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

Jun Yamada, Alexander L Mitchell, Jack Collins, Ingmar Posner
Applied AI Lab
Oxford Robotics Institute
University of Oxford

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. Existing RL methods struggle with task complexity, and collecting expert demonstrations is often impractical. 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*), an 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, where gradients from a jointly trained value function refine the constraint policy during RL training to improve bimanual coordination and task performance. Lastly, *COMBO-Grasp* employs teacher-student policy distillation to effectively deploy vision-based policies in real-world environments. Experiments show that *COMBO-Grasp* significantly outperforms baselines and generalises to unseen objects in both simulation and real environments. Videos can be found at: https://combo-grasp.github.io.

Keywords: Bimanual Occluded Grasping, Reinforcement Learning

1 Introduction

Grasping objects with kinematically infeasible grasp poses due to environmental collisions, known as occluded grasping [1], presents a significant challenge in robotics. Such 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) [2, 3].

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 [4] to enable sim-to-real transfer and make 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 [5] encompassing a diverse range of objects to achieve generalisation to unseen objects.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

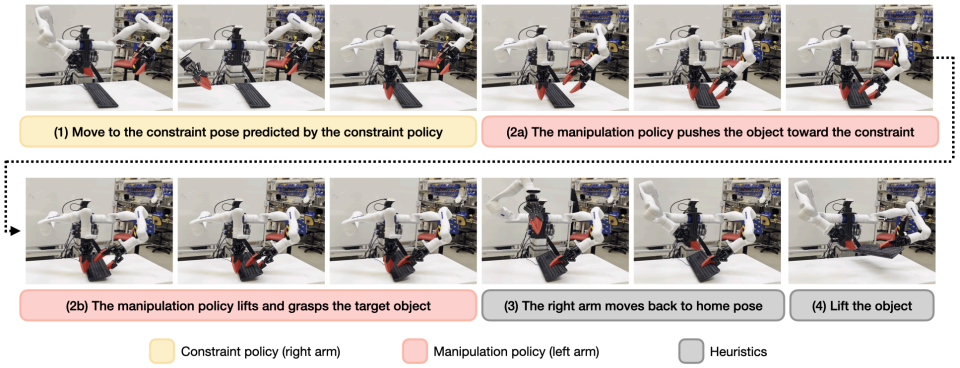

Figure 1: *COMBO-Grasp* uses two coordinated policies to tackle occluded grasping tasks. A constraint policy predicts a support pose for the right arm to assist the left arm controlled by a grasping policy. The task execution sequence is: (1) the right arm moves to the support pose, (2) the left arm grasps the object, (3) the right arm returns home, and (4) the left arm lifts the object.

We present **Co**nstraint-based **M**anipulation for **B**imanual **O**ccluded Grasping (*COMBO-Grasp*), a system designed to address occluded grasping using bimanual robot systems. Inspired by human bimanual strategies, where one hand stabilises an object while the other performs the manipulation [6, 7, 8], *COMBO-Grasp* uses two coordinated policies: a *constraint policy*, trained from dataset collected in a self-supervised manner, that generates stabilising poses, and a *grasping policy* trained using RL that reorients and grasps the target. By stabilising with one arm before grasping with the other, this coordination improves data efficiency and accelerates training for occluded grasping tasks. *COMBO-Grasp* also introduces value-guided policy coordination to refine the constraint pose, improving bimanual coordination. In particular, during RL training, gradients from the value function, trained alongside the grasping policy, optimise the constraint pose to increase grasp success. This alignment enhances object stability during bimanual grasping.

COMBO-Grasp achieves effective sim-to-real transfer via teacher-student policy distillation. A teacher trained with privileged information in simulation is distilled into a student policy that operates on point clouds. Unlike single-policy RL or LfD, *COMBO-Grasp* enables efficient bimanual coordination and generalises to unseen objects without 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 object stabilisation as a signal for self-supervised data collection, enabling training of a constraint policy that accelerates subsequent RL grasping policy learning.
- 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.

