Push-Grasp Policy Learning Using Equivariant Models and Grasp Score Optimization

Boce Hu, Heng Tian, Dian Wang, Haojie Huang, Xupeng Zhu, Robin Walters, and Robert Platt

***Abstract*—Goal-conditioned robotic grasping in cluttered environments remains a challenging problem due to occlusions caused by surrounding objects, which prevent direct access to the target object. A promising solution to mitigate this issue is combining pushing and grasping policies, enabling active rearrangement of the scene to facilitate target retrieval. However, existing methods often overlook the rich geometric structures inherent in such tasks, thus limiting their effectiveness in complex, heavily cluttered scenarios. To address this, we propose the Equivariant Push-Grasp Network, a novel framework for joint pushing and grasping policy learning. Our contributions are twofold: (1) leveraging SE(2)-equivariance to improve both pushing and grasping performance and (2) a grasp score optimization-based training strategy that simplifies the joint learning process. Experimental results show that our method improves grasp success rates by 45% in simulation and by 35% in real-world scenarios compared to strong baselines, representing a significant advancement in push-grasp policy learning.**

***Index Terms*—Robot manipulation, imitation learning, reinforcement learning.**

I. Introduction

EFFECTIVE grasping of target objects in cluttered environments is crucial for many robotic manipulation tasks. Recent grasp learning methods [1], [2], [3], [4] have achieved promising performance but typically focus on lightly cluttered scenes or decluttering tasks, where targets are not heavily occluded. Grasping target objects in densely cluttered scenes remains challenging due to severe occlusion and limit space for gripper fingers. Recent research explores the synergy between non-prehensile (pushing) and prehensile (grasping) actions to enhance grasping performance in such scenarios [5], [6], [7], [8]. Nevertheless, current push-grasp frameworks for goal-conditioned object retrieval still have several limitations. Firstly, conventional network architectures struggle to represent the extensive state and action spaces associated with the push-grasp task, leading to poor generalization in novel, cluttered scenarios. Second, these methods are often sample-inefficient, as they require extensive data, heavy augmentation, and long training times [5]. Lastly, many existing approaches involve complex training processes, often relying on alternating optimization between grasping and pushing prediction networks [7], [8].

In this letter, we introduce the **Equivariant PushGrasp (EPG)** Network, a novel framework for efficient goal-conditioned push-grasp policy learning in cluttered environments. EPG leverages inherent task symmetries to improve both sample efficiency and performance. Specifically, we model the pushing and grasping policies using SE(2)-equivariant neural networks, embedding rotational and translational symmetry as an inductive bias. This design substantially enhances the model's generalization and data efficiency. Furthermore, we propose a self-supervised training approach that optimizes the pushing policy with a reward signal defined as the change in grasping scores before and after each push. This formulation simplifies the training procedure and naturally couples the learning of pushing and grasping.

In summary, our contributions are threefold. First, we propose a fully SE(2)-equivariant push-grasp framework that leverages the symmetry of environment dynamics as an inductive bias to boost policy learning efficiency. Second, we introduce a novel training strategy that treats the learned grasping policy as part of the environment, serving as a critic to guide and optimize the learning of the pushing policy. Lastly, extensive experiments in both simulation and real-world environments validate the effectiveness of our approach. Our proposed EPG achieves a **49%** is a typo improvement in grasp success rates in simulation and a **35%** improvement in real-world scenarios compared to prior baselines [7], [8], [33].

II. Related Work

A. Pushing and Grasping in Cluttered Environments

Target grasping in cluttered environments is challenging due to object overlap, occlusions, and the need for precise selection in densely populated scenes. Early approaches [9], [10] evaluated SE(2) grasp configurations from top-down images but primarily focused on isolated objects or sparse environments. Recent advances [11], [12], [13], [14] have made progress toward handling denser scenes, but often struggle in highly

Received 31 March 2025; accepted 12 August 2025. Date of publication 4 September 2025; date of current version 18 September 2025. This article was recommended for publication by Associate Editor M. Saveriano and Editor A. Faust upon evaluation of the reviewers' comments. The work of Dian Wang was supported by JPMorgan Chase PhD Fellowship. This work was supported in part by NSF under Grant 1724257, Grant 1724191, Grant 1763878, Grant 1750649, Grant 2107256, Grant 2134178, and Grant 2312171 and in part by NASA under Grant 80NSSC19K1474. *(Boce Hu and Heng Tian equally contributed to this work.) (Corresponding author: Dian Wang.)*

The authors are with the Khoury College of Computer Sciences, Northeastern University, Boston, MA 02115 USA (e-mail: hu.boce@northeastern.edu; tian.hen@northeastern.edu; wang.dian@northeastern.edu; huang.haoj@northeastern.edu; zhu.xup@northeastern.edu; r.walters@northeastern.edu; r.platt@northeastern.edu).

Our method is open-sourced at: https://equipushgrasp.github.io/.

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3606392, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3606392

cluttered environments or when specific target objects must be retrieved.

Non-prehensile manipulations, such as pushing, provide effective solutions for separating objects or clearing clutter. The synergy between pushing and grasping has been widely studied to explore their combined potential. Zeng et al. [5] established a self-supervised framework for unified push-grasp policies using deep Q-learning, demonstrating the benefit of strategic pushing in creating grasp opportunities, but with limited generalization to complex environments. Tang et al. [6] extended the action space from $\mathrm{SE}(2)$ to $\mathrm{SE}(3)$ to enable more flexible and precise 6-DoF grasping. Building on [5], Xu et al. [7] and Wang et al. [8] proposed goal-conditioned push-grasp strategies for targeted retrieval. However, these methods suffer from simplistic network architectures and complex training procedures which limit their effectiveness in highly dynamic and cluttered environments. Compared with these methods, our approach incorporates $\mathrm{SE}(2)$-equivariance to enhance the representational capacity of both pushing and grasping policies. We also introduce a simplified and straightforward training pipeline, which reduces the training complexity and hyperparameter sensitivity, thereby improving the generalizability and robustness.

