Proceedings of Machine Learning Research vol 242:615–627, 2024 6th Annual Conference on Learning for Dynamics and Control

Efficient Skill Acquisition for Insertion Tasks in Obstructed Environments

Jun Yamada JYAMADA@ROBOTS.OX.AC.UK
Jack Collins JCOLLINS@ROBOTS.OX.AC.UK
Ingmar Posner INGMAR@ROBOTS.OX.AC.UK
Oxford Robotics Institute, 23 Banbury Road, Oxford, UK

Editors: A. Abate, M. Cannon, K. Margellos, A. Papachristodoulou

Abstract

Data efficiency in robotic skill acquisition is crucial for operating robots in varied small-batch assembly settings. To operate in such environments, robots must have robust obstacle avoidance and versatile goal conditioning acquired from only a few simple demonstrations. Existing approaches, however, fall short of these requirements. Deep reinforcement learning (RL) enables a robot to learn complex manipulation tasks but is often limited to small task spaces in the real world due to sample inefficiency and safety concerns. Motion planning (MP) can generate collision-free paths in obstructed environments, but cannot solve complex manipulation tasks and requires goal states often specified by a user or object-specific pose estimator. In this work, we propose a robust system for efficient skill acquisition designed to address complex insertion tasks in obstructed environments. Our system leverages an object-centric generative model (OCGM) for versatile goal identification to specify a goal for MP combined with RL to solve complex manipulation tasks in obstructed environments. Particularly, OCGM enables one-shot target object identification and re-identification in new scenes, allowing MP to guide the robot to the target object while avoiding obstacles. This is combined with a skill transition network, which bridges the gap between terminal states of MP and feasible start states of a sample-efficient RL policy. The experiments demonstrate that our OCGM-based one-shot goal identification provides competitive accuracy to other baseline approaches and that our modular framework outperforms competitive baselines, including a state-of-the-art RL algorithm, by a significant margin for complex manipulation tasks in obstructed environments.

Keywords: Robotic Manipulation, Integrated Planning and Learning, Reinforcement Learning, Motion Planning, Learning from Demonstration

1. Introduction

Teaching new skills to robots using limited supervision is essential for maximising the up-time and productivity of robots, leading to faster return on investment. Small-batch manufacturing, where there are a limited number of parts to be produced, is an exemplary environment that would greatly benefit from efficient skill acquisition. In a small-batch setting, a robot must learn to manipulate new objects while maintaining data efficiency in potentially arbitrarily obstructed environments. However, existing methods for controlling a manipulator such as motion planning and reinforcement learning individually struggle to satisfy such requirements.

Motion planning (MP) (Amato and Wu, 1996; LaValle, 1998) generates collision-free paths capable of guiding a robot safely in obstructed environments given an explicit state of the environment and goal. However, MP is not designed to plan through complex manipulation tasks requiring environmental interaction. Furthermore, MP necessitates the specification of a goal state in the robot's frame of reference, which is typically accomplished through manual engineering (Khodeir et al.,

© 2024 J. Yamada, J. Collins & I. Posner.

2021), template matching (Le et al., 2019), or an object-specific pose estimator (Lee et al., 2020) trained on manually labelled supervised data.

Deep reinforcement learning (RL), on the other hand, has shown promising outcomes in learning to control a robot for complex manipulation tasks such as grasping (Kalashnikov et al., 2018; Zhan et al., 2020) and insertion (Luo et al., 2021). However, prior works often limit operation to simulated environments (Haarnoja et al., 2018) or heavily restrict and regulate operating spaces by executing with a short horizon without obstructions (Luo et al., 2021; Zhan et al., 2020) due to the sample inefficiency and potential of executing unsafe policies.

Combining MP and RL has been investigated by several prior works (Yamada et al., 2020; Lee et al., 2020) and shows the potential of leveraging the strengths of both methods to solve manipulation tasks in obstructed environments. Yet, goal specification for MP in prior work has relied on either sample-inefficient interaction with the environment or an object-specific pose estimator, which needs retraining for each new target object. Notably, MoPA-RL (Yamada et al., 2020) attempts to solve similarly complex manipulation tasks but requires more than 1M samples to train the RL policy from state-based observations with fixed obstacle positions, limiting the real-world application.

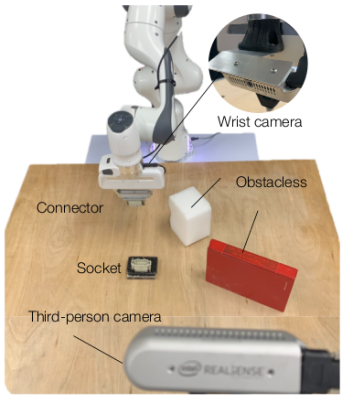

Figure 1: **Task setup.** We solve complex manipulation tasks within the entire operational space of a robot by leveraging an OCGM for versatile and efficient goal acquisition paired with MP and RL. Note that obstacles and a socket are randomly placed on the table.

