Using Proximal Policy Optimization with Adversarial Motion Priors to Aid Rehabilitation

Jamal Akhras

Master of Science in Computer Science and Artificial Intelligence
The University of Bath
2023-2024

Using Proximal Policy Optimization with Adversarial Motion Priors to Aid Rehabilitation

Submitted by: Jamal Akhras

Copyright

Declaration

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of Bachelor of Science in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Abstract

This paper explores the application of Proximal Policy Optimization(PPO) combined with Adversarial Motion Priors (AMP) to enhance rehabilitation processes through the training of an exoskeleton. Rehabilitation often requires precise and adaptive assistance to recover motor functions, particularly for individuals with mobility impairments. The integration of PPO, a reinforcement learning technique known for its stability and efficiency, with AMP, which ensures the generation of realistic and effective movement patterns, aims to create a robust framework for exoskeleton control. This study details the methodology of incorporating adversarially trained motion priors to refine the policy learning process, ensuring the exoskeleton mimics natural human movements. The methodology to incorporate these priors into the training process is outlined, with a discussion on potential improvements in the adaptability and efficacy of exoskeleton-assisted rehabilitation. Although experimental validation remains a future endeavor, this exploration lays the groundwork for developing more natural and effective rehabilitation aids, potentially accelerating recovery, and improving patient outcomes and overall quality of life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Innovations in rehabilitation technology are rapidly transforming the field of physical therapy, with reinforcement learning emerging as a promising approach to enhance the control and effectiveness of exoskeletons. By leveraging principles similar to human learning, this dissertation delves into the application of combining traditional reinforcement learning algorithms and imitation learning to optimize the control of mechanisms of exoskeletons, thereby revolutionizing rehabilitation practices. Wearable robots, such as lower-limb exoskeletons, hold significant promise for rehabilitation and enhancing human capabilities Pons (2010). Scientific and technological work on wearable exoskeletons began in the 1960's, however, only recently have they been applied to gait assistance rehabilitation and to assist those suffering from motor disorders. Exoskeletons that only offer partial assistance usually are lighter and target people with less severe handicaps. Additionally, they can be used to assist healthy individuals for performance augmentation purposesBaud et al. (2021). Lower limb rehabilitation exoskeletons (LLREs) are used more often nowadays and have shown significant benefits in the improvement of mobility in individuals with a variety of neuromuscular disorders, including but not limited to, muscle weakness and paralysis Huo et al. (2016); Banala et al. (2009). The development of LLREs to aid those with neruomuscular disorders is crucial in the field of rehabilitation exoskeletons Deng et al. (2019); Moreno, Figueiredo and Pons (2018).

This dissertation aims to tackle the issue of improving rehabilitation for those with mobility impairments through the use of reinforcement and imitation learning using advanced exoskeletons. The primary aim is to enhance the adaptability and efficacy of exoskeleton-related rehabilitation by combining Proximal Policy Optimization(PPO) with Adversarial Motion Priors(AMP). PPO is utilized to dynamically adjuct the exoskeleton's resistance profile based on the patient's force output, ensuring precise and tailored assistance. Simultaneously, AMP ensures that the exoskeleton will adhere to the human limits by using a separate network that gives the model a "style" reward informed by a provided dataset of conventional movement techniques. The currently available exoskeletons include ReWalk (ReWalk Robotics), Ekso (Ekso bionics), Indego (Parker Hannifin), TWIICE Vouga et al. (2017), VariLeg Schrade et al. (2018) and LFMAS Huang et al. (2018). The ReWalk machine uses the tilt angle of the upper body to initiate walking, while Ekso uses accelerometers on crutches and pressure sensors placed on the patients shoes to identify the walking intention of the patient. Although convenient, holding the crutches with the hands limits the patients ability to interact with the environment Baud et al. (2019),hinders the patients ability to respond quickly in the case of an emergency and adds unnecessary strain to the patients upper body. There are a limited number of LLREs

that are able to assist human walking without the need of crutches or helpers like Rex (Rex Bionics) and Atalante (Wandercraft). Although these LLREs do not require the use of the patients hands, they are limited by their slow walking speed, heavy weights, 38kg and 60kg respectively, and extremely high costs Vouga et al. (2017). The creation of a more realistic and generalizable exoskeleton control system intends to speed up the rehabilitation process and improve patient outcomes.

1.1 Contributions

This dissertation makes several contributions to the field of exoskeleton-assisted rehabilitation, particularly through the integration of reinforcement learning techniques. The key contributions are:

- **A Novel Framework for Dynamic Exoskeleton Control**: A novel theoretical framework that integrates proximal policy optimization with adversarial motion priors that dynamically adjusts the resistance profile of the exoskeleton based on the patient's force output ensuring movements remain within natural human patterns is proposed.
- **The Incorporation of Patient Effort into the Reward Function**: The approach includes patient effort into the reward function, encouraging active participation from the patient during the rehabilitation process.
- **Comparison of Current Methods vs. the Proposed Method**: The dissertation highlights the limitations of the current methods used and the potential improvement utilizing the proposed methods.
- **Future Research Directions**: The dissertation identifies the key areas for future research, including the exploration of other reinforcement learning algorithms and motion prior techniques, and the importance of collecting comprehensive datasets of natural human motions.
- **Advancing the Field of Rehabilitation Technology**: By p[providing a theoretically sound and adaptable framework, this study contributes to the development of more responsive, personalized, and generalizable rehabilitation aids, with the potential to significantly improve the quality of patient care]

1.2 Dissertation Structure

This section provides an outline of the chapters covered in this dissertation.

- **Literature and Technology Review**: This chapter will provide background into the field of rehabilitation technologies and the use of reinforcement learning in this field. The algorithms chosen, namely proximal policy optimization and adversarial motion priors, and how they can interact with each other will be clarified. This chapter will also identify the current challenges and limitations in the use of PPO with AMP for rehabilitation.
- **Methodology and Implementation**: This chapter presents a comprehensive approach to integrating PPO with adversarial motion priors to enhance rehabilitation outcomes. The data that will be used as the priors will be outlined and described. Within the reinforcement learning framework, the environment, agent, actions, and rewards are

defined to model the rehabilitation task, alongside the details of the implementations proximal policy optimization and the integration of adversarial motion priors.

- **Results and Analysis**: This chapter will present and evaluate the performance of a traditional reinforcement learning algorithm against a model that incorporates adversarial motion priors and analyze the implications of the results for rehabilitation outcomes.
- **Discussion and Future Works**: Lastly an interpretation of the results is given, analyzing the successes limitations and scalability of the project. In addition, the chapter proposes future research directions, such as leveraging emerging technologies to further enhance rehabilitation outcomes and other potential uses of this technology.

Chapter 2

Literature and Technology Review

This chapter is split into two sections, the first of which are the preliminaries which introduce the key theory and notation used in the dissertation. The second is the related works section, which discusses the existing literature and technology relating to the dissertation.

2.1 Preliminaries

Markov Decision Processes(MDP) Puterman (1994) are an intuitive and fundamental formalism for decision-theoretic planning(DTP) Boutilier, Dean and Hanks (2011) and other learning problems in stochastic domains. In this case the environment is modelled as a set of states and actions can be performed to control the state. The goal is to control the system in a way that maximises some performance criteria. MDPs have since become the standard formalism for learning sequential decision making. MDPs consist of states, actions, transitions between states, and a defined reward function.

- **States:** States are a unique characterization of everything that is important of the problem that is modeled. They are usually denoted as $s \in S$, where a state s is a state in the set of all states S.
- **Actions:** Actions are what the agent uses to interact with and control the state space and is commonly denoted as $A(s)$ where A is an action applied to state s.
- **Transitions:** The application of action $a \in A$ in a state $s \in S$ results in the transition from s to a new state s' that also belongs to the state space S. A system is called *Markovian* if the result of an action does not depend on the previous action taken, the idea is that that current state s contains enough information for the agent to make an optimal decision.
- **The Reward Function:** The reward function specifies the reward for being in a state or performing an action in that state. This is an important part of MDPs that implicitly specifies the goal of learning. Rewards are often not limited to positive integers which can be understood as sub-goals for learning, essentially telling the agent that the last action or state resulted in the opposite of what is intended to be done.

Combining all these elements defines a Markov decision process. The policy is a part of the agent which aims to control the environment and maximize the reward function. A deterministic policy is a function that outputs for each state an action, however a stochastic policy is used,

which maps the actions along a probability distribution with the most probable action being the most likely to yield a desirable outcome (Wiering and Otterlo (2012)).

