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

Sym-Servo: Disambiguate Symmetric Object Pose by End-to-End Optimal Visual Servo

Shuxin Li[1], Anzhe Chen[1], Haojian Lu[1], Rong Xiong[1,2], Yue Wang[1]

***Abstract*— Controlling symmetric objects is an indispensable but challenging task in robotic manipulation. Mainstream perception-action frameworks rely on accurate 6D pose estimation to guide the controller. However, the majority of existing 6D pose estimation methods for symmetric objects are designed to output a single pose, which can flicker between multiple equivalent solutions across consecutive frames, leading to instability in the control loop. While some approaches can output multiple hypotheses to represent the ambiguity, above methods generally cannot achieve model-free manner and strong generalization simultaneously. In this paper, we formulate the problem from a multi-solution task in pose space to an end-to-end visual servo task that admits a unique optimal solution. We propose a visual servo framework Sym-Servo. Sym-Servo uses a joint learning mechanism where a deterministic policy is trained with a diffusion-based generator to encourage the shared vision encoder to learn a symmetry-aware representation, and the policy is then refined via reinforcement and self-imitation learning to produce an efficient and stable final policy. We validate Sym-Servo with simulations and real-world experiments, demonstrating its efficiency and generalization in controlling symmetric objects in a model-free manner.**

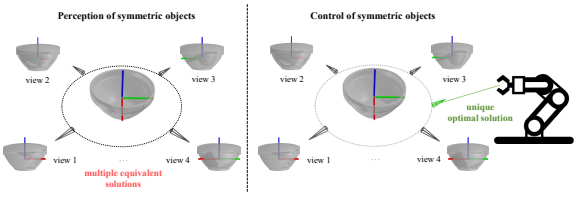
I. Introduction

Accurate 6D pose estimation is a fundamental prerequisite for robotic tasks such as grasping, manipulation, and assembly [1], [2]. In scenarios such as industrial automation and domestic services, robots are required to perceive object poses with high precision. However, a wide range of commonly encountered objects, ranging from flanges and nuts in industrial production to bowls and bottles in daily life, exhibit inherent geometric symmetries. Such symmetries introduce the fundamental challenge of *multi-hypothesis issue* [2] in 6D pose estimation, where a single visual observation may correspond to multiple, or even infinitely many, physically valid poses.

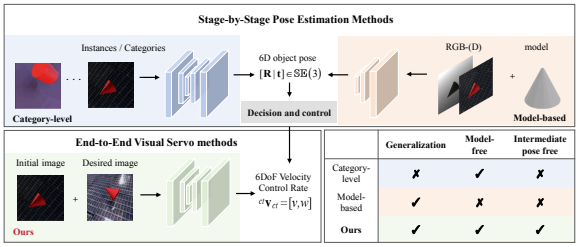
Pose ambiguity is not merely a challenge at the perception level, it poses a severe threat to downstream robotic control tasks. For symmetric objects, an effective control strategy typically requires selecting the most optimal action from multiple valid possibilities in robotic manipulation. This process first involves identifying all symmetry-equivalent poses and then choosing the one minimizes joint movement or maximizes reachability.

However, conventional 6D pose estimation methods are often insufficient in handling this challenge. While some studies have focused on the symmetric objects, most of them only produce a single solution. Certain methods introduce symmetry-aware metrics during network training [3], [4], [5], [6] or learning many-to-many correspondences [7] to make the network converge to the correct state, but the single pose they output can flicker between multiple equivalent solutions across consecutive control frames. Such discontinuities in perception can be directly propagated to the control commands, resulting in oscillations and instabilities in the trajectory of the robot's end-effector. Other methods do attempt to explicitly model the set of symmetry-equivalent poses to output a distribution rather than a single estimation [2], [8]. These methods provide the necessary hypotheses for control, but their applicability is also limited. Above methods are either struggle to generalize to novel, unseen symmetric objects, or they rely heavily on precise 3D CAD models to define the symmetries, making them unsuitable for model-free, open-world scenarios. In other words, simultaneously handling symmetric objects, generalizing to novel instances, and being model-free remains a challenge in the current research. This brings up a question: *Can we enable more stable control of symmetric objects while retaining generalization ability in a model-free manner?*

From another perspective, when the pose estimation problem is considered jointly with the control objective, the situation changes fundamentally. Although symmetric objects may have multiple equivalent poses, in a specific manipula-



(a) Perception problem in pose space vs Control problem in task space



(b) Comparison of different control frameworks

Fig. 1: We formulate the problem of controlling symmetric objects as a visual servo task to convert its multi-solution in the pose space into a single solution in task space.

[1]Zhejiang University. [2]Zhejiang Humanoid Robot Innovation Center. Yue Wang is the corresponding author: wangyue@iipc.zju.edu.cn. This work was supported by the National Natural Science Foundation of China (Grant No. 62522317 and U24A20128), Research and Development Project of Zhejiang Province under Grant No. 2025C01205(SD2), and Zhejiang Provincial Natural Science Foundation of China under Grant No. LD25F030001.

tion task there often exists a unique optimal trajectory to reach the goal. Therefore, for closed-loop control tasks, the key question should not be *what is the exact pose of the object*, which is inherently ambiguous, but rather *how can the robot reach the desired visual state in an optimal way*, which admits a unique solution. In this way, the pose ambiguity present in the pose space can be naturally resolved in the task space, transforming it into a single-solution problem and opening new possibilities for achieving generalization to symmetric objects under model-free conditions.

