2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Adapting Skills to Novel Grasps: A Self-Supervised Approach

Georgios Papagiannis[1], Kamil Dreczkowski[1], Vitalis Vosylius[1] and Edward Johns[1]

Abstract— **In this paper, we study the problem of adapting manipulation trajectories involving grasped objects (e.g. tools) defined for a *single* grasp pose to *novel* grasp poses. A common approach to address this is to define a new trajectory for each possible grasp explicitly, but this is highly inefficient. Instead, we propose a method to adapt such trajectories directly while only requiring a period of self-supervised data collection, during which a camera observes the robot's end-effector moving with the object rigidly grasped. Importantly, our method requires no prior knowledge of the grasped object (such as a 3D CAD model), it can work with RGB images, depth images, or both, and it requires no camera calibration. Through a series of real-world experiments involving 1360 evaluations, we find that self-supervised RGB data consistently outperforms alternatives that rely on depth images including several state-of-the-art pose estimation methods. Compared to the best-performing baseline, our method results in an average of 28.5% higher success rate when adapting manipulation trajectories to novel grasps on several everyday tasks. The appendix accompanying the paper and videos of the experiments are available on our webpage at www.robot-learning.uk/adapting-skills.**

I. Introduction

Consider a robot that has acquired a skill involving a grasped object, such as using a hammer to hammer in a nail, as shown in Figure 1 (a). Such a skill can be defined with various methods, such as imitation learning [1], and comprises a trajectory of end-effector (EEF) poses, and as a result, a trajectory of poses followed by the grasped object. For that skill to be widely applicable, it must generalise to the novel grasp poses that may occur at skill deployment (i.e. to different poses of the hammer within the robot's gripper). However, generalising a skill to novel grasp poses typically requires manually defining a new trajectory for each grasp pose (e.g. by providing multiple grasp-specific demonstrations), which is highly laborious. In this work, we address this problem and develop a method that enables a robot to autonomously adapt a skill's trajectory defined for an object grasped at a single grasp pose, to any novel grasp.

Past work has addressed this problem mainly by regrasping the object at a pose that aligns with the skill's requirements [2, 3, 4]. However, the majority of these methods assume prior knowledge about the object's 3D CAD model [2, 3] which is usually unavailable in unstructured environments, and methods that bypass this requirement rely on depth images that can cause failures due to missing depth data [4]. Moreover, regrasping methods often rely on a sequence of pick and place operations making them slow to deploy [4].

[1]The Robot Learning Lab at Imperial College London. Email to: g.papagiannis21@imperial.ac.uk

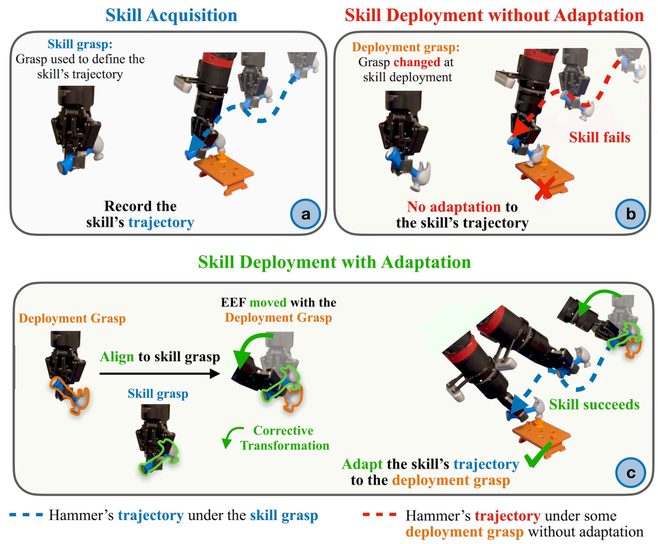

Fig. 1: *Skill Acquisition:* (a) With the hammer grasped at the skill grasp, the skill's trajectory (defined e.g. via a human demonstration) specifies how to hammer the nail. *Skill Deployment without Adaptation:* (b) If the hammer is grasped differently to the skill grasp, executing the skill's trajectory leads to task failure. *Skill Deployment with Adaptation:* (c) A corrective transformation is applied to the skill's EEF trajectory such that (without changing the grasp pose) the hammer under the deployment grasp follows the same trajectory it followed under the skill grasp, to successfully complete the task.

Instead, we are interested in methods that can adapt skills immediately, without regrasping or any prior object knowledge. To adapt a skill's trajectory to some grasp without regrasping, a potential solution is to determine a *corrective transformation* that changes the EEF's trajectory during skill deployment, such that the grasped object follows the same trajectory it followed under the grasp pose that was used to define the skill, as shown in Figure 1 (a) and (c). This can be achieved with the use of existing pose estimation methods, to first estimate the relative pose of the grasped object between the skill acquisition and deployment phases, using RGB or depth images of that object captured from a camera. Then, by using the estimated relative pose, the corrective transformation can be obtained trivially using knowledge of the camera's extrinsics [5]. Given images of the skill and deployment grasps, the required pose estimation could be achieved by establishing correspondences directly [6, 7], though learned descriptors [8, 9, 5] or canonical object representations [10], or by explicitly estimating an object's pose [11, 12, 13, 14, 15]. However, similarly to regrasping approaches, the majority of these methods rely on prior knowledge of the object's 3D CAD model [10, 12] or semantic category-specific training data [5, 10, 13], which

979-8-3503-7769-9/24/$31.00 ©2024 IEEE

may not be readily available. And methods that assume no prior object knowledge when learning descriptors [9, 8, 11] or estimating an object's pose [11, 14, 15, 7] still rely on depth images, which can be problematic due to noisy or partially missing depth data [16]. Finally, as these methods rely on estimating the pose of the grasped object, to determine the corrective transformation they require precise knowledge of a camera's extrinsic parameters (relative to the EEF), which can negatively affect performance due to challenges with camera calibration [17].

Our contributions. Motivated by the above challenges, this paper develops a method to address them by **bypassing the need to explicitly estimate the pose of a grasped object**. Instead, it **directly determines the corrective transformation** to adapt a skill trajectory defined for a single grasp pose to any novel grasp. As a result, our method: (1) **assumes no prior object knowledge, such a 3D CAD model**, (2) **can operate with only RGB images if necessary**, (3) **is robust to noisy or partially missing depth data if using depth images**, and (4) **does not require any camera calibration**.

