A Simple Approach to Continual Learning by Transferring Skill Parameters

A Preprint

K.R. Zentner*, Ryan Julian*, Ujjwal Puri, Yulun Zhang, and Gaurav S. Sukhatme †

October 22, 2021

Abstract

In order to be effective general purpose machines in real world environments, robots not only will need to adapt their existing manipulation skills to new circumstances, they will need to acquire entirely new skills on-the-fly. A great promise of continual learning is to endow robots with this ability, by using their accumulated knowledge and experience from prior skills. We take a fresh look at this problem, by considering a setting in which the robot is limited to storing that knowledge and experience only in the form of learned skill policies. We show that storing skill policies, careful pre-training, and appropriately choosing when to transfer those skill policies is sufficient to build a continual learner in the context of robotic manipulation. We analyze which conditions are needed to transfer skills in the challenging Meta-World simulation benchmark. Using this analysis, we introduce a pair-wise metric relating skills that allows us to predict the effectiveness of skill transfer between tasks, and use it to reduce the problem of continual learning to curriculum selection. Given an appropriate curriculum, we show how to continually acquire robotic manipulation skills without forgetting, and using far fewer samples than needed to train them from scratch.

1 Introduction

Reinforcement learning (RL) with rich function approximators—so-called "deep" reinforcement learning (DRL)—has been used in recent years to automate tasks which were previously-impossible with computers, such as beating humans in board and video games Mnih et al. [2013, 2015] and navigating high-altitude baloons Bellemare et al. [2020]. In the field of robotics, DRL has shown the promise by allowing robots to automatically learn sophisticated manipulation behaviors end-to-end from high-dimensional multi-modal sensor streams Levine et al. [2016], and to quickly adapt these behaviors to new environments and circumstances Julian et al. [2020]. What remains to be seen is whether DRL can bridge the significant gap from efficiently adapting existing skills to efficiently acquiring entirely new skills. If such a capability could be applied repeatedly throughout the life of a robot (*i.e.* continual learning), we stand the chance of unlocking new possibilities for physical automation with general purpose robots, much as general purpose computers unlocked theretofore unforeseen possibilities for information automation half a century ago.

While not the only relevant formulation, we believe that episodic, continual, multi-task reinforcement learning is a worthy problem setting, because it describes this skill acquisition capability we seek. In this setting, we ask the robot to acquire new manipulation skills repeatedly, using time-delineated experiences of attempts at those skills (episodes), and some durable store of previously-acquired knowledge. The possibilities for the form of this store seem endless, but are actually bound to only two possibilities by construction: an RL system consumes raw data in the form of *experiences*, and outputs processed data in the form of *parameters*, which either directly specify a policy function, or condition a policy decision rule by specifying one or more other functions (*e.g.* a value function, Q-function, transition model, etc.). So a continual reinforcement learning robot can store one or both of (1) experience data or (2) parameters.

*Equal contribution

†All authors are with the Department of Computer Science, University of Southern California, Los Angeles, CA 90089. kzentner|rjulian|ujjwalpu|yulunzha|gaurav@usc.edu. GS holds concurrent appointments as a Professor at USC and as an Amazon Scholar.

Among these two options, there are many reasons to prefer parameters over data. Keeping a comprehensive dataset of all prior experience in local storage quickly becomes intractable for a single robot. Even if stored remotely in a "cloud" and retrieved, no server could quickly locate and retrieve a subset of that data relevant to a new task without first processing it into parameters itself. Parameters not only allow a single robot to store all of its skills locally, they are also more practical to share and disseminate than datasets (precisely because they are already processed). Consider the success of pre-trained models in language and computer vision: these are computed at great expense by institutions with immense datasets, storage, and compute resources. These institutions share them for the benefit of the entire community, who can then quickly re-use them for myriad applications. The parameters for state-of-the-art language and computer vision models fit on a cell phone in the palm of your hand, but require warehouse-sized machines to compute. While the best continual learning robot will likely store a mix of both data and parameters, we believe parameters deserve an especially-enthusiastic study. For the sake of simplicity, in this work we shall focus on them in isolation.

In this work, will show that under this skill storage-only assumption, efficient skill acquisition can be performed if the appropriate skills to transfer to each new task are known. We formalize our continual learning setting in Section 3, and describe how it maps onto the simulated robotic manipulation benchmark Meta-World. In Section 5 we investigate the precise conditions required to both transfer old skills and learn new ones using on-policy fine-tuning with Proximal Policy Optimization (PPO) Schulman et al. [2017]. We introduce a measure across ordered pairs of tasks that describes how efficiently on-policy transfer can learn a new task using skills from a single prior task. We then show how this measure allows constructing a curriculum predicted to be at least as efficient as learning each task without transfer, and we empirically verify that these predictions hold on the Meta-World benchmark.

2 Related Work

Reinforcement learning for robotics Reinforcement learning has been studied for decades as an approach for learning robotic capabilities Kober et al. [2013], Mahadevan and Connell [1992], Lin [1992], Smart and Kaelbling [2002]. In addition to manipulation skills Levine et al. [2018], Kalashnikov et al. [2018], Pinto and Gupta [2016], Gullapalli et al. [1994], Ghadirzadeh et al. [2017], Zeng et al. [2018], RL has been used for learning locomotion Kohl and Stone [2004a,b], Xie et al. [2019], Haarnoja et al. [2019], navigation Beom and Cho [1995], Zhu et al. [2017], motion planning Singh et al. [1994], Everett et al. [2018], autonomous helicopter flight Bagnell and Schneider [2001], Abbeel et al. [2007], Ng et al. [2003], and multi-robot coordination Matarić [1997], Yang and Gu [2004], Long et al. [2018]. The recent resurgence of interest in neural networks for use in supervised learning domains such as computer vision and natural language processing, (*i.e.* "deep learning" (DL)) Bengio et al. [2017], corresponded with a resurgence of interest in neural networks for reinforcement learning (*i.e.* "deep reinforcement learning" (DRL)) François-Lavet et al. [2018], Mnih et al. [2013]. With it came a wave of new research on using RL for learning in robotics and continuous control Mnih et al. [2015], Lillicrap et al. [2015], though the fields of neural networks, reinforcement learning, and robotics have overlapped continuously since each of their inceptions Kober et al. [2013], Hadsell et al. [2009].

Transfer, continual, and lifelong learning for robotics Transfer learning is a heavily-studied problem outside the robotics domain Donahue et al. [2014], Howard and Ruder [2018], Devlin et al. [2018], Dai et al. [2007], Raina et al. [2007]. Many approaches have been proposed for rapid transfer of robot skill policies to new domains, including residual policy learning Silver et al. [2018], simultaneously learning across multiple goals and tasks Ruder [2017], Rusu et al. [2016a], methods which use model-based RL Finn and Levine [2017], Yen-Chen et al. [2019], Nagabandi et al. [2019], Chatzilygeroudis and Mouret [2018], Ha and Schmidhuber [2018], Dasari et al. [2019], Chatzilygeroudis et al. [2018], Cully et al. [2015], Kaushik et al. [2020], Merel et al. [2019], Rastogi et al. [2018], Jeong et al. [2019], and goal-conditioned RL Agrawal et al. [2016], Nair et al. [2018], Pathak et al. [2018], Pong et al. [2019], Yu et al. [2019a]. All of these share data and representations across multiple goals and objects, but not skills per se. Similarly, work in robotic meta-learning focuses on learning representations which can be quickly adapted to new dynamics Nagabandi et al. [2018a], Alet et al. [2018], Nagabandi et al. [2018b] and objects Finn et al. [2017], James et al. [2018], Yu et al. [2018], Bonardi et al. [2019], but has thus far been less successful for skill-skill transfer Yu et al. [2019b]. Pre-training methods are particularly popular, including pre-training with supervised learning Deng et al. [2009], Levine et al. [2016], Finn et al. [2016], Gupta et al. [2018], Pinto and Gupta [2016], experience in simulation Sadeghi and Levine [2017], Tobin et al. [2017], Sadeghi et al. [2018], Tan et al. [2018], OpenAI et al. [2019], Rusu et al. [2016b], Peng et al. [2018], Higuera et al. [2017], Hämäläinen et al. [2019], auxiliary losses Riedmiller et al. [2018], Mirowski et al. [2016], Sax et al. [2019], and other methods Sermanet et al. [2017], Hazara and Kyrki [2019]. While successful, these methods are often designed for domain transfer rather than skill-skill transfer, require significant engineering by hand to anticipate specific domain shifts, and are designed for single-step rather than continual transfer. Similar to Julian et al. and Nair et al., our work uses the very simple approach of on-line fine-tuning to achieve rapid adaptation.

