Feature Selection for Learning Versatile Manipulation Skills based on Observed and Desired Trajectories

Oliver Kroemer, and Gaurav S. Sukhatme

Abstract— **For a manipulation skill to be applicable to a wide range of scenarios, it must generalize between different objects and object configurations. Robots should therefore learn skills that adapt to features describing the objects being manipulated. Most of these object features will however be irrelevant for generalizing the skill and, hence, the robot should select a small set of relevant features for adapting the skill.**

We use a framework for learning versatile manipulation skills that adapt to a sparse set of object features. Skills are initially learned from demonstrations and subsequently improved using reinforcement learning. The robot also learns a meta prior over the features' relevances to guide the feature selection process. In this paper, we explore using either desired trajectories or observed trajectories for selecting the relevant features. The framework was evaluated on placing, tilting, and wiping tasks. The evaluations showed that using the desired trajectories to select the relevant features lead to better skill learning performance.

I. Introduction

In order to generalize manipulations between different scenarios, robots will need to be capable of adapting their manipulation skills to the objects that they are manipulating. For example, the robot may need to adapt a grasping skill to the location of an object's handle, or a placing skill to the size and location of a supporting surface. These variations in the objects and their poses can be represented by a set of *object features*. The robot can then learn versatile manipulation skills that adapt to these object features.

The majority of the object features will not be relevant for generalizing a manipulation skill. Hence, only a sparse set of object features should be selected for adapting the manipulation skill. Rather than relying on a human to define the set of relevant features, the robot should select the features autonomously. The robot can select the features based on example trajectories from different situations. These trajectories may be obtained from human demonstrations, or acquired autonomously through trial and error.

We explore feature selection for manipulation skills using a skill learning framework that incorporates imitation, reinforcement, and transfer learning [1]. An overview of the framework is shown in Fig. 1 and its components are explained in Section III. Motor primitive skills are initially learned from kinesthetic demonstrations. These example trajectories are used as training data for selecting the relevant features using stochastic search variable selection (SSVS) [2]. When the robot encounters a new scenario, it computes the initial desired trajectory according to the current skill. The skill execution is subsequently improved through trial and error using policy search reinforcement learning. The final trajectory is used as an additional training sample for learning the skill and selecting the relevant features. To guide the feature selection process, the robot learns a meta prior for transferring the relevance of features from previous skills. The meta-level prior allows the robot to predict the relevance of the new skill's object features based on *meta features* extracted from the initial demonstrations. These meta features represent characteristics of the object features and how they relate to the skill being learned.

In this paper, we use our skill learning framework [1] to investigate the effects of using either the desired hand trajectories or the hand trajectories observed during the skill executions to select the relevant features. Using the desired trajectory is the more direct approach to selecting the object features for generalizing the desired trajectory. However, the desired trajectory is not bound by the physical constraints of the task. For a compliant robot, the observed trajectory will deviate from the desired trajectory when it encounters a constraint, e.g., when it pushes against an immovable object. The observed trajectory may therefore include additional information regarding the task's constraints, which the robot should use to adapt the skill's desired trajectory. We investigate both of these approaches in this paper.

The two approaches were evaluated on placing and tilting tasks, as described in Section IV. Using the desired trajectories to select the relevant features lead to improved skill learning performance. Using a meta prior to guide the feature selection process also resulted in better performance. In the second experiment, the robot successfully learned wiping skills using a meta prior learned from the skills acquired in the first experiment.

II. Related Work

Our skill learning framework combines imitation, reinforcement, and transfer learning [1]. Recent work in imitation learning and reinforcement learning have been used to learn and execute complicated robot motor skills, e.g., playing ball games [3], [4], [5], [6], scrubbing surfaces [7], [8], opening doors [9], [10], and manipulating objects [11], [12], [13]. The learned skills adapt to the object features of the task, e.g., the goal location or the position of the ball, but the features used to generalize the skill are usually manually preselected. These approaches therefore focus on learning how to adapt to the object features, and not on selecting

This work was funded in part by the NSF under grant CNS-1213128 and the ONR under grant N00014-14-1-0536.

Oliver Kroemer and Gaurav S. Sukhatme are both members of the Robotic Embedded Systems Lab at the University of Southern California, CA, USA {okroemer,gaurav}@usc.edu

