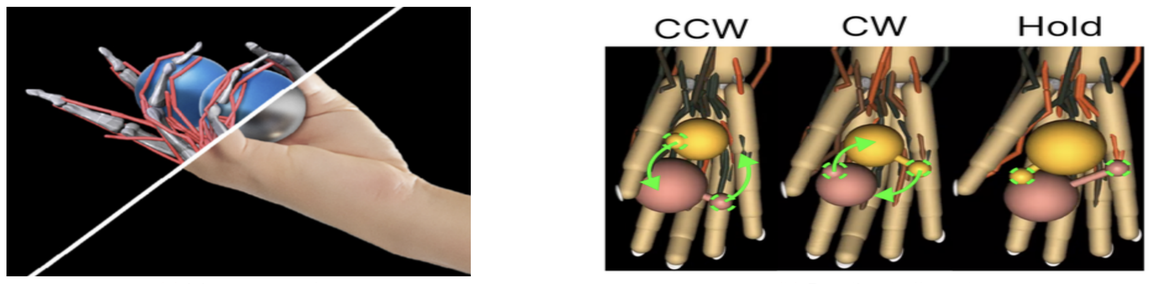
Exploring the transition from novice to expert in RL policies for motor skill

Oussama Gabouj, Ahmed Aziz Ben Haj Hmida, Salim Boussofara

Introduction

Understanding biological motor control is a major problem facing neuroscience today. The complex coordination of muscles required for tasks ranging from daily activities to athletic achievements demonstrates the remarkable capabilities of biological systems. Using computer-based tools like musculoskeletal simulators and reinforcement learning (RL) algorithms helps us learn about these processes and create better artificial motor control systems. Our project is based on the work of Chiappa, Tano, Patel et al., who introduced a novel curriculum-based RL method for motor control.



(a) Motor control (b) Baoding balls tasks

Figure 1. NeurIPS MyoChallenge for balls boading

Their Static to Dynamic Stabilization (SDS) curriculum emulates human learning by teaching an RL agent to stabilize static configurations before dynamic transitions, improving learning efficiency and performance. Chiappa et al.'s study suggests that combining physiologically-detailed simulators with RL algorithms can address complex motor control challenges. They also hypothesized that synergies can be extracted from artificial agents similar to biological muscles via dimensionality reduction in motor control. Understanding dimensionality of motor control tasks could lead to more efficient RL algorithms operating in reduced dimensional spaces, resulting in faster learning and better generalization.

Experimental Environment

The environment used for the experiments is the *Myosuite* baoding balls task which is a challenging motor control problem, divided into two distinct phases:

- **Phase I:** focuses on counter-clockwise rotations with fixed task parameters.
- **Phase II:** introduces additional complexities such as clockwise rotations, hold conditions, and random variations in task parameters like rotation period, ball size, and friction.

The interaction between the control policy and the MuJoCo physics simulator is formulated as a Partially-Observable Markov Decision Process, represented as $M = \langle S, A, O, T, R, \gamma \rangle$.

Environment variables

- **State Space (S):** The complete set of possible states of the system.
- **Observation Function (O):** Maps state to observation vector ($O : S \to \mathbb{R}^{86}$): 23 joint angles, 6 positions - 6 velocities (ball), 6 positions - 6 velocities (targets), 39 previous muscles' activations.
- **Action Space (A):** 39 muscle-tendon activations controlling human forearm model ($A \subset \mathbb{R}^{39}$).
- **Transition Function (T):** Defines how the environment evolves ($T : S \times A \to S$).
- **Reward Function (R):** Associates rewards with state transitions ($R : S \times A \times S \to \mathbb{R}$).
- **Discount Factor (γ):** Balances immediate and future rewards.

Methods

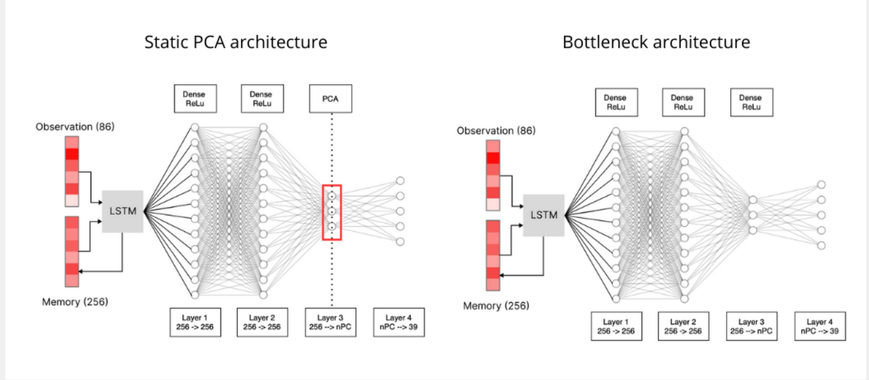
Analyze the components of motor synergies and understand how variance behaves in order to find ways to **distill expert motor strategies and effectively transfer them to novice agents** to achieve comparable performance in the second phase. The reinforcement algorithm used to train the novice agent is the **recurrent proximal policy optimization** which balances exploration and exploitation effectively.