Our method involves the robot collecting images (RGB or depth) of the grasped object by moving its EEF in front of a single external camera, in a self-supervised manner. As we will later show, this allows us to derive a framework that leverages these images to train a neural network that directly obtains the corrective transformation at skill deployment. As a result, with a few minutes of self-supervised data collection added to a standard pipeline that equips robots with skills, we can now adapt trajectories defined for a single grasp pose to different possible grasps.

We demonstrate the significance of our method through a series of real-world experiments, which quantify its accuracy and ability to adapt skill trajectories to novel grasp poses. Our experiments include adapting manually scripted trajectories for precise peg-in-hole insertion, as well as trajectories obtained with imitation learning for 7 everyday tasks, such as hammering in a nail, aligning a wrench with a nut, and inserting bread into a toaster. By evaluating 3 variants of our method and 5 depth-based pose estimation baselines we conclude that self-supervised RGB data results in the most accurate estimation of the corrective transformation. Specifically, we perform a total of 1360 real-world evaluations which show that, compared to the best-performing baseline, our method yields an average of **28.5%** higher success rate when adapting skill trajectories to novel grasps.

II. Problem Formulation

Notations. We define the following frames: $\{W\}$ which corresponds to the world frame, $\{E\}$ which corresponds to the end-effector's (EEF) frame and $\{O\}$ which is the frame of the grasped object. A transformation $\mathbf{T}_{WE} \in SE(3)$ defines the pose of frame E relative to frame W. Also, we refer to the transformation $\mathbf{T}_{EO}$ as a **grasp pose**. We denote the different relative transformations between the same two frames using superscript notation. For example, ${}^{\mathrm{P}}\mathbf{T}_{EO}$ and ${}^{\mathrm{Q}}\mathbf{T}_{EO}$ denote two different grasp pose instantiations of the same object (for examples see Figure 2), ${}^{\mathrm{M}}\mathbf{T}_{WE}$ and ${}^{\mathrm{K}}\mathbf{T}_{WE}$ denote two different EEF poses expressed in the world frame, and so on. Additionally, we define the transformation $\mathbf{T}_{EE'}$ expressed in frame $\{E\}$, to denote **displacement**, that is the EEF moves by $\mathbf{T}_{EE'}$, from its current pose $\mathbf{T}_{WE}$ to the pose $\mathbf{T}_{WE}\mathbf{T}_{EE'}$. We distinguish across different instantiations of EEF displacements using superscript notation, e.g. ${}^{\mathrm{X}}\mathbf{T}_{EE'}$. Finally, we refer to the EEF displacement that **aligns** an object grasped at pose ${}^{\mathrm{P}}\mathbf{T}_{EO}$ with another grasp pose ${}^{\mathrm{Q}}\mathbf{T}_{EO}$ as the transformation ${}^{\mathrm{Z}}\mathbf{T}_{EE'}$, that moves the grasped object under ${}^{\mathrm{P}}\mathbf{T}_{EO}$ to match the pose of ${}^{\mathrm{Q}}\mathbf{T}_{EO}$ in the world frame; that is, it satisfies: $\mathbf{T}_{WE}{}^{\mathrm{Z}}\mathbf{T}_{EE'}{}^{\mathrm{P}}\mathbf{T}_{EO} = \mathbf{T}_{WE}{}^{\mathrm{Q}}\mathbf{T}_{EO}$ for any $\mathbf{T}_{WE}$.

Motivation. Consider a skill that enables the robot to manipulate an object (e.g. a hammer) grasped at a pose ${}^{\mathrm{S}}\mathbf{T}_{EO}$ to complete some task (e.g., to hammer a nail). We refer to ${}^{\mathrm{S}}\mathbf{T}_{EO}$ as the **skill grasp** and it denotes the grasp pose used to define the skill's trajectory (see Figure 1 (a)). A skill's trajectory comprises a sequence of H EEF displacements $\{{}^{t}\mathbf{T}_{EE'}\}_{t=1}^{H}$ that make the grasped object track a sequence of H poses $\{{}^{t}\mathbf{T}_{EE'}{}^{\mathrm{S}}\mathbf{T}_{EO}\}_{t=1}^{H}$ relative to some initial EEF pose $\mathbf{T}_{WE}$. Examples of such trajectories tracked by a grasped hammer can be seen in Figure 1.

Now, consider a scenario during deployment of the skill, where the object is grasped differently to the skill grasp, at the grasp pose ${}^{\mathrm{D}}\mathbf{T}_{EO}$, referred to as the **deployment grasp** (see Figure 1 (b) and (c)). This can commonly occur due to the robot autonomously grasping the object (e.g., with GraspNet [18]), due to external perturbations on the grasped object or due to a human handing the object to the robot.

Simply executing the skill's trajectory with the deployment grasp will likely result in task failure, as the object will follow a different trajectory to that of the skill grasp, as shown in Figure 1 (b). Specifically, as ${}^{\mathrm{D}}\mathbf{T}_{EO} \neq {}^{\mathrm{S}}\mathbf{T}_{EO}$, and hence the trajectory $\{{}^{t}\mathbf{T}_{EE'}{}^{\mathrm{D}}\mathbf{T}_{EO}\}_{t=1}^{H} \neq \{{}^{t}\mathbf{T}_{EE'}{}^{\mathrm{S}}\mathbf{T}_{EO}\}_{t=1}^{H}$ the skill will no longer be suitable to complete its designated task under the deployment grasp (Figure 1 (b)). Motivated by this, we are interested in determining a **corrective transformation** ${}^{\mathrm{C}}\mathbf{T}_{EE'}$ that changes the EEF's trajectory during skill deployment such that the object follows the same trajectory it followed under the skill grasp (as shown in Figure 1 (a) and (c)). That is ${}^{\mathrm{C}}\mathbf{T}_{EE'}$ aligns ${}^{\mathrm{D}}\mathbf{T}_{EO}$ to ${}^{\mathrm{S}}\mathbf{T}_{EO}$ for any EEF pose $\mathbf{T}_{WE}$:

$$\begin{aligned} \mathbf{T}_{WE}{}^{\mathrm{C}}\mathbf{T}_{EE'}{}^{\mathrm{D}}\mathbf{T}_{EO} = \mathbf{T}_{WE}{}^{\mathrm{S}}\mathbf{T}_{EO} \Rightarrow \\ {}^{\mathrm{C}}\mathbf{T}_{EE'} = {}^{\mathrm{S}}\mathbf{T}_{EO}{}^{\mathrm{D}}\mathbf{T}_{EO}^{-1} \,. \end{aligned} \tag{1}$$

With the corrective transformation, we can adapt the skill's trajectory during skill deployment such that $\{{}^{t}\mathbf{T}_{EE'}{}^{\mathrm{C}}\mathbf{T}_{EE'}{}^{\mathrm{D}}\mathbf{T}_{EO}\}_{t=1}^{H} = \{{}^{t}\mathbf{T}_{EE'}{}^{\mathrm{S}}\mathbf{T}_{EO}\}_{t=1}^{H}$. Figure 1 (c) demonstrates the adapted sequence of poses followed by a hammer after applying the corrective transformation to the EEF's trajectory of Figure 1 (b). We note that adapting a skill using the corrective transformation *only* changes the EEF's trajectory; unlike regrasping methods, it does not change the grasp pose of the object in the robot's gripper.

