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

Oussama Gabouj
oussama.gabouj@epfl.ch
Ahmed Aziz Ben Haj Hmida
ahmed.benhajhmida@epfl.ch
Salim Boussofara
salim.boussofara@epfl.ch

Abstract

This study investigates the use of reinforcement learning for motor control, building on the curriculum-based RL approach introduced by Chiappa et al [1]. It explores the evolution from novice to expert reinforcement learning policies in motor skill acquisition, focusing on the manipulation of Baoding balls, which is a task requiring complex motor skills. We employ recurrent PPO [2] with LSTM layers to handle partial observability. Our study, also, simplifies the learning process by distilling complex expert strategies into novice agents using various dimensionality reduction techniques involving both static and dynamic reductions of observation and action spaces with the aim to find a balance that maintains essential information while enhancing learning efficiency. This report provides an overview of the methods and insights gained into the effectiveness of dimensionality reduction in training RL agents for intricate motor control tasks.

1 Introduction

1.1 Context and motivation

Understanding biological motor control is a major problem facing neuroscience today. The complex coordination of muscles required for various tasks ranging from daily activities to athletic achievements demonstrates the remarkable capabilities of biological systems. Using computer-based tools like musculoskeletal simulators and RL algorithms can help us learn about these processes and create better artificial motor control systems.

1.2 Background

Our project build on the foundational work achieved by Chiappa et al. [1]. They presented a new way of using curriculum-based RL for motor control. Their method, the Static to Dynamic Stabilization (SDS) curriculum, won the NeurIPS MyoChallenge for its effectiveness in training a model to manipulate Baoding balls, a task requiring complex motor skills. This innovative approach mirrors human learning by progressively teaching an RL agent to stabilize static configurations before moving on to dynamic transitions, thereby enhancing learning efficiency and performance. The study by Chiappa et al. [1] highlights the potential of combining physiologically-detailed simulators with powerful RL algorithms to tackle complex motor control challenges.

An important aspect of their work is to verify the hypothesis that it is possible to extract synergies from artificial agents as in biological muscles via dimensionality reduction in motor control. Indeed, The human motor system has numerous degrees of freedom, making the control problem highly

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

complex. The SDS curriculum implicitly reduces this complexity by breaking down the learning process into manageable stages. However, understanding the intrinsic dimensionality of motor control tasks can further enhance learning efficiency. Identifying the key components or synergies that govern effective motor control could help to develop more efficient RL algorithms that operate in a reduced dimensional space, leading to faster learning and better generalization.

2 Experimental environment

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

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

The agent receives a comprehensive set of observations, described by an 86-dimensional observation vector, which provides a detailed representation of the system's state. This vector includes 23 joint angles, positions and velocities of the balls, positions and distances of the targets and previous activations of the 39 muscles. These observations enable the agent to understand the current state and make informed decisions about subsequent actions. Indeed in response to the observations, the agent generates actions within a 39-dimensional action space, corresponding to the activations of the muscle-tendon units controlling the human forearm model. These actions are influenced by the internal state representations formed by the LSTM layers, which accumulate information over time.

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

- **State Space (S):** The complete set of possible states of the system.
- **Observation Function (O):** Maps the state to an 86-dimensional observation vector ($O : S \to \mathbb{R}^{86}$).
- **Action Space (A):** Represents the possible muscle activations ($A \subset \mathbb{R}^{39}$).
- **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.

3 Methodology

The primary task of our project is to analyze the components of motor synergies and understand how variance behaves with the aim 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. The PPO configuration includes a recurrent neural network architecture with LSTM layer to handle the partial observability of the environment.

To analyze the motor synergies of the novice and the expert agents, we conducted a series of structured experiments.

3.1 Train an agent on phase I using the architecture used for phase II

To ensure consistency, we began by training an agent on Phase I of the Baoding balls task using the architecture designed for Phase II. The training process starts with the agent having no prior knowledge, learning solely from rewards received during interactions. The agent trains in the Phase I environment, focusing on counter-clockwise rotations of the Baoding balls over multiple episodes, each lasting 200 time steps (5 seconds). Early termination is used if the balls fall below the palm to focus learning on productive interactions. The agent's performance is based on rewards for maintaining the balls' proximity to the target trajectory, with cumulative rewards updating the policy

through PPO. Using the advanced Phase II architecture in Phase I training is expected to improve learning efficiency and performance, providing a baseline for further analysis.

