Acquiring musculoskeletal skills with curriculum-based reinforcement learning

Alberto Silvio Chiappa[†,1,2], **Pablo Tano**[†,3], **Nisheet Patel**[†,3], **Abigaïl Ingster**[1,2], **Alexandre Pouget**[3], **and Alexander Mathis**[1,2]

[1]Brain Mind Institute, School of Life Sciences, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
[2]Neuro-X Institute, School of Life Sciences, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
[3]Department of Fundamental Neuroscience, University of Geneva, Geneva, Switzerland
[†]co-first authors; correspondence: alexander.mathis@epfl.ch

Efficient, physiologically-detailed musculoskeletal simulators and powerful learning algorithms provide new computational tools to tackle the grand challenge of understanding biological motor control. Our winning solution for the first NeurIPS MyoChallenge leverages an approach mirroring human learning and showcases reinforcement and curriculum learning as mechanisms to find motor control policies in complex object manipulation tasks. Analyzing the policy against data from human subjects reveals insights into efficient control of complex biological systems. Overall, our work highlights the new possibilities emerging at the interface of musculoskeletal physics engines, reinforcement learning and neuroscience.

Introduction

Motor control, the art of coordinating muscles to produce intricate movements, is a marvel of biological intelligence. From the graceful dance of a ballerina to the dexterous manipulation of objects, these movements are a testament to the brain's prowess of mastering numerous degrees of freedom in motor tasks (1–4) - which can take years of training and coaching to master, involving both explicit and implicit skill learning (5, 6). Yet, understanding how the brain achieves skilled behavior remains one of the fundamental challenges in neuroscience. While significant strides have been made, much of the research has been confined to relatively simple behavioral tasks (5–7). Moreover, computational modeling of motor control and learning is usually limited to simplified models of the musculoskeletal systems (e.g. (8–11)). As a result of these experimental and computational limitations, the complexities of more intricate skills like dexterous object manipulation remain largely uncharted.

Biomechanical simulators like OpenSim (12) have offered researchers many insights into musculoskeletal control (13, 14). While deep reinforcement learning (RL) recently succeeded in training control policies for complex tasks (15–21), the computational cost for combining it with biomechanics simulators like OpenSim has been a bottleneck (22, 23). MyoSuite (24), built on the efficient MuJoCo physics simulator (25), revolutionizes this space. It is not only vastly faster (by up to 4000x) than its predecessors (24), but also introduces intricate object manipulation tasks like Baoding balls, which has long fascinated motor control researchers due to its demand on coordinated and dynamic fine motor skills (3).

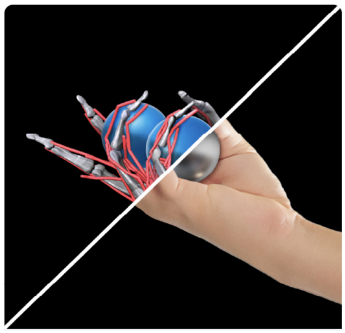
Our contribution in this article is twofold. First, we introduce a novel learning approach - the Static to Dynamic Stabilization (SDS) curriculum - drawing inspiration from human skill learning. Our approach, SDS, won the NeurIPS MyoChallenge in December 2022 for BoadingBalls (26); here we describe our solution in detail. Then, we analyze the learned policy, comparing and contrasting it with data from humans, unveiling the motor synergies and dynamics. Our work showcases that simulation-based approaches provide key insights into motor skill learning and control that complement experiments by providing a window into the representations underlying control, thereby allowing for detailed, high-throughput investigation across multiple scales of abstraction. With these advancements, we are better equipped to tackle deeper challenges in the realm of motor control, particularly the alignment between artificial and biological systems (27).

Results

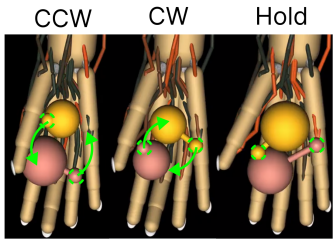
Task description. The Baoding balls task, as featured in the MyoSuite, offers a rigorous testbed that captures the essence of contact-rich motor control challenges (24, 26). It presents a biologically-realistic model of the human forearm, complete with a skeletal structure encompassing 23 joints that are actuated by 39 independent muscles, and two freely moving balls subject to physics (Fig. 1A). The objective of the task is to maneuver the pair of balls in the hand, making them rotate in tandem along a circular trajectory. Maximum reward is achieved when the controller can guide the balls to follow a pair of moving targets (small spheres in Fig. 1B).

The MyoChallenge unfolded in two distinct phases. In Phase I, only *counter-clockwise* (CCW) rotations had to be achieved. In contrast, Phase II introduced several layers of complexity. Not only did it add the *clockwise* (CW) rotation and *stay still* (Hold) conditions, but it also necessitated complex decision-making right at the episode's onset. The task's condition is not inherently evident from the observed variables; instead, it must be inferred by the agent. Adding to the challenge, the initial target position of the balls might not align with the actual position of the balls at the onset of an episode. This misalignment demanded strategic decisions to reach a high reward: whether to initiate the balls' rotation in the reverse or forward direction, when to switch directions, decelerate, or maintain position. Further amplifying the task's difficulty, Phase II introduced variability by randomizing parameters like the required targets' rotation radius and speed, as well as the balls' size, weight, and texture (friction). Each episode (in RL nomenclature, but more aptly described as 'trial' in Neuroscience) began with these parameters being randomly sampled from a predetermined range (Fig. 1B).

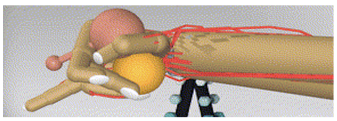
A The musculoskeletal hand



B Task variations

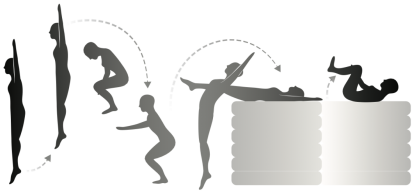


C Example failure modes

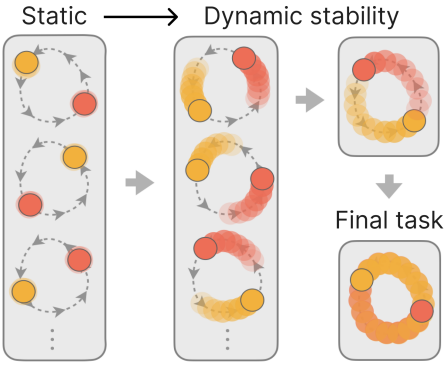


D States of a dynamic skill: the backflip

E Learning static movement motifs



F Schematic of the SDS curriculum



G Ablation results

Curriculum	Performance	
	Phase 1	Phase 2
None	41%	0%
Location only	42%	4%
Speed only	45%	0%
SDS *(ours)*	**100%**	**55%**

H Challenge leaderboard

Rank	Performance	
	Phase 1	Phase 2
1 *(ours)*	**100%**	**55%**
2	98%	41%
3	48%	15%
4	62%	14%
5	12%	3%

I

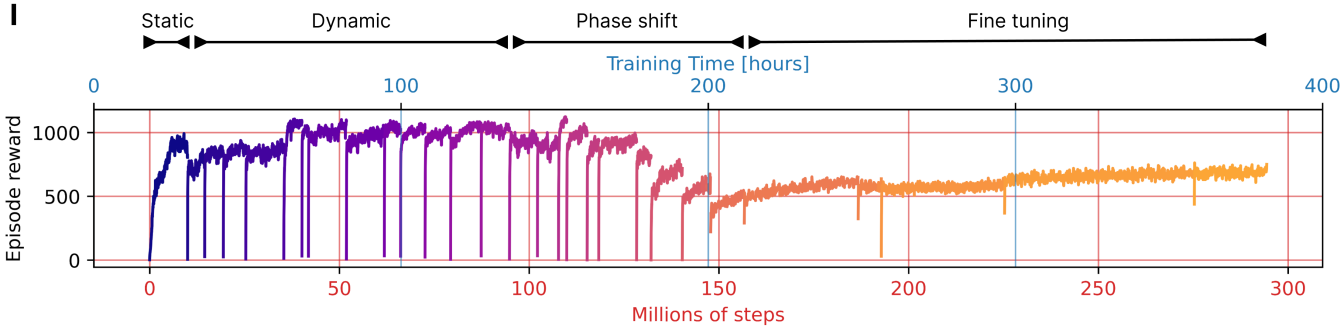


Fig. 1. Task definition, approach and benchmarking. **A** Schematic of the Baoding balls task with a musculoskeletal hand. Figure adapted from (24). **B** Task variations include direction of rotation (counter-clockwise (CCW, left), clockwise (CW, middle), and hold (Hold, right)) along with domain randomization, i.e., varying initial locations of targets (dotted green circles) as well as varying ball size, mass, and friction. **C** Visualization of a problematic local optimum during curriculum learning. **D** Illustration of the states traversed during a backflip - a complex, whole-body skill performed by humans as well as **E** Steps involved in a recommended training routine. Illustrations in D and E by Julia Kuhl inspired by backflip training materials (28). **F** Schematic of our static to dynamic stability (SDS) curriculum proposed to tackle the Baoding balls task by analogy to human skill learning (D-E). **G** Performance results from an ablation study demonstrate the necessity of the curriculum, especially in noisy environments with multiple tasks (Phase II). Performance is measured as the fraction of time steps in which the balls overlap with the targets. **H** Performance benchmarks and the MyoChallenge leaderboard (each row is a team). **I** Learning curve for the 32 curriculum steps, showing Episode reward vs. training time and environment interactions. Episode reward is measured as the total number of time steps in which the balls overlap with the targets. The first phases (Static and Dynamic) refer to the corresponding periods of the SDS curriculum. The second part of the training focused on dealing with the phase shift between the initial position of the Baoding balls and the targets and with the variable environment physics. Note that the maximum possible performance decreases during the Phase shift tasks (Phase II), as the targets do not overlap with the balls at the beginning of the episode.