Inspired by the challenges faced in small-batch manufacturing problems, we introduce a system that builds upon existing MP and RL algorithms, integrating them with an object-centric generative model (OCGM) (Wu et al., 2021) to overcome the limitations of existing methods. We posit that the integration of an OCGM leads to versatile, one-shot goal identification and re-identification, allowing for insertion tasks to be solved from a limited number of simple human demonstrations.

Specifically, we identify a target object from a *single* demonstration using an OCGM, pre-trained on diverse synthetic scenes. Matching the target object's object-centric representation to those in new scenes leads to robust object re-identification. Using the object's position as a goal, the motion planner generates a collision-free path to the target object while avoiding obstacles before a learned RL policy is executed to complete the insertion tasks. We train an RL policy for each insertion skill from a sparse reward to eliminate the need for reward engineering using specialist knowledge. We also leverage a handful of easy-to-collect demonstrations to guide exploration to achieve efficient RL policy learning. To maximise performance, we also introduce a skill transition network to reduce failures that occur when transitioning from MP to the learned RL policy.

The contributions of our work are fourfold: (1) we propose a system for efficient skill acquisition in obstructed environments that leverages an OCGM for object-agnostic to overcome the limitations of existing methods, *one-shot* goal specification, (2) we introduce a transition network that smoothly interpolates between terminal states of motion planning and feasible start states of a learned RL policy to significantly improve the successes rate of the approach, (3) we show that our OCGM-based one-shot goal specification method achieves comparable accuracy against several traditional and object-specific goal identification baselines, and (4) we demonstrate that our

system performs significantly better in real-world environments compared to baselines, including a state-of-the-art RL algorithm. In summary, our paper introduces a novel system that leverages prior MP and RL methods while distinguishing itself by eliminating the need for prior knowledge such as object geometry or object-specific detectors. We demonstrate the effectiveness of using unsupervised OCGMs to combine MP and an RL policy, making our approach particularly valuable in small-batch settings, which is our specific focus. While the individual building blocks exist, their seamless integration in a real-world robot system remains a challenging and novel achievement.

2. Related Works

Recent success in deep RL (Kalashnikov et al., 2018; Haarnoja et al., 2018) enables a robot to learn complex manipulation tasks such as grasping (Kalashnikov et al., 2018; Zhan et al., 2020) and insertion (Luo et al., 2021; Vecerik et al., 2018; Lee et al., 2018; Davchev et al., 2022; Carvalho et al., 2022) driven by a reward. To avoid the requirement of specialist knowledge for reward engineering, several prior works have proposed sample-efficient RL methods that can learn complex manipulation skills from a sparse reward by leveraging a small number of demonstrations for guided exploration (Zhan et al., 2020; Luo et al., 2021; Vecerik et al., 2017, 2018). However, due to the sample inefficiency of sparse rewards, studies have been primarily conducted in simulated environments or within limited task spaces in the real world. Learning from demonstration (LfD) (Schaal, 1999; Billard et al., 2008; Groth et al., 2021) is an alternative method for a robot to learn manipulation tasks by imitating behaviour in expert demonstrations collected by a human operator, but it often requires a large number of demonstrations to acquire manipulation skills. While InsertionNet (Spector and Di Castro, 2021) enables a robot to solve insertion tasks within the entire operational space of a robot manipulator from a small number of demonstrations, it is evaluated in a clean environment without obstruction. Successful insertion is also made possible by a small initiation set for the learnt skill. Adaptive LfD for insertion has also been proposed (Wen et al., 2022), allowing a policy to quickly adapt to new insertion objects from the same category seen in training using only a single demonstration and the object mesh. However, such mesh information is not readily available, limiting real-world applications. In our work, a manipulation skill is learnt using Framework for Efficient Robot Learning (FERM)(Zhan et al., 2020).

Motion planning (MP) (Amato and Wu, 1996; Kavraki and Latombe, 1994; LaValle, 1998) can effectively generate a collision-free path from a robot's initial configuration to a goal pose using an explicit model of the robot and environment. However, such a goal pose is often specified by a user or object-specific pose estimator. Further, complex manipulation tasks such as insertion are out of the scope of MP as MP does not model the dynamics of the surrounding environment or objects.

Several previous works (Yamada et al., 2020; Lee et al., 2020; Kuo et al., 2021) combine MP and RL to leverage the benefits of both methods to solve manipulation tasks in unstructured environments. However, these preceding works limit their real-world applicability by requiring a large number of samples to learn a goal estimator (Yamada et al., 2020) or by retraining an object-specific predictor for each new goal object (Lee et al., 2020). While MoPA-RL (Yamada et al., 2020) is most closely related to our method in spirit, it requires more than 1M samples to train an RL policy from state-based observations with fixed obstacle positions. Thus, the prior work is not directly comparable to our work due to its sample inefficiency and the need for inaccessible state observations in the real world.

