $\mathscr{F}_{handle}(\theta)$ and door to ensure the robustness of the manipulation policy. The forces are formulated as:

$$\mathscr{F}_{door}(\theta_d, \theta_h) = \begin{cases} F_f, & \theta_h \leq thre \\ k_1 \theta_d, & \theta_h > thre \end{cases} \tag{1}$$

$$\mathscr{F}_{handle}(\theta_h) = k_2 \theta_h \tag{2}$$

Where θ_d represents the door joint angle. To better emulate the real-world and improve policy robustness, we incorporate comprehensive domain randomization on robot and door properties including k_1, k_2, F_f. Instead of a parallel or suction gripper, we utilize the whole robot arm with a mobile base and parallel gripper as the agent to manipulate the doors. In this case, collision between doors and robot arm as well as the joint limitation of the robot arm itself must be taken into consideration. Unlike previous works using the pseudo perfect observation [26], [6], [5], [3] or partial observation [7], [4], [8] which don't consider the occlusion caused by the door itself or the robot arm, we put a fixed camera in front of the door and robot arm and acquire the partially occluded point cloud generated by the depth image as the observation. Tab.II shows the detailed comparison between our environment and others from various aspects.

IV. METHOD

As illustrated in Fig.2, we propose a novel framework that disentangles door manipulation into three distinct but related stages, each with a corresponding universal manipulation policy (section IV-A). We leverage a conditional framework to train these policies, as they have inter-dependencies, and thus they can be integrated into a unified universal policy (section IV-B). In the first stage, we employ generalizable point-level visual affordance [11] to propose stable grasp poses (section IV-C). In the second stage, we train a universal policy covering multiple handle manipulation mechanisms in our proposed realistic environment (section IV-D). In the third stage, we train a policy to open doors with unlocked handles (section IV-E). Additionally, we provide a comprehensive description of our data collection and training strategy(section IV-F).

A. *Problem Formulation*

We disentangle the door manipulation task into three stages: handle grasping, handle manipulation and door opening. Following UMPNet [4], we define the policy π for each stage to be generating robot actions a_t, given the partial point cloud observation O_t and the robot state S_t at time t. Here, the robot action $a_t = (p_t, r_t)$ indicates the next end-effector pose of robot arm, consisting of position $p_t \in \mathbb{R}^3$