2 Related Works

Learning to Grasp Objects. Grasping is a fundamental robotic skill crucial for downstream manipulation tasks [9, 10, 11]. Many prior works focus on learning grasp pose predictors with open-loop planning [11, 12, 13], typically assuming that collision-free poses are reachable via motion planning. However, these methods are often inadequate for occluded grasping scenarios, where environmental constraints may obstruct the target grasp poses. Closed-loop policies using reinforcement learning (RL)[14, 15, 16] and imitation learning (IL)[17, 18] provide an alternative. *COMBO-Grasp* builds on this direction, addressing more challenging occluded grasping tasks that require non-prehensile manipulation before grasping. Some prior works [19, 20, 21] address occluded grasping via extrin-

sic dexterity using a single arm. Sun et al. [19] employ dual arms for object reorientation, though it still relies on external constraints such as a wall. In contrast, *COMBO-Grasp* operates without such constraints, using one arm to stabilise the object while the other performs reorientation.

Bimanual Robotic Systems. Bimanual robotic manipulation [22, 23, 8, 24] has gained increasing attention due to its flexibility and capability to handle complex tasks. RL approaches [3, 2] often require extensive exploration, particularly for high-DoF bimanual tasks. Alternatively, IL often demands a large number of expert demonstrations [5], which is often costly and impractical for complex bimanual systems, especially in non-prehensile manipulation scenarios. Several works [25, 26, 27] address these challenges by incorporating inductive biases into RL. Similarly, *COMBO-Grasp* introduces a constraint policy as an inductive bias, specifically tailored for occluded grasping tasks. Inspired by studies in biopsychology [28, 6, 7], *COMBO-Grasp* uses one arm to stabilise 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 [29, 24]. However, these prior works require expert demonstrations [29] or nested optimisation loops [24], limiting scalability due to high supervision cost or sample inefficiency. In contrast, *COMBO-Grasp* eliminates the need for expert demonstrations or nested optimisation by using self-supervised simulation data to train a constraint policy, which stabilises objects and accelerates RL training for occluded grasping. Crucially, *COMBO-Grasp* uses value function-guided policy coordination to refine constraint poses by leveraging gradients from the grasping policy's value function during RL training. This allows the constraint policy to adapt poses that better align with the grasping policy, enhancing coordination for bimanual occluded grasping tasks.

3 Task and System Setup

Task description. 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 [29]. It is worth noting that 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.

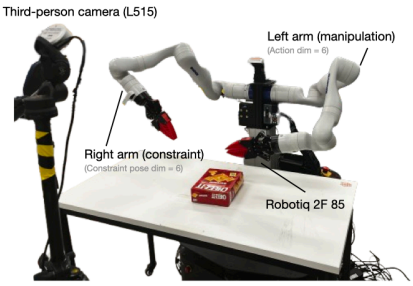

Figure 2: **Real-world system setup.** The system uses two Kinova Gen3 arms mounted perpendicularly, each with a Robotiq 2F-85 gripper and soft fingertips [30] for improved grip. A third-person RealSense L515 camera provides visual observations.

Action Space The teacher and student policies share the same action space. The grasping policy controls the left arm and outputs a six-dimensional delta pose, including translation and rotation in axis-angle representation. The constraint policy controls the right arm and outputs a six-dimensional absolute pose. Following prior work [24], our experiments assume the end-effector remains at a fixed z-coordinate, as it is typically placed on the table, with variations only in its x-y position and orientation. Thus, the first two dimensions correspond to the x and y positions, and the remaining four specify orientation as a quaternion.

Real-World Setup. We design a system for bimanual occluded grasping (Fig.2) comprising two Kinova Gen3 arms with Robotiq 2F-85 grippers, mounted perpendicularly on a central body. The grippers use deformable fingertips [31] for improved grip, replacing the original rigid ones. A calibrated Realsense L515 camera provides third-person point clouds for the vision-based student policies. To control the arms, we use a hybrid task and joint space impedance controller [32, 33].

Simulation Setup. Isaac Sim [34] is used to train teacher policies for the occluded grasping task. To train policies, 48 objects selected from the Google Scanned Objects dataset [35] are spawned into