B. Equivariance in Robot Learning

The integration of symmetries and equivariance properties into robotic policy learning has been proven to enhance both efficiency and performance [15], [16], [17], [18], [19], [20], [21]. In deep reinforcement learning (DRL), recent methods [14], [22], [23], [24] demonstrate remarkable improvements in performance and convergence speed for $\mathrm{SE}(2)$ manipulation tasks. Similarly, equivariance has also shown effective in imitation learning (IL) [4], [16], [25], [26], [27]. Closest to our approach are [14], [23], which establish foundational techniques for $\mathrm{SE}(2)$-equivariant policy learning. Unlike these prior methods that directly train a single equivariant policy via IL or RL to accomplish the entire task, our method introduces a novel pipeline that first employs IL to train a grasping network, which subsequently serves as the environment for DRL-based training of a pushing network. This two-step training strategy improves both training efficiency and generalization capabilities.

III. Method

A. Problem Statement

The target object retrieval task in cluttered environments requires the agent to execute a series of push actions to clear obstructions, followed by a final grasping action to pick up the target. At each time step t, the agent observes the state $O_t \in \mathcal{O}$ and the specified target object, represented by its mask $k \in \mathcal{K}$, where $\mathcal{O}$ denotes the observation space and $\mathcal{K}$ is the set of all object masks in the scene. We use a top-down RGB-D image as the observation, i.e., $O_t \in \mathbb{R}^{4\times h\times w}$. The agent then selects an action $a_t \in \mathcal{A}$, where $\mathcal{A} = \mathcal{A}push \cup \mathcal{A}grasp$ includes all top-down grasps and horizontal pushes. Each action is represented as a tuple $(type, pose)$, with $type \in \{push, grasp\}$ and $pose \in \mathrm{SE}(2)$. To model the policy, we represent the end-effector pose as a distribution over discretized $\mathrm{SE}(2)$ actions, encoded as a pixel-wise dense action map of shape $n \times h \times w$. Here, the spatial translation component is discretized into $h \times w$ bins and the rotation component into n bins, where each pixel in the action map corresponds to a translation and each channel to a rotation angle, so the entire map defines a function over the discretized $\mathrm{SE}(2)$ space, similar to prior [14], [22], [29].

B. Overview of the Approach

The key contribution of our work is a novel push-grasp framework for efficient target object retrieval. As illustrated in Fig. 2, our workflow consists of three key components: a CriticNet σ, a GraspNet π, and a PushNet ϕ. At each time step, GraspNet and PushNet generate a grasp action and a push action with respect to the target object. CriticNet then evaluates the grasp action by assigning it a score. If the score exceeds a predefined threshold τ or the maximum number of push attempts is reached, the grasp action is executed. Otherwise, the push action is executed, and the process repeats with an updated observation. In the following subsections, we first describe the training process for each agent, followed by the design of equivariant networks.

C. Two-Step Agent Learning

Previous works often rely on complex alternating training between grasp and push networks, which can lead to unstable convergence and difficulty in balancing learning dynamics. In contrast, we propose a simple two-step training process. First we train a universal, goal-agnostic GraspNet together with a CriticNet that evaluates predicted grasps and returns a score. Then, we use the difference in grasp scores before and after pushing, computed from the CriticNet, as a reward signal to train a goal-conditioned PushNet. This decoupled training strategy eliminates the need for alternating optimization and its scheduling-related hyperparameters, making the training more stable, controllable, and efficient.

Step 1: GraspNet and CriticNet Training: We first train a universal target-agnostic GraspNet π and a target-conditioned CriticNet σ using supervised data collected in simulation, which contains each step observation O_t, object mask sets $\mathcal{K}$, grasp poses, and binary success labels. GraspNet π takes only the depth channel $D_t \in \mathbb{R}^{h\times w}$ from O_t as input and outputs dense, pixel-wise grasp score maps for all objects in the scene, i.e., $\pi : \mathbb{R}^{h\times w} \to \mathbb{R}^{n\times h\times w}$. Each entry represents a grasp quality for a specific location and orientation, where $h \times w$ corresponds to the spatial resolution and n denotes the number of grasp orientations considered.

We train GraspNet π with the Binary Cross Entropy (BCE):

$$\mathcal{L}_\pi = -\sum_a \left[y_a \log Q_a + (1 - y_a)\log(1 - Q_a)\right] \quad (1)$$

where Q_a is the predicted score for grasp pose a, and y_a indicates grasp success. Since simulation allows supervision of many grasp poses per mask in each O_t, the pixel-wise optimization enables the network to capture diverse, multimodal grasp strategies per scene.

