Meta-level Priors for Learning Manipulation Skills with Sparse Features

Oliver Kroemer and Gaurav Sukhatme

University of Southern California,
3710 McClintock Ave., Los Angeles, CA 90089, USA
{okroemer,gaurav}@usc.edu
http://www.robotics.usc.edu/resl/

Abstract. Manipulation skills need to adapt to the geometric features of the objects that they are manipulating, e.g. the position or length of an action-relevant part of an object. However, only a sparse set of the objects' features will be relevant for generalizing the manipulation skill between different scenarios and objects. Rather than relying on a human to select the relevant features, our work focuses on incorporating feature selection into the skill learning process. An informative prior over the features' relevance can guide the robot's feature selection process. This prior is computed using a meta-level prior, which is learned from previous skills. The meta-level prior is transferred to new skills using meta features. Our robot experiments show that using a meta-level prior results in better generalization performance and more efficient skill learning.

Keywords: Manipulation, reinforcement learning, imitation learning, feature selection, transfer learning, skill generalization, meta priors

1 Introduction

Robots need to manipulate objects in their environment to perform a variety of different tasks. Manipulation skills can be learned in an efficient manner using imitation and reinforcement learning [1, 2]. In order to be versatile and generalize between different scenarios, these skills need to adapt to the features of the objects being manipulated. Although there will be many features describing the set of objects, only a few of them will be relevant to adapting the skill. The robot should therefore select a sparse set of relevant features for generalizing the manipulation skill.

Most of the work on skill learning has used manual feature selection by a human [3, 4]. In some cases, features are learned to localize task-relevant elements in the scenes [5, 6]. However, these approaches tend to focus on a fixed set of objects and the features are not learned to generalize between objects. Rather than relying on prior knowledge from a human, the robot should select the relevant features based on its own experiences and prior knowledge.

In this paper, we explore the use of meta-level priors [7] for efficiently selecting features in an imitation and reinforcement learning framework. The meta prior

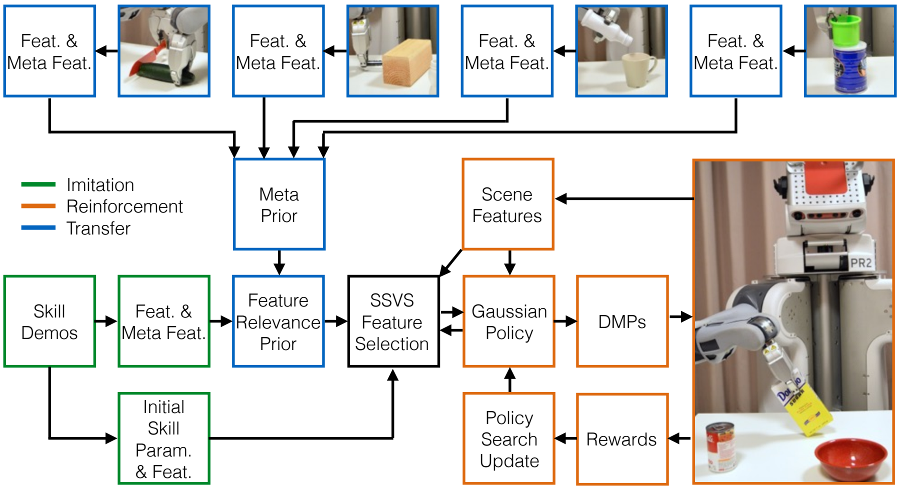


Fig. 1. An overview of the proposed framework. A new skill is initialized from demonstrations using imitation learning (green). The a priori feature relevance is computed using a meta-level prior, which is learned from previously learned skills (blue). The robot refines the skill by using policy search reinforcement learning (orange).

allows the robot to predict the relevance of a feature based on the feature's properties, i.e., its meta features [8]. We use meta features that describe the proximity and alignment of the features to the robot's hand. As part of a transfer learning framework, the meta prior is learned from the features of previously acquired skills.

