MyoDex: A Generalizable Prior for Dexterous Manipulation

Vittorio Caggiano [1] **Sudeep Dasari** [2] **Vikash Kumar** [1]

Abstract

Human dexterity is a hallmark of motor control behaviors. Our hands can rapidly synthesize new behaviors despite the complexity (multi-articular and multi-joints, with 23 joints controlled by more than 40 muscles) of mosculoskeletal control. In this work, we take inspiration from how human dexterity builds on a diversity of prior experiences, instead of being acquired through a single task. Motivated by this observation, we set out to develop agents that can build upon previous experience to quickly acquire new (previously unattainable) behaviors. Specifically, our approach leverages multi-task learning to implicitly capture a task-agnostic behavioral priors (*MyoDex*) for human-like dexterity, using a physiologically realistic human hand model – MyoHand. We demonstrate *MyoDex*'s effectiveness in few-shot generalization as well as positive transfer to a large repertoire of unseen dexterous manipulation tasks. *MyoDex* can solve approximately 3x more tasks and it can accelerate the achievement of solutions by about 4x in comparison to a distillation baseline. While prior work has synthesized single musculoskeletal control behaviors, *MyoDex* is the first *generalizable* manipulation prior that catalyzes the learning of dexterous physiological control across a large variety of contact-rich behaviors.

Webpage: https://sites.google.com/view/myodex

1. Introduction

Human dexterity (and its complexity) is a hallmark of intelligent behavior that set us apart from other primates species (Sobinov & Bensmaia). Human hands are complex and require the coordination of various muscles to impart effective manipulation abilities. New skills do not form by simple random exploration of all possible behaviors. Instead, human motor system relies on previous experience (Heldstab et al.) to build "behavioral priors" that allow rapid skill learning (Yang et al.; Dominici et al.; Cheung et al.).

The key to learning such a prior might reside in the complexity of the actuation. Manipulation behaviors are incredibly sophisticated as they evolve in a high-dimensional search space (overactuated musculoskeletal system) populated with intermittent contact dynamics between the hands' degrees of freedom and the object. The human motor system deals with this complexity by activating different muscles as a shared unit. This phenomenon is known as a "muscle synergy" (Bizzi & Cheung). Synergies allow the biological motor system – via the modular organization of the movements in the spinal cord (Bizzi & Cheung; Caggiano et al., a) – to simplify the control problem, solving tasks by building on a limited number of shared solutions (d'Avella et al.; d'Avella & Bizzi). Those shared synergies are suggested to be the fundamental building blocks for quickly learning new and more complex motor behaviors (Yang et al.; Dominici et al.; Cheung et al.). Is it possible for us to learn similar building blocks (i.e. a behavioral prior) for general dexterous manipulation on a simulated musculoskeletal hand?

In this work, we present *MyoDex*, a behavioral prior that allows agents to quickly build dynamic, dexterous, contact-rich manipulation behaviors with multiple objects and a variety of tasks – e.g. drinking from a cup or playing with toys (see Figure 1). While we do not claim to have solved physiological dexterous manipulation, the manipulation abilities demonstrated here significantly advance the state of the art of the bio-mechanics and neuroscience fields. More specifically, our main contributions are: **(1)** We (for the first time) demonstrate control of a (simulated) musculoskeletal human hand to accomplish 57 different contact-rich skilled manipulation behaviors, despite the complexity (high degrees of freedom, third-order muscle dynamics, etc.). **(2)** We recover a task agnostic physiological behavioral prior – *MyoDex* – that exhibits positive transfer while solving unseen out-of-domain tasks. Leveraging *MyoDex*, we are able to solve 37 previously unsolved tasks. **(3)** Our ablation study reveals a tradeoff between the generality and specialization of the *MyoDex* prior. The final system is configured to maximize *generalization* and *transfer* instead of zero-shot

[1] FAIR, Meta AI [2] CMU. Correspondence to: Vittorio Caggiano <caggiano@gmail.com>, Sudeep Dasari <sdasari@andrew.cmu.edu>, Vikash Kumar <vikashplus@gmail.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

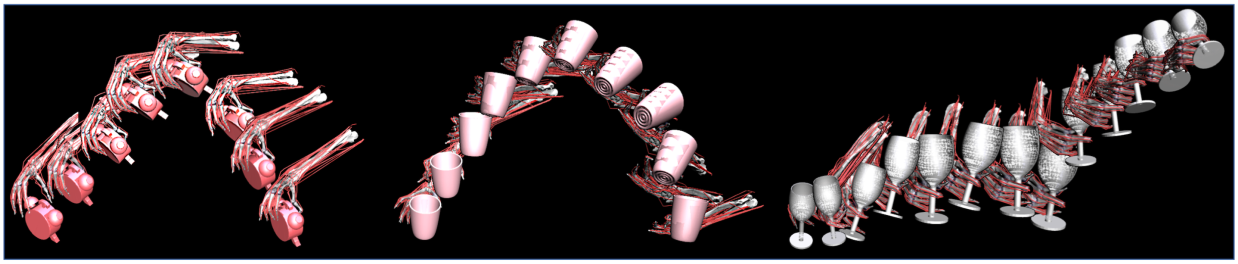

Figure 1: Contact rich manipulation behaviors acquired by *MyoDex* with a physiological *MyoHand*

out-of-the-box performance. **(4)** We demonstrate the generality of our approach by applying it to learn behaviors in other high-dimensional systems, such as multi-finger robotic hands. We construct *AdroitDex* (equivanet to *MyoDex* for the *AdroitHand* (Kumar, 2016)), which achieves 5x better sample efficiency over SOTA in the TCDM benchmark (Dasari et al., a).

2. Related Works

Dexterous manipulations has been approached independently by the biomechanics field to study the synthesis of movements of the overactuated musculoskeletal system, and roboticists looking to develop, mostly via data-driven methods, skilled dexterous robots and a-priori representations for generalizable skill learning. Here, we discuss those approaches.

Over-redundant biomechanic actuation. Musculoskeletal models (McFarland et al.; Lee et al., a; Saul et al.; Delp et al.; Seth et al.) have been developed to simulate kinematic information of the muscles and physiological joints. Nevertheless, the intensive computational needs and restricted contact forces have prevented the study of complex hand-object interactions and otherwise limited the use mostly to optimization methods. Recently, a new hand and wrist model – *MyoHand* (Caggiano et al., b; Wang et al., a) – overcomes some limitations of alternative biomechanical hand models: allows contact-rich interactions and it is suitable for computationally intensive data-driven explorations. Indeed, it has been shown that MyoHand can be trained to solve individual in-hand tasks on very simple geometries (ball, pen, cube) (Caggiano et al., b; 2022). Here, we leveraged and extended the MyoHand model to perform hand-object manouvers on a large variaty of complex realistic objects.

Behavioral synthesis. Data-driven approaches have consistently used Reinforcement Learning (RL) on joint-based control to solve complex dexterous manipulation in robotics (Rajeswaran et al.; Kumar et al.; Nagabandi et al.; Chen et al.). In order to yield more naturalistic movements, different methods have leveraged motion capture data (Merel et al., b;a; Hasenclever et al.). By means of those approaches, it has been possible to learn complex movements and athletic skills such as high jumps (Yin et al.), boxing and fencing (Won et al.) or playing basketball (Liu & Hodgins).

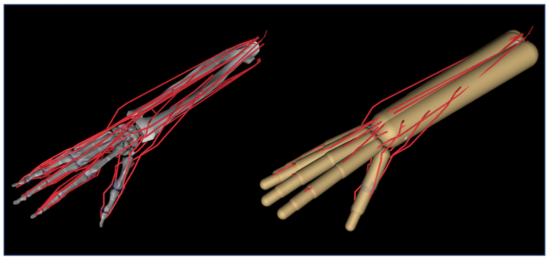

Figure 2: *MyoHand* - Musculoskeletal Hand model(Caggiano et al., b). On the left, rendering of the musculoskeletal structure illustrating bone – in gray – and muscle – in red. On the right a skin-like surface for soft contacts is overlaid to the musculoskeletal model.

In contrast to joint-based control, in biomechanical models, machine learning has been applied to muscle actuators to control movements and produce more naturalistic behaviors. This is a fundamentally different problem than robotic control as the overactuated control space of biomechanical systems leads to ineffective explorations (Schumacher et al.). Direct optimization (Wang et al., b; Geijtenbeek et al.; Al Borno et al.; Rückert & d'Avella) and deep reinforcement learning (Jiang et al.; Joos et al.; Schumacher et al.; Ikkala et al.; Caggiano et al., b; Wang et al., a; Song et al.; Park et al.) have been used to synthesize walking and running, reaching movements, in-hand manipulations, biped locomotion and other highly stylistic movements (Lee et al., c;b). Nevertheless, complex dexterous hand-object manipulations beyond in-hand object rotation (Caggiano et al., b; Berg et al., 2023) have not been accomplished so far.

Manipulation priors. Previous attempts have tried to solve

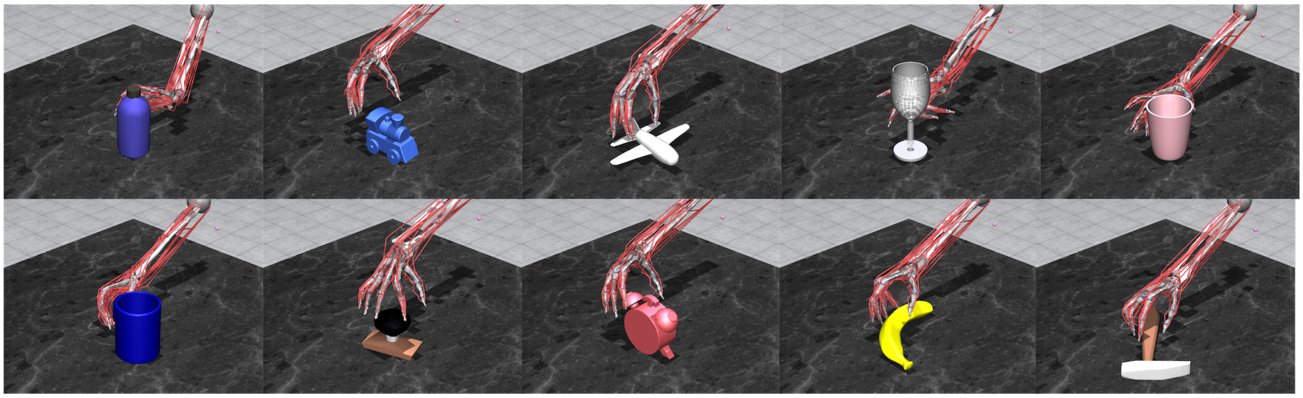

Figure 3: Task setup and a subset of *object-hand* pair from our task-set. Every task setup consisted of a tabletop environment, an object, and the MyoHand. The MyoHand was shaped with a compatible posture and positioned near an object (i.e. pre-grasp posture).

complex tasks by building priors but this approach has been limited to games and robotics. The idea of efficiently representing and utilizing previously acquired skills has been explored in robotics by looking into features across different manipulation skills e.g. Associative Skill Memories (Pastor et al.) and meta-level priors (Kroemer & Sukhatme). Another approach has been to extract movement primitives (Rueckert et al.) to identify a lower-dimensionality set of fundamental control variables that can be reused in a probabilistic framework to develop more robust movements.