Building on this idea, we formulate the problem as an end-to-end visual servo task. By learning a visuomotor policy that directly maps the current and desired images to the optimal action, the ambiguity introduced by object symmetries can be addressed implicitly. While reinforcement learning (RL) is ideal for discovering optimal policies, it is unstable to train from scratch. A pretrained model is essential, but imitation learning for symmetric objects is also difficult because of multimodal distributions in ground-truth actions. To resolve this, we propose a three-stage training framework. In the first stage, we address the multimodal distributions problem by jointly training a regression policy with a diffusion-based generator, which encourages the model to learn a symmetry-aware representation. This is followed by a RL stage to improve efficiency. In the third stage, we use self-imitation learning, where the policy learns to stop by imitating its own successful trajectories with actions close to desired set to zero, to refine for stability. This refined behavior is then stabilized through a second round of RL with an increased weight of action regularization reward. Finally, at inference time, given only the current and desired images, our approach can directly produce a 6DoF velocity control rate to the goal.

Our contributions can be summarized as follows:

- We formulate symmetric object's manipulation as an end-to-end visual servo problem rather than a pose estimation problem, enabling direct mapping from current and desired images to velocity control rate.
- We introduce a joint learning framework that integrates the deterministic regression policy with a diffusion-based probabilistic generator. Reinforcement learning and self-imitation learning are further employed to fine-tune the regression policy on symmetric objects.
- We validate our approach both in simulation and real-world, demonstrating stable control of symmetric objects under model-free conditions, while preserving strong generalization ability.

II. Related Works

6D Pose Estimation and Symmetry Problem. Existing 6D object pose estimation approaches are broadly categorized into model-based and model-free methods. Model-based methods rely on CAD models, solving pose either via direct regression [5], [9], 2D–3D/3D–3D correspondences [1], [10], [11], [12], or template matching [13], [14], [15], [16]. Although accurate, this dependency limits their open-world applicability. Alternatively, model-free methods [17], [18], [19], [20] use reference images and few-shot learning pipelines, improving flexibility but often struggling in textureless or occluded scenes, and showing limited generalization.

A challenge for both categories is multi-hypothesis issue introduced by symmetry. Prior work has addressed this by introducing symmetry-aware loss functions [3], [4], [5], [6], predicting multiple hypotheses [2], [8], or learning dense surface correspondences to encode symmetry implicitly [7] and [21] learns a direct mapping to a latent rotation representation, making them inherently symmetry-agnostic. However, these solutions either exhibit limited generalization ability to unseen symmetric objects or depend on models, leaving robust and model-free handling of symmetry a problem.

Visual Servo. Classical visual servo techniques, such as Image-Based Visual Servo (IBVS) and Position-Based Visual Servo (PBVS), have been foundational in robot control. IBVS uses 2D image features to control motion, while PBVS relies on the 3D pose difference between the current and desired positions. Despite their effectiveness, these methods struggle with challenges like Jacobian singularities, local minima [22], or sensitivity to the errors of camera intrinsic.

To overcome these limitations, learning-based approaches have emerged, shifting the focus to deep neural networks for feature extraction and pose estimation. Keypoint-based methods [23], [24], [25], which use neural networks to predict keypoints or optical flow, improve robustness but are still hindered by the need for accurate matching, especially in textureless environments. End-to-end learning methods [26], [13], [27], [28] aim to learn the entire process, improving robustness to noise and image occlusions but struggling to generalize to unseen scenes. [29] uses dense probabilistic matching, which provides strong robustness in textureless conditions and strong generalization ability. However, it still struggles to capture the multimodal action distributions arising from object symmetries.

Imitation and Reinforcement Learning for Robotic Manipulation. While Imitation Learning (IL) can rapidly learn a competent initial policy from demonstrations, methods like Behavioral Cloning (BC) often suffer from compounding errors due to covariate shift [30], [31]. Reinforcement Learning, on the other hand, can discover optimal policies through trial-and-error, but is sample-inefficient for visuomotor control from scratch [32]. A highly effective paradigm is therefore to combine these approaches, for instance, by augmenting the replay buffer with expert data [33] or by using demonstrations to shape the reward function [34].

III. Methods

The task is defined as generating the velocity control rate $[{}^c\boldsymbol{\nu}_c; {}^c\boldsymbol{\omega}_c]$ that guide the robot from the current observation $\mathbf{I}_c$ to match a given goal image $\mathbf{I}_d$. Conventional pipelines rely on explicit pose estimation followed by control, which suffers from multi-hypothesis issue in symmetric objects. We address this challenge by formulating the problem as an end-to-end visual servo task and learning a visuomotor policy that directly maps $\{\mathbf{I}_c, \mathbf{I}_d\}$ to velocity command. An overview of our framework is shown in Fig. 2, which is described in the following subsections.