The evaluation criterion of the task measures the fraction of time within an episode when the balls successfully trace the desired trajectory marked by the moving targets (see Methods).

Reinforcement and curriculum learning. Recently, deep RL algorithms made great strides in learning motor skills from scratch without a model of the environment dynamics (16, 17, 21). Such model-free RL algorithms have proven to scale robustly across increasingly complex environments. This robust scalability can be attributed to their effective use of powerful neural networks for processing high-dimensional data, alongside their high computational efficiency, broad applicability across tasks, and ease of 'out-of-the-box' imple-

mentation—advantages not as prevalent in more structured, model-based strategies. However, state-of-the-art model-free methods for RL faced significant challenges in solving the Baoding balls task, with proximal policy optimization (PPO) and a recurrent neural network architecture (Supp. Fig. 1) only reaching 41% performance in Phase I and 0% in Phase II (see Methods). The initial, sparse reward function proved insufficient for the agent to develop effective policies. Due to gravity and low friction, the balls easily rolled off the hand at the beginning of the episode, and the PPO agent was often not able to hold them at all. While adjusting the reward function to encourage long episodes provided an incentive to keep the balls on the palm (see Methods), the trained policy was highly susceptible to failure modes, like grabbing the balls to avoid them rolling off the hand (Fig. 1C), preventing any meaningful behavior from emerging. These optimization challenges are inherent to RL: as opposed to supervised learning, where the agent has access to the gradient of the task objective with respect to the control parameters, the RL agent must infer this gradient using the reward function, which makes them less sample-efficient and more prone to exploration-exploitation failures. Hence, enhancing the model-free optimization process is crucial, which lead us to focus on curriculum learning (29, 30).

Drawing parallels to human learning can provide valuable inspiration. Consider the complex skill of performing a backflip (Fig. 1D). For a novice, a backflip is a daunting sequence of (partially) unfamiliar states that must be seamlessly integrated into one fluid movement. Direct trial and error, or simply learning from one's mistakes, can be a dangerous and inefficient approach. Athletes often undergo a structured training regimen: they first familiarize themselves with the specific bodily states required by the skill (Fig. 1E). Once comfortable, they then piece these states together into a singular movement, initially slow and deliberate, eventually reaching full speed and fluidity (28).

Inspired by this human learning paradigm, we propose the Static to Dynamic Stabilization (SDS) curriculum for reinforcement learning. Analogous to the athlete's training, in SDS the RL agent is tasked to first learn to hold the balls statically in various configurations along the desired trajectory (Fig. 1F, left). Once the controller can stabilize those states, the agent is gradually trained to dynamically transition between them, creating a continuous movement that mirrors the desired trajectory (Fig. 1F, right). This curriculum not only addresses the exploration-exploitation dilemma but also provides the agent with a structured learning pathway, akin to the way humans approach complex motor tasks. In the final phases of the training we also performed domain randomization, i.e. varying the task parameters (see Methods). Overall, the SDS policy carried out 300 million environment interactions and it took almost 400 hours to train the model (Fig. 1I). Due to the curriculum, the training cannot be parallelized.

The SDS policy achieved a 100% score in Phase I (Video 1), and 55% score in phase II (Video 2). To the best of our knowledge, our demonstration provided the first successful example of fully-learned musculoskeletal control in a highly skilled object manipulation task.

Performance benchmarks. To showcase the importance of the SDS curriculum, we performed an ablation analysis, systematically stripping away key components of the curriculum to observe the resultant impact on learning efficacy. We evaluated three alternative training procedures, maintaining the network architecture and RL algorithm, but devoid of the full SDS curriculum (see Methods). We evaluated the performance of PPO without a curriculum (*None*, Fig. 1G); a curriculum that attempts to rotate the balls at full speed but departing from multiple initial locations (*Location only*, Fig. 1G); and a curriculum that gradually increases the target speed but departing from a unique initial location (*Speed only*, Fig. 1G). In the simpler Phase I, the performance drops by more than half upon removing either component (100% for SDS vs. 41-45% for *None*, *Location only*, *Speed only* curricula), whereas it all but entirely fails to learn anything in the more complex task of Phase II (55% for SDS vs. 0-4% for others). Thus the high performance of the SDS curriculum requires its two main components: learning multiple static configurations and gradually merging these configurations via increasingly faster dynamic trajectories.

To further contextualize our achievements, we also juxtaposed our performance against those of the top contenders of the competition (Fig. 1H). More than 40 teams took part and more than 340 entries were submitted (26). While most top solutions in the MyoChallenge incorporate RL with a curriculum, often complemented by reward shaping (26), it was our human-inspired SDS curriculum (Fig. 1I, Static and Dynamic) that distinguished our approach and elevated our results above the competition. Furthermore, we reviewed the literature with respect to new results since the December 2022 challenge: More recent exploration methods that do not use any kind of curriculum learning, such as generalized state-dependent exploration (31) and Lattice (32), marginally improve performance but fail to solve the full task (Table 4), further highlighting the necessity of the curriculum learning approach.

Analysis of the SDS policy.

Analysis of synergies. Mastering the many degrees of freedom inherent in motor control, often referred to as *Bernstein's problem* (1–4), is a central challenge in biological motor control. Having trained a policy network that solves the task with 39 muscles, we can interrogate the network state, muscle activations, and hand kinematics (Fig. 2A) exhibited during the task. How does the artificial agent compare to the intricate mechanisms of human motor control?

Classical studies on human subjects have illuminated a fascinating phenomenon: the vast realm of possible hand poses is often distilled into a more manageable, low-dimensional subspace (1–4). This dimensionality reduction, or the emergence of motor synergies, is seen as a strategic simplification