Reinforcement learning (RL) is a subset of machine learning that allows an AI system (referred to as an AI agent) to learn how to complete a task through trial and error using feedback from the actions it takes in an environment Sutton and Barto (2020). Initially reinforcement learning was limited to discrete state and action spaces but have since been adapted to handle more complicated environments through the introduction of deep reinforcement learning. The first major successful implementation of this method was deep Q-networks(DQN) Mnih et al. (2015), a deep learning implementation of Watkins and Dayan (1992) Q-network algorithm. DQN was used to train an agent to complete tasks given high dimensional state spaces and rewards. Another successful implementation came when AlphaGO, a hybrid of deep reinforcement learning, supervised learning, and heuristic search approach, was able to defeat the Go world champion (Silver et al. (2016)). Since then several methods of deep learning have been combined with reinforcement learning, leading to advancements in many fields, including robotics Kober, Bagnell and Peters (2013) and autonomous drivingKiran et al. (2022). Despite the successes of deep Q-networks, they are limited to discrete action spaces.

Schulman et al. (2017) later introduced proximal policy optimization algorithms (PPO), which are policy-based methods that are capable of handling both discrete and continuous action spaces. proximal policy optimization consists of two neural networks that are simultaneously trained to optimize a stochastic policy and approximate its value function. The stochastic policy to a Gaussian distribution representing different actions, this is what allows the model to handle continuous action spaces. PPO, an approximation of the earlier trust-region policy optimization algorithm Schulman et al. (2015), works by clipping the updated policies, limiting the amount it can change after each iteration resulting in more stable training. There are other algorithms that adopt a two network structure, like deep deterministic policy gradient(DDPG) Lillicrap et al. (2015), the key difference is that DDPG uses a deterministic policy that maps states directly to actions, this works best in an environment where there is a best action given a state at the cost of instability during training. Robotic control tasks usually have continuous state and action spaces, with the states made up of robot joint positions and velocities and the actions controlling the change of the joint angles and speeds. While possible to discretisize the actions to a limited set, it was found that continuous control offers far more control and often are far superior Doya (2000). While all the mentioned algorithms were capable of learning robot control policies in the field of robot locomotion Rudin et al. (2022) the project will leverage proximal policy optimization as in a comparison over a set of robotic tasks it was found to be the best Fan et al. (2018).

Although proximal policy optimization was able to successfully train a policy capable of robot locomotion it required complex reward functions that encourage physically plausible behaviors, crafting these reward functions required a tedious labour intensive process that did not generalize well. Escontrela et al. (2022) proposed substituting these complex reward functions with "style rewards" learned from a dataset of motion capture demonstrations. The learned style reward can then be used alongside an arbitrary task to train policies to perform tasks following more natural strategies, this method is called Adversarial Motion Priors(AMP). This method combined the flexibility and scalability of adversarial imitation learning Abbeel and Ng (2004) with auxiliary task objectives enabling simulated agents to perform high-level tasks while following behaviours from large unstructured motion datasets Escontrela et al. (2022). This is done by training a discriminator to tell the difference between the actions

produced by the policy and the behaviours shown in the demonstration data.

2.2 Related Work

The fast advancement of robotics and technology in recent years has furthered the development of many robotic applications, particularly exoskeletons. These exoskeletons have been developed to increase the efficiency of rehabilitation therapy by providing more intensive patient training, better quantitative feedback and improved functional outcomes for patients Che (n.d.). A drawback of the majority of existing rehabilitation exoskeletons is that they need the operator to use crutches for support, causing unnecessary stress on the upper body, a helper to help the user avoid falling down, or restrict the users ability to react in the case of an emergency. The assistance provided by these machines are often done using pre-planned movements of gait and lack the ability to perform diverse human motions. Some of these exoskeletons include the TWIICE Vouga et al. (2017) and VariLeg Schrade et al. (2018). These devices limit the patients ability to interact with the environment Baud et al. (2019), an example of this is the unlikelihood of the patient performing actions like squatting using a cane or crutches. The exoskeltons that can be operated without occupying the users hand come at the cost of slow walking speeds, increased weight, and are very expensive Vouga et al. (2017). Research into robust controllers against large and uncertain forces is not commonly carried out as bipedal balance control without external forces has been shown to be a challenging task in itself. The existing balance controller designs for similar lower body rehabilitation exoskeletons primarily revolve around the trajectory tracking method, conventional control like Proportional-Integral-Derivative (PID) Xiong (2014), model-based predictive control Shi et al. (2019), fuzzy control AYAS and ALTAS (2017), impedance control Karunakaran et al. (2020), and momentum-based control for standing Bayon et al. (2020). These methods can be added to regular motions relatively easily, however, they lack robustness against large external forces.

PPO with AMP can be leveraged to train an exoskeleton to perform tasks in a human like fashion in simulation, mitigating the risks to equipment and the patients. In simulation the forces likely to influence the machine can be replicated and handled by the policy to minimize potential risks to the users. Reinforcement learning methods that utilize adversarial motion priors have been used before to make the movements of video game characters more realistic Peng et al. (2021), and have yielded impressive results like training models that are able to navigate complex settings with simple reward functions. To my knowledge, this has not been used in real life yet and has the potential to vastly improve the field of robot aided rehabilitation.

More research into the application of PPO with AMP can(something something everyday use something something improve quality of life something something benefit healthy people too by giving more strength and replicating the motions of the best athletes in given sports)

More research into the application of proximal policy optimization with adversarial motion priors holds significant promise for improving rehabilitation processes for those that need it and quality of life for the general population. The benefits of these technologies is not limited to those in rehabilitation; healthy individuals can also stand to benefit. Using PPO with AMP it is possible to develop training programs that increase strengths and replicate the motions of elite athletes in various sports, leading to better performance and reducing the risk of injury Fang et al. (2022).

Further research should be conducted to assess and maximise the potential of using PPO with AMP. Studies should be done to explore the scalability of this technology to ensure their ability to be used across different populations and conditions. Additionally, more research can be done to develop user-friendly and wearable devices that integrate PPO with AMP making it easier for everyone top to benefit from these advancements. Lastly, some collaboration between computer scientists and sports scientists will be crucial in refining these technologies and discovering new applications, ultimately paving the way for more innovative and effective solutions in rehabilitation and physical training.

Chapter 3

Methodology and Implementation

This section details the methodological framework and implementation strategies employed in this study, including the integration of Proximal Policy Optimization (PPO) with Adversarial Motion Priors (AMP), the design of the reward function, and the use of Isaac Gym's GPU-based physics engine for efficient simulation and training of the exoskeleton control policies.

3.1 Environment Design

In order to effectively optimize the control mechanisms of the exoskeleton for rehabilitation purposes, a Markov Decision Process (MDP) framework is adopted. MDPs provide a formal mathematical framework for modelling decision-making processes in dynamic environments, making them well suited for representing the complex complex interactions inherent in the rehabilitation task. Structuring the problem as an MDP effectively captures the sequential nature of rehabilitation actions, the uncertainty in patient responses, and the dynamic nature of the environment. In this section the key components of the MDP are formulated, including the states, actions, transitions, rewards, and the policy, and clarify how this framework guides the approach taken to optimizing exoskeleton control using proximal policy optimization with adversarial motion priors.

1. The states in this MDP represent the various configurations of the exoskeleton and the patient's body during the process. Each state contains information about the current position, orientation, and movement of the exoskeleton and the patient.
2. The actions correspond to the control inputs being applied to the exoskeleton, these actions include adjusting joint angles, applying forces or torques to the specific parts of the exoskeleton, or activating/deactivating certain assistive features.
3. Transitions between the states are governed by the dynamics of the rehabilitation environment. These dynamics capture the physical interactions between the exoskeleton, the patient, and the external environment. Transitions are affected by the actions taken and the current state of the system.
4. The rewards in the MDP serve as feedback signals indicating the effectiveness of the rehabilitation process. Positive rewards may be given for achieving rehabilitation goals, such as correct technique or successful completion of a task. Negative rewards may be given for incorrect movements or not following the the desired trajectories.

5. The policy in the MDP represents the strategy or set of rules that are following to select actions based on the current state. In the context of this project the policy is learned using PPO with adversarial motion priors, aiming to optimize rehabilitation outcomes by selecting outcomes that maximize the added rewards over time.