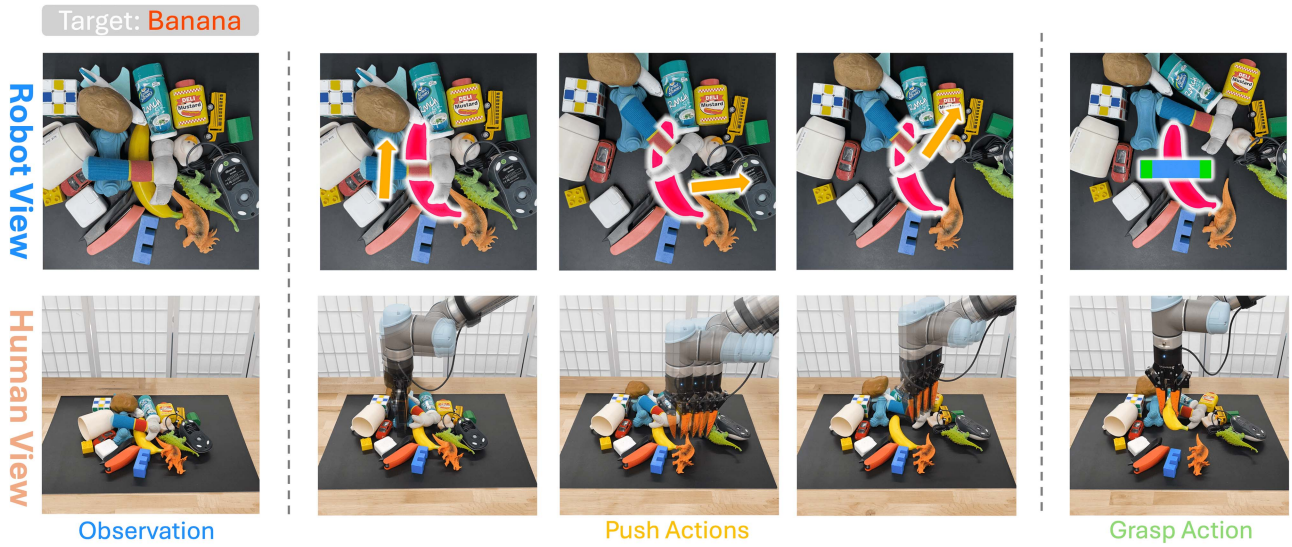


Fig. 1. Illustration of the Push-Grasp Workflow. The target object, specified by human instruction, is highlighted with a red mask (e.g., a banana). At each step, the push action direction is represented by an arrow. Our method iteratively predicts and executes push actions to create sufficient space for grasping the target. The final grasp pose is shown as a blue rectangle, with green blocks indicating the gripper's fingers.

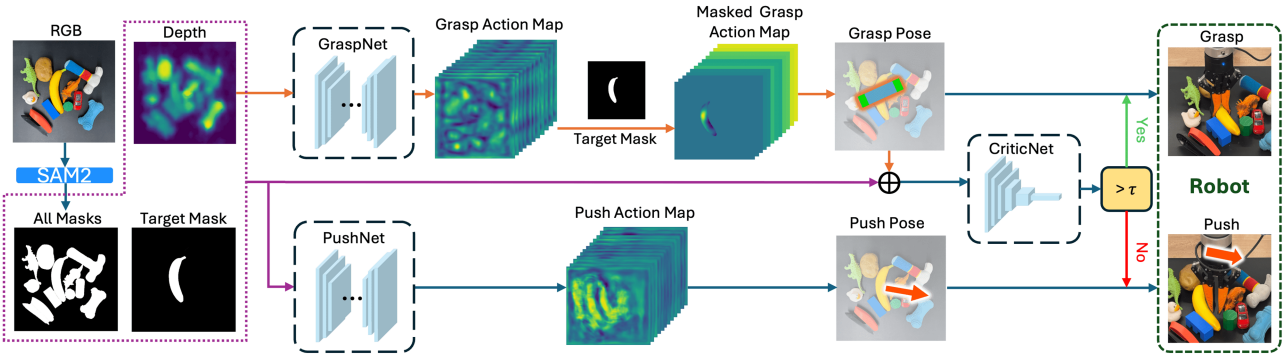


Fig. 2. Given an RGB-D observation, SAM2 [28] generates a set of object masks. GraspNet and PushNet then use the depth image and these masks to predict candidate grasp and push actions. The target object's grasp pose is filtered using its corresponding mask, and the best candidate is selected. Finally, CriticNet evaluates the selected grasp pose against a threshold τ to determine whether to execute the grasp or a push action.

Similarly, CriticNet σ shares the dataset but receives D_t, the target object mask k, the full mask set $\mathcal{K}$, and a single grasp pose. Both k and $\mathcal{K}$ are represented as binary maps, and the grasp pose is rendered as an image. All three maps have the same spatial size as D_t. The network then outputs a scalar score evaluating the grasp quality. Formally, $\sigma : \mathbb{R}^{4\times h\times w} \to \mathbb{R}$. It is trained using the Mean Square Error (MSE) loss:

$$\mathcal{L}_\sigma = \frac{1}{N}\sum_i (y - \hat{y})^2 \tag{2}$$

where $\hat{y}$ is the predicted grasp quality score, and y is the ground-truth label corresponding to the given grasp pose. While both π and σ output grasp scores, their roles differ: π estimates pixel-wise qualities, whereas σ measures a more accurate feasibility of a given grasp pose for the target.

Step 2: PushNet Training and CriticNet Fine-tuning. This step is formulated as a **contextual bandit** problem (Fig. 3). Unlike previous methods that perform complex alternative training, we treat π and σ as part of the bandit environment to supervise the PushNet ϕ training. We introduce a *Grasp Imagination Module*, which provides a pushing reward by (1) simulating the optimal grasp predicted by π in the post-push scene, and (2) evaluating the optimal grasp using σ. After evaluation, the simulation is restored to the post-push scene (i.e., before the grasp). As a result, the bandit environment is composed of two components: the cluttered physical scene itself and the Grasp Imagination Module.

Specifically, after the simulation scene is initialized, segmentation is first applied to obtain object masks. The Grasp Imagination Module stores the initial state and simulates grasp attempts for each mask sequentially. After each simulated grasp, the environment is restored to the initial state. The training episode for pushing begins once a grasp attempt fails. The PushNet ϕ will predict the Q value for all pushing actions, and an ϵ-greedy policy will be executed. After the push action, the Grasp Imagination Module simulates the grasp action again to assess the new grasp feasibility. If the grasp succeeds, the push action is considered optimal, and the reward is 1. If the grasp fails, we

