MyoDex: GENERALIZABLE REPRESENTATIONS FOR DEXTEROUS PHYSIOLOGICAL MANIPULATION

Anonymous authors
Paper under double-blind review

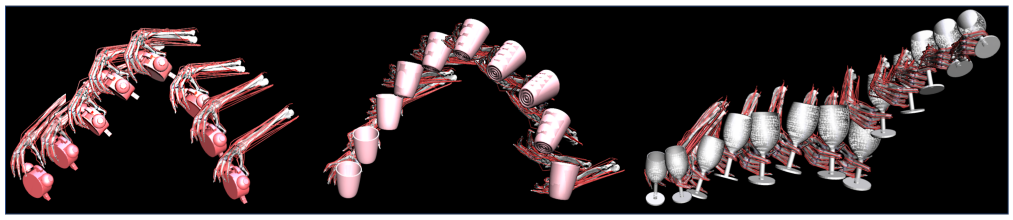


Figure 1: Contact rich manipulation behaviors acquired by *MyoDex* with physiological *MyoHand*

ABSTRACT

The complexity of human dexterity builds on the coordinated actuation of a large number of muscles. Still, much is to be understood about how the control of such overactuated system for hand manipulation behaviors emerge and quickly and flexibly adapts to new behaviours. In this work we aim at learning generalizable representations for dexterous manipulation behaviors with a physiologically realistic hand model: *MyoHand*. In contrast to prior works demonstrating isolated postural and force control, here we demonstrate musculoskeletal agents (*MyoDex*) exhibiting contact-rich dynamic dexterous manipulation behaviors in simulation. Furthermore, to demonstrate generalization, we show that a single *MyoDex* agent can be trained to solve up-to 14 different contact-rich tasks. Aligned with human development, simultaneous learning of multiple tasks imparts physiological coordinated muscle contractions i.e., muscle synergies, that are not only shared amongst those in-domain tasks but are also effective to a large series of new out-of-domain tasks. By leveraging these pre-trained manipulation synergies, we show generalization to 38 additional previously unsolved tasks. While physiological behaviors with large muscle groups (such as legged-locomotion, arm-reaching, etc) have been demonstrated before, to the best of our knowledge nimble behaviors of this complexity with smaller muscle groups and generalizable representations for the control of the overactuated human hand are being demonstrated for the first time.
Project Webpage: https://sites.google.com/view/myodex

Musculoskeletal, muscle synergies, Machine Learning, human dexterity

1 INTRODUCTION

Human hands are astonishingly complex and require effective coordination of various muscle groups to impart effective manipulation abilities. Manipulation behaviors are incredibly sophisticated as, because of the overactuated musculoskeletal system, they evolve in a high-dimensional search space populated with intermittent contact dynamics between the hands' degrees of freedom and the object. Indeed, even in the field of robotics where joints and actuations are simpler, finding effective manipulation strategies nonetheless remains a challenge Kumar et al. (2016); Rajeswaran et al. (2018); Nagabandi et al. (2020).

The human hand consists of 29 bones, 23 joints, and more than 50 muscles Sobinov & Bensmaia (2021). The complex multi-articular, multi-joint, pulling-only properties of the musculoskeletal system Sobinov & Bensmaia (2021) make physiological dexterous manipulation a very different and unique problem as opposed to joint based control typically adopted in robotics. In biology, the control of such complex musculoskeletal system is made possible by the fact that muscles are

not activated in isolation, but rather, that different muscles are activated in a proportional way as a unit. This phenomenon is known as muscle synergy Bizzi & Cheung (2013). Synergies allows the biological motor system – via the modular organization of the movements in the spinal cord Bizzi & Cheung (2013); Caggiano et al. (2016) – to simplify the control problem, solving tasks by building on a limited number of shared solutions d'Avella et al. (2003); d'Avella & Bizzi (2005). Those shared synergies are suggested to be the fundamental building blocks for quickly learning new and more complex motor behaviours Yang et al. (2019); Dominici et al. (2011); Cheung et al. (2020). Manipulation behaviors, the subject of this investigation, are further complicated because they unfold on a sequence of phases: reaching to the object, hand-object contact, and manipulation with object maneuvers. Before the hand-object contact, the human hand is pre-shaped to conform to the object such that it is often possible to predict the object that is going to be grasped just by observing the hand pose before hand-object contact Jeannerod (1988); Santello et al. (2002); Thakur et al. (2008); Yan et al. (2020). Contact and manipulation of the object are goal-driven so that the way the object is held depends on both the object affordance and the intermediate task goals Jeannerod (1988).

In this work, we seek to further our understanding of physiological dexterity by imparting dexterous manipulation ability to an anatomically realistic hand-fore-arm model Caggiano et al. (2022). While prior works have not been able to scale beyond dexterous grasping McFarland et al. (2021); Mirakhorlo et al. (2018); Saito et al. (2021); Crouch & Huang (2015); Engelhardt et al. (2021) in a controlled setting with a physiologically realistic models of the hand, here we present *MyoDex* agents capable of dynamic dexterous contact rich manipulation behaviors with multiple objects and a variety of tasks e.g. drinking from a cup, playing with toys, etc. Furthermore, by jointly training multiple tasks, we capture reusable synergies in form of a general pre-trained policy that can be further fine-tuned to manipulate 38 previously unsolved tasks with non-trivial affordances. We provide a detailed analysis of emergent physiological details in our achieved behaviors.

While we do not claim to have solved physiological dexterous manipulation, we emphasize that manipulation abilities demonstrated here significantly advance the state of the art of the bio-mechanics and neuroscience fields. Along these lines, this investigation is among the first to yield robust control policies exhibiting basic physiological constructs such as synergistic activations of muscle groups during dexterous manipulations. Nevertheless, further work is required to rigorously ground them in experimental validation. More specifically, our main contributions are:

- We show for the first time that despite the high numbers of degrees of freedom, the multi-articular-multi-joint and the third order muscle dynamics of muscle control, **it is possible to control a physiologically realistic musculoskeletal model of the hand to perform contact-rich skilled manipulation** behaviors on up-to 14 different tasks.
- We show that joint multi-task learning facilitates the **learning of physiological representations that exploit muscle coordination in a lower-dimensional space of synergies** to solve specific tasks.
- Our framework ***MyoDex*** **leverages joint multi-task learning to recover reusable representations (synergies) that allows for easier fine-tuning in both in-domain and out-of-domain tasks** (including one/few shot learning). Leveraging these synergies the *MyoDex* solves 38 previously unsolved tasks.

2 Related Works

Experimental studies of functional hand manipulations have been limited both by challenges in sensing, the discontinuous hand-object interactions and because of the limited ability to record many muscles of the hand simultaneously. Musculoskeletal models of the hand McFarland et al. (2021); Lee et al. (2015); Saul et al. (2015) have been developed to overcome some of the experimental limitations and produce insights on the kinematic information of the muscles and joints. While musculoskelatal models of large muscle groups have been extensively developed and used Delp et al. (2007); Seth et al. (2018), models of the hand have been more challenging both because of the smaller muscle groups involved and the complexity of the behaviour they can produce. Indeed, simulations of the hand mostly focus on fingertips, pinch force McFarland et al. (2021), kinematic motion McFarland et al. (2021), and passive grasping McFarland et al. (2021). Furthermore, most of those studies are also limited by intensive computational needs and restricted contact forces. Those

conditions prevented the study of complex hand object interactions and limited by using optimization methods that could not leverage data-driven state of the art.

Recently, a new hand and wrist model – MyoHand Caggiano et al. (2022); Wang et al. (2022) – has been developed. This model overcomes some limitations of alternative hand models 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).