Lifelong and continual learning have long been recognized as an important capability for autonomous robotics Thrun and Mitchell [1995]. Like Taylor et al., our approach to continual learning relies on rapidly adapting policies for an already-acquired skill into a policy for a new skill. Much like Cao et al., Bodnar et al., and Kumar et al., this work uses experiments to analyze different transfer techniques from a geometric perspective on the skill-skill adaptation problem. As in prior work Luna Gutierrez and Leonetti [2020], Yen-Chen et al. [2020], this study observes that the selection of pre-training tasks is essential for preparing RL agents for rapid adaptation. Our work uses experiments to formulate a decision rule for how to pre-train our skills. A comprehensive overview of literature in continual reinforcement learning beyond robotics is beyond the scope of this work, but please see Khetarpal et al. for an excellent survey.

Reusable skill libraries for efficient learning and transfer Learning reusable skill libraries is a classic approach Gullapalli et al. [1994] for efficient acquisition and transfer of robot motion policies. Prior to the popularity of DRL-based methods, Associative Skill Memories Pastor et al. [2012] and Probabilistic Movement Primitives Rueckert et al. [2015], Zhou et al. [2020] were proposed for acquiring a set of reusable skills for robotic manipulation. In addition to manipulation Tanneberg et al. [2021], Yang et al. [2020], Ichter et al. [2020], Wulfmeier et al. [2020], Vezzani et al. [2020], Camacho et al. [2020], Li et al. [2021], Lu et al. [2021], Kroemer et al. [2015], DRL-based skill decomposition methods are particularly popular today for learning and adaptation in locomotion and whole-body humanoid control Peng et al. [2019], Hasenclever et al. [2020], Merel et al. [2020], Li et al. [2020], Tirumala et al. [2020]. Our work argues that once decomposed, these skill libraries are useful for rapid adaptation, and ultimately continual learning for manipulation with real robots. Hausman et al. proposed learning reusable libraries of robotic manipulation skills in simulation using RL and learned latent spaces, and Julian et al. showed these skill latent spaces could be used for efficient simulation-to-real transfer and rapid hierarchical task acquisition with real robots. As we also study in this work, learning reusable skill libraries requires exploring how new skills are related to old ones. As other works have pointed out Benureau and Oudeyer [2016], Singh et al. [2020a], Biza et al. [2021], Singh et al. [2020b], Allshire et al. [2021], we believe this can be achieved efficiently by re-using policies, representations, and data from already-acquired skills.

Continual robot learning with skill libraries and curriculums Like ours, recent works have begun to use DRL with skill libraries for continual robot learning. They have explored maintaining a skill library in form of factorized policy model classes Mendez et al. [2020], learned latent spaces Lu et al. [2020], Koenig and Matarić [2017], Hazara et al. [2019], options Hawasly and Ramamoorthy [2013], policy models which partition the state space Xiong et al. [2021], movement primitives Maeda et al. [2017], or as per-skill or all-skill datasets Traoré et al. [2019], Lu et al. [2020], Hazara et al. [2019]. As in our work and others Traoré et al. [2019], Stulp [2012], Fernández and Veloso proposed directly storing and re-using policies for continual learning in the context of robot soccer. Once acquired, these works propose various methods for reusing these skills, such as via online model-based planning Lu et al. [2020], via sequencing, mixture, selection, or generation with online inference Xiong et al. [2021], Stulp [2012], Hazara et al. [2019], Maeda et al. [2017], as a high-level action space for hierarchical RL Hawasly and Ramamoorthy [2013], and (as in our work) keeping a specific policy network for each new skill Mendez et al. [2020], Traoré et al. [2019], Koenig and Matarić [2017]. Intertwined with how to maintain such skill libraries is the question of how to update them throughout the life of the robot. Recent works have proposed using on-policy RL algorithms to directly update skills Mendez et al. [2020], Xiong et al. [2021], Stulp [2012], Koenig and Matarić [2017], Maeda et al. [2017], using a continually-growing skill data buffer to update skill networks Lu et al. [2020], Hazara et al. [2019], and repeatedly distilling the policy library Hawasly and Ramamoorthy [2013], Traoré et al. [2019].

We believe that continual learning for manipulation is achievable by using modular skill libraries and repeated efficient adaptation to new tasks. Alet et al., Sharma et al., and Raziei and Moghaddam have all recently proposed rapid adaptation methods for manipulation which make use of modular skill learning and re-use. This work seeks to extend some of those ideas, in simplified form, to the continual learning setting.

See Narvekar et al. for a survey of curriculum learning in reinforcement learning. Like Fabisch et al., our work observes that continual skill learning is an active learning problem, and that measuring task novelty is an important capability for efficient active skill learning. Like and Foglino et al., we highlight the importance of skill curriculum, and propose a method for computing the optimal skill curriculum given an oracle for relative skill novelty, and show that these curriculums indeed make continual learning more efficient.

3 Setting

We formalize our continual learning problem as iterated transfer learning for multi-task reinforcement learning (MTRL) on a possibly-unbounded discrete space of tasks $\mathcal{T}$. As we are interested in learning robot manipulation policies, we presume all tasks in $\mathcal{T}$ share a single continuous state space $\mathcal{S}$ and continuous action space $\mathcal{A}$, and the

MTRL problem is defined by the tuple $(\mathcal{T}, \mathcal{S}, \mathcal{A})$ Each task $\tau \in \mathcal{T}$ is an infinite-horizon Markov decision process (MDP) defined by the tuple $\tau = (\mathcal{S}, \mathcal{A}, p_\tau(s, a, s'), r_\tau(s, a, s'))$. As tasks are differentiated only by their reward functions r_τ and state transition dynamics p_τ, we may abbreviate this definition to simply $\tau = (r_\tau, p_\tau)$.

Importantly, we do not presume that the robot ever has access to all tasks in $\mathcal{T}$ at once, or even a representative sample thereof, and can only access one task at a time. We shall refer to time between task transitions an "epoch" and count them from 0, but in general two different epochs can be assigned the same task (*i.e.* tasks may reappear). When solving a task τ (hereafter, the "target task"), the robot only has access to skill policies acquired while solving prior tasks $\mathcal{M}$ (the "skill library"). When only a single prior task is used to solve a new task, we will refer to that task as the "prior task." Extending our problem to include these assumptions, we can say that a single epoch of this continual multi-task learning problem is defined by an infinite-horizon MDP $(\mathcal{S}, \mathcal{A}, \mathcal{M}_i, p_{\tau_i}, r_{\tau_i})$, where i is the epoch number and $\mathcal{M}_0$ is the (possibly-empty) set of manipulation skills with which the robot is initialized.

In this work, we will assume that the robot can choose which task τ to learn in each epoch, and also when to stop learning that task and begin a new epoch. In Section 5 we will discuss at length the implications of such decisions.

4 Simple Continual Learning with Skill Transfer

Now that we have defined our setting in detail, we can describe our proposed continual learning procedure in abstract. We begin with a (potentially empty) set of pre-trained skills $\mathcal{M}_0$. Then, in each epoch i, we choose a target task τ and base skill policy $\pi_{base} \in \mathcal{M}_i$. We then run an algorithm $\mathcal{F}$ to train a clone of π_{base} to solve τ. This may either accept π_{base} and return a new policy π_{target} or reject the selected π_{base}, in which case a new τ and π_{base} is chosen.