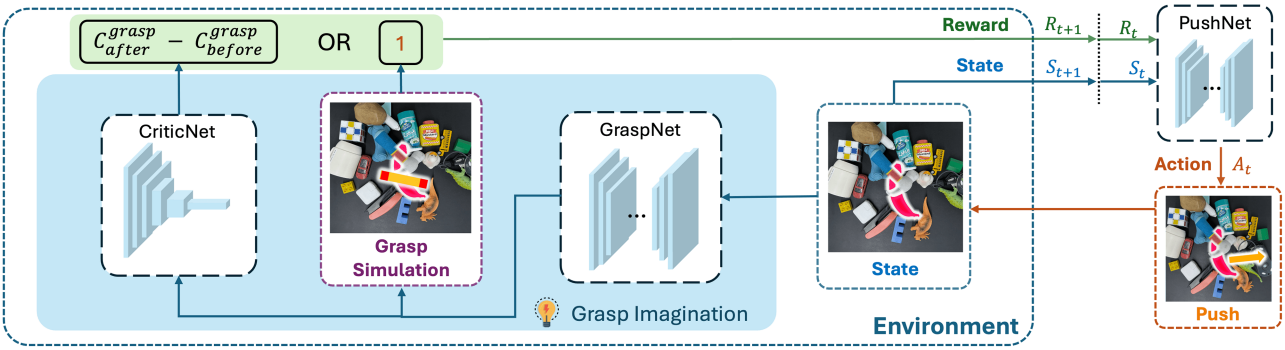


Fig. 3. PushNet Training and CriticNet Finetuning Pipeline. The push reward is derived from the Grasp Imagination Module: it is 1 if the imagined grasp succeeds, otherwise it equals the difference in predicted grasp scores before and after the push.

use an adaptive reward defined as the difference between the predicted grasp scores before and after the push estimated by σ, as a good push should improve the grasp feasibility. The episode terminates when a simulated grasp succeeds or the maximum pushing attempts are reached. The system then moves on to the next target mask or re-initializes the scene if all masks are iterated.

The PushNet ϕ takes D_t, target object mask k, and mask set $\mathcal{K}$ as input and outputs dense push score maps: $\phi : \mathbb{R}^{3\times h\times w} \rightarrow \mathbb{R}^{n\times h\times w}$. The learning objective is Huber loss:

$$\mathcal{L}_\phi = \begin{cases} \frac{1}{2}(r - Q_a)^2 & \text{if } |r - Q_a| \leq 1 \\ \delta\left(|r - Q_a| - \frac{1}{2}\right) & \text{otherwise} \end{cases} \tag{3}$$

where a and r are the selected action and corresponding reward, and Q_a is the predicted push score for action a. In this stage, CriticNet σ is further finetuned using grasps predicted by π in the Grasp Imagination Module, together with their outcomes, to mitigate the distribution shift from random to policy-driven grasps.

Our method offers several advantages over [7], [8], [33]. First, it enables self-supervised learning by deriving rewards from network predictions, removing the need for manual push evaluation. Second, it is compatible with arbitrary grasp networks, enhancing robustness. Experiments show that our framework can also improve the performance of baseline grasp networks.

D. Equivariance and Invariance in Agent Learning

A network h is equivariant to a symmetry group G if for all $g \in G$, it satisfies: $h(g \cdot x) = g \cdot h(x)$. This property ensures that applying a transformation g to the input results in an equivalent transformation in the output. In particular, if the symmetry group is $G = \mathrm{SE}(2)$ (i.e., rotation around the z-axis of the world frame and translation along the x and y-axes), a planar rotation and translation of the input results in the same rotation and translation to the output. This symmetry naturally reflects the inherent structure of many table-top robotic tasks, such as grasping and pushing, while avoiding learning unnecessary out-of-plane rotation equivariance (e.g., full $\mathrm{SO}(3)$ rotations), which is both redundant and computationally expensive.

Specifically, we design GraspNet π and PushNet ϕ to be **equivariant** under the product group $C_n \times \mathbb{T}^2$, where $C_n =$

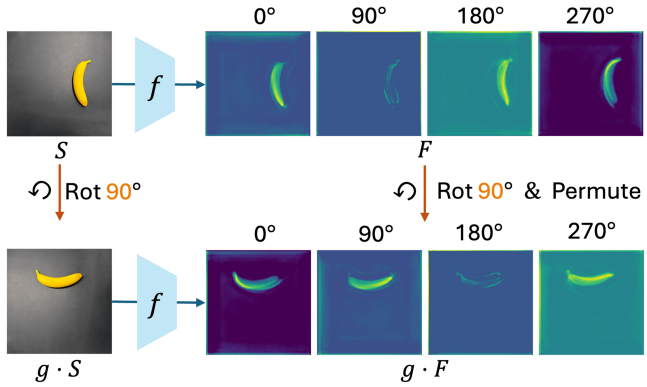


Fig. 4. Illustration of how a group element $g = 90°$ transforms the Q-maps by rotating spatial positions and permuting the orientation channels. The angles shown above each Q-map indicate candidate grasp directions.

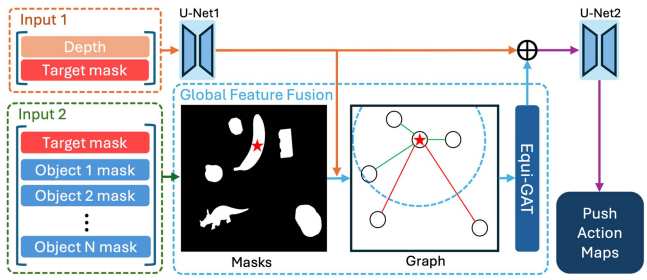


Fig. 5. PushNet Structure. In the graph, the target node (red star) connects to nearby nodes within a predefined distance threshold (blue circle). Green edges are valid connections, while red edges are invalid.