By adopting a Markov Decision Process framework a structured approach to tackling the complexities of optimizing exoskeleton control for rehabilitation is given. Systematically defining the states, actions, transitions, rewards, and polices, the groundwork for an effective decision making process is set.

3.2 Simulation Environments

Choosing the right environment for training is crucial as it ensures realism and accuracy, presents relevant challenges, facilitates efficient learning, enhances transferability to real-world scenarios, ensures safety, supports reproducibility, allows for comprehensive testing, and aligns with the specific objectives of the task.

3.2.1 PyBullet

The simulation for PPO implementation is built in the PyBullet Coumans and Bai (2016) physics engine and designed to replicate an authentic physical environment. PyBullet is efficient, allowing for rapid simulations and reduced training times, and includes visualization tools for real time debugging. The state space for the `pybulletHumanoid` model that is used includes a comprehensive set of variables that describe the current status of the humanoid robot. These elements provide a detailed representation of the humanoid's configuration, crucial for the reinforcement learning algorithm to make informed decisions. The components of the state space include the joint angles and velocities, the position and orientation of the humanoid in the environment, the linear and angular velocities of the humanoid, and binary values representing which parts of the humanoid are in contact with the ground. For the given implementation this space usually contains 44 values.

The `HumanoidPyBulletEnv-v0` model is a detailed simulation of a humanoid robot, designed for use with the PyBullet physics engine. It features a human-like structure with multiple degrees of freedom, including joints for shoulders, elbows, hips, knees, and ankles, as shown by figure 3.2. The model includes sensors for position, orientation, force, and torque, providing comprehensive feedback for control algorithms. PyBullet's accurate physics simulation enables realistic interactions with the environment, making the model ideal for reinforcement learning research.

Vectorized Environments

PyBullet works with stable-baselines' "*vectorizedenvironments*", a method for stacking several independent environments into a single environment allowing the agent to train on n environments per step significantly decreasing training times, in the case of this implementation $n = 5$ effectively meaning that the agent was learning from 5 separate experiences per step.

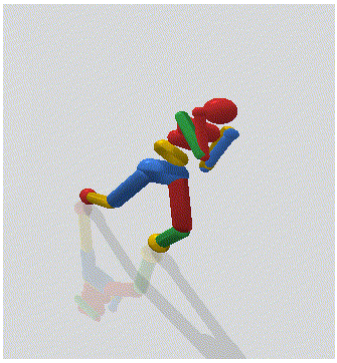

Figure 3.1: HumanoidPyBulletEnv-v0

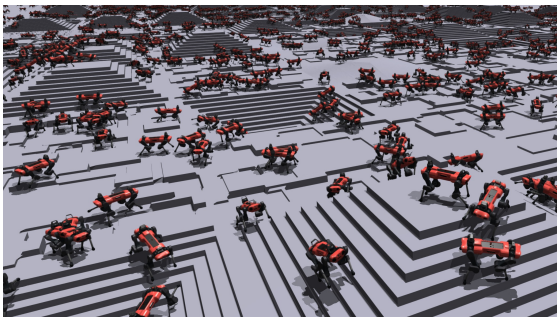

Figure 3.2: Isaacgym Parallelization
Makoviychuk et al. (2021)

3.2.2 Isaacgym

Isaac Gym is an end-to-end high performance robotics simulation platform. It leverages NVIDIAS PhysX physics engine to provide a GPU-accelerated simulation, enabling it to train at 2-3 orders of magnitude faster than CPU based environments in continuous control tasks. This allows it to gather experience data required for robotics RL at rates only achievable using a high degree of parallelism Makoviychuk et al. (2021). Implementing PPO with AMP in this environment will allow for more accurate and quicker training of the exoskeleton and train it to complete much more complex tasks, such as sports movements, in a reasonable time frame.

3.3 Experiment Design

A custom implementation of proximal policy optimization was used for several reasons, mainly for the customization. The custom implementation allows for the flexibility to customize every aspect of the algorithm, including the hyperparameters and the neural network architecture.

3.3.1 Network Structure

The input layer of the network takes in the observation space or states as input and is connected to the first hidden layer.

The first hidden layer of the network is a linear transformation (fully connected layer) with *inputDims* number of input neurons and 128 output neurons. 128 neurons were chosen based on experiments conducted by Stamford. It applies the transformation to the input observations

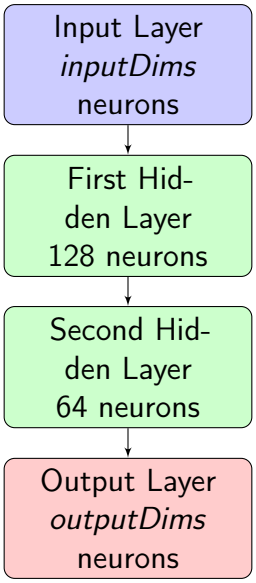

Figure 3.3: Neural Network Structure

and passes the result to the next layer after applying the ReLU activation function.

The second layer also consists of a linear transformation, taking the 128 output neurons from the first hidden layer as input and producing 64 output neurons. Similar to the first hidden layer, it applies the ReLU activation function to its output.

The output layer is the final layer of the network. It consists of a linear transformation with 64 input neurons and *outputDims* number of output neurons. The output dimensions depend on whether the network is used for the actor (distribution over actions) or the critic (value for state). No activation function is applied to the output layer, as it is typically used to produce raw output values.

3.3.2 Priors Dataset

The dataset that will be used by the adversarial motion prior will be generated using motion capture technology(MoCaP) and is integrated into the model using acclaim skeleton format(asf) and adaptive modulation and coding(amc) files. In the context of motion capture, asf files are used to define the structure of a skeletal model. These files specify the hierarchy, segment lengths, and joint constraints of the skeleton. ASF files describe each bone with attributes such as length, orientation, and degrees of freedom, ensuring realistic movements by defining parent-child relationships and joint constraints. Typically used alongside AMC files that contain the actual motion data to carry out the desired task.

In the context of this project, the prior dataset will comprise of a single asf file detailing the exoskeleton alongside multiple amc files of the exoskeleton carrying out tasks set by the needed rehab, which may include walking, running, or whatever is required for the rehabilitation process.

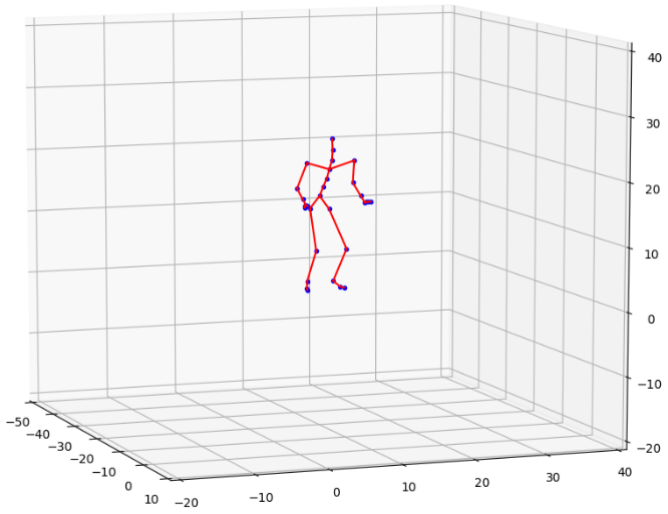

Figure 3.4: ASF visualization of humanoid

3.3.3 Proximal Policy Optimization with Adversarial Motion Priors

The reward function is designed to ensure that the simulated exoskeleton not only achieves specific therapeutic tasks but also mimics desired motion styles that are beneficial for the patient's recovery. The total reward r_t at time step t is a weighted sum of style-rewards r_{S_t} and task-rewards r_{G_t}:

$$r_t = \alpha r_{S_t} + \beta r_{G_t}$$

where:

- r_{S_t} represents the style-reward at time step t, encouraging the exoskeleton to produce movements that match the reference motions deemed beneficial for rehabilitation.
- r_{G_t} represents the task-reward at time step t, incentivizing the exoskeleton to achieve specific therapeutic goals.
- α and β are weight coefficients that balance the importance of style and task rewards.

Style-Reward r_{S_t}

The style-reward r_{S_t} ensures that the exoskeleton's movements align with the reference motions from the rehabilitation dataset. This may include:

- **Pose Matching**: Encourages the exoskeleton's pose to be similar to the reference pose, helping the patient achieve proper postures.

  $$r_{pose} = -\|\mathbf{p}_t - \mathbf{p}_{ref}\|^2$$

  where $\mathbf{p}_t$ is the exoskeleton's pose at time step t and $\mathbf{p}_{ref}$ is the reference pose.