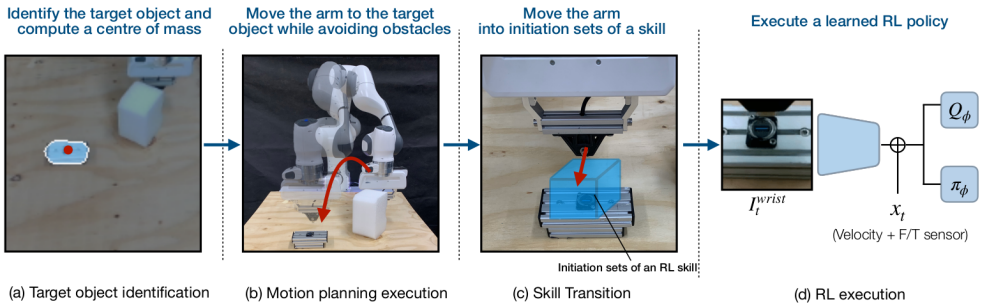

Figure 2: **Our framework architecture.** (a) We leverage an OCGM to re-identify a target object such that its object-centric representation matches one extracted from a single demonstration. The goal state is specified in the robot's reference frame using an external RGB-D camera with calibrated extrinsics. (b) Given the goal state acquired in (a), a motion planner generates a collision-free path to the goal. (c) A skill transition network guides the arm from the terminal state of the motion planning (MP) to the initiation set of the RL policy. (d) Given a wrist camera image $\boldsymbol{I}_t^{wrist}$ and robot's internal state $\boldsymbol{x}_t$, a learned RL policy executes the final interaction until task completion.

Our work leverages unsupervised OCGMs (Wu et al., 2021; Locatello et al., 2020; Lin et al., 2020) to find a target object for MP, negating the need for object-specific goal estimators. OCGMs learn structured representations of objects within complex scenes and provide a set of object-centric embeddings useful for instance matching. In contrast to goal specification methods that require human intervention or a large, object-specific datasets, i.e. template matching or object classifier, OCGMs hold the promise of versatile target object identification. While several prior works (Kirillov et al., 2023; Xie et al., 2021) introduce instance segmentation methods for unseen objects, these methods do not provide a dense description of objects suitable for instance matching. Specifically, this work leverages APEX (Wu et al., 2021) an unsupervised model trained on a wide distribution of simulated data to assist with generalisation to real-world environments.

3. Methodology

In this work, we present an efficient solution for solving insertion tasks in obstructed, real-world environments by leveraging an OCGM. We demonstrate our system on several insertion tasks as they require learning complex insertion skills and also require the identification of the target socket to complete the tasks (see Figure 1 for our task setup). We break our method down into pre-training, task-specific skill training and execution in the following subsections. The pre-training component of our method is only completed *once* and can be reused for all future insertion tasks. Task-specific skill training is required for each new insertion task and execution describes the process for autonomous task execution after training.

3.1. Pre-training

Pre-training is required for APEX (Wu et al., 2021), our choice of unsupervised OCGM, to achieve versatile one-shot target object acquisition, but it only needs to be done once as it is trained on a diverse synthetic dataset collected in simulation to encourage generalisation to a variety of real-world objects. APEX is formulated as a set of VAEs, and takes a video sequence $\boldsymbol{I}_{1:T}$ as input. Each

frame is decomposed into a set of latent representations for each discovered object j consisting of object location $\boldsymbol{z}_{t,j}^{where}$, appearance $\boldsymbol{z}_{t,j}^{what}$, and presence $z_{t,j}^{pre} \in [0, 1]$, where T is the number of frames in a video sequence. We train APEX on a synthetic dataset consisting of a set of trajectories in which a robot interacts with a diverse set of primitive shapes of differing colour and size. In order to successfully transfer APEX trained on synthetic data to the real world, we add a small amount of noise to the camera pose for each trajectory, leading to variations in the images. As a result, APEX is successfully applied to real-world scenes with similar background textures.

3.2. Task-Specific Skill Training

This section details the task-specific data and training required by our method. The data must be collected for each new task that the robot is taught, however, the supervised component only requires about 10 minutes to collect. First, a single demonstration, $\mathcal{D}^{goal} = \{(\boldsymbol{I}_t^{ext}, \boldsymbol{x}_t^{ee}), \dots\}$ consisting of a sequence of images $\boldsymbol{I}^{ext}$ from the third-person camera and robot end-effector positions $\boldsymbol{x}_t^{ee}$, of a successful task completion from anywhere within the robot's operational space is collected for goal specification using the pre-trained APEX. Additionally, 25 demonstrations, $D^{RL} = \{(\boldsymbol{I}_t^{wrist}, \boldsymbol{x}_t, \boldsymbol{a}_t), \dots\}^{|25|}$, of the insertion skill for efficient RL training, are collected from within a limited task space such that the connector is always within sight of the wrist camera, where $\boldsymbol{I}_t^{wrist}$, $\boldsymbol{x}_t$, $\boldsymbol{a}_t$ are wrist camera image, robot states including 3-dimensional Cartesian end-effector velocity and F/T sensor data, and action at time step t.