Multi-task learning, where a model is trained on multiple tasks simultaneously (Caruana), has been also shown to improve the model's ability to extract features that generalize well (Zhang & Yeung; Dai et al.; Liu et al.). Multi-task reinforcement learning (RL) has been used in robotics to propose representations-based methods for exploration and generalization in games and robotics (Goyal et al.; Hausman et al.). However, training on multiple tasks can lead to negative transfer. As a consequence, performance on one task is negatively impacted by training on another task (Sun et al.). Nevertheless, it has been argued that in (over)redundant control such as the physiological one, multi-task learning might facilitate learning of generalizable solutions (Caruana). In this work, in addition to showing that nimble contact-rich manipulation using detailed physiological hand with musculoskeletal dynamics is possible, we present evidence that a generalizable physiological representation via Multi-task reinforcement learning – *MyoDex* – can be acquired and used as priors to facilitate both learning and generalization across complex contact rich dexterous tasks.

3. Overactuated Physiological Dexterity

Human hand dexterity builds on the fundamental characteristics of physiological actuation: muscles are multi-articular and multi-joints, the dynamics of the muscle are of the third order, muscles have pulling-only capabilities, and effectors have intermittent contact with objects. To further our understanding of physiological dexterity, we embed the same control challenges – by controlling a physiologically accurate musculoskeletal model of the hand (see Sec. 3.1) – in complex manipulation tasks (see Sec. 3.2).

3.1. MyoHand: A Physiologically Accurate Hand Model

In order to simulate a physiologically accurate hand model, a complex musculoskeletal hand model comprised of 29 bones, 23 joints, and 39 muscles-tendon units (Wang et al., a) – MyoHand model – implemented in the MuJoCo physics simulator (Todorov et al.) was used (see Figure 2). This hand model has previously been shown to exhibit a few dexterous *in-hand* manipulation tasks (Caggiano et al., b), which makes it a good candidate for our study seeking generalization in dexterous manipulation.

We extended the MyoHand model to include translations and rotations at the level of the shoulder. We limited the translation on the frontal (range between $[-0.07,\ 0.03]$) and longitudinal (range between $[-0.05,\ 0.05]$) axis to support the natural shoulder and wrist rotation (elbow is considered maximally extended i.e. a straight arm). For the rest of the paper we will refer to the whole system as *MyoHand*.

3.2. Dexterous Behaviors Studied

In this study, we need a large variability of manipulations to explore the generality of our method against a wide range of solutions, hence it was important to include 1) objects with different shapes, and 2) complexity in terms of desired behaviors requiring simultaneous effective coordination of finger, wrist, as well as arm movements.

Our task set *MyoDM* (inspired by TCDM benchmarks (Dasari et al., b)) is implemented in the MuJoCo physics

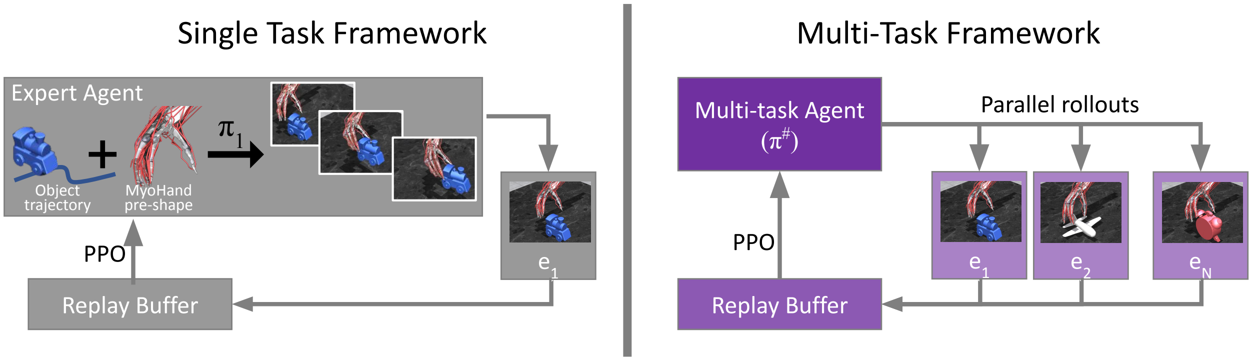

Figure 4: Learning Frameworks. Left - Single Task Framework: policies were obtained by training policies to solve the individual tasks. Right - Multi-task framework: A single policy (*MyoDex*) was obtained by learning all tasks at once.

engine (Todorov et al.) and consists of 33 objects and 57 different behaviors. Every task setup (see Figure 3) consists of a tabletop environment, an object from the ContactDB dataset (Brahmbhatt et al.), and the MyoHand.

Dexterous manipulation is often posed as a problem of achieving the final desired configuration of an object. In addition to the final posture, in this study, we are also interested in capturing the detailed temporal aspect of the entire manipulation behavior. Tasks like drinking, playing, or cyclic movement like hammering, sweeping, etc., that are hard to capture simply as goal-reaching, can be handled by our formulation (Sec. 4) and are well represented in the *MyoDM*.

The tasks considered in *MyoDM* entail a diverse variety of object manipulation (relocations+reorientations) behaviors requiring synchronized coordination of arm, wrist, as well as in-hand movements to achieve the desired object behaviors involving simultaneous translation as well as rotation (average ± std, $28° \pm 21°$). The range of motions of the shoulder with fixed elbow alone is not sufficient to enable the entire range of desired object rotations without involving in-hand and wrist maneuvers. The angle between the palm and object ranges upwards of $20°$ in our final acquired behaviors. The wrist is one of the most complex joints to control because it is affected simultaneously by the balanced activation of more than 20 muscles whose activations also control finger movements. Careful maneuvering of objects within the hand requires simultaneous synchronization of numerous antagonistic finger muscle pairs, failing which leads to loss of object controllability; highlighting the complexities of controlling a physiological musculoskeletal hand during these complex manipulations.

4. Learning Controllers for Physiological Hands

In this section, we discuss our approach to build agents that can learn contact-rich manipulation behaviors and generalize across tasks.

4.1. Problem formulation

A manipulation task can be formulated as a Markov Decisions Process (MDP) (Sutton & Barto) and solved via Reinforcement Learning (RL). In RL paradigms, the Markov decision process is defined as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \rho, \gamma)$, where $\mathcal{S} \subseteq \mathbb{R}^n$ and $\mathcal{A} \subseteq \mathbb{R}^m$ represents the continuous state and action spaces respectively. The unknown transition dynamics are described by $s' \sim \mathcal{T}(\cdot|s, a)$. $\mathcal{R} : \mathcal{S} \to [0, R_{\max}]$, denotes the reward function, $\gamma \in [0, 1)$ denotes the discount factor, and and ρ the initial state distribution. In RL, a policy is a mapping from states to a probability distribution over actions, i.e. $\pi : \mathcal{S} \to P(\mathcal{A})$, which is parameterized by θ. The goal of the agent is to learn a policy $\pi_\theta(a|s) = argmax_\theta[J(\pi, \mathcal{M})]$, where $J = \max_\theta \mathbb{E}_{s_0 \sim \rho(s), a \sim \pi_\theta(a_t|s_t)}[\sum_t R(s_t, a_t)]$

4.2. Learning Single-Task Controllers

Single task agents. The single task agents are tasked with picking a series of actions ($[a_0, a_1, ..., a_T]$), in response of the evolving states ($[s_0, s_1, ..., s_T]$) to achieve their corresponding object's desired behavior $\hat{X}_{object} = [\hat{x}_0, ..., \hat{x}_T]$.

We adopt a standard RL algorithm *PPO* (Schulman et al.) to acquire a goal-conditioned policy $\pi_\theta(a_t|s_t, \hat{X}_{object})$ as our single task agents. Details on state, actions, rewards, etc are provided in Section 5. Owing to the third-order non-linear actuation dynamics and high dimensionality of the search space, direct optimization on $\mathcal{M}$ leads to no meaningful behaviors.

Pre-grasps implicitly incorporate information pertaining to the object and its associated affordance with respect to the desired task (Jeannerod; Santello et al.). We leveraged (Dasari et al., b)'s approach of leveraging pregrasp towards dexterous manipulation with robotic (Adroit (Kumar, 2016)) hand and extend it towards MyoHand. The approach uses the hand-pose directly preceding the initiation of contact with an object i.e. a proxy to pre-grasp, to guide search in the high dimensional space in which dexterous behaviors evolve. This approach yeilds a set of single-task expert agents π_i with $i \in I$ where I is the set of tasks (see Figure 4-left).

4.3. Framework for Multi-Task Physiological Learning

Multi-task agent. Ideally, an agent would be able to solve multiple tasks using a goal-conditioning variable. Thus, we additionally train a single agent to solve a subset of tasks in parallel (see Figure 4-right). This approach proceeds in a similar fashion as the single-task learner, but agent's experiences are sampled from the multiple tasks in *parallel*. All other details of the agent $\pi_\theta^{\#}(a_t|s_t, \hat{X}_{object})$ (e.g. hyperparameters, algorithm, etc.) stay the same.

Similar to the single task agents, we encode manipulation behaviors in terms of goal-conditioned policies $\pi_\theta(a_t|s_t, \hat{X}_{object})$ and employ a standard implementation of the PPO (Schulman et al.) from Stable-Baselines (Raffin et al.) and pre-grasp informed formulation from (Dasari et al., b)'s to guide the search for our multi-task agents as well. See Section A.4.2 for details. The hyperparameters were kept the same for all tasks (see Appendix Table A.1).