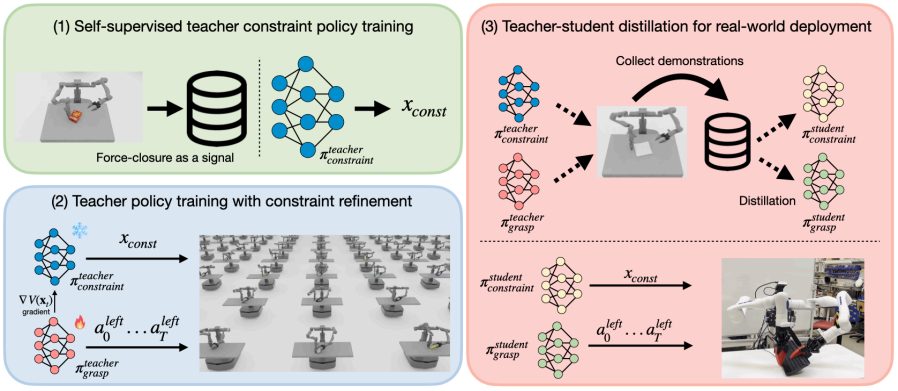

Figure 3: **Method Overview.** (1) *COMBO-Grasp* first collects a synthetic dataset in a self-supervised manner in simulation to train a state-based teacher constraint policy that outputs a right arm end-effector pose. (2) This constraint policy is frozen, and a state-based teacher grasping policy $\pi_{teacher}$ is trained with RL. To improve performance, we propose value-guided policy coordination, refining the constraint output via gradients from a jointly trained value function. (3) Both teacher policies are then distilled into vision-based student policies using point clouds, proprioception, and optionally a desired grasp pose to tackle real-world bimanual occluded grasping.

the environment (see Figure 9). We use an operational space controller [36] to control robot arms. Further information regarding the simulation setup can be found in Appendix B.

4 Approach

In this section we present *COMBO-Grasp*, a system designed to solve challenging bimanual occluded grasping tasks. 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.

We first present a self-supervised data collection method in simulation (Section 4.1) to train the teacher constraint policy (Section 4.2). Section 4.3 details the training of the teacher grasping policy, including value function-guided coordination for refining constraint poses. Finally, teacher-student distillation for real-world deployment is described in Section 4.5.

4.1 Self-Supervised Data Collection for Constraint Policy

Instead of relying on costly expert demonstrations, this work introduces a self-supervised data collection method using force-closure signals in simulation to train the constraint policy across diverse objects (see Figure 3 (1)). Target occluded grasp poses are generated via antipodal sampling [37] for 48 objects from the Google Scanned Objects dataset [35] (see Figure 9 in Appendix B). These poses are also used during RL training for the grasping policy (Section 4.3).

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 whether a stabilising pose is achieved, the object's velocity is used as an approximation. A stabilising pose is considered successful if, after applying force, the object's velocity remains below a predefined threshold for the given 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 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.

4.2 Teacher Constraint Policy Training

One of the central contributions of this work lies in value function-guided policy coordination, which builds upon classifier guidance used 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 4.1). 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) [38]. 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, $x^k, x^{k-1}, ...x^0$. 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

$$\mathcal{L}_{constraint} = \mathit{MSE}(\epsilon^k, \epsilon_\theta(\mathbf{x}^0_{const} + \epsilon^k, k)) \tag{1}$$

where $\mathbf{x}_{const}$ is the constraint pose for the right arm. An MLP-based denoising model is used as the backbone for the diffusion policy (see Appendix B.1 for further details of the architecture).

The constraint policy takes as input the object pose, desired grasp pose, and object IDs. To represent Object IDs, an autoencoder [39] is trained to reconstruct object point clouds using the Chamfer distance. The resulting compact latent code replaces one-hot vectors, reducing observation dimensionality for large object sets. The state-based teacher constraint policy is used only during teacher grasping policy training (Section 4.3) and is distilled into a vision-based student policy for sim-to-real transfer.

4.3 Teacher Grasping Policy

After the constraint policy is trained, a teacher grasping policy $\pi_{grasp}^{teacher}$ is trained using Proximal Policy Optimisation (PPO) [2] on 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 D.1. The teacher grasping policy receives as input the robot's proprioceptive states, object pose, object velocity, desired grasp poses, object IDs (see Section 4.2), object's mass and friction parameters, and the PID gains for the operational space control(OSC).

At the beginning of each training episode, the teacher constraint policy $\pi_{const}^{teacher}$ 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 the CuRobo IK solver [40]. 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.

We design a reward function with six components: (1) position and (2) orientation distance to the target grasp pose for a left end-effector, (3) action penalty to penalise the large actions, (4) collision penalty (including self- and table collisions), (5) lift reward to expose occluded grasps, and (6) sparse grasp success reward. The collision penalty term is computed using the signed distance field provided by CuRobo. The final reward r is

$$r = \alpha_1 r_{dist\_pos} + \alpha_2 r_{dist\_ori} - \alpha_3 r_{collision} - \alpha_4 r_{action} + \alpha_5 r_{lift} + \alpha_6 r_{success} \tag{2}$$

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.

