COMBO-Grasp: Learning Constraint-Based Manipulation for Bimanual Occluded Grasping

Jun Yamada Alexander L. Mitchell Jack Collins Ingmar Posner
University of Oxford

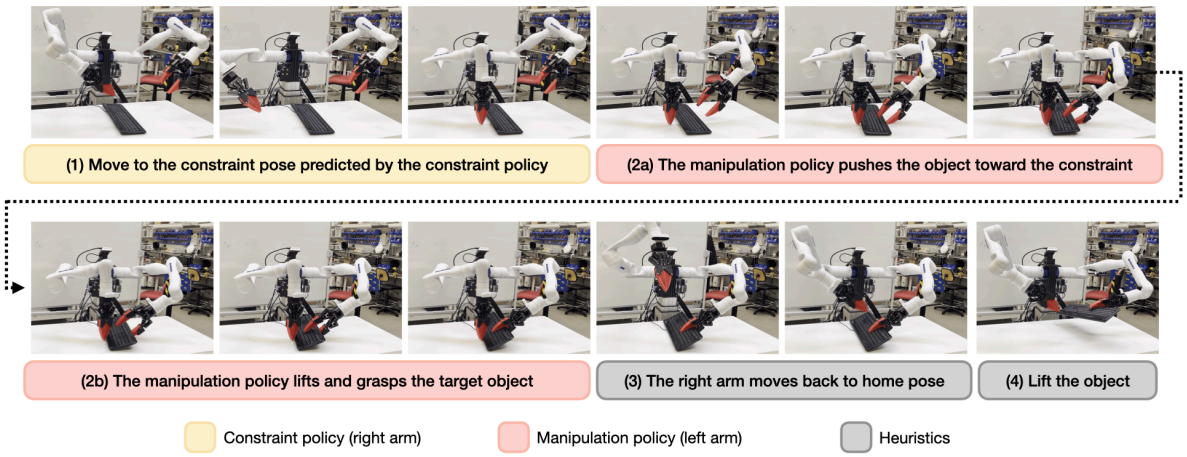

Fig. 1: We introduce *COMBO-Grasp*, a bimanual robotic system that uses two coordinated policies to address the challenges of grasping objects when the grasp pose is occluded. The system leverages a constraint policy that predicts the pose for the right arm to support the left arm during manipulation. Task execution unfolds in the following sequence: (1) the right arm moves to the predicted support pose using motion planning, (2) the left arm uses the constraint to grasp the target object, (3) the right arm returns to its home position, and (4) the left arm lifts the target object to complete the task.

***Abstract*—This paper addresses the challenge of *occluded* robot grasping, i.e. grasping in situations where the desired grasp poses are kinematically infeasible due to environmental constraints such as surface collisions. Traditional robot manipulation approaches struggle with the complexity of non-prehensile or bimanual strategies commonly used by humans in these circumstances. State-of-the-art reinforcement learning (RL) methods are unsuitable due to the inherent complexity of the task. In contrast, learning from demonstration requires collecting a significant number of expert demonstrations, which is often infeasible. Instead, inspired by human bimanual manipulation strategies, where two hands coordinate to stabilise and reorient objects, we focus on a bimanual robotic setup to tackle this challenge. In particular, we introduce Constraint-based Manipulation for Bimanual Occluded Grasping (*COMBO-Grasp*), a learning-based approach which leverages two coordinated policies: a constraint policy trained using self-supervised datasets to generate stabilising poses and a grasping policy trained using RL that reorients and grasps the target object. A key contribution lies in value function-guided policy coordination. Specifically, during RL training for the grasping policy, the constraint policy's output is refined through gradients from a jointly trained value function, improving bimanual coordination and task performance. Lastly, *COMBO-Grasp* employs teacher-student policy distillation to effectively deploy point cloud-based policies in real-world environments. Empirical evaluations demonstrate that *COMBO-Grasp* significantly improves task success rates compared to competitive baseline approaches, with successful generalisation to unseen objects in both simulated and real-world environments.**

I. Introduction

Grasping objects with kinematically infeasible grasp poses due to environmental collisions, known as occluded grasping [44], presents a significant challenge in robotics. Indeed, kinematic infeasibility arises from supporting surfaces, such as the table that the object is resting on. For example, grasping a keyboard that rests on a desk requires reorienting the keyboard with regard to the desk surface (nonprehensile manipulation) to reveal the grasp pose (see Figure 1). Humans exhibit exceptional dexterity in solving such occluded grasping problems through coordinated bimanual manipulation, seamlessly using both hands to reposition objects for grasping. However, learning to acquire such coordinated skills for a bimanual robotic system poses significant challenges, particularly when using reinforcement learning (RL) [31, 18].

Specifically, compared to single-handed applications, bimanual manipulation exhibits a significantly increased action space with coordination requirements adding to task complexity. These challenges are exacerbated when using domain randomisation [36] to enable sim-to-real transfer and render RL approaches infeasible due to sample inefficiency. For the occluded grasping task, these challenges are particularly pronounced as the policies must enable one arm to stabilise the object while the other reorients and grasps it. More

importantly, designing a reward function that facilitates the emergence of such coordinated behaviour is nontrivial. Compared to RL, learning from demonstration (LfD) necessitates a large number of expert demonstrations [45] encompassing a diverse range of objects to achieve generalisation to unseen objects.

In this work, we present **Co**nstraint-based **M**anipulation for **B**imanual **O**ccluded Grasping (*COMBO-Grasp*), a system specifically designed to address the challenges of occluded grasping using bimanual robot systems. Drawing inspiration from human bimanual strategies, where the nondominant hand stabilizes an object while the dominant hand performs complex manipulations [2, 1, 15], *COMBO-Grasp* consists of two coordinated policies: a *constraint policy* trained using self-supervised learning to generate stabilising poses, and a *grasping policy* trained using RL that reorients and grasps the target object. The constraint policy generates object-stabilising poses for one of the arms before the grasping policy controls the other arm to attempt grasping. The use of the constraint policy coupled with the grasping policy accelerates RL training as both policies work in coordination to solve occulded grasping problems in a data-efficient manner.

The constraint policy is trained on a synthetic dataset collected in a self-supervised manner within a simulation designed to leverage force closure as a signal. At the core of *COMBO-Grasp* is value function-guided policy coordination that refines the constraint pose generated by the constraint policy to improve bimanual coordination, thereby enhancing task performance. During RL training for the grasping policy, using gradients from the value function trained in tandem with the grasping policy, *COMBO-Grasp* optimises the constraint pose to increase the likelihood of a successful grasp once the grasping policy is executed. This value function-guided policy coordination ensures that the output of the constraint policy is refined to align with the goals of the grasping policy, leading to improved stability of objects during the bimanual grasping.

COMBO-Grasp achieves effective sim-to-real transfer by adopting a teacher-student policy distillation approach. The teacher policy, trained with privileged information for diverse objects in simulation, is distilled into a student policy that operates on point clouds for effective sim-to-real transfer. In contrast to training a single RL policy or LfD, *COMBO-Grasp* learns effective bimanual coordination with improved sampled efficiency and generalizes to unseen objects without relying on any expert demonstrations.

In summary, our contributions are four-fold:

- *COMBO-Grasp*, a novel approach to bimanual manipulation comprising two coordinated policies to solve occluded grasping problems.
- The use of force-closure as a signal to train a self-supervised constraint policy, which accelerates the subsequent RL grasping policy training.
- Value function-guided policy coordination that refines generated constraint poses using gradients from the value function to improve coordination during RL training for the grasping policy.
- Empirically demonstrating that *COMBO-Grasp* successfully grasps seen and unseen objects in both simulated and real-world environments.

II. Related Works

A. Learning to Grasp Objects

Grasping is a fundamental robot skill essential for various subsequent manipulation tasks [40, 12, 42]. Many prior works focus on learning grasp pose prediction models with open-loop planning controls [42, 26, 3]. These methods typically assume that there exist grasp poses that are not in collision with obstacles, such as a table, so that a robot can reach these poses using motion planning. Consequently, such open-loop approaches are inadequate for handling occluded grasping tasks where environmental constraints may block or interfere with the target grasp poses.