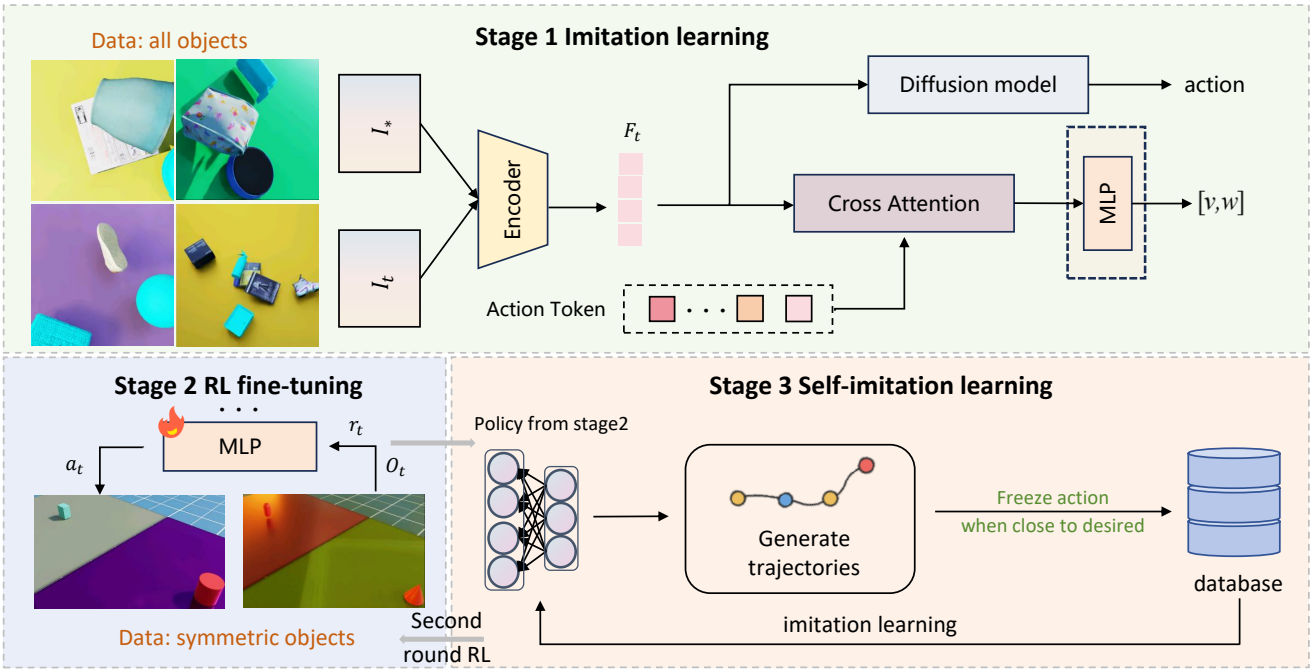


Fig. 2: Overview of our framework. Our policy is trained through three stages. (1) Imitation Learning Pretraining: The policy is trained using a joint learning mechanism based on CNSv2 with auxiliary diffusion model to learn a more robust representation. (2) RL Fine-tuning: The policy is then optimized for more efficient action and higher accuracy via reinforcement learning. (3) Final Refinement: A final stage combining Self-Imitation Learning and another RL encourages more stable terminal behavior, reducing oscillations near the goal.

A. *Joint Learning with Diffusion Model*

In this section, we describe how we integrate the diffusion model with the deterministic regression policy to improve visual servo performance. The key idea is to enhance the encoder's ability to capture symmetry-aware features, addressing the ambiguity introduced by symmetric objects.

Regression-based deterministic visual servo model. To provide context, we briefly review the core architecture of CNSv2 [29], which forms the backbone of our approach. It consists of a probabilistic matching network and a regression policy head.

The matching network comprises the foundation vision model (ViT), the transformer with several self/cross attention layers, and the module that computes the translation-equivariant probabilistic matching representation from the score matrix $\mathcal{S}$. The coarse features extracted by the process will be sent to several self attention layers with the fine features extracted by CNN layers to obtain the final matching features $\mathbf{F}$, which will be shared between two policy heads.

The regression head has a feature fusion module with cross-attention layers and a Multi-Layer Perceptron (MLP) head to regress the final velocity command. The training loss is defined as an L1-loss on the log norm of the velocity and a cosine similarity loss on its direction:

$$\mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{norm}} + \lambda_d \mathcal{L}_{\text{dir}} \tag{1}$$

where λ_d is the loss weight.

Diffusion-based Probabilistic action Generator. We leverage a Denoising Diffusion Implicit Model (DDIM) [35] to implicitly model the multimodal action space. It learns to model a distribution $p(x|\mathbf{c})$ conditioned on input $\mathbf{c}$, where x represents the relative transformation from current pose to desired pose of the camera on the end-effector and $\mathbf{c}$ is the probabilistic matching features $\mathbf{F}$ of current and desired images. In the forward process, the initial data x_0 is gradually corrupted to noise x_T through a predefined noise schedule α_t, where $\alpha_t \in (0, 1]$ decreases monotonically. For any timestep t, the noisy state x_t can be directly computed from x_0:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{1}) \tag{2}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ controls the noise intensity at each time step. The neural network, denoted as $\hat{\epsilon} = \epsilon_\theta(x_t, t, \mathbf{c})$, is designed to model the denoising process. It takes the noisy state x_t, the noise level t, and the condition $\mathbf{c}$ as inputs and attempts to predict the noise component ϵ. The training objective is to minimize the L1 loss:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{x_0, \mathbf{c}, \epsilon, t}\left[\|\epsilon - \epsilon_\theta(x_t, t, \mathbf{c})\|_1\right] \tag{3}$$

Joint Learning Mechanism. The deterministic head and the diffusion action generator are trained end-to-end. The parameters of the shared feature backbone are updated by a composite loss function:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{reg}} + \lambda \mathcal{L}_{\text{diff}} \tag{4}$$

where λ is a hyperparameter that balances the contribution of the two tasks. To ensure the model learns a generalizable policy, we follow the training principle of CNSv2 and make no specific distinction between symmetric and asymmetric objects during this training stage.

B. Reinforcement Learning Fine-tuning for Symmetric Objects

While our joint learning mechanism enables the feature encoder with an implicit understanding of symmetry, the deterministic MLP head must still map this multimodal representation to a single action. This can lead to suboptimal behavior for symmetric objects. For instance, the policy might average between two valid rotational paths, resulting in an invalid action or stay put, or it may exhibit hesitation by switching between different potential trajectories from one timestep to the next. To resolve this, we introduce a RL fine-tuning stage, which allows the policy to learn a consistent and cost-efficient strategy through direct environmental interaction.

RL Formulation. We formalize the fine-tuning problem as a finite-horizon discounted Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$. The agent's goal is to learn a policy $\pi(a_t|s_t)$ that maximizes the expected cumulative discounted reward.