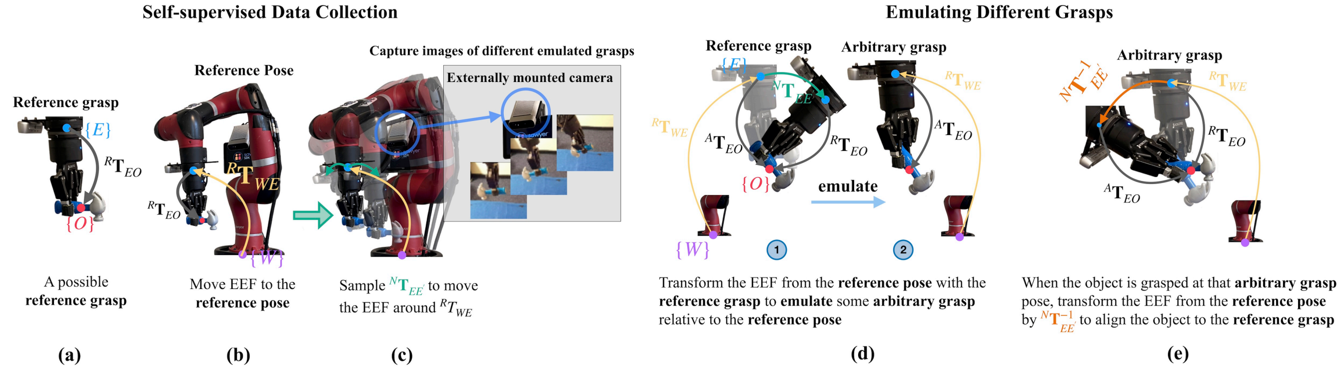

Fig. 2: *Self-supervised Data Collection:* (a) An example of a possible reference grasp. (b) The EEF at a potential reference pose ${}^{\mathrm{R}}\mathbf{T}_{WE}$ in front of the external camera. The object is at the reference grasp. (c) With the object *rigidly* grasped at the reference grasp, we sample and move the EEF to random poses ${}^{\mathrm{N}}\mathbf{T}_{EE'}$ relative to the reference pose to collect image-transformation pairs in a self-supervised manner that emulate different grasps. *Emulating Different Grasps:* (d.1) By transforming the EEF and the object at the reference grasp by ${}^{\mathrm{N}}\mathbf{T}_{EE'}$, we emulate an arbitrary grasp with some grasp pose ${}^{\mathrm{A}}\mathbf{T}_{EO}$ relative to the reference pose as it is shown in (d.2). (e) Then, if the object is grasped at that arbitrary grasp pose ${}^{\mathrm{A}}\mathbf{T}_{EO}$ emulated by ${}^{\mathrm{N}}\mathbf{T}_{EE'}$, moving the EEF to ${}^{\mathrm{N}}\mathbf{T}^{-1}_{EE'}$ relative to the reference pose aligns the object to the reference grasp.

III. Method

In this section, we present a method to determine the corrective transformation between *any* pair of skill and deployment grasps of an object by leveraging images collected in a self-supervised manner in the real-world. This process requires no prior object knowledge or human time, no camera calibration, and it can work with either RGB or depth modalities or both.

The process begins with a user handing the object to the robot's gripper at some arbitrary grasp pose which we refer to as the **reference grasp**. Then, the robot moves to different poses in front of an external camera in a self-supervised manner to emulate different possible grasps. By capturing images from the external camera and leveraging the robot's forward kinematics we train a vision-based alignment network that predicts an EEF displacement that can align any grasp to the reference grasp. As we later show, this allows us to obtain the corrective transformation between any pair of skill and deployment grasps by first aligning them to the reference grasp. This results in a task agnostic method where we can collect data for a grasped object *once* and leverage that data to adapt skills across *any* task for which the object is used and for *any* skill or deployment grasps.

An appendix accompanying our method can be found on our webpage: www.robot-learning.uk/adapting-skills.

A. Aligning grasps to a reference grasp

First, we define the **reference grasp** ${}^{\mathrm{R}}\mathbf{T}_{EO}$. ${}^{\mathrm{R}}\mathbf{T}_{EO}$ can be *any* grasp pose, including that of the skill grasp. For generality, we assume that they are different, and in our experiments to define a reference grasp the user simply places the object in the EEF in a natural-looking pose for that object. Figure 2 (a) shows an example of a possible reference grasp for a hammer. Note that in practice as we do not assume any prior object knowledge or access to a 3D CAD model we do not have access to the numerical value of the pose ${}^{\mathrm{R}}\mathbf{T}_{EO}$. Our initial goal is to determine how to align an object grasped at any possible grasp pose, i.e., any skill or deployment grasp, to the reference grasp, in the absence of any human intervention or prior object knowledge, and without requiring depth images or camera calibration, all of which are hard to achieve with existing pose estimation methods.

Self-supervised data collection. Towards this end, we seek a way to *emulate* the appearance of different possible grasps in the robot's EEF autonomously, in a self-supervised manner. To achieve this, we begin by moving the EEF to a **reference pose** ${}^{\mathrm{R}}\mathbf{T}_{WE}$ with the object grasped at the reference grasp, as shown in Figure 2 (b). The reference pose can be arbitrary as long as the grasped object is clearly visible to the camera. Then, from ${}^{\mathrm{R}}\mathbf{T}_{WE}$, we sample and move the robot to random poses ${}^{\mathrm{N}}\mathbf{T}_{EE'}$ relative to the reference pose. Throughout this process we do not manually move the object in the gripper, instead, the object remains *rigidly* grasped at the reference grasp, requiring no human intervention. At every EEF pose ${}^{\mathrm{R}}\mathbf{T}_{WE}{}^{\mathrm{N}}\mathbf{T}_{EE'}$ we capture an image of the grasped object, I, and record the inverse transformation ${}^{\mathrm{N}}\mathbf{T}^{-1}_{EE'}$, as can be seen in Figure 2 (c) and Figure I.1 in the appendix. This way we record a dataset of M image-transformation pairs $\mathcal{D} := \{(I, {}^{\mathrm{N}}\mathbf{T}^{-1}_{EE'})_i\}_{i=1}^{M}$ in a self-supervised manner.