Hand dexterity has been also a very active field in robotics. Robotic hands have been shown to be able to perform complex in-hand manipulation of real-world objects OpenAI et al. (2019); Huang et al. (2021); Chen et al. (2021) and solve complex manipulation tasks, such as *HandManipulateEgg* and *HandManipulatePen* Plappert et al. (2018). Still, both the hardware and the control of robotic hands do not match the level of dexterity of human hands and remain limited to in-hand movements.

Data driven approaches have consistently used Reinforcement Learning (RL) on joint-based control to solve simple locomotion tasks Miki et al. (2022), in animation of physics based characters Heess et al. (2017) and to solve complex dexterous manipulation in robotics Rajeswaran et al. (2018); Kumar et al. (2016); Nagabandi et al. (2019); Chen et al. (2021). Typically, in order to yield more naturalistic movements, different methods have leveraged motion capture data Merel et al. (2017; 2019); Hasenclever et al. (2020). By means of those approaches, it has been possible to learn complex movements and athletic skills such as high jumps Yin et al. (2021), boxing and fencing Won et al. (2021) or playing basketball Liu & Hodgins (2018). More recently, approaches that prime models with hand pre-shaped for a specific task have been shown to be successful at simplifying the search of RL solutions on complex robotic manipulations Dasari et al. (2022). In contrast to joint-based control, in biomechanical models machine learning has been applied on 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. (2022). Wang et al. Wang et al. (2012), Geijtenbeek et al. Geijtenbeek et al. (2013), Borno et al. 2020 Al Borno et al. (2020), and Ruckert et al. Rückert & d'Avella (2013), have been using optimization methods on biomechanical models to synthesize walking and running, reaching movements, and biped locomotion. More recently, deep reinforcement learning has been used to either map the muscle-actuation to joint-actuation control to produce movements that are more human-like than those generated by torque-based control at the joints Jiang et al. (2019), in order to directly control shoulder and arm muscles for isometric arm movements Joos et al. (2020) and reaching Schumacher et al. (2022); Ikkala et al. (2022), hand muscles for hand dexterous manipulations Caggiano et al. (2022), co-learning elbow exoskeleton movements Caggiano et al. (2022); Wang et al. (2022), walking/running Song et al. (2020); Park et al. (2022), or to produce movements such as juggling, weight lifting, cart-wheeling and other highly stylistic movements Lee et al. (2018; 2019). Musculoskeletal models have been used also to improve the realism of simulated animal movements; for example, in controlling movements in animal models of a dog Stark et al. (2021) and more recently, of an ostrich Barbera et al. (2021); Schumacher et al. (2022).

While musculoskeletal control with large muscles groups have been demonstrated Song et al. (2020; 2021); Schumacher et al. (2022); Ikkala et al. (2022), nimble contact rich musculoskeletal behaviors with smaller muscle groups such as hand-manipulation remains an open challengeCaggiano et al. (2022). *MyoDex*, in addition of showing that indeed it is possible, presents evidence that the learned physiological representations share muscle coordination across tasks which, like human synergistic control, facilitate both learning and generalization across tasks.

3 Physiological Dexterity

Human hand dexterity builds on the fundamental characteristics of the physiological actuation: muscle are multi-articular and multi-joints, the dynamics of the muscle is of the third order, muscle have pulling only capabilities, and coordinated synergistic muscle control with intermittent contact with objects. Furthermore, the key aspect of the control of such physiological effectors is that the human central nervous system optimizes movements through coordinated muscle contraction – muscle synergies – which are meant to simplify the control problem, allowing generalization. Fields of bio-mechanics, rehabilitation, neuro-surgery, etc. have long benefited from physiological understanding of neuro-mechanical control. To further our understanding, here we embed the same

control challenges e.g. by controlling a physiologically accurate musculoskeletal model of the hand (see Sec. 3.1) in complex manipulation tasks (see Sec. 3.2). This allows a window to peek into the mechanisms behind human dexterity that enables generalization across different tasks.

3.1 PHYSIOLOGICALLY ACCURATE MYOHAND

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. (2022) - MyoHand model - implemented in the MuJoCo physics simulator Todorov et al. (2012) was used (see Figure A.1). This hand model has been shown to allow dexterous in-hand manipulation of one or multiple objects when trained using reinforcement learning Caggiano et al. (2022).

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 favor shoulder and wrist rotation.

3.2 TASK

Dexterous manipulation is often posed as a problem of achieving the final configuration of the object. In this study we are interested to capture the whole continuous aspects of the manipulation behaviour with object maneuver e.g., drinking, playing, or cyclic movement like hammering. Those tasks are hard to capture as goal reaching. To effectively capture the temporal behaviors, we instead define dexterous manipulation as a task of realizing a desired object trajectory ($\hat{X}$). We use two metrics to measure task performance. The object error metric $E(\hat{X})$ calculates the average Euclidean distance between the object's center-of-mass position[1], and the desired position from the desired trajectory: $E(\hat{X}) = \frac{1}{T}\sum_{t=0}^{T} \|x_t^p - \hat{x}_t^p\|_2$. In addition, 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: $S(\hat{X}) = \frac{1}{T}\sum_{t=0}^{T} \mathbb{1}\|x_t^p - \hat{x}_t^p\|_2 < \epsilon$

3.3 DEXTERITY OBJECTIVES

While the complexity of human level dexterity can be hard to fully quantify, none the less we outline a few objective measures we consider in this work. First, a dexterous agent should be capable of exhibiting contact-rich manipulation behaviors. Next, the agent's behavior should seamlessly generalize to multiple tasks/objects without additional assumptions. Finally, these agents should exhibit coordinated muscles movements (synergies) that are shared amongst different behaviors, as well as generalize to new unseen tasks.

4 *MyoDex*: ACQUIRING DEXTERITY

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 (2018) 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 $\mathcal{A} \subseteq \mathbb{R}^m$ represent the continuous state and action spaces respectively. The unknown transition dynamics is 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 $J = \max_\theta \mathbb{E}_{s_0\sim\rho(s), a\sim\pi_\theta(a_t|s_t)}[\sum_t R(s_t, a_t)]$

[1]For interpretabiliy, we omit orientations because center-of-mass error and orientation error were highly correlated in practice

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 arms joints and velocity $\dot{\phi}$, and object pose ψ and velocity $\dot{\psi}$. In addition, positional encoding τ Vaswani et al. (2017), 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 which consists of continuous activations for 39 muscles of 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 wider 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 for a fixed time horizon. Similar to Dasari et al. (2022), this is translated into an optimization problem where we are searching for a policy that can match a desired object trajectory $\hat{X} = [\hat{x}^0, ..., \hat{x}^T]$, which is captured using the following reward function:

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

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

4.2 PREGRASP TO SIMPLIFY SEARCH SPACES

Owing to the third order non linear actuation dynamics and high dimensionality of the search space, direct optimisation of $\mathcal{M}$ leads to no meaningful behaviors. We leverage the state directly preceding the hand initiating contact with an object – i.e. pre-grasp – to greatly decrease the complexity of learning dexterous behaviors Dasari et al. (2022). Pre-grasp implicitly incorporates information pertaining to the shape of the object and its associated affordance with respect to the desired task Jeannerod (1988); Santello et al. (2002). Additionally, pre-grasps can be used in our experiments without additional assumptions as they can be easily mined from MoCap recordings, annotated by human labelers, or even predicted by learned models Taheri et al. (2020a). We choose to adopt the technique from Dasari et al. (2022) and break the learning into two phases. In the first phase the hand learns to reach the pre-grasp pose using a free-space planners (no object conditioning required). Next, RL agents are trained to perform either a single target task or a family of tasks. We now describe these approaches in detail.