1) State Space. To leverage the representation power of the pretrained backbone, the state s_t at time t is defined as the matching feature $\mathbf{F}$ extracted by the frozen shared encoder from the current and desired images. The backbone thus serves as a fixed, high-quality state encoder, significantly simplifying the learning problem for the RL algorithm.

2) Action Space. The action a_t is the 6-DoF velocity control rate $[\boldsymbol{v}; \boldsymbol{\omega}]$ output by the policy's terminal MLP head.

3) Reward Function. The reward function is carefully designed to balance task completion, efficiency, and robustness in challenging symmetric object scenarios:

IoU reward. We adopt the Intersection over Union (IoU) between the current image $\mathbf{I_c}$ and the desired image $\mathbf{I_d}$ as a main reward signal:

$$r_{\text{IoU}} = \text{IoU}(\mathbf{I_c}, \mathbf{I_d}) - 1 \tag{5}$$

which encourages the agent to continuously increase the overlap between the current and desired object masks.

Mask-Center penalty. Since IoU may remain zero for several steps at the beginning of an episode, we introduce an auxiliary penalty based on the distance between the object mask centers:

$$r_{\text{center}} = -\lambda_c \, \|c(\mathbf{I_c}) - c(\mathbf{I_d})\|_2 \tag{6}$$

where $c(\mathbf{I})$ denotes the 2D center of the mask in image $\mathbf{I}$.

Visibility penalty. If the object is out of the camera's field of view, we apply a constant penalty:

$$r_{\text{vis}} = \begin{cases} -\lambda_v, & \text{if the object is out of view} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Action Regularization. To encourage smooth and natural motion, we regularize the executed actions to suppress abrupt and unstable trajectories:

$$r_{\text{act}} = -\lambda_a \|a_t\|_2 \tag{8}$$

Task completion reward. Once the IoU exceeds a threshold (e.g., 0.98), a large sparse reward is provided to signal task completion:

$$r_{\text{succ}} = \begin{cases} R_{\text{succ}}, & \text{if IoU}(\mathbf{I_c}, \mathbf{I_d}) > 0.98 \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

The overall reward at each timestep t is defined as:

$$r_t = r_{\text{IoU}} + r_{\text{center}} + r_{\text{vis}} + r_{\text{act}} + r_{\text{succ}} \tag{10}$$

Fine-tuning Strategy. We use the Proximal Policy Optimization (PPO) [36] algorithm to optimize the policy. To stabilize the initial phase of training, we first warm up the value network for a small number of epochs before starting the full PPO updates. Crucially, this fine-tuning is performed in a parameter-efficient manner where only the parameters of the terminal MLP head are updated during RL. This allows the model to rapidly adapt its final decision-making process to the distinction of symmetric objects without risking catastrophic forgetting of its generalization.

The fine-tuning is conducted exclusively in simulation environments populated with a set of challenging symmetric geometries and objects (e.g., cone, cube, cylinder). This focused training enables the policy to specialize in resolving the specific ambiguities and inefficiencies associated with these types of symmetry axis.

C. Self-Imitation Learning and RL Refinement

Although the RL fine-tuning described above improves performance on symmetric objects, we observe that the policy tends to be instable when approaching the desired state. As the agent nears the goal, the reward function continues to incentivize marginal improvements, which can lead to oscillations or unnecessary rotations around an object's axis of symmetry. The policy performs well at reaching the goal, but not at staying there. To address this, we introduce a final, two-stage refinement process to encourage stable terminal behavior.

Self-Imitation Learning from Successful Trajectories. The first stage uses Self-Imitation Learning (SIL) to teach the policy how to remain stable near the goal. This is achieved by training the agent to imitate its own successful past experiences.

We use the policy obtained from the RL fine-tuning stage to generate a large set of trajectories in the symmetric object environments. We then create an expert dataset $\mathcal{D}_{\text{expert}}$ by filtering these trajectories. A trajectory $\tau = \{(s_t, a_t)\}_{t=0}^{T}$ is regarded successful and included in the dataset if at any point its IoU exceeds a high success threshold, i.e., $\exists t' \in [0, T]$ such that $\text{IoU}(s_{t'}) > 0.95$. For each successful trajectory τ in our dataset, we construct a corresponding expert trajectory τ_{expert}. We first identify the earliest timestep t_{success}, where the success condition is met. The expert trajectory is then defined as:

$$a_t^{\text{expert}} = \begin{cases} a_t, & \text{if } t < t_{\text{success}} \\ \mathbf{0}, & \text{if } t \geq t_{\text{success}} \end{cases} \tag{11}$$

This explicitly provides the supervision signal: "if the goal is reached, the optimal action is to stop." Keeping the feature backbone frozen, we fine-tune the regression policy head on

this expert dataset via supervised learning. The loss function is same as CNSv2:

$$\mathcal{L}_{\text{SIL}} = \mathbb{E}_{\tau_{\text{expert}} \in \mathcal{D}_{\text{expert}}} \left[\sum_{t=0}^{T} (\mathcal{L}_{\text{norm}} + \lambda_d \mathcal{L}_{\text{dir}}) \right] \tag{12}$$

Final RL Refinement. Training with SIL can sometimes make a policy overly conservative, potentially causing it to decelerate prematurely and slow down convergence. Therefore, we conduct a final phase of reinforcement learning to integrate the newly learned terminal stability with the policy's existing goal-reaching capabilities. Starting with the weights from the SIL-trained model, we run the PPO algorithm for a limited number of iterations using the same RL formulation as before but with a higher λ_a. This final RL stage allows the policy to find a balance, ensuring it remains efficient in reaching the target while also demonstrating more stable behavior when it is close enough to the goal. The resulting policy is capable of both rapid convergence and more stable termination.