5. Task Details

Next, we provide details required to instantiate our *MyoDM* task suite–

State Space. The state vector $s_t = \{\phi_t, \dot{\phi}_t, \psi_t, \dot{\psi}_t, \tau_t\}$ consisted of ϕ a 29-dimensional vector of 23 hand and 6 arm joints and velocity $\dot{\phi}$, and object pose ψ and velocity $\dot{\psi}$. In addition, positional encoding τ (Vaswani et al.), used to mark the current simulation timestep, was appended to the end of the state vector. This was needed for learning tasks with cyclic motions such as hammering.

Action Space. The action space a_t was a 45-dimensional vector that consists of continuous activations for 39 muscles of the wrist and fingers (to contract muscles), together with 3D translation (to allow for displacement in space), and 3D rotation of the shoulder (to allow for a natural range of arm movements).

Reward Function. The manipulation tasks we consider involved approaching the object and manipulating it in free air after lifting it off a horizontal surface. The hand interacts with the object adjusting its positions and orientation ($X = [x_0, ..., x_T]$) for a fixed time horizon. Similar to (Dasari et al., b), this is translated into an optimization problem where we are searching for a policy that is conditioned on desired object trajectory $\hat{X} = [\hat{x}_0, ..., \hat{x}_T]$ and optimized using the following reward function:

$$R(x_t, \hat{x}_t) := \lambda_1 exp\{-\alpha \|x_t^{(p)} - \hat{x}_t^{(p)}\|_2 - \beta |\angle x_t^{(o)} - \hat{x}_t^{(o)}|\} + \lambda_2 \mathbb{1}\{lifted\} - \lambda_3 \left\|\overline{m}_t\right\|_2 \quad (1)$$

where $\angle$ is the quaternion angle between the two orientations, $\hat{x}_t^{(p)}$ is the desired object position, $\hat{x}_t^{(o)}$ is the desired object orientation, $\mathbb{1}\{lifted\}$ encourages object lifting, and $\overline{m}_t$ the is overall muscle effort.

Progress metrics. To effectively capture the temporal behaviors, we treat dexterous manipulation as a task of realizing desired object trajectories ($\hat{X}$). To capture temporal progress, similar to (Dasari et al., b), we use three metrics to measure task performance. The *success metric*, $S(\hat{X})$ reports the fraction of time steps where object error is below a $\epsilon = 1cm$ threshold. It is defined as: $S(\hat{X}) = \frac{1}{T}\sum_{t=0}^{T} \mathbb{1} \left\|x_t^{(p)} - \hat{x}_t^{(p)}\right\|_2 < \epsilon.$ The *object error metric* $E(\hat{X})$ calculates the average Euclidean distance between the object's center-of-mass position and the desired position from the desired trajectory: $E(\hat{X}) = \frac{1}{T}\sum_{t=0}^{T} \left\|x_t^{(p)} - \hat{x}_t^{(p)}\right\|_2$. In addition, we also used the *object orientation metric*: $O(\hat{X}) = \frac{1}{T}\angle_{t=0}^{T}(x_t^{(o)} - \hat{x}_t^{(o)})$ [1].

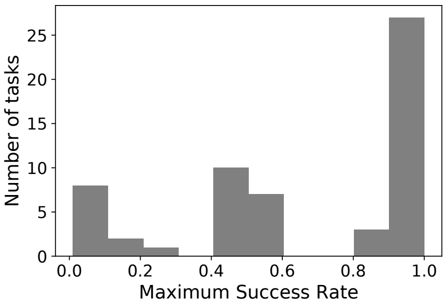

Figure 5: Distribution of single task solutions. Distribution of maximums success rate for single-task solutions on 57 different tasks. Only 32 out of 57 tasks i.e. 56%, were solved with a success rate above 80%. Training performed over $12.5k$ iterations.

[1] For interpretability, we often omit orientations because center-of-mass error and orientation error were highly correlated in practice i.e. Pearson-correlation > 0.785

6. Results

First, we study if we can solve the *MyoDM* task set, one task at a time (see Sec. 6.1). Next, we illustrate that our *MyoDex* representation can be used as *a prior* for accelerating learning novel, out-of-domain tasks (see Sec. 6.2). Finally, we present a series of ablation studies to understand various design choices of our approach (see Sec. 6.4).

6.1. Learning Expert Solutions for Single-Task Setting

We begin by asking, is it possible to learn a series of complex dexterous manipulation behaviors (see Sec. 3.2) using a MyoHand? Our single-task learning framework is applied to solve a set of 57 *MyoDM* tasks independently, without any object or task-specific tuning (see Table A.1). The resulting "expert policies" were able to properly manipulate only a subset of those objects, while moving and rotating them to follow the target trajectory (see Figure 1 for a sequence of snapshots). This was quantified by using 2 metrics (Sec. 5): a Success Metric and an Error Metric. Our single-task framework achieves an average success rate of 66% solving 32 out of 57 tasks (see Fig. 5 and experts in Fig. A.3) and an average (ecludean distance) error of 0.021. We encourage readers to check our project website for videos and further qualitative analysis of the learned behaviors.

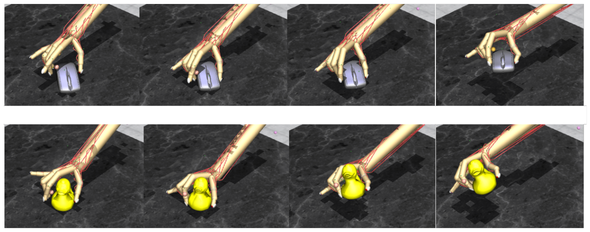

Figure 6: Zero-shot generalization. *MyoDex* successfully initiated manipulations on new objects and trajectories. Hand rendering includes skin-like contact surfaces (see Fig. 2)

6.2. Accelerating Out-of-Domain Learning via *MyoDex*

The single-task framework was not able to solve all task in our task set, even individually which further establishes complexity of behavior acquisition with high DoF MyoHand and the difficulty of our *MyoDM* task set. Furthermore, it creates controllers that can only function within a specific scenario/task. Next, we will demonstrate that by simultaneously training on multiple tasks during the reinforcement learning loop we can achieve a *MyoDex* prior that can overcome single-task limitations. *MyoDex* is a prior that can be fine-tuned to solve a larger amount of tasks. In addition, a single multi-task policy based on training *MyoDex* at convergence can solve multiple tasks.

For building the *MyoDex* prior, we consider a subset of 14 *MyoDM* tasks with a large variability of object and movements (see Sec. 6.4.3 for the effects of task choice) and we trained one policy to solve all the set of tasks at once. We stopped the training at $12.5k$ iterations (at the beginning of the error plateau – see Figure A.1). At this iteration, we tested potential *zero-shot* generalization capabilities with the MyoHand positioned near a novel object with a compatible posture and conditioned on a new task trajectory. While the policy was not able to zero-shot solve these new tasks (success rate $\leq$ 10%), we do observe (see Fig. 6) that the hand can succesfully grasp and lift the unseen objects. This leads us to believe that the *MyoDex* representation can be used as a *prior* for accelerating transfer learning.

However, this is not the only way to accomplish a general multi-task representation. An established baseline is a student-teacher distillation (see Sec. A.1), which trains a single student policy to imitate the 14 expert policies (from prior experiments) via behavior cloning.

We fine-tune both the *MyoDex* and the student policy on the remaining out-of-domain set of 43 *MyoDM* tasks (using single-task RL) for additional iterations. Figure 7 presents learning curves for the fine-tuned models based on *MyoDex*, fine-tuned student baselines, and trained (from scratch) single-task expert policies in terms of success rate and errors, respectively. Note how the *MyoDex* based policy is able to learn the tasks significantly faster than either the baseline or the single-task policies. Among the solved out-of-domain tasks, *MyoDex* based policies were about 4x faster than student based policy ($1.9k$ vs $7.7k$), and approximately 3x fastern than single-task expert policy ($1.9k$ vs $5.7k$, Table 1). Additionally, it achieves a *higher overall task performance in comparision to the single-task experts*, which plateau at a significantly lower success rate, likely due to exploration challenges. Table 1 shows this trend in extra detail. The *MyoDex* representation allows to solve more tasks (37 vs 22, see Table 1 and Table A.2) and achieve a higher overall success rate (0.89 vs 0.69) than the single-task expert, which in turn outperforms the student baseline. This leads us to conclude that the *MyoDex* representation can act as a generalizable prior for learning dexterous manipulation policies on a musculoskeletal MyoHand. It is both able substantially accelerate learning new tasks, and indeed leads to a *stronger* transfer to new tasks.

Based on	Solved	Success	Iter. to solve
Expert	51% (22/43)	0.69 ± 0.30	$5.7k \pm 1.5k$
Student	30% (13/43)	0.54 ± 0.35	$7.7k \pm 1.9k$
MyoDex	**86% (37/43)**	$\mathbf{0.89 \pm 0.25}$	$\mathbf{1.9k \pm 2.1k}$

Table 1: *MyoDex* transfer statistics on unseen (43) tasks – *Solved* indicates the percentage (ratio) of solved tasks (success $\geq$ 80%). *Success* indicates the success metric stats on all 43 tasks at $12.5k$ iterations. *Iter. to solve* indicates the stats on min iterations required by the solved task to achieve $\geq$ 80% success. Values are expressed as average $\pm$ std.

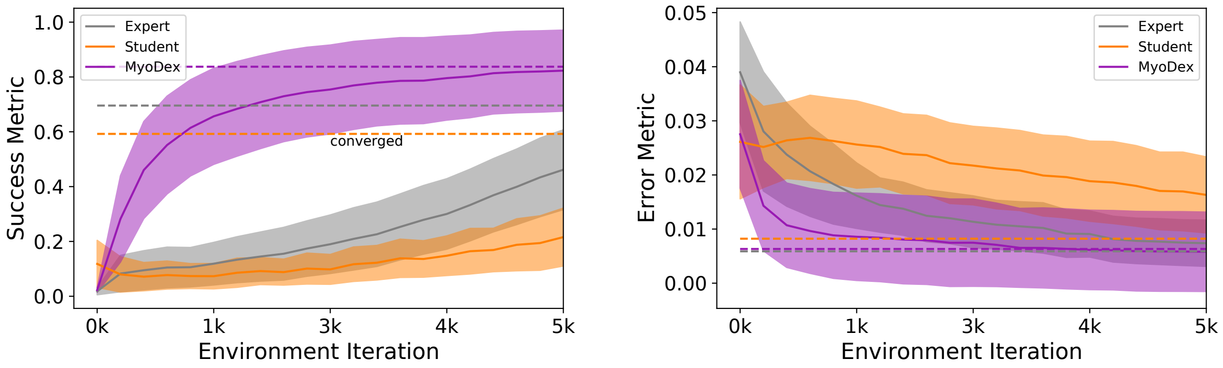

Figure 7: Fine-tuning on 43 Out-of-domain tasks. Metrics until $5k$ iterations of the fine tuning of 43 out-of-domain tasks. Convergence is assumed at $12.5k$ iterations. Left - Success Metric. Right - Error Metric. Continuous lines show average and shaded areas the standard deviation of success and error metrics. The dashed line represents the value at convergence i.e. $12.5k$ iterations.