Every random EEF displacement ${}^{\mathrm{N}}\mathbf{T}_{EE'}$ emulates a different, arbitrary grasp with some grasp pose ${}^{\mathrm{A}}\mathbf{T}_{EO}$ (whose numerical value is unknown) *relative to the reference pose*, as shown in Figure 2 (d.1) and (d.2). As a result, every sample $(I, {}^{\mathrm{N}}\mathbf{T}^{-1}_{EE'})_i$ in our dataset $\mathcal{D}$ contains an image I that depicts how the grasped object would appear to the camera if the object was grasped at that arbitrary grasp pose ${}^{\mathrm{A}}\mathbf{T}_{EO}$ *and the EEF was at the reference pose*, as shown in Figure 2 (d.2). And every ${}^{\mathrm{N}}\mathbf{T}^{-1}_{EE'}$ corresponds to the transformation that we need to apply to the robot's EEF at the reference pose to align the object at that arbitrary grasp pose ${}^{\mathrm{A}}\mathbf{T}_{EO}$ to the reference grasp, as shown in Figure 2 (e). This is true since ${}^{\mathrm{R}}\mathbf{T}_{WE}{}^{\mathrm{N}}\mathbf{T}_{EE'}{}^{\mathrm{R}}\mathbf{T}_{EO} = {}^{\mathrm{R}}\mathbf{T}_{WE}{}^{\mathrm{A}}\mathbf{T}_{EO}$ and as a result ${}^{\mathrm{R}}\mathbf{T}_{WE}{}^{\mathrm{N}}\mathbf{T}^{-1}_{EE'}{}^{\mathrm{A}}\mathbf{T}_{EO} = {}^{\mathrm{R}}\mathbf{T}_{WE}{}^{\mathrm{R}}\mathbf{T}_{EO}$.

For pseudocode detailing our data collection procedure

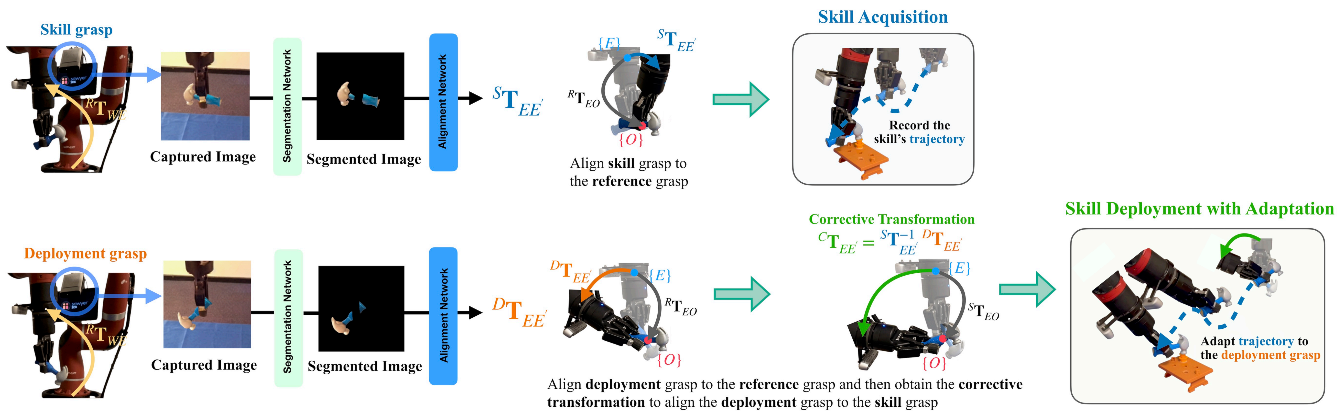

Fig. 3: *Top row:* Given a skill grasp, the EEF moves to the reference pose. Using our trained alignment network we obtain the transformation that aligns the skill grasp to the reference grasp. Then, a method suitable for equipping robots with skills is used to define the skill's trajectory. *Bottom row:* During skill deployment for *any* deployment grasp, we first obtain the transformation that aligns the deployment grasp to the reference grasp using our alignment network. This allows us to compute the corrective transformation which we use to adapt the skill's trajectory to the given deployment grasp.

please see Algorithm 1 in the appendix.

B. Alignment Network

After collecting our dataset $\mathcal{D}$ we train an **alignment network** which is a function parameterised by θ, $f_\theta : \mathbb{R}^{H\times W\times C} \rightarrow SE(3)$ using supervised learning to predict poses ${}^{\mathrm{N}}\mathbf{T}^{-1}_{EE'}$ given camera images (H: height, W: width of the image and $C=3$ for RGB and $C=1$ for depth).

In practice, when the robot grasps the object at a pose different from the reference grasp, the gripper's fingers will occlude parts of the object differently to the occlusions captured in images in $\mathcal{D}$. Hence, we need to ensure that the alignment network's predictions are robust to any occlusion caused by the gripper when grasping the object. To achieve this, before training f_θ, we segment the EEF and background from each image I in $\mathcal{D}$ using a pre-trained optical flow network [19] which we deploy similarly to [20] and perform additional data augmentations. This way, we ensure that our alignment network's predictions are *object-centric* and robust to any object occlusions caused by the gripper or previously unseen image variations. For more implementation details we refer the reader to the appendix I.A-I.C.

At test time, given a grasp, to obtain the transformation ${}^{\mathrm{N}}\mathbf{T}^{-1}_{EE'}$, we first move the EEF to the reference pose ${}^{\mathrm{R}}\mathbf{T}_{WE}$. Then, we capture a live image and segment everything but the grasped object. The segmented image is then passed through f_θ to obtain ${}^{\mathrm{N}}\mathbf{T}^{-1}_{EE'}$. In practice, we obtain ${}^{\mathrm{N}}\mathbf{T}^{-1}_{EE'}$ in a visual servoing (VS) manner. During the VS process, we leverage our robot's redundant DOFs and motion-planning to ensure that we avoid self-collisions and joint limits. We refer the reader to the appendix I.F for a derivation of the VS process using our alignment network's predictions. Finally, as the external camera is rigidly mounted to the robot during data collection and deployment of f_θ, our method is also *independent* of camera calibration.