We employ FERM (Zhan et al., 2020) to train the RL policy π_θ along with a critic function Q_ϕ parameterised by π to complete the insertion task. This takes between 60 to 90 minutes to train on a desktop computer with an i7 processor and a Nvidia Titan X GPU. FERM is composed of Soft Actor Critic (Haarnoja et al., 2018), contrastive learning (Laskin et al., 2020b), and image augmentation (Laskin et al., 2020a). In addition to a gray-scale image from the wrist-mounted camera ($\boldsymbol{I}^{wrist[H\times W]}$ where H and W are 64 pixels), the policy takes as input the end-effector Cartesian velocity and F/T sensor data (see Figure 2 (d)) and outputs the desired 3-dimensional Cartesian end-effector velocity for the robot. Because the policy takes as input local information, it is able to generalise to any location in the robot's operational space. The RL policy is trained using a sparse reward $r_t = \mathbb{1}[\boldsymbol{s} \in S_g]$ where S_g is a set of goal states defined as the average termination state of the collected demonstrations within a $1cm$ tolerance. To accelerate training of the RL policy, we leverage the task demonstrations D^{RL} to initialise a replay buffer for guided exploration similar to FERM and train the policy and critic asynchronously, as inspired by prior work (Luo et al., 2021). We also limit the task space for this stage, such that the socket is always within sight of the wrist camera for training, improving sample efficiency and reducing the chance of unsafe interactions. The initial states of the policy are positioned above the socket, with a small random noise added to their positions, sampled from a uniform distribution between -0.02cm and 0.02cm.

We also introduce a skill transition network, inspired by prior work (Johns, 2021), for each new insertion task to improve the task success rate. The terminal state of the MP is not guaranteed to be within the feasible start states of the RL policy, defined as *the initiation set of the skill* (represented as a blue box in Figure 2 (c)), due to the error in estimating the MP goal state in 3D space, caused by errors in camera extrinsics and noisy depth estimation from the RGB-D camera. To mitigate this issue, a simple convolutional neural network (CNN) is trained on data collected in a self-supervised manner to predict the Cartesian offset required to move the end-effector from the terminal states of MP to the initiation set of the RL policy. The dataset is collected in less than 30 minutes by sampling

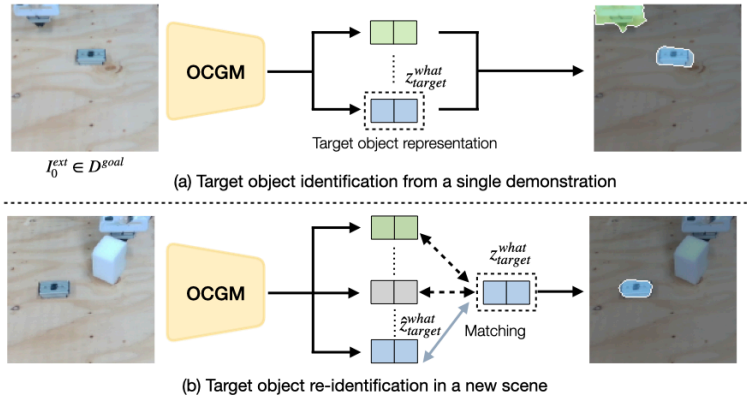

Figure 3: **Target object identification and re-identification using an OCGM.** (a) We leverage a pre-trained OCGM to extract an object-centric representation from a single task demonstration. The target object is identified in the demonstration as the object mask closest to the robot end-effector position at the end of the trajectory $\boldsymbol{x}^{ee}_T$ (see Eq 1). (b) Given a new scene, the OCGM is used to acquire object-centric representations of all objects and compare these with the already identified target object representation to re-identify the target object.

random Cartesian poses around the target object and recording the sequences of wrist camera images $\boldsymbol{I}^{wrist}_t$ and offsets between the current end-effector pose and an initial pose used for RL training. The collected data contains only local information conditioned on the wrist camera which allows the transition network to generalise to unseen target positions. Crucially, while an RL policy could be trained with a wider initial state distribution, this is well understood in literature (Yamada et al., 2020; Lee et al., 2020; Nair et al., 2018), to be sample inefficient. Instead, we train a skill transition network using a labelled dataset collected in a self-supervised fashion which is akin to Behaviour Cloning (Zhang et al., 2018).

3.3. Execution

The execution of the task can be completed from *anywhere* within the robot's operational space to any goal location. Execution follows four steps (see Figure 2) that are completed autonomously: (i) goal identification via OCGM, (ii) MP, (iii) skill transition network, and (iv) RL policy.