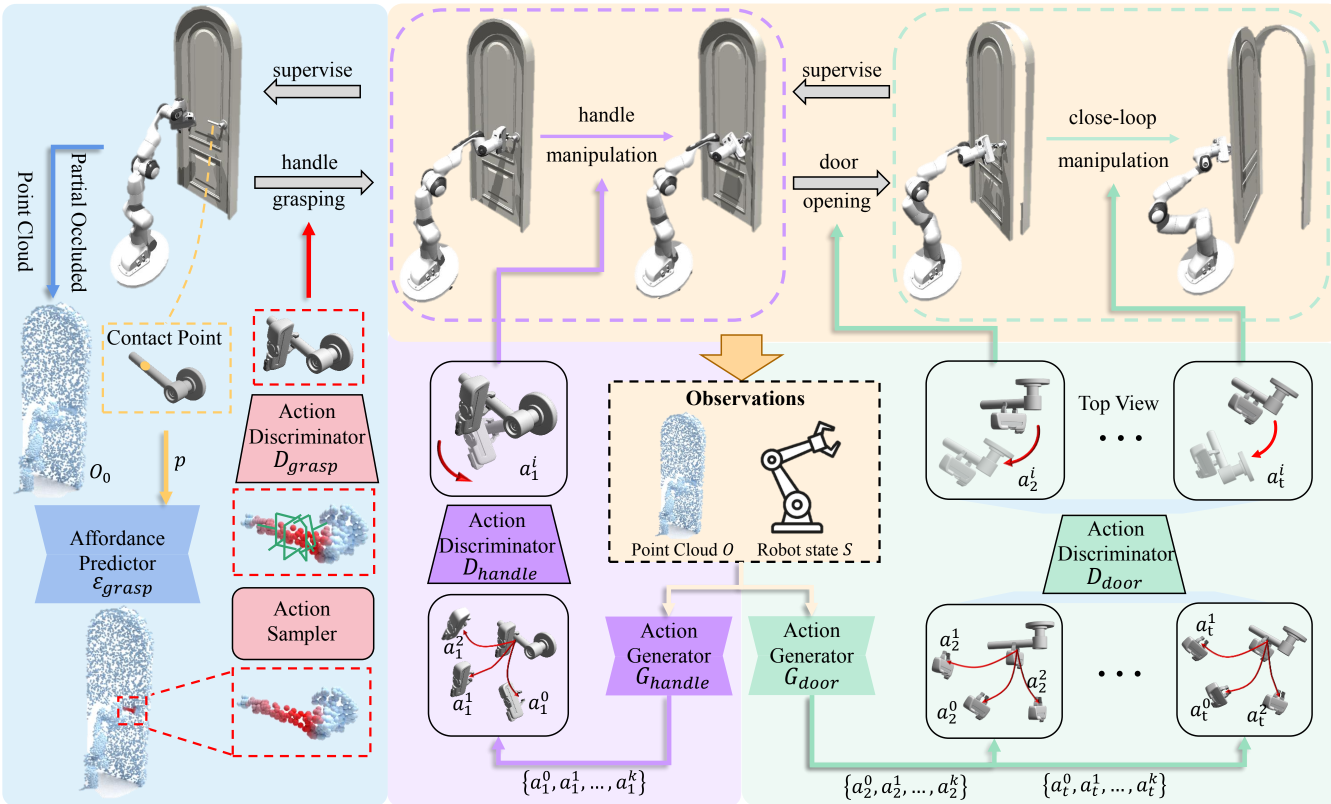


Fig. 2: **UniDoorManip Framework**. We disentangle the entire door manipulation process into three stages: handle grasping, handle manipulation, and door opening. We predict affordance map to grasp the handle and employ a similar formulation but train separate policies for handle manipulation and door opening. Conditional training integrates these three policies.

and orientation $r_t \in \mathbb{SO}(3)$. The robot state S_t consists of its joint angles and velocities. In **stage one**, given the initial point cloud observation O_0 and the contact point $p_0 \in O_0$, the model outputs a per-point affordance map A_{O_0} where a higher score on p_0 indicates a greater chance of successful door manipulation after grasping p_0 with action a_0. In **stage two and three**, given O_t and S_t, we train two separate but similar manipulation policies to generate the action a_t for diverse handle mechanisms and door opening.

B. Disentanglement and Conditional Training

As door manipulation includes multiple different stages, it is difficult to directly train a universal policy in an end-to-end manner covering all the manipulation policies in those diverse stages. Therefore, we disentangle the door manipulation into three distinct stages. Due to the similarity of the manipulation policies in each stage, it is easy to train a universal and generalizable policy in each stage and merge them into a whole universal manipulation policy.

These stages have internal dependencies. For example, if the robot grasps the handle too far outward, it may result in disengagement while rotating the handle. Also, if the robot has a strange state in handle manipulation, it's easy to collide with the door while the door is opening. Hence, to integrate all policies for each stage, we employ a conditional training formulation leveraging these internal relations. Considering that all policies aiming to successful door opening, we train our models in the reverse order of inference.

C. Handle Grasping

Due to the diverse geometries of different handle types, it's challenging to train a universal and generalizable manipulation policy for initial handle grasping. While existing heuristic methods leveraging the pose estimation [6] or keypoint representation [2] can not generalize to objects with irregular shapes, we employ point-level visual affordance [11] for manipulation, which predicts a point-level score map on the target object, indicating actionability for downstream tasks. The policy for handle grasping consists of three models: affordance predictor $\mathscr{E}_{\text{grasp}}$, action sampler and action discriminator $\mathscr{D}_{grasp}$. During inference, taking the initial observed point cloud $O_0 \in \mathbb{R}^{3\times 4096}$ and the contact point $p \in \mathbb{R}^3$ as input, the affordance predictor $\mathscr{E}_{\text{grasp}}$ predicts a point-level score map where we choose point with highest score as the grasp point. We employ PointNet++ [32] to extract a per-point feature from O_0. Based on the grasp point, the action sampler proposes alternative actions $\{a_0^0, a_0^1, \ldots, a_0^k\}$ by sampling random orientations from the normal plane of the handle axis and combining each of them with the grasp point. Taking the concatenation of the point cloud feature and the action feature extracted by another MLP, the action discriminator $\mathscr{D}_{grasp}$ outputs an action score. We select the action with the highest action score to grasp the handle.