C. Corrective Transformation

Now that we have a way to align any grasp to the reference grasp, consider a skill grasp ${}^{\mathrm{S}}\mathbf{T}_{EO}$. First given the skill grasp we deploy our alignment network f_θ to obtain the transformation that aligns the skill grasp to the reference grasp. For clarity, we denote that transformation as ${}^{\mathrm{S}}\mathbf{T}_{EE'}$ (instead of ${}^{\mathrm{N}}\mathbf{T}^{-1}_{EE'}$). Then, a method suitable for equipping robots with skills is used to define the skill's trajectory (e.g., an imitation learning method like [1, 21]), as shown in the top row of Figure 3. Then, at skill deployment, given *any* novel deployment grasp, we obtain the transformation that aligns that grasp to the reference grasp in an identical manner using the alignment network. For clarity, we denote that transformation as ${}^{\mathrm{D}}\mathbf{T}_{EE'}$. Given ${}^{\mathrm{S}}\mathbf{T}_{EE'}$ and ${}^{\mathrm{D}}\mathbf{T}_{EE'}$, we can calculate the corrective transformation ${}^{\mathrm{C}}\mathbf{T}_{EE'}$ trivially, that is ${}^{\mathrm{C}}\mathbf{T}_{EE'} = {}^{\mathrm{S}}\mathbf{T}^{-1}_{EE'}\,{}^{\mathrm{D}}\mathbf{T}_{EE'}$. Then, given the corrective transformation, we can immediately adapt a skill's trajectory, as shown in the bottom row of Figure 3 and discussed in Section II.

To see why we can obtain the corrective transformation this way, note that ${}^{\mathrm{S}}\mathbf{T}_{EE'}$ and ${}^{\mathrm{D}}\mathbf{T}_{EE'}$ align the skill grasp ${}^{\mathrm{S}}\mathbf{T}_{EO}$ and deployment grasp ${}^{\mathrm{D}}\mathbf{T}_{EO}$ to the reference grasp ${}^{\mathrm{R}}\mathbf{T}_{EO}$ respectively, that is: ${}^{\mathrm{S}}\mathbf{T}_{EE'}\,{}^{\mathrm{S}}\mathbf{T}_{EO} = {}^{\mathrm{R}}\mathbf{T}_{EO} = {}^{\mathrm{D}}\mathbf{T}_{EE'}\,{}^{\mathrm{D}}\mathbf{T}_{EO}$. Hence, it follows that: ${}^{\mathrm{S}}\mathbf{T}_{EO}\,{}^{\mathrm{D}}\mathbf{T}^{-1}_{EO} = {}^{\mathrm{S}}\mathbf{T}^{-1}_{EE'}\,{}^{\mathrm{D}}\mathbf{T}_{EE'}$ and as a result from Equation 1 we can see that ${}^{\mathrm{C}}\mathbf{T}_{EE'} = {}^{\mathrm{S}}\mathbf{T}^{-1}_{EE'}\,{}^{\mathrm{D}}\mathbf{T}_{EE'}$. Note that the corrective transformation is not dependent to the reference pose or any EEF pose relative to the world frame. This allows us to adapt the skill's trajectory across the whole task space of the robot.

For pseudocode detailing the deployment of our method please see Algorithm 3 in the appendix.

Finally, we note that the alignment network is agnostic to the skill's trajectory and its designated task. This is very important: once the alignment network is trained for a particular grasped object, we can re-use that same alignment network for *any* task with that same object, without further training. A video demonstrating this ability can be found at the bottom

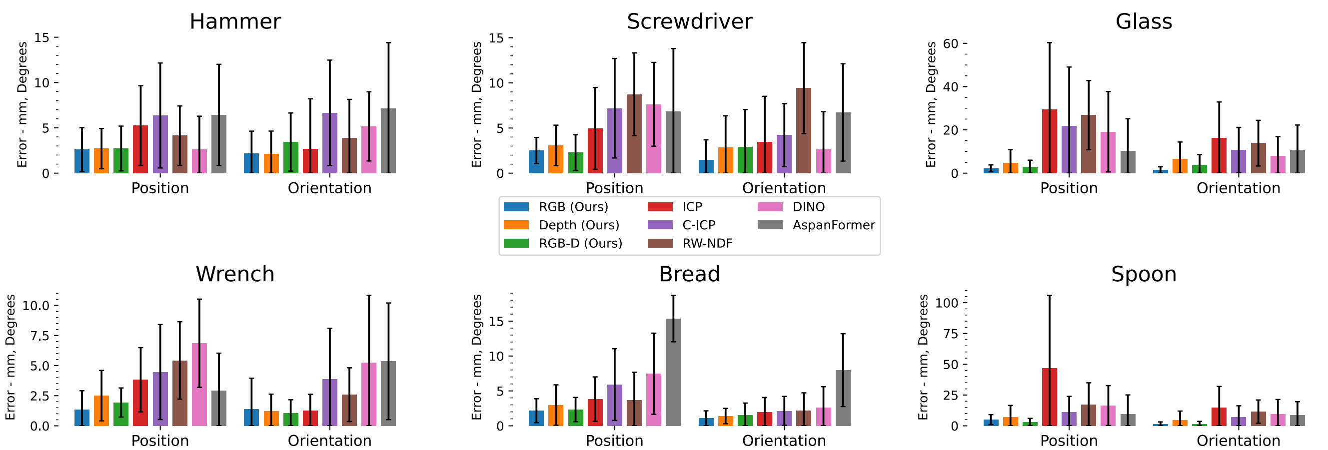

Fig. 4: Mean and standard deviation error in computing the corrective transformation between grasps averaged across all DoFs for the position and orientation for the 6 everyday objects (lower is better).

of our webpage at www.robot-learning.uk/adapting-skills.

IV. Experiments

We evaluate our method by performing three sets of real-world experiments totaling 1360 real-world evaluations. Through these experiments, we answer the following questions: 1) How accurate are the corrective transformations obtained by our method? 2) Can our method adapt skills to novel deployment grasps for (a) precise tasks and (b) tasks learned with imitation learning? 3) What is the best modality to determine the corrective transformation in the real-world; RGB, depth or both combined (RGB-D)? Videos of our real-world experiments can be found on our webpage at www.robot-learning.uk/adapting-skills.