Goal Identification via OCGM As MP requires a goal pose to plan a collision-free path through the scene, we leverage the pretrained OCGM to identify the target object from the single demonstration D^{goal} and re-identify it in the current scene to specify the goal for MP. To identify the target object from D^{goal}, we first acquire a set of object-centric representations by encoding the first external camera image $\boldsymbol{I}^{ext}_0$ in the demonstration D^{goal} (see Figure 3 (a)). We determine the target object-centric representation $\boldsymbol{z}^{what}_{target}$ such that the object is present at the beginning of the trajectory, i.e. $p(z^{pre}_{0,j}) \geq 0.5$ and such that it is the closest to the robot end-effector position $\boldsymbol{x}^{ee}_T$ at the end of the demonstration D^{goal}. To calculate the 3D position of objects in the robot's reference frame, the centre of the object mask predicted by the OCGM is converted to Cartesian coordinates using the RGB-D camera's depth plane and the known camera extrinsics. The closest object to the robot

end-effector position at the end of the demonstration D^{goal} is calculated using L_2 distance:

$$target = \underset{j,j=1..N}{\arg\min} ||\boldsymbol{x}_T^{ee} - \boldsymbol{o}_j||_2 \tag{1}$$

where o_j is the 3D object positions in the robot reference frame and N is the number of objects discovered by the OCGM in the scene $\boldsymbol{I}_0^{ext}$. In order to re-identify the target object in the current scene (see Figure 3 (b)), we compare the target object-centric representation $\boldsymbol{z}_{target}^{what}$ with each object-centric representation $\hat{\boldsymbol{z}}_j^{what}$ discovered in the new scene (see Figure 3 (b)) using the L_2 distance and choose the object that has the most similar representation:

$$\hat{\boldsymbol{z}}_{target}^{what} = \underset{j=1..N}{\arg\min} ||\boldsymbol{z}_{target}^{what} - \hat{\boldsymbol{z}}_j^{what}||_2 \tag{2}$$

MP + Transition Policy + RL Policy Using the target object's pose $\boldsymbol{o}_{target}$ in the robot's reference frame, we use an RRT-connect motion planner to guide the robot's end-effector to the location of the target object (see Figure 2 (b)). To avoid collisions during the MP phase, an occupancy map, OctoMap (Hornung et al., 2013), is created using the point clouds captured by the calibrated external camera. After the execution of MP, we leverage the trained skill transition network to guide the arm into the initiation set of the skill (see Figure 2(c)) to maximise the outcomes of the RL policy. Finally, the learned RL policy completes the manipulation task.

4. Experiments

Our experiments are designed to answer the following guiding questions: (1) does the use of an OCGM achieve versatile and efficient target object identification for MP in the real world? (2) how well does our system perform insertion tasks in obstructed environments? (3) does a skill transition network increase task success rate?

4.1. Experimental Setup

Several insertion tasks, inspired by the NIST assembly boards (Kimble et al., 2020), are used within our experiments (see Figure 4). To verify the robustness of the target object identification using an OCGM, sockets and obstacles with different colours and sizes are used. In our experiments, we use a Franka Panda robot (7-DOF robot arm) and rigidly attach each connector to the robot's end-effector similar to prior work (Luo et al., 2021). Each phase of our framework uses different controllers: a joint position controller to follow a trajectory planned by the motion planner, a Cartesian pose controller for the skill transition network, and a Cartesian velocity impedance controller for the RL policy. For each evaluation trial, the socket, robot arm, and one or two obstacles are randomly placed in the robot's operational space.

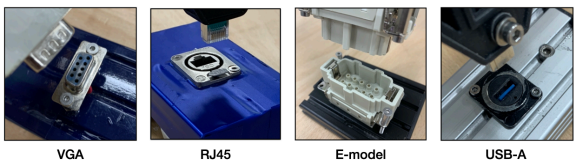

Figure 4: **Insertion tasks.** We evaluate our framework on four insertion tasks. Each socket is attached to a mount of varying size and colour to demonstrate the versatility and efficiency of our one-shot goal specification using an OCGM.

Given the pre-trained OCGM, for each insertion skill, our modular framework requires a total of 10 minutes of human-supervised demonstrations and a maximum of 130 minutes of unsupervised

	VGA		RJ45		E-model		USB-A		#Data	Intervention
Method	Accuracy	WSI	Accuracy	WSI	Accuracy	WSI	Accuracy	WSI		
Template matching	70.0%	54.7/81.9%	35.0%	22.1/50.5%	87.5%	73.9/94.5%	55.0%	39.8/69.3%	1	yes
Feature-based matching	40.0%	26.3/55.4%	87.5%	73.9/94.5%	7.5%	2.6/19.9%	77.5%	62.5/87.7%	1	yes
Object-specific classifier	80.0%	65.2/89.5%	**100.0%**	91.2/100.0%	87.5%	73.9/94.5%	75.0%	59.8/85.8%	2.5K	yes
OCGM identifier (Ours)	**82.5%**	68.0/91.3%	95.0%	83.5/98.6%	**95.0%**	83.5/98.6%	**92.5%**	80.1/97.4%	1	no

Table 1: **Accuracy of target object identification.** We evaluate our method and two baselines on 160 test scenes (40 scenes per connector), and report the accuracy, Lower Limit (LL) and Upper Limit (UL) of the Wilson score interval (WSI) with confidence interval of 95%. While template matching and object-specific classification requires human intervention, such as cropping a reference target object image and labelling training data, our OCGM identifier successfully identifies the target object from only a single demonstration without such human intervention.

training comprising of: up to 90 minutes for RL policy training and 40 minutes for data collection and training of the skill transition network.