Multi-task transfer learning for robot skills has often focused on tasks that involve similar manipulations and feature spaces [9, 10], e.g., throwing a ball to different locations. In our experiments, the robot learns skills for tasks with distinct feature spaces: placing, tilting, cutting, pouring, and wiping. We compare the performance of skills learned using the proposed meta-prior approach versus a standard uniform prior. The experiments show that the meta-level prior results in improved generalization performance and more efficient skill learning.

2 Technical Approach

The goal of our proposed approach is to learn a mapping from object features to the skill's shape parameters. The object features and the parameterized skill representation are described in Section 2.1. Only a sparse set of the object features will be relevant for generalizing the skill between different scenarios. The robot must therefore select a set of relevant object features for each skill component. Prior knowledge of each feature's relevance can be used to guide the feature selection process. This prior knowledge is transferred from previously learned skills using meta features and a meta-level prior, as explained in Section 2.2.

The robot uses stochastic search variable selection (SSVS), with a meta prior, to select a set of relevant features for generalizing each component of the skill, as described in Section 2.3.

Section 2.4 describes how the robot can learn to improve the skill's performance through experience. Each new skill is initially learned through imitation learning and subsequently improved through reinforcement learning. As the robot learns to perform the skill in new scenarios, the corresponding shape parameters are used as training samples for selecting the relevant features and computing the sparse skill parameters. Similarly, once a skill has been mastered and its relevant features have been extracted, it can be used to learn an improved meta prior for learning future skills.

An overview of the proposed framework is shown in Fig. 1. The learning framework consists of imitation learning (green), learning a meta-level prior (blue), and policy search reinforcement learning (orange). The feature selection (black) is at the core of the framework.

2.1 Object Features and Skill Representations

In order to execute a skill in a given scenario, the robot must first extract a set of object features describing the scene and the objects within it. These features describe the geometry of the objects that the robot is manipulating, e.g., a feature may define the x position of a container's opening or the width of the opening in the y direction. We assume that we have 3D point cloud models that capture the coarse shapes of the manipulated objects. These object models are first segmented into parts based on demonstrations of the skill using the GrabCut algorithm [11]. The segmentation process is initialized by detecting points that are in close proximity to other objects during the demonstrations [12]. Examples of object parts extracted for different tasks are shown in Fig. 2.

Each of the objects' parts is used to compute a generic set of object features ϕ. In our experiments, the six object features for each part consist of the 3D position of the part and the 3D size of its axis-aligned bounding box. The positions of the parts are defined relative to the robot's hand. The features from all of the parts are concatenated into a single set of M features $\phi_{ji} = \phi_j(S_i) \forall j \in \{1, ...M\}$ to represent scene S_i. The number of features M and the types of features may vary between different tasks. It is therefore usually not possible to define a 1-to-1 mapping between features of different tasks.

The manipulation skills are represented using dynamic movement primitives (DMPs) [13]. The task-space DMPs are defined relative to the initial position of the robot's hand. We focus on learning the three x-y-z translational components of the skill's movement. The standard DMP formulation uses a set of shape parameters $\tilde{w}_k \forall k \in \{1, ..., K\}$ and a goal offset $g - y_0 = \tilde{w}_0$ to define the shape of the skill's trajectory. In our reformulation, these shape parameters are represented as linear combinations of the object features $\tilde{w}_{ik} = \sum_{j=1}^{M} w_{jk} \phi_{ji} \forall k \in \{0, ..., K\}$. Thus, each feature ϕ_j has its own set of skill parameters w_{jk}, which define how the trajectory is adapted to the feature's value. For example, if the knife's length ϕ_j is doubled, then the amplitude of the cutting movement described by