Algorithm 1 Proposed Continual Learning Framework

Input: Initial skill library $\mathcal{M}_0$, target task space $\mathcal{T}$, RL algorithm $\mathcal{F} \rightarrow (\pi, \rho)$, target task rule `ChooseTargetTask`, base skill rule `ChooseBaseSkill`
$i \leftarrow 1$
while not done **do**
  $\tau \leftarrow$ `ChooseTargetTask`$(\mathcal{T}, \mathcal{M}_{i-1})$
  while π_{target} not solved **do**
    $\pi_{base} \leftarrow$ `ChooseBaseSkill`$(\mathcal{T}, \mathcal{M}_{i-1})$
    $\pi_{target}, \cdot \leftarrow \mathcal{F}(\tau,$ `clone`$(\pi_{base}))$
  end while
  $\mathcal{M}_i \leftarrow \{\pi_{target}\} \cup \mathcal{M}_{i-1}$
  $i \leftarrow i + 1$
end while
Output: Skill library $\mathcal{M}_i$

In this work, we primarily use PPO for the RL algorithm $\mathcal{F}$, but in general $\mathcal{F}$ may be any parametric RL algorithm, including an off-policy algorithm. However, we find that we need to augment PPO in a few minor ways in order for it to perform adequately. These augmentations are general enough that they could be applied to most on-policy DRL algorithms.

"Warm-Up" Procedure for Value Function Transfer In order to tune a skill on a task using on-policy reinforcement learning, we also need a value function $V_{\tau, pi_{\tau'}}(s)$ which estimates the expected return of that skill policy on the task τ. Although PPO learns a value function as it trains the policy, we found that using a value function not fitted to the current task destroyed the skill policy's parameters before they could be transferred by PPO. Copying value functions of the same skill on prior tasks is particularly ineffective, since those value functions are very likely to overestimate initial performance, and are thus not admissible. To avoid this issue, before applying any gradient updates to the policy, we sample a batch of the skill's behavior on the new task, and train a new value function to convergence on the Monte Carlo return estimates from those samples. We found that this "value function warm-up" procedure was sufficient to produce an accurate value function of any skill on the new task, and was necessary to perform transfer effectively with PPO.

Rejecting Bad Transfers With Value Function Warm-Up, it is possible to transfer a skill policy π_{base} to a new task τ. However, this process is never completely reliable. If π_{base} is particularly unsuitable for solving τ, then PPO may be completely unable to transfer π_{base} to solve τ. Because we desire a *continual* learner, we must be able to continue learning without spending too much time on bad transfers. To this end, we make use of a rejecting rule that can stop training at any point in time, and request a new π_{base}. In this work, we use a rejection rule that compares the current

average reward of π_{base} on τ to the average reward of training a policy from-scratch to solve τ using 1.2 million (10% of total training time) fewer timesteps than have been used so far in the transfer process. If the transferred policy π_{base} ever "falls too far behind" the from-scratch policy, then we reject π_{base} and select a new π_{base} (possibly π_{random}).

4.1 Using Meta-World for Continual Learning

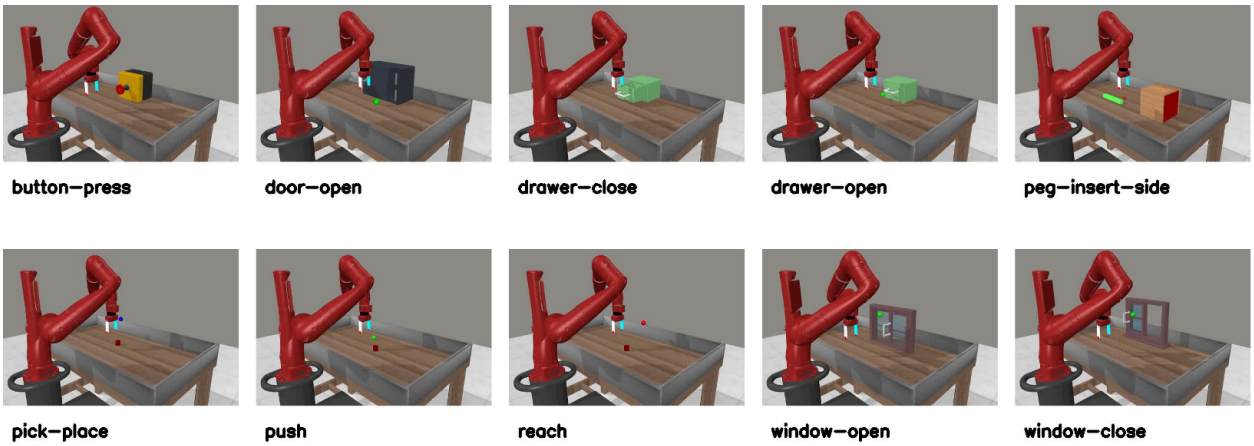

Figure 1: Images of Meta-World MT10 environments we use for experiments.

In order to perform useful continual learning experiments for robotic manipulation, we need a benchmark that allows us to learn multiple distinct robotic manipulation tasks. In this work we use environments from the Meta-World MT10 benchmarks. In particular, we consider each distinct environment in Meta-World as a seperate task (e.g. `pick-place`, `reach`, `push`, etc.). We use the fully observable variants of those environments, as found in the MT10 benchmark. This ensures that each "task" in our experiments internally contains parametric variation that skill policies must encode how to handle.

4.2 Learning Continually using Random Skill Transfer

With these augmentations, we can learn continually (if inefficiently), by transferring skills from random prior tasks. The performance of this method is equivalent to a "random curriculum" as presented in the next section. See Figure 4 for details on the performance of Random Skill Transfer.

5 Efficient Continual Learning with Skill Curriculum

Now that we have show that Random Skill Transfer can produce a continual learner that can learn by only transferring skill policies from one task to another, we would like to make a continual learner that is more efficient than from-scratch learning. We do this by constructing a "curriculum", which will be used by the function `ChooseBaseSkill` to choose a base skill for each target task. In the next Section, we will introduce a skill policy model for online curriculum selection, and share some promising results towards using it for continual learning.

We first develop a notion of skill-skill transfer cost, by counting the number of samples needed to acquire a target skill starting from a given base skill.

We then show that given this metric, we can reduce efficient continual learning to solving a Directed Minimum Spanning Tree (DMST) problem. We show the effectiveness of our skill curriculum selection algorithm by using it to continually learn all skills in the Meta-World MT10 benchmark using a fraction of the total samples needed to learn each skill from scratch.

5.1 Measuring Skill-Skill Transfer Cost

We define efficiency in continual learning as acquiring a skill policies for a given set of tasks while consuming the lowest number of environment samples possible.

Since we are reducing the continual multi-task learning problem to one of repeated skill-skill transfer, it follows that efficient continual learning is equivalent to minimizing the sum of the environment steps used for each adaptation

step. Without any prior on the relationship between two tasks, estimating such a quantity is difficult Sinapov et al. [2015]. This is compounded by the fact that skill-skill transitions are not independent: the robot only has access to skill policies it acquired during adaptation to tasks it has already seen, so the lowest-cost skill-skill transition for any given epoch depends on the skills which were acquired in previous epochs.

To make progress on our central question, we make two simplifying assumptions: (1) before continual learning begins, we can access the task space to build a cost metric for skill-skill transfer and (2) we assume that skill-skill transfer costs are conditionally-independent (*i.e.* as long as the robot has a skill policy for a manipulation task, the skill-skill transfer cost is independent of the skill transfer sequence it used to acquire that policy). Our experiments below with Meta-World MT10 will validate the conditional independence assumption.

To build our offline skill-skill cost metric, before continual learning begins, we train a skill policy from scratch for each task $\tau \in \mathcal{T}$. We then use these as base skill policies, and retrain a copy of each to solve each other target task. We considered several ways we could define the cost of transferring a skill to a new task. Our initial experiments used the inverse of the average success rate throughout a fixed training interval as the cost. This metric is convenient because it is always well defined, even if the transferred skill fails to learn the new task. However, because transferred skills often exhibits a sudden improvement in performance after a fixed number of environment timesteps, this metric is highly sensitive to the size of the training interval used to compute the performance ratio. It also does not accurately represent the resources needed in the learning process (namely, the number of environment steps needed to train). Because most skills that succeed during transfer eventually reach a $> 90\%$ success rate, we chose to use that threshold as a performance criteria instead, and use the number of timesteps required to reach it as the skill-skill transfer cost C (See Equation 1). This is similar to the "jumpstart" measure used for skill curriculum inference by Sinapov et al..

Note that in Yu et al. [2019b], several tasks which we include in our experiments cannot be reliably solved to this success rate. However, we find that our restarting rule allows us to learn all skills we include in our experiments to the required success rate.