Implementation details. For our experiments, we use a 7 DoF Rethink Sawyer Robot to which we mount a Microsoft Azure Kinect camera. To determine the reference pose ${}^{\mathrm{R}}\mathbf{T}_{WE}$, we empirically evaluate the quality of depth images for the objects used for evaluation at various EEF poses and select the best one. We found this to be crucial to obtain good performance for the baselines, as the quality of the depth images varied significantly based on an object's pose, unlike our method that is robust to this. We allow an average of approximately 5 minutes of real-world data collection, during which we sample random poses around ${}^{\mathrm{R}}\mathbf{T}_{WE}$ in the range of 30cm for each of the DoFs relating to position, and 60° for each of the DoFs relating to orientation. This range is not limiting and can be trivially adjusted to any value to accommodate any task requirements; we found that this range was sufficient for our tasks. Preprocessing the collected data and training takes approximately 15 minutes on an NVIDIA GeForce RTX 3080 Ti. We refer the reader to appendix I.C, I.D and I.E for further implementation details, a description of our network architectures and a discussion on the effect of data collection time on our method's performance.

A. How accurate are the corrective transformations obtained by our method?

In this experiment, we quantify the accuracy of our method in determining the corrective transformation between pairs of object grasps. To perform our evaluation, we use 6 objects for the everyday tasks shown in Figure 5. That is, a plastic **hammer**, a plastic **screwdriver**, a plastic **bread**, a metallic **spoon**, a plastic **wrench**, and a semi-transparent wine **glass**.

Evaluation procedure. As we do not have access to the true pose of the objects, we evaluate our method using the robot's forward kinematics. We refer the reader to appendix II.A for a detailed description of our evaluation procedure. For each of the six objects, we randomly set 4 skill grasps and for each skill grasp, we evaluate the corrective transformation for 5 random deployment grasps leading to 20 evaluations per object. For each evaluation, we allow our method 5 seconds of visual servoing to align each grasp to the reference grasp.

Baselines. As we have no information about the objects' CAD models, we compare all variants of our method (RGB, Depth, RGB-D) on the same objects against 5 baselines that can be deployed without requiring any prior object knowledge: 1) ICP, 2) Color-ICP (C-ICP), 3) a real-world variant of neural descriptor fields [5] (RW-NDF) that we implement, and two correspondence based methods that leverage 4) DINO ViT (DINO) [22, 8] and 5) AspanFormer[6].

For ICP and C-ICP we capture a segmented depth image of the grasped object under the skill and deployment grasps and compute the relative pose between them in the camera's frame. Then, by leveraging the camera's extrinsics we obtain the corrective transformation. Additionally, we note that we tried scanning the object by moving it in front of the external camera to obtain a more complete 3D object model before using ICP and C-ICP, but performance did not improve and was almost identical. For DINO and AspanFormer we first obtain correspondences between the RGB images of the skill and deployment grasps and obtain their relative pose by leveraging the corresponding depth images and singular value decomposition [23]. Identically to our method's evaluation, we allow each baseline 5 seconds of visual servoing. We refer the reader to the appendix II.B for more details on the baselines implementations.

Results. Figure 4 shows the mean and standard deviation error of the corrective transformation averaged separately across the 20 evaluations for the DoFs relating to position and orientation for each object and method. The quantitative results of Figure 4 can be seen in the appendix Table B1. As shown, the RGB, Depth and RGB-D variants of our method perform better, on average, than the 5 baselines both with respect to position and orientation error. Specifically, the best-performing variant overall from our methods is RGB with a 2.65mm mean position error and 1.50° mean orientation error. In addition to the high accuracy, the results of Figure 4 also confirm that our method is robust to object occlusions caused by the gripper's fingers under different grasps. From the baselines, AspanFormer obtains the lowest mean error in position (8.59mm) and DINO for orientation (5.56°). Hence, RGB obtains a 69.1% increase in position accuracy compared to AspanFormer and a 73.0% increase in orientation accuracy compared to DINO, averaging at least a **71.1**% increase in accuracy when compared to all the baselines.

The performance difference is most profound for the spoon and glass objects (see Figure II.1 in the appendix), where all the baselines struggle to accurately determine the corrective transformation. We attribute this difference to the missing depth data for these objects which is due to their textures; that is the spoon is metallic and the glass semi-transparent. As shown in the appendix Figures II.3 and II.4 the point clouds for these objects have very low quality. On the other hand, our depth-based variant shows robustness to missing depth data as it was trained *directly* on the real-world depth images for each object. Nevertheless, it is still less accurate when compared to the RGB and RGB-D variants.

B. Can our method adapt skills to novel deployment grasps for precise tasks?

In this experiment, we evaluate the ability of our method to adapt skill trajectories for precise tasks. To this end, we 3D printed a base with 4 different holes and a peg. The 4 holes have insertion tolerances of $\{2, 4, 8, 12\}$mm on each side of the peg, as shown in Figure 5. In this setting, we assume that the pose of the base is known. This setup may correspond to an industrial assembly setting where the pose of the insertion hole is known, but there is uncertainty on the grasp pose of the object (e.g. because the robot autonomously grasps it). In this setting, manually programming a separate trajectory for every possible grasp is infeasible. Instead, it is significantly more efficient to design a single insertion skill for some skill grasp and deploy our method to adapt the insertion trajectory for each deployment grasp.

Evaluation procedure. First, for each hole, we hand the peg to the robot to define a skill grasp and we manually program a trajectory of poses to insert the peg in each of the 4 holes which we track with a position controller. Then, we randomly change the pose of the peg in the gripper and consider that as a deployment grasp. Given the deployment grasp, we deploy our method to adapt the insertion trajectory as discussed in section II. We evaluate 5 different deployment grasps for each hole totalling 20 evaluations for each method.

Hole Tolerance	RGB	Depth	RGB-D	ICP	DINO
2mm	**80%**	40%	40%	40%	60%
4mm	**100%**	60%	80%	40%	20%
8mm	**100%**	80%	**100%**	**100%**	80%
12mm	**100%**	100%	**100%**	**100%**	**100%**
Average	**95%**	70%	80	70%	65%

TABLE I: Skill adaptation success rate to various deployment grasps for peg-in-hole insertion for 4 hole tolerances.

Baselines. Based on the results of Section IV-A, we compare our method against the best non-learning-based baseline, ICP, and the best learning-based baseline, DINO. As we are interested in success rate, we selected ICP because it outperforms C-ICP in 4 out of the 6 objects, although its average error is higher as it is negatively affected by the "Glass" and "Spoon" objects. Similarly, we selected DINO as it outperforms RW-NDF and AspanFormer in more objects when accounting both for the translation and orientation errors. We evaluate the baselines identically to our method.