In the reverse order of inference, we train the model of action discriminator $\mathscr{D}_{grasp}$ firstly which is supervised by the final door joint angle θ_d using L_1 loss. For affordance predictor $\mathcal{E}_{\text{grasp}}$, we sample 50 random actions based on the contact point p_0 selected from O_0. Then we estimate their action scores using $\mathcal{D}_{grasp}$ and regress the prediction to the mean score of the top-10 rated action with L_1 loss.

D. Handle Manipulation

To deal with diverse manipulation mechanisms like lever, round, key and valve in Fig.1 (c), we employ a generative network architecture to learn the universal manipulation policy for this stage. Firstly, we use an action generator $\mathcal{G}_{handle}$ which is implemented as a conditional variational autoencoder (cVAE) [33] composed of the action encoder and decoder to generate the diverse actions for different objects. Due to the unstable action generated from the sampled noise, we further employ an action discriminator $\mathcal{D}_{handle}$ to regularize the action generation. Given the alternative actions $\left\{a_1^0, a_1^1, \ldots, a_1^k\right\}$ generated by $\mathcal{G}_{handle}$, plus O_t and S_t, $\mathcal{D}_{handle}$ predicts scores for each action, which we select the action with the highest score as the manipulation action.

For training $\mathcal{G}_{handle}$, we use the KL divergence loss $\mathcal{L}_{kl}$ for regularizing Gaussian bottleneck noises, the L_1 loss $\mathcal{L}_{pos}$ to regress the action position p_t and the a 6D-rotation loss [34] $\mathcal{L}_{rot}$ for supervise the action orientation r_t. The total loss can be formulated as:

$$\mathcal{L}_h = \lambda_{KL}\mathcal{L}_{kl} + \lambda_{pos}\mathcal{L}_{pos} + \lambda_{rot}\mathcal{L}_{rot} \tag{3}$$

where we use a set of hyper-parameters to adjust the loss balance. For training $\mathcal{D}_{handle}$, we use L_1 loss to supervise the action score with the residual handle joint angle between before and after the action execution.

E. Door Opening

To tackle the diverse geometries and long action sequences, instead of open-loop manipulation [5], we employ a closed-loop formulation that iteratively generates the next action a_t for door opening based on the current observation for door opening stage. We follow the implementation of the policy for handle manipulation and employ an action generator $\mathcal{G}_{door}$ and an action discriminator $\mathcal{D}_{door}$ to output the door opening action. In addition, we employ a loss function formulation similar to that used for handle manipulation to train both $\mathcal{G}door$ and $\mathcal{D}door$. Unlike the previous discriminator, here the action score is supervised using the residual door joint angle.

F. Data Collection and Training Strategy

To train policies for the three stages, we collect data including the input observation and ground truth supervision leveraging the rule-based method. For planning rules of different manipulation mechanisms and stages, we use the door states like the joint axis of the handle and body which can be acquired in the simulation environment to calculate the next action.

We train the policies in reverse order of inference, beginning with the door opening policy. This policy is supervised using the final door state, enabling it to generate actions that maximize the door opening angle. Once trained, we replace the door opening rule with this policy and proceed to collect data for training the handle manipulation policy. This policy is thus optimized to produce actions that support the subsequent door opening stage. Finally, we replace both the handle and door opening rules with the trained policies and gather data for training the affordance prediction model. By incorporating supervision from the final door joint angle into affordance score, the initial grasping policy learns to predict foresightful grasp poses that mitigate failure modes in later stages, like disengagement between the end-effector and the handle, or collisions between the robot arm and the door.