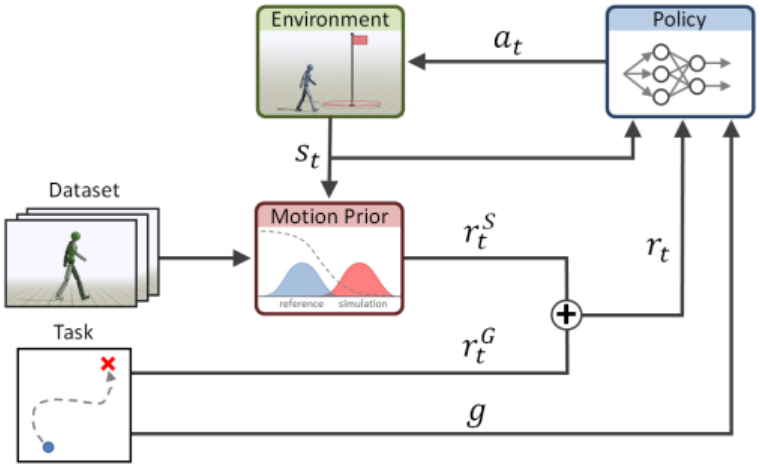

Figure 3.5: Overview of the system. Starting with a motion dataset that defines the desired motion style for a character, the system trains a motion prior which provides style-rewards r_{S_t} during policy training. These style-rewards are combined with task-rewards r_{G_t} to train a policy that enables the simulated character to achieve task-specific goals g while also replicating the motion styles from the reference dataset. The policy in this case is given by the actor critic model of proximal policy optimization

Peng et al. (2021)

- **Velocity Matching**: Ensures that the exoskeleton's movement speed matches the reference velocity, promoting smooth and controlled motions.

$$r_{velocity} = -\|\mathbf{v}_t - \mathbf{v}_{ref}\|^2$$

  where $\mathbf{v}_t$ is the exoskeleton's velocity and $\mathbf{v}_{ref}$ is the reference velocity.

Task-Reward r_{G_t}

The task-reward r_{G_t} focuses on achieving therapeutic goals specific to the patient's rehabilitation needs. This can include:

- **Goal Achievement**: Rewards the exoskeleton for helping the patient perform specific tasks, such as walking a certain distance or maintaining balance.

$$r_{goal} = f(\mathbf{g}, \mathbf{s}_t)$$

  where $\mathbf{g}$ represents the therapeutic goal and $\mathbf{s}_t$ is the current state of the exoskeleton.

- **Penalty for Unwanted Behavior**: Penalizes the exoskeleton for movements that could be detrimental to the patient's recovery, such as jerky or unstable motions.

$$r_{penalty} = -\sum_i \mathbf{c}_i$$

  where $\mathbf{c}_i$ are different penalty terms.

Combined Reward Function

The combined reward function that guides the training of the exoskeleton is thus:

$$r_t = \alpha \left(-\|\mathbf{p}_t - \mathbf{p}_{ref}\|^2 - \|\mathbf{v}_t - \mathbf{v}_{ref}\|^2 \right) + \beta \left(f(\mathbf{g}, \mathbf{s}_t) - \sum_i \mathbf{c}_i \right)$$

Here, α and β are hyperparameters that balance the contributions of style and task rewards, ensuring that the exoskeleton performs therapeutic tasks effectively while promoting beneficial movement styles.

3.4 Training Details

Algorithm 1: PPO-Clip

Input: initial policy parameters θ_0, initial value function parameters ϕ_0

for $k = 0, 1, 2, \ldots$ **do**

Collect set of trajectories $\mathcal{D}_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment;

Compute rewards-to-go $\hat{R}_t$;

Compute advantage estimates $\hat{A}_t$ based on the current value function V_{ϕ_k};

Update the policy by maximizing the PPO-Clip objective:

$$\theta_{k+1} = \arg\max_\theta \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min \left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} \hat{A}_t^{\pi_k}(s_t, a_t), g(\epsilon, \hat{A}_t^{\pi_k}(s_t, a_t)) \right)$$

typically via stochastic gradient ascent with Adam;

Fit value function by regression on mean-squared error:

$$\phi_{k+1} = \arg\min_\phi \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t \right)^2$$

typically via some gradient descent algorithm;

The training loop was used to train a humanoid to walk in the PyBullet environment mentioned later in the dissertation to demonstrate the shortcomings of traditional reinforcement learning methods when it comes to accurately replicating human locomotion.

The PPO-Clip algorithm's training loop begins with initializing the policy parameters θ_0 and value function parameters ϕ_0. For each iteration k, the current policy $\pi_k = \pi(\theta_k)$ is used to collect a set of trajectories $\mathcal{D}_k = \{\tau_i\}$ by interacting with the environment, which provides the necessary experience data consisting of states, actions, and rewards. Next, rewards-to-go $\hat{R}_t$ are computed for each time step t within the trajectories, representing the cumulative future rewards. Advantage estimates $\hat{A}_t$ are then calculated based on the current value function V_{ϕ_k}, indicating how much better or worse each action is compared to the expected value of the state. The policy is updated by maximizing the PPO-Clip objective:

$$\theta_{k+1} = \arg\max_\theta \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min \left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} \hat{A}_t^{\pi_k}(s_t, a_t), g(\epsilon, \hat{A}_t^{\pi_k}(s_t, a_t)) \right),$$

where $g(\epsilon, \hat{A}_t^{\pi_k}(s_t, a_t))$ is a clipping function to ensure stable updates. This update is typically performed via stochastic gradient ascent using the Adam optimizer. Concurrently, the value function is updated by minimizing the mean-squared error between the predicted values $V_\phi(s_t)$ and the rewards-to-go $\hat{R}_t$:

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t \right)^2,$$

typically using gradient descent. This iterative process continues, refining the policy and value function parameters until the policy converges or the desired performance is achieved, resulting in an effective and stable policy for the given task.

Using reinforcement learning with adversarial motion priors was initially done for stylized control of video game characters and was trained in the Isaac Gym environment. The training loop of their implementation followed the following psuedocodeHuang et al. (2018).

Algorithm 2: Training Vanilla Gradient Descent with AMP

Input: $\mathcal{M}$: dataset of reference motions

$D \leftarrow$ initialize discriminator;
$\pi \leftarrow$ initialize policy;
$V \leftarrow$ initialize value function;
$\mathcal{B} \leftarrow \emptyset$ initialize replay buffer;
while *not done* **do**
  for *trajectory* $i = 1, \ldots, m$ **do**
    $\tau^i \leftarrow \{(s_t, a_t, r_t^G, r_t^S, g)_t^{T-1}\}$ collect trajectory with π;
    for *time step* $t = 0, \ldots, T-1$ **do**
      $d_t \leftarrow D(\phi(s_t), \phi(s_{t+1}))$;
      $r_t^S \leftarrow$ calculate style reward d_t;
      $r_t \leftarrow w_G r_t^G + w_S r_t^S$;
      record r_t in τ^i;
    end
    store τ^i in $\mathcal{B}$;
  end
  for *update step* $l = 1, \ldots, n$ **do**
    $b^M \leftarrow$ sample batch of K transitions $\{(s_j, s_j')\}_{j=1}^K$ from $\mathcal{M}$;
    $b^\tau \leftarrow$ sample batch of K transitions $\{(s_j, s_j')\}_{j=1}^K$ from $\mathcal{B}$;
    update D using b^M and b^τ;
  end
  update V and π using data from trajectories $\{\tau^i\}_{i=1}^m$;
end

Adjusting the loop to utilize proximal policy optimizations stochastic gradient descent rather than vanilla gradient descent would involve modifying the loss function to include a clipping objective, value function loss, and entropy bonus, performing multiple epochs of updates on mini-batches. These changes stabilize the training and improve performance, addressing of the issues commonly faced with the high variance and instability of vanilla policy gradient methods. Considering these changes the pseudo-code would be better reflected by the following:

Algorithm 3: Training PPO with AMP

Input: $\mathcal{M}$: dataset of reference motions

$D \leftarrow$ initialize discriminator;
$\pi \leftarrow$ initialize policy;
$V \leftarrow$ initialize value function;
$\mathcal{B} \leftarrow \emptyset$ initialize replay buffer;
while *not done* **do**
  for *trajectory* $i = 1, \dots, m$ **do**
    $\tau^i \leftarrow \{(s_t, a_t, r_t^G, r_t^S, g)_t^{T-1}\}$ collect trajectory with π;
    for *time step* $t = 0, \dots, T-1$ **do**
      $d_t \leftarrow D(\phi(s_t), \phi(s_{t+1}))$;
      $r_t^S \leftarrow$ calculate style reward d_t;
      $r_t \leftarrow w_G r_t^G + w_S r_t^S$;
      record r_t in τ^i;
    end
    store τ^i in $\mathcal{B}$;
  end
  for *update step* $l = 1, \dots, n$ **do**
    $b^M \leftarrow$ sample batch of K transitions $\{(s_j, s'_j)\}_{j=1}^K$ from $\mathcal{M}$;
    $b^\tau \leftarrow$ sample batch of K transitions $\{(s_j, s'_j)\}_{j=1}^K$ from $\mathcal{B}$;
    update D using b^M and b^τ;
  end
  for *epoch* $e = 1, \dots,$ *num_epochs* **do**
    for *mini-batch* $b = 1, \dots,$ *num_mini_batches* **do**
      $b^\pi \leftarrow$ sample mini-batch of transitions from $\mathcal{B}$;
      compute advantages $\hat{A}_t$ for b^π;
      old$_p$*robs*$\leftarrow$ get action probabilities from π_{old};
      new$_p$*robs*$\leftarrow$ get action probabilities from π using b^π;
      $r_t(\theta) \leftarrow \frac{\text{new\_probs}}{\text{old\_probs}}$;
      clip_advantages $\leftarrow \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t$;
      surrogate_loss $\leftarrow -\text{mean}(\min(r_t(\theta)\hat{A}_t, \text{clip\_advantages}))$;
      value_loss $\leftarrow \text{mean}((V(s_t) - R_t)^2)$;
      entropy_bonus $\leftarrow \text{mean}(\mathcal{H}(\pi(\cdot|s_t)))$;
      total_loss $\leftarrow$ surrogate_loss $+ c_1 \cdot$ value_loss $- c_2 \cdot$ entropy_bonus;
      perform gradient descent step on total_loss;
    end
  end
  update V using data from trajectories $\{\tau^i\}_{i=1}^m$;
end

3.5 Hyperparameter Tuning

Hyperparameter tuning in reinforcement learning is essential for optimizing performance, ensuring convergence stability, enhancing generalization, improving training efficiency, and achieving a balance between exploration and exploitation. Properly tuned hyperparameters allow RL agents to adapt to different environments, cater to algorithm-specific requirements, and avoid issues like divergence or sub-optimal policies. This process is crucial for developing effective and efficient RL agents that can learn optimally and perform reliably across various tasks and environments Zhang et al. (2021).

The hyperparameters chosen for the model are shown in the table in *appednixB* The hyperparameters were tuned using manual hyperparameter tuning, which is the process of manually selecting and adjusting the hyperparameters to improve performance. The initial hyperparameters used were adopted from the official RL-Zoo GitHub repository. Although the method is time consuming it allowed for the finer control of the training process and due to the limited computational resources proved to be a strong choice.

Chapter 4

Results and Analysis

This section will discuss and analyze the results obtained by the PPO implementation in relation to the research question and hypothesis. A comparison will be made of those results against those of PPO with AMP and will consider the potential implications it has for the field of rehabilitation.

4.0.1 PPO Progressive Rewards and Actor Loss

The PPO model was trained over 48 hours, completing 17000 time steps of learning. Figure 4.1 shows the progressive rewards obtained during the training of the PPO model. The increasing trend in the rewards indicates that the model is learning to optimize the given reward function over time. This positive trend demonstrates that the PPO implementation is effectively improving its policy to achieve higher rewards.

During the training period, it was observed that the actor loss did not change significantly. This does not mean that the agent is not learning, since the ultimate goal is to maximize the return, which it does, this indicates that the agent is learning successfully. Some potential reasons for the loss not changing are:

- **Learning Rate Issues:** A learning rate that is too low may result in very small updates to the policy parameters, causing minimal changes in the actor loss. Conversely, a very high learning rate can cause oscillations or instability, potentially leading to a plateau in the actor loss.
- **Plateau in Learning:** The training process might be stuck in a local minimum or saddle point, where the gradient is close to zero, leading to a stagnant actor loss. Additionally, if the neural network's activation's are saturated, the gradients can become very small, slowing down learning.
- **Sub-optimal Reward Function:** If the reward function does not provide a strong or clear enough signal, the policy may not improve significantly, resulting in little change in the actor loss. Misaligned rewards can also cause the agent to learn sub-optimal behaviors.
- **Exploration-Exploitation Trade-off:** Insufficient exploration can prevent the agent from discovering better policies, leading to minimal changes in the actor loss. If the initial policy performance is already high, there may be limited room for improvement.

- **Algorithmic and Implementation Issues:** In PPO, clipping can restrict the magnitude of policy updates. If the clipping threshold is too tight, it may result in conservative updates and a stable actor loss. Implementation errors in gradient calculation can also lead to unexpected behavior.
- **Training Stability:** A stable actor loss might indicate that the learning dynamics have stabilized, with the policy consistently improving or maintaining performance without drastic changes in the loss.

Understanding these factors is crucial for diagnosing and addressing issues in the training process, ensuring that the reinforcement learning model can effectively learn and improve its policy.

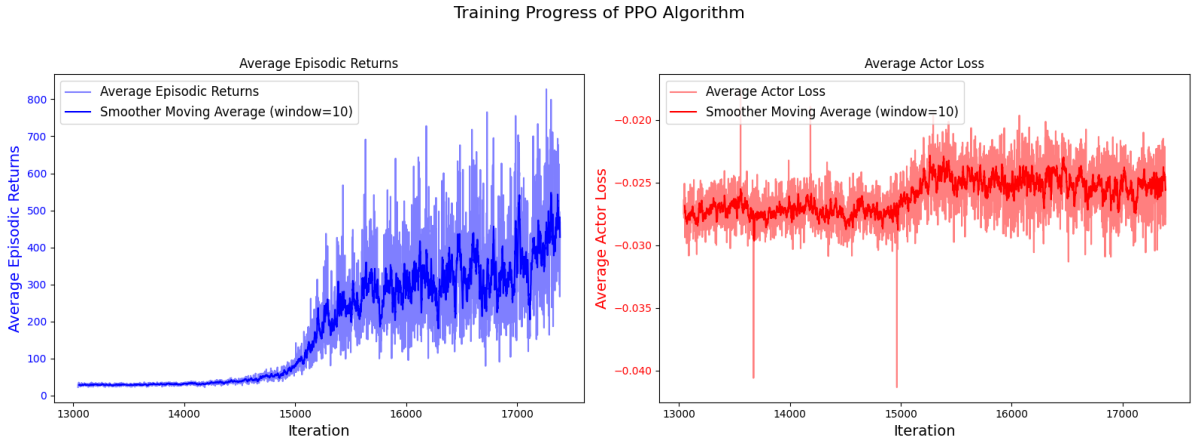

Figure 4.1: Progressive rewards and loss obtained during PPO training, indicating the model's learning progression.

4.0.2 PPO Walking Realism

Although the increasing returns indicate successful learning, the quality of the resulting walking behavior is not sufficient for practical rehabilitation purposes. Upon closer analysis, it was observed that the walking motion generated by the PPO policy lacks several critical aspects of human gait, such as smooth transitions between steps, natural joint movements, and stability.