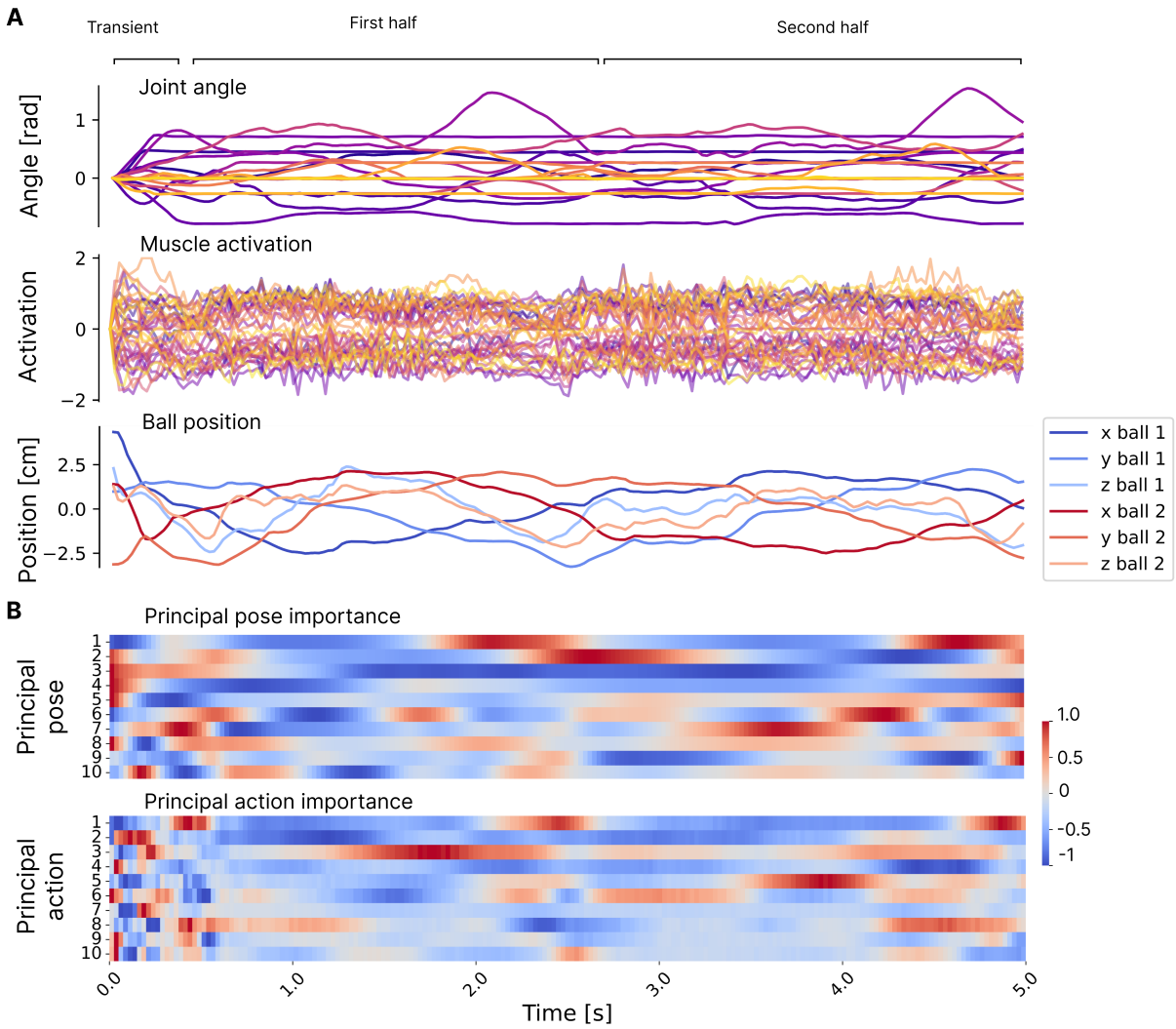
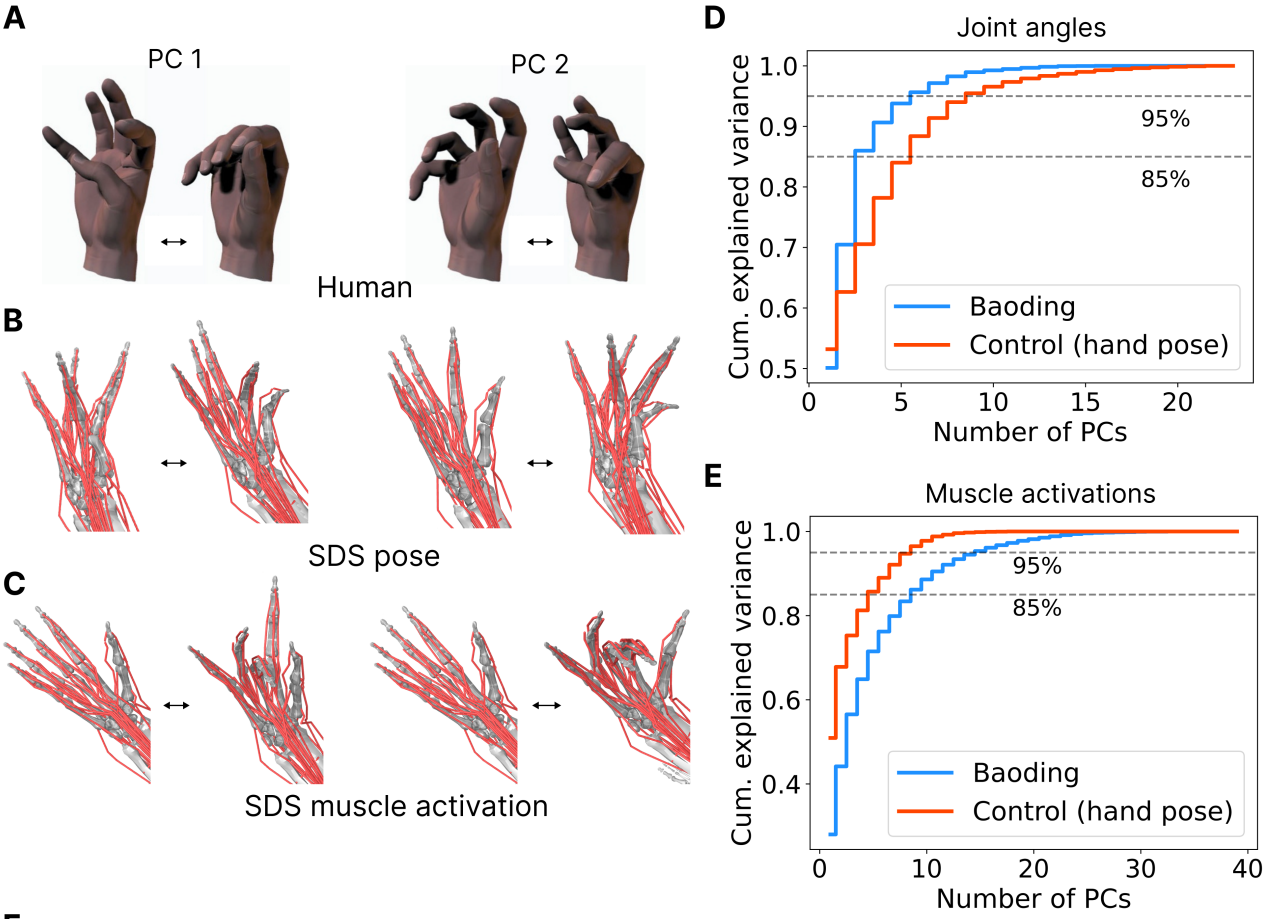


Fig. 2. Kinematic and dynamic motives within one Baoding balls rotation. A Time series of the hand kinematics (top), the muscle activations (middle), and the position of the Baoding balls (bottom), during a full rotation of both balls (5 seconds) for one episode of the SDS policy. **B** Relative importance of the 10 main principal components of the hand pose (top) and of the muscle activations (bottom), named *principal poses* and *principal actions*, as a function of the time step within a full rotation of both balls. The activations are averaged across all the episodes of the Baoding-SV dataset featuring a clockwise rotation (see Methods). Apart from the initial transient period, many components exhibit a periodic behavior, suggesting the emergence of motor primitives which are repeated in the first and the second half of the rotation. Indeed, a full Boading balls cycle is completed when the position of the two balls is swapped twice.

by the central nervous system to master the complexities of redundant biomechanical systems. We hypothesize that similarly to biological agents, artificial agents learn to operate in a reduced kinematic (pose) and dynamic (muscles) space too, learning a set of reusable motor primitives.

We begin by comparing the SDS policy with human movement data from Todorov and Ghahramani, who used a CyberGlove to capture hand movements during object manipulation tasks, including the Baoding balls task (3). Their approach, using Principal Component Analysis (PCA) on the hand kinematics, sought to unveil the number of motor synergies at play (see Methods). The CyberGlove, however, does not record any information about the muscle dynamics, which is instead fully accessible in the MyoSuite simulator. Therefore, we can analyze which activation patterns the SDS policy employs to control the hand, something that is more challenging to record in humans.

By projecting the hand poses (i.e., joint angles) and the policy's actions (i.e., muscle activations) onto the principal axes, we can gauge the significance of each principal component during the Baoding balls' rotation cycle. Comparing the top principal poses and actions between the First and Second half, we see that the same movement primitive is reused to swap the positions of the Baoding balls (Fig. 2B), indicating that the SDS policy has effectively generalized across similar phases of the task. To better grasp the meaning of the principal components of the hand pose, we can visualize them as shifts from the mean pose. Comparing the first two principal components computed from a human subject (Fig 3A) and from the SDS policy (Fig. 3B), the SDS policy's movements appear more rigid and localized to a subset of fingers. Furthermore, we can visualize the effect of individual principal components of the muscle activations by repeatedly applying them as actions, starting from an open hand position (Fig. 3C). In the SDS policy, the first principal action causes a



F

SDS

Human

	Position	Velocity	Muscle act.
Baoding	4.5	8.5	12
Control	8	5.5	7
Baoding	5	6	
Control	8.5	10	

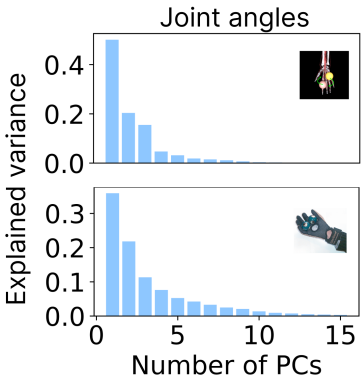


Fig. 3. Dimensionality of the control policy in the Baoding balls Task. **A** Pose variation corresponding to the first two principal components (PCs) of the hand poses recorded from humans rotating two Baoding balls (Figure adapted from (3)). **B** Same as A, but extracting the PCs from the hand poses of the SDS policy rotating the Baoding balls counter-clockwise. **C** Impact of applying the muscle activation patterns corresponding to the first two principal components of the control policy of the MyoChallenge competition for 15 steps (0.6 s), starting from an open hand position. **D** Cumulative explained variance of the PCs of the joint angles for an artificial agent trained to rotate two Baoding balls (SDS policy) and one trained to reach random poses (Control). For Baoding, the results in this plot (as well as in E and F) refer to the subset of episodes with counter-clockwise rotation direction. Also including the clockwise rotations gives similar results (Supp. Table 3). **E** Same as D, but the PCs are extracted from the agent's actions, corresponding to the muscle activations. **F** Left: Comparison between the number of independent degrees of freedom or *synergies* observed in the RL model (SDS) and experimental data (Human) (3) for the Baoding balls task as well as the control (hand/joint pose) task. Consistent with (3), the values are obtained by averaging the number of principal components necessary to explain 85% and 95% of the variance of the joint positions, the joint velocities and the muscle activations respectively. Right: Graphs of the explained variance for the first 15 principal components of the joint angles, for the SDS policy (top) and for a human subject (bottom, reproduced from (3)).

flexion of the middle finger and the thumb, and a wrist movement, while the second principal action orchestrates a unique flexion pattern of the thumb away from the palm and the other fingers towards it. The SDS policy primarily engages certain fingers, almost sidelining others (see videos 1 and 2). This contrasts with Todorov and Ghahramani's findings, where the first two principal components of the hand poses influenced all fingers (Fig.3A).