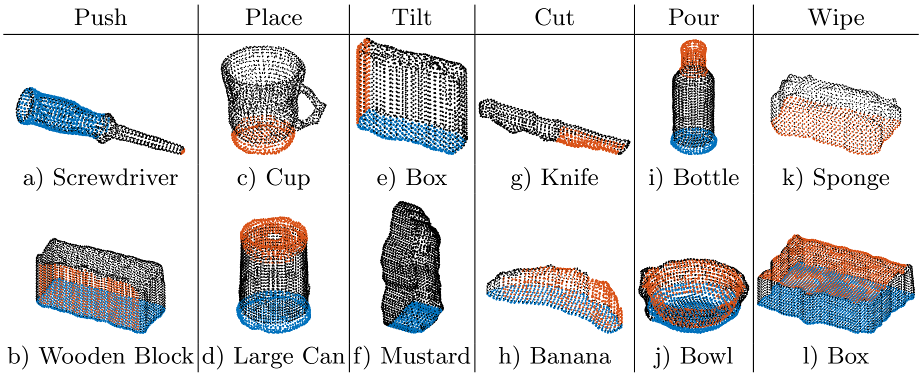


Fig. 2. Examples of object parts extracted from demonstrations of manipulations. The points show the objects' point cloud models, and the colors indicate different parts.

the corresponding skill parameters $w_{j0:K}$ will also double. The ultimate goal of our framework is to learn the skill parameters w_{jk} that map from the object features ϕ_{ji} to the shape parameters $\tilde{w}_{ik}$.

2.2 Meta Features and Meta Priors

The majority of the extracted features ϕ_j will not be relevant for generalizing the skill between different scenarios. The robot therefore needs to perform feature selection to select a sparse set of relevant features. We define a binary relevance variable $\gamma_j \in \{0, 1\}$ as $\gamma_j = 1$ iff ϕ_j is a relevant feature. An informative prior over the feature's relevance $p(\gamma_j)$ can guide the feature selection process and, thus, improve the skill's generalization performance. Rather than manually defining this prior, the robot should predict the relevance of individual features based on knowledge from previously learned skills.

The robot predicts the relevance of a feature ϕ_j using meta features $\varphi_{jh} \forall h \in \{1, ..., H\}$, which are extracted from the initial skill demonstrations. Meta features model characteristics of the object features and how they relate to the skill being learned. For example, each of our features ϕ_j is associated with a part of an object. We therefore define meta features that represent the initial and final distances between the position of the feature's part p_{jf} and the robot's hand p_h. These meta features represent the *proximity* of the feature. Using these meta features, the robot can predict that a feature is more relevant if the hand is near, or moves towards, the feature's part during the demonstrations. Each feature ϕ_j is also associated with a direction d_{jf} in which the length or position feature is computed. We include the inner product between the feature's direction and the skill component's axis d_a as another meta feature. These meta features represent the features' *alignment*. They can be used to predict the relevance of features for each x-y-z component of the DMPs. The first six meta features are illustrated in Fig. 3a. The seventh meta feature φ_{j7} has a value of $\varphi_{j7} = 1$ if the object feature describes the position of a part and $\varphi_{j7} = 0$ if it describes a part's length.

Given a set of meta features φ_{jh} for a new skill, the prior over the feature's relevance is computed as $p(\gamma_j) = (1+\exp(\sum_h \theta_h \varphi_{jh}))^{-1}$. The parameters θ_h are known as the meta prior [7]. The meta prior is learned from the meta features and feature relevances of previous skills using iteratively reweighted least squares.

2.3 Feature Selection

The robot selects a set of relevant features using stochastic search variable selection (SSVS) [14] with a meta-level prior. The graphical model for this framework is shown in Fig. 3b. A separate set of relevant features is selected for each x-y-z component of the skill.

The relevance variable γ_j of each feature ϕ_{ji} determines the prior variance over the feature's skill parameters w_{jk} such that $p(w_{jk}|\gamma_j = 0, \check{s}, \hat{s}) = \mathcal{N}(0, \check{s}^2)$ and $p(w_{jk}|\gamma_j = 1, \check{s}, \hat{s}) = \mathcal{N}(0, \hat{s}^2)$, where $\check{s}^2$ and $\hat{s}^2$ define narrow and broad Gaussians respectively. In this manner, an irrelevant feature's skill parameters are more likely to have values close to zero. In our experiments, we set $\check{s}^2 = 0.0225$ and $\hat{s}^2 = 1.125$. The prior over the feature relevance is given by the distribution $p(\gamma_j) = (1 + \exp(\sum_h \theta_h \varphi_{jh}))^{-1}$, as explained in Section 2.2