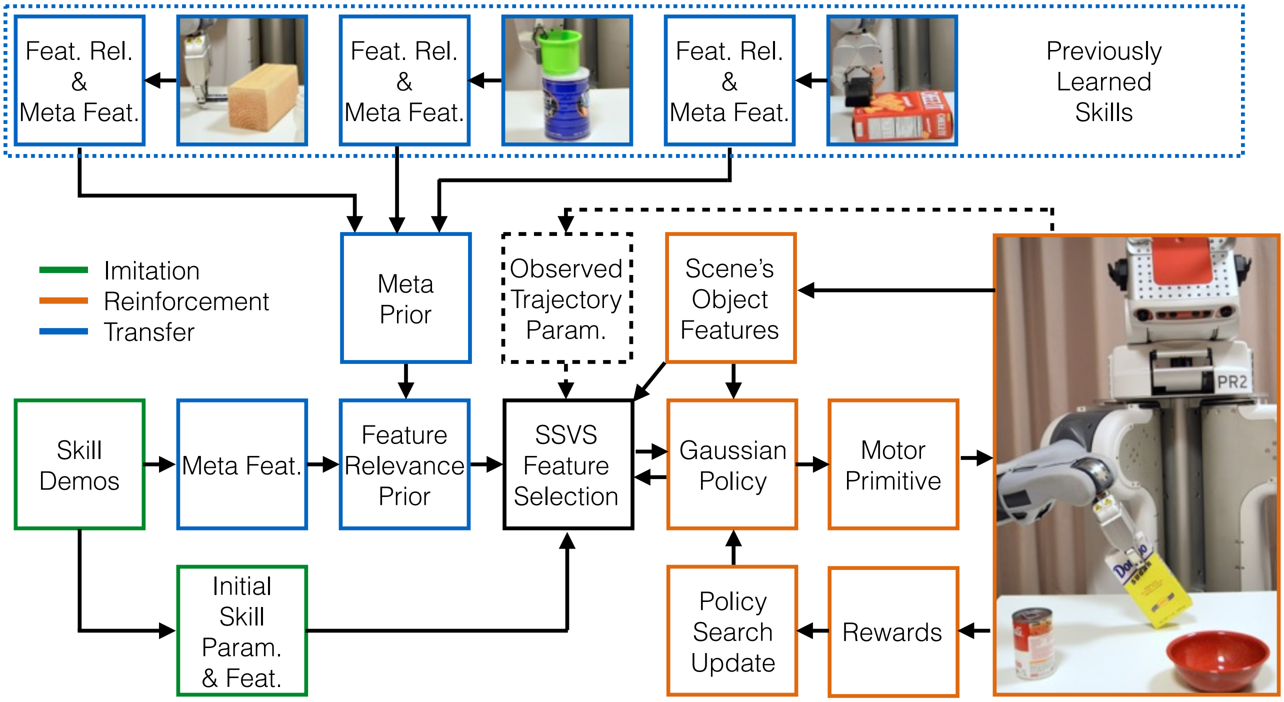


Fig. 1. An overview of the manipulation skill learning framework. The robot learns a mapping from the object features to the motor primitive's parameters. The features and parameterized motor primitives are described in Sections III-A and III-B. The framework uses stochastic search variable selection (SSVS) to select a sparse set of relevant object features for generalizing the skill, as explained in Section III-D. The selection process is guided by a prior on the relevance of the features. This prior is computed from meta features extracted from demonstrations and a meta prior learned from the feature relevances and meta features of previously acquired skills, as explained in Section III-C. The framework incorporates reinforcement learning to optimize the skill for new scenarios and thus acquire more training samples for the feature selection and skill learning process.

the relevant features. Our approach uses a set of predefined rules to automatically generate large feature pools from task demonstrations. The robot subsequently learns to select a subset of relevant features from the pool.

Some frameworks learn features to improve action selection in ambiguous situations [14], [15]. These methods often result in an implicit pose estimation of an object or part that the robot then uses to generalize the skill. The features are usually learned for specific objects and not to generalize between different objects with various shapes and sizes.

Motor skills can be adapted to different situations by using suitable task frames. Potential task frames are often defined relative to objects. The robot can select a task frame from demonstrations based on the variances of the trajectories in each candidate task frame [16], [17]. Task frames can be generalized between objects by detecting object parts with similar shapes and estimating their poses [18], [19], [20].

A related challenge for skill learning is selecting the set of relevant objects for a manipulation [21]. Relevant objects can be extracted from demonstrations using visual cues, such as motionese [22]. The object selection problem is distinct from our feature selection problem, as not all of the features of a relevant object will themselves be relevant.

Several works have explored multi-task and transfer learning in the field of robotics [23], [24], [25]. These approaches often focus on transferring trajectories or controllers between tasks. The tasks are often similar and share the same feature space, e.g., different target locations for reaching may be considered as different tasks. Our tasks have distinct features and the robot learns a meta-level prior for transferring the relevance of the features between skills. Meta features have been used to transfer knowledge about features between tasks in other applications including predicting movie ratings, text classification, and object recognition [26], [27], [28].

III. Skill Learning with Feature Selection

The object features and skill representations are described in Sections III-A and III-B respectively. The robot predicts the relevance of the individual features using a meta prior, as described in Section III-C, and then selects a sparse set of features using stochastic search variable selection, as explained in Section III-D. The robot improves the skill execution for new situations using reinforcement learning, as described in Section III-E.

A. Object Features

To generalize skills between objects, the robot requires a suitable representation for describing the manipulated objects. We use object features to model the objects' shapes, although the set of features could be extended to include other properties such as mass. Manipulations often depend on parts of objects [29], e.g., the blade of a knife, or the opening of a container. Our features therefore describe the relevant parts of the objects rather than the objects as a whole.

To extract a set of object parts, we provide the robot with coarse 3D point cloud models of the objects and

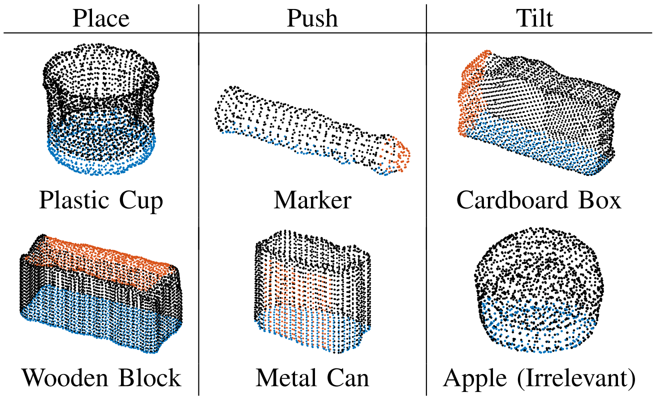


Fig. 2. Examples of object parts (colored points) that were extracted using GrabCut. Each part is described by six features: 3D positions and 3D sizes.

demonstrations of the manipulation task using the objects. The robot estimates the object parts by detecting points that come into close proximity to other objects during the demonstrations. Proximity often indicates an interaction between objects. These points are used as the initial part estimates for segmenting the point cloud into part and non-part regions using GrabCut [30]. Additional details of the part extraction are provided in our previous work [31]. Examples of the extracted parts for the pushing, placing, and tilting tasks used in our experiments are shown in Fig. 2.