4.4 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 the RL policy training. Since the teacher constraint policy is initially trained on datasets collected using a signal indicating whether a moving object is stabilised, 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 we 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 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

$$\mathbf{x}_{const}^{k-1} = \alpha(\mathbf{x}_{const}^{k} - \gamma\epsilon_\theta(\mathbf{x}_{const}^{k}, k) - w\nabla V(\mathbf{x}) + \mathcal{N}(0, \sigma^2 I)) \tag{3}$$

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.

4.5 Policy Distillation for Sim-to-Real Transfer

To deploy policies in real-world environments, leveraging visual observations as input is essential. Teacher-student policy distillation [41, 42] is used to transfer knowledge from trained teacher constraints and grasping policies to student policies. These student policies process point cloud observations along with state information, such as proprioceptive data and, optionally, a desired grasp pose. In *COMBO-Grasp*, we adopt a diffusion policy as the student grasping policy, similar to prior work [43]. Specifically, DP3 [43] and MLP encoders process point cloud and state observations, respectively, as illustrated in Figure 11 (Appendix C). The encoder outputs are concatenated to condition the diffusion policy. For simplicity, the student constraint policy employs a Gaussian Mixture Model (GMM). Unlike the teacher constraint policy, it does not require output steering, making the GMM approach effective and straightforward.

To distil the teacher to the student policy, we rollout the teacher in simulation and collect $10K$ expert demonstrations with visual observations. During distillation, we apply small perturbations to point cloud observations to simulate real-world noise. For further details, see Appendix C.

5 Experimental Results: Simulation

Our experiments 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 experiments, see Appendix A.

5.1 Evaluation Metric and Baselines

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.

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

- **PPO**: A PPO [44] 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 with a modified reward function that adds a distance-based term between the right end-effector and the target object's center. This encourages the right arm to act as a constraint, assisting the left arm in grasping. The policy thus avoids undesirable behaviors seen with the original reward, such as high-velocity grasps by the left end-effector without support.
- ***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.

5.2 Sample Efficiency in Teacher Policy Training

***COMBO-Grasp* achieves higher performance and sample efficiency.** We first evaluate teacher policy training in simulation. As shown in Fig. 4, *COMBO-Grasp* solves the occluded grasping task more efficiently and achieves better overall performance. In contrast, *PPO* struggles due to task and system complexity. More critically, it often exhibits unrealistic behaviours—e.g., the left arm grasping aggressively without right-arm support—by exploiting simulator inaccuracies, which fail to transfer to the real world.

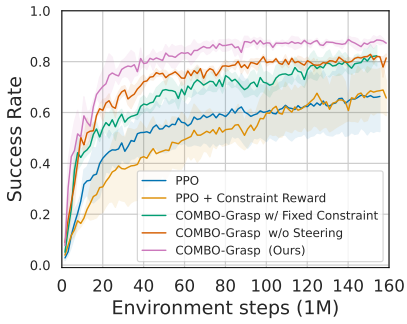

Figure 4: **Teacher policy training.** We run 3 seeds per method, with shaded regions showing standard deviation. *COMBO-Grasp* significantly outperforms baselines in both performance and sample efficiency.

Reward shaping alone is insufficient for coordination. *PPO + Constraint Reward* partially addresses these issues using a distance-based reward, but effective constraint poses are not known a priori and depend on coordinated behaviour between arms. This highlights the difficulty of inducing such coordination through reward engineering alone.

Constraint learning and coordination drive performance. *COMBO-Grasp* with fixed constraints performs poorly, as static poses may not generalise across tasks. Similarly, removing refinement degrades performance. These findings emphasise the importance of both pre-training and refining the constraint policy. In general, our coordinated approach, learning separate constraints and grasping policies, yields faster training and higher success rates than the RL baselines of a single policy.

5.3 Student Policy Performance in Simulation

***COMBO-Grasp* generalises well to both seen and unseen objects.** We evaluate the distilled student policies in simulation (Fig.5). *COMBO-Grasp* handles occluded grasping effectively across object sets. Without the target grasp pose as input, performance drops but remains competitive.

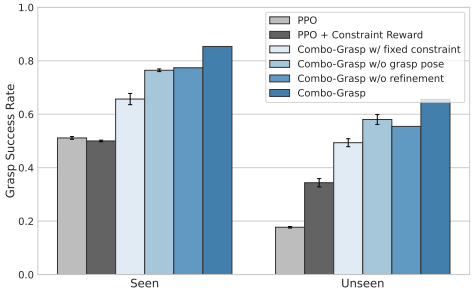

Figure 5: Student Policy Performance averaged over 3 seeds in Simulated environments. We evaluate each approach for 50 times using both seen and unseen objects.