Unlike biological agents, the artificial policy has been optimized exclusively for the Baoding task, which might explain the more localized and rigid movements. However, despite the differences in the specific movement synergies used between the biological and artificial agents, having a successful artificial policy at our disposal allows us to study some of the core principles of motor control, which are possibly shared across the majority of musculoskeletal policies that can successfully perform object manipulation. For example, echoing the methodology in Todorov and Ghahramani, we estimated the dimensionality of the movements generated by SDS (Fig. 3D,E *Baoding*), as well as a policy on a control task, called Hand Pose (Fig. 3D,E, *Control*), designed to span the maximum dimensionality of the hand in joint-space. Namely, to solve the hand-pose task one needs to actuate the joints to random target hand postures (see Methods).

Our findings painted a compelling picture: the dimensionality of hand poses during Baoding balls rotation was consistently lower than during control kinematics, both in our RL controller and in Todorov and Ghahramani's experiments (Fig. 3F, *Position* and *Velocity* columns). This result also validates the experimental design of the Control task and suggests that object manipulation tasks, despite their complexity, tend to reuse familiar movement patterns, both in artificial and human subjects. The number of synergies in joint angle space computed for the simulated agent closely resembles the one measured for human subjects (Fig. 3F, Position column), an observation also supported by the fraction of explained variance of individual components (Fig. 3F, right).

However, a divergence between artificial and human subjects emerged when analyzing the angular velocities of the joints. The velocities from the simulated policy do not closely match the experimental data, both for Baoding balls and Control tasks. Specifically, the joint velocities produced by the Control policy are considerably lower-dimensional in simulation than in a human subject (Fig. 3F, *Velocity* column). This discrepancy likely stems from the fact that the Control policy attempts to reach the target pose as fast as possible, while human subjects are not instructed about the speed at which they have to move individual joints.

Task relevance of the low-variance PCs via "Virtual MuscleManifold Optogenetics". Diving deeper, we performed an analysis that is less feasible with biological subjects: probing the dimensionality of the control manifold in muscle-space. Strikingly, the dimensionality of the control policy in muscle-space showed a different pattern than in joint-space. Object manipulation demanded *more* degrees of freedom than pose reaching (12 vs. 7, Fig. 3F). This finding introduces a critical, yet previously overlooked perspective: assessing the complexity of control solely from motion capture observations might prove misleading. It is likely that this result, namely that the muscle-space dimensionality during free-hand movement is lower than during object manipulation, derives from the musculoskeletal structure of the hand, which has pairs of muscles antagonizing each other such that reaching desired hand positions requires a low-dimensional control signal over the muscles. In contrast, robustly manipulating objects like two Baoding balls, might require a more complex co-activation of antagonist muscles, leading to a higher effective muscle-space dimensionality.

It is tempting to categorize the high- and low-variance PCs as the task-relevant and task-irrelevant muscle co-activation manifolds, respectively. Indeed, projecting the hand kinematics on the first six principal components accounts for over 95% of the variance of the hand poses observed in the Baoding-SV dataset (Fig. 3). Yet, the role of low-variance (muscle) PCs in mastering the skill is unclear. With an experiment akin to selective activation and inactivation of individual neurons via optogenetics, we can remove the component of the control signal, which lies on a specific subspace of the muscle activation space. By projecting the muscle activation pattern output by the control policy onto the manifold spawned by a subset of the principal directions of variability, we force the control signal to lie in a lower-dimensional subspace. In this way, we can measure how the task performance varies as a function of the dimensionality of the enforced space.

We find that twelve principal components out of 39 retain 50% of the task performance, which saturates after 25 components are allowed (blue curve in Fig. 4A). Our analysis also revealed that low-variance principal components of the hand pose are task-dependent, as they retain discriminative power regarding what task is being performed (clockwise or counter-clockwise rotation in Fig. 4B). This result is consistent with the observations of Yan et al. (33), who showed evidence for task-relevance even in low-variance PCs for (primate) hand kinematics. However, with our computational model, we moved one step further, by quantifying the role of the low-variance (muscle) PCs of the control signal in terms of task performance, which is currently impossible to do experimentally. While the high-variance principal components are crucial for task performance (red curve in Fig. 4A), our findings suggest that a large subspace of the muscle activation space does not affect task performance (blue curve in Fig. 4A).

Analysis of the SDS policy's computation. The policy network, with its Long-Short Term Memory (LSTM) layer and two fully connected layers (Supp. Fig. 1), offers a unique lens into how information is transformed from the hand's proprioceptive state to the output control signal in terms of muscle activations. We embedded the activity of the policy network into a 3D space with UMAP (Fig. 4C, see Methods).