Given the object parts, the robot uses a set of predefined rules to generate object features for each of the parts. Our evaluation tasks can be performed using motions aligned with the Cartesian robot frame. We also assume that the objects are prealigned with this task frame at the start of the skill execution. The object features are therefore computed by fitting an axis-aligned bounding box to the point cloud points associated with each part. The 3D x-y-z position of the bounding box's center relative to the robot's hand defines the first three features. The x-y-z lengths of the bounding box's sides define another three features, giving a total of six features per extracted part. The features from all of the parts are concatenated to form a set of M object features $\phi_{ji} \forall j \in \{1, ..., M\}$ that describe the ith scene $\mathcal{S}_i$. The correspondences between the objects and parts across different scenes are given, and the features are concatenated in the same order to create a consistent feature vector. Most of the generated features will not be relevant for generalizing the manipulation skill, and the robot will need to select a sparse set of relevant features.

B. Adaptive Skills

Having extracted a set of object features, the robot now needs to learn manipulation skills that adapt to these features. The skills are represented using dynamic motor primitives (DMPs) [32]. The Cartesian DMPs use a separate linear dynamical system to represent each of the three x-y-z components of the skills' desired trajectory. We focus on learning the translational components of the skills. The shape of the trajectory for each skill component is defined by a set of $K+1$ shape parameters $\tilde{w}_{ik} \forall k \in \{0, ..., K\}$, for scene S_i, which includes a goal offset $\tilde{w}_{i0} = g - y_0$ between the initial state y_0 and the goal state g.

To generalize the manipulation skill, the robot learns a linear mapping from object features to shape parameters

$$\tilde{w}_{ik} = \textstyle\sum_{j=1}^{M} w_{jk} \phi_{ji}, \tag{1}$$

where w_{jk} is the skill parameter that determines how the jth feature affects the kth shape parameter. Even though this linear representation is not as flexible as a nonlinear mapping, it can be used for adapting a wide variety of skills. For example, the robot can learn a pouring skill that adapts to the position and width of a container's opening, or it can learn a cutting skill that scales with the length of the blade.

Given a set of N training examples, each of which includes K shape parameter values $\tilde{w}_{ik}$ and M object feature values ϕ_{ji}, the robot can learn the skill parameters w_{jk} using linear ridge regression. However, most of the extracted object features will not be relevant for generalizing the manipulation skill. We therefore include a feature relevance variable $\gamma_j \in \{0, 1\} \forall j \in \{1, ..., M\}$ that defines if the jth feature ϕ_{ji} is relevant $\gamma_j = 1$ or irrelevant $\gamma_j = 0$. Since the set of relevant features can vary between the x-y-z components of the manipulation skill, the robot selects a separate set of relevant features for each skill component. The feature selection process, as described in Section III-D, estimates the relevance of each feature γ_j based on the example trajectories. The linear regression for computing the skill parameters w_{ij} is performed using only the features ϕ_{ji} that are considered to be relevant $\gamma_j = 1$, and the skill parameters for the irrelevant features are set to zero.

C. Meta Features and Meta Priors

To guide the skill learning process, the robot learns a meta-level prior [26] to compute a prior over the feature relevances $p(\gamma_j)$. The prior has a logistic regression form

$$p(\gamma_j = 1) = \left(1 + \exp\left(\textstyle\sum_{h=1}^{H} \theta_h \varphi_{jh}\right)\right)^{-1}, \tag{2}$$

where the set of H parameters $\theta_h \forall h \in \{1, ..., H\}$ is the meta prior, and $\varphi_{jh} \forall h \in \{1, ..., H\}$ are the H meta features for the jth object feature ϕ_{ji}.

The meta features φ_{jh} represent properties of their respective object feature ϕ_j and how it relates to the skill. We extract the meta features from the initial and final scenes of individual demonstrations, as illustrated in Fig. 3. Each object feature ϕ_j is associated with an object part and a direction in which the feature is computed. We define the position of the part at the start of the demonstration as $p_j \in \mathbb{R}^3$ and the direction of the feature as $d_j \in \mathbb{R}^3$. The features describing the position or length of a part in the x direction will thus have $d_j = [1\ 0\ 0]^T$. Similarly, we define the initial position of the active robot hand as $\widehat{p} \in \mathbb{R}^3$, and the direction of the skill component being learned as $\widehat{d} \in \mathbb{R}^3$. Thus, when learning the y component of the manipulation skill the direction is $\widehat{d} = [0\ 1\ 0]^T$. We assume that the direction of the skill component $\widehat{d}$ remains constant, but the position and direction of the feature moves with the object

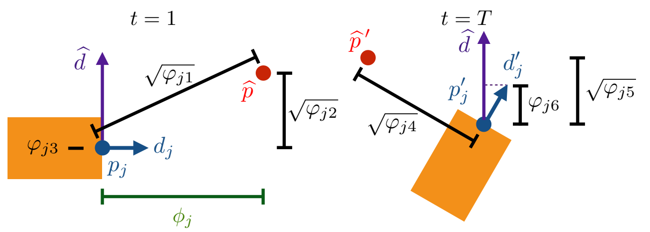


Fig. 3. The figure illustrates the six geometric meta features φ_{jh} for the feature ϕ_j. The initial and final scenes of the demonstration are shown on the left and right. The hand is located at $\widehat{p}$ and the feature's part is located at p_j. The horizontal feature defines a distance in the d_j direction, and the robot is currently learning the skill component for the vertical $\widehat{d}$ direction.

part to p_j' and d_j', as shown in Fig. 3. The hand position at the end of the demonstration is given by $\widehat{p}'$.

Given these properties, we define six meta features as

$$\begin{aligned} \varphi_{j1} &= ||p_j - \widehat{p}||^2 & \varphi_{j4} &= ||p_j' - \widehat{p}'||^2 \\ \varphi_{j2} &= ||\widehat{d}^T(p_j - \widehat{p})||^2 & \varphi_{j5} &= ||\widehat{d}^T(p_j' - \widehat{p}')||^2 \\ \varphi_{j3} &= |\widehat{d}^T d_j| & \varphi_{j6} &= |\widehat{d}^T d_j'| \end{aligned} \quad (3)$$

where the meta features in the first two rows represent the *proximity* of the object feature to the robot's hand, and the meta features in the third row capture the *alignment* of the object feature with the skill component. These meta features allow the robot to predict if, for example, a feature is more likely to be relevant because the hand moved closer or further away from its part during the demonstration, or because the feature was aligned with the skill component at the end of the demonstration. We include a binary seventh meta feature, which indicates whether the feature is representing the position or the length of a part, as well as a bias term.