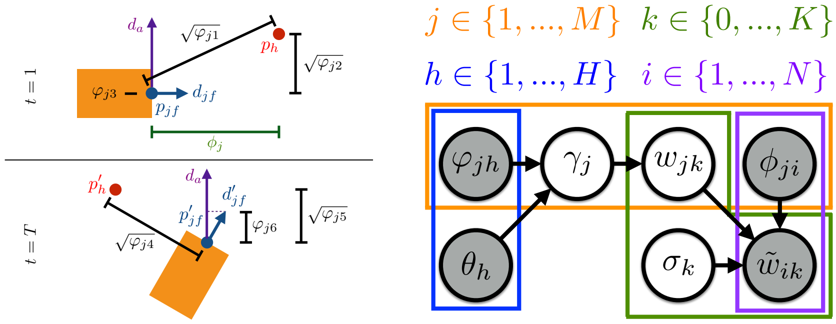
The distribution over the skill parameters is inferred from a set of N training samples. The i^{th} sample includes a set of values for the object features $\phi_{ji} \forall j \in \{1, ..M\}$ and a set of shape parameter values $\tilde{w}_{ik} \forall k \in \{0, ..K\}$ that define the desired trajectory of the hand. The object features and shape parameters are normalized during the data preprocessing. The distribution over the shape parameters is modelled using a Gaussian distribution such that $\tilde{w}_{ik} = \sum_{j=1}^{M} w_{jk}\phi_{ji} + \epsilon_{ik}$, where $\epsilon_{ik} \sim \mathcal{N}(0, \sigma_k^2)$. We model the distribution over the output variances σ_k^2 using an inverse gamma distribution with constant shape and scale parameters set to three in our experiments.

In order to select a set of relevant features, the robot estimates the posterior distribution over the relevance variables γ_j using a Gibbs sampling approach [15, 12]. For details on using Gibbs sampling for SSVS, we refer the reader to the paper of George and McCulloch [14].

Given the posterior distribution over the relevance parameters computed using Gibbs sampling, the robot computes the maximum a posteriori estimate of the relevance parameters. Hence, the robot selects a feature ϕ_j to be relevant iff the majority of the samples from the Gibbs sampling were $\gamma_j = 1$. Once the robot has selected a set of relevant features, the final skill parameters w_{jk} for the relevant features are learned using linear ridge regression. The skill parameters for the irrelevant features are set to zero. In this manner, the robot obtains a sparse set of parameters for generalizing the skill between different scenarios.

2.4 Learning from Experience

In order to master a skill and adapt to novel situations, a robot should be capable of learning from its own experiences.The robot can learn better estimates of the skill parameters w_{jk} by obtaining additional sets of object features ϕ_{ji} and corresponding shape parameters $\tilde{w}_{ik}$.



(a) Meta Features (b) Graphical Model

Fig. 3. The figure on the left (a) illustrates the meta features φ_{jh} computed for an object feature ϕ_j and hand position p_h at the start (top) and end (bottom) of a demonstrated trajectory. The figure on the right (b) shows the SSVS graphical model with the meta-level prior. Shaded nodes correspond to observed variables. Best viewed in color.

The robot extracts a new set of object feature values when it is presented with a novel scene in which to perform the skill. The robot can then also compute an initial estimate of the shape parameters $\tilde{w}^0_{ik}$ for this scenario using the current skill parameters. However, this initial estimate may perform poorly given that the robot is in the process of learning the skill. The robot can improve the shape parameters through trial and error learning. In particular, the robot uses a relative entropy policy search (REPS) approach to locally optimize the shape parameters [16]. We assume that the robot can attempt the skill execution multiple times in the same scenario, i.e., the object features are constant. This assumption allows the robot to optimize the shape parameters for the particular scene $\tilde{w}_{ik}$ rather than the larger set of skill parameters w_{jk} for generalization.