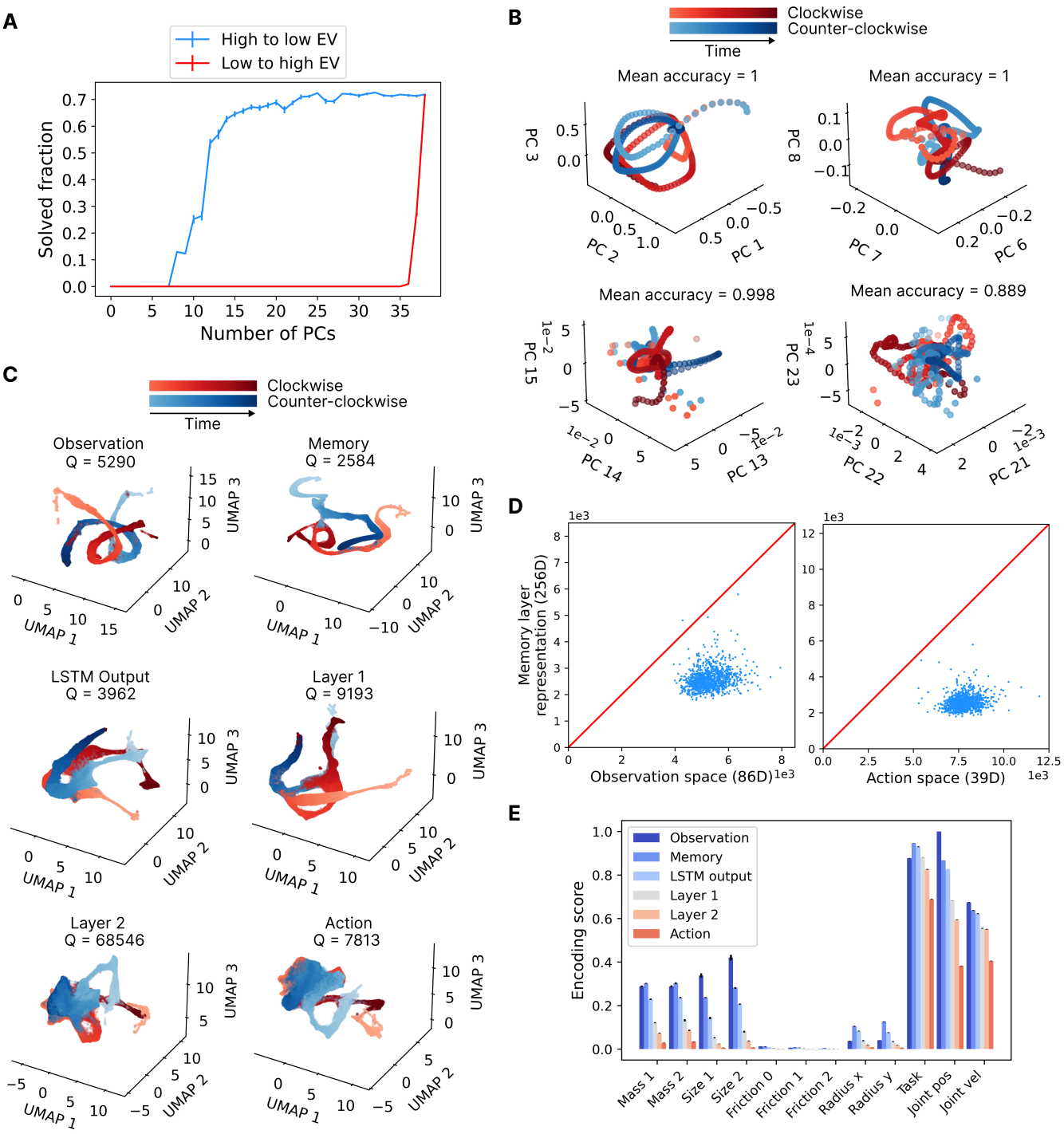


Fig. 4. Analysis of the computation performed by the policy network during the Baoding balls task. **A** Average performance of the policy (± s. e. m. over 100 episodes) when the actions are projected onto progressively larger subspaces of the action space, defined by the principal components. Blue curve: the principal axes are added from the most to the least important in terms of explained variance. Red curve: the principal axes are added from the least to the most important. **B** Average trajectory of the hand pose (500 episodes per rotation direction, see Methods) when projected onto different 3-dimensional subspaces defined by progressively less important principal components. The mean accuracy score refers to the decoding via logistic regression of the task direction from each of the 1000 episodes considered to obtain the average trajectory (see Methods). While the accuracy decreases with the importance of the PCs, one can also reliably decode the task from the least important ones. **C** UMAP embedding of the population activity of all the layers of the policy network (Supp. Fig. 1), extracted from the Baoding-SV dataset (see Methods). The rotational dynamics and the separation between clockwise and counter-clockwise rotation progressively disappear along the processing hierarchy; Tangling metric Q displayed. **D** Pair-wise comparison of the tangling of the population activity in the Memory layer versus the Actions (left) and the Observations (right). Each point indicates the average tangling of one episode (see Methods). The Memory layer consistently untangles the input Observation, while the deeper layers transform it into more tangled Actions. **E** Decoding analysis of the policy network, showcasing where task-relevant quantities are best encoded in each layer. Each time point is separated by 25 ms. The error bars indicate the standard deviation across 5-fold cross validation.

We quantified the degree to which the manifolds of the two tasks are entangled using Russo et al.'s tangling metric Q (see Methods). First, note that in the deeper layers (Layer 1 and Layer 2) and in the output (Action), the activity of the two

tasks (blue and red) is more tangled than in the input (Observation) and in the early layers (Memory and LSTM Output), indicating that common motor synergies (at the Action subspace) are being reused independently of the task. Furthermore, the manifolds of the two tasks are more separated in the Memory subspace than in the Observation subspace (compare top-right with top-left). Indeed, two states that are similar in the observation space, are less similar when integrating past information. Two states from different tasks that are similar in the observation space might require rather different motor plans, which forces the system to separate the manifolds of the two tasks in the memory space. Overall, we see that the system first distills the information from the observation space into two separate simple manifolds, effectively separating states that are similar in observation space but require different motor plans. Then, the manifolds merge back together towards the action space, as the system recruits similar motor synergies independently of the task (Fig. 4C).

The population activity of the Memory layer of the LSTM block presents the least tangled trajectories in the policy network. In particular, it is consistently less tangled than the action and observation dynamics (Fig. 4D) when carrying out the analysis in a per-episode manner. This result qualitatively resembles Russo et al.'s insight that motor cortical dynamics avoid tangling more than muscle dynamics (EMG) (34).

Finally, we sought to determine (if and) where in the network different task-relevant quantities are best encoded (Fig. 4E). While certain quantities, like joint angular positions, can be directly decoded from the observation, others, not part of the observation space, like the physical properties of the balls, are usually best decoded from the LSTM layer's memory vector. This suggests that the SDS policy accumulates data over time to form a representation of the system's non-observable state, which then conditions the control signal; several variables cannot be well decoded suggesting that they are implicitly represented.

Discussion

The intersection of efficient musculoskeletal simulators and powerful learning algorithms heralds a new era of understanding in the realm of biological motor control. By leveraging these models, we can delve deeper into the core principles of biological skill learning. This approach can not only offer insights into how animals incl. humans learn and execute complex motor tasks, but also generate both behavioral and neural predictions.

The SDS curriculum, inspired by human learning, emphasizes the significance of reinforcement and curriculum learning in training motor control policies for complex tasks. This not only showcases the potential of RL in developing high-fidelity sensorimotor models but also provides a platform to juxtapose artificial and biological control systems. Such comparisons, as we've demonstrated, can yield insights into emergent movement dynamics, offering a fresh perspective on motor learning and control.

Yet, as we push the boundaries of what's possible with digital simulations in deep RL, we are confronted with a deeper, more fundamental challenge: model alignment in sensorimotor systems. How do we bridge the gap between artificial systems and the intricate, naturalistic movements of a living organism? While supervised learning offers some solutions, allowing us to enforce specific movement repertoires, as demonstrated in previous models of the motor (35), visuomotor (36) and proprioceptive system (37, 38), RL presents a unique conundrum. Without explicit guidance, a model must *bootstrap* itself to discover viable solutions. Our SDS curriculum is more than just a technical achievement in that it is a step towards aligning artificial and biological motor control. Indeed, the SDS policy exhibits a number of properties that have been observed in primates. Firstly, we found that a low dimensional posture and control space, reminiscent in humans performing the Boading ball task (3). While the specific synergies are not the same, it is worth noting that the SDS policy has only been trained on this single task. Secondly, we found that the controller is robust to activity perturbations and that low-variance PCs still contain task-relevant signals, akin to (33). Thirdly, we found lower tangling of the dynamics in the learned controller than in the action space, akin to what Russo et al. found for the motor cortex vs. EMG dynamics (34).

Our policy analysis further illuminates the complexities of motor control. The emergence of motor synergies, as a strategic simplification by the central nervous system, is evident in both our artificial agent and human subjects. Yet, the dimensionality of control in muscle-space and joint-space offers a nuanced perspective, suggesting that assessing complexity solely from motion observations might be misleading. This highlights the importance of delving deeper into the underlying representations of control, beyond mere observations.

Furthermore, our analysis of the SDS policy's computation provides a unique vantage point into the transformation process of motor control. The LSTM layer's ability to integrate information over time to form a representation of the system's non-observable state is particularly intriguing. This suggests that the policy network is adept at accumulating and processing data to effectively condition the control signal.

However, the journey is not without its challenges. Despite all recent successes, optimization remains a key concern for deep learning, and while innovations such as LSTMs (39), ResNet's skipped-connections (40), reward shaping (41) and sensorimotor priors (42, 43) have made strides in addressing these challenges, the true potential of RL lies in its adaptability. Our SDS curriculum, inspired by human learning paradigms, curriculum learning (29, 30) and with elements of deliberate practice (44), exemplifies this adaptability, offering a structured learning pathway that aligns with biological motor control. The success of curriculum learning, mirrored

by other top-performing models for the 2022 MyoChallenge, underscores its efficacy (26). We note that, A.S.C and A.M. as well as Alessandro Marin Vargas won the MyoChallenge in 2023, with a solution also relying on curriculum learning, suggesting that currently, this is a necessary ingredient for mastering complex skills with reinforcement learning, despite recent advances in exploration for musculoskeletal systems (32, 45).

In conclusion, our work demonstrates the potential of highly realistic simulation-based approaches in motor control research. By bridging the gap between artificial and biological systems, we are poised to tackle deeper challenges in motor control, offering insights that complements experimental findings. As we continue to push the boundaries of musculoskeletal simulations in RL, the quest for model alignment in sensorimotor systems remains at the forefront, promising a future where the intricate movements of living organisms can be seamlessly replicated and understood in the digital realm.

Methods

Myosuite and Boading ball challenge. The model of the human forearm used to simulate the Baoding ball task is *myoHand* from the library MyoSuite (24). It is comprised of 29 bones, 23 joints and 39 muscle-tendon units. The interaction between the control policy and the MuJoCo physics simulator can be formulated as a Partially-Observable Markov Decision Process (POMDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{R}, \gamma, \rangle$. The process $\mathcal{M}$ is identified by the state space $\mathcal{S}$, the observation function $\mathcal{O}$, the action space $\mathcal{A}$, the transition function $\mathcal{T}$, the reward function $\mathcal{R}$ and the discount factor γ. The action space $\mathcal{A} \subset \mathbb{R}^{39}$ is the space of the possible activations of the 39 muscles controlling the human arm model. The observation function $\mathcal{O} : \mathcal{S} \rightarrow \mathbb{R}^{86}$ maps the state to the observation vector, which includes the angular position of each joint (23 elements), the positions (6 elements) and velocities (6 elements) of the two balls, the positions (6 elements) and the distances (6 elements) of the targets and the activation of each muscle at the previous time step (39 elements). The transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ maps a state and an action to a new environment state, defining how the environment evolves depending on the agent's decision. An agent seeks to maximize the discounted cumulative reward $R = \sum_{t=0}^{N} \gamma^t r_t$, where r_t is the reward at time step t, associated by the reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ to the transition from the state s_t to the state s_{t+1} via the action a_t.