Coordinated strategies improve generalisation to unseen objects. While *PPO* and *PPO + Constraint Reward* perform similarly on seen objects, the latter significantly outperforms on unseen ones by leveraging the right arm as a constraint instead of exploiting simulator flaws. However, it still lags behind *COMBO-Grasp* due to the difficulty of reward shaping for effective constraint learning, which limits the teacher policy and thus the student's performance.

6 Experimental Results: Real-World

We evaluate a student policy trained on simulated data in real-world settings to address (1) How does *COMBO-Grasp* perform on seen and unseen real-world objects? (2) Does conditioning on a desired grasp pose improve its performance?

6.1 Experiment Setup

Student policies are evaluated on both seen and unseen objects with diverse shapes, sizes, and weights (Figure 6). To facilitate grasping, we scan objects to reconstruct 3D meshes and generate grasp poses via antipodal sampling, avoiding the need for grasp pose prediction models [12], which are outside our evaluation scope. However, *COMBO-Grasp* is compatible with any grasp pose prediction models. When student policies are conditioned on desired grasp poses, object pose is estimated in real-time using FoundationPose[45], enabling grasp pose inference during manipulation [46].

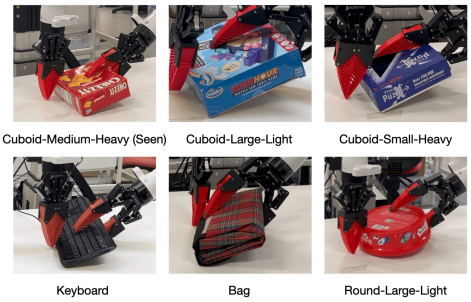

Figure 6: Selected objects of varying sizes and weights requiring occluded grasping are used to evaluate *COMBO-Grasp* in the real world.

Constraint poses from the student constraint policy are converted to joint positions using CuRobo's IK solver [40], and MoveIt [47] controls the right arm accordingly. Once positioned, the left arm executes the student grasping policy.

6.2 Results

***COMBO-Grasp* is effective in real-world occluded grasping, with trade-offs depending on input.** As shown in Table 1, *COMBO-Grasp* handles occluded grasping well for both seen and unseen objects. It struggles with the round box due to stability challenges, and performance slightly declines without the target grasp pose. In this setting, the policy cannot recover from failed nonprehensile manipulation; for instance, pushing a keyboard often fails due to its thin shape, leading to a 40% success rate.

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

The target grasp pose improves robustness, but removing it increases practicality. Providing the desired grasp pose enables retries and improves success by guiding the left arm more effectively. However, omitting it increases deployment flexibility, removing the need for real-time pose estimation, which is useful in environments where tracking is infeasible. The complete baseline results, including *PPO* and various ablations of *COMBO-Grasp*, are presented in Table 2 in Appendix A.3.

7 Conclusion

We present *COMBO-Grasp*, a bimanual robotic system for occluded grasping tasks. By introducing a constraint policy and value function-guided policy coordination, which refines the constraint pose using value gradients, we show that coordinated policies efficiently solve challenging occluded grasping tasks. Furthermore, the trained teacher policies are then distilled into vision-based student policies 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.

8 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.

Finally, while COMBO-Grasp is tailored for bimanual occluded grasping, we view this as a foundational step toward solving a broader range of bimanual tasks, such as threading a needle, painting, or cutting—where one arm must constrain the object while the other performs precise ma- nipulation. We see COMBO-Grasp with value-guided implicit coordination as a step towards efficiently solving this class of problem.

Acknowledgments

This work was supported by a UKRI/EPSRC Programme Grant [EP/V000748/1]. We would like to acknowledge the use of the University of Oxford Advanced Research Computing (ARC) and SCAN facilities in carrying out this work.

References

[1] A. Zhan, R. Zhao, L. Pinto, P. Abbeel, and M. Laskin. Learning visual robotic control efficiently with contrastive pre-training and data augmentation, 2022.

[2] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

[3] T. Haarnoja, A. Zhou, P. Abbeel, and S. 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.

[4] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. 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.

[5] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, K. Ghasemipour, C. Finn, and A. Wahid. Aloha unleashed: A simple recipe for robot dexterity, 2024. URL https://arxiv.org/abs/2410.13126.