6.3. Multi-Task Learning with *MyoDex*

Additionally, *MyoDex* can also be used to recover one single policy that can solve multiple tasks. We compared the results of the *MyoDex* training at convergence against the student policy (from the distillation of experts) on the same set of 14 tasks. See a summary of the results Figure A.2. The converged *MyoDex* based policy's success rate improves by $>$ 2x over the student policy. We present an explanation in Section 8 of why distilling from experts that have acquired incompatible behaviors in an over-redundant musculoskeletal system fails at learning multi-task policies. Indeed, expert policies found a local solution that does not help to learn other tasks e.g. experts used as a-priori do not help to fine-tune other tasks (see Fig. A.5). In contrast, our multi-task framework avoids this pitfall, since it simultaneously learns one policy without any implicit bias, and can reach similar levels as reached by individual experts in isolation.

6.4. *MyoDex* Ablation Study

The previous set of experiments demonstrated that *MyoDex* contains generalizable priors for dexterous manipulation. The following ablation study investigates how changing the number of pre-training iterations as well as the number of tasks during pre-training affect the *MyoDex*'s capabilities.

6.4.1. Effects of iterations on the *MyoDex* representation

In our experiment, the multi-task policy at $12.5k$ iterations is defined as the *MyoDex* prior. At this number of iterations, the policy was able to achieve $\sim 35\%$ success rate (see Fig. A.1). This solution provided both few-shot learning (task solved within the first environment iteration) and able to solve most of the *MyoDM* set of 57 tasks. Here, in order to probe the sensitivity of *MyoDex* prior to the stage of learning at which the representation is extracted, we compared *MyoDex* against representations obtained earlier i.e. $2.5k$ and $7.5k$, and one later i.e. $37.5k$ stages of learning. Figure 8 shows the results on the fine-tuning of all the 57 tasks for the 4 different representations. Early representations are slower but, with enough iterations, they are able to solve almost all tasks (98% (56 / 57) and 91% (52 / 57) respectively for the representations at $2.5k$ and $7.5k$). Conversely, later representations, show few-shot learning (10 tasks) but they are able to learn only a reduced amount of tasks (61% (35 / 57)). Hence, *MyoDex* trained at $12.5k$ iterations strikes a balance, facilitating fast initial learning (including few-shots) while being general enough to support a diverse collection of out-of-domain tasks (see Figure 8).

Another way to look at the effect of learning and generalizable solutions over the iterations is to look to muscle synergies as they express the amount of muscle co-contraction shared across tasks. In our study, we utilized the concept of Variance Accounted For (VAF, see Sec. A.4) to quantify the number of synergies needed to reconstruct the needed muscle activations to solve the task. Higher VAF achieved with fewer muscle synergies indicates that it is possible to use fewer combinations of muscle co-contractions to generate the needed muscle activations. Our findings indicate that early on in the training process (i.e., around 2.5k iterations, see Figure A.8), a substantial number of synergies (more than 12) is needed to achieve a high level of signal reconstruction. This suggests that while the policy is capable of discovering some solutions in the muscle space, synergies are not able to cover all the variability of the signal. Indeed, this representation helps to overcome some local minima hence it is particularly well-suited for facilitating transfer to new tasks.

Around 12.5k iterations, we observed a peak in the capacity of fewer synergies to account for most of the signal (see Figure A.8). At this point we have identified solutions in the muscle space that are highly reusable across multiple tasks.

However, at 37.5k iterations, we found that a greater number of synergies were required to explain most of the original signals. This indicates that specialized co-contractions are emerging to address specific tasks demands. While these synergies are effective at solving similar tasks with few or zero shots, their specialization may limit their ability to tackle dissimilar tasks.

Overall, our results suggest that our representation of synergies is a powerful tool for facilitating transfer learning, especially in the early stages of training when more generalized solutions are required. As training progresses, the emergence of specialized co-contractions enables efficient learning and transfer to similar tasks. Still, with even more training, specialized solutions are developed.

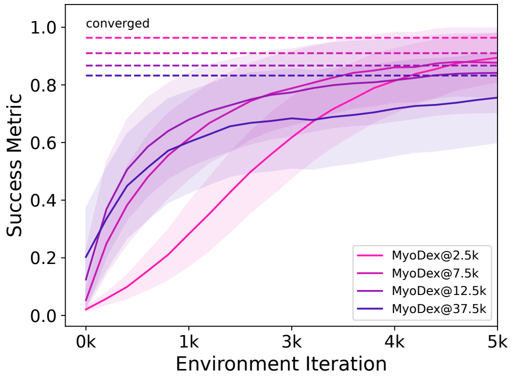

Figure 8: Fine-tuning from representations obtained at different iterations. Fine-tuning from representations obtained earlier i.e. $2.5k$ and $7.5k$ iterations, *MyoDex* i.e. $12.5k$ iterations, and later i.e. $37.5k$ iterations. Earlier representations show no few-shot generalization but better coverage with 56 out of 57 tasks solved, while later representations show few-shot generalizations but have less coverage with 35 out of 57 tasks solved. The continuous line represents the average and the shaded area is the standard deviation of the success metrics. The dashed line represents the value at convergence i.e. 12.5k iterations.

6.4.2. Effect of the number of environments on *MyoDex* training

In the above experiment, we showed the *MyoDex* representation based on 14 environments. An analysis showing the effect of multi-task learning on environment diversity illustrates that the use of 14 environments represented a balance between trainign the multi-task policy effectively and transfer/generalization ability it possses. We compared *MyoDex* trained on 6, 14, and 18 environments at $12.5k$ iterations and tested on a set of 39 new environments. *MyoDex* based on 6 and 18 environments leads to lower performance with respect to 14 environments both in terms of success rate and the number of solved environments (see Table 2).

Based on	Success	Solved
*MyoDex*6	0.78 ± 0.32	72% $(28/39)$
***MyoDex*14**	$\mathbf{0.92 \pm 0.21}$	$\mathbf{95\%}$ $\mathbf{(37/39)}$
*MyoDex*18	0.91 ± 0.2	87% $(34/39)$

Table 2: Fine-tuning statistics based on different *MyoDex* priors. *MyoDex* trained with different environments as priors and fine-tuned on 39 environments. Results reported in terms of average and standard deviation of success and percentage of solved tasks i.e. $\geq 80\%$.

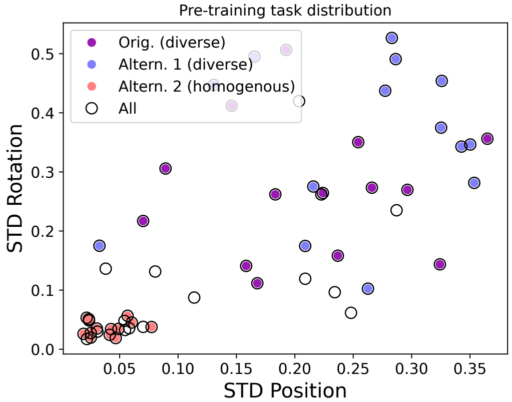

Figure 9: Pre-Training task distribution. The distributions of our task collection in terms of its variability (standard deviation - STD). Represented on each axes are the STD of the absolute positional (X-Axis) and rotational (Y-axis) displacements from the respective initial object poses in the desired object trajectories in our task set. In circle are all the 57 tasks involved in our study In pink [Orig.(diverse)] are the original tasks used for training MyoDex. In blue [Altern.1(diverse)] is a new task set we use for training an alternate instance of MyoDex prior used in ablation studies.

6.4.3. How Training Tasks Affect *MyoDex*

The choice of objects and tasks to train *MyoDex* can significantly impact the effectiveness of the representation. We study the effect of pre-training task distribution on the effectiveness of MyoDex priors. We selected two new task sets. First, a *diverse* tasks collection – *MyoDex Alt Diverse* (Figure 9 in blue) with the same similar attributes of the original dataset (in pink). Second, a *homogenous* task collection – *MyoDex Alt Homogenous* (Figure 9 in red) – with tasks with little motion variance (e.g. mostly lifting). We found that *MyoDex Alt Diverse* – trained on the alternative diverse tasks – was able to improve performance over time, while *MyoDex Alt Homogenous* – trained on the alternative homogenous tasks – had its performance plateau early on during training (see Figure A.7). Indeed, when used for transfer on new tasks, *MyoDex Alt Diverse* is able to match the original *MyoDex* performance, while *MyoDex Alt Homogenous* does not (see Figure 10). This shows that the variety of manipulation/tasks in the pretraining is fundamental to achieve

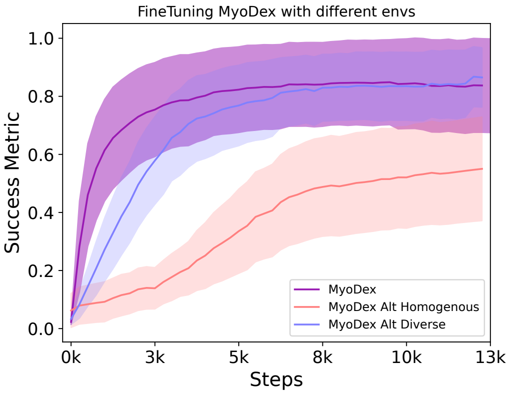

Figure 10: Effect of pre-training task distribution on *MyoDex* performance. *MyoDex Alt Diverse* (trained on tasks of similar diversity – in blue) is able to better match the original *MyoDex* performance in comparision to *MyoDex Alt Homogenous* (trained on homogenous tasks collection).