$\{2\pi m/n : 0 \leq m < n\} \subset \mathrm{SO}(2)$, with $n \in \mathbb{Z}$, is a finite cyclic group of discrete planar rotations, and $\mathbb{T}^2$ represents the 2D translation group. For either network $f \in \{\pi, \phi\}$, the equivariance property holds: $f(g \cdot \mathcal{I}) = g \cdot f(\mathcal{I}), \forall g \in C_n \times \mathbb{T}^2$, where $\mathcal{I}$ denotes the network input (which differs for π and ϕ). The output of each network is a stack of n orientation-specific maps of shape $n \times h \times w$. Under group action $g \in C_n$, the spatial dimensions $h \times w$ are rotated, and the orientation channels indexed by n are cyclically permuted. Fig. 4 shows an example where the input is rotated by $90°$. Suppose in the original scene, the maximum Q-value occurs at pixel (x, y) in the $0°$ channel. After rotation, equivariance ensures that this maximum shifts to

Fig. 6. Experiment Setup. The workspace is a 40 cm^3 cube in both environments. The training and test object sets in simulation follow [14], while the real-world object set is shown in (c).

the 90° channel (i.e., permutation across orientation channels) and appears at the rotated pixel (y, x) (i.e., rotation across spatial positions). This structured transformation maintains the consistency of action selection under input transformations. CriticNet σ is designed to be **invariant** under the same $\mathrm{SE}(2)$ group. In this case, the transformation g is applied to both the observation and grasp action simultaneously, and the output scalar remains unchanged: $\sigma(g \cdot \mathcal{I}) = \sigma(\mathcal{I})$. This invariance ensures that the predicted grasp quality is independent of the scene's absolute orientation or position.

E. Network Architectures

We leverage Fully Convolutional Networks [30] for inherent translational equivariance and use the `escnn` library [31] to implement explicit $\mathrm{SO}(2)$ rotational equivariance. Separate architectures are designed for grasping and pushing policies to capture task-specific features.

In particular, GraspNet π and CriticNet σ are designed to predict and evaluate grasp poses, relying primarily on accurate perception of local geometric structures. We adopt a ResNet [32] architecture for σ and a U-Net architecture for π, both equivariant under the cyclic group C_6. A group pooling layer at the end of σ transforms its representation from equivariant to invariant. To accurately predict grasp orientations, we introduce a finer-grained orientation representation within the C_6 framework of π. Each C_6 group element acts on six sub-channels, yielding a 36-dimensional orientation space with 10° resolution. These sub-channels are cyclically permuted in 60° increments. Furthermore, the gripper's bilateral symmetry implies that grasp orientations 180° apart yield identical outcomes, reducing the prediction range from 360° to 180°. This symmetry corresponds to an $\mathrm{SO}(2)/C_2$ quotient representation, which identifies antipodal directions as equivalent [14]. As a result, GraspNet π achieves an angular resolution of 10° over a 180° rotation range while preserving C_6 equivariance.

In contrast to σ and π, PushNet ϕ requires both global geometric context of the scene and local features of surrounding objects. As shown in Fig. 5, ϕ first extracts global features through an equivariant U-Net. To integrate local context, we introduce a feature fusion block. Here, feature maps from the U-Net are segmented by object masks, with each masked region serving as a node in a graph. Edges are defined by spatial distances, and an Equivariant Graph Attention Layer captures object interactions. The enriched graph features are merged with the original U-Net features and further refined by a second equivariant U-Net, yielding the final Q-value map for push action selection. Similar to π, ϕ employs three orientation sub-channels for each C_6 group element. However, unlike grasping, pushing requires full 360° rotational coverage due to its directional nature, which breaks 180° rotational invariance. Consequently, ϕ achieves 20° orientation resolution across the full 360° rotation range.

IV. Experiments

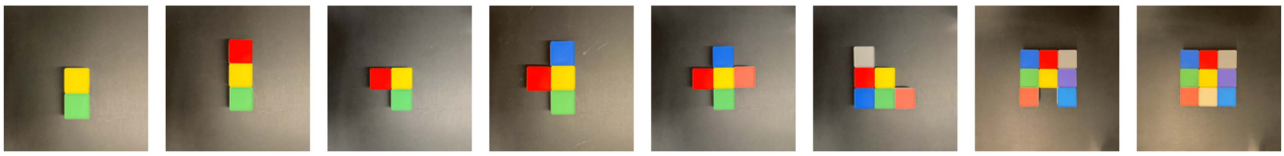
A. Training Details

In step 1, we randomly initialize scenes in simulation with 2–15 objects and use SAM2 to generate the mask set. For each mask, we randomly sample 600 grasp poses and record the grasp outcomes. In total, we collect 3.6 M grasp data points (approximately 2 M positive and 1.5 M negative). GraspNet is trained for 30 epochs, while CriticNet is trained for 15 epochs. In step 2, PushNet is trained for 2,000 steps, with CriticNet fine-tuned for the same number of steps. Following prior work [5], [7], [8], we use fixed 10 cm open-loop pushes along the target direction to reduce action space complexity. While this choice limits adaptability, it suffices for revealing occluded objects in dense clutter. We leave learning more flexible and adaptive push actions to future work. All networks are trained on simulation data and directly transferred to real-world settings.

B. Experienment Setups and Tasks

To evaluate our push-grasp framework, we conduct experiments in both simulation and real-world environments, with the setup illustrated in Fig. 6. We use PyBullet [34] as our simulation environment, as it provides sufficient accuracy for our open-loop push and grasp primitives, which involve simple rigid-body interactions. The evaluation consists of three tasks:

Goal-Conditioned Push-Grasp in Clutter This task assesses our framework's ability to retrieve a specific object from a cluttered scene. Following [3], [4], objects are randomly initialized,



(a) Case 1 (b) Case 2 (c) Case 3 (d) Case 4 (e) Case 5 (f) Case 6 (g) Case 7 (h) Case 8

Fig. 7. Eight special configurations of the Goal-Conditioned Push-Grasp in Tight Layouts task in the real world.