- **Train an Agent on Phase I using phase II architecture → Baseline**: The training starts with the agent having no prior knowledge, learning solely from rewards received during interactions.
- **Extract PCs of the Expert Agent from Phase II:** PCA is performed on expert agent's features and actions from Baoding task (Phase II) to identify key components and underlying motor synergies.
  - **Feature Space**: Extract PCs from the last hidden layer and map into a smaller feature space.
  - **Action Space**: Extract PCs from the action space (i.e muscle activations space).
- **Policy Distillation Strategies** 2 strategies to explore transition from novice to expert in RL policies for motor skill: reducing observation mapping space or action space dimensionality.

Policy Distillation Strategies

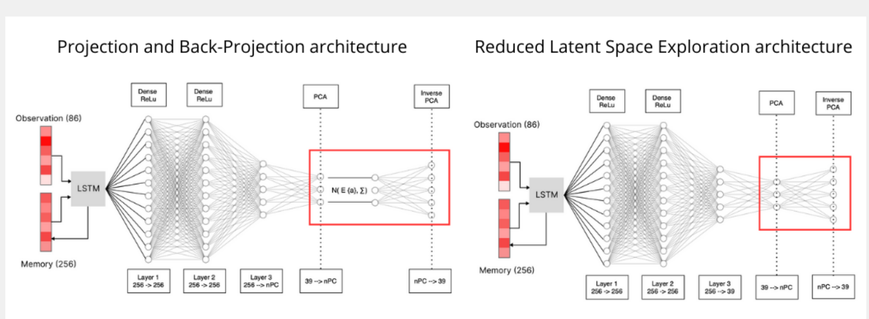
1- Reduce observation mapping space guiding agent to take best actions. Constrain agent's exploration for more efficient learning. Address curse of dimensionality, simplifying observation mapping space.

Reduced observation mapping space:



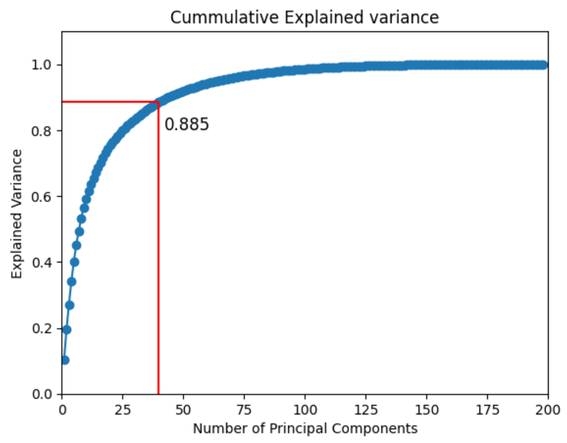
2- Reduce the action space dimensionality. Constrain the agent's exploration so that it only explores the most probable action space derived from the expert agent's experience.

Action Space Dimensionality Reduction:



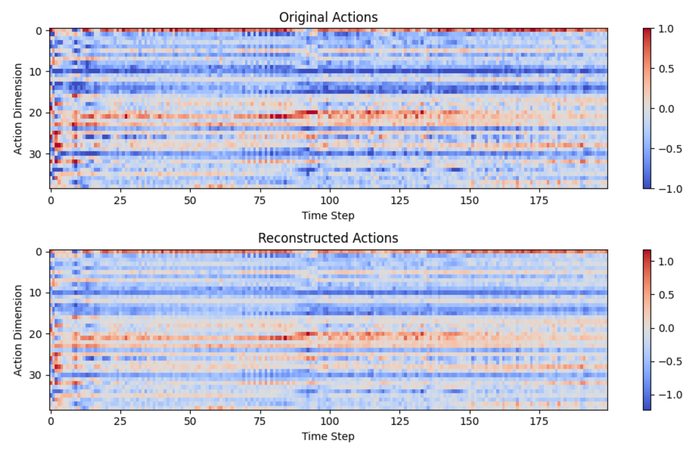
Results Analysis

PCA Analysis:



A suitable number of extracted PCs is 16 in action space and 40 in feature space, as their cumulative explained variance is greater than 0.9 and 0.88, respectively. 16 components are effective in representing action space with a high level of variance retention. For the feature space, achieving a cumulative variance of over 0.9 requires adding at least 10 components, covering only 0.02 of the variance. 40 components maintain a higher number of features compared to subsequent layer's 39 action features, ensuring that PCA doesn't overly simplify the feature space prior to its usage in further analyses.