The performance of the agent for the Boading ball challenge is evaluated in terms of the *solved fraction* SF, corresponding to the number of steps in which both balls are in proximity of the two moving target balls:

$$SF = \frac{\sum_{t=1}^{T} I_{x<\epsilon}(||\mathbf{p}_{B_1}^{(t)} - \mathbf{p}_{G_1}^{(t)}||) I_{x<\epsilon}(||\mathbf{p}_{B_2}^{(t)} - \mathbf{p}_{G_2}^{(t)}||)}{T}, \quad \textbf{(1)}$$

where T is the maximum number of timesteps in the episode, $I.(x)$ is the indicator function, ϵ is the distance threshold between the balls and the targets (1.5 cm), while $\mathbf{p}_{B_i}^{(t)}$ and $\mathbf{p}_{G_i}^{(t)}$, $i \in \{1, 2\}$ correspond to the position vector of the ball and the goal i at time step t, respectively.

Phase I of the MyoChallenge had the objective of rotating the two balls counter-clockwise, with a constant rotation period of 5 seconds. The simulation was chunked into episodes of 200 time steps, corresponding to 5 seconds of simulation. This means that the agent had sufficient time to complete a full rotation of the two balls, before the simulation terminated. An early termination condition was set to accelerate the training, when the balls fell below the palm. In this way, no computation was wasted simulating the environment with the balls in a hopeless position (namely, too far from the hand), which would not provide any useful experience for learning. At the beginning of every episode, the balls started from the same initial location, corresponding to the first target position.

Phase II of the MyoChallenge introduced many complications compared to Phase I. At the beginning of every episode,

a random task was sampled (clockwise or counter-clockwise rotation, holding the balls in place) and a target trajectory generated accordingly. Differently from Phase I, the starting point of the target trajectory might be different from the initial position of the balls. This meant that in Phase II it was impossible for the model to score 100% *solved*, because in most episodes the targets would spawn away from the initial position of the balls, and the model would require some time steps before being able to reach them. This complications were addressed with several intermediate curriculum steps (Fig. 1I, Phase shift). Besides the random task selection, further variability was provided by the randomization of the task parameters, namely: target rotation period, ellipse axes of the rotation trajectory, mass, size and texture of the balls. Each of these parameters was sampled independently, uniformly at random in a predefined range (Table 1). The MyoChallenge is also described in Caggiano et al. (26).

Reward engineering. The performance score (Eq. 1) provided a too sparse signal to be directly optimized via reinforcement learning, as an agent which had not yet learnt how to rotate the balls would almost always collect a vanishing score. For this reason, we designed a dense reward function $\mathcal{R} : \mathcal{S} \rightarrow \mathbb{R}$, which associates a meaningful performance score to all states. The reward was computed as a weighted sum of four values: the distance between each ball and the corresponding target (2 values), the indicator function representing whether both balls are at most 1.5 cm away from the respective target and the indicator function representing whether the balls are still on the palm. This last reward component proved fundamental for the agent to learn not to drop the balls, which would cause an early termination of the episode (and thus a lower cumulative reward). We assigned equal weight to all the reward components, penalizing the distance between the balls and the targets (weight: -1) and promoting the balls being on the palm and close to the targets (weight: 1).

Reinforcement learning details. We used the powerful on-policy RL algorithm PPO (16) from the Stable Baselines 3 library (46) with a recurrent architecture that has LSTM layers (39) in both the actor and critic, which allowed us to deal with the partially observable environment (Fig.1). The neural networks of the actor and the critic were implemented in PyTorch (47). The parameters of PPO and of the network architecture are listed in Table 2.

This policy was trained as a baseline. To win the challenge we developed the following curriculum learning strategy.

Static to dynamic stabilization curriculum. The schematic of our training curriculum, Static to Dynamic Stabilization (SDS), is illustrated in Fig. 1F. For clarity we only show the CCW trials, but the full curriculum includes CW and Hold trials. The key idea of the SDS curriculum is to learn stable postures at intermediate states along the desired trajectory *before* having to learn how to reach those states. The benefits of learning stable intermediate postures are twofold: (1) they serve as safe arriving and departing subgoals for an increasingly complex and unstable policy and (2) they robustly shape the value function of the RL agent such that it assigns a high value to these intermediate states, effectively acting as attractors for the final policy.

Parameter	Phase I	Phase II
Num episode steps	200	200
Drop threshold [m]	1.3	1.3
Target threshold (ϵ) [cm]	1.5	1.5
Radius x min [cm]	2.5	2
Radus x max [cm]	2.5	3
Radiux y min [cm]	2.8	2.2
Radius y max [cm]	2.8	3.2
Balls mass min [kg]	0.043	0.030
Balls mass max [kg]	0.043	0.300
Balls size min [cm]	2.2	1.8
Balls size max [cm]	2.2	2.4
Friction coeff. 1 min []	1	0.8
Friction coeff. 1 max []	1	1.2
Friction coeff. 2 min []	0.005	0.004
Friction coeff. 2 max []	0.005	0.006
Friction coeff. 3 min []	0.0001	0.00008
Friction coeff. 3 max []	0.0001	0.00012
Rotation period min [s]	5	4
Rotation period max [s]	5	6
Possible tasks	CCW	CCW, CW, Hold

Table 1. Parameters of the environments of Phase I and II.

The SDS curriculum imposes several subtasks that the agent must solve before moving to the next subtask. In the first subtask, the balls are initialized at random phases along the desired rotation cycle, and the goal of the agent is simply to hold them still at the initial position (first panel). In the following subtasks, the balls are also initialized randomly, but now the task of the agent is to move them following the desired trajectory, gradually increasing the target speed from subtask to subtask. As the curriculum advances and the targets speed up, at one point it is not beneficial to use random initialization anymore, as the policy can benefit from exploiting the inertia of the balls. At this point, SDS initializes the balls at the original initial position of the task (second to last panel) and continues speeding up the targets until it reaches the final task.

The code for training the policy including additional details is available at `https://github.com/amathislab/myochallenge`.

Dataset generation. For analyzing the SDS policy, we created two datasets including 1000 sessions each, in which the artificial agent trained for the MyoChallenge rotates the two Baoding balls for 5 seconds. In the first dataset, Baoding-SV (small variations), we sampled in each session the size and the mass of the balls within a small range of values, to introduce some variability in the task while maintaining consistent movements. This dataset is used for all the analyses illustrated in Fig. 2, Fig. 3 and in Fig. 4A-D.

Parameter	Before step 25	After step 25
Discount factor γ	0.99	0.99
GAE λ	0.9	0.95
Entropy regularization	3.62109e-6	3e-5
PPO clipping	0.3	0.2
Optimizer	Adam	Adam
Learning rate	2.6e-5	2.5e-5
Batch size	2048	65536
Minibatch size	32	1024
Max grad norm	0.835671	0.8
Num parallel envs	16	16
LSTM layer size	256	256
Layer 1 size	256	256
Layer 2 size	256	256
Activation	ReLU	ReLU

Table 2. Parameters of PPO and of the network architecture (the same parameters are used for both the actor and the critic). Some parameters changed starting from step 25 of the training curriculum.

The second dataset, Baoding-LV (large variations), features the same experimental conditions as the Phase II of the MyoChallenge. Baoding-LV was used to linearly decode the unobservable environment variables from the policy's neural population (Fig.4E).

Furthermore, we created a third dataset (Control) comprising 1000 episodes collected with a policy that also relies on PPO and recurrent networks, trained by Chiappa et al. (32) on the same musculoskeletal forearm model, but to reach a target pose (i.e., a random angular position of each joint per episode). The policy achieves a solved fraction of 68.3%, meaning that the hand successfully reaches the target pose more than 2/3 of the time. Data extracted from the Control dataset are utilized in Fig. 3D-F.

In each dataset, we recorded the joint angles, the joint velocities and the muscle activations with a frequency of 40 Hz. Furthermore, we recorded all the internal activations of the policy network, namely, the memory state and the output of the LSTM layer, and the output of the two fully connected layers (Fig. 4C). We validated the size of the dataset verifying that the dimensionality of the joint angle, joint velocity and muscle activation trajectories was stable when only using a subset of the data (Supp. Fig. 2).

Estimation of the number of motor synergies. Motor synergies emerge from the coordination between muscles (1–4). The correlation between muscle activations, hand poses or hand velocities can be quantified via principal component analysis (PCA). The first principal components capture the motor synergies that maximally explain the variability of the dataset. We applied PCA on the hand poses, hand velocities and muscle activations of the sessions included in the Baoding-SV and Control datasets. For the Baoding-SV dataset, we also analyzed how these quantities change when considering the sub-dataset of episodes where the balls rotate in the same direction (Table 3). For consistency with Todorov and Ghahramani, we calculated the number of degrees of freedom by averaging the number of principal components necessary to achieve an explained variance higher than 85% and 95%. The same steps were applied to the Control dataset, to estimate the motor synergies emerging from free hand motion (target pose reaching).

Table 3. Dimensionality of the joint angular positions (Position), joint angular velocities (Velocity) and actions (Muscle activation) when including the clockwise (CW), counter-clockwise (CCW) or both (Both) episodes of the Baoding-LV dataset (see Methods), for different thresholds of the explained variance.