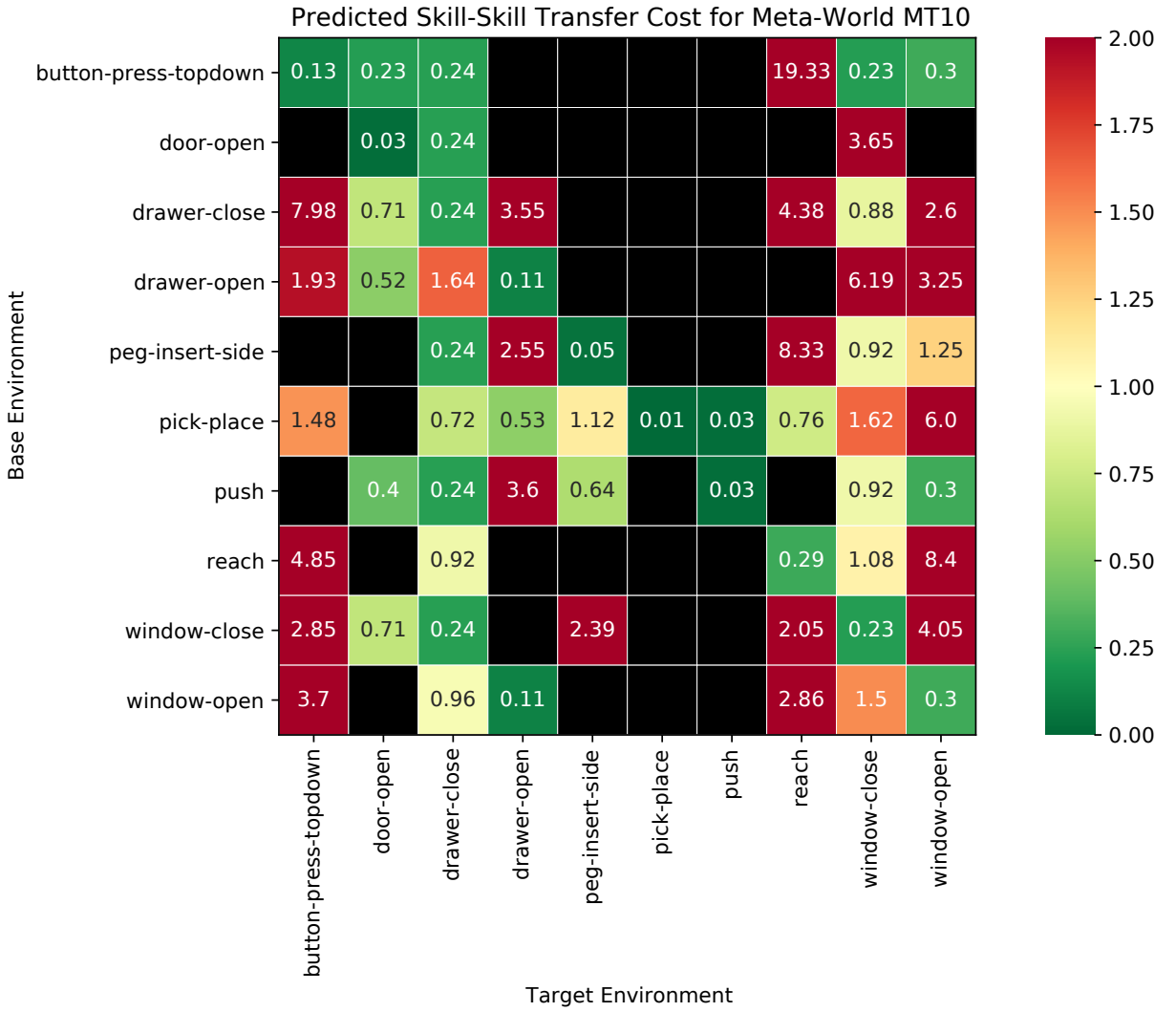

Figure 2: This figure shows $A_{base \to target}$, as defined in Equation 1, the ratio of time steps required to learn a task in MT10 by skill transfer to learning the same task from scratch, using each other possible skill policy as a base skill. Note that we always run at least a single training iteration, and use at most 12 million timesteps. This prevents the matrix from containing 0 along the diagonal. Black cells in the matrix correspond to transfers that did not reach a 90% success rate.

$$
\begin{aligned}
C_{base \to target} &= \text{EnvironmentSteps}_{base \to target} \\
C_{scratch \to target} &= \text{EnvironmentSteps}_{scratch \to target} \\
A_{base \to target} &= \frac{C_{base \to target}}{C_{scratch \to target}}
\end{aligned}
\tag{1}
$$

5.2 Curriculum Selection Algorithm

Given a skill-skill transfer cost for each task $\tau \in \mathcal{T}$, we can generate a curriculum which minimizes the transfer cost of learning each $\tau \in \mathcal{T}$. We refer to the choice of skill to transfer on each task as "the curriculum." Our curriculum selection algorithm is based on the observation that we can interpret the skill-skill transfer cost matrix in Figure 2 (which shows $A_{base \to target}$) as the weighted adjacency matrix of a densely-connected directed graph, with our skill-skill transfer cost metric $C_{base \to target}$ as the directed edge weights. Under this interpretation, we can extract the lowest cost of visiting all tasks by solving for the Directed Minimum Spanning Tree (DMST) using Kruskal's Algorithm Kruskal [1956]. The resulting tree will have a total edge weight equal to the minimal number of environment steps required to learn all tasks, as predicted by the skill-skill transfer cost metric.

To complete this DMST formulation, we need to take into account the possibility that it is most efficient to train a skill policy from scratch, rather than starting learning with one which already exists in the skill library $\mathcal{M}$. To achieve this, we add a "scratch" vertex to our graph representation, with out-directed edges from the scratch vertex to each task with edge weight equal to the number of training steps needed to learn that task from scratch. This would be represented in Figure 2 as an addition row, whose values are all 1.0.

In addition to solving for the DMST to find the optimal curriculum predicted by our cost metric, we can also solve for the Directed Maximum Spanning Tree to find a predicted pessimal curriculum. See Figure 3 for examples of optimal and pessimal curriculum trees computed using our DMST-based method for Meta-World MT10, using the skill-skill transfer cost data in Figure 2.

Our experience indicates that these graphs make intuitive sense, and tell us a few things about the structure of the task space and the most efficient tasks with which to pre-train for continual learning. For instance, the optimal curriculum features two distinct sub-trees, with roots corresponding to the two rows of Figure 2 that contain the largest number of useful transfers ($A_{base \to target} < 1$).

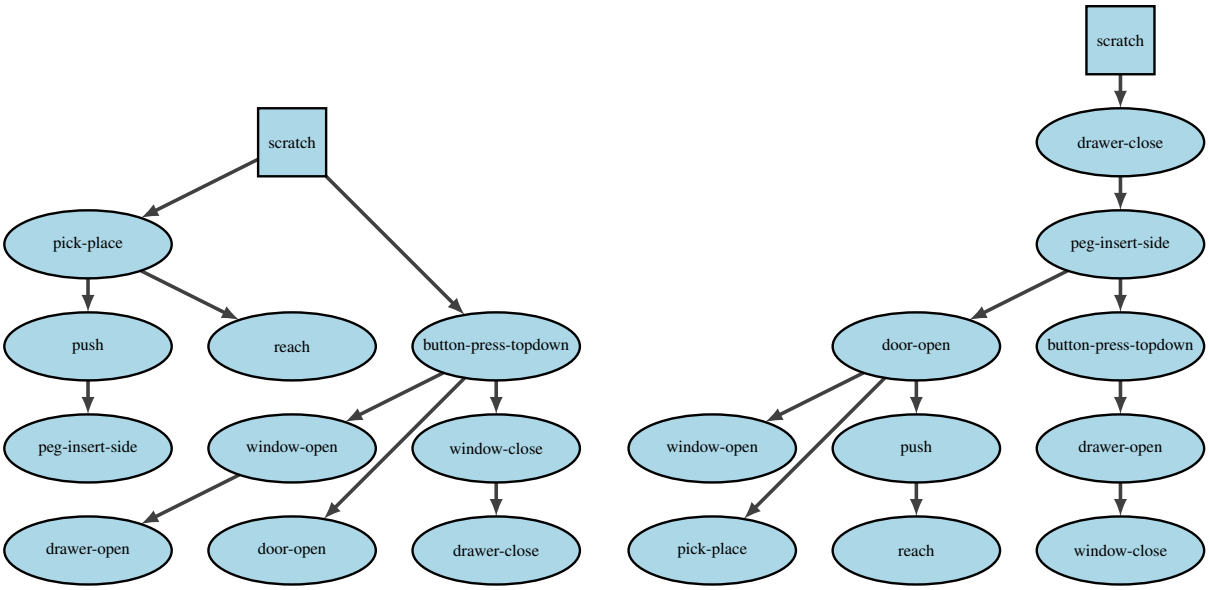

Figure 3: **Left:** Predicted optimal curriculum DMST for MT10. Note that the curriculum begins by learning the *hardest* tasks first, then transfers those skills to easier tasks. **Right:** Predicted pessimal curriculum DMST for MT10.