These meta features allow the robot to compare features from different tasks, and thus transfer the prior knowledge of the features' relevances between distinct tasks. The meta prior $\theta_h \forall h \in \{1, ..., H\}$ indicates which meta features are informative for selecting relevant features. The meta prior is learned from previous skills using iteratively reweighted least squares. The training data for learning the meta prior includes the meta feature values φ and the feature relevances γ from the previous skills.

Given multiple demonstrations, the robot first computes the meta features for each demonstration and their respective estimates of the prior. The robot then takes the mean of these estimates to obtain the prior used for the feature selection.

D. Feature Selection

Using the example demonstrations and the prior over the features' relevances, the robot selects a set of relevant features for generalizing the manipulation skill. The robot uses the stochastic search variable selection (SSVS) algorithm to perform the feature selection. The graphical model of the SSVS with the meta prior is shown in Fig. 4. The distribution over the feature relevances γ_j is given by Eq. 2. The distribution over the shape parameter w_{jk} is given by zero-mean Gaussians with relevance-dependent variances

$$p(w_{jk}|\gamma_j) = \begin{cases} \mathcal{N}(0, \hat{s}^2) & \text{if } \gamma_j = 0 \\ \mathcal{N}(0, \check{s}^2) & \text{if } \gamma_j = 1, \end{cases} \quad (4)$$

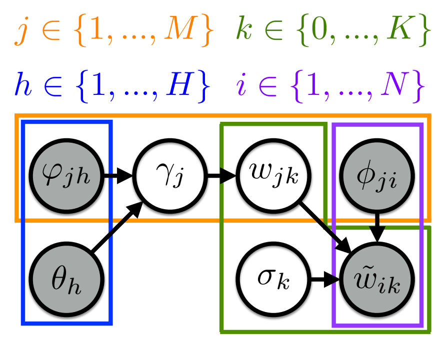


Fig. 4. The graphical model for the stochastic search variable selection (SSVS) with the meta-level prior. The feature selection is performed using N training samples. Each sample includes M object feature values ϕ_{ij} and the desired trajectory's $K+1$ shape parameter values $\tilde{w}_{ik}$. We also evaluated using the observed trajectory's shape parameters $\breve{w}_{ik}$ to select the relevant features. Each shape parameter is associated with a standard deviation σ_k, and the mapping from object features to shape parameters is defined by the skill parameters w_{jk}. Each of the M object features is associated with a binary relevance variable γ_j. The prior over the relevance variable γ_j is computed using H meta features φ_{jh} extracted from a demonstration and H meta prior parameters ϕ_h learned from previous skills.

where the variances were set to $\hat{s}^2 = 0.0225$ and $\check{s}^2 = 1.125$ for our experiments. Thus, the shape parameters corresponding to irrelevant features $\gamma = 0$ are more likely to be close to zero. The distribution over the shape parameters $\tilde{w}_{ik}$ is

$$p(\tilde{w}_{ik}|w_{(1:M)k}\phi_{(1:M)i}, \sigma_k^2) = \mathcal{N}(\textstyle\sum_{j=1}^{M} w_{jk}\phi_{ji}, \sigma_k^2) \quad (5)$$

where $\phi_{(1:M)i}$ are the M object feature values for the ith sample, and σ_k^2 is the output variance of the kth shape parameter. We model the distribution over the variances as

$$\sigma_k^2 \sim \text{Inv-Gamma}(a, b), \quad (6)$$

where we set the inverse-gamma distribution's shape parameter $a = 3$ and scale parameter $b = 3$ to constants.

The SSVS algorithm selects the relevant features based on the posterior distribution over the relevance features γ_j. This posterior distribution is approximated using Gibbs sampling [2], [33]. A feature is considered to be relevant if its posterior distribution is over 0.5, i.e., it had a relevance value of $\gamma_j = 1$ for the majority of the samples obtained through Gibbs sampling. Once the robot has selected a set of relevant object features, it computes the skill parameters w_{jk} for this sparse set of features using linear regression.

Rather than using the parameters for the desired trajectories $\tilde{w}_{ik}$ to select the relevant features, the robot can alternatively use the shape parameters $\breve{w}_{ik}$ extracted from the *observed* trajectories. Since the robot's arms are compliant, these trajectories will be bound by the physical constraints of the task. The robot may thus be able to detect these constraints better and generalize the skill accordingly. Once a set of relevant features has been extracted, the robot uses them to learn the mapping from the object features to the desired shape parameters as before. The robot can use the observed trajectories to compute the relevant features, but not for computing the final skill parameters.

E. Learning from Experience

Given the skill parameters w_{ij} and a set of object features ϕ_{ji} for a new scene, the robot computes an initial set of shape

Fig. 5. Objects used for learning manipulation skills in the experiments.

parameters $\tilde{w}_{ik}^0$ for the DMP. This initial skill's performance could potentially be improved. Our skill learning framework therefore incorporates reinforcement learning to improve the skill through experience. We use relative entropy policy search (REPS) to refine the shape parameters for a given scenario [34]. We assume that the robot can attempt the task repeatedly in the same scenario with constant object features. This assumption allows the robot to optimize the shape parameters $\tilde{w}_{ik}$ instead of the larger set of skill parameters w_{jk} used for generalization.