V. EXPERIMENTS

In this section, we carefully evaluate the performance of our proposed framework in our simulation environments and real world. Our evaluation aims to address the following questions:

- Can visual affordance effectively indicate optimal handle grasping pose for different handle types?
- Can the disentanglement and conditional training design be helpful to learn universal policy and improve performance?
- How does the mobile base affect the task?
- Is the universal policy robust enough for large door joint angles?
- Can the universal policy learned in simulation be applied to real-world scenarios?

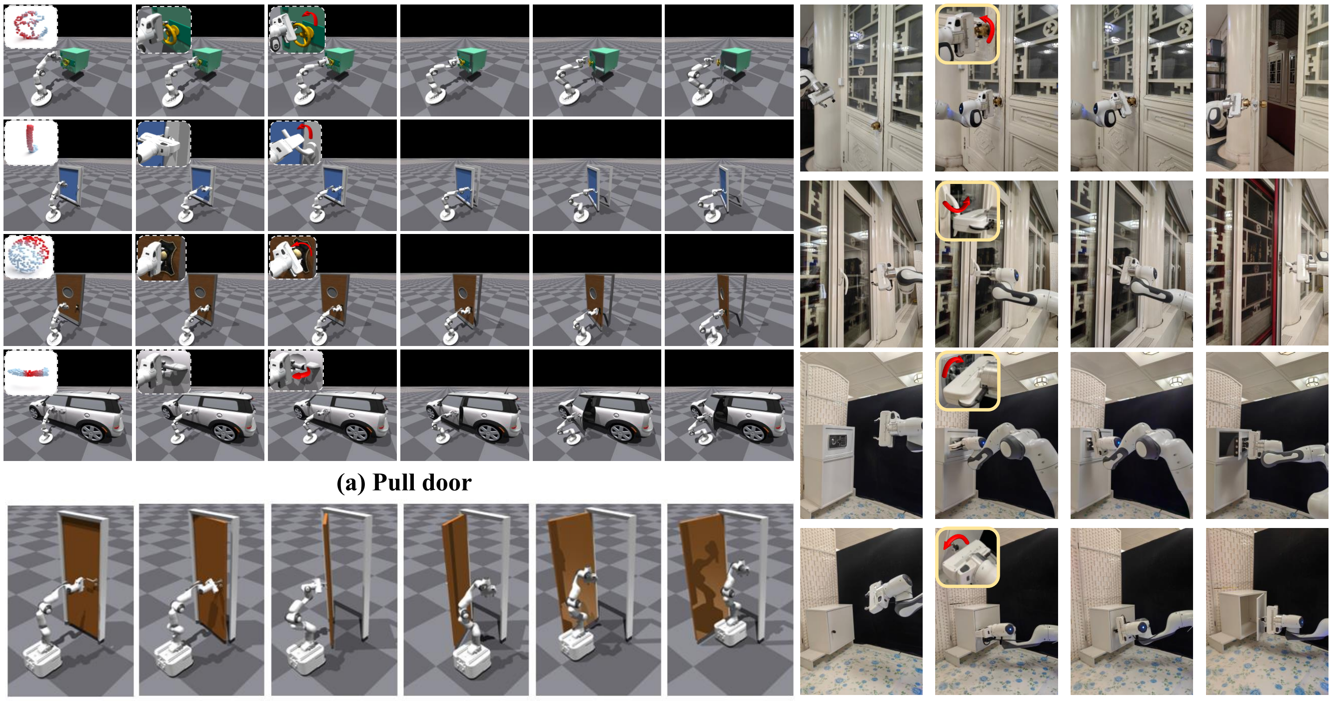
A. Task, Metric, Evaluation and Settings

We conduct our experiments on two challenging tasks: **pull door** and **move through door**. Notice that the initially closed door can only be unlocked and pulled to open. For **evaluation metric**, we use the task success rate as our main evaluation metric. We consider the success of the **pull door** task when the door joint angle θ_d is larger than a threshold $thre_{door}$. Here, we set $thre_{door}$ to be 45°. For the **move through door** task, the robot arm is required to pull the door and move through it.