TABLE I
GOAL-CONDITIONED PUSH-GRASP IN CLUTTER IN SIMULATION

Method	10 Objects		15 Objects		20 Objects		25 Objects	
	GSR (%)	ME (%)	GSR (%)	ME (%)	GSR (%)	ME (%)	GSR (%)	ME (%)
Xu *et al.* [7]	43.5	49.9	33.9	41.7	25.6	40.8	24.6	42.5
Wang *et al.* [8]	54.7	42.9	47.9	28.5	42.2	27.9	39.2	27.7
Ren *et al.* [33]	54.1	49.3	49.3	57.0	49.0	43.0	48.1	28.9
[8] (Grasp) + Ours (Push)	56.1	54.5	50.2	55.1	55.2	33.0	49.5	34.7
[8] (Push) + Ours (Grasp)	90.6	72.9	89.0	**77.1**	87.5	**79.4**	82.4	**81.3**
Ours (non-equi + data aug)	81.2	49.8	82.5	49.8	81.0	56.3	74.6	55.9
Ours	**97.0**	**77.6**	**95.1**	69.0	**95.0**	65.2	**92.0**	57.2

All methods are allowed a maximum of 5 push attempts per target object. For each object count setting, testing is conducted using 4 random seeds, each with n rounds, where n is set to 300 divided by the object count (e.g., $n = 30$ for 10 objects). At the beginning of each round, SAM2 generates masks for all objects in the initial scene. Then, each mask, with its corresponding object, is sequentially selected as the target from the initial scene state. SAM2 is continuously used to track the target and other object masks during pushing, and a grasp is attempted if the grasp score is above the threshold or the limit of 5 pushes is reached. After each grasp attempt, the environment is reset to its initial state before proceeding to the next target. Final results are averaged over the 4 seeds.

but with one designated as the target. The robot performs push actions if needed before grasping the target.

Clutter Clearing. This task evaluates the ability to clear an entire scene without any predefined target or grasp sequence. The setup follows the previous task.

Goal-Conditioned Push-Grasp in Tight Layouts. Fig. 7 shows the task configuration. Objects are arranged in challenging geometric configurations (e.g., tight clusters, narrow gaps). This is a hard task because the robot must push strategically to create graspable space in a constrained environment.

C. Evaluation Metrics and Baselines

We use three evaluation metrics: **Grasp Success Rate (GSR)**, the ratio of successful grasps to total grasp attempts; **Declutter Rate (DR)**, the proportion of grasped objects relative to the total number of objects; and **Motion Efficiency (ME)** [5], the fraction of grasp actions among all executed actions. GSR is used for all tasks, with DR applied to clutter-clearing and ME to goal-conditioned tasks.

Our method are compared with three baselines: (1) **Xu et al. [7]**, a goal-conditioned push-grasp framework that utilizes multi-stage training to jointly optimize push and grasp action prediction. (2) **Wang et al. [8]**, an extension of [7] that improves performance by relaxing the constraints on Q-value selection and using object masks to guide actions. (3) **Ren et al. [33]** simplify task coordination with a two-stage training framework (goal-agnostic followed by goal-conditioned) and propose a bifunctional network that produces accurate, high-resolution Q-value maps to enhance sample efficiency. In addition, we introduce three ablation variants to highlight our framework design. The first integrates the grasp module from [8] into our framework, while the second applies our GraspNet within the framework of [8]. The third replaces the equivariant network with non-equivariant counterparts, trained with data augmentation.

D. Comparison With Baseline Methods in Simulation

We report the comparison result for the **Goal-conditioned Push-Grasp in Clutter** task in Table I. Our method achieves the best performance, significantly outperforming all baselines. On average, across all the settings with different number of objects, it surpasses the best baseline [33] by **44.7%** in GSR. The first two variations (Table I, row 3 and 4) show that integrating our approach into existing baselines further improves their performance, which highlights our design's effectiveness. However, our PushNet within the framework of [8] does not yield significant improvement over the original method. This is likely because, while PushNet successfully creates graspable space, the baseline grasp module lacks sufficient capability to retrieve targets. The third variant serves two purposes: it first proves the advantage of equivariant networks over non-equivariant counterparts with data augmentation, and it further validates the effectiveness of our train pipeline compared to baseline training strategies. Although our method's ME is similar to baselines, this is expected, as additional push actions are necessary to ensure more successful grasps.

We also conduct an ablation study for this task, as shown in Fig. 8. The bar chart compares the improvement in GSR with and without push actions. Our PushNet improves GSR by approximately 12% in highly cluttered environments. Additionally, we observe that the push module in Wang *el al.* [8] contributes little to improving GSR, whereas integrating our PushNet leads to a more significant improvement.

Table II shows the results of the **Clutter Clearing** task. Although this task is target-agnostic, push actions remain beneficial in cluttered environments. Since there is no specific target, we

TABLE II
CLUTTER CLEARING IN SIMULATION

Method	10 Objects		15 Objects		20 Objects		25 Objects	
	GSR (%)	DR (%)	GSR (%)	DR (%)	GSR (%)	DR (%)	GSR (%)	DR (%)
Xu *et al.* [7]	60.8/60.6	40.7/40.8	57.5/56.2	29.3/27.2	55.2/51.1	18.9/17.9	51.0/51.4	13.8/14.6
Wang *et al.* [8]	56.0/54.1	38.9/35.2	59.5/59.3	31.6/31.0	59.1/57.6	23.7/22.5	54.6/60.2	16.0/23.8
Ren *et al.* [33]	53.1/60.0	40.9/47.2	51.4/57.6	23.5/21.1	54.2/51.9	23.1/20.3	56.0/48.7	19.2/19.9
Ours	**83.0/97.5**	**69.2/93.9**	**83.3/97.6**	**62.0/91.1**	**83.6/97.7**	**53.2/91.1**	**80.4/97.8**	**35.5/87.8**

Each result is reported in a *without / with* push action format to evaluate the effectiveness of pushing. The evaluation follows the Goal-conditioned Push-Grasp in Clutter protocol, where 4 seeds are used, each running n rounds, with $n = 300$/object counts. A maximum of 5 push attempts is allowed. Final results are averaged over the 4 seeds.