Results. Table I shows the success rate results for all methods. All variants of our method, outperform on average both ICP and DINO. Both ICP and DINO successfully adapt most deployment grasps for the holes with tolerances 8mm and 12mm, but fail to adapt the majority of the deployment grasps for the 2mm and 4mm holes. As shown, the RGB variant of our method performs best overall, outperforming ICP and DINO on average by **25**% and **30**% respectively, only failing to adapt one deployment grasp for the hole with the lowest tolerance of 2mm. The RGB-D variant also performs well but fails to adapt 3 deployment grasps for the 2mm tolerance hole. On the other hand, the Depth variant has the lowest performance across our method's variants.

The results suggest that our method can accurately determine the corrective transformation between grasps to adapt skill trajectories even for tasks with low error tolerance. However, our results also indicate that the RGB modality is crucial for high precision. Interestingly, when combining RGB with depth, the performance seems to be better than using only depth but lower compared to using only RGB. This is likely because, during training, our alignment network is optimized to give a certain weight to RGB and a certain weight to the depth modality based on the depth values recorded in the training dataset. However, if the depth images observed during testing do not match those observed during training, likely due to the high level of noise in the depth signal, this can negatively affect the alignment network's predictions.

C. Can our method adapt skills learned with imitation learning to novel deployment grasps?

In this experiment, we are interested in evaluating our method as a modular component added to a skill acquisition pipeline where skill trajectories are obtained using imitation learning. This setup corresponds to a realistic robot learning setup where a user may teach a robot skills using human demonstrations. Specifically, we use the one-shot imitation learning method DOME [1] to equip our robot with skills to solve the 6 everyday tasks shown in Figure 5. DOME

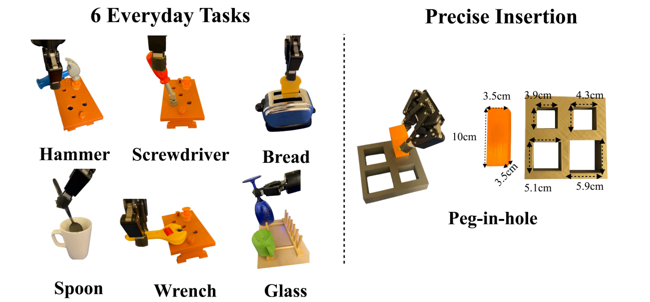

Fig. 5: The 6 everyday objects and tasks and the peg-in-hole task used in the experiments.

is a vision-based imitation learning method whose action space comprises EEF twists. We provide a brief overview of DOME in appendix II.D and refer the reader to [1] for more details.

Evaluation procedure. We teach the robot skills using DOME to solve the following six everyday tasks: A *Hammer* task requiring the robot to knock a plastic nail using a plastic hammer into a receptacle. A *Screwdriver* task requiring the robot to fully insert the tip of the plastic screwdriver into the slit of a screw. This task has very low tolerance, requiring millimetre precision. A *Bread* task requiring the robot to insert a plastic bread into the slit of a toaster. A *Spoon* task requiring the robot to insert the spoon into a mug and stir. A *Wrench* task that requires the robot to insert a plastic nut into the wrench's head. Finally, a *Glass* task where the robot needs to place a wine glass standing upright on a wooden rack. We chose these tasks to represent a variety of challenges and tolerances ranging from several cm of tolerance (*Spoon*) to around 1 mm of tolerance (*Screwdriver*). For each task, first, we hand the relevant object to the robot's EEF to define a skill grasp and provide a demonstration using DOME. Then, we manually sample novel deployment grasps approximately within 10cm and 90° of the skill grasp for all the axes not constrained by the gripper's fingers and deploy DOME along with our method to adapt each skill's twists as described in section II. We perform this process for 10 different grasps for each task and record the success rate of each skill adaptation trial, totaling 60 evaluations per method. For each evaluation, we also randomize the task space configuration and consider a trial successful if the task is completed as in the demonstration. Finally, as in section IV-B, we compare our method against the ICP and DINO baselines.

Results. Table II shows the skill adaptation success rate for all the methods. All variants of our method outperform the baselines in the majority of trajectory adaptation trials, apart from the Screwdriver task where ICP outperforms both the Depth and RGB-D variants and the Bread task where DINO is the best-performing method along with RGB-D. Overall, the best-performing variant of our methods is RGB with an average success rate of 75 %, outperforming on average ICP and DINO by **32**% and **37**% respectively. We observe that all variants of our method can successfully adapt trajectories obtained by DOME to most deployment grasps, especially for the Hammer, Spoon and Glass tasks. The depth and

Tasks	RGB	Depth	RGB-D	ICP	DINO
Hammer	**100%**	**100%**	**100%**	80%	60%
Screwdriver	**30%**	10%	10%	20%	0%
Bread	50%	60%	**70%**	60%	**70%**
Spoon	**100%**	**100%**	**100%**	30%	40%
Wrench	70%	70%	**80%**	50%	20%
Glass	**100%**	80%	80%	20%	40%
Average	**75%**	70%	73%	43%	38%

TABLE II: Skill adaptation success rate to various deployment grasps for imitation learning skills taught using DOME

RGB-D variants failed twice to adapt the trajectory for the Glass task. These failures likely relate to the semi-transparent texture of the glass which yields poor depth quality; an observation also made in section IV-A.

All variants of our method fail almost completely to adapt the trajectory obtained by DOME for the Screwdriver task. We attribute this failure mainly to DOME's error in accurately approaching the screw as our method yielded small inaccuracies in determining the corrective transformation in the order shown in Figure 4. Although small, the errors from both methods can result in failure when accumulated, especially for a high-precision task like the Screwdriver task. Similar reasoning applies to the Bread task, which, however, has a higher tolerance to error compared to the Screwdriver and, subsequently, a higher success rate.

Following these results and the results obtained in Section IV-B we can see that the RGB variant yields on average a **28.5**% higher success rate compared to ICP and **33.5**% when compared to DINO.

D. What is the best modality to determine the corrective transformation between pairs of object grasps in the real-world, RGB, depth or both combined (RGB-D)?

In our experiments, we observed that all modalities perform similarly when trained using our method. Overall, RGB is crucial when dealing with objects for which the depth quality is poor, especially when compared to depth-based registration methods as demonstrated in the previous sections. Further, even for the objects for which the depth quality was high, the performance was on par with that of RGB. In fact, we observed that RGB performed better for the peg-in-hole task. Hence, we can conclude that the RGB modality is the most robust overall. It performs well for all types of objects, causing no performance degradation compared to depth on all the experiments we conducted.