Prior art has also investigated closed-loop grasping policies using reinforcement learning (RL) [21, 38] and imitation learning (IL) [46, 33]. Recent advancements using deep RL have empowered robots to acquire complex manipulation skills driven by predefined reward functions. For example, Wang et al. [38] proposes learning a 6-DoF policy to grasp objects in tabletop environments. Moreover, QT-Opt [21] learns to grasp objects in a box by collecting diverse samples from multiple real-world robots. *COMBO-Grasp* is positioned amongst the works aiming to learn a closed-loop grasp policy, but focuses on more challenging occluded grasping scenarios that necessitate nonprehensile manipulation before grasping.

B. Occluded Grasping

Several prior work [34, 47] attempt to solve occluded grasping tasks through extrinsic dexterity while using only a single arm. Zhou and Held [47] train a policy using RL from state observations in simulation only for a rectangular object to learn extrinsic dexterity via interaction with a wall to reorient the object for grasping. Sun et al. [34] address the challenge of occluded grasping by utilising two arms; however, both arms are primarily employed for object reorientation, and the approach still relies on external constraints, such as a supporting wall. In contrast to these methods that rely on external constraints, *COMBO-Grasp* considers scenarios without external constraints, necessitating the use of one arm to stabilise the object while the other arm re-orients the object.

C. Bimanual Robotic Systems

Bimanual robotic manipulation has gained increasing attention due to its flexibility and capability to handle complex tasks such as dynamic handovers [20], twisting bottle lids [25], and assembly tasks [15, 32]. RL [18, 31] offers a reward-driven approach to skill acquisition but requires extensive exploration, particularly for high-DoF bimanual tasks. Alternatively, learning from demonstrations often demands a large number of expert demonstrations [45], which is often costly and impractical for complex bimanual systems, especially in non-prehensile manipulation scenarios.

Several works address these challenges by incorporating inductive biases into RL frameworks. For instance, predefined parameterised skills [10], intrinsic motivation for efficient exploration [9], and symmetry-aware actor-critic networks [24] have shown promise. Similarly, *COMBO-Grasp* introduces a constraint policy as an inductive bias, specifically tailored for occluded grasping tasks. In particular, inspired by studies in biopsychology [29, 2, 1], which suggest that the non-dominant arm tends to stabilise while the dominant arm performs more complex movements, *COMBO-Grasp* adopts a similar principle. One arm is used to stabilise and secure the object, while the other performs non-prehensile manipulation for occluded grasping.

Stabilising an object with one arm to assist the other in manipulation is a well-established strategy. For example, Grannen et al. [17] propose learning two policies: one for stabilising the object's position and the other for manipulation, applied to a peg-insertion task. Similarly, Shao et al. [32] adopt a dual-loop learning framework, where the inner loop optimises the grasping policy, and the outer loop refines the constraint policy based on the inner-loop performance. This approach, though effective, suffers from significant sample inefficiency.

In contrast, *COMBO-Grasp* eliminates the reliance on expert demonstrations or dual-loop training. Instead, it utilises self-supervised data collection in simulation to train a constraint policy that stabilises objects, thereby accelerating RL training of the grasping policy for occluded grasping of diverse objects. Crucially, *COMBO-Grasp* introduces value function-guided policy coordination to refine the generated constraint poses by leveraging gradients derived from maximising the value function of the grasping policy during the RL training. This refinement process enables the constraint policy to adapt the constraint pose such that it is more suitable for the grasping policy, thereby improving the coordination between the constraint policy and the grasping policy for bimanual occluded grasping tasks.

III. Task and System Setup

Task description. This work tackles bimanual occluded grasping problems. Occluded grasping refers to the inability to execute a desired grasp pose due to collisions between the robot and its environment.

In order to grasp a target object given a desired grasp pose that is occluded, one arm is needed to prevent the object from moving, while the dominant arm attempts to reorient and grasp the object. In this work, the left robot arm (dominant arm) always attempts to grasp a target object while the right arm (non-dominant arm) stabilises the object to assist the left arm. We leave dynamic role assignment of left and right arms to future work, similar to [17]. Moreover, the gripper of the left arm autonomously closes at the end of each episode to grasp the target object, and the left end-effector moves upward to lift the object.

We formulate two distinct Markov Decision Processes (MDP) for the occluded grasping task. The first MDP is fully observable and defined by a tuple $\mathcal{M}_1 = (\mathcal{S}_1, \mathcal{A}, \mathcal{R}, \gamma, \mathcal{P}_1)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{R}$ is the reward function, γ is a discount factor, and $\mathcal{P}$ is the environment dynamics. The second MDP is partially observable, a POMDP, and defined by a tuple $\mathcal{M}_2 = (\mathcal{S}_2, \mathcal{O}_2, \mathcal{A}, \mathcal{R}, \gamma, \mathcal{P}_2)$, where $\mathcal{O}$ is the observation space which includes point clouds. The state-based teacher constraint and grasping policies are trained in simulation under the MDP $\mathcal{M}_1$. On the other hand, the vision-based student constraint and grasping policies are evaluated in the simulated and real-world environments under the MDP $\mathcal{M}_2$. The RL policy is trained to maximise the cumulative reward $\sum_{t=0}^{T-1} \gamma^t \nabla(\mathbf{s}_t, \mathbf{a}_t)$ where $r \in \mathcal{R}$.

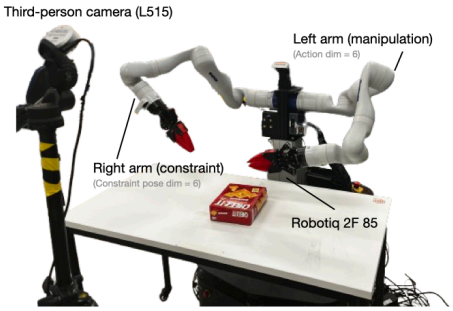

Fig. 2: **Real-world system setup.** The system comprises two Kinova Gen3 robotic arms mounted perpendicularly to the main body. Each arm is equipped with a Robotiq 2F-85 gripper. To enhance grasping performance, the grippers are fitted with soft fingertips [8] instead of the standard ones. Visual observations are captured using a third-person RealSense L515 camera positioned in front of the robot.

Action Space The teacher and student policies share the same action space. A grasping policy that controls the left arm outputs six-dimensional delta poses, including translations and rotations represented in axis-angle format. As described in Section III, the robot autonomously closes the left gripper at the end of the episode. Thus, the gasping policy does not include a discrete action for closing and opening the gripper. A constraint policy for the right arm produces a six-dimensional absolute pose for the right end-effector. As in prior work [32], the constraint policy focuses on the x-y plane, given that the constrained end-effector is typically positioned on the table. Thus, the first two dimensions correspond to the x and y coordinates of the end-effector, while the remaining four define its orientation as a quaternion. For the baseline RL policy that jointly controls both arms, the policy outputs 12-dimensional delta poses. The first six dimensions correspond to the delta actions for the left arm, while the remaining six dimensions are allocated to the right arm.

Real-World Setup. We design a system for bimanual occluded grasping as shown in Fig. 2. The system consists of two Kinova Gen3 arms with Robotq 2F-85 grippers. The robot arms are perpendicularly attached to a body. Instead of the original rigid fingertips attached to the Robotiq 2F-85 grippers, deformable fingertips [7] designed for a more secure grip are used. An L515 Realsense camera, with known extrinsic parameters, serves as the third-person camera and

provides point clouds to vision-based student policies. To control a robot, we use a hybrid task and joint space impedance controller [23].

Simulation Setup. Isaac Sim [27] is used to train teacher policies for the occluded grasping task. To train policies, 48 objects selected from the Google Scanned Objects dataset [14] are spawned into the environment (see Figure 10). The trained teacher policies are distilled to student policies that take as input point cloud observations obtained from a third-person camera. We use an operational space controller [22] to control robot arms. Further information regarding the simulation setup can be found in Appendix B.

IV. Approach

In this section we present *COMBO-Grasp*, a system designed to solve challenging bimanual occluded grasping tasks. Bimanual occluded grasping refers to scenarios where a desired grasp pose is initially occluded and in collision with external environments such as a table. COMBO-Grasp utilises two coordinated policies: a constraint policy trained on a dataset collected without human supervision within a simulation to stabilise the target object using one arm, and a grasping policy trained using RL to control the other arm and reorient the object for successful grasping.

This section first introduces a novel self-supervised data collection method (Section IV-A) that creates the data in simulation to train the teacher constraint policy (Section IV-B). The training procedure for the teacher grasping policy is then described in Section IV-C. Value function-guided policy coordination that refines the generated constraint pose during RL training for the teacher grasping policy is described in Section IV-C. Finally, the process of teacher-student distillation for real-world deployment is detailed in Section IV-E.