IV. Experiments

A. Simulation Environment Setup

In simulation experiment, we randomly place a symmetric object from the GSO [37] dataset or a symmetric geometry in a scene, generating a total of 300 scenes for testing. For each scene, we randomly sample an initial pose and a desired pose, then render the corresponding images in IsaacSim. We compare our method with three pose estimation approaches: Genpose [2], SC6D [21], and FoundationPose [16], as well as a visual servo method, CNSv2 [29]. For the visual servo approach, we use the norm of the integrated motion of the predicted velocity over a unit time step (dT norm) as the criterion for termination, stopping the process once the norm is below the threshold. For the pose estimation methods, we directly move the robot to the desired pose based on the predicted pose.

Metrics: (1) IoU: The average final IoU between the object in the final image and the desired image of each episode, reported with a 95% confidence interval. (2) Success Rate at IoU Thresholds (IoU_{xx}): We define success rate at different precision levels, where IoU_{xx} represents the percentage of episodes that achieve a final IoU higher than a threshold of xx%. We report this metric for thresholds of 50%, 75%, 90%, and 95%.

B. Simulation Results

Fig. 3 shows the results in simulation environment. Our method demonstrates superior overall performance, achieving the highest mean IoU across all test scenes. In addition, we evaluated the sensitivity of pose estimation methods to the accuracy of the desired pose. We tested two scenarios: one using the ground-truth pose from the desired image (desired-gt), and another using a pose estimated from the desired image (desired-est). Despite having the lowest overall accuracy, SC6D shows a small performance drop when switching to estimated desired pose. This suggests that while its absolute pose predictions are less inaccurate, its estimation errors may be consistent across different viewpoints. When calculating

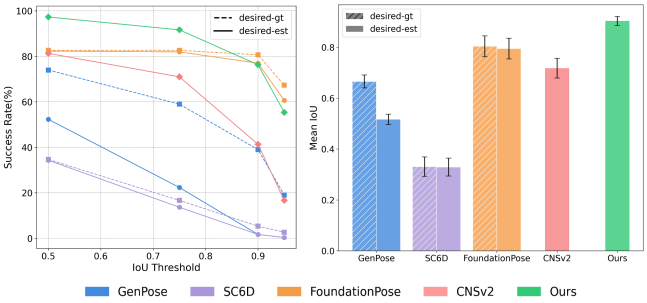


Fig. 3: Comparisons in simulation. CNSv2 and our methods are end-to-end visual servo methods. Genpose, SC6D and FoundationPose are pose estimation methods, among which Genpose and SC6D are model-free but category/instance-level methods and FoundationPose can generalize but is model-based. 'Desired-gt', 'desired-est' denotes using the ground-truth desired pose and using the estimated pose from the desired image, respectively.

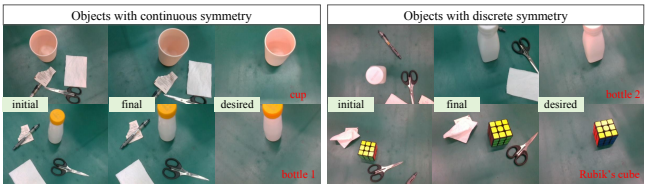


Fig. 4: Objects used in real-world experiments. The final image is generated by our method.

the relative transformation from the current to the desired pose, these consistent biases can partially cancel each other out. Conversely, the significant performance drop of GenPose suggests its errors may be highly viewpoint-dependent and inconsistent. FoundationPose's strong performance in both scenarios indicates that it is not only highly accurate but also highly consistent.

C. Real-world Environment Setup

We evaluate methods on 4 unseen symmetric objects (cup, bottle 1, bottle 2, Rubik's cube), among which two have continuous symmetry and two have discrete symmetry (Fig. 4). We sample 25 desired and initial pose pairs for each object. To test the robustness of each method in more realistic, cluttered environment, when sampling initial frames, we randomly place some distractions on the background. Here we compare our method just with FoundationPose [16] and CNSv2 [29] that can generalize to unseen objects.

We define some additional metrics for real-world experiments: (1) TT: total convergence time of one servo episode. (2) FPS: frames per second, evaluating the neural network's inference speed. (3) Episode energy: total norm of the executed transformation of one episode, evaluating the efficiency of the trajectory.

In addition to the quantitative evaluation, we also designed a qualitative grasping scene with distractions to further analyze the robustness of the compared methods.

D. Real-world Environment Results

Quantitative evaluation. As shown in Table I, our method outperforms both CNSv2 and FoundationPose in precision and efficiency. We observed cases where CNSv2 would take a long path to unnecessarily circle around the object's axis of

TABLE I: Statistics of real-world experiments. We use the same termination criterion as simulation environment. Our method achieves the highest precision and efficiency across all objects.