We define an initial Gaussian policy over the DMP's shape parameters as $\tilde{w}_i \sim \mathcal{N}(\tilde{w}_i^0, \Sigma^0)$, where $\tilde{w}_i^0$ is the vector of initial shape parameters $[\tilde{w}_i^0]_k = \sum_{j=1}^{M} w_{jk}\phi_{ji}$, and Σ^0 is an exploratory covariance matrix. We used diagonal covariance matrices and set the diagonal elements of Σ^0 to 50. The robot samples shape parameters from the policy and evaluates them by executing the skill and receiving a task-specific reward. After evaluating multiple samples from the current policy $\mathcal{N}(\tilde{w}^q, \Sigma^q)$, the robot computes an improved policy $\mathcal{N}(\tilde{w}^{q+1}, \Sigma^{q+1})$ based on the acquired rewards. The REPS algorithm maximizes the expected reward of the new policy while limiting the Kullback-Leibler divergence between the two consecutive policies. This bound leads to an improved policy convergence behavior.

Once the robot has learned the skill for the scenario, the final mean of the policy $\tilde{w}_{ik}^Q$ and the object feature values ϕ_{ji} are used as additional training data for selecting the relevant features and learning the skill parameters w_{jk}. Thus, the robot uses the knowledge acquired from this scenario to improve the generalization of the skill in the future.

IV. Experiments and Evaluations

The first experiment explores how the choice of trajectories and prior used to select the relevant features affects the skill learning process. In the second experiment, the robot learns a new meta prior from the first experiment's skills. The updated meta prior is used to learn basic wiping skills. The experiments were performed with assorted objects, including YCB objects [35], as shown in Fig. 5.

A. Learning to Tilt and Place Experiments

The first experiment evaluates the effects of using the desired trajectories versus the observed trajectories for selecting the relevant features together with either a meta-level prior or a uniform prior. The evaluations were performed using placing and tilting tasks. In the placing task, the robot has to place a held object on top of another object in the environment. The robot received a quadratic cost for the final distance between the middle of the bottom of the held object and the middle of the top of the supporting object. The robot also incurred a quadratic cost for shifting the supporting object, and a small quadratic cost for the distance moved in each time step of the trajectory. For the tilting task, we defined two points on the pivotal corners of the boxes being tilted. The robot received a quadratic cost for deviations of these points from their initial locations during the task executions. The robot also incurred a cost for the angle between the final object pose and the desired 90 degree rotation. We assumed a rigid grasp and used the robot's forward kinematics to track the held objects. Other objects were tracked with Niekum's Alvar AR tags ROS package.

To learn the placing and tilting skills, the robot was initially provided with six kinesthetic demonstrations of each task. The demonstrations were performed with three different sets of objects, with two demonstrations per object set. Each scene includes three objects, although only one or two objects were relevant for the tilting and placing tasks respectively. The rotational components of the demonstrations were similar for each skill. The robot therefore learned these components using a single constant bias feature. The robot learned DMPs with five shape parameters for each component. The placing and tilting tasks both use 24 object features plus a bias term, resulting in 125 skill parameters w_{jk} per translational component.

After learning the initial skills from demonstrations, the robot was presented with nine additional scenarios in which to improve the skills through trial and error. The policy search reinforcement learning was performed using three policy updates with five skill executions between each update. After each policy update, the robot executed the skill using the current mean of the policy to evaluate its performance.

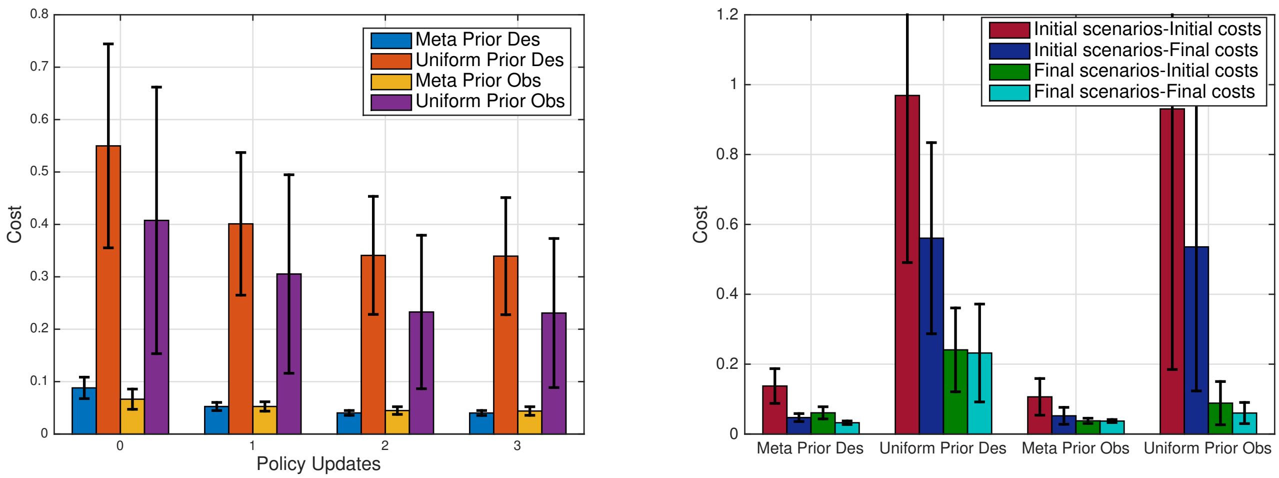
For the meta prior approach, the robot was provided with fifteen demonstrations of a pushing task, which included one skill for making contact with the side of the object and another for pushing the object. The robot used these demonstrations to extract meta features. The relevance of the features γ for the pushing task were manually labeled. The robot used these labels to learn a meta prior for predicting the relevance of the features for the placing and tilting tasks. The uniform prior is given by $p(\gamma_j) = 0.1 \forall j \in \{1, ..., M\}$.

The results of the experiment are shown in Fig. 6. Examples of the initial and final skills are shown in Fig. 7 and 8.