	Position			Velocity			Muscle activation		
Exp. var. thresh.	CW	CCW	Both	CW	CCW	Both	CW	CCW	Both
0.85	4.0	3.0	4.0	6.0	6.0	6.0	9.0	9.0	9.0
0.95	6.0	6.0	7.0	10.0	11.0	11.0	15.0	15.0	15.0
Average	5.0	4.5	5.5	8.0	8.5	8.5	12.0	12.0	12.0

Decoding the rotation direction from the PCs of the hand pose. We assessed whether progressively less important principal components retain information about the rotation direction by decoding it via logistic regression. To this end, we considered four different subsets of three principal components of the hand pose, extracted from the Baoding-SV dataset (Fig 4B). One data point of the training dataset consisted in a flattened vector of three principal components of the pose for all the 200 time steps of one episodes (600 features in total). The mean accuracy was obtained by averaging the logistic regression score via 5-fold cross validation on random shuffles of the Baoding-SV dataset. We used the Python library Scikit-Learn (48).

Time-dependent importance of the motor primitives. The relative importance of the PCs of the hand pose and of the muscle activations varies throughout the rotation of the Baoding balls. To visualize at which point each component is more relevant, we considered the subset of the Baoding-SV dataset featuring clockwise rotations and averaged the hand poses and the actions. In this way we obtained the average hand pose and muscle activation across sessions at each time step. For clearer visualization, these coefficients were then rescaled between -1 and 1 (Fig. 2B). A full rotation of the Baoding balls corresponds to two periods in the hand pose and muscle dynamics, which are evident in several principal components. In fact, approximately the same movement primitives swap the first and the second ball and vice versa, to complete a rotation. The re-utilization of movement primitives is indicative that the RL policy has achieved appropriate generalization capabilities: if the identity of the balls is ignored in the input (which is irrelevant for task performance), the states at the beginning of the first and second half of the cycle are similar, so the movement primitives that achieve the desired goal should also be similar.

Visualization of the network activations. We embedded the observations, the actions and the network activations into a 3-dimensional space using Uniform Manifold Approximation Projection (UMAP) (49). Each point of the graphs of Fig. 4C is relative to a single time step.

Tangling of the population activity at variable depth. We quantified the tangling of the observations, actions and population activity of each layer of the SDS policy (Fig. 4) employing a metric introduced by Russo et al. (34). They propose to measure the time-dependent tangling of a trajectory $\mathbf{x} : \mathbb{R} \to \mathbb{R}^d$ with the scalar function $Q(t)$, defined as

$$Q(t) = \max_{t'} \frac{||\dot{\mathbf{x}}(t) - \dot{\mathbf{x}}(t')||^2}{||\mathbf{x}(t) - \mathbf{x}(t')||^2 + \epsilon}, \quad \textbf{(2)}$$

where $\dot{\mathbf{x}}(t)$ is temporal derivative of the trajectory, while ϵ is a small constant such that division by 0 is avoided (10^{-10} in our analysis). A single value per plot (Fig. 4C) is computed by averaging $Q(t)$ over M trajectories and N time steps:

$$Q = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=0}^{N} Q_{m_i}(t_j). \quad \textbf{(3)}$$

Decoding the environment variables from the population activity. Some unobservable (i.e., not part of the observation provided to the agent) parameters of the environment, such as the mass of the balls, have an impact on the dynamics of the environment and thus on the task performance. Such parameters are encoded with variable accuracy in the population activity of the different layers of the SDS policy. Using the Baoding-LV dataset, We computed the encoding score of each environment parameter proportionally to how well such parameter could be linearly decoded from each layer of the policy network. For continuous parameters (Mass, Size, Friction, Radius, Joint Positions, Joint Velocities) we considered the coefficient of determination R^2 of a linear regression, while for the only categorical parameter (Task) we considered the mean accuracy of a logistic regression. To compute robust scores and estimate their variability, we used 5-fold cross validation on random shuffles of the Baoding-LV dataset, using the Python library Scikit-Learn (48). The results of this analysis are illustrated in Fig. 4E.

Acknowledgments:

We thank members of the Mathis Group for helpful feedback. AM is appreciative to the Kavli Institute for Theoretical Physics (KITP) in Santa Barbara, where part of the manuscript was written.

Funding: AC and AM are funded by Swiss SNF grant (310030_212516). AI acknowledges a EPFL's Summer in the Lab fellowship to join the Mathis Group. AM was supported in part by grants NSF PHY-1748958 and PHY-2309135 to the KITP. PT and NP were supported by University of Geneva internal funding.

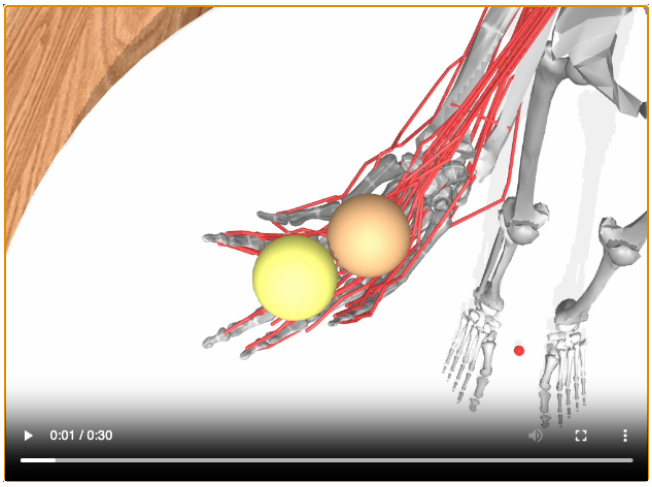
Declaration of interests: The authors declare no competing interests.

References

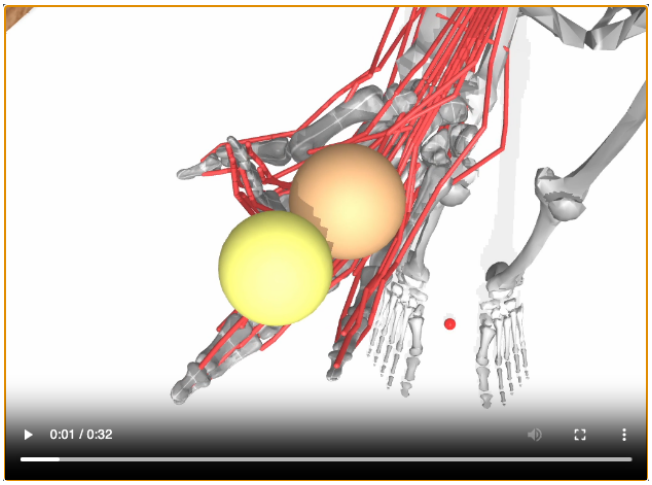
1. Michael T Turvey. Coordination. *American psychologist*, 45(8):938, 1990.
2. Andrea d'Avella, Philippe Saltiel, and Emilio Bizzi. Combinations of muscle synergies in the construction of a natural motor behavior. *Nature neuroscience*, 6(3):300–308, 2003.
3. Emanuel Todorov and Zoubin Ghahramani. Analysis of the synergies underlying complex hand manipulation. In *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 2, pages 4637–4640. IEEE, 2004.
4. Nicholai A Bernstein. *Dexterity and its development*. Psychology Press, 2014.
5. John W Krakauer, Alkis M Hadjiosif, Jing Xu, Aaron L Wong, and Adrian M Haith. Motor learning. *Compr Physiol*, 9(2):613–663, 2019.
6. Daniel M Wolpert, Jörn Diedrichsen, and J Randall Flanagan. Principles of sensorimotor learning. *Nature reviews neuroscience*, 12(12):739–751, 2011.
7. Mackenzie Weygandt Mathis and Alexander Mathis. Deep learning tools for the measurement of animal behavior in neuroscience. *Current opinion in neurobiology*, 60:1–11, 2020.
8. Emanuel Todorov and Michael I Jordan. Optimal feedback control as a theory of motor coordination. *Nature neuroscience*, 5(11):1226–1235, 2002.
9. Timothy P Lillicrap and Stephen H Scott. Preference distributions of primary motor cortex neurons reflect control solutions optimized for limb biomechanics. *Neuron*, 77(1):168–179, 2013.
10. Frédéric Crevecoeur, Stephen H Scott, and Tyler Cluff. Robust control in human reaching movements: a model-free strategy to compensate for unpredictable disturbances. *Journal of Neuroscience*, 39(41):8135–8148, 2019.
11. James B Heald, Máté Lengyel, and Daniel M Wolpert. Contextual inference underlies the learning of sensorimotor repertoires. *Nature*, 600(7889):489–493, 2021.
12. 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.
13. David A. Winter. *Biomechanics and motor control of human movement*. John Wiley & Sons, 2009.
14. Thomas K Uchida and Scott L Delp. *Biomechanics of movement: the science of sports, robotics, and rehabilitation*. Mit Press, 2021.
15. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.
16. John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
17. Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.
18. Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. In *International Conference on Learning Representations*, 2019.
19. Matthew Botvinick, Jane X Wang, Will Dabney, Kevin J Miller, and Zeb Kurth-Nelson. Deep reinforcement learning and its neuroscientific implications. *Neuron*, 107(4):603–616, 2020.
20. Tuomas Haarnoja, Ben Moran, Guy Lever, Sandy H Huang, Dhruva Tirumala, Markus Wulfmeier, Jan Humplik, Saran Tunyasuvunakool, Noah Y Siegel, Roland Hafner, et al. Learning agile soccer skills for a bipedal robot with deep reinforcement learning. *arXiv preprint arXiv:2304.13653*, 2023.
21. Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. *arXiv preprint arXiv:2310.12931*, 2023.
22. Florian Fischer, Miroslav Bachinski, Markus Klar, Arthur Fleig, and Jörg Müller. Reinforcement learning control of a biomechanical model of the upper extremity. *Scientific Reports*, 11(1):14445, 2021.
23. 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*, 18:1–17, 2021.
24. Vittorio Caggiano, Huawei Wang, Guillaume Durandau, Massimo Sartori, and Vikash Kumar. MyoSuite – a contact-rich simulation suite for musculoskeletal motor control, 2022.
25. Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ*, pages 5026–5033. IEEE, 2012.
26. Vittorio Caggiano, Guillaume Durandau, Huwawei Wang, Alberto Chiappa, Alexander Mathis, Pablo Tano, Nisheet Patel, Alexandre Pouget, Pierre Schumacher, Georg Martius, et al. Myochallenge 2022: Learning contact-rich manipulation using a musculoskeletal hand. In *NeurIPS 2022 Competition Track*, pages 233–250. Proceedings of Machine Learning Research, 2022.
27. Sébastien B Hausmann, Alessandro Marin Vargas, Alexander Mathis, and Mackenzie W Mathis. Measuring and modeling the motor system with machine learning. *Current opinion in neurobiology*, 70:11–23, 2021.
28. Backfliplab. https://backfliplab.com/backflip/standing_backflip.
29. Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.
30. Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576, 2021.
31. Antonin Raffin, Jens Kober, and Freek Stulp. Smooth exploration for robotic reinforcement learning. In *Conference on Robot Learning*, pages 1634–1644. PMLR, 2022.
32. Alberto Silvio Chiappa, Alessandro Marin Vargas, Ann Zixiang Huang, and Alexander Mathis. Latent exploration for reinforcement learning. *Advances in Neural Information Processing Systems*, 2023.
33. 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,