The first sub-tree contains all tasks which can be solved by grasping an object (and reach, which is equivalent to grasping a "fictional" object in midair). It begins with `pick-place`, which is empirically the most-challenging task in the benchmark, and contains the grasp skill while covering as much of the state space as possible. It then transfer `pick-place` to `push`, which uses the grasp skill to move the grasp object along the table surface, and `reach`, which it solves trivially. The second sub-tree contains all other tasks, all of which involve manipulating some object which is fixed to the table across from the table and below the robot gripper's initial position. In this subtree, `window-open` is transferred to `drawer-open` and `window-close` is transferred `drawer-close`. This is somewhat surprising, since one might expect transfer to happen most readily between tasks with the same objects and therefore most similar

state distributions. However, this structure instead corresponds to transferring from more challenging tasks to easier ones with similar motions.

For the pessimal curriculum, similar observations about inter-task structure hold, but with reversed consequences. The pessimal curriculum instead begins by learning the empirically-easiest task, `drawer-close` which can be solved very quickly buy simply reaching forward. It then solves one of the hardest tasks, `peg-insert-side`, which requires essentially the same number of samples as scratch training when starting from `drawer-close`. The remainder of the curriculum alternates between solving easier tasks and harder tasks, terminating in harder tasks. Each harder-easier transition destroys behaviors useful for future tasks, making the learning process take as long as possible.

Not only do these results suggest that our pairwise transfer cost metric discovers structure in the task space useful for learning skills, it also suggests a somewhat counter-intuitive result: that the most efficient curriculum begins by learning the hardest tasks first, then using them to solve the easier tasks. Most curriculum learning methods in RL instead learn the easiest tasks first, then use them to generalize to harder tasks. These results suggest we should initialize our skill library $\mathcal{M}$ by learning policies for the hardest tasks first. We empirically confirm these results with continual learning experiments using these curricula.

Given the solved optimal curriculum tree $T_{optimal}$, Curriculum Skill Transfer learns all tasks continually by learning skill in the sequence produced by the tree traversal on $T_{optimal}$, starting from the root "scratch" node (Algorithm 2). Note that if we reject a transfer that was predicted to succeed by our cost metric, we remove that edge from the graph, re-compute the curriculum using DMST to find a better curriculum online, and continue learning under the new curriculum.

Algorithm 2 DMST-Based Curriculum Transfer

Input: Initial skill library $\mathcal{M}_0$, target task space $\mathcal{T}$, RL algorithm $\mathcal{F} \rightarrow (\pi, C)$
$V \leftarrow \mathcal{T} \cup scratch$
$E \leftarrow \{\}$
for $\tau_{base} \in \mathcal{T}$ **do**
  $E \leftarrow (scratch, \tau_{base}, -1.0)$
  $\pi_{base}, C_{scratch \rightarrow target} \leftarrow \mathcal{F}(\tau_{base}, \pi_{random})$
  for $\tau_{target} \in \mathcal{T}$ **do**
    $\cdot, C_{base \rightarrow target} = \mathcal{F}(\tau_{target}, \pi_{base})$
    $E \leftarrow E(\tau_{base}, \tau_{target}, C_{base \rightarrow target}) \cup E$
  end for
end for
$T_{optimal} \leftarrow \texttt{kruskal}((V, E))$
$i \leftarrow 1$
$\pi_{base} \leftarrow \pi_{random}$
for $\tau_{target} \in \texttt{traverse}(T_{optimal})$ **do**
  while τ_{target} not solved **do**
    $\pi_{target}, \cdot \leftarrow \mathcal{F}(\tau_{target}, \texttt{clone}(\pi_{base}))$
    if τ_{target} not solved **then**
      $E \leftarrow E \setminus (\tau_{base}, \tau_{target})$
      $T_{optimal} \leftarrow \texttt{kruskal}((V, E))$
    end if
  end while
  $\mathcal{M}_i \leftarrow \{\pi_{target}\} \cup \mathcal{M}_{i-1}$
  $i \leftarrow i + 1$
end for
Output: Skill library $\mathcal{M}$

5.3 Measuring the Effectiveness of Curriculum Selection

We demonstrate the effectiveness of our curriculum selection algorithm in Figure 4, in which the total cost in environment steps is shown for each success rate we could use as a success criteria, calculated using the data from the skill-skill cost metric training experiments. The optimal curriculum computed using our curriculum selection algorithm matches or outperforms training from scratch for success rates between 80% and 95%.

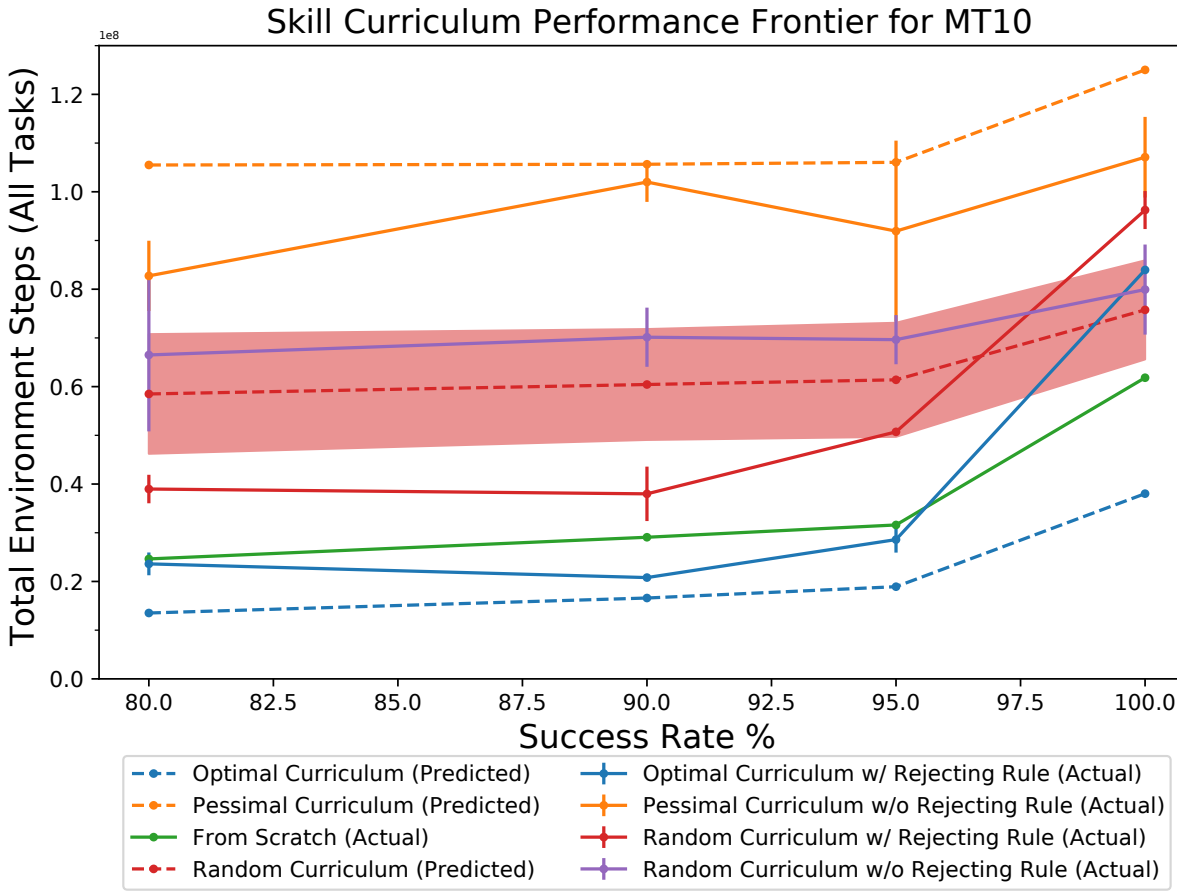

Figure 4: Comparison of the performance-sample efficiency frontier for several curricula for Meta-World MT10. The high level of agreement between the actual and predicted curricula indicate that our skill-skill transfer cost metric accurately predicts the true cost to transfer skills. Note that the success rate is presented on the X-axis, since it is chosen as the point from which to stop learning one task and begin learning another. Error bars show one standard deviation in number of total environment steps require to solve all tasks. However, runs using the rejecting rule exhibit very low variance between runs.