3.2 Extract the principal components of the expert agent (Phase II)

Next, we extracted the PCs from the expert agent trained in Phase II to identify key components driving effective performance and understand underlying motor synergies. This step is foundational for the subsequent distillation processes, as it highlights the most significant features and actions that contribute to successful task execution. PCA is performed on the expert agent's features and actions from Phase II of the Baoding balls task.

- **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 muscles activation's space) and map into a smaller action space.

Using an analysis of the cumulative explained variance, which we will discuss later in the report, we decomposed the high-dimensional feature and action spaces of the expert agent into a small set of orthogonal components capturing the most of the variance in the agent's behavior. This allows us to identify the key factors of the expert's performance, representing motor synergies [3] and efficient strategies that can be transferred to novice agents. This understanding helps in reducing the dimensionality of observation and action spaces, combining hand movements and simplifying the learning process for novice agents.

3.3 Policy distillation strategies

To explore the transition from novice to expert in RL policies for motor skill acquisition, we employ two primary strategies for policy distillation: reducing the observation mapping space and reducing the action space dimensionality.

3.3.1 Feature space dimensionality reduction

In the first part, we focus on reducing the observation mapping space to guide the agent in taking the best possible actions. By reducing the dimensionality of the observation space, we aim to constrain the agent's exploration, helping it to learn more efficiently. This approach addresses the curse of dimensionality by simplifying the complex observation mapping space.

Static PCA: We project the features of the novice agent into a lower-dimensional space using the principal components (PCs) of the observation mapping space derived from the expert agent as illustrated in the Figure 1. The PCA layer as indicated by red box is a frozen layer. By reducing the feature space to these key components, we create a simplified representation that the novice agent can use to learn the task. The agent is, then, trained to take actions given the reduced observation mapping space, effectively focusing its learning on the most relevant aspects of the task as determined by the expert agent's experience. The reduced feature space helps limit the exploration of the novice agent to the most promising regions.

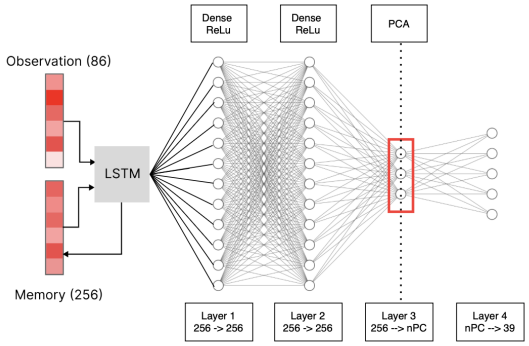


Figure 1: Static PCA architecture

Bottleneck: Instead of using a static PCA, we employ an additional fully connected layer to the policy neural network to dynamically learn the best feature mapping before action selection. The Figure 2 illustrates the architecture of the modified network. It involves adding a bottleneck layer, just before the output layer, which forces the network to compress the information into a lower-dimensional representation. It should allow the network to adaptively determine the most important features, potentially capturing more nuanced relationships than static PCA. This bottleneck architecture serves as a baseline for comparison. It allows us to evaluate the effectiveness of PCA-reduced observation space from the expert agent against a dynamically learned reduced observation mapping space.

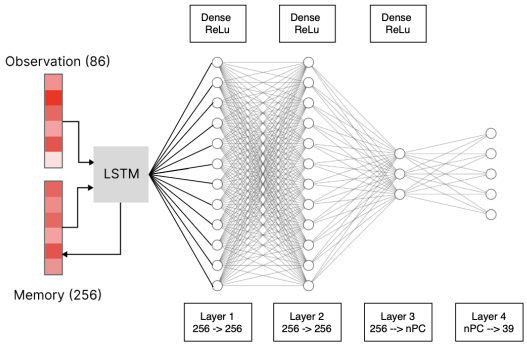


Figure 2: Bottleneck architecture

3.3.2 Action space dimensionality reduction

The second approach focuses on directly reducing the action space dimensionality. We aim to constrain the agent's exploration so that it only explores the most probable and relevant action space derived from the expert agent's experience. By reducing the dimensionality of the action space, we add constraints to guide the agent's exploration towards the best possible actions, captured by the PC of the expert.