2020.
34. Abigail A Russo, Sean R Bittner, Sean M Perkins, Jeffrey S Seely, Brian M London, Antonio H Lara, Andrew Miri, Najja J Marshall, Adam Kohn, Thomas M Jessell, et al. Motor cortex embeds muscle-like commands in an untangled population response. *Neuron*, 97 (4):953–966, 2018.
35. David Sussillo, Mark M Churchland, Matthew T Kaufman, and Krishna V Shenoy. A neural network that finds a naturalistic solution for the production of muscle activity. *Nature neuroscience*, 18(7):1025–1033, 2015.
36. Jonathan A Michaels, Stefan Schaffelhofer, Andres Agudelo-Toro, and Hansjörg Scherberger. A goal-driven modular neural network predicts parietofrontal neural dynamics during grasping. *Proceedings of the national academy of sciences*, 117(50):32124–32135, 2020.
37. Kai J Sandbrink, Pranav Mamidanna, Claudio Michaelis, Matthias Bethge, Mackenzie W Mathis, and Alexander Mathis. Contrasting action and posture coding with hierarchical deep neural network models of proprioception. *Elife*, 12:e81499, 2023.
38. Alessandro Marin Vargas, Axel Bisi, Alberto Silvio Chiappa, Christopher Versteeg, Lee E Miller, and Alexander Mathis. Task-driven neural network models predict neural dynamics of proprioception. *bioRxiv*, pages 2023–06, 2023.
39. Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
40. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
41. Pierre Schumacher, Thomas Geijtenbeek, Vittorio Caggiano, Vikash Kumar, Syn Schmitt, Georg Martius, and Daniel FB Haeufle. Natural and robust walking using reinforcement learning without demonstrations in high-dimensional musculoskeletal models. *arXiv preprint arXiv:2309.02976*, 2023.
42. Alexei Koulakov, Sergey Shuvaev, Divyansha Lachi, and Anthony Zador. Encoding innate ability through a genomic bottleneck. *BiorXiv*, pages 2021–03, 2021.
43. Alberto Silvio Chiappa, Alessandro Marin Vargas, and Alexander Mathis. Dmap: a distributed morphological attention policy for learning to locomote with a changing body. *Advances in Neural Information Processing Systems*, 35:37214–37227, 2022.
44. Yue Du, John W Krakauer, and Adrian M Haith. The relationship between habits and motor skills in humans. *Trends in cognitive sciences*, 26(5):371–387, 2022.
45. Pierre Schumacher, Daniel Haeufle, Dieter Büchler, Syn Schmitt, and Georg Martius. Deprl: Embodied exploration for reinforcement learning in overactuated and musculoskeletal systems. In *The Eleventh International Conference on Learning Representations*, 2022.
46. 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.
47. Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
48. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12: 2825–2830, 2011.
49. Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Videos of the policies. Here we provide videos of several episodes in Phase I and Phase II.



Video 1. Here: A single frame a video of the policy which scored 100% in Phase I. The attached video shows a visualization of the hand and ball dynamics. Several episodes were concatenated. Video available at: https://youtu.be/RFIWLEUUqbw

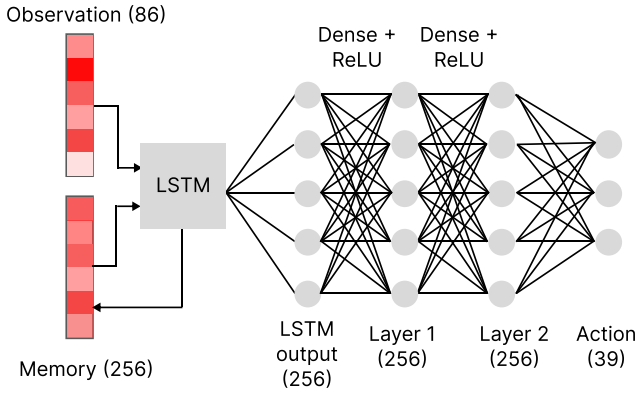


Video 2. Here: A single frame a video of the policy which scored 55% in Phase II. The attached video shows a visualization of the hand and ball dynamics. Several episodes with different hyperparameters and conditions (CCW, CW, hold) were concatenated. The sometimes visible small balls indicate the target position. When the Baoding balls are both close to the target positions, they turn brighter. The score is given by the fraction of time steps in which such condition is satisfied. Video available at: https://youtu.be/fF72wf0JK-4

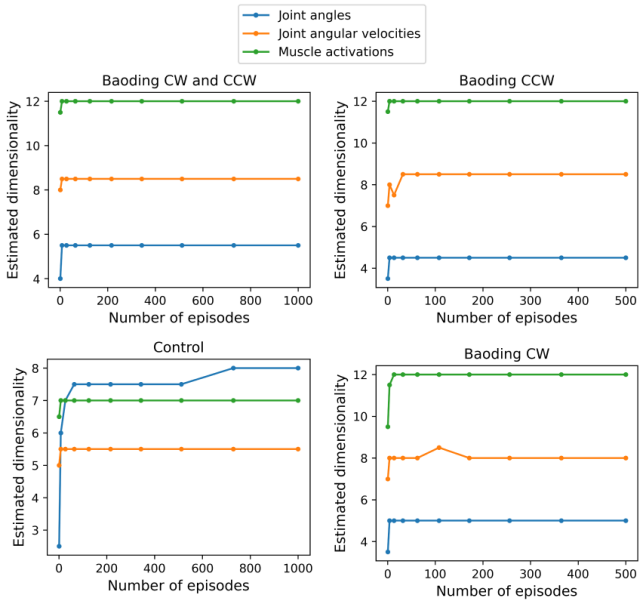
Supplementary Figures. Here we provide a diagram of the policy and critic network and a validation of the size of the dataset.

Table 4. Muscle effort, reward and solved fraction of different policies trained on Baoding Phase I with different exploration noises. Results are adapted from Chiappa et al. with Generalized State-Dependent Exploration (gSDE) described in Raffin et al.. The results are reported as mean ± s.e.m. across 5 random seeds.

	Baoding Effort	Reward	Solved
PPO	0.08 ± 0.00	573.46 ± 73.24	0.4 ± 0.07
PPO-gSDE period 4	0.07 ± 0.01	437.12 ± 91.91	0.26 ± 0.09
PPO-Lattice	0.04 ± 0.00	627.31 ± 69.01	0.44 ± 0.07
PPO-Lattice period 4	0.07 ± 0.01	471.16 ± 99.66	0.29 ± 0.1



Supplementary Figure 1. Architecture of the policy network. The observations are fed into a single LSTM layer, followed by two fully connected layers from where the action is predicted. Both the actor and the critic have the same architecture. The numbers between parentheses indicate the size of each layer.



Supplementary Figure 2. Dataset size validation. Number of independent degrees of freedom (as in Figure 3F) vs. dataset size (number of episodes). For all the quantities whose dimensionality we estimated in the analysis, the number of episodes in the Baoding-SV dataset is sufficiently large to achieve stable values.