For **training and testing**, to show the generalizability of our proposed framework across diverse categories, geometries and manipulation mechanisms, we train the universal manipulation policy on the door category including Interior, Window, Safe and Car. We carefully split the object assets of the aforementioned categories into the train and test part which ensures the universal policy is tested on unseen shapes. Furthermore, we also leave two more categories, Cabinet and Refrigerator, for extensive tests on unseen novel categories. Because only the Interior door can be moved through, we evaluate the **move through door** task on it.

B. Baselines and Ablations

We compare our method with the following baselines: 1) **GAPartNet + GT**: following the heuristic manipulation method of GAPartNet, we directly obtain the initial ground truth like poses of the handle and board from simulation and implement a heuristic motion planning method for door manipulation. 2) **DoorGym**: we introduce the state-based



(a) Pull door

(b) Move through the door

(c) Real world experiment

Fig. 3: **Qualitative Results in Different Configurations.**

PPO [35] implemented in DoorGym as our baseline. Here we use a flying gripper as the agent and put the gripper close to the handle in the beginning. 3) **PartManip**: we introduce a visual-based PPO used in PartManip as our baseline. The setting is similar to **DoorGym**. 4) **VAT-MART**: a method that proposes visual action trajectory for downstream manipulation task in an open-loop formulation. 5) **DP3**: a end-to-end imitation learning method that extends diffusion policy with 3D visual represetations.

To further evaluate the importance of different components of our method, we conducted an ablation study by comparing our method with four ablations: 1) **Ours w/o disentangle.**: instead of training policies for the latter two stages separately, we use a single policy to generate the manipulation action after handle grasping. 2) **Ours w/o condition.**: ours without conditional training. 3) **Ours w/o state.**: ours without the input observation of robot state. 4) **Ours w/o mobile.**: ours that uses a fixed base robot arm.

C. Result Analysis

1) *Visual Affordance*: The results in Fig.3 (a) visualizes the affordance map of different handle types across various geometries and categories in the first column pictures. Results of **GAPartNet + GT** in table III also demonstrate that part pose estimation can't generalize to handles with irregular shapes especially on Car and Safe.

2) *Disentanglement and Conditional Training*: Compared with end-to-end policy (**DoorGym**, **PartManip**, **DP3**), open-loop policy (**VAT-MART**), and the ablation **Ours w/o disentangle**, our policy significantly improves the success rate across the train and test categories as shown in Tab.III.

TABLE III: **Quantitative Results of Baselines, Ablations and Our Method in Different Tasks.** In the **Method** column, **Train** means testing on unseen shapes in training categories. **Test** means testing on unseen shapes in unseen test categories.

Task	Pull Door					
Method	Train				Test	
GAPartNet [6]+GT	0.62	0.88	0.41	0.44	0.52	0.26
DoorGym [1]	0.56	0.72	0.61	0.41	0.19	0.23
PartManip [7]	0.47	0.61	0.54	0.34	0.42	0.19
VAT-MART [5]	0.59	0.62	0.57	0.43	0.51	0.25
DP3 [36]	0.58	0.71	0.62	0.36	0.53	0.28
Ours w/o disentangle.	0.44	0.88	0.20	0.19	0.05	0.22
Ours w/o condition.	0.77	0.31	0.58	0.51	0.54	0.33
Ours w/o state.	0.73	0.59	0.16	0.36	0.45	0.37
Ours w/o mobile.	0.87	0.60	0.00	0.43	0.50	0.81
Ours	**0.99**	**0.91**	**0.81**	**0.72**	**0.75**	**0.89**

Hence, the disentanglement design can overcome the challenges of object diversity and large exploration of action space. Results of **Ours w/o condition** further demonstrate our disentangled policies fit each other well through the conditional training.