The robot creates an initial Gaussian policy over the shape parameters $\tilde{w}_i \sim \mathcal{N}(\tilde{w}^0_i, \Sigma^0)$, where the $\mathbb{R}^{K+1}$ vector $\tilde{w}^0_i = \sum_{j=1}^M w_j \phi_{ji}$ contains the initial estimates of the shape parameters, and Σ^0 is an initial exploratory covariance matrix. In our experiments, we assumed diagonal covariance matrices and initialized the diagonal elements with 50. The robot uses the policy to sample skill parameters from he policy and evaluates them on the task. After performing multiple executions with the current policy $\mathcal{N}(\tilde{w}^q, \Sigma^q)$, the robot computes an updated policy $\mathcal{N}(\tilde{w}^{q+1}, \Sigma^{q+1})$ based on the task rewards obtained for the sampled skill executions. REPS maximizes the expected reward of the new policy while limiting the Kullback-Leibler divergence between the previous and new policies, which leads to improved policy convergence behaviour.

Once the robot has learned to perform the skill for a given scenario, the final mean values of the Gaussian policy $\tilde{w}^Q_{ik}$ and the object feature values ϕ_{ji} are used as additional training data for selecting relevant features and learning the skill parameters w_{jk}. Similarly, once the robot has obtained a sufficient number of training samples and learned the relevant features for the skill, it can use the skill as additional training data for learning the meta-level prior. Determining the relevance of features with small variances across different scenarios is difficult, as they are prone to noise and their influence can often be incorrectly represented by a constant value. We therefore omitted features with standard deviations of less than 0.015 from the meta-prior training data.

3 Experiments

The proposed framework was evaluated in two experiments using a PR2 robot. The first experiment required the robot to learn skills using imitation and reinforcement learning. In the second experiment, the robot learned a meta prior based on the skills from the first experiment in order to predict the relevant features for new skills. In both experiments, we compared the robot's performance when using the meta-prior approach versus a standard uniform prior over the features' relevance.

3.1 Skill Learning Experiment

In the first experiment, the robot had to learn two skills: placing an object on another object (place), and tilting a box from one side to an adjacent side (tilt). The skills were learned through imitation and reinforcement learning, as described in Section 2. The PR2 robot was initially provided with six demonstrations of each skill using kinaesthetic teaching. Each skill was demonstrated with three sets of different objects, with two demonstrations per object set. We set $K = 5$. The rotation trajectories were consistent between demonstrations and could be modelled using constant shape parameters. The robot therefore only needed to learn the skills' translational components.

For each task, the robot was presented with nine novel scenarios in which to perform the skill. The scenarios included different sets of objects and different object locations. Each scenario included three objects, with at least one irrelevant object. In each scenario, the robot learned a set of shape parameters, as explained in Section 2.4. Between each scenario, the robot reselected the relevant features and recomputed the corresponding skill parameters using the new data.

In the placing task, the robot received a quadratic cost for the final distance between the middle of the bottom of the grasped object and the middle of the supporting surface, and for the distance that the supporting object was moved. For the tilting task, we defined two points along the pivotal edge of each box. The robot incurred a quadratic cost for deviations of these points from their initial locations, as well as a linear cost for the difference in rotation angle from the desired 90 degrees. The robot performed three policy updates for each scenario, with five skill executions between each update. The objects in the scenes were replaced at the end of each skill execution. The first skill execution after each update used the policy's mean parameters to evaluate the learning progress.

We evaluated the performance of the robot using both a uniform relevance prior $p(\gamma_j) = 0.1 \forall j \in \{1, ..., M\}$ and a meta-level prior. In order to learn the

