

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

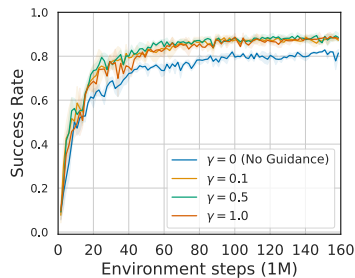


Figure 7: **Guidance scaling ablation.** We compare the guidance scaling parameter to steer the output of the constraint policy. This result indicates that *COMBO-Grasp* without guidance shows worse performance and *COMBO-Grasp* is robust to a wide range of guidance scaling parameters to achieve better 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 [45].

	<i>COMBO-Grasp</i>	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 [46] 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 $[\mathbf{x}_{robot}, \mathbf{x}_{goal}, \mathbf{x}_{obj}] \in \mathbb{R}^{64}$. The robot proprioceptive states, \mathbf{x}_{robot} , include the left end-effector pose, $\mathbf{x}_{left} \in \mathbb{R}^9$, the right end-effector pose, $\mathbf{x}_{right} \in \mathbb{R}^8$, and the translational and rotational action scale parameters for the operational space controller, $\mathbf{x}_{control} \in \mathbb{R}^2$. 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}_{goal} , consist of the desired grasp pose, $\mathbf{x}_{grasp} \in \mathbb{R}^7$, the distance between the left end-effector and the desired grasp position, $\mathbf{x}_{dist} \in \mathbb{R}^3$, and the orientation distance between the left end-effector and the desired grasp orientation in the axis-angle representation, $\mathbf{x}_{dist-ori} \in \mathbb{R}^3$. The object states, \mathbf{x}_{obj} , comprise the object pose, $\mathbf{x}_{obj-pose} \in \mathbb{R}^7$, the object velocity, $\mathbf{x}_{obj-vel} \in \mathbb{R}^6$, the friction parameters, $\mathbf{x}_{friction} \in \mathbb{R}^2$, the object’s mass, $x_{mass} \in \mathbb{R}^1$, and the object’s ID, $\mathbf{x}_{obj-id} \in \mathbb{R}^{16}$.

We train the policy using an Adam optimiser with an adaptive learning rate scheduler¹ 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

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

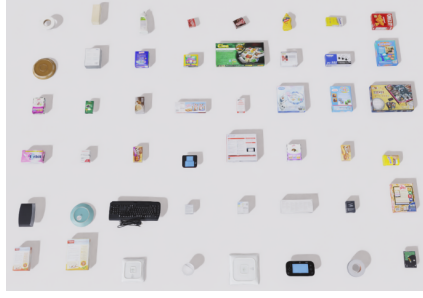


Figure 9: **Training objects.** We choose 48 training objects from the Google Scanned Object Dataset [32].

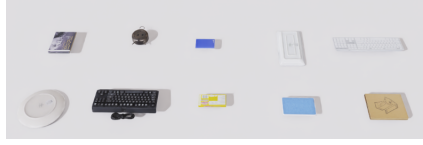


Figure 10: **Test objects.** We evaluate 10 held-out objects from the Google Scanned Object Dataset.

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:

$$r = \alpha_1 r_{dist_pos} + \alpha_2 r_{dist_ori} - \alpha_3 r_{collision} - \alpha_4 r_{action} + \alpha_5 r_{lift} + \alpha_6 r_{success} \quad (4)$$

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:

$$r_{dist_pos} = 1 - \tanh(4 \cdot \|\mathbf{p}_{left} - \mathbf{p}_{grasp}\|_2), \quad (5)$$

where $\mathbf{p}_{left} \in \mathbb{R}^3$ and $\mathbf{p}_{grasp} \in \mathbb{R}^3$ 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 space⁴. The reward is computed as:

$$r_{dist_ori} = 1 - \tanh(0.2 \cdot \|\boldsymbol{\theta}_{left} - \boldsymbol{\theta}_{grasp}\|_2), \quad (6)$$

where $\boldsymbol{\theta}_{left} \in \mathbb{R}^3$ and $\boldsymbol{\theta}_{grasp} \in \mathbb{R}^3$ 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:

$$r_{action} = \|\mathbf{a}\|_2. \quad (7)$$

- **Collision Penalty ($r_{collision}$):** To prevent self-collisions and contact with the table, we compute the signed distance (SD) using CuRobo [37]. The collision penalty is given by:

$$r_{collision} = SD_{self_col} + SD_{table}. \quad (8)$$

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:

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

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:

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

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 [40] 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 $\mathbf{x}_{\text{grasp}}$.

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) [40] 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 $\mathbf{x}_{\text{robot}}$ and optionally $\mathbf{x}_{\text{grasp}}$ 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. To train teacher policies efficiently, we spawn 1024 robots and objects in the simulated environment.

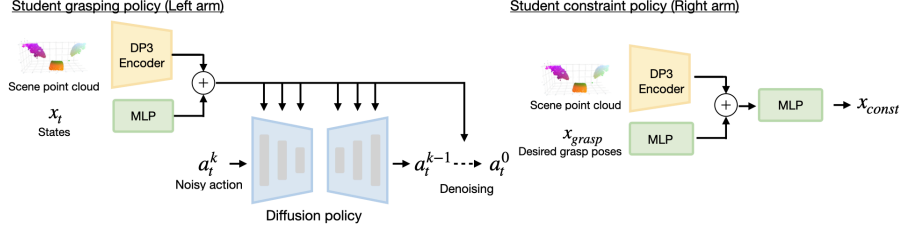


Figure 11: **Student policy architecture.** We utilize DP3 [40] as the backbone for the grasping policy. The DP3 encoder processes the scene point cloud, and its output is concatenated with a state feature vector obtained by a multi-layer perceptron (MLP). The resulting concatenated vector serves as 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.

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 $\mathcal{U}(-0.2\pi \text{ rad}, 0.2\pi \text{ rad})$ 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 [40], 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.

597 **E.1.1 Desired Occluded Grasp Pose Generation**

598 In order to scan an object to reconstruct a mesh, we use Polycam, an application that captures
599 pictures of objects and reconstructs an object mesh using Neural Radiance Fields (NeRF). Using the
600 reconstructed mesh, we generate desired occluded grasp poses using antipodal sampling.

601 **F Baseline Method Details**

602 **F.1 PPO**

603 We train a policy using Proximal Policy Optimization (PPO) [41], where the policy outputs 12-
604 dimensional delta end-effector poses corresponding to both the left and right arms. We use the same
605 hyperparameters employed for training *COMBO-Grasp*, except for the entropy coefficient, which is
606 set to 0.003. This modification was made because using the original entropy coefficient caused a
607 continuous increase in the policy’s standard deviation, resulting in the policy’s inability to exploit a
608 stable and effective strategy during training.

609 **F.2 PPO + Constraint Reward**

610 Similar to the *PPO* baseline, but we introduce an additional reward term that encourages the right
611 arm to be used as a constraint. In particular, we add a reward $r_{right.dist} = ||T^{obj} - T^{RightEE}||_2$.

612 **F.3 *COMBO-Grasp* w/ Fixed Constraint**

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