B. Discussion

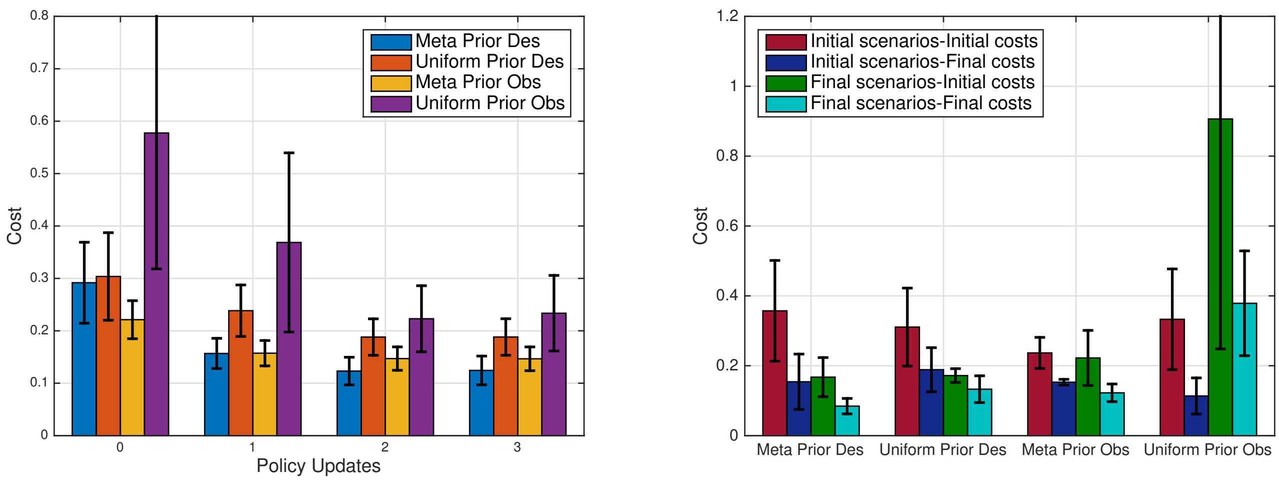
The downward trends in the left and right plots of Fig. 6 show that the skills are being improved through reinforcement learning and autonomous sample gathering respectively. The meta priors result in lower costs for the placing skill from the beginning. The MetaDes approach selected the x-y-z position of the supporting surface as the relevant features

Placing Experiment Results



Final	Initial Trial	Meta Des	Unif Des	Meta Obs	Unif Obs	Final Trial	Meta Des	Unif Des	Meta Obs	Unif Obs
Place	Initial Scn	6.48cm	27.4cm	5.96cm	24.6cm	Initial Scn	4.35cm	20.7cm	4.40cm	17.90cm
Errors	Final Scn	4.37cm	14.3cm	3.52cm	2.70cm	Final Scn	1.80cm	12.9cm	2.94cm	1.46cm

Tilting Experiment Results



Mean	Initial Trial	Meta Des	Unif Des	Meta Obs	Unif Obs	Final Trial	Meta Des	Unif Des	Meta Obs	Unif Obs
Pivot	Initial Scn	4.49cm	4.44cm	4.03cm	4.41cm	Initial Scn	3.13cm	3.62cm	3.35cm	2.68cm
Errors	Final Scn	2.86cm	3.47cm	3.47cm	7.29cm	Final Scn	2.13cm	2.93cm	2.91cm	5.06cm

Fig. 6. The top and bottom plots show the results for the placing and tilting experiments. The plots show the performances when using the meta prior versus the uniform prior approach, and the desired trajectory versus the observed trajectory, for selecting the relevant features. Lower costs indicate better performance and the error bars represent one standard error. The plots on the left show the reinforcement learning curves averaged over the nine different task scenarios. The plots on the right show the performance between the initial three (dark red and dark blue) and the final three (green and light blue) scenarios. The dark red and green bars correspond to zero policy updates, while the dark and light blue bars indicate the performance after three policy updates in the given scenarios. The top and bottom tables show the final position error after placing the object and the mean displacement of the pivot while tilting the object respectively. The middle rows correspond to the errors for the first three scenarios, while the bottom rows correspond to the last three scenarios. The left and right sides of the tables show the errors before and after applying reinforcement learning in the scenarios.

for the x-y-z skill components. It also included the redundant y position of the supporting object's bottom surface for the y component. All of these features, except for the y position of the supporting surface, were already selected for the first scenario by using the meta-level prior.

By contrast, the uniform priors selected suitable features for the skills' horizontal x component in the initial scenario, but the horizontal y component was either constant or incorrectly depended on the z position of the supporting surface. Given more samples, the uniform prior approaches ultimately included the y position of the supporting object. However the poor initial samples can reduce the quality of the learned skill. We will explore methods for weighting samples based on their performance in the future. The relatively high final placing cost of UnifDes is due to including the vertical z position of the irrelevant object for the y movement.

The UnifDes and MetaObs approaches did not select the z position of the supporting surface for the z component. Instead, these skills relied on the compliance of the robot to adapt to the surface's height. This result indicates that using the observed trajectory does not improve the feature selection for the constrained directions. Including a penalty

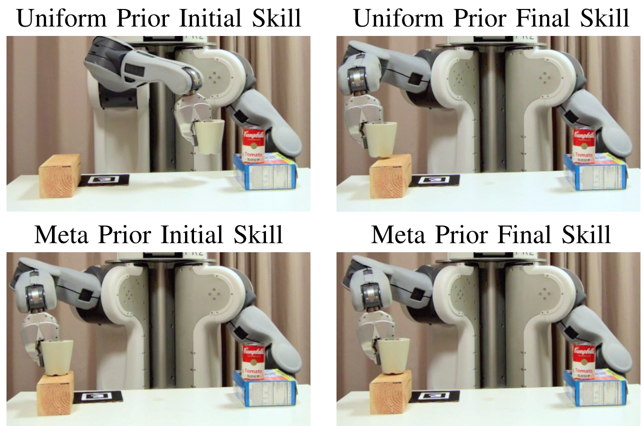


Fig. 7. The pictures show the performances of the placing skills. The robot should place the cup on the middle of the wooden block. The pictures on the left show the initial skills learned from six demonstrations. The pictures on the right show the final skills that were improved over nine additional scenarios using reinforcement learning. The top and bottom pictures show the skills learned using uniform and meta-level priors.