3) *Mobile Base*: The ablation **Ours w/o mobile.** in Tab.III shows that the success rate falls in all categories and achieves zero in the Car category. This suggests that the absence of a mobile base makes the robot arm prone to collide with doors (Fig. 5 (a)). Thanks to the mobile base, we evaluate the performance of our policy for the **move through door** task. By combining our manipulation policy with task and motion planning (TAMP [37]), the robot can adjust the joint

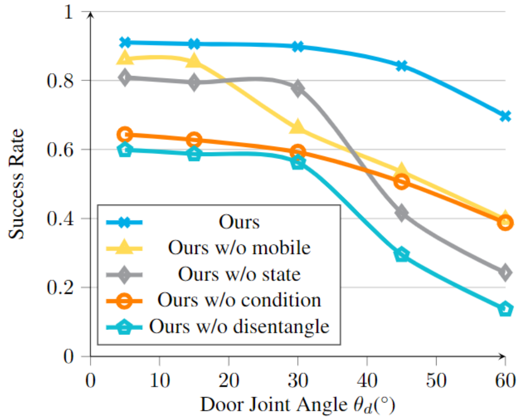


Fig. 4: **Comparison of the ablations and ours for different door joint angles**. For each door joint angle, we do experiments on all categories and get the average success rate.

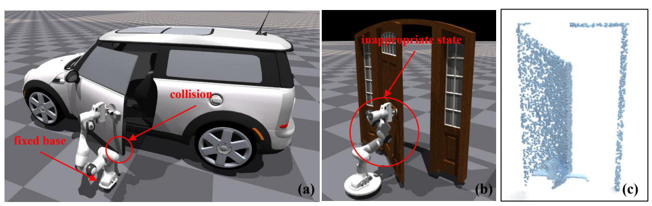


Fig. 5: **Failure cases of ablations.**

states to facilitate the base to move through the door when the door is open enough as shown in Fig.3 (b), with a success rate of **74%** in this more realistic and challenging task.

4) *Policy Robustness*: To evaluate the robustness of our policy at large door joint angles, we compare the performance of ours and the ablations for different door joint angles. The results in Fig.4 demonstrate that with the increase of door joint angle, ours can sustain the above 0.7 high success rate while the ablations suffers a significantly fall. **Ours w/o state.** achieve nearly 0.8 success rate when the door joint angle is smaller than 30° but fall when it becomes larger. This proves that the inclusion of robot state can mitigate the degradation of visual observation. Besides, the results from Tab.III show that without robot state as input, the success rate is lower in all categories due to the occluded point cloud. We further investigate the reason by visualizing the whole manipulation and visual observation. As Fig.5 (b) and (c) shows, the joints of the robot arm are easy to fall into strange states, which is not conducive to the execution of subsequent actions. Due to the inverse kinematics solution to transform the output end-effector actions into the joint states, adding robot state as input can regularize the model to output more consecutive end-effector actions.

TABLE IV: **Real-world Success Rates.**

Category	Interior	Window	Safe	Cabinet
Fixed base	7/15	8/15	10/15	12/15
Mobile base	9/15	/	/	12/15

5) *Real-world Experiments*: To investigate the generalizability of our policy to real-world scenarios, we conduct experiments on the task similar to the simulation on both fixed-base and mobile-base robot systems.

For the fixed-base robot settings, we employ a Franka Emika Panda Robot Arm as our agent to manipulate the door. An Azure Kinect DK camera is placed beside the robot arm to capture the visual observation. Taking the real-time point clouds from the depth camera and the robot state from the robot arm as the input, our policy outputs the end-effector action in a close loop fashion. To transform the end-effector action to joint force/torque, we employ the impedance control to control the robot. We select various real-world doors including Interior, Window, Safe, and Cabinet, and take 15 trials for each category. Fig.3 (c) demonstrates the whole manipulation process.