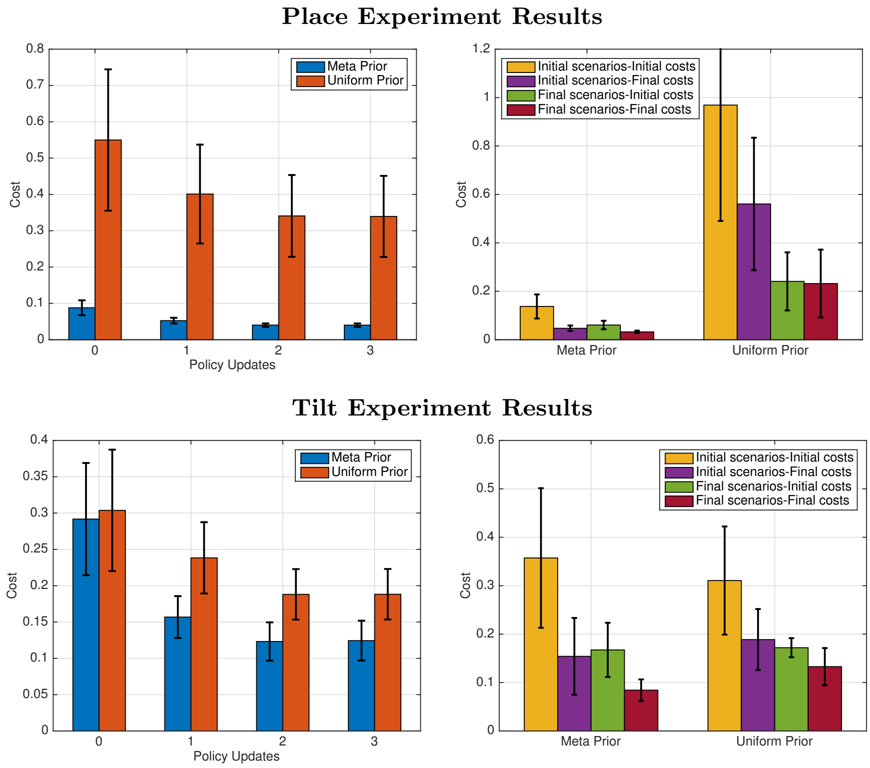


Fig. 4. The top and bottom plots show the results for the placing and tilting experiments respectively. Lower values indicate better performance. The plots on the left show the reinforcement learning curves averaged over the nine different task scenarios. The plots on the right show the performance between the initial three (yellow and purple) and the final three (green and red) scenarios. The yellow and green correspond to zero policy updates, while the purple and red bars correspond to the performance after three policy updates in the given scenario.

meta prior, the robot was also provided with 15 demonstrations of a pushing task. The task was divided into two skills: making contact with the side of the object, and pushing the object. The relevant features for these skills were manually labelled. The robot learned the meta prior using these relevance labels and the meta features extracted from the demonstrations.

The results of the placing and tilting experiments are shown in Fig. 4. The plots show the expected costs observed during the experiments, with lower values indicating better performance.

3.2 Skill Learning Experiment Discussion

The plots on the left of Fig. 4 show that the reinforcement learning improved the shape parameters for the individual scenarios, as the expected cost is decreasing in both tasks and for both priors. The plots on the right show the performance improvement across the different scenarios. The differences between the yellow and green bars is particularly important, as they show the initial performance of the DMPs before they are improved for the specific scenarios. These results show that the robot's ability to predict suitable skill parameters $\tilde{w}_{ik}$ improves as it obtains more training examples through reinforcement learning.

The meta-prior and uniform-prior approaches had similar performance levels for the tilting task. However, the meta-level prior performed considerably better than the uniform prior for the placing task. This performance gain is due to the similarity between the pushing and placing tasks. These two skills both involve moving a held object into contact with another object. The relevant features are also aligned with the skills' components, i.e., the skills' x components depend on x positions and not y or z positions. In contrast to the placing and pushing tasks, the tilting task involves a 90 degree rotation of the objects. As a result, the goal for the vertical z component of the DMP depends on the initial y position and length of the bottom surface. The prior computed for these features is lower than for the aligned features. However, the meta-prior approach still favours features associated with the grasped boxes due to their proximity to the robot's hand. The robot should therefore ideally use a meta prior that has been trained on a wide variety of different tasks. The robot should use more samples and a Gaussian policy with a larger variance to learn these initial skills.

In some cases, the shape parameters acquired through reinforcement learning did not fulfill the task requirements, e.g., the object was placed above the support surface and not on it. These shape parameters still performed better than the initial estimates $\tilde{w}_{ik}^0$ and were included as training samples for predicting shape parameters for future scenarios. However, as the robot masters the skill, it should remove samples that performed poorly compared to other samples in similar situations. In the future, we will explore methods of reweighting samples for the SSVS based on their performance.