Obj.	Methods	IoU_{50}(%)	IoU_{75}(%)	IoU_{90}(%)	IoU_{95}(%)	IoU	TT(s)	FPS	Episode Energy
Cup	FoundationPose	21/25	21/25	11/25	4/25	0.773±0.133	6.194±0.863	1.276±0.004	1.177±0.361
	CNSv2	17/25	17/25	12/25	8/25	0.743±0.108	11.580±1.107	**36.112±0.317**	2.976±1.170
	Ours	**25/25**	**25/25**	**25/25**	**23/25**	**0.967±0.005**	**5.632±0.912**	31.670±0.291	**0.589±0.172**
Bottle 1	FoundationPose	23/25	21/25	2/25	0/25	0.766±0.091	7.516±0.911	1.274±0.003	1.712±0.389
	CNSv2	23/25	22/25	15/25	13/25	0.839±0.108	8.980±1.170	**36.733±0.516**	1.492±0.713
	Ours	**25/25**	**25/25**	**25/25**	**19/25**	**0.965±0.007**	**5.245±1.085**	31.866±0.303	**0.673±0.180**
Bottle 2	FoundationPose	**25/25**	24/25	12/25	2/25	0.878±0.022	6.908±0.911	1.271±0.004	1.486±0.378
	CNSv2	20/25	18/25	11/25	4/25	0.742±0.113	11.560±1.250	**36.631±0.602**	2.150±0.889
	Ours	**25/25**	**25/25**	**25/25**	**21/25**	**0.966±0.006**	**6.227±0.742**	31.916±0.118	**0.860±0.171**
Rubik's cube	FoundationPose	20/25	20/25	6/25	0/25	0.692±0.137	7.598±0.930	1.201±0.003	1.719±0.397
	CNSv2	24/25	24/25	17/25	7/25	0.890±0.047	11.267±0.964	**37.032±0.185**	1.578±0.630
	Ours	**25/25**	**25/25**	**23/25**	**16/25**	**0.952±0.012**	**6.873±1.377**	31.921±0.149	**0.941±0.208**

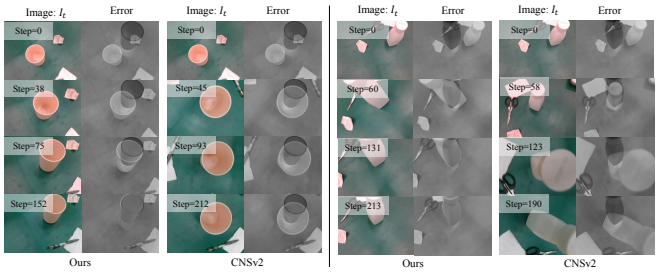


Fig. 5: The comparision of CNSv2 and our method on two real-world scenes. Our method converges to desired state efficiently while CNSv2 would take a long path to circle around the object's axis of symmetry and is hard to converge.

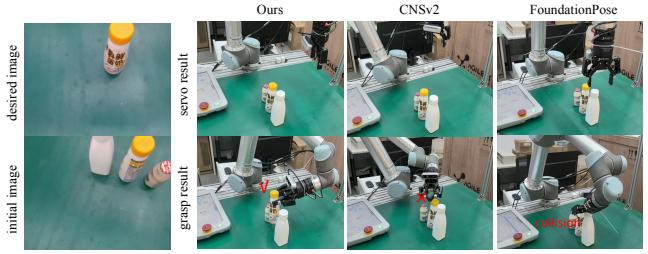


Fig. 6: Qualitative comparison in a grasping scene. The target bottle is rotated and flanked by two distractive bottles during testing. Our method achieves a successful grasp by focusing on the correct desired geometry. CNSv2 fails due to oscillatory movements caused by hesitation between aligning the bottle's texture and geometry. FoundationPose collides with a distraction during its approach.

symmetry without converging (Fig. 5). This hesitation significantly increased its episode energy. As for FoundationPose, its performance is constrained by the inability to obtain precise 3D models in real world and its open-loop control that cannot correct for estimation errors.

Qualitative Analysis. We conducted a challenging grasping experiment with a beverage bottle to visualize the robustness of our method. First, a target view of the bottle was captured. The robot was then guided via teleoperation to a successful grasp pose to establish the relative transformation between the target view and the grasp pose. For the test, the bottle was rotated, and two distractive bottles were placed around (Fig. 6).

As shown in Fig. 6, our method robustly converges to the correct grasping pose, successfully grasping the bottle despite the changes in orientation. In contrast, CNSv2 failed to grasp, which is because it exhibited oscillatory behavior as it hesitated between aligning the bottle's texture and geometry. FoundationPose converged to an equivalent but incorrect pose hypothesis where the gripper's approach direction was obstructed by a distraction, resulting in a collision.

E. Ablation Study

Training stages. As shown in Table II, transitioning from the stage1-IL to the stage2-RL policy significantly improves IoU metrics by discovering more efficient trajectories. However, we observe that the stage2-RL policy has the longest mean episode completion time and highest episode energy. While it excels at reaching the goal, it struggles with stability. It tends to make continuous small corrective movements near the goal instead of stopping. Fig. 7 has supported this behavior, where the final dT norm of stage2-RL policy is the highest. These constant adjustments delay the satisfaction of the termination criterion, thereby increasing the overall task time.

The stage3 policy, refined with SIL, addresses the instability while maintaining its high performance. The final IoU of this policy is comparable to that of stage2-RL, and it achieves the shortest mean episode completion time, demonstrating a reduction in time spent during the terminal phase. It can reduce the final dT norm (Fig. 7), thus reducing oscillations near the goal. While the episode energy is lower than that of stage2-RL, it remains slightly higher than the stage1-IL policy. Combined with Fig. 7, the reason is that the stage3 policy moves more decisively and rapidly towards the goal to achieve higher precision than the more cautious IL policy. Moreover, compared with stage1-IL policy, the policy can generate more direct trajectory and converges to the most energy-efficient desired pose (Fig. 8).

Effect of diffusion head. To analyze the impact of our proposed diffusion-based joint learning mechanism, we trained a model using the same three-stage pipeline but without the auxiliary diffusion loss during the initial pretraining. We then compare the final performance of both models after the RL fine-tuning stage. As demonstrated in Table III, the

TABLE II: The ablation study on the training stages of our framework. 'IL', 'RL1', 'RL2' denotes the trained policy of the three stages. Each statistic is obtained from 300 scenes in simulation environment.