A. Self-Supervised Data Collection for Constraint Policy

Rather than relying on costly expert demonstrations for training, this work introduces a novel self-supervised data collection approach that uses force-closure as a signal in simulation to train the constraint policy across a diverse set of objects (see Figure 3 (1)). Firstly, target occluded grasp pose annotations for objects are generated using antipodal sampling [16]. The desired grasp poses are collected for 48 objects selected from the Google Scanned Objects dataset [14] (see Figure 10 in Appendix B). These desired grasp poses are also used during RL training for a grasping policy (see Section IV-C).

End-effector poses for the right arm are randomly sampled near the target object placed on a table, while the left arm remains fixed in its initial position. A force of $25N \times$ mass along the approach vector of a desired grasp pose is applied to the object. To assess force closure, the object's velocity is used as an approximation, as accurately evaluating force closure is often challenging, particularly in scenarios involving multiple contacts. Instead, force closure is considered successful if, after applying force, the object's velocity remains below a predefined threshold, given the specific grasp and constraint poses. The sampled end-effector pose, the corresponding desired grasp pose, and the object pose are then added to the dataset. By iterating this process in the simulation, $3K$ constraint poses per object are collected. With 48 objects, this results in a total of $144K$ samples. By leveraging the object's motion as a proxy measure for the success of a constraint pose, we can generate a rich set of training data to train the constraint policy.

B. Teacher Constraint Policy Training

One of the central contributions of this work lies in value function-guided policy coordination that integrates classifier guidance in diffusion models to refine the generated constraint pose during the training of the state-based teacher grasping policy. This is achieved by employing a diffusion model for the state-based teacher constraint policy, denoted as $\pi_{const}^{teacher}$, trained from the privileged information in the dataset (see Section IV-A). This approach leverages gradients from the value function to steer the teacher constraint policy's output, optimising the stabilising poses to align with the grasping policy's objectives to improve task performance and sample efficiency.

The teacher constraint policy uses a diffusion model formulated as a Denoising Diffusion Probabilistic Model (DDPM) [19]. DDPMs are a class of generative models where the output generation is modelled as a denoising process. Starting from x^K sampled from Gaussian noise, the DDPM performs K denoising iterations to generate a series of intermediate samples with decreasing levels of noise, $x^k, x^{k-1}, ...x^0$. This process is formulated as

$$\mathbf{x}^{k-1} = \alpha(\mathbf{x}^k - \gamma\epsilon_\theta(\mathbf{x}^k, k) + \mathcal{N}(0, \sigma^2 I)) \tag{1}$$

where ϵ_θ is a noise prediction network parameterised by θ. The choice of α, γ, σ is determined by a noise scheduler. To train the constraint policy, a forward diffusion process is applied to add noise to an unmodified sample, x^0, from the dataset by randomly sampling a denoising iteration k and random noise ϵ^k. The noise prediction model ϵ_θ is then trained to estimate the noise added to a sample during the forward diffusion process. Thus, the training loss is formulated as

$$\mathcal{L}_{constraint} = \mathit{MSE}(\epsilon^k, \epsilon_\theta(\mathbf{x}^0_{const} + \epsilon^k, k)) \tag{2}$$

where $\mathbf{x}_{const}$ is the constraint pose for the right arm. In this work, we employ an MLP-based denoising model as the backbone for the diffusion policy (see Appendix B1 for further details of the architecture).

The constraint policy takes as input an object pose, a desired grasp pose, and object IDs. For the Object IDs, we train an autoencoder [28] by reconstructing point clouds of objects using the Chamfer distance to learn compact representations similar to prior work [37]. Then, we use the learnt compact latent representation as the Object ID rather than using a one-hot vector, as this reduces the dimensionality of the observation space, especially when considering greater numbers of objects.

The state-based teacher constraint policy is used exclusively during the teacher grasping policy training phase, as described

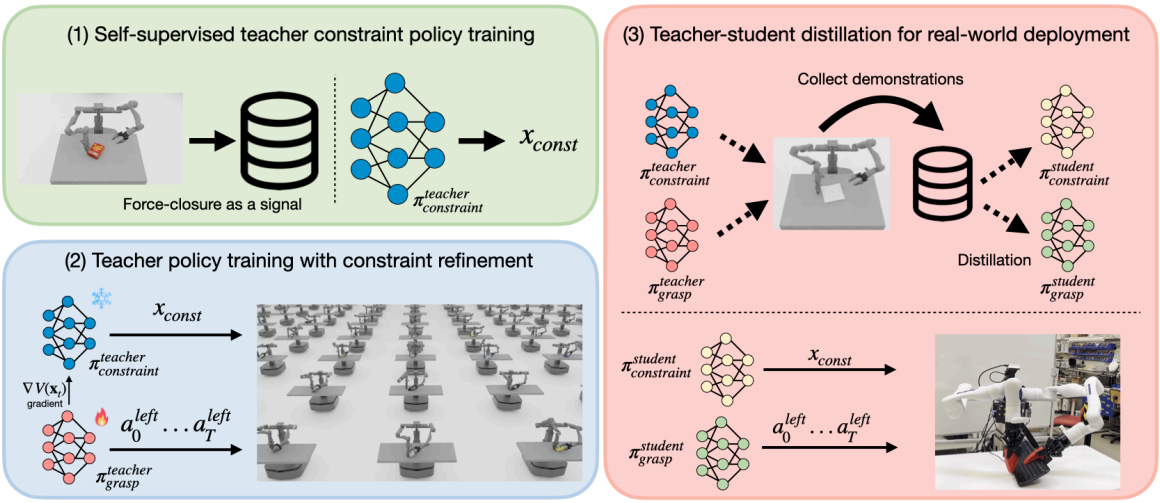

Fig. 3: **Method Overview.** (1) *COMBO-Grasp* first collects a synthetic dataset in a self-supervised manner in simulation to train the state-based teacher constraint policy. The teacher constraint policy outputs an end-effector pose for the right arm, given the privileged information available in the simulation. (2) The weights of the trained teacher constraint policy are frozen, and a teacher grasping policy, $\pi_{teacher}$, is trained using RL from privileged information in simulation. To maximise the performance, we propose value function-guided policy coordination that refines the output of the constraint policy using gradients propagated from the value function that is jointly trained with the grasping policy by maximising its value. (3) The teacher grasping policy and the teacher constraint policy are distilled into vision-based student policies that leverage point cloud observations, robot proprioceptive states, and, optionally, a desired grasp pose to address bimanual occluded grasping tasks in real-world environments.

in Section IV-C. At a later stage, the teacher constraint policy is distilled into a vision-based student constraint policy for sim-to-real transfer, ensuring robust performance across deployment scenarios.

C. Teacher Grasping Policy

After the constraint policy is trained, a teacher grasping policy $\pi^{teacher}_{grasp}$ is trained using Proximal Policy Optimisation (PPO) [31] for diverse objects from privileged information in simulation. To train a robust teacher grasping policy capable of performing in real-world environments, we employ domain randomisation, incorporating additive Gaussian noise into low-dimensional observations, as well as randomising the physics parameters of the target object and the controller parameters during policy training. For further information about the domain randomisation, see Appendix D1. The teacher grasping policy receives as input the robot's proprioceptive states, object pose, object velocity, desired grasp poses, object IDs (see Section IV-B), object's mass and friction parameters, and the PID gains for the OSC controller.

At the beginning of each training episode, the teacher constraint policy $\pi^{teacher}_{const}$ generates a constraint end-effector pose $\mathbf{x}_{const}$ for the right arm. Given the constraint end-effector pose, the joint positions of the right arm are computed using an inverse kinematics solver in CuRobo [35]. Then, the right arm moves to the computed desired constraint joint positions. Once the right arm is positioned, the grasping policy controls the left arm to attempt the occluded grasping task.

In this work, we design a reward function using six components: (1) distance reward between a left end-effector position and a desired grasp position, (2) distance reward between the left end-effector orientation and the desired grasp orientation, (3) action penalty to penalise the prediction of large actions by the grasping policy, (4) collision penalty including both self-collision and collision between robot arms and the table, (5) lift reward for incentivising the left arm to reorient the target object to expose the occluded target grasp pose, and (6) a sparse grasp success reward. The collision penalty term is computed using the sign distance field provided by CuRobo. The final reward r is

$$r = \alpha_1 r_{dist\_pos} + \alpha_2 r_{dist\_ori} - \alpha_3 r_{collision} - \alpha_4 r_{action} + \alpha_5 r_{lift} + \alpha_6 r_{success} \quad (3)$$

where α_i is a coefficient for each reward term. For more details on teacher policy training, domain randomisation, and each reward term with the coefficient value, see Appendix B.