4.2. Efficient and Versatile Target Object Identification

First, the OCGM identifier is evaluated against several baselines on 160 test scenes (40 for each of the four sockets) with the target locations hand-labelled with bounding boxes for quantitative comparison. During testing, if the intersection of union (IoU) between a ground truth and the returned bounding box from the tested algorithm is greater than 0.5, we count it as successful (Everingham et al., 2015).

We evaluate our proposed goal identification approach against three baselines. *Template matching* finds the target object in the current scene by calculating a correlation coefficient given a manually cropped target object reference image. *Feature-based matching* finds a pair of the best matched keypoints between the manually cropped target object reference image and the current scene using FLANN-based matching (Muja and Lowe, 2009) and SIFT descriptor (Lowe, 2004). *Object-specific classifier* trained on a dataset of manually cropped object images with binary labels, is queried with cropped images found using a region proposal method (Uijlings et al., 2013). Lastly, we evaluate our method by retrieving the minimum bounding box of the target object mask predicted by the OCGM.

Results. We report the accuracy of target object identification in Table 1. Our method achieves commensurate or better performance compared to other baselines, whilst not requiring human intervention or an object-specific dataset needing laborious manual data labelling. This result motivates the use of OCGMs for efficient goal acquisition for MP. The occlusion of objects that are sometimes considered as one object in APEX can lead to unsuccessful object-centric representation matching, resulting in a failure to identify the target. Also, if the object shape and colour look similar in an image, object-centric representation matching may fail. Template matching often struggles to find a target object with high confidence, potentially due to the complex scene composition and slanted third-person camera angle (see Fig. 1). Feature-based matching also shows a lower success rate for several objects due to a lack of distinguishing features, especially for small objects in a scene. The object-specific classifier, on the other hand, generally performs well because it is tailored to a single object, and could be further improved by collecting more data. However, such classifiers

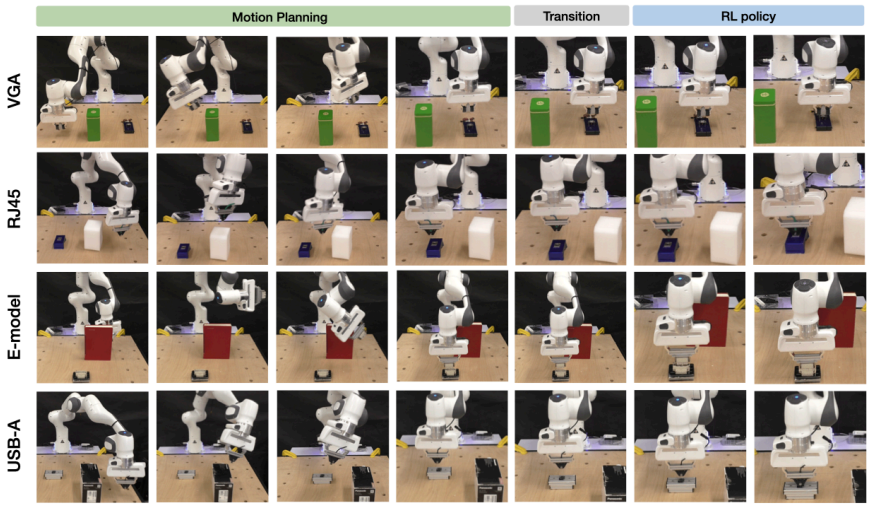

Figure 5: **Real-world industrial assembly tasks in obstructed environments.** The OCGM is used to specify a goal for MP, followed by a skill transition network and a learned RL policy. Our method successfully solves complex manipulation tasks with a high success rate.

require re-training on a new dataset manually labelled for each new object, limiting the versatility and efficiency of goal specification.