V. Discussion

A. Limitations

We now highlight some important limitations of our method. Firstly, to obtain an alignment network for an object our method needs to collect data for a few minutes. As such, for basic low-tolerance tasks, simple baselines such as ICP may be preferred as they do not require data collection. However, for real-world tasks requiring precision, our method is *necessary* to overcome errors from calibration or missing depth data. And, as our alignment network can be reused across *any* task for an object without further training, if we aim to adapt skills across multiple grasps and tasks the required data collection time becomes comparably negligible.

Secondly, our method assumes that the object under the deployment grasp is grasped such that the gripper's fingers do not obstruct task execution. To address this assumption future work can trivially couple our method with affordance-based grasping approaches, e.g., [24] or imitation learning approaches like DOME [1] that can show to the robot which parts of an object to grasp. Additionally, this assumption is not practically limiting as most deployment grasps can be adapted, and as we showed in our experiments we can adapt skill trajectories over a wide range of grasp poses. Also, future work could investigate extending our method to detect such scenarios and perform regrasping by leveraging the corrective transformation.

Thirdly, our method assumes that for a given deployment grasp executing the adapted trajectory results in the same kinodynamic interaction with the environment as with the skill grasp. In most scenarios, this can be achieved by leveraging our robot's redundant degrees of freedom and motion planning as we do in our experiments. However, for grasps where the adapted trajectory cannot complete a task due to issues relating to the robot's kinematics, future work can couple our method with approaches that determine the initial robot configuration such that skill execution succeeds[25].

Finally, compared to methods such as [9, 5, 8, 11], in the current setup same-category generalization is not possible with our approach. However, these methods assume prior access to category-specific training data which may not be readily available, and rely on depth images which hinder the performance as shown in our experiments. Instead, our method addresses these issues. In future work, we aim to study whether training our method directly on image descriptors extracted from our dataset using pertained vision models, such as DINO ViTs [22], allows our method to generalize to novel objects of the same category.

B. Conclusions

In this work, we proposed an autonomous, self-supervised method that enables the adaptation of skill trajectories defined for a single object grasp pose to any novel grasp pose at skill deployment, without any prior knowledge about the grasped object. Through multiple real-world experiments, we show that our method enables skills acquired through imitation learning, for several everyday tasks, to be adapted to different grasps at deployment without the robot needing to re-learn or fine-tune the skill itself. Importantly, our results demonstrate that RGB data collected in a self-supervised manner is the best modality that obtains the highest skill adaptation performance when compared to depth-based alternatives including several state-of-the-art pose estimation methods.

References

[1] E. Valassakis *et al.*, "Demonstrate once, imitate immediately (dome): Learning visual servoing for one-shot imitation learning," *IROS*, 2022.

[2] W. Wan, H. Igawa, K. Harada, H. Onda, K. Nagata, and N. Yamanobe, "A regrasp planning component for object reorientation," *Auton. Robots*, vol. 43, no. 5, pp. 1101–1115, Jun. 2019.

[3] A. Nguyen *et al.*, "Preparatory object reorientation for task-oriented grasping," in *IROS*, 2016.

[4] S. Cheng, K. Mo, and L. Shao, "Learning to regrasp by learning to place," *CoRR*, vol. abs/2109.08817, 2021. arXiv: 2109.08817.

[5] A. Simeonov *et al.*, "Neural descriptor fields: Se(3)-equivariant object representations for manipulation," *ICRA*, 2022.

[6] H. Chen *et al.*, "Aspanformer: Detector-free image matching with adaptive span transformer," *ECCV*, 2022.

[7] S. Rusinkiewicz *et al.*, "Efficient variants of the icp algorithm," *3rd Intl. Conf. on 3D Digital Imaging and Modeling*, 2001.

[8] S. Amir *et al.*, "Deep vit features as dense visual descriptors," *ECCVW What is Motion For?*, 2022.

[9] P. R. Florence, L. Manuelli, and R. Tedrake, "Dense object nets: Learning dense visual object descriptors by and for robotic manipulation," *arXiv preprint arXiv:1806.08756*, 2018.

[10] B. Wen, W. Lian, K. E. Bekris, and S. Schaal, "You only demonstrate once: Category-level manipulation from single visual demonstration," *ArXiv*, vol. abs/2201.12716, 2022.

[11] W. Goodwin *et al.*, "You only look at one: Category-level object representations for pose estimation from a single example," in *CoRL*, 2023.

[12] X. Deng, Y. Xiang, A. Mousavian, C. Eppner, T. Bretl, and D. Fox, "Self-supervised 6d object pose estimation for robot manipulation," in *ICRA*, 2020.

[13] X. Li, H. Wang, L. Yi, L. Guibas, A. L. Abbott, and S. Song, "Category-level articulated object pose estimation," *CVPR*, 2020.

[14] S. Devgon *et al.*, "Orienting novel 3d objects using self-supervised learning of rotation transforms," *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 2020.

[15] H. Yisheng, W. Yao, F. Haoqiang, C. Qifeng, and S. Jian, "Fs6d: Few-shot 6d pose estimation of novel objects," *CVPR*, 2022.

[16] A. Kadambi, A. Bhandari, and R. Raskar, "3d depth cameras in vision: Benefits and limitations of the hardware," in *Computer Vision and Machine Learning with RGB-D Sensors*. 2014.

[17] E. Valassakis, K. Dreczkowski, and E. Johns, "Learning eye-in-hand calibration from a single image," in *CoRL*, 2021.

[18] H. Fang *et al.*, "Graspnet-1billion: A large-scale benchmark for general object grasping," *2020 CVPR*, pp. 11 441–11 450, 2020.

[19] H. Xu *et al.*, *Unifying flow, stereo and depth estimation*, 2022. arXiv: 2211.05783 [cs.CV].

[20] W. Boerdijk, M. Sundermeyer, M. Durner, and R. Triebel, "Self-supervised object-in-gripper segmentation from robotic motions," in *Conference on Robot Learning*, 2020.

[21] E. Johns, "Coarse-to-fine imitation learning: Robot manipulation from a single demonstration," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[22] M. Caron *et al.*, "Emerging properties in self-supervised vision transformers," *2021 ICCV)*, 2021.

[23] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, pp. 698–700, 1987.

[24] D. Hadjivelichkov *et al.*, "One-Shot Transfer of Affordance Regions? AffCorrs!" In *CoRL*, 2023.

[25] V. Vosylius and E. Johns, "Where to start? collision-free transfer of skills to new environments," in *CoRL*, 2022.

Appendix

For the appendix accompanying our paper and videos demonstrating the method please visit our website: www.robot-learning.uk/adapting-skills.