D. Value Function-guided Policy Coordination

A key aspect of *COMBO-Grasp* is to induce effective bimanual coordination using the trained constraint policy, thereby improving task performance and enhancing the sample efficiency of RL policy training. Since the teacher constraint policy is initially trained on datasets collected using force closure as a signal, it does not inherently guarantee the generation of an optimal constraint for the grasping policy. To address this limitation, *COMBO-Grasp* draws inspiration from classifier guidance in diffusion models and propose value function-guided policy coordination that refines the generated constraint pose using gradients from a value function $V(\mathbf{x}_t)$, which is trained alongside the grasping policy using RL. The

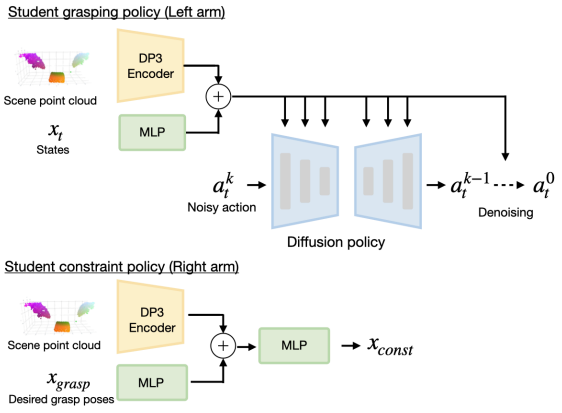

Fig. 4: **Student policy architecture.** We utilize DP3 [43] as the backbone for the grasping policy. The DP3 encoder processes the scene point cloud, and its output is concatenated with a state feature vector obtained by a multi-layer perceptron (MLP). The resulting concatenated vector serves as the conditioning input for the diffusion-based policy. Similarly, the constraint student policy employs the DP3 encoder and an MLP, but it takes a desired grasp pose as input. Unlike the grasping policy, the constraint student policy employs a Gaussian Mixture Model (GMM)-based policy.

value function of the grasping policy acts as a classifier in the classifier guidance framework, and the gradients for guidance are obtained by maximising the estimated value. This approach effectively refines the generated constraint poses to align more closely with the grasping policy's requirements, leading to improved overall performance and sample efficiency.

By incorporating gradients from the value function by maximisation, the denoising process for the constraint policy is formulated as

$$\mathbf{x}_{const}^{k-1} = \alpha(\mathbf{x}_{const}^{k} - \gamma\epsilon_\theta(\mathbf{x}_{const}^{k}, k) - w\nabla V(\mathbf{x}) + \mathcal{N}(0, \sigma^2 I)) \tag{4}$$

where w is a scaling parameter, $\mathbf{x}$ is low-dimensional observation used as input to the value function $V(\cdot)$, and the constraint pose $\mathbf{x}_{const}$ is a subset of the input state $\mathbf{x}$ for the value function (i.e., $\mathbf{x}_{const} \in \mathbf{x}$). For further details on value function-guided policy coordination, see Appendix B.

E. Policy Distillation for Sim-to-Real Transfer

To deploy a policy in real-world environments and grasp unseen objects, it is essential to leverage visual observations as an input modality to the policy. To achieve this, teacher-student policy distillation [41, 4] is used to transfer knowledge from the learnt teacher constraint and grasping policies to student policies. These student policies are designed to process point cloud observations and state observations, such as proprioceptive information and, optionally, a desired grasp pose. While the inclusion of the desired grasp pose slightly improves performance by biasing the arm's movements toward the target, omitting them still enables effective execution, albeit with a minor performance drop. In *COMBO-Grasp*, we employ a diffusion policy as the student grasping policy, similar to that used in the prior work [43]. The student policy consists of DP3 [43] and MLP encoders to process point cloud and state observations, as shown in Figure 4. The resulting vectors from these encoders are concatenated and used as the conditioning input for the diffusion policy. In contrast to the student grasping policy, the student constraint policy uses a Gaussian Mixture Model (GMM)-based policy for its simplicity. Since the student constraint policy does not require output steering like the teacher policy, the GMM approach is both effective and straightforward.

To distil the teacher to the student policy, we rollout the teacher policy in the simulation and collect expert demonstrations. The expert demonstrations include visual observations. We collect $10K$ expert demonstrations for use in the distillation process. During distillation, we apply a small perturbation to point cloud observations to simulate noise seen in real-world scenes. For further details on the student policy training, see Appendix C.

V. Experimental Results: Simulation

Our experimental evaluation aims to address the following questions: (1) How successful is *COMBO-Grasp* in learning a teacher policy compared to competitive baselines? (2) How well does *COMBO-Grasp* generalise to unseen objects? (3) How does the value function-guided policy coordination affect *COMBO-Grasp*'s overall performance? For further analysis of the experiment, see Appendix A.

A. Baselines

We compare *COMBO-Grasp* with the following baselines:

- **PPO**: A PPO [30] policy that controls both arms. The policy outputs 12-dimensional actions. Compared to *COMBO-Grasp* which employs two coordinated policies, this baseline requires more extensive exploration to solve the task.
- **PPO + Constraint Reward**: A PPO policy trained using a reward function that adds a distance-based reward between the right end-effector and the centre of the target object into the original reward function. This modification encourages the right arm to act as a constraint, assisting the left arm in grasping tasks. This enables the policy to avoid undesirable physical behaviours observed when using the original reward function, such as the left end-effector attempting to grasp the target object with high velocity without using the constraint.
- ***COMBO-Grasp* with a fixed constraint**: A PPO policy is trained to control the left arm, while the right arm remains fixed in a predefined pose in contrast to *COMBO-Grasp*. This showcases the importance of a constraint policy.
- ***COMBO-Grasp* without refinmenet**: *COMBO-Grasp* without value function-guided policy coordination. This demonstrates the necessity of refining the constraint pose generated by the constraint policy to further improve performance.

B. Evaluation Metric

For evaluation, we assess the success rate of grasping. In particular, a trial is considered successful if the robot's left arm securely grasps and lifts the target object at least $8\ cm$ at the end of the episode.

C. Sample Efficiency in Teacher Policy Training

Firstly, we evaluate the performance of teacher policy training in simulation. As shown in Figure 5, *COMBO-Grasp* solves the complex occluded grasping task with more sample efficiency and overall achieves a higher performance. On the other hand, *PPO* struggles to achieve similar performance due to the task and system complexity. More crucially, *PPO* trained with the original reward function often demonstrates undesirable physical behaviours, such that the left arm attempts to grasp the target object with high velocity without leveraging the right arm as a constraint. Such behaviour exploits the imperfect physics in the simulator and cannot be transferred to real-world environments. *PPO + Constraint Reward* alleviates this issue by adding a distance-based reward function. However, defining a reward function to induce a suitable constraint is challenging because such constraint poses are not known beforehand as this is a cooperative task dependent on the approach taken by both arms. Thus, engineering an appropriate reward function to elicit desired behaviour is not straightforward. These findings suggest that coordinated policies consisting of the constraint policy and the grasping policy effectively accelerate training and lead to higher success rates compared to a single policy trained using the state-of-the-art RL method.

COMBO-Grasp w/ fixed constraints demonstrate mediocre performance, as fixed constraints are sometimes not ideal for the grasping policy to solve occluded grasping tasks. Similarly, *COMBO-Grasp* w/o refinement shows worse performance compared to that of *COMBO-Grasp*. These results indicate that pre-training a constraint policy and refining the output during the teacher policy training are essential components in *COMBO-Grasp*.

D. Student Policy Performance in Simulation

We assess the performance of the distilled student policies on both seen and unseen objects in the simulation. Figure 6 shows the performance of the student policies in the simulated environments. *COMBO-Grasp* effectively addresses the challenging occluded grasping tasks for both seen and unseen objects. Without the desired grasp pose as input, *COMBO-Grasp* has reduced performance; however, it still demonstrates relatively performant results.