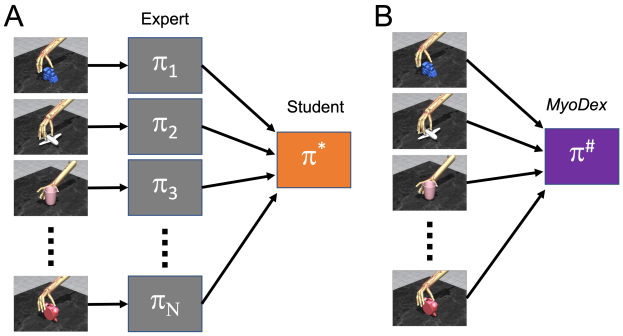


Figure 2: Learning Paradigms. A - Single Experts policies were obtained by training policies to solve the individual tasks. Then, by means of an expert-student approach, a unified Student policy was distilled. B - A single policy (*MyoDex*) was obtained by learning all tasks at once.

Single task agents. In our first setting, we adopt a standard RL algorithm (see 4.1) to learn a goal-conditioned policy $\pi_\theta(a_t|\phi_t, \dot{\phi}_t, \psi_t, \dot{\psi}_t, \tau_t, \hat{X}_{object}, \phi_{object}^{pregrasp})$ (see notation in Sec. 4.1) that can solve a single task. This approach will define a set of expert agents defined as π_i with $i \in I$ where I is the set of tasks (see Figure 2A).

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 all 14 tasks in parallel (see Figure 2B). This approach proceeds in a similar fashion as the single-task learner, but trajectory rollouts are sampled from the 14 tasks in *parallel*. All other details of the agent $\pi_\theta^\#(a_t|\phi_t, \dot{\phi}_t, \psi_t, \dot{\psi}_t, \tau_t, \hat{X}_{object}, \phi_{object}^{pregrasp})$ (e.g. hyperparameters, algorithm, etc.) stay the same.

We encode manipulation behaviors in term of goal-conditioned policies $\pi_\theta(a_t|s_t)$. A standard implementation of the PPO Schulman et al. (2017) method from Stable-Baselines Raffin et al. (2021) was used. Same hyperparameters were used for all tasks (see Appendix Table A.2).

Imitation learning. In addition to *MyoDex* $\pi^{\#}$, we also train a baseline agent using π^* expert-student method Jain et al. (2019); Chen et al. (2021) (see Figure 2A). 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 policy. Then, we extended the observation vector to include the vector τ_{task} representing the object and trajectory. Finally, a neural network similar to Dasari et al Dasari et al. (2022) was trained via supervised learning to learn the association between observations and actions (see hyperparameters in Appendix A.1) to obtain a single policy $\pi^*(a_t|\phi_t, \dot{\phi}_t, \psi_t, \dot{\psi}_t, \tau_t, \tau_{task})$ capable of multiple task behaviors (see Figure 2A).

5 Experimental Design

5.1 Task design

In this study, we need a large variability of manipulations, hence it was important to include 1) objects with different shapes and weights, 2) complexity both in terms of translation and rotation of the object. Also, having different movements on the same objects allows us to investigate how the different hand pre-shapes and object trajectory affected the solution.

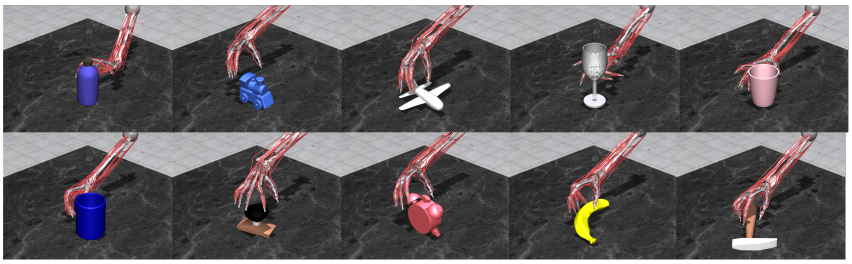


Figure 3: A subset of *object-pregrasp* pair from our task-set. See Table A.1 for a complete description

A set of 11 Objects and 14 different behaviors (see Table A.1) similar to the TDCM dataset presented by Dasari et al Dasari et al. (2022) were used. The setup (see Figure 3) consisted of a table-top environment, an object from the ContactDB dataset Brahmbhatt et al. (2019) and the MyoHand Caggiano et al. (2022). This dataset was implemented in the MuJoCo physics engine Todorov et al. (2012). To define our tasks, we adopted Dasari et al Dasari et al. (2022) solution where the desired object trajectory $\hat{X} = [\hat{x}^0, ..., \hat{x}^T]$ and the hand-object pre-grasp posture $\phi_{object}^{pregrasp} = [j_0, ..., j_n]$ where needed. We extracted these information from the GRAB motion capture Taheri et al. (2020b) dataset which contains high quality human-object interactions.

Following the same approach of Dasari et al Dasari et al. (2022), hand postures were computed by matching the human fingertip of the ContactDB dataset and MyoHand by means of Inverse Kinematics. In the context of this work, only the hand pre-shaped to grab the object before the initial contact (see Figure 3) was considered. For each task, a pre-shaped hand was used to initialize the posture of the hand and the goal was to follow a given trajectory of the object. This allow us to avoid any physical or geometric information about the object. Each tasks consisted of a pair of one trajectory of an object and its associated pre-shaped posture.

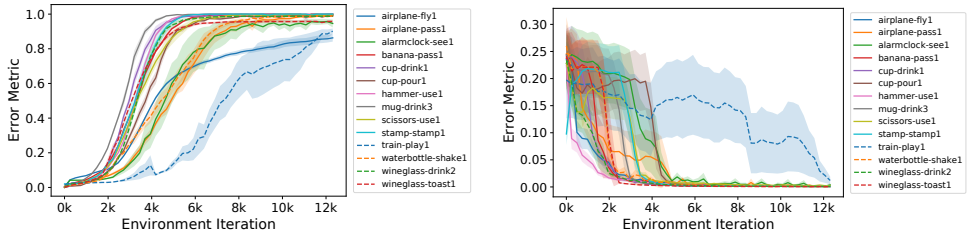
5.2 Synergy probing

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. (2006). 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 regularities on the muscle activations whose linear combination minimize muscle reconstruction Bizzi & Cheung (2013). 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.6.



(a) Success Metric **(b)** Error Metric

Figure 4: Single task expert task-performance. Metrics: (a) Success and (b) Error rate over iteration steps. Average and standard deviation (shaded areas) over 4 seeds are shown.

6 Experimental Results

First we present how by leveraging pregrasps in standard RL pipeline, it was possible to control the physiological MyoHand to perform various tasks involving contact rich manipulation of objects (Sec.6.1). Then, we detail a deeper investigation illustrating how the muscle coordination evolves and changes as function of task conditions, learning (Sec. 6.3) and support generalization (Sec. 6.4).

6.1 *MyoDex*'s task dexterity

First, we wanted to explore if we can learn a series of complex dexterous object manipulations required for performing specific tasks (see Sec. 5.1). A set of agents were trained to solve each task independently: expert solutions. The same pipeline and parameters were used to solve all tasks without any object or task-specific tuning (see Table A.2). Qualitatively, all objects in the sample were properly manipulated while moving them to follow the target trajectory (see Figure 1 for a sequence of snapshots). This was quantified by means of 2 metrics (section 3.2): Success Metric (Figure 4a) and Error Metric (Figure 4b). In all cases we achieved greater than 80% success and, overall, an error below 0.01. These analysis indicated that *MyoDex* is able to effectively drive a musculoskeletal model of the hand to learn stable object manipulations within very tight margins. To the best of our knowledge, this is the first demonstration of such nimble manipulation (see project website for behavior videos) with physiological musculoskeletal hand.