For the mobile-base setup, we utilize an ARX-x7s robot equipped with a dual-arm system and a wheeled mobile base. Similar to the fixed-base case, a RealSense camera mounted on the robot's head captures 3D point clouds. The policy uses this visual input along with the robot's state to generate end-effector actions. To execute these actions, we treat the mobile base as two additional joints representing motion in the xy-plane and solve for the inverse kinematics accordingly. The resulting arm joint commands are sent directly to the manipulator, while the base motion is converted into wheel velocities via a PID controller. Experiments are performed on Interior and Cabinet doors, with 15 trials per category. Fig.6 demonstrates the whole manipulation process.

A trial is considered successful if the door is opened beyond 30°. The quantitative results are shown in Tab.IV, which provides evidence that our policy can be applied to the real world. We observe higher success rates for cabinet and safe doors compared to interior and window doors, which we attribute to their lower joint friction and reverse torque. Compared with the fixed-base configuration, the mobile-base robot achieves better performance on Interior but not on Cabinet. We attribute this difference to the distinct characteristics of the door categories. The mobile base provides a larger workspace, which is particularly beneficial for opening Interior doors that often require a wider range of motion. For the failure case in real-world, we notice that the slippery surface of the handle is hard for the parallel gripper to grasp steadily in the entire door opening process.

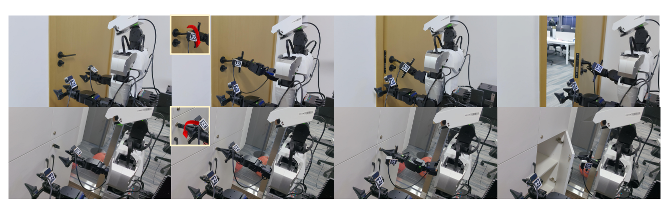


Fig. 6: **Real-World Experiments with Mobile Robot.**

VI. Conclusion

We aim to learn a universal manipulation policy with diverse door categories, geometries and mechanisms. We first build a novel environment including a large-scale door dataset and diverse realistic mechanisms. Then, we present a novel framework for universal policy learning which disentangles the entire manipulation process into three stages and integrates them by conditional training. Extensive experiments validate the strong performance of our framework.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China (No.62576046, No.62301066, No.62406028) and Beijing Academy of Artificial Intelligence (Z251100008125041).

REFERENCES

[1] Y. Urakami, A. Hodgkinson, C. Carlin, R. Leu, L. Rigazio, and P. Abbeel, "Doorgym: A scalable door opening environment and baseline agent," *arXiv preprint arXiv:1908.01887*, 2019.

[2] J. Wang, S. Lin, C. Hu, Y. Zhu, and L. Zhu, "Learning semantic keypoint representations for door opening manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6980–6987, 2020.

[3] K. Mo, L. J. Guibas, M. Mukadam, A. Gupta, and S. Tulsiani, "Where2act: From pixels to actions for articulated 3d objects," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 6813–6823.

[4] Z. Xu, H. Zhanpeng, and S. Song, "Umpnet: Universal manipulation policy network for articulated objects," *IEEE Robotics and Automation Letters*, 2022.

[5] R. Wu, Y. Zhao, K. Mo, Z. Guo, Y. Wang, T. Wu, Q. Fan, X. Chen, L. Guibas, and H. Dong, "VAT-mart: Learning visual action trajectory proposals for manipulating 3d ARTiculated objects," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=iEx3PiooLy

[6] H. Geng, H. Xu, C. Zhao, C. Xu, L. Yi, S. Huang, and H. Wang, "Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7081–7091.

[7] H. Geng, Z. Li, Y. Geng, J. Chen, H. Dong, and H. Wang, "Partmanip: Learning cross-category generalizable part manipulation policy from point cloud observations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2978–2988.

[8] B. Eisner, H. Zhang, and D. Held, "Flowbot3d: Learning 3d articulation flow to manipulate articulated objects," *arXiv preprint arXiv:2205.04382*, 2022.

[9] L. Liu, W. Xu, H. Fu, S. Qian, Q. Yu, Y. Han, and C. Lu, "Akb-48: A real-world articulated object knowledge base," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 809–14 818.

[10] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang *et al.*, "Sapien: A simulated part-based interactive environment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 097–11 107.