	Policy	IoU_{50}(%)	IoU_{75}(%)	IoU_{90}(%)	IoU_{95}(%)	IoU	TT(s)	Episode energy
proposed	IL	94.33	79.00	23.67	3.00	0.788±0.021	8.678±0.795	**3.295±0.592**
	RL1	96.33	90.67	74.33	52.67	0.897±0.018	22.597±1.430	6.728±0.959
	RL2	**97.33**	**91.67**	**76.33**	**55.33**	**0.904±0.017**	**7.802±0.974**	4.581±0.796

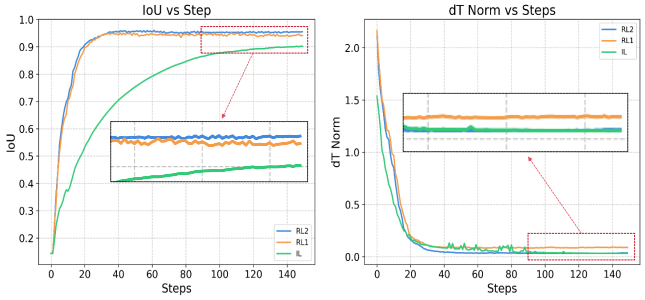


Fig. 7: We sample 30 scenes converged across all policies to visualize mean of their IoU and dT norm against steps. Each scene runs 150 steps for every policy. RL2 policy can converge rapidly to high IoU and reducing oscillations near the goal.

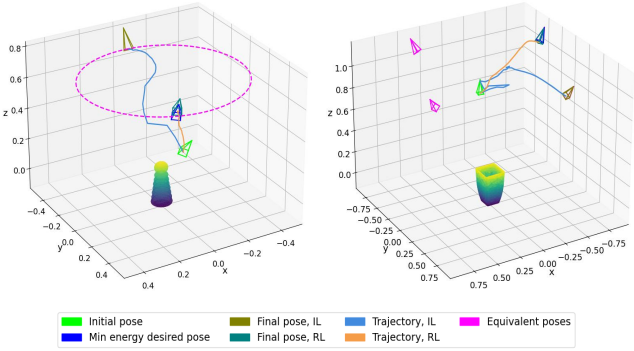


Fig. 8: Two samples of the trajectories generated by IL and RL2 on a discrete symmetric object and a continuous symmetric object. We define the min energy desired pose as the pose that minimizes the geodesic distance between the equivalent desired pose and the initial pose. For the continuous object, we use Levenberg-Marquardt algorithm to calculate the min energy desired pose.

policy pretrained with the diffusion head achieves better final performance after RL fine-tuning. The reason might be that diffusion model's ability to model multimodal distribution can encourage the feature encoder to learn a symmetry-aware representation, and this provides a superior foundation for the subsequent RL stage. Consequently, the RL agent can explore more effectively and converge to a higher-quality policy.

Effect of pretraining. We conduct two ablations to validate our pretraining and fine-tuning paradigm. In Fig. 9, When the entire network is initialized randomly and trained solely with RL, The policy will lead to 'nan' values in the action distribution, making it unable to continue trining. This highlights the difficulty of exploration and policy update in our task without a strong behavioral prior provided by imitation learning. In addition, as demonstrated in Table III, we use the pretrained feature backbone but randomly initialize the terminal MLP head before RL fine-tuning. This setup also resulted in a performance drop across all metrics. While the model benefits from a powerful vision encoder, the lack of a

TABLE III: Ablation study on key components of our framework on stage2-RL policy. '-diffusion' denotes making RL fine-tuning from the imitation policy without diffusion head and '-MLP pretrain' denotes training RL without the pretrained parameters of MLP head from stage 1.

Metrics	RL1 (base)	-diffusion	-MLP pretrain
IoU_{50}(%)	**96.33**	85.67	81.67
IoU_{75}(%)	**90.67**	72.00	74.67
IoU_{90}(%)	**74.33**	51.00	43.33
IoU_{95}(%)	**52.67**	33.67	11.33
IoU	**0.897±0.018**	0.7635±0.034	0.722±0.038
TT(s)	22.597±1.430	**16.186±1.215**	25.424±0.052
Episode energy	**6.728±0.959**	7.631±0.912	23.980±2.654

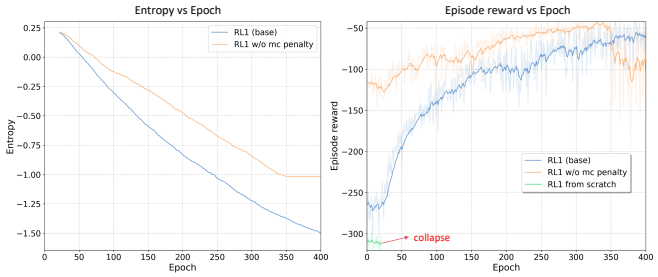


Fig. 9: Effect of mask-center penalty and pretraining on stage2 RL fine-tuning. 'RL1 w/o mc penalty': train RL without the mask-center penalty; 'RL1 from scratch': the entire network was initialized randomly to traine with RL. After several epochs, RL1 w/o mc penalty results in reward collapse. RL1 from scratch can lead to 'nan' values in the action distribution and the policy is unable to continue training.

coherent decision prior from a pretrained MLP reduced the benefits of imitation. Above two experiments confirm that a comprehensive pretrained policy is critical for stable and effective learning.

Mask-center penalty helps convergence. Our ablation study in Fig. 9 demonstrates that the policy trained without mask-center penalty suffers from unstable training process: after several epochs, its reward catastrophically drops and the corresponding entropy curve shows a slower decay followed by an abrupt flattening upon failure. This is because the sparse IoU reward provides an insufficient gradient for exploration, leading to high-entropy, random actions that eventually make the agent to a local optimum. In contrast, adding the mask-center penalty can provide a dense and clear learning signal to ensure more stable convergence in the training process.