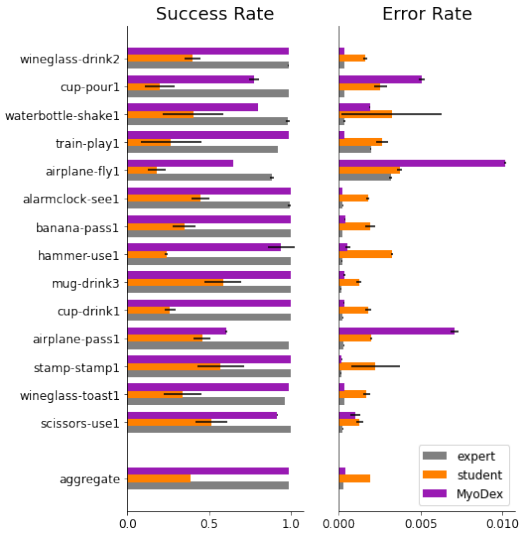


Figure 5: Baselines: Success and error rate metrics for Expert π_i, student π^*, and *MyoDex* $\pi^{\#}$ policies.

6.2 *MyoDex*'s multi task learning

Next, to investigate *MyoDex* task generalization, we trained one single policy to handle all tasks simultaneously. We approached this problem in two ways. First, we used the above policies trained

to solve each single task as expert policies and, by means of an expert (teacher)-student learning paradigm, we distilled the experts into one policy that solved all tasks at the same time (see Sec.4.2). Second, by means of joint multi-task learning, we obtained one policy – *MyoDex* – that solved multiple tasks (see Sec.4.2). In both cases, the policies were able to solve the tasks but with key differences. The student policy, when compared with the *MyoDex* multi-task policy, showed a reduced success rate (see Figure 5, median success rate: expert 0.99, student 0.37, *MyoDex* 0.98) and overall greater error (see Figure 5, median error rate: expert 0.00031, student 0.00195, *MyoDex* 0.00046). In particular, the multi-task policy (see Figure A.2a) reached an error below 0.01 in $9.1k$ iterations while the expert policies reached that error cumulatively around $34k$ iterations i.e. 4x slower. On the other hand, expert policies reached success rate of 80% cumulatively in $70k$ vs $123k$ iterations needed for the *MyoDex* policy i.e. double the time. This indicates that jointly learning multiple tasks greatly facilitates the initial phase of learning, while slowing the learning of detailed aspects of each specific tasks. This is likely because tasks like *airplane-pass*, *airplane-fly* and *cup-pour* require task specific and unique wrist rotations to be accomplished.

While the student policy – obtained with imitation learning – produced muscle activations similar (Figure A.3) to that of the respective task expert (Figure A.5) but it effectiveness was quite low in task metrics.

6.3 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.

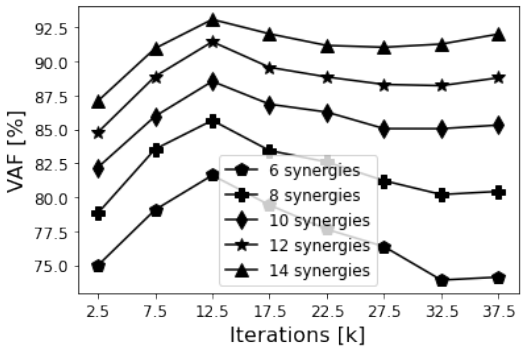


Figure 6: Muscle Synergies over learning iterations for the joint multi-task policy. Variance of the muscle activations (see Sec. 5.2) explained as function of the number of synergies at different steps of the learning process.

For *MyoDex* where agent has to simultaneously learn multiple manipulations / tasks, common patterns emerges and fewer synergies i.e. 12 (Figure 6), can explain the more than 80% of the variance of the data (see Figure A.4). Furthermore, we observe that tasks start sharing more synergies (on average 6, see Figure A.6). 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.

As expected, the expert policies shared fewer common muscle activations as indicated by fewer synergies shared between tasks (on average 2, see Figure A.6) 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 (see Figure A.4). Similar results were obtained with the student policy (on average 1 similar synergies between tasks, see Figure A.6).

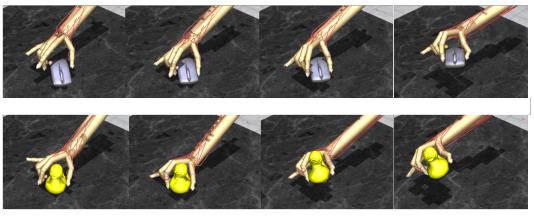


Figure 7: Zero-shot generalization. *MyoDex* successfully initiated manipulations on new objects and trajectories. Hand rendering includes skin (see Fig. A.1)

6.4 *MyoDex* Out of domain generalization

The joint multi-task learning yields a policy that, by only knowing the hand posture (pre-grasp) and without information about the object, shows generalization to unseen objects and trajectories.

Qualitatively in a zero-shot out-of-domain task (Figure 8), the *MyoDex* policy initiates effective manipulations of new objects. Nevertheless, it's is not able to achieve success rates above 0.1. The lack of complete generalization is very likely due to the missing sensory information e.g. from skin receptors, needed in order to properly hold objects with complex geometries. Indeed, in humans, when sensory information at the finger tips are inhibited, proper manipulation cannot be accomplished.

Given this initial indication of generalization capabilities, we want to explore the possibility of using the shared representation provided by the multi-task policy for 1) improving performance on the single tasks i.e. fine-tuning, 2) learning new out-of-domain tasks which were experts policies were not able to learn. We use earlier learned models i.e. $12.5k$ iterations, which provide the most general representation of coordinated movements as shown by the greater variance explained by fewer muscles synergies (see Figure 6). First, for most of the tasks, fine-tuning the multi-task representation allows faster learning of the in-domain tasks (see Table A.3). Indeed, it was possible to achieve 80% success using almost less than half of the iterations ($2.8k$ vs. $5k$, fine-tuned vs experts) required for experts trained without the same model initialization. Second, we used the multi-task representation on a series of 44 new tasks (see Figure 8 and Table A.3). In most of those tasks, the shared representation allowed to quickly learn them. To be noticed, it was not possible to learn expert policies for most of those tasks without this initialization of the model. This indicates that the representation obtained by jointly learning multiple tasks helps to initialize a solution space that avoids local minima.

7 Conclusion

In this manuscript we showed how it is possible to control a musculoskeletal model of the human hand to learn skilled dexterous manipulation of complex objects. We were able to learn these tasks by imposing a pre-shaping of the hand that reduced the search space. In addition, by means of the joint multi-task learning we showed that it is possible to extract generalizable representations that leverage synergies – muscles that are activated as a unit – which allows both faster fine-tuning on downstream in-domain and out-of-domain tasks. All in all, 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 of motion capture data to imitate specific behaviour.

8 Limitations and Future Work

While we have been able to show that we can produce realistic behavior without the need of fitting 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 combination 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 of physiological feasibility of the result, these findings have yet to be properly validated. Future works will need to consider the ability to synthesize new motor behaviors while simultaneously providing muscle validation.

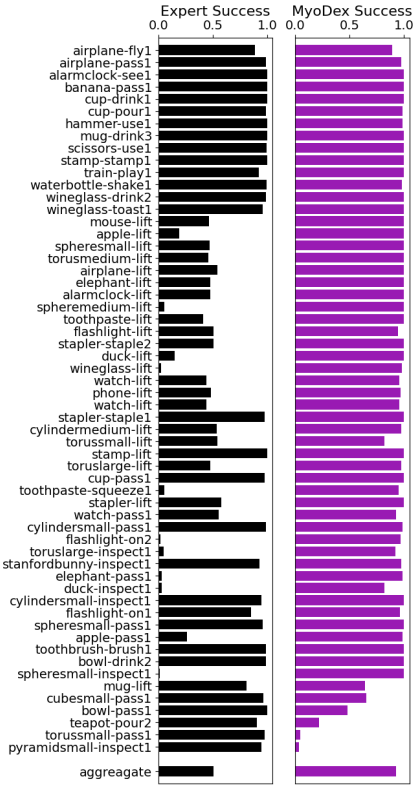


Figure 8: Summary of all Tasks. Left column tasks solved by single expert policies. Right columns, task fine tuning based on *MyoDex*. Aggregate success Expert vs *MyoDex* 0.51 vs 0.93. See also Table A.3.