[11] J. J. Gibson, "The theory of affordances," *Hilldale, USA*, vol. 1, no. 2, pp. 67–82, 1977.

[12] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu, "robosuite: A modular simulation framework and benchmark for robot learning," *arXiv preprint arXiv:2009.12293*, 2020.

[13] B. Abbatematteo, S. Tellex, and G. Konidaris, "Learning to generalize kinematic models to novel objects," in *Proceedings of the 3rd Conference on Robot Learning*, 2019.

[14] K. Hausman, S. Niekum, S. Osentoski, and G. S. Sukhatme, "Active articulation model estimation through interactive perception," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3305–3312.

[15] D. Katz, M. Kazemi, J. A. Bagnell, and A. Stentz, "Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 5003–5010.

[16] N. Nie, S. Y. Gadre, K. Ehsani, and S. Song, "Structure from action: Learning interactions for articulated object 3d structure discovery," *arXiv preprint arXiv:2207.08997*, 2022.

[17] J. Lv, Q. Yu, L. Shao, W. Liu, W. Xu, and C. Lu, "Sagci-system: Towards sample-efficient, generalizable, compositional, and incremental robot learning," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 98–105.

[18] Z. Jiang, C.-C. Hsu, and Y. Zhu, "Ditto: Building digital twins of articulated objects from interaction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5616–5626.

[19] A. Jain, R. Lioutikov, C. Chuck, and S. Niekum, "Screwnet: Category-independent articulation model estimation from depth images using screw theory," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 670–13 677.

[20] Q. Liu, W. Qiu, W. Wang, G. D. Hager, and A. L. Yuille, "Nothing but geometric constraints: A model-free method for articulated object pose estimation," *arXiv preprint arXiv:2012.00088*, 2020.

[21] X. Li, H. Wang, L. Yi, L. J. Guibas, A. L. Abbott, and S. Song, "Category-level articulated object pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3706–3715.

[22] Y. Weng, H. Wang, Q. Zhou, Y. Qin, Y. Duan, Q. Fan, B. Chen, H. Su, and L. J. Guibas, "Captra: Category-level pose tracking for rigid and articulated objects from point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 209–13 218.

[23] D. Katz and O. Brock, "Manipulating articulated objects with interactive perception," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 272–277.

[24] T. Mu, Z. Ling, F. Xiang, D. Yang, X. Li, S. Tao, Z. Huang, Z. Jia, and H. Su, "Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations," *arXiv preprint arXiv:2107.14483*, 2021.

[25] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao *et al.*, "Maniskill2: A unified benchmark for generalizable manipulation skills," *arXiv preprint arXiv:2302.04659*, 2023.

[26] Y. Geng, B. An, H. Geng, Y. Chen, Y. Yang, and H. Dong, "End-to-end affordance learning for robotic manipulation," *arXiv preprint arXiv:2209.12941*, 2022.

[27] J. Borja-Diaz, O. Mees, G. Kalweit, L. Hermann, J. Boedecker, and W. Burgard, "Affordance learning from play for sample-efficient policy learning," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6372–6378.

[28] G. Schiavi, P. Wulkop, G. Rizzi, L. Ott, R. Siegwart, and J. J. Chung, "Learning agent-aware affordances for closed-loop interaction with articulated objects," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5916–5922.

[29] Y. Wang, R. Wu, K. Mo, J. Ke, Q. Fan, L. Guibas, and H. Dong, "Adaafford: Learning to adapt manipulation affordance for 3d articulated objects via few-shot interactions," *European conference on computer vision (ECCV 2022)*, 2022.

[30] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[31] K. Cheng, R. Wu, Y. Shen, C. Ning, G. Zhan, and H. Dong, "Learning environment-aware affordance for 3d articulated object manipulation under occlusions," *arXiv preprint arXiv:2309.07510*, 2023.

[32] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[33] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems*, vol. 28, 2015.

[34] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5745–5753.

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

[36] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy," *ArXiv*, vol. abs/2403.03954, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:268253298

[37] C. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *arXiv: Robotics,arXiv: Robotics*, Oct 2020.