The success rates of *PPO* and *PPO + Constraint Reward* are similar for seen objects, but *PPO + Constraint Reward* performs significantly better on unseen objects. This improvement arises because *PPO + Constraint Reward* avoids exploiting imperfect physics and instead uses the right arm as a constraint induced by the constraint reward, which makes it more transferable to unseen objects. These findings suggest that learning a coordinated strategy is crucial for better transfer

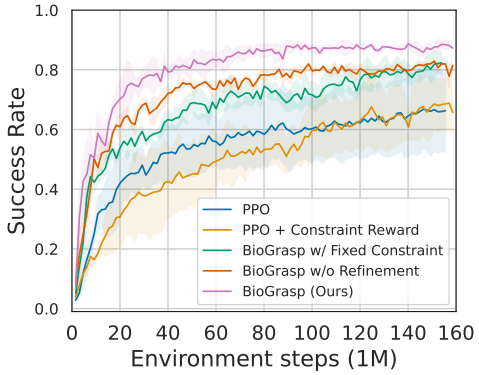

Fig. 5: **Teacher policy training.** We run 3 seeds for each method, and the shaded region represents the standard deviation. *COMBO-Grasp* significantly outperforms competitive baselines in both performance and sample efficiency.

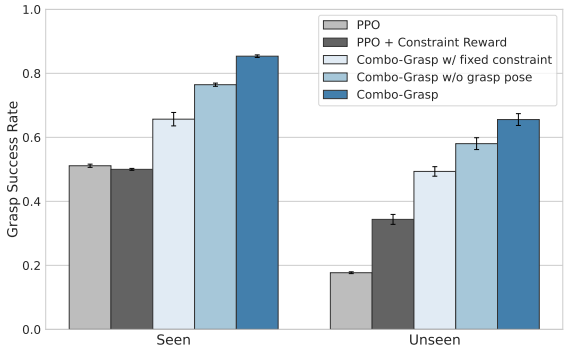

Fig. 6: Student Policy Performance averaged over 3 seeds in Simulated environments. We evaluate each approach for 50 times using both seen and unseen objects.

performance when dealing with unseen objects. Nonetheless, the performance of *PPO + Constraint Reward* remains inferior to *COMBO-Grasp* due to the difficulty of training a teacher policy that effectively learns a better constraint. The challenge of reward engineering limits the teacher's capability, which in turn reduces the performance of the student policy.

E. Ablation of the Value Function-guided Policy Coordination

The degree to which the value function-guided policy coordination improves the task success rate is investigated here. Concretely, the impact of the scaling parameter w on the constraint diffusion policy (see Eq. 4) during teacher policy training is investigated. As illustrated in Figure 7, the teacher policy's performance decreases when value function policy coordination is not applied (i.e., $\lambda = 0$). On the other hand, incorporating value function policy coordination consistently enhances the teacher policy's overall performance. This finding suggests that the constraint policy occasionally generates constraint poses that are suboptimal for the grasping policy. Consequently, value function policy coordination promotes on-the-fly adjustments and this is cooperation between the two arms achieves higher success rates.

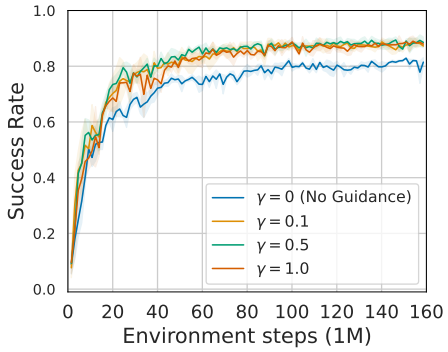

Fig. 7: **Guidance scaling ablation.** We compare the guidance scaling parameter to steer the output of the constraint policy. This result indicates that *COMBO-Grasp* without guidance shows worse performance and *COMBO-Grasp*is robust to a wide range of guidance scaling parameters to achieve better performance.

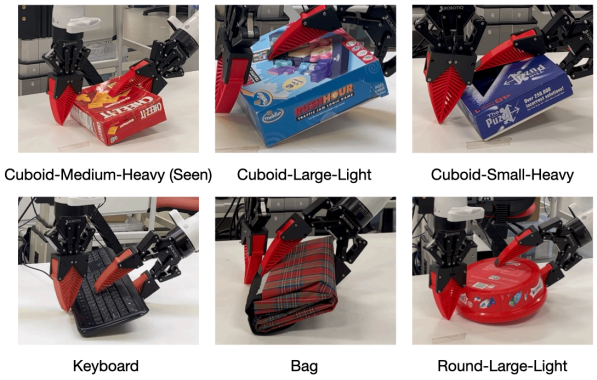

Fig. 8: We select several objects with differing sizes and weights that necessitate occluded grasping to evaluate *COMBO-Grasp* in the real-world environment.

VI. Experimental Results: Real-World

In this section, we evaluate a student policy trained using simulated data in real-world environments. We design experiments to address (1) What is the performance of *COMBO-Grasp* when operating over seen and unseen objects in the real-world? (2) Does the performance of *COMBO-Grasp* improve when conditioned on a desired grasp pose?

A. Experiment Setup

In this experiment, the student policies are evaluated using both seen and unseen objects with varying shapes, weights, and sizes, as illustrated in Figure 8. To facilitate grasping, we scan the objects to reconstruct their 3D meshes and employ antipodal sampling to generate desired grasp poses. This avoids reliance on off-the-shelf grasp pose prediction models [26], as addressing the inaccuracies of such models is beyond the scope of our evaluation. Nonetheless, *COMBO-Grasp* is compatible with any grasp pose prediction model that takes a segmented object point cloud as input.

	COMBO-Grasp	*COMBO-Grasp* w/o grasp pose
Cuboid-Medium-Heavy (Seen)	80% (8/10)	80% (8/10)
Cuboid-Large-Light	90% (9/10)	80% (8/10)
Cuboid-Small-Heavy	50% (5/10)	60%(6/10)
Keyboard	80% (8/10)	40%(4/10)
Bag	80% (8/10)	80% (8/10)
Round-Large-Light	30% (3/10)	10% (1/10)
Average	**68.3% (41/60)**	58.3% (35/60)

TABLE I: Performance of *COMBO-Grasp* in real-world environments for seen and unseen objects with varying shapes, sizes, and weights.

When the student policies are conditioned on the desired grasp poses, we estimate the object pose and thus can infer the desired grasp pose during manipulation in real-time, similar to prior work [37]. To achieve this, FoundationPose [39] is used to track object pose. Additioanlly, SAMTrack [5] is used to generate a segmentation mask to extract the target object point cloud used as input for the policies.

After generating a constraint pose using the student constraint policy, the desired joint positions for the right arm are obtained using the IK solver in CuRobo [35]. Then, MoveIt [11] is used to control the right arm to the desired positions. After the right arm is positioned in the constraint pose, the left arm begins the student grasping policy rollout.

B. Results

Table I shows the performance of the student policy in the real world. *COMBO-Grasp* effectively tackles occluded grasping tasks for both seen and unseen objects. Nonetheless, it encounters challenges with the round box, which is particularly difficult to stabilize in real-world conditions. While performance shows a slight decline, *COMBO-Grasp* still achieves a reasonable level of success, even without the target grasp pose as input. *COMBO-Grasp*, when operating without the target grasp pose, struggles to recover from failed non-prehensile manipulation attempts. When an initial push fails to move the object as intended, it is unable to retract and retry the push. For instance, pushing a keyboard towards a constraint is particularly challenging due to its thin profile, resulting in a low success rate of only 40%, as the left arm often fails to effectively position the keyboard.

In contrast, incorporating the desired grasp pose enables *COMBO-Grasp* to recover from such failures by reattempting the task until the left arm successfully reorients and completes the grasp. This demonstrates the importance of the desired grasp pose as a critical input for guiding the policy and improving performance. However, the version of *COMBO-Grasp* without the desired grasp pose offers increased flexibility, as it eliminates the need for real-time object pose estimation. This trade-off makes it more practical for scenarios where tracking the target grasp pose is infeasible.

VII. Limitations

COMBO-Grasp offers notable improvements in learning efficiency and generalisation compared to baselines and prior

occluded grasping methods. However, there are some limitations to consider. Firstly, *COMBO-Grasp* struggles with unseen objects of significantly different shapes, which could be addressed by training the teacher and student policy with a more diverse set of geometries. Additionally, *COMBO-Grasp* faces challenges with round objects in the real world, where stabilisation during occluded grasping is difficult. This issue could be mitigated through a closed-loop control approach, such as learning a residual policy for real-time constraint pose adjustments.