Projection and Back-Projection: In this strategy, actions are projected into a reduced space formed by the PCs that were extracted from the expert. The obtained vectors are, then, projected back to the original high-dimensional action space. This process, illustrated in Figure 3, involves splitting the projection and back-projection phases to reduce small variations and noise. The idea is that by reconstructing from the projected space, the basic actions remain consistent, but the intensity of the activated muscles is smoothed, removing small variations. This smoothing effect helps the agent to capture the overall behavior more effectively.

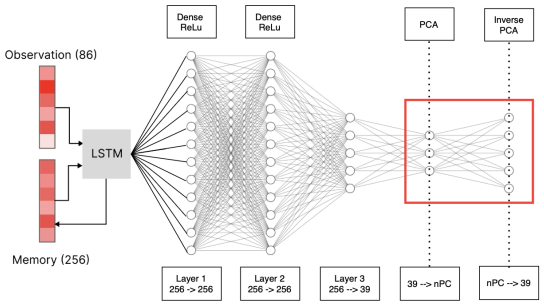


Figure 3: Projection and Back-Projection architecture

Reduced Latent Space Exploration: This approach enforces exploration constraints by projecting actions into a reduced latent space, conducting exploration within this space, and then projecting back to the original action space. The key difference from the previous method is the focus on latent space, which represents a more abstract and potentially more informative reduced space for exploration. By operating within this latent space, the agent can explore variations around the high-variance

components identified from the expert's policy. This method ensures that the agent's exploration is primarily within the most critical regions of the action space, promoting efficient learning and potentially faster convergence to an optimal policy. This approach is illustrated in Figure 4, where the exploration is conducted after PCA and then projected back to ensure the agent focuses on the most informative components.

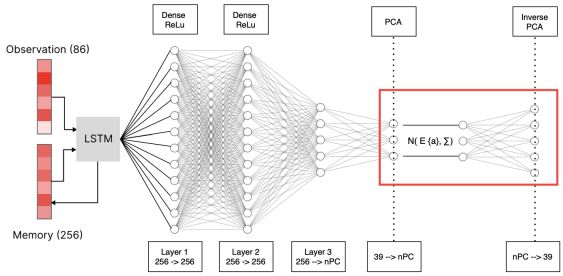


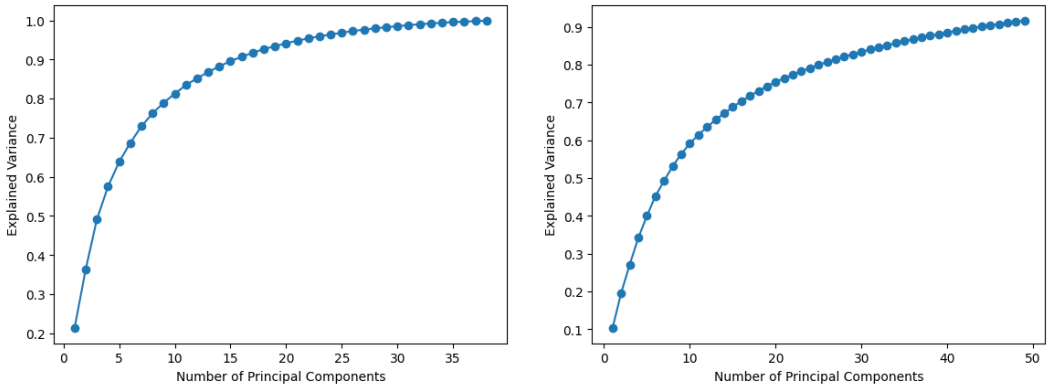
Figure 4: Latent Space Exploration architecture

4 Results Analysis

4.1 PCA Analysis

After analyzing the figures showcasing the cumulative explained variance by principal components in both the action space and feature space, we can conclude that a suitable number of principal components (PCs) for each domain can be determined by considering their cumulative contribution to the variance. In the action space, a suitable number of PCs to be extracted is 16, as their cumulated contribution to the variance is greater than 90%. For the feature space, a suitable number of PCs to be extracted is 40, accounting for a substantial 88% of the variance. We can note that achieving a cumulative variance of over 0.9 requires adding at least 10 components, covering only 0.02 of the variance. Keeping only 40 PCs seems, therefore, to be a good tradeoff.