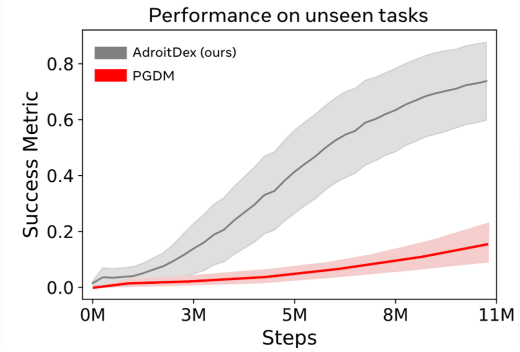

Figure 11: Fine-tuning a generalizable representation on Adroit subtasks: *AdroitDex*. A general representation of manipulation on the same 14 tasks used for trainign *MyoDex* was finetuned on 34 unseen tasks on the TCDM benchmark (Dasari et al., a). Curves shows average (continuous) and std (shaded area). *AdroitDex* beats previously reported SOTA on TCDM benchmarks while being 5x more sample efficient.

high performance in a larger set of downstream tasks and *MyoDex* is not sensitive to the specific choice of tasks.

6.5. Extension to other high dimensional system

To further investigate the applicability of our approach to other high dimensional systems, we set out to build a generalizable representation for the robotic *Adroit Hand* (Rajeswaran et al.) commonly studied in robot learning. Adroit is a 24 degree-of-freedom (DoF) modified shadow hand with 4 extra DoF at the distal joint. Following the approach behind *MyoDex*, a general representation of manipulation prior - *AdroitDex* - was obtained. We use the same 14 tasks that we used for training *MyoDex*. In the Figure 11 we show the performance of *AdroitDex* on 34 unseen tasks on the TCDM benchmark (Dasari et al., a). *AdroitDex* achieves a success rate of 74.5% in about 10M iteration steps, which is approvimately 5x faster than the PGDM baseline (Dasari et al., a), which needed 50M iteration steps to achieve the same result (see Figure 11).

7. Conclusion

In this manuscript, we learn skilled dexterous manipulation of complex objects on a musculoskeletal model of the human hand. In addition, by means of joint multi-task learning, we showed that it is possible to extract generalizable representations (*MyoDex*) which allow faster fine-tuning on out-of-domain tasks and multi-task solutions. Ultimately, this study provides strong bases for how physiologically realistic hand manipulations can be obtained by pure exploration via Reinforcement Learning i.e. without the need for motion capture data to imitate specific behavior.

8. Discussion on the role of Synergies

Why does *MyoDex* help the overactuated musculoskeletal system to solve multiple tasks? If we look at the coordination of muscle activations – muscle synergies (see Appendix A.4) – we notice that *MyoDex* shows a larger number of similar activations (see Figure A.4) vs experts/distilled policies. This is because the expert solutions find one mode/solution to solve a task that does not incorporate information from other tasks. Naiive distillation propogates this effect to the student policy. In contrast, *MyoDex* learns to coordinate muscle contraction. Indeed, fewer muscle coordination/synergies seem to explain most of the behavior (see Figure A.8, at 12.5K iterations). All in all, those observations are in line with the neuroscience literature where muscle synergies have been suggested as the physiological substrate to obtain faster and more effective skill transfer (Yang et al.; Cheung et al.; Dominici et al.; Berger et al.).

9. Limitations and Future work

While we demonstrated that *MyoDex* can produce realistic behavior without human data, one important limitation is understanding and matching the results with physiological data. Indeed, our exploration method via RL produced only one of the very high dimensional combinations of possible ways that a human hand could hypothetically grab and manipulate an object. For example, there are several valid ways to hold a cup e.g. by using the thumb and one or multiple fingers. Although our investigation points us in the right direction regarding the physiological feasibility of the result, these findings have yet to be properly validated with clinical data and user studies. Future works will need to consider the ability to synthesize new motor behaviors while simultaneously providing muscle validation.

References

Al Borno, M., Vyas, S., Shenoy, K. V., and Delp, S. L. High-fidelity musculoskeletal modeling reveals that motor planning variability contributes to the speed-accuracy tradeoff. 9:e57021. ISSN 2050-084X. doi: 10.7554/eLife.57021. URL https://doi.org/10.7554/eLife.57021.

Berg, C., Caggiano, V., and Kumar, V. Sar: Generalization of dexterity via synergistic action representation. https://sites.google.com/view/sar-rl/home, 2023.

Berger, D. J., Gentner, R., Edmunds, T., Pai, D. K., and d'Avella, A. Differences in adaptation rates after virtual surgeries provide direct evidence for modularity. 33(30):12384–12394. ISSN 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.0122-13.2013. URL https://www.jneurosci.org/content/33/30/12384. Publisher: Society for Neuroscience Section: Articles.

Bizzi, E. and Cheung, V. C. The neural origin of muscle synergies. 7. ISSN 1662-5188. doi: 10.3389/fncom.2013.00051. URL https://www.frontiersin.org/article/10.3389/fncom.2013.00051.

Brahmbhatt, S., Ham, C., Kemp, C. C., and Hays, J. ContactDB: Analyzing and predicting grasp contact via thermal imaging. pp. 8701–8711.

Caggiano, V., Cheung, V. C. K., and Bizzi, E. An optogenetic demonstration of motor modularity in the mammalian spinal cord. 6(1):35185, a. ISSN 2045-2322. doi: 10.1038/srep35185. URL https://doi.org/10.1038/srep35185.

Caggiano, V., Wang, H., Durandau, G., Sartori, M., and Kumar, V. MyoSuite – a contact-rich simulation suite for musculoskeletal motor control, b. URL http://arxiv.org/abs/2205.13600.

Caggiano, V., Wang, H., Durandau, G., Song, S., Tassa, Y., Sartori, M., and Kumar, V. Myochallenge: Learning contact-rich manipulation using a musculoskeletal hand. https://sites.google.com/view/myochallenge, 2022.

Caruana, R. Multitask learning.

Chen, T., Xu, J., and Agrawal, P. A system for general in-hand object re-orientation.

Cheung, V. C. K., Cheung, B. M. F., Zhang, J. H., Chan, Z. Y. S., Ha, S. C. W., Chen, C.-Y., and Cheung, R. T. H. Plasticity of muscle synergies through fractionation and merging during development and training of human runners. 11(1):4356. ISSN 2041-1723. doi: 10.1038/s41467-020-18210-4. URL https://doi.org/10.1038/s41467-020-18210-4.

Dai, J., He, K., and Sun, J. Instance-aware semantic segmentation via multi-task network cascades. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3150–3158. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.343. URL http://ieeexplore.ieee.org/document/7780712/.

Dasari, S., Gupta, A., and Kumar, V. Learning dexterous manipulation from exemplar object trajectories and pregrasps, a. URL https://arxiv.org/abs/2209.11221.

Dasari, S., Gupta, A., and Kumar, V. Learning dexterous manipulation from exemplar object trajectories and pregrasps. In *IEEE International Conference on Robotics and Automation 2023*, b.

d'Avella, A. and Bizzi, E. Shared and specific muscle synergies in natural motor behaviors. 102(8):3076–3081. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.0500199102. URL https://pnas.org/doi/full/10.1073/pnas.0500199102.

d'Avella, A., Saltiel, P., and Bizzi, E. Combinations of muscle synergies in the construction of a natural motor behavior. 6(3):300–308.

Delp, S. L., Anderson, F. C., Arnold, A. S., Loan, P., Habib, A., John, C. T., Guendelman, E., and Thelen, D. G. OpenSim: Open-source software to create and analyze dynamic simulations of movement. 54(11):1940–1950. doi: 10.1109/TBME.2007.901024.

Dominici, N., Ivanenko, Y. P., Cappellini, G., d'Avella, A., Mondì, V., Cicchese, M., Fabiano, A., Silei, T., Paolo, A. D., Giannini, C., Poppele, R. E., and Lacquaniti, F. Locomotor primitives in newborn babies and their development. 334(6058):997–999. doi: 10.1126/science.1210617. URL https://www.science.org/doi/abs/10.1126/science.1210617.

Geijtenbeek, T., van de Panne, M., and van der Stappen, A. F. Flexible muscle-based locomotion for bipedal creatures. 32(6):1–11. ISSN 0730-0301, 1557-7368. doi: 10.1145/2508363.2508399. URL https://dl.acm.org/doi/10.1145/2508363.2508399.

Goyal, A., Islam, R., Strouse, D., Ahmed, Z., Botvinick, M., Larochelle, H., Bengio, Y., and Levine, S. InfoBot: Transfer and exploration via the information bottleneck. URL http://arxiv.org/abs/1901.10902.

Hasenclever, L., Pardo, F., Hadsell, R., Heess, N., and Merel, J. CoMic: Complementary task learning & mimicry for reusable skills. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Hausman, K., Springenberg, J. T., Wang, Z., Heess, N., and Riedmiller, M. LEARNING AN EMBEDDING SPACE FOR TRANSFERABLE ROBOT SKILLS.

Heldstab, S. A., Isler, K., Schuppli, C., and van Schaik, C. P. When ontogeny recapitulates phylogeny: Fixed neurodevelopmental sequence of manipulative skills among primates. 6(30):eabb4685. doi: 10.1126/sciadv.abb4685. URL https://www.science.org/doi/10.1126/sciadv.abb4685. Publisher: American Association for the Advancement of Science.

Ikkala, A., Fischer, F., Klar, M., Bachinski, M., Fleig, A., Howes, A., Hämäläinen, P., Müller, J., Murray-Smith, R., and Oulasvirta, A. Breathing life into biomechanical user models. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, UIST '22. Association for Computing Machinery. ISBN 978-1-4503-9320-1. doi: 10.1145/3526113.3545689. URL https://doi.org/10.1145/3526113.3545689. event-place: Bend, OR, USA.

Jain, D., Li, A., Singhal, S., Rajeswaran, A., Kumar, V., and Todorov, E. Learning deep visuomotor policies for dexterous hand manipulation. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3636–3643. doi: 10.1109/ICRA.2019.8794033.

Jeannerod, M. *The neural and behavioural organization of goal-directed movements*. Clarendon Press/Oxford University Press.

Jiang, Y., Van Wouwe, T., De Groote, F., and Liu, C. K. Synthesis of biologically realistic human motion using joint torque actuation. 38(4):1–12. ISSN 0730-0301, 1557-7368. doi: 10.1145/3306346.3322966. URL https://dl.acm.org/doi/10.1145/3306346.3322966.