Detecting relevant features that correspond to constraints on the robot's movements presented a challenge to the feature selection framework. For example, when using the uniform prior approach, the robot learned to use constant shape parameters for the vertical z-component of the placing skill. Rather than explicitly adapting the skill to the height of the supporting object, the robot learned to exploit the compliance in its arm to place the objects. These approaches could potentially be avoided by including a penalty based on the forces exerted on the objects. In the future, we will explore selecting the relevant features based on the observed trajectories rather than the desired trajectories.

The experiment showed that the robot can use reinforcement learning to acquire new training samples for selecting relevant features and learning the corresponding skill parameters. The meta prior can help the robot select relevant features and, hence, perform better when given limited number of training samples. The benefit of the meta-level prior is greater when the new skill shares similarities with the previous set of skills.

3.3 Transfer Learning Experiment

Having learned a set of relevant parameters for the placing and tilting tasks, the robot can now use these skills as additional examples for learning an improved meta prior θ_h. In the second experiment, the robot used a meta prior learned from the four previous skills to learn skills for three new tasks: cutting an object using a knife (cut), emptying a container into another container (pour), and wiping an object's flat surface with another object (wipe). The cutting and wiping tasks were divided into two skills each, with the first skill making contact between the two objects (prep cut and prep wipe) and the second performing the actual manipulation (cut and wipe). The prep cut skill was ultimately omitted from the evaluations due to safety concerns. The safe demonstrations of the cut prep skill lacked variety, which resulted in poor generalization performance.

The meta prior was trained using the meta features extracted from six demonstrations of each of the four previous skills. For comparison, the robot also learned a set of skills using a uniform prior. Each new skill was learned from six demonstrations using three sets of different objects. The set of manipulated objects used in the experiments are shown in Fig. 5a. The focus of this experiment is on investigating the initial performance of skills when using a meta-level prior versus a uniform prior. The robot therefore does not use reinforcement learning to improve the skills.

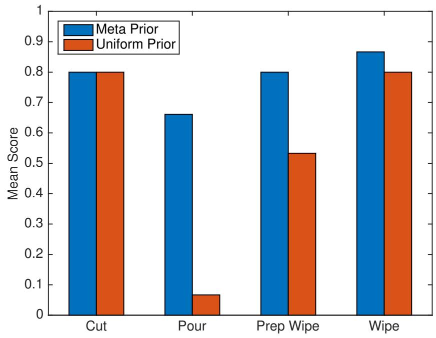
Each skill was executed in 15 different scenarios using various sets of three objects in different configurations. In order to compare the performances of the two approaches, the robot received a score between zero and one for each skill execution. For the cutting skill, the robot received one point if the knife cut into the object without slipping off or break contact with the object. The cutting motion consists of a single back-and-forth movement, and is not meant to directly cut through the object. For the pouring task, the robot received a score based on the proportion of the poured nuts that were transferred to the second container. For the prep wiping skills, the robot received a score of one if the tool made contact with the bottom right quadrant of the surface to be wiped. For the wiping skill, the robot received a point if it moved over the surface without breaking contact or slipping off. High scores do not indicate exceptional performance or mastery of the skill. Instead, they indicate that the computed shape parameters fulfill the basic goals of the skill and, hence, provide a suitable initialization for further refinement through trial and error. The average scores for the skills are shown in Fig. 5, with higher scores indicating better performance.

3.4 Transfer Learning Experiment Discussion

The results show that the robot could successfully perform the task for the majority of the presented scenarios when using the meta-level prior, with an average score of 0.78 across the four skills. In contrast, the uniform prior approach resulted in an average score of 0.55.