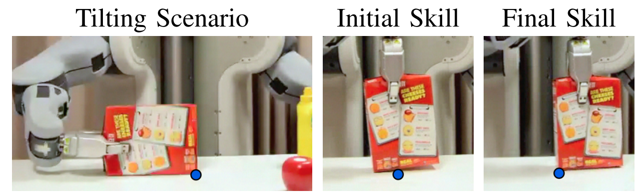


Fig. 8. The pictures show the initial and final tilting skills learned using the uniform prior. The skills learned using the meta-level prior are similar. The blue dot indicates the initial corner point, i.e., the desired pivot point. (Left) The initial evaluation scene. (Middle) The final pose using the skill learned from six demonstrations. (Right) The final pose using the skill learned from six demonstrations and nine reinforcement learning scenarios.

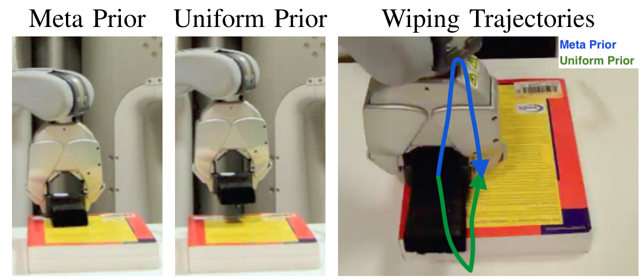


Fig. 9. (Left and Middle) Example goal states from the wipe preparation skills. The uniform prior's skill fails to make contact with the surface. (Right) Illustration of the wiping skills' trajectories. The meta prior's skill moves towards the center of the surface.

for large forces or deviations between desired and observed trajectories could improve the feature selection.

The robot tended to omit features corresponding to the cups and bowls being placed. The variances over these features are comparable to tracking errors and other sources of noise. As a result, their relevance is difficult to determine. Using a wider range of objects could alleviate this issue.

The surface centers used for the reward function are not the same as those extracted for the features. The features may therefore only be able to predict the placing position to within a couple of centimeters for the initial trials. Given tracking errors and other sources of noise, a placement error of less than 5cm is generally acceptable for this task. The robot could successfully perform the task even with larger errors, as shown in Fig. 7.

The meta prior provided less benefits for the tilting task than for the placing task. All four approaches used constant trajectories for the initial tilting movements in the y-z plane. Placing is similar to pushing, as both skills involve creating contacts between two objects. By contrast, tilting is a single-object skill that involves a 90 degree rotation. The benefit of the meta prior is therefore greater when the new skill is similar to the previously acquired skill.

The UnifObs learned a final skill that performed worse than the initial skill. The poor performance of this skill is due to the robot selecting two x direction features from the box and the z position of the irrelevant object as the features for the y movement. These selection errors could be attributed to nonlinearities in the observed trajectories resulting from the object-table interactions. The MetaObs approach avoided these errors by selecting aligned features and omitting features from irrelevant objects based on proximity.

Although the skill learning framework performed well and could learn the skills, individual components of the framework could be improved in the future. The skill learning framework currently assumes axis-aligned tasks and that the relative positions of objects are similar, e.g., the object is placed at the center of the supporting surface. Our future research will explore using more advanced features and incorporating variable rotations for aligning objects. We will also investigate using pose synthesis methods to predict suitable object-object poses for establishing contacts between objects [36], [19]. These predicted poses can then be used as virtual objects to generate additional features. The current meta features capture general concepts of proximity and alignment. We will explore using more specialized meta features in the future, e.g., meta features indicating if object parts are currently in contact with other objects.

The results of the experiment show that the robot should use the shape parameters from the desired trajectory to select relevant features for generalizing manipulation skills.

C. Transfer to Wiping Skill Experiment and Discussion

The robot used the learned placing and tilting skills to learn a new meta prior. The robot extracted meta features from six demonstrations of each of the pushing, placing, and tilting skills. The relevant features for placing and tilting were computed using SSVS with the meta prior and desired trajectories. Features with standard deviations of less than 0.015m were excluded from the training data, as their relevance cannot be reliably determined due to sensory noise.

Using the new meta prior, the robot learned basic wiping skills from six demonstrations each. The wiping task was divided into two skills: making contact with the surface, and wiping across the surface. The robot only used imitation learning for this experiment and did not refine the skills with reinforcement learning. We again compared using a uniform prior to the meta prior approach.

The skills were evaluated on 15 different scenarios using different sets of objects. The robot successfully made contact with the surface in 12 of the wipe preparation trials using the meta prior, and 8 trials when using the uniform prior. The robot successfully wiped the surface in 13 trials using the meta prior and 12 trials using the uniform prior. The objects were manually positioned at the start of the wiping task such that the results are independent of the wiping preparation skills.

The decreased performance of the uniform prior's wiping preparation skill is due to the inclusion of irrelevant features. The uniform prior selected the horizontal x length of the grasped object for adapting the vertical z movements, which resulted in the object being placed too high, as shown in Fig. 9. The meta prior approach successfully avoided these misalignment errors, resulting in improved performance for both skills.

For the wiping skill, the uniform prior approach selected a constant amplitude for the back-and-forth movement in the x direction. The meta prior approach selected the x position of the wiped surface as a relevant feature, due to it alignment and proximity to the hand. In this manner, the amplitude of the wiping motion adapts to the initial position of the object on the surface and always moves towards the center of the surface, as shown in Fig. 9. The results of the experiment show that the learned meta prior is useful for both rejecting irrelevant features as well as selecting relevant features.

V. Conclusion

We explored feature selection for learning manipulation skills. We used a skill learning framework that combines imitation, reinforcement, and transfer learning to efficiently learn versatile manipulation skills that adapt to different objects and scenarios. We explored performing the feature selection based on the desired and the observed trajectories. For each approach, we also evaluated the performance when using a uniform prior or a meta-level prior that allows the robot to predict the relevance of features based on the relevance of features from previously learned skills.