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

[7] L. Bagesteiro and R. 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.

[8] M. Drolet, S. Stepputtis, S. Kailas, A. Jain, J. Peters, S. Schaal, and H. B. Amor. A comparison of imitation learning algorithms for bimanual manipulation. *IEEE Robotics and Automation Letters*, 2024.

[9] J. Yamada, J. Collins, and I. Posner. Efficient skill acquisition for complex manipulation tasks in obstructed environments, 2023.

[10] J. Collins, M. Robson, J. Yamada, M. Sridharan, K. Janik, and I. Posner. Ramp: A benchmark for evaluating robotic assembly manipulation and planning. *IEEE Robotics and Automation Letters*, 2023.

[11] W. Yuan, A. Murali, A. Mousavian, and D. Fox. M2t2: Multi-task masked transformer for object-centric pick and place. *arXiv preprint arXiv:2311.00926*, 2023.

[12] A. Mousavian, C. Eppner, and D. 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.

[13] K. R. Barad, A. Orsula, A. Richard, J. Dentler, M. Olivares-Mendez, and C. Martinez. Graspldm: Generative 6-dof grasp synthesis using latent diffusion models. *IEEE Access*, 2024.

[14] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on robot learning*, pages 651–673. PMLR, 2018.

[15] L. Wang, Y. Xiang, W. Yang, A. Mousavian, and D. Fox. Goal-auxiliary actor-critic for 6d robotic grasping with point clouds. In *Conference on Robot Learning*, pages 70–80. PMLR, 2022.

[16] T. G. W. Lum, M. Matak, V. Makoviychuk, A. Handa, A. Allshire, T. Hermans, N. D. Ratliff, and K. V. Wyk. Dextrah-g: Pixels-to-action dexterous arm-hand grasping with geometric fabrics, 2024. URL https://arxiv.org/abs/2407.02274.

[17] A. Zhou, M. J. Kim, L. Wang, P. Florence, and C. 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.

[18] S. Song, A. Zeng, J. Lee, and T. 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.

[19] Z. Sun, K. Yuan, W. Hu, C. Yang, and Z. Li. Learning pregrasp manipulation of objects from ungraspable poses, 2020.

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

[21] T. G. W. Lum, O. Y. Lee, C. K. Liu, and J. Bohg. Crossing the human-robot embodiment gap with sim-to-real rl using one human demonstration, 2025. URL https://arxiv.org/abs/2504.12609.

[22] T. Lin, Z.-H. Yin, H. Qi, P. Abbeel, and J. Malik. Twisting lids off with two hands. *arXiv preprint arXiv:2403.02338*, 2024.

[23] B. Huang, Y. Chen, T. Wang, Y. Qin, Y. Yang, N. Atanasov, and X. Wang. Dynamic handover: Throw and catch with bimanual hands, 2023.

[24] L. Shao, T. Migimatsu, and J. 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.

[25] R. Chitnis, S. Tulsiani, S. Gupta, and A. Gupta. Efficient bimanual manipulation using learned task schemas. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1149–1155. IEEE, 2020.

[26] R. Chitnis, S. Tulsiani, S. Gupta, and A. Gupta. Intrinsic motivation for encouraging synergistic behavior. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJleNCNtDH.

[27] Y. Li, C. Pan, H. Xu, X. Wang, and Y. 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.

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

[29] J. Grannen, Y. Wu, B. Vu, and D. 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.

[30] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots, 2024. URL https://arxiv.org/abs/2402.10329.

[31] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[32] M. J. Kim, F. Beck, C. Ott, and A. 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.

[33] A. L. Mitchell, T. Flatscher, and I. Posner. Task and joint space dual-arm compliant control, 2025. URL https://arxiv.org/abs/2504.21159.

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

[35] 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, 2022. URL https://arxiv.org/abs/2204.11918.

[36] O. 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.

[37] C. Eppner, A. Mousavian, and D. 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.

[38] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[39] 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.

[40] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. Van Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, et al. Curobo: Parallelized collision-free minimum-jerk robot motion generation. *arXiv preprint arXiv:2310.17274*, 2023.

[41] J. Yamada, M. Rigter, J. Collins, and I. 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] J. Brosseit, B. Hahner, F. Muratore, M. Gienger, and J. Peters. Distilled domain randomization. *arXiv preprint arXiv:2112.03149*, 2021.

[43] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu. 3d diffusion policy. *arXiv preprint arXiv:2403.03954*, 2024.

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

[45] 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*, pages 17868–17879, 2024.