By selecting these 16 PCs for the action space and 40 PCs for the feature space, we effectively capture the most significant features, enabling proper dimensionality reduction while preserving the essential characteristics of both spaces for our analysis.



(a) Action Space (b) Feature Space

Figure 5: Cumulative explained variance of the PCs

4.2 Experiments

All four models were evaluated under stochastic conditions. The mean reward indicates the average reward that a model achieved throughout training, and it varies within the range specified by the

standard deviation STD Reward. Similarly, the mean episode length, which measures the duration before failure (e.g., balls dropping) also fluctuates within the range defined by its standard deviation STD Episode Length.

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

4.2.1 Feature space dimensionality reduction

Experiment 1: It involved integrating a PCA layer into the original neural network architecture to reduce the feature space from 256 dimensions 40 which is the number of principal components (PCs), and then mapping these PCs to the final 39 features. This model achieved a mean reward of 485 with a standard deviation of 7.3 indicates that this model performed robustly under varied conditions, consistently earning high rewards. The relatively stable mean episode length of 121.4 with a standard deviation of 3.5 suggests that the model was effective in sustaining performance over time before failure occurred. The integration of a PCA layer demonstrated a substantial improvement in performance compared to the more complex models in subsequent experiments. This suggests that feature reduction at this level of dimensionality effectively balances complexity and performance, offering a sweet spot for this specific task.

Experiment 2: Similar to the first experiment, but replacing the static PCA layer with a fully connected layer that dynamically learns the best feature mapping for action selection. The mean reward decreased to 321 with a standard deviation of 7.8, and the mean episode length also reduced to 111.3 with a standard deviation of 4.2. These changes suggest that, while the network was adapting, it might have faced challenges in stabilizing its performance, possibly due to the increased complexity or overfitting to the training, as the network adjusts the feature representation based on feedback from the environment, potentially capturing more complex patterns beneficial for the task.

4.2.2 Action space dimensionality reduction

Experiment 3: This retained the original neural network structure but added a PCA layer that projected the 39 output features into a smaller feature space of only 16 dimensions, previously defined by the expert, before projecting them back to the 39 output features. The results showed a significant drop in performance, with a mean reward of 41 and a standard deviation of 2. The episode length was also much shorter at 22.5 with a standard deviation of 1.5, indicating a quicker failure rate and less robustness in maintaining performance.The drastic reduction in feature space seems to have been too severe, potentially omitting necessary information for making effective decisions. This suggests that there is a critical threshold below which further reduction in dimensionality adversely affects the model's capability to function effectively.

Experiment 4: This setup is built on the third experiment by adding exploration of the reduced feature space before mapping back to the original action space. The slight decrease in mean reward to 39 and an episode length of 21.5, both with small standard deviations of 2.1 and 0.9, indicates a continued struggle in leveraging the reduced feature space effectively, even with the added exploration phase. The results suggest that simply spending more time within this space is not sufficient to compensate for the loss of critical information due to excessive compression, although the exploration of the reduced feature space was a logical step.

4.2.3 Evaluation:

The first two experiments demonstrated significantly higher rewards and episode lengths compared to the last two. The higher performance suggests that, in these initial experiments, the models were at least able to maintain control over the task (e.g., holding the balls) for longer periods, which is the initial step of the curriculum learning SDS according to the project paper, although they struggled with more complex manipulations like correctly rotating the balls, as indicated in

the skeletal simulations. The significantly lower rewards and episode durations in the latter two experiments suggest difficulties in basic task retention, such as holding the balls, which was visually confirmed through the simulations.