	VGA		RJ45		E-model		USB-A	
Method	Success	WSI	Success	WSI	Success	WSI	Success	WSI
SAC	0.0%	0.0/16.1%	0.0%	0.0/16.1%	0.0%	0.0/16.1%	0.0%	0.0/16.1%
MP + Demonstration Replay	3.3%	1.0/16.7%	0.0%	0.0%/16.1%	3.3%	1.0/16.7%	0.0%	0.0/16.1%
MP + BC	16.7%	7.3%/33.6%	16.7	7.3/33.6%	23.3%	11.8/40.9%	26.7%	14.2/44.5%
MP + Heuristic	10.0%	3.5/25.6%	16.7	7.3/33.6%	36.7%	21.9/54.5%	43.3%	27.4/60.8%
MP + RL w/o skill transition	73.3%	55.6/85.8%	46.7%	30.2/63.8%	80.0%	62.7/90.5%	70.0%	52.1/83.3%
MP + RL (our method)	**86.7%**	70.3/94.7%	**83.3%**	66.4/92.7%	**93.3%**	78.7/98.2%	**96.7%**	83.3/99.4%

Table 2: **Real-world assembly results.** We report the success rate, Lower Limit (LL) and Upper Limit (UL) of the Wilson score interval (WSI) with confidence interval of 95% over 30 trials. Our method outperforms, by a significant margin, all of the other methods including a state-of-the-art RL method and several comparable instantiations of our method.

4.3. Insertion Tasks in Obstructed Environments

We evaluate our proposed system on several insertion tasks in obstructed environments against a series of baselines composed of competing methods. All baselines that utilise MP make use of the OCGM for target object identification. For each task, we conduct 30 trials and report the success rate in Table 2. Figure 5 illustrates the execution of our method for each insertion task in the obstructed environments.

We compare the performance of our approach against a state-of-the-art RL algorithm and four comparable instantiations of our approach. *Soft Actor-Critic (SAC)* a state-of-the-art RL algorithm that predicts the desired Cartesian velocity from sparse rewards, trained with the same number of environmental interactions as our proposed method and similarly with 25 demonstrations preloaded

into the replay buffer, following FERM (Zhan et al., 2020). *MP+Demonstration Replay* substitutes replaying a single expert demonstration for the learned RL policy execution in our method, inspired by previous work (Johns, 2021). *MP+BC* replaces the learned RL policy in our method with Behaviour Cloning (BC) (Zhang et al., 2018), trained from 25 demonstrations. *MP+Heuristic* uses a manually designed heuristic policy (Luo et al., 2021) instead of the learned RL policy in our method to solve the task. Lastly, we evaluate our method without a skill transition network (*MP+RL w/o skill transition*) for comparison.

Results. As described in Table 2, our method (MP+RL) as outlined in Section 3 records the highest success rate for all tasks. The results for the *SAC* baseline show that it is unable to solve any of the tasks, likely because it requires a large number of samples to train the policies in the robot's operational space with obstructions. *MP+Demonstration Replay* is the most data-efficient method, however, it mostly fails to solve any of the tasks because it requires very accurate estimation of pose offsets for the demonstration replay to be successful. *MP+BC* is another efficient skill acquisition method because it does not require any additional interactions with the environments other than the given demonstrations to learn manipulation skills. However, due to the narrow state coverage, it struggles to solve the tasks. While *MP+Heuristic* is able to solve some insertion tasks, such as USB-A and E-model, almost one-third of the time, it fails to solve the tasks the majority of the time due to the need for accurate pose offset (the same reason for failure as *MP+Demonstration Replay*). While our method achieves high success rate over 4 industrial insertion tasks, the main failure case is caused by the misidentification of the target object by the OCGM. These failure modes can be readily eliminated by extended and/or augmenting the OCGM training.

Examining whether the transition network is useful for our system to solve complex manipulation tasks in obstructed environments (see Table 2), the results verify that using the skill transition network results in higher success rates than without the skill transition policy. Due to errors caused by the OCGM, camera extrinsics and estimation of the 3D goal poses, a terminal state of MP can often be outside of the initiation set of the learned RL skill. Therefore, by introducing the skill transition module to move the robot arm into the initiation set of the skill, we can mitigate these issues and achieve better performance.

5. Conclusion

In this work, we propose a modular system that leverages an OCGM for one-shot goal identification and re-identification as a vital component to combine MP and RL to solve complex manipulation tasks in obstructed environments. Specifically, the OCGM extracts a target object from only a single demonstration and re-identifies the object to determine a goal pose for MP without the need of fine-tuning on an object-specific dataset. The experimental results show that our method for goal specification using an OCGM achieves better versatility and comparable accuracy to other tested baselines. In addition, our method successfully solves real-world insertion tasks in obstructed environments from few demonstrations.

While the rotation around the z-axis of the sockets is consistent across all of the evaluation trials, we can readily extend our system to accommodate cases where the socket is rotated. We leave this extension to future work and anticipate overcoming the orientation misalignment by extending the skill transition network to additionally predict a z-axis displacement, correctly orientating the peg with respect to the socket. Any further small orientation errors could be overcome using an impedance controller and an RL policy trained with small perturbations in the z-axis orientation.

ACKNOWLEDGMENT

This work was supported by a UKRI/EPSRC Programme Grant [EP/V000748/1], we would also like to acknowledge the use of the University of Oxford Advanced Research Computing (ARC) (http://dx.doi.org/10.5281/zenodo.22558) and the SCAN facility in carrying out this work.

References

Nancy M Amato and Yan Wu. A randomized roadmap method for path and manipulation planning. In *IEEE International Conference on Robotics and Automation*, 1996.