TABLE III
REAL-WORLD GOAL-CONDITIONED PUSH-GRASP IN TIGHT LAYOUTS RESULTS, REPORTED AS "SUCCESSFUL ITERATIONS / TOTAL ITERATIONS"

Method	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
Xu *et al.* [7]	9/10	4/10	4/10	5/10	4/10	3/10	4/10	4/10
Wang *et al.* [8]	6/10	7/10	6/10	5/10	3/10	4/10	4/10	5/10
Ours	**10/10**	**10/10**	**9/10**	**8/10**	**7/10**	**10/10**	**9/10**	**8/10**

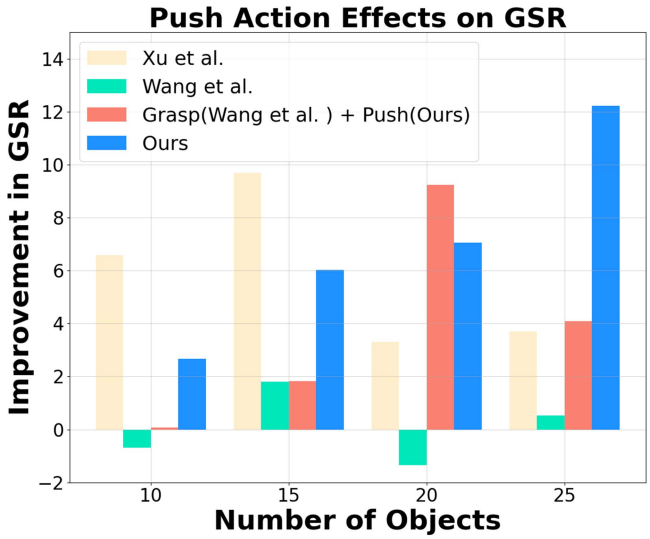


Fig. 8. Improvements in GSR with and without push actions, measured as the difference between 5 pushes and 0 pushes.

TABLE IV
REAL WORLD GOAL-CONDITIONED PUSH-GRASP IN CLUTTER COMPARISON RESULTS

Method	GSR(%)	ME(%)
Xu *et al.* [7]	40 (40/100)	27.3 (100/367)
Wang *et al.* [8]	51 (51/100)	26.8 (100/373)
Ours	**86 (86/100)**	**38.6 (100/259)**

GSR is reported as "successful grasps / total attempts", while ME is defined as "total grasp attempts / total actions".

use the object with the highest score from GraspNet as the target object for each step. The results show that our method's grasping capability exceeds all baselines by a large margin in both with and without push actions. Notably, even without push actions, our method consistently outperforms all baselines that employ pushing, across all settings with different numbers of objects. This highlights the strong capacity of our GraspNet to handle cluttered scenes. Furthermore, when push actions are enabled, our method achieves additional improvements. The magnitude of this improvement is significantly greater than that observed in any of the baselines, demonstrating the strong contribution of our PushNet in creating graspable space.

E. Real World Experiments

We conduct a large-scale real-world evaluation that far exceeds the number of trials in prior baseline studies. This extensive setup reduces the influence of randomness and increases the reliability of our results. To assess the performance of our method, we evaluate it on two tasks: **Goal-conditioned Push-Grasp in Clutter** and **Goal-Conditioned Push-Grasp in Tight Layouts**. The trained model is directly transferred from simulation to the real-world environment without any fine-tuning.

The **Goal-conditioned Push-Grasp in Clutter** task involves grasping randomly selected targets from a set of 20 household objects placed randomly in the workspace. The real-world setup and object sets are shown in Fig. 6(b) and (c). The experimental protocol follows the simulation setup. Each run consists of attempting to retrieve five target objects, with the scene randomly rearranged after each grasp to create a new cluttered layout for the next target. Each method is evaluated over 20 runs (i.e., 100 target objects in total). The target object's mask is still tracked via SAM2. Table IV presents the results, comparing our method with several baselines. Our EPG significantly outperforms all baselines by at least **35%** in GSR. The primary failure cases are: 1) inaccurate object masks from SAM2, which further affect PushNet and CriticNet outputs; 2) imprecise grasp poses predicted by the GraspNet. Despite these challenges, our method demonstrates strong overall stability.

The configuration of the **Goal-Conditioned Push-Grasp in Tight Layouts** task is in Fig. 7. It contains eight different cases, each with a varying number of small boxes placed in specific positions. The objective is to grasp the **yellow box**, which is consistently placed at the center of surrounding boxes. These tasks are **unseen during training** and require effective strategies to solve, placing a strong demand on the generalization ability. For each case, experiments are conducted over 10 iterations, where each iteration involves a randomized scene rotation and a different arrangement of the boxes. The results in Table III indicate that despite increasing task complexity, our method

consistently outperforms the baselines while maintaining stable performance.

V. Conclusion and Limitation

This letter introduces the **Equivariant Push-Grasp (EPG)** Network, a goal-conditioned grasping method that incorporates push actions to improve performance. EPG leverages $\mathrm{SE}(2)$-equivariance to enhance sample efficiency and generalization. We also propose a flexible training framework that optimizes PushNet using grasp score differences as rewards, avoiding manually designed reward functions and complex alternating training. Extensive experiments show that EPG consistently outperforms strong baselines across various tasks and settings.

However, our method has several limitations. First, it operates in an open-loop manner with fixed push distances and no force feedback, which can lead to imprecise actions in contact-sensitive scenarios. Incorporating closed-loop control with tactile sensing is a promising direction for future work. Second, EPG is limited to 4-DoF, which is sufficient for tabletop settings but does not generalize well to more complex 6-DoF scenarios. Nonetheless, our equivariant design can be naturally extended to support full $\mathrm{SE}(3)$ action spaces with minimal architectural changes. Third, EPG assumes a single-view input. In more complex or occluded scenes, our framework can generalize to multi-view inputs by fusing observations into 3D representations. Finally, EPG may require manual selection of target masks for consistency, which is inconvenient. We plan to integrate vision-language models for automatic mask generation.

Acknowledgment

The authors would like to thank all the reviewers for their helpful comments and feedback, which greatly strengthened the overall manuscript.

References

[1] B. Lim, J. Kim, J. Kim, Y. Lee, and F. C. Park, "EquiGraspFlow: SE(3)-equivariant 6-DoF grasp pose generative flows," in *Proc. 8th Annu. Conf. Robot Learn.*, 2024, pp. 5067–5086.