VIII. Conclusion

We present *COMBO-Grasp*, a bimanual robotic system for occluded grasping tasks. By introducing a constraint policy and value function-guided policy coordination that refine the generated constraint pose using gradients from a value function, we demonstrate that the coordinated policies can efficiently learn to solve challenging occluded grasping tasks. Furthermore, the trained teacher policies are distilled into student policies without dependence on privileged information for real-world deployment. Through empirical evaluation, we show that *COMBO-Grasp* achieves significantly better performance compared to a state-of-the-art baseline and instantiations of *COMBO-Grasp* in both simulated and real-world environments.

Acknowledgments

This work was supported by a UKRI/EPSRC Programme Grant [EP/V000748/1]. We would also like to thank the SCAN facility for carrying out this work (http://dx.doi.org/10.5281/zenodo.22558).

References

[1] Leia Bagesteiro and Robert Sainburg. Nondominant arm advantages in load compensation during rapid elbow joint movements. *Journal of neurophysiology*, 90:1503–13, 10 2003. doi: 10.1152/jn.00189.2003.

[2] Leia B Bagesteiro and Robert L Sainburg. Handedness: dominant arm advantages in control of limb dynamics. *Journal of neurophysiology*, 88(5):2408–2421, 2002.

[3] Kuldeep R Barad, Andrej Orsula, Antoine Richard, Jan Dentler, Miguel Olivares-Mendez, and Carol Martinez. Graspldm: Generative 6-dof grasp synthesis using latent diffusion models. *IEEE Access*, 2024.

[4] Julien Brosseit, Benedikt Hahner, Fabio Muratore, Michael Gienger, and Jan Peters. Distilled domain randomization. *arXiv preprint arXiv:2112.03149*, 2021.

[5] Yangming Cheng, Liulei Li, Yuanyou Xu, Xiaodi Li, Zongxin Yang, Wenguan Wang, and Yi Yang. Segment and track anything. *arXiv preprint arXiv:2305.06558*, 2023.

[6] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[7] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[8] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots, 2024. URL https://arxiv.org/abs/2402.10329.

[9] Rohan Chitnis, Shubham Tulsiani, Saurabh Gupta, and Abhinav Gupta. Intrinsic motivation for encouraging synergistic behavior. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJleNCNtDH.

[10] Rohan Chitnis, Shubham Tulsiani, Saurabh Gupta, and Abhinav Gupta. Efficient bimanual manipulation using learned task schemas. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1149–1155. IEEE, 2020.

[11] D.M. Coleman, Ioan Alexandru Sucan, Sachin Chitta, and Nikolaus Correll. Reducing the barrier to entry of complex robotic software: a moveit! case study. *ArXiv*, abs/1404.3785, 2014. URL https://api.semanticscholar.org/CorpusID:13939653.

[12] Jack Collins, Mark Robson, Jun Yamada, Mohan Sridharan, Karol Janik, and Ingmar Posner. Ramp: A benchmark for evaluating robotic assembly manipulation and planning. *IEEE Robotics and Automation Letters*, 2023.

[13] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023.

[14] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B. McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items, 2022. URL https://arxiv.org/abs/2204.11918.

[15] Michael Drolet, Simon Stepputtis, Siva Kailas, Ajinkya Jain, Jan Peters, Stefan Schaal, and Heni Ben Amor. A comparison of imitation learning algorithms for bimanual manipulation. *IEEE Robotics and Automation Letters*, 2024.

[16] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. A billion ways to grasp: An evaluation of grasp sampling schemes on a dense, physics-based grasp data set. In *The International Symposium of Robotics Research*, pages 890–905. Springer, 2019.

[17] Jennifer Grannen, Yilin Wu, Brandon Vu, and Dorsa Sadigh. Stabilize to act: Learning to coordinate for bimanual manipulation. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/

forum?id=86aMPJn6hX9F.

[18] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[20] Binghao Huang, Yuanpei Chen, Tianyu Wang, Yuzhe Qin, Yaodong Yang, Nikolay Atanasov, and Xiaolong Wang. Dynamic handover: Throw and catch with bimanual hands, 2023.

[21] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on robot learning*, pages 651–673. PMLR, 2018.

[22] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

[23] Min Jun Kim, Fabian Beck, Christian Ott, and Alin Albu-Schäffer. Model-free friction observers for flexible joint robots with torque measurements. *IEEE Transactions on Robotics*, 35(6):1508–1515, 2019. doi: 10.1109/TRO.2019.2926496.

[24] Yunfei Li, Chaoyi Pan, Huazhe Xu, Xiaolong Wang, and Yi Wu. Efficient bimanual handover and rearrangement via symmetry-aware actor-critic learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3867–3874, 2023. doi: 10.1109/ICRA48891.2023.10160739.

[25] Toru Lin, Zhao-Heng Yin, Haozhi Qi, Pieter Abbeel, and Jitendra Malik. Twisting lids off with two hands. *arXiv preprint arXiv:2403.02338*, 2024.

[26] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2901–2910, 2019.

[27] NVIDIA. Nvidia isaac sim. URL https://developer.nvidia.com/isaac-sim.

[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning internal representations by error propagation*, page 318–362. MIT Press, Cambridge, MA, USA, 1986. ISBN 026268053X.

[29] Robert L. Sainburg. Evidence for a dynamic-dominance hypothesis of handedness. *Experimental Brain Research*, 142:241–258, 2001. URL https://api.semanticscholar.org/CorpusID:206924666.

[30] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[31] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

[32] Lin Shao, Toki Migimatsu, and Jeannette Bohg. Learning to scaffold the development of robotic manipulation skills. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5671–5677, 2020. doi: 10.1109/ICRA40945.2020.9197134.

[33] Shuran Song, Andy Zeng, Johnny Lee, and Thomas A. Funkhouser. Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations. *IEEE Robotics and Automation Letters*, 5:4978–4985, 2019. URL https://api.semanticscholar.org/CorpusID:209140715.

[34] Zhaole Sun, Kai Yuan, Wenbin Hu, Chuanyu Yang, and Zhibin Li. Learning pregrasp manipulation of objects from ungraspable poses, 2020.

[35] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, et al. Curobo: Parallelized collision-free minimum-jerk robot motion generation. *arXiv preprint arXiv:2310.17274*, 2023.

[36] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[37] Weikang Wan, Haoran Geng, Yun Liu, Zikang Shan, Yaodong Yang, Li Yi, and He Wang. Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning, 2023. URL https://arxiv.org/abs/2304.00464.

[38] Lirui Wang, Yu Xiang, Wei Yang, Arsalan Mousavian, and Dieter Fox. Goal-auxiliary actor-critic for 6d robotic grasping with point clouds. In *Conference on Robot Learning*, pages 70–80. PMLR, 2022.

[39] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17868–17879, 2024.

[40] Jun Yamada, Jack Collins, and Ingmar Posner. Efficient skill acquisition for complex manipulation tasks in obstructed environments, 2023.

[41] Jun Yamada, Marc Rigter, Jack Collins, and Ingmar Posner. Twist: Teacher-student world model distillation for efficient sim-to-real transfer. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9190–9196. IEEE, 2024.

[42] Wentao Yuan, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. M2t2: Multi-task masked transformer for object-centric pick and place. *arXiv preprint arXiv:2311.00926*, 2023.

[43] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy. *arXiv*

preprint arXiv:2403.03954, 2024.

[44] Albert Zhan, Ruihan Zhao, Lerrel Pinto, Pieter Abbeel, and Michael Laskin. Learning visual robotic control efficiently with contrastive pre-training and data augmentation, 2022.

[45] Tony Z. Zhao, Jonathan Tompson, Danny Driess, Pete Florence, Kamyar Ghasemipour, Chelsea Finn, and Ayzaan Wahid. Aloha unleashed: A simple recipe for robot dexterity, 2024. URL https://arxiv.org/abs/2410.13126.

[46] Allan Zhou, Moo Jin Kim, Lirui Wang, Pete Florence, and Chelsea Finn. Nerf in the palm of your hand: Corrective augmentation for robotics via novel-view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17907–17917, 2023.

[47] Wenxuan Zhou and David Held. Learning to grasp the ungraspable with emergent extrinsic dexterity. In *Conference on Robot Learning*, pages 150–160. PMLR, 2023.