Using the meta prior helped the robot avoid selecting irrelevant features. The majority of the errors for the wipe prep skill with the uniform prior are due

(a) Objects used in Experiments



(b) Transfer Learning Results

Fig. 5. The left picture (a) shows the objects manipulated in the experiments. The plot on the right (b) shows the average scores obtained in the transfer learning experiment.

to placing the tool too high and thus failing to make contact with the surface. This offset is due to the robot incorrectly selecting the horizontal x width of the tool's part for the vertical z component of the skill. Similarly, the uniform prior resulted in poor performance for the pouring task because the y position of the second container's opening was selected as a relevant feature for the x component of the skill. Both of these errors were avoided by the meta-level prior, which had learned from previous skills that the relevant features are usually aligned with the direction of the skill component.

The meta-level prior also helped the robot to select relevant features for improving generalization. The wiping skills learned using the two priors were qualitatively different despite achieving similar scores. The uniform prior resulted in a skill with a constant amplitude for the back-and-forth movement of the wiping skill. By contrast, the meta prior approach selected the position of the middle of the wiped surface as a relevant feature. As a result, the wiping movement implicitly adapted to the size of the surface and was even inverted when the tool was placed on the far side of the wiped surface.

The experiment has shown the benefit of using a meta-level prior to transfer feature relevance knowledge between skills. Although the skills learned still require refinement through trial and error, the meta prior resulted in better initialization for the novel skills given the limited training data.

4 Conclusion

In this paper, we presented a framework for efficiently learning versatile manipulation skills by transferring knowledge from previous skills. The skills are initially learned through imitation learning and subsequently refined using reinforcement learning. The framework uses SSVS to select a sparse set of object features for generalizing the skill parameters. The feature selection process is guided by a meta-level prior on the relevance of each feature. The meta prior is learned from previously acquired skills and their relevant features. The robot predicts the relevance of object features for novel skills based on a set of seven meta features, which describe the proximity and alignment of the features to the hand.

The proposed framework was successfully evaluated on a variety of manipulation skills including placing, tilting, and pouring. The experiments show that the meta-level prior results in more efficient learning and better performance for novel skills that share similarities with previously learned skills.

Acknowledgements

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

References

1. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. RAS **57**(5) (May 2009) 469–483
2. Kober, J., Bagnell, D., Peters, J.: Reinforcement learning in robotics: A survey. (11) (2013) 1238–1274
3. da Silva, B.C., Baldassarre, G., Konidaris, G.D., Barto, A.: Learning parameterized motor skills on a humanoid robot. In: ICRA. (2014)
4. Muelling, K., Kober, J., Kroemer, O., Peters, J.: Learning to select and generalize striking movements in robot table tennis. IJRR (2013)
5. Levine, S., Finn, C., Darrell, T., Abbeel, P.: End-to-end training of deep visuomotor policies. JMLR (2016)
6. Piater, J., Jodogne, S., Detry, R., Kraft, D., Krueger, N., Kroemer, O., Peters, J.: Learning visual representations for perception-action systems. IJRR (2011)
7. Lee, S., Chatalbashev, V., Vickrey, D., Koller, D.: Learning a meta-level prior for feature relevance from multiple related tasks. In: ICML. (2007)
8. Krupka, E., Navot, A., Tishby, N.: Learning to Select Features using their Properties. JMLR (2008)
9. Deisenroth, M.P., Englert, P., Peters, J., Fox, D.: Multi-task policy search for robotics. In: ICRA. (2014)
10. Metzen, J.H., Fabisch, A., Senger, L., de Gea Fernandez, J., Kirchner, E.A.: Towards learning of generic skills for robotic manipulation. Germ. Journ. of AI (2014)
11. Rother, C., Kolmogorov, V., Blake, A.: GrabCut -interactive foreground extraction using iterated graph cuts. SIGGRAPH (2004)
12. Kroemer, O., Sukhatme, G.S.: Learning relevant features for manipulation skills using meta-level priors. ArXiv-CoRR (2016)
13. Schaal, S., Peters, J., Nakanishi, J., Ijspeert, A.: learning movement primitives. In: ISRR. (2004)
14. George, E.I., McCulloch, R.E.: Variable selection via Gibbs sampling. Journal of the American Statistical Association (1993)
15. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. PAMI (1984)
16. Peters, J., Muelling, K., Altun, Y.: Relative entropy policy search. In: AAAI. (2010)