[46] W. Wan, H. Geng, Y. Liu, Z. Shan, Y. Yang, L. Yi, and H. Wang. Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning, 2023. URL https://arxiv.org/abs/2304.00464.

[47] D. Coleman, I. A. Sucan, S. Chitta, and N. 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.

[48] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. 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.

[49] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

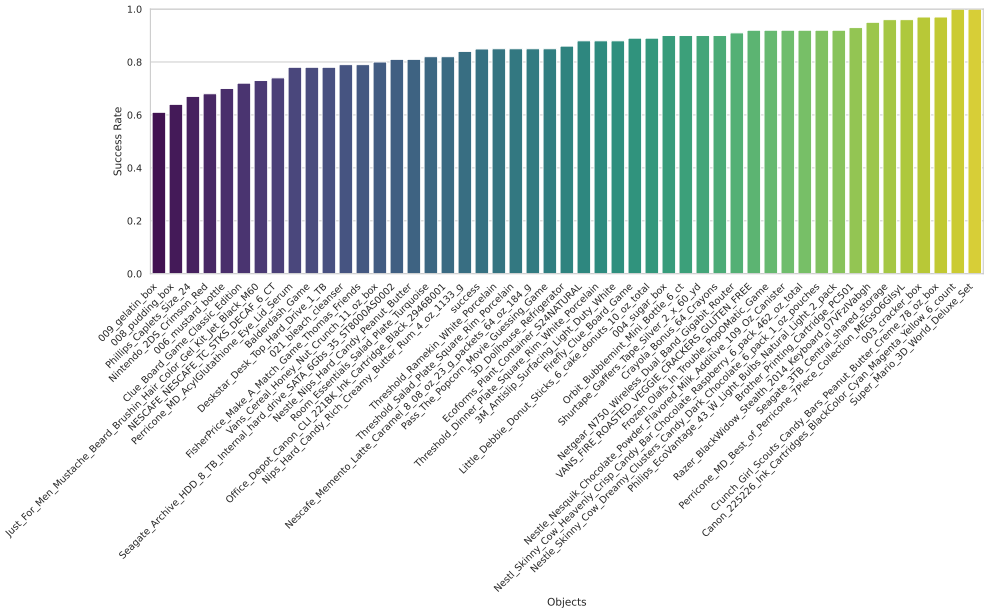

Figure 8: **Grasp success rates of *COMBO-Grasp*'s student policies.** The success rate is reported for each object in simulation, averaged over 50 trials per object.

A Additional Analysis for Experiments

A.1 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. 3) 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.

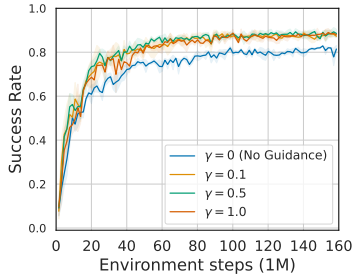

Figure 7: **Guidance scaling ablation.** We compare the guidance scaling parameter to steer the output of the constraint policy. This result indicates that *COMBO-Grasp* without guidance shows worse performance and *COMBO-Grasp*is robust to a wide range of guidance scaling parameters to achieve better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

A.2 Student Policy Performance per Object

Figure 8 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 [48].

	COMBO-Grasp	w/o grasp pose	w/ fixed constraint	w/o refinement	PPO
Cuboid-Medium-Heavy (Seen)	80% (8/10)	80% (8/10)	60% (6/10)	60% (6/10)	60% (6/10)
Cuboid-Large-Light	90% (9/10)	80% (8/10)	40%(4/10)	30% (3/10)	30% (3/10)
Cuboid-Small-Heavy	50% (5/10)	60% (6/10)	50% (5/10)	50% (5/10)	40% (4/10)
Keyboard	80% (8/10)	40%(4/10)	40% (4/10)	30% (3/10)	10% (1/10)
Bag	80% (8/10)	80% (8/10)	60% (4/10)	80% (8/10)	40% (4/10)
Round-Large-Light	30% (3/10)	10% (1/10)	0% (0/10)	10% (1/10)	0% (0/10)
Average	**68.3**% (**41**/**60**)	58.3% (35/60)	38.3%(23/60)	43.3%(26/60)	30.0% (18/60)

Table 2: Performance of *COMBO-Grasp* in real-world environments for seen and unseen objects with varying shapes, sizes, and weights.