5 Discussion

5.1 Evaluating Task-Specific Performance Through PCA

The series of experiments underscores the critical role of selecting an appropriate number of principal components (PCs) in feature reduction strategies. This choice must align with the complexity of the tasks to ensure effective performance. In these experiments, the reconstruction from the reduced feature space maintains consistent basic actions but smooths out the intensity of muscle activation, eliminating minor variations. This smoothing occurs because the agent struggles to fully execute all its tasks within the constricted feature space

For instance, while the agent learns to hold the balls, a task that does not require fine variations, it fails to effectively rotate the balls, which demands maximal muscle extension. This limitation in the model's learning process induced by the smoothing prevents it from extending the muscles sufficiently to perform complex tasks like ball rotation. Consequently, while this effect allows the agent to capture overall behavior more efficiently, it overlooks crucial, nuanced movements essential for complete task execution.

Given that Experiment 1 demonstrated a 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 such as ball rotation and then boarding. Since training to hold the balls was accomplished in just 4 hours, it is projected that training the model to perform all three tasks, holding, rotating, and boarding, could be completed in less than 24 hours. This is a substantial reduction from the 400 hours initially cited in the literature, indicating a significant efficiency improvement in the training process.

5.2 Optimizing Dimensionality Reduction Strategies for Enhanced Model Training

The strategic selection of 16 principal components (PCs) for the action space and 40 PCs for the feature space of the last hidden layer was intended to maintain explained variances above 0.9 and 0.88 respectively. However, the model's persistent struggles, particularly with tasks involving ball rotation, suggest that the lower variations excluded from the model (those accounting for the remaining 0.1 and 0.12 of variance) are indeed critical for complete training success. Since the explained variance is highly task-specific and our model manages intricate details like training 39 distinct muscles, it appears necessary to include more variance by increasing the explained variance percentage for both action and feature spaces.

In light of this, discussions with the supervisor have showed that it was recommended to adjust the number of components for the action space to 24 instead of 16. This adjustment aims to preserve more of the essential variations needed for comprehensive model training.

Moreover, the use of Principal Component Analysis (PCA) itself may need reconsideration. While PCA is effective for reducing dimensionality by capturing maximum variance within fewer dimensions, it might not be the most suitable method for tasks that involve complex, dependent actions like the ones our model is trained on. Alternative dimensionality reduction techniques might be more appropriate like Independent Component Analysis (ICA) which, unlike PCA, that prioritizes directions that maximize variance, focuses on finding components that are statistically independent from each other. This could be particularly advantageous in tasks where independent features contribute uniquely to performance outcomes.

Additionally, experimenting with the positioning of the PCA layer within the neural network architecture might yield improvements. Shifting the PCA layer to different points between other layers could potentially optimize the variance explained, ensuring that essential information is not lost during dimension reduction. This approach may validate whether an explained variance of 0.9, for example, is sufficient to maintain the necessary components for the model to learn effectively without significant loss of crucial details.

6 Conclusion

The experiments conducted in this study reveal significant insights into the efficiency of dimensionality reduction and policy distillation strategies for training RL agents in motor control tasks. While reducing the observation and action spaces aids in simplifying the learning environment, it is crucial to balance the extent of reduction to avoid losing critical information necessary for task execution. The results indicate that a careful selection of the number of principal components and strategically placing the projection layer within the neural network architecture are keys to maintaining an effective learning process. Moreover, transitioning from novice to expert performance can be optimized by adjusting the dimensional reduction techniques to enhance learning outcomes. Future work should explore alternative dimensionality reduction techniques and different configurations of the learning architecture to further improve the efficiency and effectiveness of training RL agents in complex motor tasks.

References

[1] Nisheet Patel Abigaïl Ingster Alexandre Pouget Alexander Mathis Alberto Silvio Chiappa, Pablo Tano. Acquiring musculoskeletal skills with curriculum-based reinforcement learning. 2024.

[2] Frank Zimmer Mike Preuss Marco Pleines, Matthias Pallasch. Generalization, mayhems and limits in recurrent proximal policy optimization. 2022.

[3] Zhang Z Atzori M Müller H Xie Z Scano A Zhao K, Wen H. Evaluation of methods for the extraction of spatial muscle synergies reinforcement learning. 2022.