REFERENCES

Mazen Al Borno, Saurabh Vyas, Krishna V Shenoy, and Scott L Delp. High-fidelity musculoskeletal modeling reveals that motor planning variability contributes to the speed-accuracy tradeoff. *eLife*, 9:e57021, December 2020. ISSN 2050-084X. doi: 10.7554/eLife.57021. URL https://doi.org/10.7554/eLife.57021. Publisher: eLife Sciences Publications, Ltd.

Vittorio La Barbera, Fabio Pardo, Yuval Tassa, Monica Daley, Christopher Richards, Petar Kormushev, and John Hutchinson. OstrichRL: A Musculoskeletal Ostrich Simulation to Study Bio-mechanical Locomotion. In *Deep RL Workshop NeurIPS 2021*, 2021. URL https://openreview.net/forum?id=7KzszSyQP0D.

Emilio Bizzi and Vincent CK Cheung. The neural origin of muscle synergies. *Frontiers in Computational Neuroscience*, 7, 2013. ISSN 1662-5188. doi: 10.3389/fncom.2013.00051. URL https://www.frontiersin.org/article/10.3389/fncom.2013.00051.

Samarth Brahmbhatt, Cusuh Ham, Charles C. Kemp, and James Hays. ContactDB: Analyzing and Predicting Grasp Contact via Thermal Imaging. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8701–8711, 2019.

Vittorio Caggiano, Vincent C. K. Cheung, and Emilio Bizzi. An Optogenetic Demonstration of Motor Modularity in the Mammalian Spinal Cord. *Scientific Reports*, 6(1):35185, October 2016. ISSN 2045-2322. doi: 10.1038/srep35185. URL https://doi.org/10.1038/srep35185.

Vittorio Caggiano, Huawei Wang, Guillaume Durandau, Massimo Sartori, and Vikash Kumar. MyoSuite – A contact-rich simulation suite for musculoskeletal motor control, May 2022. URL http://arxiv.org/abs/2205.13600. Number: arXiv:2205.13600 arXiv:2205.13600 [cs].

Tao Chen, Jie Xu, and Pulkit Agrawal. A System for General In-Hand Object Re-Orientation. *arXiv preprint arXiv:2111.03043*, 2021.

Vincent C. K. Cheung, Ben M. F. Cheung, Janet H. Zhang, Zoe Y. S. Chan, Sophia C. W. Ha, Chao-Ying Chen, and Roy T. H. Cheung. Plasticity of muscle synergies through fractionation and merging during development and training of human runners. *Nature Communications*, 11 (1):4356, August 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-18210-4. URL https://doi.org/10.1038/s41467-020-18210-4.

Dustin L. Crouch and He Huang. Musculoskeletal model predicts multi-joint wrist and hand movement from limited EMG control signals. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1132–1135, 2015. doi: 10.1109/EMBC.2015.7318565.

Sudeep Dasari, Abhinav Gupta, and Vikash Kumar. Learning Dexterous Manipulation from Exemplar Object Trajectories and Pre-Grasps, 2022. URL https://arxiv.org/abs/2209.11221.

Andrea d'Avella and Emilio Bizzi. Shared and specific muscle synergies in natural motor behaviors. *Proceedings of the National Academy of Sciences*, 102(8):3076–3081, February 2005. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.0500199102. URL https://pnas.org/doi/full/10.1073/pnas.0500199102.

Andrea d'Avella, Philippe Saltiel, and Emilio Bizzi. Combinations of muscle synergies in the construction of a natural motor behavior. *Nature neuroscience*, 6(3):300–308, 2003. Publisher: Nature Publishing Group.

Scott L. Delp, Frank C. Anderson, Allison S. Arnold, Peter Loan, Ayman Habib, Chand T. John, Eran Guendelman, and Darryl G. Thelen. OpenSim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement. *IEEE Transactions on Biomedical Engineering*, 54(11): 1940–1950, 2007. doi: 10.1109/TBME.2007.901024.

Nadia Dominici, Yuri P. Ivanenko, Germana Cappellini, Andrea d'Avella, Vito Mondì, Marika Cicchese, Adele Fabiano, Tiziana Silei, Ambrogio Di Paolo, Carlo Giannini, Richard E. Poppele, and Francesco Lacquaniti. Locomotor Primitives in Newborn Babies and Their Development. *Science*, 334(6058):997–999, 2011. doi: 10.1126/science.1210617. URL https://www.science.org/doi/abs/10.1126/science.1210617. _eprint: https://www.science.org/doi/pdf/10.1126/science.1210617.

Lucas Engelhardt, Maximilian Melzner, Linda Havelkova, Pavel Fiala, Patrik Christen, Sebastian Dendorfer, and Ulrich Simon. A new musculoskeletal AnyBody™ detailed hand model. *Computer Methods in Biomechanics and Biomedical Engineering*, 24 (7):777–787, 2021. doi: 10.1080/10255842.2020.1851367. URL https://doi.org/10.1080/10255842.2020.1851367. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10255842.2020.1851367.

Thomas Geijtenbeek, Michiel van de Panne, and A. Frank van der Stappen. Flexible muscle-based locomotion for bipedal creatures. *ACM Transactions on Graphics*, 32(6):1–11, November 2013. ISSN 0730-0301, 1557-7368. doi: 10.1145/2508363.2508399. URL https://dl.acm.org/doi/10.1145/2508363.2508399.

Leonard Hasenclever, Fabio Pardo, Raia Hadsell, Nicolas Heess, and Josh Merel. CoMic: Complementary Task Learning & Mimicry for Reusable Skills. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Nicolas Heess, Dhruva TB, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, S. M. Ali Eslami, Martin Riedmiller, and David Silver. Emergence of Locomotion Behaviours in Rich Environments, July 2017. URL http://arxiv.org/abs/1707.02286. arXiv:1707.02286 [cs].

Wenlong Huang, Igor Mordatch, Pieter Abbeel, and Deepak Pathak. Generalization in Dexterous Manipulation via Geometry-Aware Multi-Task Learning. *arXiv preprint arXiv:2111.03062*, 2021.

Aleksi Ikkala, Florian Fischer, Markus Klar, Miroslav Bachinski, Arthur Fleig, Andrew Howes, Perttu Hämäläinen, Jörg Müller, Roderick Murray-Smith, and Antti Oulasvirta. Breathing life into biomechanical user models. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, UIST '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393201. doi: 10.1145/3526113.3545689. URL https://doi.org/10.1145/3526113.3545689.

Divye Jain, Andrew Li, Shivam Singhal, Aravind Rajeswaran, Vikash Kumar, and Emanuel Todorov. Learning Deep Visuomotor Policies for Dexterous Hand Manipulation. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3636–3643, 2019. doi: 10.1109/ICRA.2019.8794033.

Marc Jeannerod. *The neural and behavioural organization of goal-directed movements*. Clarendon Press/Oxford University Press., 1988.

Yifeng Jiang, Tom Van Wouwe, Friedl De Groote, and C. Karen Liu. Synthesis of biologically realistic human motion using joint torque actuation. *ACM Transactions on Graphics*, 38(4):1–12, August 2019. ISSN 0730-0301, 1557-7368. doi: 10.1145/3306346.3322966. URL https://dl.acm.org/doi/10.1145/3306346.3322966.