6 Conclusion

In this work, we introduced a simple approach to continual learning for robotic manipulation, based around a growing library of skill policies and repeated skill-skill fine-tuning. We first introduced the method as an abstract framework, and showed that a naïve implementation of that framework (Random Skill Transfer) achieves continual skill learning without forgetting, but is not more efficient than training all skills from scratch. We discussed the importance of skill curricula for efficient continual learning, reduced the efficient continual learning problem to minimizing total skill-skill transfer cost, developed an offline metric for measuring that cost based on skill-skill fine-tuning performance, and used this metric to solve for optimal and pessimal curricula using an DMST-based algorithm. We found that the optimal and pessimal curricula produced by this DMST-based curriculum algorithm are not only intuitive, but they make clear a rather unintuitive result: that it is best to pre-train continual learners with the hardest manipulation skills first. Finally, we tested these curricula to continually learn Meta-World MT10, and verified that continual learning with the optimal curriculum outperforms training from scratch, and the pessimal curriculum performs much worse than both training scratch and a random curriculum.

In future work, we will investigate ways of approximating our cost metric, as well as policy model classes that can transfer skills without explicit cost metrics. We look forward to developing a method for active online curriculum selection, and using it to achieve efficient continual learning.

References

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. ISSN 14764687. doi: 10.1038/nature14236. URL https://www.nature.com/nature/journal/v518/n7540/pdf/nature1423\6.pdf.

Marc G Bellemare, Salvatore Candido, Pablo Samuel Castro, Jun Gong, Marlos C Machado, Subhodeep Moitra, Sameera S Ponda, and Ziyu Wang. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588(7836):77–82, 2020.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Ryan Julian, Benjamin Swanson, Gaurav S Sukhatme, Sergey Levine, Chelsea Finn, and Karol Hausman. Never stop learning: The effectiveness of fine-tuning in robotic reinforcement learning. *arXiv e-prints*, pages arXiv–2004, 2020.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

Sridhar Mahadevan and Jonathan Connell. Automatic programming of behavior-based robots using reinforcement learning. *Artificial intelligence*, 55(2-3):311–365, 1992.

Long-Ji Lin. Reinforcement learning for robots using neural networks, 1992.

William D Smart and L Pack Kaelbling. Effective reinforcement learning for mobile robots. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 4, pages 3404–3410. IEEE, 2002.

Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4-5):421–436, 2018. doi: 10.1177/0278364917710318. URL https://doi.org/10.1177/0278364917710318.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*, pages 651–673, 2018.

Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 3406–3413. IEEE, 2016.

Vijaykumar Gullapalli, Judy A Franklin, and Hamid Benbrahim. Acquiring robot skills via reinforcement learning. *IEEE Control Systems Magazine*, 14(1):13–24, 1994.

Ali Ghadirzadeh, Atsuto Maki, Danica Kragic, and Mårten Björkman. Deep predictive policy training using reinforcement learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2351–2358. IEEE, 2017.

Andy Zeng, Shuran Song, Stefan Welker, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. *arXiv preprint arXiv:1803.09956*, 2018.

Nate Kohl and Peter Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 3, pages 2619–2624. IEEE, 2004a.

Nate Kohl and Peter Stone. Machine learning for fast quadrupedal locomotion. In *The Nineteenth National Conference on Artificial Intelligence*, pages 611–616, July 2004b.

Zhaoming Xie, Patrick Clary, Jeremy Dao, Pedro Morais, Jonathan Hurst, and Michiel van de Panne. Iterative reinforcement learning based design of dynamic locomotion skills for cassie. *arXiv preprint arXiv:1903.09537*, 2019.

Tuomas Haarnoja, Sehoon Ha, Aurick Zhou, Jie Tan, George Tucker, and Sergey Levine. Learning to walk via deep reinforcement learning. In *Robotics: Science and Systems*, 2019.

Hee Rak Beom and Hyung Suck Cho. A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning. *IEEE transactions on Systems, Man, and Cybernetics*, 25(3):464–477, 1995.

Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3357–3364. IEEE, 2017.

Satinder P Singh, Andrew G Barto, Roderic Grupen, Christopher Connolly, et al. Robust reinforcement learning in motion planning. *Advances in neural information processing systems*, pages 655–655, 1994.

Michael Everett, Yu Fan Chen, and Jonathan P How. Motion planning among dynamic, decision-making agents with deep reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3052–3059. IEEE, 2018.

J Andrew Bagnell and Jeff G Schneider. Autonomous helicopter control using reinforcement learning policy search methods. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 2, pages 1615–1620. IEEE, 2001.

Pieter Abbeel, Adam Coates, Morgan Quigley, and Andrew Y Ng. An application of reinforcement learning to aerobatic helicopter flight. *Advances in neural information processing systems*, 19:1, 2007.

Andrew Y Ng, H Jin Kim, Michael I Jordan, and Shankar Sastry. Autonomous helicopter flight via reinforcement learning. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, pages 799–806, 2003.

Maja J Matarić. Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4(1):73–83, 1997.

Erfu Yang and Dongbing Gu. Multiagent reinforcement learning for multi-robot systems: A survey. Technical report, Tech Report, 2004.

Pinxin Long, Tingxiang Fan, Xinyi Liao, Wenxi Liu, Hao Zhang, and Jia Pan. Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6252–6259. IEEE, 2018.

Yoshua Bengio, Ian Goodfellow, and Aaron Courville. *Deep learning*, volume 1. MIT press Massachusetts, USA:, 2017.

Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G Bellemare, and Joelle Pineau. An introduction to deep reinforcement learning. *arXiv preprint arXiv:1811.12560*, 2018.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Raia Hadsell, Pierre Sermanet, Jan Ben, Ayse Erkan, Marco Scoffier, Koray Kavukcuoglu, Urs Muller, and Yann LeCun. Learning long-range vision for autonomous off-road driving. *Journal of Field Robotics*, 26(2):120–144, 2009.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning*, pages 193–200, 2007.

Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pages 759–766, 2007.

Tom Silver, Kelsey Allen, Josh Tenenbaum, and Leslie Kaelbling. Residual policy learning. *arXiv preprint arXiv:1812.06298*, 2018.

Sebastian Ruder. An overview of multi-task learning in deep neural networks, 2017.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, abs/1606.04671, 2016a. URL `http://arxiv.org/abs/1606.04671`.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2786–2793. IEEE, 2017.

Lin Yen-Chen, Maria Bauza, and Phillip Isola. Experience-embedded visual foresight. *arXiv preprint arXiv:1911.05071*, 2019.

Anusha Nagabandi, Kurt Konoglie, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. *arXiv preprint arXiv:1909.11652*, 2019.

Konstantinos Chatzilygeroudis and Jean-Baptiste Mouret. Using parameterized black-box priors to scale up model-based policy search for robotics. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9. IEEE, 2018.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, pages 2450–2462, 2018.

Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning, 2019.

Konstantinos Chatzilygeroudis, Vassilis Vassiliades, and Jean-Baptiste Mouret. Reset-free trial-and-error learning for robot damage recovery. *Robotics and Autonomous Systems*, 100:236–250, 2018.

Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.

Rituraj Kaushik, Pierre Desreumaux, and Jean-Baptiste Mouret. Adaptive prior selection for repertoire-based online adaptation in robotics. *Frontiers in Robotics and AI*, 6:151, 2020.