A.3 Real-world Experiments

Table 2 presents the full results of the real-world experiments, including comparisons with all baseline methods. We observe that baseline approaches often fail to solve the tasks due to poor coordination between the left and right arms, likely because such coordination is not adequately learned during training in simulation, and consequently does not transfer well to real-world environments.

B Teacher Policy Details

B.1 Teacher Constraint Policy

We employ a diffusion policy [49] 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}.

B.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

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

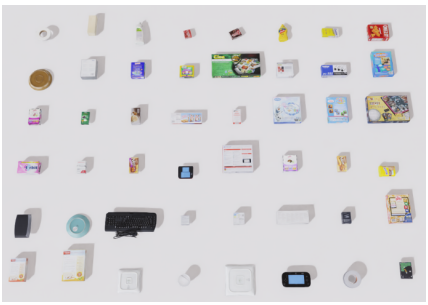

Figure 9: **Training objects.** We choose 48 training objects from the Google Scanned Object Dataset [35].

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

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.

B.3 Reward function

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

$$\begin{aligned} r = \alpha_1 r_{dist\_pos} + \alpha_2 r_{dist\_ori} - \alpha_3 r_{collision} \\ - \alpha_4 r_{action} + \alpha_5 r_{lift} + \alpha_6 r_{success} \end{aligned} \tag{4}$$

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:

$$r_{\text{dist\_pos}} = 1 - \tanh(4 \cdot ||\mathbf{p}_{\text{left}} - \mathbf{p}_{\text{grasp}}||_2), \tag{5}$$

  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:

$$r_{\text{dist\_ori}} = 1 - \tanh(0.2 \cdot ||\boldsymbol{\theta}_{\text{left}} - \boldsymbol{\theta}_{\text{grasp}}||_2), \tag{6}$$

  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:

$$r_{\text{action}} = ||\mathbf{a}||_2. \tag{7}$$

- **Collision Penalty ($r_{\text{collision}}$):** To prevent self-collisions and contact with the table, we compute the signed distance (SD) using CuRobo [40]. The collision penalty is given by:

$$r_{\text{collision}} = SD_{\text{self\_col}} + SD_{\text{table}}. \tag{8}$$

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:

$$r_{\text{lift}} = \mathbb{1}(z_{\text{grasp}} > z_{\text{grasp,init}} + 2\text{ cm}), \tag{9}$$

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:

$$r_{\text{success}} = \begin{cases} 1, & \text{if grasp and lift are successful,} \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

C Student Policy Details

We describe the architecture of the student constraint and grasping policy, as shown in Fig. 11.

C.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}.

C.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

D.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 9. We select objects such that successful picking requires occluded grasping. In particular, relatively flat objects that are difficult to grasp at the

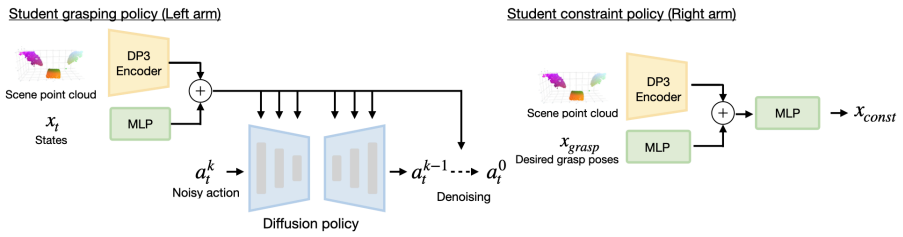

Figure 11: **Student policy architecture.** We utilize DP3 [43] as the backbone for the grasping policy. The DP3 encoder processes the scene point cloud, and its output is concatenated with a state feature vector obtained by a multi-layer perceptron (MLP). The resulting concatenated vector serves as 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.

annotated target poses are deliberately included. 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 3 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 3: 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

D.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 10).

E Real-World Experiment Setup

E.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.

E.1.1 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.

E.1.2 Execution

As illustrated in Fig. 1, the constraint policy first predicts a desired constraint pose, and the constraint arm is positioned accordingly using motion planning. Once the constraint arm is in place, the grasp policy controls the grasping arm to perform the occluded grasp. At the end of the episode, the grasping arm automatically closes its gripper while the constraint arm returns to the home pose. Finally, the grasping arm lifts the object to complete the task.

F Baseline Method Details

F.1 PPO

We train a policy using Proximal Policy Optimization (PPO) [44], 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.

F.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 $r_{right\_dist} = ||T^{obj} - T^{RightEE}||_2$.

F.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*.