A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Survey: Robot programming by demonstration. *Springer Handbook of Robotics*, pages 1371–1394, 2008.

Joao Carvalho, Dorothea Koert, Marek Daniv, and Jan Peters. Residual robot learning for object-centric probabilistic movement primitives, 2022.

Todor Davchev, Kevin Sebastian Luck, Michael Burke, Franziska Meier, Stefan Schaal, and Subramanian Ramamoorthy. Residual learning from demonstration: Adapting DMPs for contact-rich manipulation. *IEEE Robotics and Automation Letters*, 2022.

M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, January 2015.

Oliver Groth, Chia-Man Hung, Andrea Vedaldi, and Ingmar Posner. Goal-conditioned end-to-end visuomotor control for versatile skill primitives. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1319–1325, 2021.

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*, 2018.

Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013.

Edward Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

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, 2018.

Lydia Kavraki and Jean-Claude Latombe. Randomized preprocessing of configuration for fast path planning. In *IEEE International Conference on Robotics and Automation*, 1994.

Mohamed Khodeir, Ben Agro, and Florian Shkurti. Learning to search in task and motion planning with streams, 2021.

Kenneth Kimble, Joseph Falco, Elena Messina, Karl Van Wyk, Yu Sun, Mizuho Shibata, Wataru Uemura, and Yasuyoshi Yokokohji. Benchmarking protocols for evaluating small parts robotic assembly systems. (5), 2020.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023.

Cheng-Yu Kuo, Andreas Schaarschmidt, Yunduan Cui, Tamim Asfour, and Takamitsu Matsubara. Uncertainty-aware contact-safe model-based reinforcement learning. *IEEE Robotics and Automation Letters*, 6(2):3918–3925, 2021.

Michael Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *arXiv preprint arXiv:2004.14990*, 2020a.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *International Conference on Machine Learning, Vienna, Austria, PMLR 119*, 2020b.

Steven M. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical Report TR 98-11, Computer Science Department, Iowa State University, 1998.

Minh-Tri Le, Chih-Hung G. Li, Shu-Mei Guo, and Jenn-Jier James Lien. Embedded-based object matching and robot arm control. In *IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1296–1301, 2019.

Michelle A. Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks, 2018.

Michelle A Lee, Carlos Florensa, Jonathan Tremblay, Nathan Ratliff, Animesh Garg, Fabio Ramos, and Dieter Fox. Guided uncertainty-aware policy optimization: Combining learning and model-based strategies for sample-efficient policy learning. *IEEE International Conference on Robotics and Automation*, 2020.

Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. In *International Conference on Learning Representations*, 2020.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

Jianlan Luo, Oleg Sushkov, Rugile Pevceviciute, Wenzhao Lian, Chang Su, Mel Vecerik, Ning Ye, Stefan Schaal, and Jon Scholz. Robust multi-modal policies for industrial assembly via reinforcement learning and demonstrations: A large-scale study. *arXiv preprint arXiv:2103.11512*, 2021.

Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP (1)*, pages 331–340. INSTICC Press, 2009.

Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *IEEE international conference on robotics and automation (ICRA)*, pages 6292–6299, 2018.

Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3 (6):233–242, 1999. ISSN 1364-6613.

Oren Spector and Dotan Di Castro. Insertionnet – a scalable solution for insertion, 2021.

J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013.

Mel Vecerik, Todd Hester, Jonathan Scholz, Fumin Wang, Olivier Pietquin, Bilal Piot, Nicolas Heess, Thomas Rothörl, Thomas Lampe, and Martin Riedmiller. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards, 2017.

Mel Vecerik, Oleg Sushkov, David Barker, Thomas Rothörl, Todd Hester, and Jon Scholz. A practical approach to insertion with variable socket position using deep reinforcement learning, 2018.

Bowen Wen, Wenzhao Lian, Kostas Bekris, and Stefan Schaal. You only demonstrate once: Category-level manipulation from single visual demonstration. 2022.

Yizhe Wu, Oiwi Parker Jones, Martin Engelcke, and Ingmar Posner. Apex: Unsupervised, object-centric scene segmentation and tracking for robot manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3375–3382. IEEE, 2021.

Christopher Xie, Yu Xiang, Arsalan Mousavian, and Dieter Fox. Unseen object instance segmentation for robotic environments, 2021.

Jun Yamada, Youngwoon Lee, Gautam Salhotra, Karl Pertsch, Max Pflueger, Gaurav S. Sukhatme, Joseph J. Lim, and Peter Englert. Motion planner augmented reinforcement learning for obstructed environments. In *Conference on Robot Learning*, 2020.

Albert Zhan, Philip Zhao, Lerrel Pinto, Pieter Abbeel, and Michael Laskin. A framework for efficient robotic manipulation. *arXiv:2012.07975*, 2020.

Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *IEEE International Conference on Robotics and Automation*, pages 5628–5635, 2018.