Emanuel Joos, Fabien Péan, and Orcun Goksel. Reinforcement Learning of Musculoskeletal Control from Functional Simulations, July 2020. URL http://arxiv.org/abs/2007.06669. Number: arXiv:2007.06669 arXiv:2007.06669 [cs, eess].

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

Jong Hwa Lee, Deanna S. Asakawa, Jack T. Dennerlein, and Devin L. Jindrich. Finger Muscle Attachments for an OpenSim Upper-Extremity Model. *PLOS ONE*, 10(4):e0121712, April 2015. ISSN 1932-6203. doi: 10.1371/journal.pone.0121712. URL https://dx.plos.org/10.1371/journal.pone.0121712.

Seunghwan Lee, Ri Yu, Jungnam Park, Mridul Aanjaneya, Eftychios Sifakis, and Jehee Lee. Dexterous manipulation and control with volumetric muscles. *ACM Transactions on Graphics*, 37 (4):1–13, August 2018. ISSN 0730-0301, 1557-7368. doi: 10.1145/3197517.3201330. URL https://dl.acm.org/doi/10.1145/3197517.3201330.

Seunghwan Lee, Moonseok Park, Kyoungmin Lee, and Jehee Lee. Scalable muscle-actuated human simulation and control. *ACM Transactions on Graphics*, 38(4):1–13, August 2019. ISSN 0730-0301, 1557-7368. doi: 10.1145/3306346.3322972. URL https://dl.acm.org/doi/10.1145/3306346.3322972.

Libin Liu and Jessica Hodgins. Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. *ACM Transactions on Graphics*, 37(4):1–14, August 2018. ISSN 0730-0301, 1557-7368. doi: 10.1145/3197517.3201315. URL https://dl.acm.org/doi/10.1145/3197517.3201315.

Daniel C. McFarland, Benjamin I. Binder-Markey, Jennifer A. Nichols, Sarah J. Wohlman, Marije de Bruin, and Wendy M. Murray. A Musculoskeletal Model of the Hand and Wrist Capable of Simulating Functional Tasks. *bioRxiv*, pp. 2021.12.28.474357, January 2021. doi: 10.1101/2021.12.28.474357. URL http://biorxiv.org/content/early/2021/12/30/2021.12.28.474357.abstract.

Josh Merel, Yuval Tassa, Dhruva TB, Sriram Srinivasan, Jay Lemmon, Ziyu Wang, Greg Wayne, and Nicolas Heess. Learning human behaviors from motion capture by adversarial imitation, July 2017. URL http://arxiv.org/abs/1707.02201. Number: arXiv:1707.02201 arXiv:1707.02201 [cs].

Josh Merel, Leonard Hasenclever, Alexandre Galashov, Arun Ahuja, Vu Pham, Greg Wayne, Yee Whye Teh, and Nicolas Heess. Neural probabilistic motor primitives for humanoid control, January 2019. URL http://arxiv.org/abs/1811.11711. Number: arXiv:1811.11711 arXiv:1811.11711 [cs].

Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022. doi: 10.1126/scirobotics.abk2822. URL https://www.science.org/doi/abs/10.1126/scirobotics.abk2822. _eprint: https://www.science.org/doi/pdf/10.1126/scirobotics.abk2822.

M. Mirakhorlo, N. Van Beek, M. Wesseling, H. Maas, H. E. J. Veeger, and I. Jonkers. A musculoskeletal model of the hand and wrist: model definition and evaluation. *Computer methods in biomechanics and biomedical engineering*, 21(9):548–557, July 2018. ISSN 1476-8259 1025-5842. doi: 10.1080/10255842.2018.1490952. Place: England.

Anusha Nagabandi, Kurt Konoglie, Sergey Levine, and Vikash Kumar. Deep Dynamics Models for Learning Dexterous Manipulation, September 2019. URL http://arxiv.org/abs/1909.11652. arXiv:1909.11652 [cs].

Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, pp. 1101–1112. PMLR, 2020.

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. *arXiv preprint arXiv:1910.07113*, 2019.

Jungnam Park, Sehee Min, Phil Sik Chang, Jaedong Lee, Moonseok Park, and Jehee Lee. Generative GaitNet, January 2022. URL http://arxiv.org/abs/2201.12044. Number: arXiv:2201.12044 arXiv:2201.12044 [cs].

Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, Vikash Kumar, and Wojciech Zaremba. Multi-Goal Reinforcement Learning: Challenging Robotics Environments and Request for Research, 2018. _eprint: 1802.09464.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

Elmar Rückert and Andrea d'Avella. Learned parametrized dynamic movement primitives with shared synergies for controlling robotic and musculoskeletal systems. *Frontiers in computational neuroscience*, 7:138, 2013. ISSN 1662-5188. doi: 10.3389/fncom.2013.00138.

Tsuyoshi Saito, Naomichi Ogihara, Tomohiko Takei, and Kazuhiko Seki. Musculoskeletal Modeling and Inverse Dynamic Analysis of Precision Grip in the Japanese Macaque. *Frontiers in Systems Neuroscience*, 15, 2021. ISSN 1662-5137. doi: 10.3389/fnsys.2021.774596. URL https://www.frontiersin.org/articles/10.3389/fnsys.2021.774596.

Marco Santello, Martha Flanders, and John F. Soechting. Patterns of Hand Motion during Grasping and the Influence of Sensory Guidance. *The Journal of Neuroscience*, 22(4):1426, February 2002. doi: 10.1523/JNEUROSCI.22-04-01426.2002. URL http://www.jneurosci.org/content/22/4/1426.abstract.

Katherine R. Saul, Xiao Hu, Craig M. Goehler, Meghan E. Vidt, Melissa Daly, Anca Velisar, and Wendy M. Murray. Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model. *Computer Methods in Biomechanics and Biomedical Engineering*, 18(13):1445–1458, 2015. doi: 10.1080/10255842.2014.916698. URL https://doi.org/10.1080/10255842.2014.916698. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10255842.2014.916698.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, August 2017. URL http://arxiv.org/abs/1707.06347. Number: arXiv:1707.06347 arXiv:1707.06347 [cs].

Pierre Schumacher, Daniel Häufle, Dieter Büchler, Syn Schmitt, and Georg Martius. Dep-rl: Embodied exploration for reinforcement learning in overactuated and musculoskeletal systems, 2022. URL https://arxiv.org/abs/2206.00484.

Ajay Seth, Jennifer L. Hicks, Thomas K. Uchida, Ayman Habib, Christopher L. Dembia, James J. Dunne, Carmichael F. Ong, Matthew S. DeMers, Apoorva Rajagopal, Matthew Millard, Samuel R. Hamner, Edith M. Arnold, Jennifer R. Yong, Shrinidhi K. Lakshmikanth, Michael A. Sherman, Joy P. Ku, and Scott L. Delp. OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. *PLOS Computational Biology*, 14: 1–20, July 2018. doi: 10.1371/journal.pcbi.1006223. URL https://doi.org/10.1371/journal.pcbi.1006223. Publisher: Public Library of Science.

Anton R. Sobinov and Sliman J. Bensmaia. The neural mechanisms of manual dexterity. *Nature Reviews Neuroscience*, 22(12):741–757, December 2021. ISSN 1471-0048. doi: 10.1038/s41583-021-00528-7. URL https://doi.org/10.1038/s41583-021-00528-7.