Josh Merel, Saran Tunyasuvunakool, Arun Ahuja, Yuval Tassa, Leonard Hasenclever, Vu Pham, Tom Erez, Greg Wayne, and Nicolas Heess. Reusable neural skill embeddings for vision-guided whole body movement and object manipulation. *arXiv preprint arXiv:1911.06636*, 2019.

Divyam Rastogi, Ivan Koryakovskiy, and Jens Kober. Sample-efficient reinforcement learning via difference models. In *Technical Report*, 2018.

Rae Jeong, Jackie Kay, Francesco Romano, Thomas Lampe, Tom Rothorl, Abbas Abdolmaleki, Tom Erez, Yuval Tassa, and Francesco Nori. Modelling generalized forces with reinforcement learning for sim-to-real transfer. *arXiv preprint arXiv:1910.09471*, 2019.

Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. In *Advances in neural information processing systems*, pages 5074–5082, 2016.

Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. In *Advances in Neural Information Processing Systems*, pages 9191–9200, 2018.

Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Fred Shentu, Evan Shelhamer, Jitendra Malik, Alexei A. Efros, and Trevor Darrell. Zero-shot visual imitation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2018. doi: 10.1109/cvprw.2018.00278. URL `http://dx.doi.org/10.1109/CVPRW.2018.00278`.

Vitchyr H Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: State-covering self-supervised reinforcement learning. *arXiv preprint arXiv:1903.03698*, 2019.

Tianhe Yu, Gleb Shevchuk, Dorsa Sadigh, and Chelsea Finn. Unsupervised visuomotor control through distributional planning networks. *arXiv preprint arXiv:1902.05542*, 2019a.

Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *arXiv:1803.11347*, 2018a. URL `https://openreview.net/forum?id=HyztsoC5Y7`.

Ferran Alet, Tomás Lozano-Pérez, and Leslie P Kaelbling. Modular meta-learning. *arXiv preprint arXiv:1806.10166*, 2018.

Anusha Nagabandi, Chelsea Finn, and Sergey Levine. Deep online learning via meta-learning: Continual adaptation for model-based rl. *arXiv preprint arXiv:1812.07671*, 2018b.

Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. *arXiv preprint arXiv:1709.04905*, 2017.

Stephen James, Michael Bloesch, and Andrew J Davison. Task-embedded control networks for few-shot imitation learning. In *Conference on Robot Learning*, pages 783–795, 2018.

Tianhe Yu, Chelsea Finn, Annie Xie, Sudeep Dasari, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot imitation from observing humans via domain-adaptive meta-learning. In *International Conference on Learning Representations*, 2018.

Alessandro Bonardi, Stephen James, and Andrew J Davison. Learning one-shot imitation from humans without humans. *arXiv preprint arXiv:1911.01103*, 2019.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. *arXiv preprint arXiv:1910.10897*, 2019b.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 512–519. IEEE, 2016.

Abhinav Gupta, Adithyavairavan Murali, Dhiraj Prakashchand Gandhi, and Lerrel Pinto. Robot learning in homes: Improving generalization and reducing dataset bias. In *Advances in Neural Information Processing Systems*, pages 9112–9122, 2018.

Fereshteh Sadeghi and Sergey Levine. Cad2rl: Real single-image flight without a single real image. *Robotics: Science and Systems XIII*, July 2017. doi: 10.15607/rss.2017.xiii.034. URL http://dx.doi.org/10.15607/RSS.2017.XIII.034.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30, September 2017. doi: 10.1109/iros.2017.8202133. URL http://dx.doi.org/10.1109/IROS.2017.8202133.

Fereshteh Sadeghi, Alexander Toshev, Eric Jang, and Sergey Levine. Sim2real viewpoint invariant visual servoing by recurrent control. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4691–4699, 2018. doi: 10.1109/CVPR.2018.00493. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Sadeghi_Sim2Real_Viewpoint_Invariant_CVPR_2018_paper.html.

Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. *Robotics: Science and Systems XIV*, June 2018. doi: 10.15607/rss.2018.xiv.010. URL http://dx.doi.org/10.15607/RSS.2018.XIV.010.

OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving rubik's cube with a robot hand, 2019.

Andrei A. Rusu, Matej Vecerík, Thomas Rothörl, Nicolas Manfred Otto Heess, Razvan Pascanu, and Raia Hadsell. Sim-to-real robot learning from pixels with progressive nets. In *CoRL*, 2016b.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1–8. IEEE, 2018.

Juan Camilo Gamboa Higuera, David Meger, and Gregory Dudek. Adapting learned robotics behaviours through policy adjustment. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5837–5843. IEEE, 2017.

Aleksi Hämäläinen, Karol Arndt, Ali Ghadirzadeh, and Ville Kyrki. Affordance learning for end-to-end visuomotor robot control. *arXiv preprint arXiv:1903.04053*, 2019.

Martin A. Riedmiller, Roland Hafner, Thomas Lampe, Michael Neunert, Jonas Degrave, Tom Van de Wiele, Volodymyr Mnih, Nicolas Manfred Otto Heess, and Jost Tobias Springenberg. Learning by playing solving sparse reward tasks from scratch. In *ICML*, 2018.

Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, et al. Learning to navigate in complex environments. *arXiv preprint arXiv:1611.03673*, 2016.

Alexander Sax, Bradley Emi, Amir R. Zamir, Leonidas J. Guibas, Silvio Savarese, and Jitendra Malik. Mid-level visual representations improve generalization and sample efficiency for learning visuomotor policies. In *Conference on Robot Learning*, 2019.

Pierre Sermanet, Kelvin Xu, and Sergey Levine. Unsupervised perceptual rewards for imitation learning. *Proceedings of Robotics: Science and Systems (RSS)*, 2017. URL http://arxiv.org/abs/1612.06699.

Murtaza Hazara and Ville Kyrki. Transferring generalizable motor primitives from simulation to real world. *IEEE Robotics and Automation Letters*, 4(2):2172–2179, 2019.

Ashvin Nair, Murtaza Dalal, Abhishek Gupta, and Sergey Levine. Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Sebastian Thrun and Tom M Mitchell. Lifelong robot learning. *Robotics and autonomous systems*, 15(1-2):25–46, 1995.

Matthew E Taylor, Peter Stone, and Yaxin Liu. Transfer learning via inter-task mappings for temporal difference learning. *Journal of Machine Learning Research*, 8(9), 2007.

Zhangjie Cao, Minae Kwon, and Dorsa Sadigh. Transfer reinforcement learning across homotopy classes. *IEEE Robotics and Automation Letters*, 6(2):2706–2713, 2021.

Cristian Bodnar, Karol Hausman, Gabriel Dulac-Arnold, and Rico Jonschkowski. A geometric perspective on self-supervised policy adaptation. *arXiv preprint arXiv:2011.07318*, 2020.

Saurabh Kumar, Aviral Kumar, Sergey Levine, and Chelsea Finn. One solution is not all you need: Few-shot extrapolation via structured maxent rl. *Advances in Neural Information Processing Systems*, 33, 2020.

R Luna Gutierrez and M Leonetti. Information-theoretic task selection for meta-reinforcement learning. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*. Leeds, 2020.

Lin Yen-Chen, Andy Zeng, Shuran Song, Phillip Isola, and Tsung-Yi Lin. Learning to see before learning to act: Visual pre-training for manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7286–7293. IEEE, 2020.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *arXiv preprint arXiv:2012.13490*, 2020.

Peter Pastor, Mrinal Kalakrishnan, Ludovic Righetti, and Stefan Schaal. Towards associative skill memories. In *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pages 309–315, November 2012. doi: 10.1109/HUMANOIDS.2012.6651537.

E. Rueckert, J. Mundo, A. Paraschos, J. Peters, and G. Neumann. Extracting low-dimensional control variables for movement primitives. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1511–1518, May 2015. doi: 10.1109/ICRA.2015.7139390.

Xuefeng Zhou, Hongmin Wu, Juan Rojas, Zhihao Xu, and Shuai Li. Incremental learning robot task representation and identification. In *Nonparametric Bayesian Learning for Collaborative Robot Multimodal Introspection*, pages 29–49. Springer, 2020.