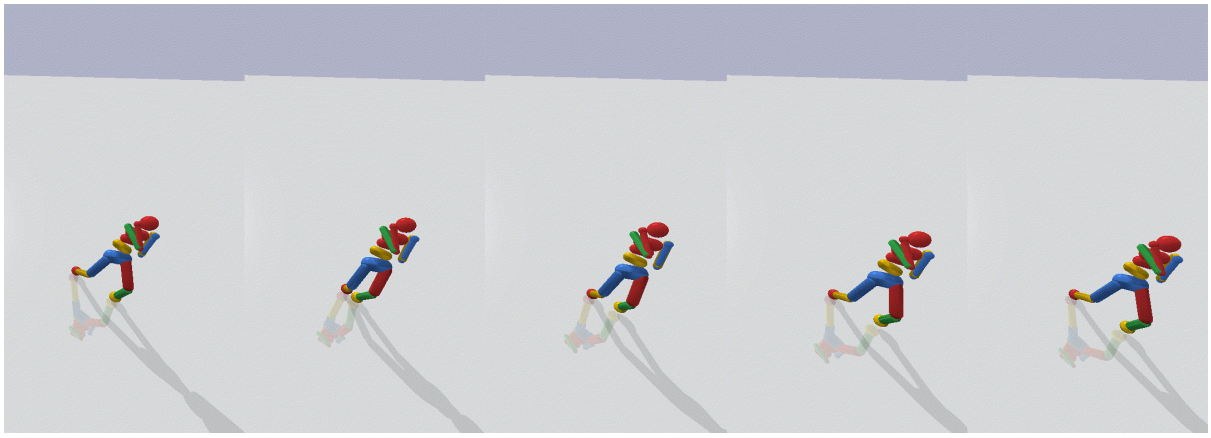

Figure 4.2: Visualization of the walking motion generated by the PPO policy.

Figure 4.2 illustrates the walking motion generated by the trained policy. While the model is able to produce a walking behavior, it does not closely resemble natural human gait. Given that the overall goal is to accurately replicate human movements through an exoskeleton, having an agent that exploits unnatural techniques to maximize the reward function is not desirable. This walking behaviour fails to accurately replicate the human movements thereby limiting the effectiveness of the exoskeleton for rehabilitation purposes and potentially causing discomfort or injury to the user. The discrepancy can be attributed to several factors:

- **Simplified Reward Function:** The reward function used in training primarily focuses on achieving forward locomotion and maintaining balance. However, it does not adequately capture the nuances of human walking dynamics, such as the synchronization of arm swings with leg movements and the maintenance of a consistent walking pace.
- **Limited Physical Realism:** The simulation environment may lack certain physical realism elements, such as accurate muscle forces, joint torques, and ground reaction forces, which are critical for replicating natural walking. The model also does not have enough joints in the feet to replicate a realistic heel toe walking action.
- **Absence of Biomechanical Constraints:** The PPO model may not fully incorporate biomechanical constraints that ensure safe and ergonomic movements, leading to unnatural or jerky motions.
- **Learned Walking Motion** The method the model adopted to walk was moving the lead leg forward and dragging the trail leg along, which is demonstrated in figure 4.1. the model also learned to maintain balance by tucking in the arms and not swinging them in a realistic fashion. This method also could not learn to walk using the traditional heel-toe technique that is considered to be normal due to the fact that the model uses a ball instead of feet, potential improvements to this would be making the ball more like a foot by adding two joints enabling the model to use dorsi and plantar flexion (the movement of pushing toes towards the ground and pulling them to the sky) to better mimic the heel toe walking action.

Given these limitations, the current PPO implementation is not suitable for use in rehabilitation without further enhancements. For the model to be practical for rehabilitation purposes, it needs to produce walking behaviors that are not only functional but also biomechanically accurate and comfortable for patients.

In summary, while the PPO implementation demonstrates promising learning capabilities as evidenced by the progressive reward graphs, the resulting walking motion lacks the realism required for effective rehabilitation. Future work should focus on refining the reward function, enhancing the physical realism of the simulation environment, and incorporating biomechanical constraints to achieve more natural walking behaviors.

4.0.3 AMP Model with Realistic Walking

In contrast, the AMP model demonstrated significantly more realistic walking behavior, closely mimicking human gait. The AMP model was trained in only 6 minutes, leveraging the Isaac Gym environment, which utilizes a GPU-based physics engine for efficient simulation.

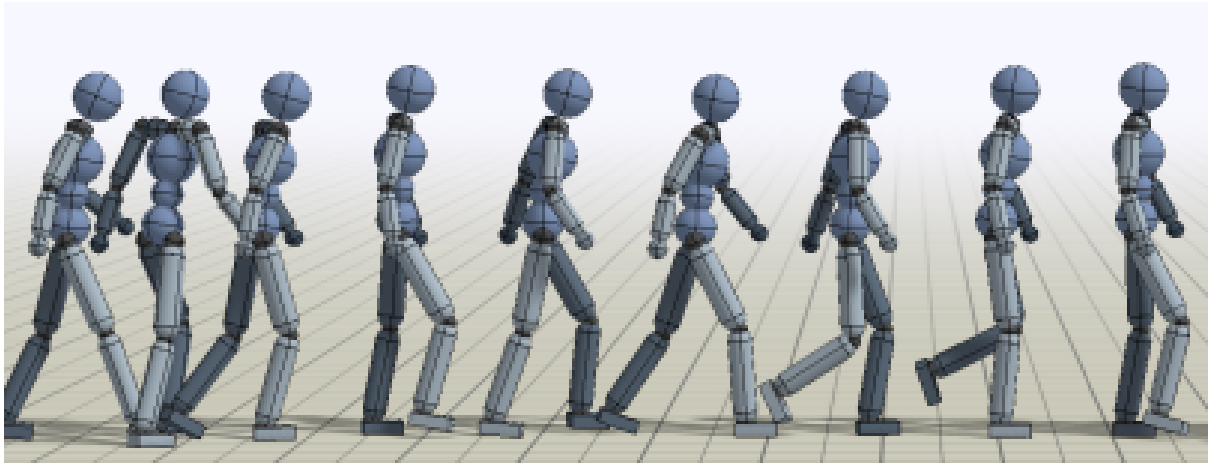

Figure 4.3: Visualization of the walking motion generated by the AMP model.
Peng et al. (2021)

Figure 4.3 shows the walking motion produced by the AMP model. The walking behavior is characterized by smooth transitions between steps, natural joint movements, and overall stability. The significant improvement in walking realism can be attributed to several factors:

- **Enhanced Simulation Realism:** The use of Isaac Gym's GPU-based physics engine allows for more accurate and detailed simulations, capturing the nuances of human gait dynamics.
- **Efficient Training:** The AMP model's training process is highly efficient, completing in just 6 minutes Peng et al. (2021). This efficiency is due to the parallel processing capabilities of GPUs, which significantly speed up the simulation and training processes.
- **Comprehensive Reward Function:** The reward function used in the AMP model incorporates more detailed biomechanical constraints and style-rewards, leading to more natural and realistic walking behaviors.

4.0.4 Implications for Rehabilitation

Given these limitations, the traditional PPO implementation is not suitable for use in rehabilitation exoskeletons without further enhancements. For the model to be practical for rehabilitation purposes, it needs to produce walking behaviors that are not only functional but also biomechanically accurate and comfortable for patients. In contrast, the PPO with AMP model shows promise for practical application in rehabilitation exoskeletons due to its realistic walking behaviors and efficient training process.

In summary, while the PPO implementation demonstrates promising learning capabilities as evidenced by the progressive reward graphs, the resulting walking motion lacks the realism required for effective rehabilitation. In contrast, the PPO with AMP model, trained in just 6 minutes using Isaac Gym's GPU-based physics engine, exhibits realistic walking behaviors

suitable for integration with rehabilitation exoskeletons. Future work should focus on refining the reward function, enhancing the physical realism of the simulation environment, and incorporating biomechanical constraints to achieve more natural walking behaviors and generalizing the application for more use cases.

Chapter 5

Discussion and Future Works

The current study demonstrates the development and implementation of a Proximal Policy Optimization (PPO) model and an Adversarial Motion Priors (AMP) model for controlling exoskeletons. While the AMP model shows promise with its realistic walking behavior, trained efficiently in just 6 minutes using Isaac Gym's GPU-based physics engine, there remain several key areas for further research and consideration.

5.1 Potential Ethical Issues

One significant concern is the potential ethical implications of the research, particularly regarding the militarization of exoskeleton technology. While the primary aim is to enhance rehabilitation for patients, the advancements in exoskeleton control could be adapted for military purposes, raising ethical questions about the use of such technology in combat situations. It is crucial for researchers and developers to consider the broader implications of their work and ensure that ethical guidelines and regulations are in place to prevent misuse.

5.2 Experimental Validation

Another limitation of the study is the lack of experimental validation. While simulation provides a valuable and risk-free environment for training and testing the models, it cannot fully replicate the complexities of real-world scenarios. Therefore, the results must be validated through experimental trials involving real exoskeletons and human participants to confirm their efficacy and safety.