Joos, E., Péan, F., and Goksel, O. Reinforcement learning of musculoskeletal control from functional simulations. URL http://arxiv.org/abs/2007.06669.

Kroemer, O. and Sukhatme, G. S. Learning relevant features for manipulation skills using meta-level priors. URL http://arxiv.org/abs/1605.04439.

Kumar, V. *Manipulators and Manipulation in high dimensional spaces*. PhD thesis, 2016.

Kumar, V., Todorov, E., and Levine, S. Optimal control with learned local models: Application to dexterous manipulation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 378–383. doi: 10.1109/ICRA.2016.7487156.

Lee, J. H., Asakawa, D. S., Dennerlein, J. T., and Jindrich, D. L. Finger muscle attachments for an OpenSim upper-extremity model. 10(4):e0121712, a. ISSN 1932-6203. doi: 10.1371/journal.pone.0121712. URL https://dx.plos.org/10.1371/journal.pone.0121712.

Lee, S., Park, M., Lee, K., and Lee, J. Scalable muscle-actuated human simulation and control. 38(4):1–13, b. ISSN 0730-0301, 1557-7368. doi: 10.1145/3306346.3322972. URL https://dl.acm.org/doi/10.1145/3306346.3322972.

Lee, S., Yu, R., Park, J., Aanjaneya, M., Sifakis, E., and Lee, J. Dexterous manipulation and control with volumetric muscles. 37(4):1–13, c. ISSN 0730-0301, 1557-7368. doi: 10.1145/3197517.3201330. URL https://dl.acm.org/doi/10.1145/3197517.3201330.

Liu, L. and Hodgins, J. Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. 37(4):1–14. ISSN 0730-0301, 1557-7368. doi: 10.1145/3197517.3201315. URL https://dl.acm.org/doi/10.1145/3197517.3201315.

Liu, S., Johns, E., and Davison, A. J. End-to-end multi-task learning with attention. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1871–1880. IEEE. ISBN 978-1-72813-293-8. doi: 10.1109/CVPR.2019.00197. URL https://ieeexplore.ieee.org/document/8954221/.

McFarland, D. C., Binder-Markey, B. I., Nichols, J. A., Wohlman, S. J., de Bruin, M., and Murray, W. M. A musculoskeletal model of the hand and wrist capable of simulating functional tasks. pp. 2021.12.28.474357. doi: 10.1101/2021.12.28.474357. URL http://biorxiv.org/content/early/2021/12/30/2021.12.28.474357.abstract.

Merel, J., Hasenclever, L., Galashov, A., Ahuja, A., Pham, V., Wayne, G., Teh, Y. W., and Heess, N. Neural probabilistic motor primitives for humanoid control, a. URL http://arxiv.org/abs/1811.11711.

Merel, J., Tassa, Y., TB, D., Srinivasan, S., Lemmon, J., Wang, Z., Wayne, G., and Heess, N. Learning human behaviors from motion capture by adversarial imitation, b. URL http://arxiv.org/abs/1707.02201.

Nagabandi, A., Konoglie, K., Levine, S., and Kumar, V. Deep dynamics models for learning dexterous manipulation. URL `http://arxiv.org/abs/1909.11652`.

Park, J., Min, S., Chang, P. S., Lee, J., Park, M., and Lee, J. Generative GaitNet. URL `http://arxiv.org/abs/2201.12044`.

Pastor, P., Kalakrishnan, M., Righetti, L., and Schaal, S. Towards associative skill memories. pp. 309–315. doi: 10.1109/HUMANOIDS.2012.6651537. URL `http://ieeexplore.ieee.org/document/6651537/`. Conference Name: 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012) ISBN: 9781467313698 Place: Osaka, Japan Publisher: IEEE.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. Stable-baselines3: Reliable reinforcement learning implementations. 22 (268):1–8. URL `http://jmlr.org/papers/v22/20-1364.html`.

Rajeswaran, A., Kumar, V., Gupta, A., Vezzani, G., Schulman, J., Todorov, E., and Levine, S. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*.

Rong, Y., Shiratori, T., and Joo, H. Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1749–1759, 2021.

Rueckert, E., Mundo, J., Paraschos, A., Peters, J., and Neumann, G. Extracting low-dimensional control variables for movement primitives. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1511–1518. doi: 10.1109/ICRA.2015.7139390. ISSN: 1050-4729.

Rückert, E. and d'Avella, A. Learned parametrized dynamic movement primitives with shared synergies for controlling robotic and musculoskeletal systems. 7:138. ISSN 1662-5188. doi: 10.3389/fncom.2013.00138.

Santello, M., Flanders, M., and Soechting, J. F. Patterns of hand motion during grasping and the influence of sensory guidance. 22(4):1426. doi: 10.1523/JNEUROSCI.22-04-01426.2002. URL `http://www.jneurosci.org/content/22/4/1426.abstract`.

Saul, K. R., Hu, X., Goehler, C. M., Vidt, M. E., Daly, M., Velisar, A., and Murray, W. M. Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model. 18(13):1445–1458. doi: 10.1080/10255842.2014.916698. URL `https://doi.org/10.1080/10255842.2014.916698`.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. URL `http://arxiv.org/abs/1707.06347`.

Schumacher, P., Häufle, D., Büchler, D., Schmitt, S., and Martius, G. DEP-RL: Embodied exploration for reinforcement learning in overactuated and musculoskeletal systems. URL `https://arxiv.org/abs/2206.00484`.

Seth, A., Hicks, J. L., Uchida, T. K., Habib, A., Dembia, C. L., Dunne, J. J., Ong, C. F., DeMers, M. S., Rajagopal, A., Millard, M., Hamner, S. R., Arnold, E. M., Yong, J. R., Lakshmikanth, S. K., Sherman, M. A., Ku, J. P., and Delp, S. L. OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. 14:1–20. doi: 10.1371/journal.pcbi.1006223. URL `https://doi.org/10.1371/journal.pcbi.1006223`.

Sobinov, A. R. and Bensmaia, S. J. The neural mechanisms of manual dexterity. 22(12):741–757. ISSN 1471-0048. doi: 10.1038/s41583-021-00528-7. URL `https://doi.org/10.1038/s41583-021-00528-7`.

Song, S., Kidziński, \., Peng, X. B., Ong, C., Hicks, J., Levine, S., Atkeson, C. G., and Delp, S. L. Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation. doi: 10.1101/2020.08.11.246801.

Sun, X., Panda, R., Feris, R., and Saenko, K. AdaShare: learning what to share for efficient deep multi-task learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, pp. 8728–8740. Curran Associates Inc. ISBN 978-1-71382-954-6.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. The MIT Press, second edition. URL `http://incompleteideas.net/book/the-book-2nd.html`.

Taheri, O., Ghorbani, N., Black, M. J., and Tzionas, D. GRAB: A dataset of whole-body human grasping of objects. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M. (eds.), *Computer Vision – ECCV 2020*, pp. 581–600. Springer International Publishing. ISBN 978-3-030-58548-8.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE.

Tresch, M. C., Cheung, V. C. K., and d'Avella, A. Matrix factorization algorithms for the identification of muscle synergies: Evaluation on simulated and experimental data sets. 95(4):2199–2212. ISSN 0022-3077, 1522-1598. doi: 10.1152/jn.00222.2005. URL https://www.physiology.org/doi/10.1152/jn.00222.2005.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wang, H., Caggiano, V., Durandau, G., Sartori, Massimo, K., and Vikash. MyoSim: Fast and physiologically realistic MuJoCo models for musculoskeletal and exoskeletal studies. In *2022 IEEE international conference on robotics and automation (ICRA)*. IEEE, a.

Wang, J. M., Hamner, S. R., Delp, S. L., and Koltun, V. Optimizing locomotion controllers using biologically-based actuators and objectives. 31(4):1–11, b. ISSN 0730-0301, 1557-7368. doi: 10.1145/2185520.2185521. URL https://dl.acm.org/doi/10.1145/2185520.2185521.

Won, J., Gopinath, D., and Hodgins, J. Control strategies for physically simulated characters performing two-player competitive sports. 40(4):1–11. ISSN 0730-0301, 1557-7368. doi: 10.1145/3450626.3459761. URL https://dl.acm.org/doi/10.1145/3450626.3459761.

Yang, Q., Logan, D., and Giszter, S. F. Motor primitives are determined in early development and are then robustly conserved into adulthood. 116(24):12025–12034. doi: 10.1073/pnas.1821455116. URL https://www.pnas.org/doi/abs/10.1073/pnas.1821455116.

Yin, Z., Yang, Z., Van De Panne, M., and Yin, K. Discovering diverse athletic jumping strategies. 40(4):1–17. ISSN 0730-0301, 1557-7368. doi: 10.1145/3450626.3459817. URL https://dl.acm.org/doi/10.1145/3450626.3459817.

Zhang, Y. and Yeung, D.-Y. A regularization approach to learning task relationships in multitask learning. 8(3):12:1–12:31. ISSN 1556-4681. doi: 10.1145/2538028. URL https://doi.org/10.1145/2538028.

A. Appendix

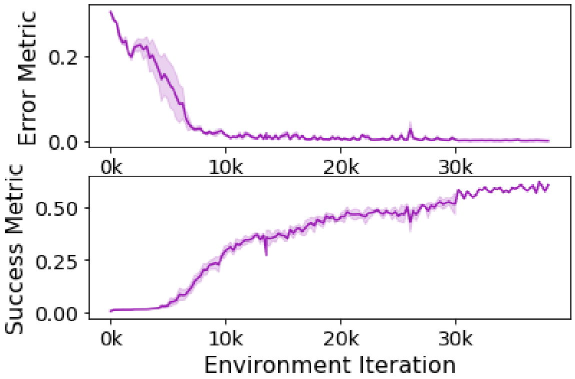

Figure A.1: Success and error metrics for the multi-task policy trained on 14 environments in the first 40k iterations on 4 seeds (line average and errors as shaded area).