APPENDIX

A. Additional Analysis for Experiments

1) Student Policy Performance per Object: Figure 9 illustrates the success rate of *COMBO-Grasp* for each object used during training. While *COMBO-Grasp* demonstrate performant success rate across diverse objects, the occluded grasp performance for small objects or objects with complex geometries is reduced when compared to that of large objects with simple geometries. In order to overcome this limitation, it is suggested that both teacher and student policies be trained using more diverse objects, such as those available in the Objaverse datasets [13].

B. Teacher Policy Details

1) Teacher Constraint Policy: We employ a diffusion policy [6] as the basis for the teacher constraint policy. The diffusion policy is implemented using a Denoising Diffusion Probabilistic Model (DDPM), with a multi-layer perceptron (MLP)-based backbone. The denoising model is built on a three-level UNet architecture, comprising residual blocks with a hidden layer size of 512. The diffusion time step is encoded as an 80-dimensional feature vector. Additionally, the desired grasp pose, $\mathbf{x} \in \mathbb{R}^9$, and the object's ID, $\mathbf{x}_{obj\_id} \in \mathbb{R}^{16}$, are encoded into an 80-dimensional vector respectively to provide task-specific context. Similarly, the noisy input representing the constraint pose is encoded into another 80-dimensional vector. These encoded vectors are summed and passed through the residual blocks. The denoising model outputs the noise added to the original input during the forward diffusion process. In this work, we use 100 diffusion time steps for both training and inference. We train the diffusion policy using an Adam optimiser with a learning rate of 1×10^{-4}.

2) Teacher Grasping Policy: We train a teacher grasping policy using Proximal Policy Optimisation (PPO). An actor network consists of an MLP with 2 hidden layers of sizes $[256, 256]$. The actor network is parameterized as a Gaussian distribution with a fixed, state-independent standard deviation. The critic network consists of an MLP with 3 hidden layers of sizes $[256, 256, 256]$.

We define the privileged information used to train the policy as $[\mathbf{x}_{\text{robot}}, \mathbf{x}_{\text{goal}}, \mathbf{x}_{\text{obj}}] \in \mathbb{R}^{64}$. The robot proprioceptive states, $\mathbf{x}_{\text{robot}}$, include the left end-effector pose, $\mathbf{x}_{\text{left}} \in \mathbb{R}^9$, the right end-effector pose, $\mathbf{x}_{\text{right}} \in \mathbb{R}^8$, and the translational and rotational action scale parameters for the operational space controller, $\mathbf{x}_{\text{control}} \in \mathbb{R}^2$. The right end-effector states, $\mathbf{x}_{right}$, exclude the z-coordinate position, as the table height remains constant, and the constraint pose is fixed at a predetermined z-coordinate. The goal-related states, $\mathbf{x}_{\text{goal}}$, consist of the desired grasp pose, $\mathbf{x}_{\text{grasp}} \in \mathbb{R}^7$, the distance between the left end-effector and the desired grasp position, $\mathbf{x}_{\text{dist}} \in \mathbb{R}^3$, and the orientation distance between the left end-effector and the desired grasp orientation in the axis-angle representation, $\mathbf{x}_{\text{dist\_ori}} \in \mathbb{R}^3$. The object states, $\mathbf{x}_{\text{obj}}$, comprise the object pose, $\mathbf{x}_{\text{obj\_pose}} \in \mathbb{R}^7$, the object velocity, $\mathbf{x}_{\text{obj\_vel}} \in \mathbb{R}^6$, the friction parameters, $\mathbf{x}_{\text{friction}} \in \mathbb{R}^2$, the object's mass, $x_{\text{mass}} \in \mathbb{R}^1$, and the object's ID, $\mathbf{x}_{\text{obj\_id}} \in \mathbb{R}^{16}$.

We train the policy using an Adam optimiser with an adaptive learning rate scheduler[1] based on the KL divergence between the current policy and the previous policy, whose maximum learning rate is 1×10^{-2} and the minimum is 1×10^{-6}. We use a discount factor of 0.99, a GAE lambda value of 0.95, and an entropy coefficient of $6e-3$. After each policy rollout, the policy is updated using a batch size of 2048 for 8 epochs.

3) Reward function: The reward function used in our experiments comprises six terms and is defined as follows:

$$\begin{aligned} r = \alpha_1 r_{dist\_pos} + \alpha_2 r_{dist\_ori} - \alpha_3 r_{collision} \\ - \alpha_4 r_{action} + \alpha_5 r_{lift} + \alpha_6 r_{success} \end{aligned} \tag{5}$$

where the weighting coefficients are set to $\alpha_1 = 0.2$, $\alpha_2 = 0.2$, $\alpha_3 = 1.0$, $\alpha_4 = 0.025$, $\alpha_5 = 0.1$, and $\alpha_6 = 40$. Each term in the reward function serves a distinct purpose in guiding the robot's behaviour:

- **Position Distance Reward (r_{dist_pos}):** This term incentivizes the left end-effector to move towards the desired grasp position. It is computed as:

$$r_{\text{dist\_pos}} = 1 - \tanh(4 \cdot ||\mathbf{p}_{\text{left}} - \mathbf{p}_{\text{grasp}}||_2), \tag{6}$$

  where $\mathbf{p}_{\text{left}} \in \mathbb{R}^3$ and $\mathbf{p}_{\text{grasp}} \in \mathbb{R}^3$ represent the current and desired positions of the left end-effector, respectively.
- **Orientation Distance Reward (r_{dist_ori}):** This term encourages the left end-effector to align its orientation with the desired grasp orientation. The orientation difference is measured in the axis-angle space‘. The reward is computed as:

$$r_{\text{dist\_ori}} = 1 - \tanh(0.2 \cdot ||\boldsymbol{\theta}_{\text{left}} - \boldsymbol{\theta}_{\text{grasp}}||_2), \tag{7}$$

  where $\boldsymbol{\theta}_{\text{left}} \in \mathbb{R}^3$ and $\boldsymbol{\theta}_{\text{grasp}} \in \mathbb{R}^3$ represent the axis-angle representations of the current and desired orientations of the left end-effector, respectively.
- **Action Penalty (r_{action}):** This term discourages large control commands by penalizing the magnitude of the action vector:

$$r_{\text{action}} = ||\mathbf{a}||_2. \tag{8}$$

- **Collision Penalty ($r_{\text{collision}}$):** To prevent self-collisions and contact with the table, we compute the signed distance (SD) using CuRobo [35]. The collision penalty is given by:

$$r_{\text{collision}} = SD_{\text{self\_col}} + SD_{\text{table}}. \tag{9}$$

  The signed distance is computed for the robot arms, excluding the grippers, since the grippers must make contact with the table for occluded grasping problems. In CuRobo, a positive signed distance indicates a collision.
- **Lift Reward (r_{lift}):** This term encourages lifting the object to expose an initially occluded grasp pose. It is defined as an indicator function:

$$r_{\text{lift}} = \mathbb{1}(z_{\text{grasp}} > z_{\text{grasp,init}} + 2 \text{ cm}), \tag{10}$$

[1] https://skrl.readthedocs.io/en/latest/api/resources/schedulers/kl_adaptive.html