5.3 Simulation Training

To minimize the risk of damage to the exoskeleton machinery and to ensure the safety of participants, the models are trained extensively in simulation environments. This approach allows us to refine the control policies and ensure robust performance before deploying them in real-world applications. The use of advanced simulation tools like Isaac Gym enables rapid and realistic training, reducing the time and cost associated with physical testing.

5.4 Challenges of Real-World Implementation

Implementing exoskeletons in real-world settings presents several challenges:

1. **Physical Realism:** Despite the advanced simulations, transferring the learned policies to physical exoskeletons may reveal discrepancies due to differences in real-world dynamics and unforeseen variables.
2. **Human Factors:** Each patient's unique biomechanics and rehabilitation needs require personalized adjustments to the exoskeleton control policies, necessitating extensive customization and fine-tuning.
3. **Safety and Reliability:** Ensuring the safety and reliability of exoskeletons in real-world use is paramount. Rigorous testing and validation are needed to prevent malfunctions that could harm users.
4. **Regulatory Approval:** Gaining regulatory approval for medical devices involves comprehensive documentation and evidence of safety and efficacy, which can be a lengthy and complex process.

5.5 Future Works

Moving forward, several areas warrant further investigation:

1. **Experimental Trials:** Conducting experimental trials with actual exoskeletons and human participants to validate the effectiveness and safety of the trained policies.
2. **Ethical Frameworks:** Developing and adhering to ethical frameworks to guide the responsible use of exoskeleton technology, with a particular focus on preventing militarization.
3. **Personalization:** Enhancing the adaptability of the exoskeleton control policies to cater to individual patient needs, improving the overall effectiveness of rehabilitation.
4. **Hybrid Training Approaches:** Combining simulation-based training with real-world fine-tuning to bridge the gap between simulated environments and actual use cases.
5. **Long-term Studies:** Investigating the long-term impacts of exoskeleton-assisted rehabilitation on patients' recovery trajectories and quality of life.

In conclusion, while the study demonstrates significant advancements in exoskeleton control using reinforcement learning models, it also highlights the need for ethical considerations, experimental validation, and real-world testing. By addressing these challenges, the development of safe, effective, and ethically sound exoskeleton technologies that can profoundly benefit rehabilitation practices is ensured.

This comprehensive approach not only advances the field of exoskeleton research but also ensures that the technology is developed responsibly and with a clear focus on improving patient outcomes.

Chapter 6

Conclusions

This dissertation presents a theoretical framework for enhancing exoskeleton-assisted rehabilitation by integrating Proximal Policy Optimization (PPO) with Adversarial Motion Priors (AMP). The primary objective was to develop a methodology that dynamically adjusts the resistance profile of the exoskeleton based on the patient's force output while ensuring adherence to natural human movement patterns.

The proposed approach leverages the strengths of PPO, a state-of-the-art reinforcement learning algorithm, to optimize the exoskeleton's control policy. By incorporating patient effort into the reward function, the system encourages active participation from the patient, which is critical for effective rehabilitation. The integration of AMP ensures that the movements facilitated by the exoskeleton remain within natural limits, enhancing both the safety and comfort of the patient. This dual approach theoretically enhances the adaptability and efficacy of rehabilitation devices, potentially leading to more personalized and effective therapy.

The significance of these findings lies in the potential to revolutionize the field of exoskeleton-assisted rehabilitation. Traditional rehabilitation methods often lack the ability to dynamically adapt to the patient's needs in real-time. The proposed methodology addresses this gap by continuously monitoring patient effort and adjusting assistance accordingly. This dynamic adjustment could accelerate recovery times, improve rehabilitation outcomes, and enhance the overall quality of life for patients undergoing therapy.

Despite the promising theoretical foundation, this study acknowledges several limitations. The absence of an actual exoskeleton model and reliance on simulated environments mean that the findings are yet to be validated experimentally. Simulations, while useful for preliminary exploration, may not fully capture the complexities and variances of real-world patient interactions. Furthermore, the effectiveness of the AMP is contingent upon the quality and comprehensiveness of the motion dataset used for training the adversarial network.

Additionally, potential ethical concerns must be considered, particularly regarding the militarization of exoskeleton technology. While our primary focus is on rehabilitation, the advancements in exoskeleton control could be adapted for military purposes, raising ethical questions about the use of such technology in combat situations. Ensuring ethical guidelines and regulations are in place is crucial to prevent misuse.

Future research should prioritize experimental validation of the proposed framework. Developing a prototype exoskeleton and conducting clinical trials will be crucial steps in testing the practical

applicability and effectiveness of the methodology. Moreover, exploring other reinforcement learning algorithms and motion prior techniques could provide further improvements and robustness to the control system. Collecting and utilizing comprehensive datasets of natural human motions will enhance the accuracy of the AMP, ensuring better adherence to natural movement patterns.

The potential challenges in real-world implementation, such as integrating sensors, processing real-time data, and ensuring patient safety, must also be addressed. Advanced sensors, improved computational efficiency, and rigorous safety protocols will be essential in overcoming these challenges. Collaborations with clinicians and rehabilitation specialists will be invaluable in refining the system to meet practical requirements and patient needs.

In conclusion, the integration of PPO with AMP represents a promising direction for the development of adaptive and effective exoskeleton-assisted rehabilitation systems. This research contributes to the foundation for future advancements in assistive technologies, with the potential to significantly impact the field of rehabilitation and improve the lives of patients requiring physical therapy. By providing a theoretically sound and adaptable framework, this study opens new avenues for creating more responsive and personalized rehabilitation aids, ultimately advancing the quality and efficacy of patient care.

This study's comprehensive approach not only advances the field of exoskeleton research but also ensures that the technology is developed responsibly, with a clear focus on improving patient outcomes and quality of life.

Number of words until this point, excluding front matter: 7208.

Bibliography

n.d.

Abbeel, P. and Ng, A.Y., 2004. Apprenticeship learning via inverse reinforcement learning [Online]. *Proceedings of the twenty-first international conference on machine learning*. New York, NY, USA: Association for Computing Machinery, ICML '04, p.1. Available from: https://doi.org/10.1145/1015330.1015430.

AYAS, M.S. and ALTAS, I.H., 2017. Fuzzy logic based adaptive admittance control of a redundantly actuated ankle rehabilitation robot. *Control engineering practice* [Online], 59, p.44–54. Available from: https://doi.org/10.1016/j.conengprac.2016.11.015.

Banala, S.K., Kim, S.H., Agrawal, S.K. and Scholz, J.P., 2009. enRobot assisted gait training with active leg exoskeleton (ALEX). *Ieee trans. neural syst. rehabil. eng.*, 17(1), pp.2–8.

Baud, R., Fasola, J., Vouga, T., Ijspeert, A. and Bouri, M., 2019. Bio-inspired standing balance controller for a full-mobilization exoskeleton [Online]. *2019 ieee 16th international conference on rehabilitation robotics (icorr)*. IEEE. Available from: https://doi.org/10.1109/icorr.2019.8779440.

Baud, R., Manzoori, A.R., Ijspeert, A. and Bouri, M., 2021. enReview of control strategies for lower-limb exoskeletons to assist gait. *J. neuroeng. rehabil.*, 18(1), p.119.

Bayon, C., Emmens, A.R., Afschrift, M., Van Wouwe, T., Keemink, A.Q.L., Kooij, H. van der and Asseldonk, E.H.F. van, 2020. Can momentum-based control predict human balance recovery strategies? *Ieee transactions on neural systems and rehabilitation engineering* [Online], 28(9), p.2015–2024. Available from: https://doi.org/10.1109/tnsre.2020.3005455.

Boutilier, C., Dean, T.L. and Hanks, S., 2011. Decision-theoretic planning: Structural assumptions and computational leverage. *Corr* [Online], abs/1105.5460. 1105.5460, Available from: http://arxiv.org/abs/1105.5460.

Coumans, E. and Bai, Y., 2016. Pybullet, a python module for physics simulation for games, robotics and machine learning.

Deng, M.Y., Ma, Z.Y., Wang, Y.N., Wang, H.S., Zhao, Y.B., Wei, Q.X., Yang, W. and Yang, C.J., 2019. enFall preventive gait trajectory planning of a lower limb rehabilitation exoskeleton based on capture point theory. *Front. inf. technol. electron. eng.*, 20(10), pp.1322–1330.

Doya, K., 2000. Reinforcement learning in continuous time and space. *Neural computation* [Online], 12(1), p.219–245. Available from: https://doi.org/10.1162/089976600300015961.