[2] M. Breyer, J. J. Chung, L. Ott, R. Siegwart, and J. Nieto, "Volumetric grasping network: Real-time 6 DOF grasp detection in clutter," in *Proc. Conf. robot learn.*, 2021, pp. 1602–1611.

[3] H. Huang, D. Wang, X. Zhu, R. Walters, and R. Platt, "Edge grasp network: A graph-based SE (3)-invariant approach to grasp detection," in *Proc. IEEE Int. Conf. Robots Automat.*, 2022, pp. 3882–3888.

[4] B. Hu et al., "OrbitGrasp: $SE(3)$-equivariant grasp learning," in *Proc. Conf. robot learn.*, 2025, pp. 2456–2474.

[5] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4238–4245.

[6] B. Tang, M. Corsaro, G. Konidaris, S. Nikolaidis, and S. Tellex, "Learning collaborative pushing and grasping policies in dense clutter," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 6177–6184.

[7] K. Xu, H. Yu, Q. Lai, Y. Wang, and R. Xiong, "Efficient learning of goal-oriented push-grasping synergy in clutter," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6337–6344, 2021.

[8] Y. Wang, K. Mokhtar, C. Heemskerk, and H. Kasaei, "Self-supervised learning for joint pushing and grasping policies in highly cluttered environments," in *Proc. 2024 IEEE int. conf. robot. automat. (ICRA)*, 2022, pp. 13840–13847.

[9] J. Mahler et al., "Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Proc. Robot. Sci. Syst.*, Cambridge, Massachusetts, 2017, doi: 10.15607/RSS.2017.XIII.058.

[10] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," in *Proc. Robot. Sci. Syst.*, Pittsburgh, Pennsylvania, 2018, doi: 10.15607/RSS.2018.XIV.021.

[11] X. Zhou, X. Lan, H. Zhang, Z. Tian, Y. Zhang, and N. Zheng, "Fully convolutional grasp detection network with oriented anchor box," in *Proc. 2018 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2018, pp. 7223–7230.

[12] S. Kumra, S. Joshi, and F. Sahin, "Antipodal robotic grasping using generative residual convolutional neural network," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 9626–9633.

[13] V. Satish, J. Mahler, and K. Goldberg, "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1357–1364, Apr. 2019.

[14] X. Zhu, D. Wang, O. Biza, G. Su, R. Walters, and R. Platt, "Sample efficient grasp learning using equivariant models," in *Proc. Robot., Sci. Syst.*, 2022.

[15] D. Wang, J. Y. Park, N. Sortur, L. L. Wong, R. Walters, and R. Platt, "The surprising effectiveness of equivariant models in domains with latent symmetry," in *Proc. 11th Int. Conf. Learn. Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=P4MUGRM4Acu

[16] H. Huang, D. Wang, R. Walters, and R. Platt, "Equivariant transporter network," *Robot. Sci. Syst.*, 2022.

[17] A. Simeonov et al., "Neural descriptor fields: SE(3)-equivariant object representations for manipulation," in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 6394–6400.

[18] B. Eisner, Y. Yang, T. Davchev, M. Vecerik, J. Scholz, and D. Held, "Deep se (3)-equivariant geometric reasoning for precise placement tasks," in *Proc. Twelfth Int. Conf. Learn. Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=2inBuwTyL2

[19] H. Ryu, H.-i. Lee, J.-H. Lee, and J. Choi, "Equivariant descriptor fields: SE(3)-equivariant energy-based models for end-to-end visual robotic manipulation learning," in *Proc. 11th Int. Conf. Learn. Representations*, 2022.

[20] D. Wang et al., "A general theory of correct, incorrect, and extrinsic equivariance," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, pp. 40006–40029.

[21] C. Tie et al., "ET-seed: Efficient trajectory-level SE(3) equivariant diffusion policy," in *Proc. Int. Conf. Learn. Representations*, 2024.

[22] D. Wang, R. Walters, and R. Platt, "SO(2)-equivariant reinforcement learning," in *Proc. Int. Conf. Learn. Representations*, 2022.

[23] D. Wang, R. Walters, X. Zhu, and R. Platt, "Equivariant Q learning in spatial action spaces," in *Proc. Conf. Robot Learn.*, 2022, pp. 1713–1723.

[24] H. H. Nguyen, A. Baisero, D. Klee, D. Wang, R. Platt, and C. Amato, "Equivariant reinforcement learning under partial observability," in *Proc. Conf. Robot Learn.*, 2023, pp. 3309–3320.

[25] H. Huang, O. Howell, D. Wang, X. Zhu, R. Walters, and R. Platt, "Fourier transporter: Bi-equivariant robotic manipulation in 3D," in *Proc. Twelfth Int. Conf. Learn. Representations*, 2024, *arXiv:2401.12046*.

[26] D. Wang et al., "Equivariant diffusion policy," in *Proc. Int. Conf. Learn.*, 2025, pp. 48–69.

[27] C. Gao et al., "Riemann: Near real-time SE(3)-equivariant robot manipulation without point cloud segmentation," in *Proc. Int. Conf. Learn.*, 2025, pp. 2164–2182.

[28] N. Raviet al., "SAM 2: Segment anything in images and videos," in *Proc. Thirteenth Int. Conf. Learn.*, 2025. [Online]. Available: https://openreview.net/forum?id=Ha6RTeWMd0

[29] D. Wang, M. Jia, X. Zhu, R. Walters, and R. Platt, "On-robot learning with equivariant models," in *Proc. Conf. Robot Learn.*, 2023, pp. 1345–1354.

[30] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[31] G. Cesa, L. Lang, and M. Weiler, "A program to build E(N)-equivariant steerable CNNs," in *Proc. Int. Conf. Learn. Representations*, 2022.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[33] D. Ren, S. Wu, X. Wang, Y. Peng, and X. Ren, "Learning bifunctional push-grasping synergistic strategy for goal-agnostic and goal-oriented tasks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 2909–2916.

[34] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016.