Seungmoon Song, \Lukasz Kidziński, Xue Bin Peng, Carmichael Ong, Jennifer Hicks, Sergey Levine, Christopher G. Atkeson, and Scott L. Delp. Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation. *bioRxiv*, 2020. doi: 10.1101/2020.08.11.246801. Publisher: Cold Spring Harbor Laboratory _eprint: https://www.biorxiv.org/content/early/2020/08/12/2020.08.11.246801.full.pdf.

Seungmoon Song, Łukasz Kidziński, Xue Bin Peng, Carmichael Ong, Jennifer Hicks, Sergey Levine, Christopher G. Atkeson, and Scott L. Delp. Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation. *Journal of NeuroEngineering and Rehabilitation*, 18(1):126, December 2021. ISSN 1743-0003. doi: 10.1186/s12984-021-00919-y. URL https://jneuroengrehab.biomedcentral.com/articles/10.1186/s12984-021-00919-y.

Heiko Stark, Martin S. Fischer, Alexander Hunt, Fletcher Young, Roger Quinn, and Emanuel Andrada. A three-dimensional musculoskeletal model of the dog. *Scientific Reports*, 11(1): 11335, May 2021. ISSN 2045-2322. doi: 10.1038/s41598-021-90058-0. URL https://doi.org/10.1038/s41598-021-90058-0.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A Dataset of Whole-Body Human Grasping of Objects. volume 12349, pp. 581–600. 2020a. doi: 10.1007/978-3-030-58548-8_34. URL http://arxiv.org/abs/2008.11200. arXiv:2008.11200 [cs].

Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A Dataset of Whole-Body Human Grasping of Objects. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, pp. 581–600, Cham, 2020b. Springer International Publishing. ISBN 978-3-030-58548-8.

Pramodsingh H. Thakur, Amy J. Bastian, and Steven S. Hsiao. Multidigit Movement Synergies of the Human Hand in an Unconstrained Haptic Exploration Task. *The Journal of Neuroscience*, 28(6):1271, February 2008. doi: 10.1523/JNEUROSCI.4512-07.2008. URL http://www.jneurosci.org/content/28/6/1271.abstract.

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

Matthew C. Tresch, Vincent C. K. Cheung, and Andrea d'Avella. Matrix Factorization Algorithms for the Identification of Muscle Synergies: Evaluation on Simulated and Experimental Data Sets. *Journal of Neurophysiology*, 95(4):2199–2212, April 2006. ISSN 0022-3077, 1522-1598. doi: 10.1152/jn.00222.2005. URL https://www.physiology.org/doi/10.1152/jn.00222.2005.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Huawei* Wang, Vittorio* Caggiano, Guillaume Durandau, Kumar Sartori, Massimo, 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, 2022.

Jack M. Wang, Samuel R. Hamner, Scott L. Delp, and Vladlen Koltun. Optimizing locomotion controllers using biologically-based actuators and objectives. *ACM Transactions on Graphics*, 31(4):1–11, August 2012. ISSN 0730-0301, 1557-7368. doi: 10.1145/2185520.2185521. URL https://dl.acm.org/doi/10.1145/2185520.2185521.

Jungdam Won, Deepak Gopinath, and Jessica Hodgins. Control strategies for physically simulated characters performing two-player competitive sports. *ACM Transactions on Graphics*, 40(4):1–11, August 2021. ISSN 0730-0301, 1557-7368. doi: 10.1145/3450626.3459761. URL https://dl.acm.org/doi/10.1145/3450626.3459761.

Yuke Yan, James M. Goodman, Dalton D. Moore, Sara A. Solla, and Sliman J. Bensmaia. Unexpected complexity of everyday manual behaviors. *Nature Communications*, 11(1):3564, July 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-17404-0. URL https://doi.org/10.1038/s41467-020-17404-0.

Qi Yang, David Logan, and Simon F. Giszter. Motor primitives are determined in early development and are then robustly conserved into adulthood. *Proceedings of the National Academy of Sciences*, 116(24):12025–12034, 2019. doi: 10.1073/pnas.1821455116.

URL https://www.pnas.org/doi/abs/10.1073/pnas.1821455116. _eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1821455116.

Zhiqi Yin, Zeshi Yang, Michiel Van De Panne, and Kangkang Yin. Discovering diverse athletic jumping strategies. *ACM Transactions on Graphics*, 40(4):1–17, August 2021. ISSN 0730-0301, 1557-7368. doi: 10.1145/3450626.3459817. URL https://dl.acm.org/doi/10.1145/3450626.3459817.

A APPENDIX

Object	Weight [g]	behaviors
airplane	172	fly
airplane	172	pass
alarmclock	542	see
banana	277	pass
cup	300	drink
cup	300	pour
mug	432	drink
stamp	210	stamp
waterbottle	364	shake
wineglass	178	drink
wineglass	178	toast
train	400	play
hammer	210	use
scissors	47	use

Table A.1: Collection of Object, weight and tasks performed on.

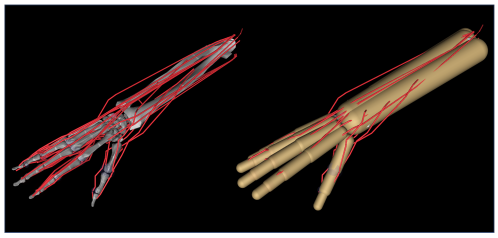
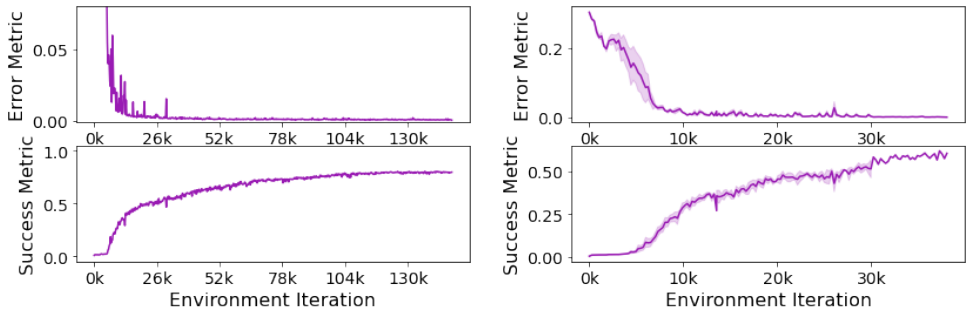


Figure A.1: Hand models. On the left, rendering of the musculoskeletal structure illustrating bone – in gray – and muscle – in red. On the right a skin like surfaces for soft contacts is overlaid to the musculoskeletal model.



(a) Success Metric

(b) Error Metric

Figure A.2: Training (rollout) of the multi-task policy i.e. jointly training on all tasks. Error metrics (top) and Success metrics (bottom) illustrate that multi-task training is very efficient at reducing smaller errors. Nevertheless, overall success is achieved more slowly than expert solutions. (a) long term training (b) magnification with average - continuous line - and standard deviation - shaded area - over 5 seeds.

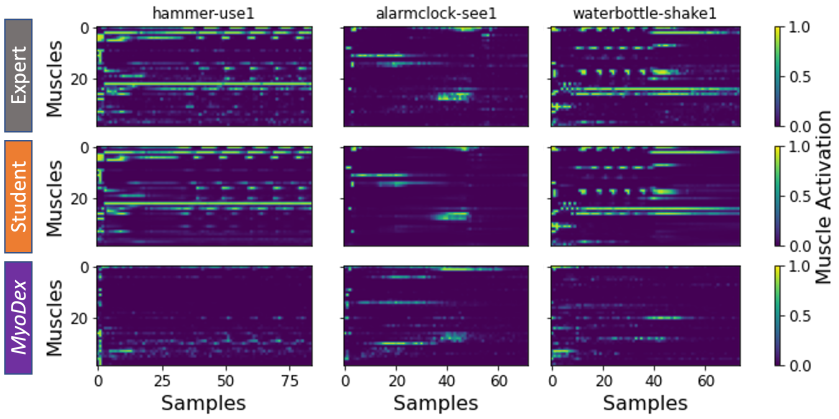


Figure A.3: Example of muscle activations. Expert (top) policies, student policies (middle) and multi-task/*MyoDex* (bottom).