The skill learning framework was evaluated on placing, tilting, and wiping tasks. The evaluations show that using the desired trajectories, rather than the observed trajectories, results in improved learning behavior and skill performance. The experiments also showed that, in both cases, the meta prior resulted in improved learning performance, especially when the new skill is similar to a previously acquired skill.

References

[1] O. Kroemer and G. S. Sukhatme, "Meta-level priors for learning manipulation skills with sparse features," in *ISER*, 2016.

[2] E. I. George and R. E. McCulloch, "Variable selection via Gibbs sampling," *Journal of the American Statistical Association*, vol. 88, no. 423, pp. 881–889, 1993.

[3] J. Kober, K. Muelling, O. Kroemer, C. Lampert, B. Schoelkopf, and J. Peters, "Movement templates for learning of hitting and batting," in *ICRA*, 2010.

[4] B. da Silva, G. Baldassarre, G. Konidaris, and A. Barto, "Learning parameterized motor skills on a humanoid robot," in *ICRA*, 2014.

[5] A. Kupcsik, M. Deisenroth, J. Peters, L. Ai Poh, V. Vadakkepat, and G. Neumann, "Model-based contextual policy search for data-efficient generalization of robot skills," *Artificial Intelligence*, 2015.

[6] J. Peters and S. Schaal, "Policy gradient methods for robotics," in *IROS*, 2006.

[7] J. E. M. Do, J. Schill and T. Asfour, "Learn to wipe: A case study of structural bootstrapping from sensorimotor experience," in *ICRA*, 2014.

[8] Y. Chebotar, O. Kroemer, and J. Peters, "Learning robot tactile sensing for object manipulation," in *IROS)*, 2014.

[9] M. Kalakrishnan, L. Righetti, P. Pastor, and S. Schaal, "Learning force control policies for compliant manipulation," in *IROS*, 2011.

[10] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto, "Autonomous skill acquisition on a mobile manipulator," in *AAAI*, 2011.

[11] P. Pastor, M. Kalakrishnan, S. Chitta, E. Theodorou, and S. Schaal, "Skill learning and task outcome prediction for manipulation," in *ICRA*, 2011.

[12] S. Levine, N. Wagener, and P. Abbeel, "Learning contact-rich manipulation skills with guided policy search," in *ICRA*, 2015.

[13] S. Calinon, T. Alizadeh, and D. G. Caldwell, "On improving the extrapolation capability of task-parameterized movement models," in *IROS*, 2013.

[14] J. Piater, S. Jodogne, R. Detry, D. Kraft, N. Krueger, O. Kroemer, and J. Peters, "Learning visual representations for perception-action systems," *International Journal of Robotics Research*, 2011.

[15] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," in *ArXiv*, 2015.

[16] A. L. Pais, K. Umezawa, Y. Nakamura, and A. Billard, "Learning robot skills through motion segmentation and constraints extraction," HRI Workshop on Collaborative Manipulation, 2013.

[17] S. Niekum, "An integrated system for learning multi-step robotic tasks from unstructured demonstrations," in *AAAI*, 2013.

[18] J. Stuckler and S. Behnke, "Adaptive tool-use strategies for anthropomorphic service robots," in *Humanoids*, 2014.

[19] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis - a survey," *Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2014.

[20] O. Kroemer, E. Ugur, E. Oztop, and J. Peters, "A kernel-based approach to direct action perception," in *ICRA*, 2012.

[21] C. Kemp, A. Edsinger, and E. Torres-Jara, "Challenges for robot manipulation in human environments," *IEEE Robotics Automation Magazine*, vol. 14, no. 1, pp. 20–29, 2007.

[22] K. H. Lee, J. Lee, A. Thomaz, and A. Bobick, "Effective robot task learning by focusing on task-relevant objects," in *IROS*, 2009.

[23] M. Stolle, H. Tappeiner, J. E. Chestnutt, and C. G. Atkeson, "Transfer of policies based on trajectory libraries." in *IROS*, 2007.

[24] J. H. Metzen, A. Fabisch, L. Senger, J. de Gea Fernandez, and E. A. Kirchner, "Towards learning of generic skills for robotic manipulation," *Künstliche Intelligenz*, vol. 28, pp. 15–20, 2014.

[25] M. P. Deisenroth, P. Englert, J. Peters, and D. Fox, "Multi-task policy search for robotics," in *ICRA*, 2014.

[26] S. Lee, V. Chatalbashev, D. Vickrey, and D. Koller, "Learning a meta-level prior for feature relevance from multiple related tasks," in *ICML*, 2007.

[27] B. Taskar, M. F. Wong, and D. Koller, "Learning on the test data: Leveraging unseen features." in *ICML*, 2003.

[28] E. Krupka, A. Navot, and N. Tishby, "Learning to Select Features using their Properties," *Journal of Machine Learning Research*, vol. 9, pp. 2349–2376, 2008.

[29] M. Tenorth, S. Profanter, F. Balint-Benczedi, and M. Beetz, "Decomposing CAD models of objects of daily use and reasoning about their functional parts," in *IROS*, 2013.

[30] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut -interactive foreground extraction using iterated graph cuts," 2004.

[31] O. Kroemer and G. S. Sukhatme, "Learning relevant features for manipulation skills using meta-level priors," *Arxiv*, 2016.

[32] A. Ijspeert, J. Nakanishi, and S. Schaal, "movement imitation with nonlinear dynamical systems in humanoid robots," in *ICRA*, 2002.

[33] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.

[34] J. Peters, K. Muelling, and Y. Altun, "Relative entropy policy search," in *AAAI*, 2010.

[35] B. Çalli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The YCB object and model set and benchmarking protocols," *Arxiv*, 2015.

[36] O. Kroemer and J. Peters, "Predicting object interactions from contact distributions," in *IROS*, 2014.