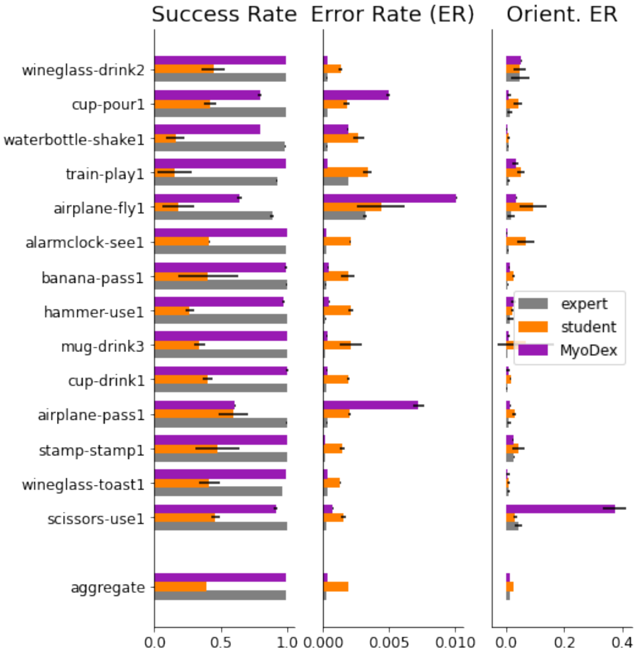

Figure A.2: Baselines: Success, error rate, orientation error metrics (in the left, middle and right columns, respectively – see definitions in Sec. 5) for Individual-Task Experts π_i, Multi-task Student policy π^*, and Multi-task *MyoDex* $\pi^\#$ policy. On the Y-axis the 14 tasks used to traing *MyoDex* are reported, in addition to an aggregate information. *MyoDex* is able to match individual-Task Experts solutions across the 3 differnt metrics. Nevertheless, the multi-task student policy was able to achieve lower perforances overall in most of the individual tasks.

A.1. Imitation learning.

In addition to *MyoDex* $\pi^\#$, we also train a baseline agent using π^* expert-student method (Jain et al.; Chen et al.). Individual task-specific policies (π_i) were used as experts. We developed a dataset with 1M samples of observation-action tuples for each of those policies. A neural network was trained via supervised learning to learn the association between observations and actions to obtain a single policy $\pi^*(a_t|s_t)$ capable of multiple task behaviors.

For distilling the single expert agents into one, a neural network of the same size of the single agent was used. We adopted a batch size of 256, and Adadelta optimizer with a learning rate of 0.25, a Discount Factor (γ) of 0.995, and 10 epochs.

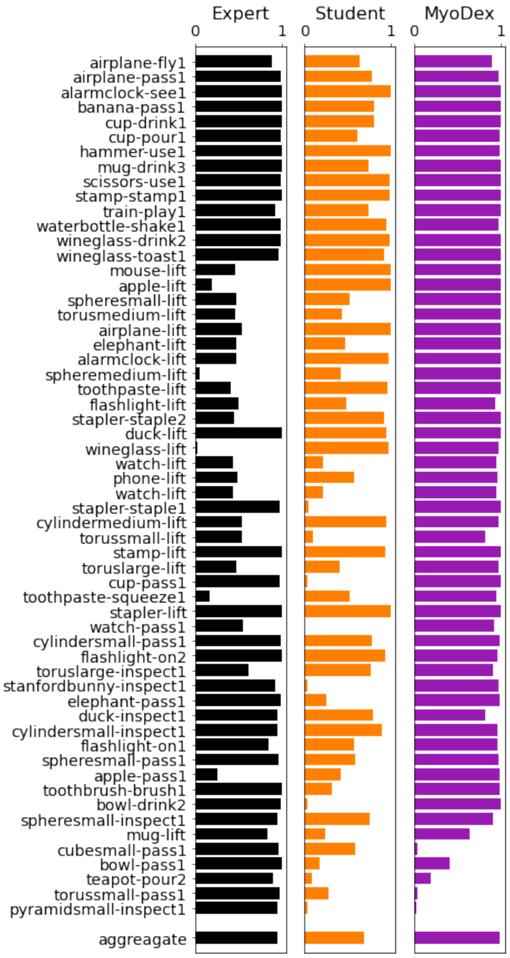

Figure A.3: Summary of all Tasks. Left column tasks solved by single expert policies. Right columns, task fine tuning based on *MyoDex*. See also Table A.2. Training values reported at $12.5k$ iterations.

A.2. Noise

In real-world scenarios, calculating the precise position, and trajectory of an object is often subject to errors. To address this issue, we conducted a study to investigate the resilience of these policies to noisy measurements. We emulate real-world tracking errors by gradually adding increasing levels of noise to the trained policies during deployment (see Figure A.6). We see that *MyoDex* policies are able to handle significant levels of noise (up to 100mm) in the observation of the object's position with limited loss in performance.

A.3. *MyoDex* Alternatives

The choice of objects and tasks can significantly impact the effectiveness of the *MyoDex* representation. To investigate this, we conducted an ablation experiment using two new sets of 14 tasks each: a diverse tasks collection with similar complexity as *MyoDex* – *MyoDex Alt Diverse* (shown in blue in Figure 9)– and the other with a homogenous task collection *MyoDex Alt Homogenous* (shown in red in Figure 9). *MyoDex Alt Homogenous* shows a quick rise in performance which saturates due to overfitting A.7. In contrast, *MyoDex Alt Diverse* observes a slow start but is gradually able to improve its performance over time. Figure 10 shows that the priors implicitly induced by the richer task diversity leads to better generalization and transfer to new unseen tasks.

Samples for Iterations	4096
Discount Factor (γ)	0.95
GAE-λ	0.95
VF Coefficient (c1)	0.5
Entropy Bonus (c2)	0.001
Clip Parameter (ϵ)	0.2
Batch Size	256
Epochs	5
Network Size	$pi = [256, 128], vf = [256, 128]$

Table A.1: Parameters adopted for the reinforcement learning models.

A.4. Synergy probing

To quantify the level of muscle coordination required for accomplishing a given task, we calculated muscle synergies by means of Non-Negative Matrix factorization (NNMF) (Tresch et al.).

After training, we played policies for 5 roll-outs to solve specific tasks and we stored the muscle activations (value between 0 and 1) required. Then, a matrix A of muscle activations over time (dimension 39 muscle x total task duration) was fed into a non-negative matrix decomposition (*sklearn*) method.

The NNMF method finds two matrices W and H that are respectively the coefficients and the basis vectors which product approximates A. Muscle synergies identified by NNMF capture the spatial regularies on the muscle activations whose linear combination minimize muscle reconstruction (Bizzi & Cheung). This method reveals the amount of variance explained by each of the components. We calculated the Variance Accounted For (VAF) as:

$$VAF = 100 \cdot \left(1 - \frac{(A - W \cdot H)^2}{A^2}\right) \tag{2}$$

Similarity of synergies between two different tasks was calculated using cosine similarity (CS) such as: $CS = w_i \cdot w_j$, where $[w_i,\ w_j] \in W$ are synergy coefficients respectively for the task i and j. We used then a threshold of 0.8 to indicate that 2 synergies were similar Appendix-A.4.

While the student policy – obtained with imitation learning – produced muscle activations similar to that of the respective task expert but it effectiveness was quite low in task metrics.

A.4.1. Does *MyoDex* produce reusable synergies?

Biological systems simplify the problem to control the redundant and complex muscuolokeletal systems by resorting on activating particular muscle groups in consort, a phenomenon known as muscle synergies. Here, we want to analyse if synergies emerge and facilitate learning.

For *MyoDex* where an agent has to simultaneously learn multiple manipulations / tasks, common patterns emerges and fewer synergies i.e. 12 (Figure A.8), can explain more than 80% of the variance of the data. Furthermore, we observe that tasks start sharing more synergies (on average 6, see Figure A.4). This is expected as each task needs a combination of shared (task-aspecific) and task-specific synergies. Common patterns of activations seems to be related with learning. Indeed, earlier in the training more synergies are needed to explain the same amount of variance of the data. The peak is reached at $12.5k$ iterations where more than 90% of the variance is explained by 12 synergies (see Figure A.8).

As expected, the expert policies shared fewer common muscle activations as indicated by fewer synergies shared between tasks (on average 2, see Figure A.4) and by the overall greater number of synergies needed to explain most of the variance: to explain more than 80% of the variance it is needed to use more than 20 synergies. Similar results were obtained with the student policy (on average 1 similar synergies between tasks, see Figure A.4).

A.4.2. PreGrasp informed Dexterous Manipulation

We adopted Dasari et al (Dasari et al., b) solution where the desired object trajectory $\hat{X} = [\hat{x}^0, ..., \hat{x}^T]$ is leveraged to capture the temporal complexities of dexterous manipulation. Additionally hand-object pre-grasp posture $\phi_{object}^{pregrasp}$ is leveraged to

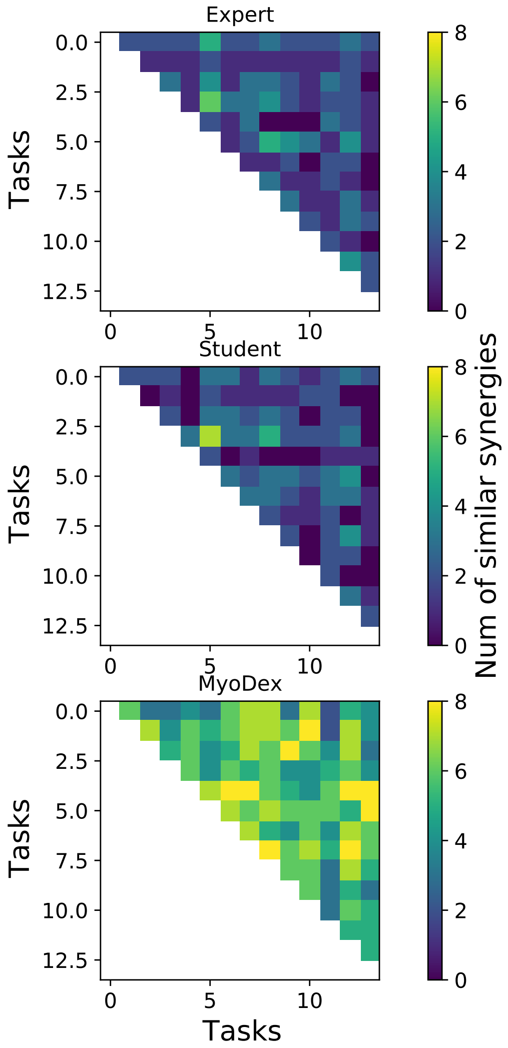

Figure A.4: Cosine Similarity between 12 synergies extracted from 14 different tasks. Top - expert policies. Middle - student policy. Bottom – *MyoDex*. On average the number of similar synergies for expert, student, *MyoDex* (mean +/- std over 10 repetitions with different random seeds) was 1.86 ± 1.19, 1.71 ± 1.29 and 5.56 ± 1.59, respectively.

guide the search space. For each task, first a trajectory planner is used to solve for the free space movement driving the hand to the pre-shape pose, and then PPO is employed to solve for achieving the desired object trajectory. We extracted relevant pregrasp informations from the GRAB motion capture (Taheri et al.) dataset which contains high-quality human-object interactions. Note that these details can also be acquired by running standard hand tackers (Rong et al., 2021) on free form human videos.