PCA-based reconstruction for action space captures essential patterns effectively, maintaining a coherent visual representation of the data after reconstruction. Despite some loss in detail, the major variations are well-represented and key characteristics are preserved. The smoother transitions in the reconstructed actions reflect a successful dimensionality reduction, simplifying data while retaining its critical structure.



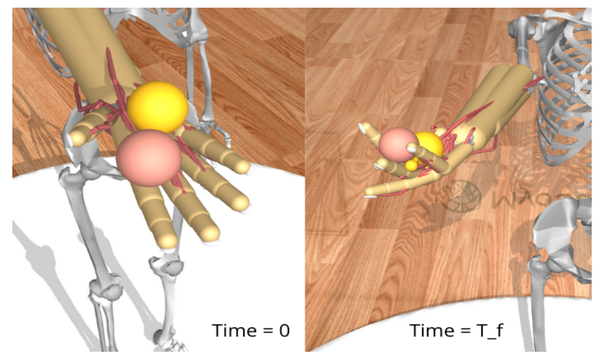
Evaluation of experiments:

All 4 models were evaluated under stochastic conditions. Mean Reward indicates the average reward achieved while training. Mean Episode Length measures the duration before failure (balls dropping).

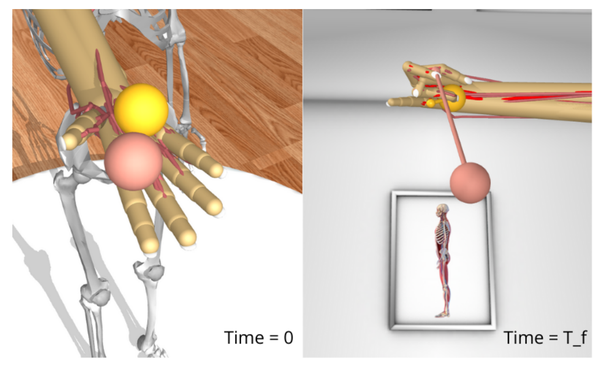
Model	Exp 1	Exp 2	Exp 3	Exp4	Expert 2 Baseline
Mean Reward	485	321	41	39	990.4
STD Reward	7.3	7.8	2	2.1	-
Mean Episode Length	121.4	111.3	22.5	21.5	200
STD Episode Length	3.5	4.2	1.5	0.9	-

Table 1. Summary of Neural Network Experiment Results

Experiment 1/2: The first two experiments demonstrated significantly higher rewards and episode lengths. Agents were at least able to maintain control over the task (holding the balls) for longer periods, which is the initial step of the curriculum learning SDS according to the foundational project paper, although they struggled with more complex manipulations (rotating the balls), as indicated by skeletal simulations.



Experiment 3/4: Significantly lower rewards and episode durations in the last two experiments suggest difficulties in basic task retention (holding the balls) and the drastic reduction in feature and action spaces was too severe, omitting necessary information for effective decision-making. Although exploring the reduced feature space was a logical step in experiment 4, results suggest that simply spending more time within this space didn't compensate for the loss of critical information due to excessive compression.



Discussion and Conclusion

Evaluating Task-Specific Performance Through PCA

- The reconstruction from reduced spaces maintains basic actions but smooths out intensity of muscle activation, causing agent to struggle to fully execute all tasks within restricted space. While the agent learns to hold the balls, a task not requiring fine variations, it fails to rotate the balls, a task demanding maximal muscle extension. This underscores the role of selecting an appropriate number of PCs.

- Given Experiment 1's demonstrated capability to achieve a reward of 400 just by holding the balls, we suggest a potential to sequentially train the model on more complex tasks: ball rotation and baoding.

Optimizing Dimensionality Reduction Strategies for Enhanced Model Training

- The number of PCs was selected to maintain high explained variances. However low-variance PCs indicate some task-specific features that could be essential for efficient learning. It could, therefore, be beneficial to find wiser ways to extract the PCs.

- PCA usage may need reconsideration. Although it is effective for reducing dimensionality by capturing maximum variance, it might not be the most suitable method for tasks that involve complex, dependent actions like those our model is trained on. Alternative techniques might be more appropriate.

- Experimenting with re-positioning PCA layer within the NN architecture might yield improvements. Shifting PCA layer to different points between other layers could potentially optimize agent learning.

References

- Nisheet Patel Abigaïl Ingster Alexandre Pouget Alexander Mathis Alberto Silvio Chiappa, Pablo Tano. Acquiring musculoskeletal skills with curriculum-based reinforcement learning. 2024.
- Prafulla Dhariwal Alec Radford Oleg Klimov John Schulman, Filip Wolski. Proximal policy optimization algorithms. 2017.
- Frank Zimmer Mike Preuss Marco Pleines, Matthias Pallasch. Generalization, mayhems and limits in recurrent proximal policy optimization. 2022.