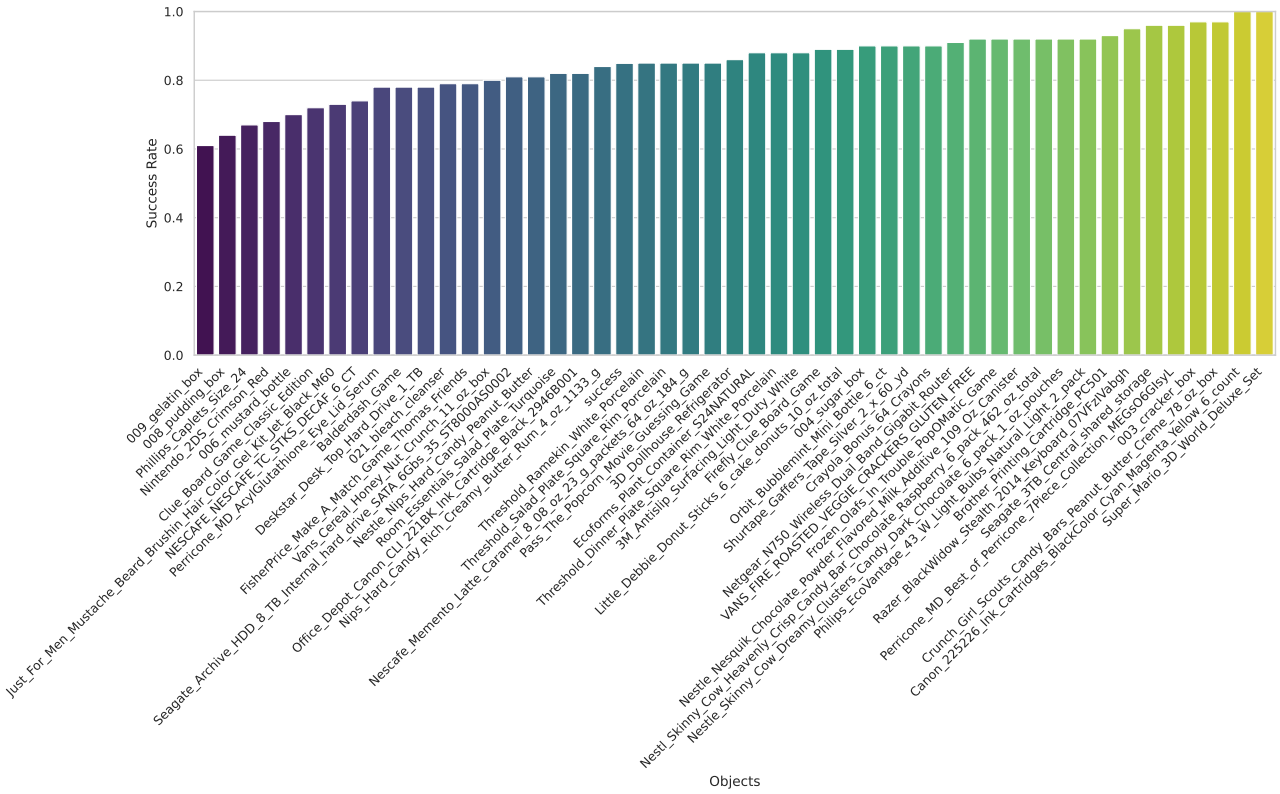

Fig. 9: **Grasp performance of *COMBO-Grasp*'s student policies for each object in simulation. The success rate is averaged over** 50 **trials.**

Fig. 10: **Training objects.** We choose 48 training objects from the Google Scanned Object Dataset [14].

where z_{grasp} and $z_{\text{grasp,init}}$ denote the current and initial heights of the desired grasp position, respectively.

- **Grasp Success Reward** ($r_{success}$)**:** At the end of an episode, a reward of 1 is assigned if the left arm successfully grasps and lifts the object; otherwise, the reward is 0:

$$r_{\text{success}} = \begin{cases} 1, & \text{if grasp and lift are successful,} \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Fig. 11: **Test objects.** We evaluate 10 held-out objects from the Google Scanned Object Dataset.

C. Student Policy Details

1) Studnet Constraint Policy: The student constraint policy integrates the DP3 encoder [43] and a state encoder to process point cloud and state observations, respectively.

The DP3 encoder comprises three fully connected layers with dimensions of $[128, 256, 384]$, followed by a max pooling operation and a final fully connected layer of size 64. Layer normalization and ReLU activations are applied after each of the initial three layers preceding the max pooling operation. The state encoder consists of two hidden layers with dimensions of $[128, 256]$. The state encoder outputs a feature vector of size 32 given the desired grasp pose $\mathbf{x}_{grasp}$.

The feature vectors produced by the DP3 and state encoders are concatenated and subsequently processed through a MLP to generate a constraint pose. For this work, the student policy utilizes a Gaussian Mixture Model (GMM)-based approach

due to its simplicity and effectiveness. Specifically, the GMM-based policy employs 5 modes, with a minimum standard deviation of 1×10^{-4}. We employ an AdamW optimiser with a learning rate of 5×10^{-5} and a weight decay of 5×10^{-5}.

2) Student Grasping Policy: We adopt the 3D Diffusion Policy (DP3) [43] as the foundation for the student grasping policy. The architecture of the DP3 encoder and the state encoder is consistent with that employed in the student constraint policy. However, the weights of these encoders are independently initialized from those of the constraint policy. Furthermore, the input dimension for the state encoder in the manipulation policy differs from that of the constraint policy. The state encoder for the manipulation policy processes $\mathbf{x}_{robot}$ and optionally $\mathbf{x}_{grasp}$ as input. During training, we employ 100 diffusion timesteps, whereas during inference a Denoising Diffusion Implicit Model (DDIMs) is used with 10 diffusion timesteps to accelerate action generation. We use an AdamW optimiser with a learning rate of 5×10^{-5} and a weight decay of 5×10^{-5}.

D. Simulation Setup

1) Training: In order to train a teacher policy from a diverse set of objects, we select 48 objects from the Google Scanned Object dataset, as illustrated in Figure 10. To train teacher policies efficiently, we spawn 1024 robots and objects in the simulated environment.

In order to train a policy robust to noises and effectively transfer it to real-world environments, we apply domain randomisation during teacher policy training. Table II describes the details of the randomisations used in our experiments. We also apply domain randomisation during the self-supervised data collection for the constraint policy.

TABLE II: Domain Randomisation Hyperparameters

Parameter	Description
Initial robot joint positions	Add noise sampled from $\mathcal{N}(0, 0.05)$
Robot base position	Add random noise sampled from $\mathcal{U}(-0.015, 0.015)$ to the z-coordinate of the robot base
PID position action scale	Sampled from $\mathcal{U}(0.03, 0.04)$
PID rotation action scale	Sampled from $\mathcal{U}(0.1, 0.2)$
Action	Add random noise sampled from $\mathcal{N}(0, 0.01)$
Object mass	Add mass sampled from $\mathcal{U}(-0.1, 0.1)$
Static and dynamic friction	Sampled from $\mathcal{U}(0.8, 1.2)$
Grasp position	Add random noise sampled from $\mathcal{N}(0, 0.005)$
Grasp translational distance	Add random noise sampled from $\mathcal{N}(0, 0.005)$
Grasp rotational distance	Add random noise sampled from $\mathcal{N}(0, 0.005)$
End-effector position	Add random noise sampled from $\mathcal{N}(0, 0.01)$
Object position	Add random noise sampled from $\mathcal{N}(0, 0.01)$
Object orientation	Add random noise sampled from $U(-0.2\pi \text{ rad}, 0.2\pi \text{ rad})$ to the yaw axis

2) Evaluation: To evaluate policies for both seen and novel objects, we also select 10 held-out objects from the Google Scanned Object dataset (see Figure 11).

E. Real-World Experiment Setup

1) Input Observation for Student Policies: The distilled student policies take point clouds as input in real-world environments. We render depth images with the size of 640×480 from a Realsense L515 camera to reconstruct point cloud observations. Similar to [43], we crop the point cloud within a pre-defined bounding box such that it includes the robot arms and the target object. Then, we remove statistical outliers from the point clouds reconstructed from depth images and apply farthest point sampling to sub-sample 1024 points.

2) Desired Occluded Grasp Pose Generation: In order to scan an object to reconstruct a mesh, we use Polycam, an application that captures pictures of objects and reconstructs an object mesh using Neural Radiance Fields (NeRF). Using the reconstructed mesh, we generate desired occluded grasp poses using antipodal sampling.

F. Baseline Method Details

1) PPO: We train a policy using Proximal Policy Optimization (PPO) [30], where the policy outputs 12-dimensional delta end-effector poses corresponding to both the left and right arms. We use the same hyperparameters employed for training *COMBO-Grasp*, except for the entropy coefficient, which is set to 0.003. This modification was made because using the original entropy coefficient caused a continuous increase in the policy's standard deviation, resulting in the policy's inability to exploit a stable and effective strategy during training.

2) PPO + Constraint Reward: Similar to the *PPO* baseline, but we introduce an additional reward term that encourages the right arm to be used as a constraint. In particular, we add a reward $r_{right\_dist} = ||T^{obj} - T^{RightEE}||_2$.

3) COMBO-Grasp *w/ Fixed Constraint:* Instead of employing a trained constraint policy, we place the right arm as a constraint at a fixed pose. To accommodate objects of varying sizes and orientations, the constraint is positioned at the right hand side of the workspace rather than at the centre. This policy is trained using the same hyperparameters as those employed by *COMBO-Grasp*.