We used the hand posture just before the initial contact with the object (see Figure 3) as the pre-grasp posture. This allows us to not require any physical or geometric information about the object. Given this tracked posture, we recover MyoHand posture via means of Inverse Kinematics over the finger tips.

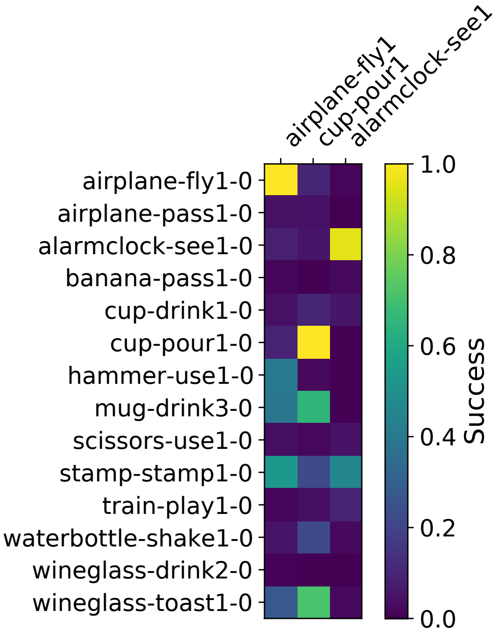

Figure A.5: Fine-tuning based on expert policies. Success rate fine-tuning experts solutions (columns) on 14 different environments. This matrix shows that the combination of pre-grasps and the initialization on a pre-trained task is not enough to generalize to new tasks.

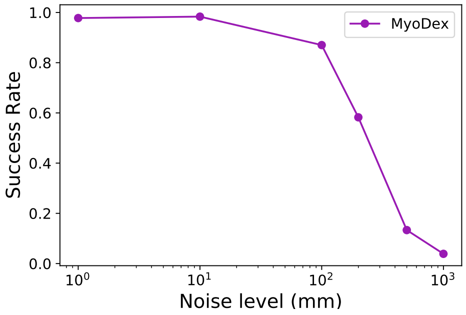

Figure A.6: Performance according to addional noise (in mm) in the observation of the object.

Task	Multi-task Success			Iter. to reach Success of 0.8	
	@ 1k Iter.	@ 2k Iter.	@ 3k Iter.	Multi-Task	Expert
stamp-stamp1	0.981538	0.997949	1.000000	247	3458
banana-pass1	0.910000	0.993333	0.998571	247	4446
cup-drink1	0.991724	0.998161	0.977471	247	3952
mug-drink3	0.978667	0.999467	1.000000	247	3458
alarmclock-see1	0.984444	0.997778	1.000000	494	4940
train-play1	0.822278	0.929114	0.987848	741	8398
scissors-use1	0.754699	0.945542	0.986988	1235	5434
wineglass-drink2	0.714943	0.924138	0.985287	1235	4446
hammer-use1	0.781429	0.870000	0.972857	1482	3952
wineglass-toast1	0.713846	0.796410	0.902051	2223	4199
cup-pour1	0.743429	0.730857	0.830286	2964	4446
waterbottle-shake1	0.574595	0.709189	0.743784	3458	5434
airplane-fly1	0.564675	0.606753	0.631169	12350	7657
airplane-pass1	0.436322	0.497011	0.509425	12597	6669
mouse-lift	1.000000	1.000000	1.000000	247	-
apple-lift	1.000000	1.000000	1.000000	247	-
spheresmall-lift	0.986667	1.000000	1.000000	247	-
torusmedium-lift	0.980571	1.000000	1.000000	247	-
airplane-lift	0.995122	1.000000	1.000000	247	-
elephant-lift	1.000000	1.000000	1.000000	247	-
alarmclock-lift	1.000000	1.000000	1.000000	247	-
spheremedium-lift	0.998947	1.000000	1.000000	494	-
toothpaste-lift	0.971818	0.952727	0.990000	494	-
flashlight-lift	0.941714	0.942857	0.942857	494	-
stapler-staple2	0.991529	1.000000	0.999529	494	-
duck-lift	0.994737	1.000000	1.000000	494	5434
wineglass-lift	0.933000	0.979500	0.980000	494	-
watch-lift	0.925333	0.955556	0.955556	741	-
phone-lift	0.960000	0.967742	0.967742	741	-
stapler-staple1	0.893809	0.989524	0.996190	988	5187
cylindermedium-lift	0.841111	0.970000	0.972222	988	-
torussmall-lift	0.690285	0.931428	0.915428	1235	-
stamp-lift	0.709756	0.980488	0.992195	1235	3211
toruslarge-lift	0.707273	0.965455	0.977273	1235	-
cup-pass1	0.609048	0.995238	1.000000	1235	4446
toothpaste-squeeze1	0.598421	0.943157	0.977368	1482	-
stapler-lift	0.650732	0.868293	0.982439	1482	2717
watch-pass1	0.492593	0.887407	0.884444	1729	-
cylindersmall-pass1	0.571200	0.826667	0.901333	1976	4693
flashlight-on2	0.168791	0.695385	0.920000	2470	6175
toruslarge-inspect1	0.251852	0.645926	0.817778	2470	-
stanfordbunny-inspect1	0.289157	0.591325	0.921446	2470	6422
elephant-pass1	0.506667	0.621235	0.834568	2964	5928
duck-inspect1	0.621299	0.624935	0.820260	2964	5681
cylindersmall-inspect1	0.420000	0.713333	0.639444	3705	6422
flashlight-on1	0.234483	0.541609	0.626207	4446	10127
spheresmall-pass1	0.191905	0.351905	0.674286	4446	5928
apple-pass1	0.344198	0.481481	0.583210	5187	-
toothbrush-brush1	0.119375	0.353125	0.589063	5434	4199
bowl-drink2	0.075714	0.089524	0.163810	7657	4693
spheresmall-inspect1	0.235676	0.332432	0.438919	8151	7163
mug-lift	0.326575	0.335342	0.397808	-	7904
cubesmall-pass1	0.024691	0.024691	0.024691	-	5928
bowl-pass1	0.114430	0.153418	0.184810	-	7163
teapot-pour2	0.137627	0.150508	0.162712	-	7657
torussmall-pass1	0.038987	0.037975	0.037975	-	5928
pyramidsmall-inspect1	0.028571	0.033333	0.035238	-	5187

Table A.2: *MyoDex* based fine-tuning and expert solutions for all 57 tasks. Expert solutions could reliably reach 0.80 success for the first 14 tasks but in many other cases they were not able to. A few exceptions at the bottom show success only for expert solutions. We indicated with '-' the lack of success in achieving the success threshold. The first 3 columns report the success rate respectively at $1k$, $2k$ and $3k$ iterations. The 4th and 5th columns, document the iterations at 0.80 success for *MyoDex* based fine-tuning and experts.

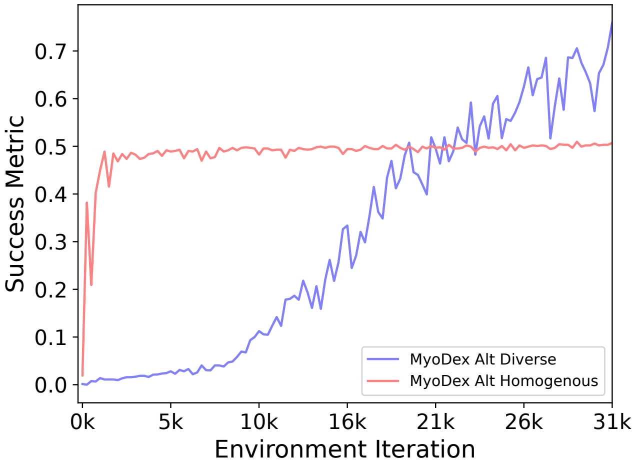

Figure A.7: Effect of pre-training task distribution on *MyoDex* training. *MyoDex Alt Homogenous* shows a quick rise in performance which saturates due to overfitting. In contrast, *MyoDex Alt Diverse* observes a slow start but is gradually able to improve its performance over time.

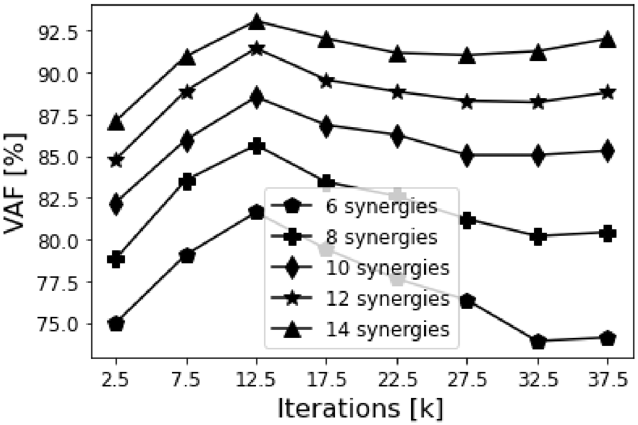

Figure A.8: Muscle Synergies over learning iterations for the joint multi-task policy. Variance of the muscle activations (see Sec. A.4) explained as function of the number of synergies at different steps of the learning process.

Object	Creator	License
waterbottle	badger	GNU GPL v2
train	Jason Shoumar	public domain
airplane	Gravity Sketch	CC BY-4.0
wine glass	Michael Spivey	CC BY 3.0
cup	Ablapo	CC BY 3.0
mug	Ryan Smith	credit, remix, non-commercial
alarm clock	Javier Ruiz	CC BY-SA 3.0
banana	Lloyd Bolts	credit, remix, non-commercial
hammer	Microsoft	CC BY 4.0
mouse	Michael Spivey	CC BY 3.0
duck	willie	CC0 1.0

Table A.3: Creators and License for the objects illustrated.