.

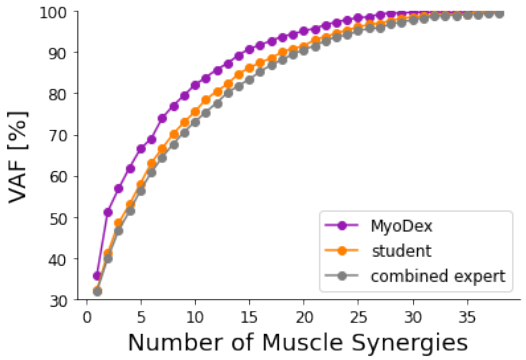


Figure A.4: Muscle Synergies. Number of muscle synergies as function of the explained variance (see Sec. 5.2) of the data shows that, given the same number of synergies, the multi-task learning can explain more variance of the data. Color coded experts (gray), student (orange) and *MyoDex*(purple).

A.1 Parameters of the neural network for the expert-student.

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 256, and Adadelta optimizer with a learning rate of 0.25, a Discount Factor (γ) 0.995, and 10 epochs.

Environment Iterations	$12k$
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.2: Parameters adopted for the reinforcement learning models.

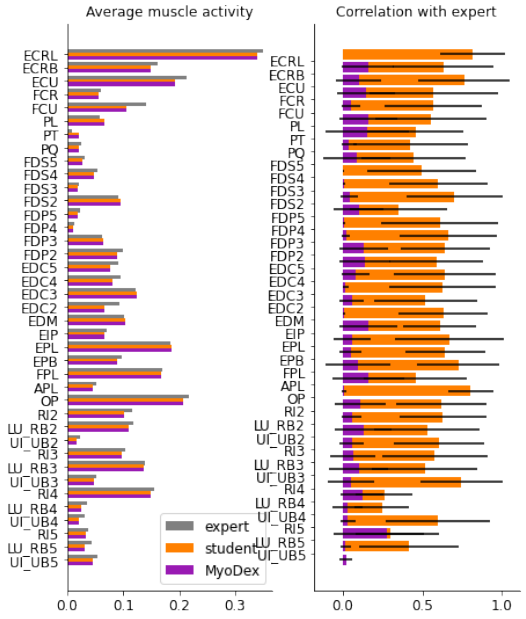


Figure A.5: Relationship between muscle activations. Left - Average Muscle Activation of experts (gray), student (orange) and *MyoDex*(purple). Right - correlation against the expert policies of student (orage) and *MyoDex* (purple).

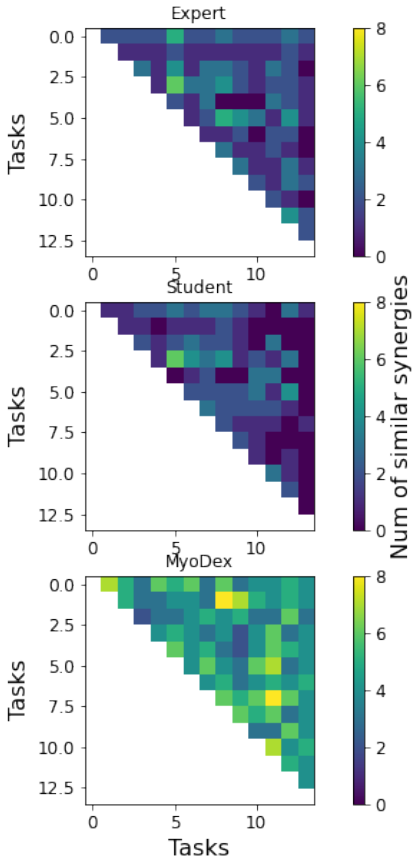


Figure A.6: Cosine Similarity between 12 synergies extracted from 14 different tasks at $37.5k$ iterations. Top - expert policies. Middle - student policy. Bottom – *MyoDex* policy. On average the number of similar synergies for expert, student, *MyoDex* (mean +/- std over 10 repetitions with different random seeds) was 1.88 ± 0.9, 1.45 ± 0.9 and 5.48 ± 0.17, respectively.

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	-
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	-
watch-lift	0.825778	0.953778	0.955556	988	-
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	-
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	-
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	-
duck-inspect1	0.621299	0.624935	0.820260	2964	-
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	-
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.3: Fine-tuning of 58 different tasks for *MyoDex* and expert agents. 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 column, document the iterations at which 0.80 success for *MyoDex* and experts was reached.

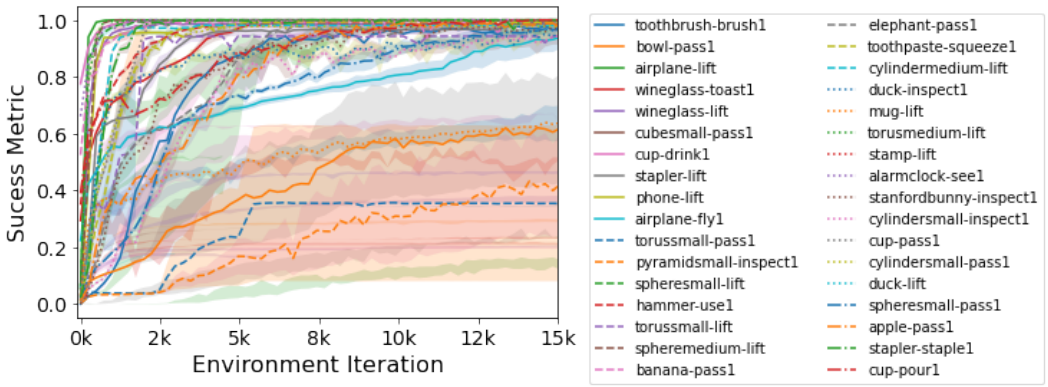


Figure A.7: Finetuning *MyoDex* on a large set of tasks. Success rate over iterations of finetuning *MyoDex*. Shaded areas indicate standard deviation over 3 seeds.

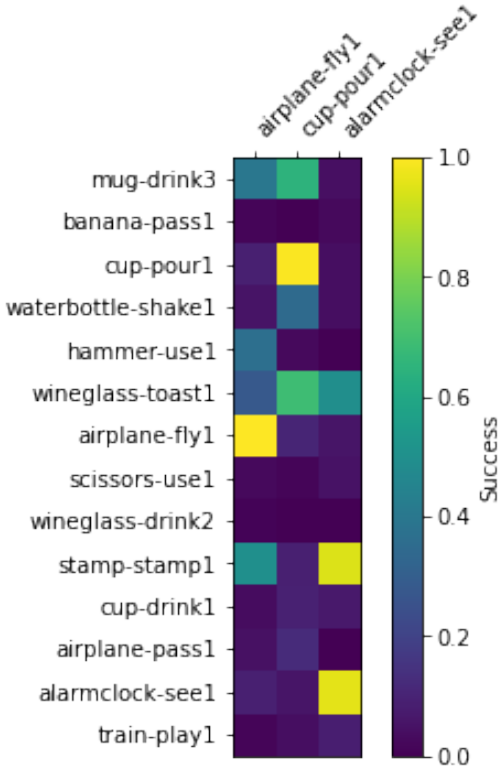


Figure A.8: Fine-tuning based on expert policies. Success rate fine-tuning experts solutions (columns) on 14 different environments. This matrix shows that the combination of pre-grasps and the initialization on a pre-trained task is not enough to generalize to new tasks.