V. Conclusion

In this work, we propose a three-stage training framework Sym-Servo to address the challenge of controlling symmetric objects in a model-free, generalizable manner. We formulate the problem from an ambiguous perception task to a control task that admits unique solution. We use a diffusion-based joint learning mechanism that enables a policy to implicitly learn

object symmetries from data. This provides a robust foundation for a subsequent refinement pipeline using reinforcement and self-imitation learning, which resolves control instabilities and optimizes for trajectory efficiency. Sym-Servo achieves superior accuracy and efficiency, offering a robust solution for high-precision visual servo of symmetric objects.

References

[1] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," *arXiv preprint arXiv:1809.10790*, 2018.

[2] J. Zhang, M. Wu, and H. Dong, "Genpose: Generative category-level object pose estimation via diffusion models," *arXiv preprint arXiv:2306.10531*, 2023.

[3] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3343–3352.

[4] G. Wang, F. Manhardt, F. Tombari, and X. Ji, "Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 16 611–16 621.

[5] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[6] N. Mo, W. Gan, N. Yokoya, and S. Chen, "Es6d: A computation efficient and symmetry-aware 6d pose regression framework," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 6718–6727.

[7] H. Zhao, S. Wei, D. Shi, W. Tan, Z. Li, Y. Ren, X. Wei, Y. Yang, and S. Pu, "Learning symmetry-aware geometry correspondences for 6d object pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 14 045–14 054.

[8] F. Manhardt, D. M. Arroyo, C. Rupprecht, B. Busam, T. Birdal, N. Navab, and F. Tombari, "Explaining the ambiguity of object detection and 6d pose from visual data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6841–6850.

[9] Z. Li, G. Wang, and X. Ji, "Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 7678–7687.

[10] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 632–11 641.

[11] Y. He, H. Huang, H. Fan, Q. Chen, and J. Sun, "Ffb6d: A full flow bidirectional fusion network for 6d pose estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3003–3013.

[12] K. Park, T. Patten, and M. Vincze, "Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 7668–7677.

[13] V. Balntas, A. Doumanoglou, C. Sahin, J. Sock, R. Kouskouridas, and T.-K. Kim, "Pose guided rgbd feature learning for 3d object pose estimation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3856–3864.

[14] D. Cai, J. Heikkilä, and E. Rahtu, "Ove6d: Object viewpoint encoding for depth-based 6d object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6803–6813.

[15] I. Shugurov, F. Li, B. Busam, and S. Ilic, "Osop: A multi-stage one shot object pose estimation framework," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6835–6844.

[16] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 868–17 879.

[17] J. Sun, Z. Wang, S. Zhang, X. He, H. Zhao, G. Zhang, and X. Zhou, "Onepose: One-shot object pose estimation without cad models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6825–6834.

[18] X. He, J. Sun, Y. Wang, D. Huang, H. Bao, and X. Zhou, "Onepose++: Keypoint-free one-shot object pose estimation without cad models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 35 103–35 115, 2022.

[19] Y. He, Y. Wang, H. Fan, J. Sun, and Q. Chen, "Fs6d: Few-shot 6d pose estimation of novel objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6814–6824.

[20] F. Li, S. R. Vutukur, H. Yu, I. Shugurov, B. Busam, S. Yang, and S. Ilic, "Nerf-pose: A first-reconstruct-then-regress approach for weakly-supervised 6d object pose estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2123–2133.

[21] D. Cai, J. Heikkilä, and E. Rahtu, "Sc6d: Symmetry-agnostic and correspondence-free 6d object pose estimation," in *2022 International Conference on 3D Vision (3DV)*. IEEE, 2022, pp. 536–546.

[22] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The confluence of vision and control*. Springer, 2007, pp. 66–78.

[23] N. Adrian, V.-T. Do, and Q.-C. Pham, "Dfbvs: Deep feature-based visual servo," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2022, pp. 1783–1789.

[24] Y. Harish, H. Pandya, A. Gaud, S. Terupally, S. Shankar, and K. M. Krishna, "Dfvs: Deep flow guided scene agnostic image based visual servoing," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9000–9006.

[25] P. Katara, Y. Harish, H. Pandya, A. Gupta, A. Sanchawala, G. Kumar, B. Bhowmick, and M. Krishna, "Deepmpcvs: Deep model predictive control for visual servoing," in *Conference on Robot Learning*. PMLR, 2021, pp. 2006–2015.

[26] A. Saxena, H. Pandya, G. Kumar, A. Gaud, and K. M. Krishna, "Exploring convolutional networks for end-to-end visual servoing," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3817–3823.

[27] S. Felton, E. Fromont, and E. Marchand, "Siame-se (3): regression in se (3) for end-to-end visual servoing," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 14 454–14 460.

[28] C. Yu, Z. Cai, H. Pham, and Q.-C. Pham, "Siamese convolutional neural network for sub-millimeter-accurate camera pose estimation and visual servoing," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 935–941.

[29] A. Chen, H. Yu, S. Li, Y. Chen, Z. Zhou, W. Sun, R. Xiong, and Y. Wang, "Cnsv2: Probabilistic correspondence encoded neural image servo," *arXiv preprint arXiv:2503.00132*, 2025.

[30] M. Bain and C. Sammut, "A framework for behavioural cloning." in *Machine intelligence 15*, 1995, pp. 103–129.

[31] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[32] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[33] M. Vecerik, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, and M. Riedmiller, "Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards," *arXiv preprint arXiv:1707.08817*, 2017.

[34] Y. Zhu, Z. Wang, J. Merel, A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramár, R. Hadsell, N. de Freitas *et al.*, "Reinforcement and imitation learning for diverse visuomotor skills," *arXiv preprint arXiv:1802.09564*, 2018.

[35] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

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

[37] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke, "Google scanned objects: A high-quality dataset of 3d scanned household items," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2553–2560.