Daniel Tanneberg, Kai Ploeger, Elmar Rueckert, and Jan Peters. Skid raw: Skill discovery from raw trajectories. *IEEE Robotics and Automation Letters*, 2021.

Ruihan Yang, Huazhe Xu, Yi Wu, and Xiaolong Wang. Multi-task reinforcement learning with soft modularization. *arXiv preprint arXiv:2003.13661*, 2020.

Brian Ichter, Pierre Sermanet, and Corey Lynch. Broadly-exploring, local-policy trees for long-horizon task planning. *arXiv preprint arXiv:2010.06491*, 2020.

Markus Wulfmeier, Dushyant Rao, Roland Hafner, Thomas Lampe, Abbas Abdolmaleki, Tim Hertweck, Michael Neunert, Dhruva Tirumala, Noah Siegel, Nicolas Heess, et al. Data-efficient hindsight off-policy option learning. *arXiv preprint arXiv:2007.15588*, 2020.

Giulia Vezzani, Michael Neunert, Markus Wulfmeier, Rae Jeong, Thomas Lampe, Noah Siegel, Roland Hafner, Abbas Abdolmaleki, Martin Riedmiller, and Francesco Nori. " what, not how"–solving an under-actuated insertion task from scratch. *arXiv preprint arXiv:2010.15492*, 2020.

Alberto Camacho, Jacob Varley, Deepali Jain, Atil Iscen, and Dmitry Kalashnikov. Disentangled planning and control in vision based robotics via reward machines. *arXiv preprint arXiv:2012.14464*, 2020.

Yunfei Li, Yilin Wu, Huazhe Xu, Xiaolong Wang, and Yi Wu. Solving compositional reinforcement learning problems via task reduction. *arXiv preprint arXiv:2103.07607*, 2021.

Yuchen Lu, Yikang Shen, Siyuan Zhou, Aaron Courville, Joshua B Tenenbaum, and Chuang Gan. Learning task decomposition with ordered memory policy network. *arXiv preprint arXiv:2103.10972*, 2021.

Oliver Kroemer, Christian Daniel, Gerhard Neumann, Herke Van Hoof, and Jan Peters. Towards learning hierarchical skills for multi-phase manipulation tasks. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1503–1510. IEEE, 2015.

Xue Bin Peng, Michael Chang, Grace Zhang, Pieter Abbeel, and Sergey Levine. Mcp: Learning composable hierarchical control with multiplicative compositional policies. *arXiv preprint arXiv:1905.09808*, 2019.

Leonard Hasenclever, Fabio Pardo, Raia Hadsell, Nicolas Heess, and Josh Merel. CoMic: Complementary task learning & mimicry for reusable skills. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4105–4115. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/hasenclever20a.html.

Josh Merel, Saran Tunyasuvunakool, Arun Ahuja, Yuval Tassa, Leonard Hasenclever, Vu Pham, Tom Erez, Greg Wayne, and Nicolas Heess. Catch & carry: reusable neural controllers for vision-guided whole-body tasks. *ACM Transactions on Graphics (TOG)*, 39(4):39–1, 2020.

Tianyu Li, Nathan Lambert, Roberto Calandra, Franziska Meier, and Akshara Rai. Learning generalizable locomotion skills with hierarchical reinforcement learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 413–419. IEEE, 2020.

Dhruva Tirumala, Alexandre Galashov, Hyeonwoo Noh, Leonard Hasenclever, Razvan Pascanu, Jonathan Schwarz, Guillaume Desjardins, Wojciech Marian Czarnecki, Arun Ahuja, Yee Whye Teh, et al. Behavior priors for efficient reinforcement learning. *arXiv preprint arXiv:2010.14274*, 2020.

Karol Hausman, Jost Tobias Springenberg, Ziyu Wang, Nicolas Heess, and Martin Riedmiller. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rk07ZXZRb.

Ryan C Julian, Eric Heiden, Zhanpeng He, Hejia Zhang, Stefan Schaal, Joseph Lim, Gaurav S Sukhatme, and Karol Hausman. Scaling simulation-to-real transfer by learning composable robot skills. In *International Symposium on Experimental Robotics*. Springer, 2018. URL https://ryanjulian.me/iser_2018.pdf.

Fabien CY Benureau and Pierre-Yves Oudeyer. Behavioral diversity generation in autonomous exploration through reuse of past experience. *Frontiers in Robotics and AI*, 3:8, 2016.

Avi Singh, Huihan Liu, Gaoyue Zhou, Albert Yu, Nicholas Rhinehart, and Sergey Levine. Parrot: Data-driven behavioral priors for reinforcement learning. *arXiv preprint arXiv:2011.10024*, 2020a.

Ondrej Biza, Dian Wang, Robert Platt, Jan-Willem van de Meent, and Lawson LS Wong. Action priors for large action spaces in robotics. *arXiv preprint arXiv:2101.04178*, 2021.

Avi Singh, Albert Yu, Jonathan Yang, Jesse Zhang, Aviral Kumar, and Sergey Levine. Cog: Connecting new skills to past experience with offline reinforcement learning. *arXiv preprint arXiv:2010.14500*, 2020b.

Arthur Allshire, Roberto Martín-Martín, Charles Lin, Shawn Manuel, Silvio Savarese, and Animesh Garg. Laser: Learning a latent action space for efficient reinforcement learning. *arXiv preprint arXiv:2103.15793*, 2021.

Jorge Mendez, Boyu Wang, and Eric Eaton. Lifelong policy gradient learning of factored policies for faster training without forgetting. *Advances in Neural Information Processing Systems*, 33, 2020.

Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. Reset-free lifelong learning with skill-space planning. *arXiv preprint arXiv:2012.03548*, 2020.

Nathan Koenig and Maja J Matarić. Robot life-long task learning from human demonstrations: a bayesian approach. *Autonomous Robots*, 41(5):1173–1188, 2017.

Murtaza Hazara, Xiaopu Li, and Ville Kyrki. Active incremental learning of a contextual skill model. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1834–1839. IEEE, 2019.

Majd Hawasly and Subramanian Ramamoorthy. Lifelong transfer learning with an option hierarchy. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1341–1346. IEEE, 2013.

Fangzhou Xiong, Zhiyong Liu, Kaizhu Huang, Xu Yang, and Hong Qiao. State primitive learning to overcome catastrophic forgetting in robotics. *Cognitive Computation*, 13(2):394–402, 2021.

Guilherme Maeda, Marco Ewerton, Takayuki Osa, Baptiste Busch, and Jan Peters. Active incremental learning of robot movement primitives. In Sergey Levine, Vincent Vanhoucke, and Ken Goldberg, editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 37–46. PMLR, 13–15 Nov 2017. URL http://proceedings.mlr.press/v78/maeda17a.html.

René Traoré, Hugo Caselles-Dupré, Timothée Lesort, Te Sun, Guanghang Cai, Natalia Díaz-Rodríguez, and David Filliat. Discorl: Continual reinforcement learning via policy distillation. *arXiv preprint arXiv:1907.05855*, 2019.

Freek Stulp. Adaptive exploration for continual reinforcement learning. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1631–1636. IEEE, 2012.

Fernando Fernández and Manuela Veloso. Probabilistic policy reuse in a reinforcement learning agent. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 720–727, 2006.

Mohit Sharma, Jacky Liang, Jialiang Zhao, Alex LaGrassa, and Oliver Kroemer. Learning to compose hierarchical object-centric controllers for robotic manipulation. *arXiv preprint arXiv:2011.04627*, 2020.

Zohreh Raziei and Mohsen Moghaddam. Adaptable automation with modular deep reinforcement learning and policy transfer. *arXiv preprint arXiv:2012.01934*, 2020.

Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181): 1–50, 2020.

Alexander Fabisch, Jan Hendrik Metzen, Mario Michael Krell, and Frank Kirchner. Accounting for task-difficulty in active multi-task robot control learning. *KI-Künstliche Intelligenz*, 29(4):369–377, 2015.

F Foglino, C Coletto Christakou, R Luna Gutierrez, and M Leonetti. Curriculum learning for cumulative return maximization. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 2308–2314. IJCAI, 2019.

Jivko Sinapov, Sanmit Narvekar, Matteo Leonetti, and Peter Stone. Learning inter-task transferability in the absence of target task samples. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 725–733, 2015.

Joseph B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.