Escontrela, A., Peng, X.B., Yu, W., Zhang, T., Iscen, A., Goldberg, K. and Abbeel, P., 2022. Adversarial motion priors make good substitutes for complex reward functions [Online]. *2022 ieee/rsj international conference on intelligent robots and systems (iros)*. pp.25–32. Available from: https://doi.org/10.1109/IROS47612.2022.9981973.

Fan, L., Zhu, Y., Zhu, J., Liu, Z., Zeng, O., Gupta, A., Creus-Costa, J., Savarese, S. and Fei-Fei, L., 2018. Surreal: Open-source reinforcement learning framework and robot manipulation benchmark [Online]. In: A. Billard, A. Dragan, J. Peters and J. Morimoto, eds. *Proceedings of the 2nd conference on robot learning*. PMLR, *Proceedings of Machine Learning Research*, vol. 87, pp.767–782. Available from: https://proceedings.mlr.press/v87/fan18a.html.

Fang, J., Lee, V.C., Ji, H. and Wang, H., 2022. Enhancing digital health services: A machine learning approach to personalized exercise goal setting [Online]. [Online]. Available from: https://doi.org/10.48550/ARXIV.2204.00961.

Huang, R., Pengl, Z., Cheng, H., Hu, J., Qiu, J., Zou, C. and Chen, Q., 2018. Learning-based walking assistance control strategy for a lower limb exoskeleton with hemiplegia patients [Online]. *2018 ieee/rsj international conference on intelligent robots and systems (iros)*. IEEE. Available from: https://doi.org/10.1109/iros.2018.8594464.

Huo, W., Mohammed, S., Moreno, J.C. and Amirat, Y., 2016. Lower limb wearable robots for assistance and rehabilitation: A state of the art. *Ieee syst. j.*, 10(3), pp.1068–1081.

Karunakaran, K.K., Abbruzzese, K., Androwis, G. and Foulds, R.A., 2020. A novel user control for lower extremity rehabilitation exoskeletons. *Frontiers in robotics and ai* [Online], 7. Available from: https://doi.org/10.3389/frobt.2020.00108.

Kiran, B.R., Sobh, I., Talpaert, V., Mannion, P., Sallab, A.A.A., Yogamani, S. and Perez, P., 2022. Deep reinforcement learning for autonomous driving: A survey. *Ieee transactions on intelligent transportation systems* [Online], 23(6), p.4909–4926. Available from: https://doi.org/10.1109/tits.2021.3054625.

Kober, J., Bagnell, J.A. and Peters, J., 2013. Reinforcement learning in robotics: A survey. *The international journal of robotics research* [Online], 32(11), p.1238–1274. Available from: https://doi.org/10.1177/0278364913495721.

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D. and Wierstra, D., 2015. Continuous control with deep reinforcement learning [Online]. Available from: https://doi.org/10.48550/ARXIV.1509.02971.

Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A. and State, G., 2021. Isaac gym: High performance gpu-based physics simulation for robot learning [Online]. Available from: https://doi.org/10.48550/ARXIV.2108.10470.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. and Hassabis, D., 2015. Human-level control through deep reinforcement learning. *Nature* [Online], 518(7540), p.529–533. Available from: https://doi.org/10.1038/nature14236.

Moreno, J.C., Figueiredo, J. and Pons, J.L., 2018. Exoskeletons for lower-limb rehabilitation. *Rehabilitation robotics*. Elsevier, pp.89–99.

Peng, X.B., Ma, Z., Abbeel, P., Levine, S. and Kanazawa, A., 2021. AMP: adversarial motion priors for stylized physics-based character control. *Corr* [Online], abs/2104.02180. 2104.02180, Available from: `https://arxiv.org/abs/2104.02180`.

Pons, J.L., 2010. Rehabilitation exoskeletal robotics. *Ieee engineering in medicine and biology magazine* [Online], 29(3), pp.57–63. Available from: `https://doi.org/10.1109/MEMB.2010.936548`.

Puterman, M.L., 1994. en*Markov decision processes*, Wiley Series in Probability & Mathematical Statistics: Applied Probability & Statistics. Nashville, TN: John Wiley & Sons.

Rudin, N., Hoeller, D., Bjelonic, M. and Hutter, M., 2022. Advanced skills by learning locomotion and local navigation end-to-end [Online]. Available from: `https://doi.org/10.48550/ARXIV.2209.12827`.

Schrade, S.O., Dätwyler, K., Stücheli, M., Studer, K., Türk, D.A., Meboldt, M., Gassert, R. and Lambercy, O., 2018. Development of varileg, an exoskeleton with variable stiffness actuation: first results and user evaluation from the cybathlon 2016. *Journal of neuroengineering and rehabilitation* [Online], 15(1). Available from: `https://doi.org/10.1186/s12984-018-0360-4`.

Schulman, J., Levine, S., Abbeel, P., Jordan, M. and Moritz, P., 2015. Trust region policy optimization [Online]. In: F. Bach and D. Blei, eds. *Proceedings of the 32nd international conference on machine learning*. Lille, France: PMLR, *Proceedings of Machine Learning Research*, vol. 37, pp.1889–1897. Available from: `https://proceedings.mlr.press/v37/schulman15.html`.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O., 2017. Proximal policy optimization algorithms [Online]. Available from: `https://doi.org/10.48550/ARXIV.1707.06347`.

Shi, D., Zhang, W., Zhang, W. and Ding, X., 2019. A review on lower limb rehabilitation exoskeleton robots. *Chinese journal of mechanical engineering* [Online], 32(1). Available from: `https://doi.org/10.1186/s10033-019-0389-8`.

Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Driessche, G. van den, Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T. and Hassabis, D., 2016. Mastering the game of go with deep neural networks and tree search. *Nature* [Online], 529(7587), p.484–489. Available from: `https://doi.org/10.1038/nature16961`.

Sutton, R.S. and Barto, A.G., 2020. *Reinforcement learning: An introduction*. The MIT Press.

Vouga, T., Baud, R., Fasola, J., Bouri, M. and Bleuler, H., 2017. Twiice — a lightweight lower-limb exoskeleton for complete paraplegics [Online]. *2017 international conference on rehabilitation robotics (icorr)*. IEEE. Available from: `https://doi.org/10.1109/icorr.2017.8009483`.

Watkins, C.J.C.H. and Dayan, P., 1992. Q-learning. *Machine learning* [Online], 8(3–4), p.279–292. Available from: `https://doi.org/10.1007/bf00992698`.

Wiering, M. and Otterlo, M.v., 2012. *Reinforcement learning state-of-the-art*. Springer Berlin Heidelberg.

Xiong, M., 2014. Research on the control system of the lower limb rehabilitation robot under the single degree of freedom. *Applied mechanics and materials* [Online], 643, p.15–20. Available from: `https://doi.org/10.4028/www.scientific.net/amm.643.15`.

Zhang, B., Rajan, R., Pineda, L., Lambert, N., Biedenkapp, A., Chua, K., Hutter, F. and Calandra, R., 2021. On the importance of hyperparameter optimization for model-based reinforcement learning [Online]. In: A. Banerjee and K. Fukumizu, eds. *Proceedings of the 24th international conference on artificial intelligence and statistics*. PMLR, *Proceedings of Machine Learning Research*, vol. 130, pp.4015–4023. Available from: `https://proceedings.mlr.press/v130/zhang21n.html`.

Appendix A

Table of Hyper-parameters

Hyperparameter	Description	Value(s) Used	Notes
Learning Rate (lr)	Step size used by the optimizer	2.5e-4	Fine-tuned for optimal performance
Num Batch Time Steps	Number of time steps in each batch	4800	Defined for each batch during training
Max Time Steps per Episode	Maximum number of time steps per episode	1600	Limits the length of each episode
Num Net Updates per Iteration	Number of network updates per iteration	20	Determines the frequency of updates
Gamma	Discount factor for future rewards	0.99	Balances immediate and future rewards
Render	Whether to render the environment	True	Used for visualization during training
Seed	Random seed for reproducibility	None	Ensures consistent results if set
Normalize	Whether to normalize input data	True	Helps in stabilizing the learning process
Clip Range	Clipping range for policy updates	0.2	Used in PPO to maintain policy stability

Table A.1: The hyperparameters used in this study were sourced from the official RL Zoo GitHub repository. The detailed hyperparameter configurations for PPO can be found in the provided link: https://github.com/araffin/rl-baselines-zoo/blob/master/hyperparams/ppo2.yml