Learning-Based Control of Transtibial Assistive Devices in Physics-Informed Locomotion Simulations

Jonathan (Jintong) He
Mechanical Engineering
Carnegie Mellon University
Pittsburgh, USA
jintongh@andrew.cmu.edu

Shawn Krishnan
Mechanical Engineering
Carnegie Mellon University
Pittsburgh, USA
shawnakk@andrew.cmu.edu

Manuel Lancastre
Mechanical Engineering
Carnegie Mellon University
Pittsburgh, USA
manuella@andrew.cmu.edu

Eric Zhao
Mechanical Engineering
Carnegie Mellon University
Pittsburgh, USA
ericzhao@andrew.cmu.edu

***Abstract*—Individuals with lower extremity impairments often face significant mobility challenges, as current control strategies for transtibial prostheses and exoskeletons may not be adequately adapted to varied environments and user-specific needs. Traditional control methods, such as finite-state machines and impedance control, can be limited in their ability to handle the dynamic and complex nature of human locomotion. To address these limitations, our research explores the application of deep learning techniques to control transtibial prostheses and exoskeletons to improve mobility for individuals with lower extremity limitations. We propose a hybrid learning pipeline that integrates imitation learning (IL) and deep reinforcement learning (RL) to develop adaptive control policies for ankle assistive devices. Using the Mujoco physics engine and Loco-Mujoco framework, we implemented a two-agent system: a Variational Adversarial Imitation Learning (VAIL) agent that controls the humanoid body and a separate agent that manages control of the ankle-level assistive device. Additionally, we created a Proximal Policy Optimization (PPO) based reinforcement learning model that can replicate natural human locomotion. We achieved promising results by combining both approaches. First establishing baseline behavior through imitation learning, then fine-tuning with reinforcement learning to incorporate environmental adaptation. We also found that simpler multilayer perceptron actor-critic models outperformed more complex transformer architectures. This work demonstrates the potential of hybrid learning approaches for developing adaptable, user-compatible prosthetic controllers that can function effectively across diverse environments and user conditions.**

I. Introduction

Approximately 150,000 people undergo a lower extremity amputation each year [1], and in people over 40 years of age with both diagnosed diabetes and lower extremity disease, 33% reported difficulty walking a quarter mile and climbing 10 steps without rest [2]. The application of machine learning for problem-solving has been revolutionized over the last decade. These advancements have led to groundbreaking developments in a multitude of fields, especially in robotics and controls. Particularly, advancements in reinforcement learning [3] have produced highly effective algorithms to solve complex control problems without the need for explicit dynamic models, which can be difficult to implement. We intend to use these advances to address mobility issues in patients with limited lower body control.

There are many modern approaches to this challenge that use traditional control methods, such as adaptive whole-body dynamics with joint torque output [4]. These strategies have proven to be highly effective. However, they lack one key requirement: adaptability and compatibility between different users and environments. Recently, an increasing number of approaches have been published that utilize reinforcement learning as the primary adaptive control algorithm to learn weights and parameters for a controller to input torques into a prosthetic or an exoskeletal device. This enables the devices to apply appropriate torques and maintain weight balance to mimic ankle kinematics [5].

Our work builds on recent advances by proposing a custom control pipeline that combines imitation learning (IL) and deep reinforcement learning (RL) to train policies to understand ankle kinematics and human gait control. We use IL to efficiently bootstrap the agent's walking behavior from expert demonstrations, providing a structured prior that avoids unstable or random exploration during early training. However, since IL alone lacks adaptability and struggles with recovery in unseen scenarios, we incorporate RL to fine-tune the learned policy. This enables the agent to adapt to environmental variability and improve long-term stability. Finally, our objective is to develop a comprehensive set of metrics to evaluate the performance of the policy in terms of adaptability, compatibility, and practical deployability for transtibial prosthetic or exoskeletal devices.

Our preliminary results show that training an ankle actuator using pure imitation learning is inefficient, as the agent is unable to walk more than a few steps in a simple flat ground, constant speed environment. Pure reinforcement learning also produces undesirable results due to unpredictable learning behavior. Our best results come from a combination of imitation and reinforcement learning, by first training a baseline model using imitation learning, which is then fine tuned using reinforcement learning. We have also produced better results in model training using a simple multilayer perception actor-critic model, as opposed to more complex models such as transformers.

II. Related Work

A. Modern Control Methods for Assistive Devices

Current assistive devices like prosthetics and exoskeletons rely on actuators to provide external forces to assist or mimic the function of an individual's limbs in everyday tasks. This means that advanced control methods have to be applied in order to increase the usability of these devices in a wide range of scenarios, from sitting and standing to running. As a result, a variety of control methods have been proposed for achieving these desired functionalities.

1) Model-based Control: Model-based control methods employ mathematical models of the system to create and test control strategies. Impedance control is widely used in exoskeletons to regulate the dynamic relationship between the device and user [6]. This approach enables compliant human-robot interactions that prioritize safety and comfort. Model predictive control (MPC) has shown effectiveness in prosthetics by anticipating future states and optimizing trajectories while respecting constraints. Manchola et al. [7] demonstrated an MPC framework for powered ankle prostheses that balances performance with power consumption. These approaches perform well when accurate models are available but struggle with the inherent complexity and variability of human movement and unpredictable environments.

2) Model-free Control: Model-free control methods overcome the limitations of explicit modeling by establishing direct mappings between sensor inputs and control outputs. Electromyography (EMG)-based control is particularly prevalent, using muscle activity signals to predict user intent. Woodward and Hargrove [8] developed a real-time EMG pattern recognition system for lower limb prostheses that classifies locomotion modes with high accuracy. Finite state machines offer another approach, as demonstrated by Young et al. [9], who implemented an intent recognition system for powered lower limb prostheses using mechanical sensors to transition between activity modes. Sensor fusion techniques that combine multiple data sources have further improved robustness and adaptability, as shown by Li et al. [10] in their real-time adaptive assistance system for exoskeletons. The drawbacks of these approaches is that they are often fine tuned to individual users and specific conditions, making them less adaptable to variations across different people and dynamic environments as well as a difficulty scaling these solutions.

3) Deep Learning Methods: Deep learning approaches represent the cutting edge in prosthetic and exoskeleton control, offering powerful tools for handling complex, high-dimensional data without explicit modeling. Reinforcement learning (RL) has emerged as particularly promising for developing adaptive controllers. Wen et al. [11] demonstrated how deep RL can learn to generate appropriate torque commands for a powered prosthetic leg across different walking conditions. Imitation learning combines aspects of supervised learning and RL to develop controllers that mimic expert behavior. Idzikowski et al. [12] used this approach to train neural network policies that reproduce natural ankle prosthesis behavior. Recent work has explored integration of these methods. Chen et al. [13] proposed a model-based reinforcement learning framework that learns dynamics models while optimizing control policies. This hybrid approach combines sample efficiency with adaptability. Zhang et al. [14] pushed this further with a personalized assistance policy that continuously adapts to user fatigue and environmental conditions using meta-learning techniques, enabling rapid adaptation to new users with minimal calibration. Unlike traditional model-free control methods that rely on reactive mappings from sensor data to actions, these learning-based approaches can generalize across users and environments by optimizing over long-term outcomes. Deep reinforcement learning methods are also more sample efficient, as data is collected from simulation as opposed to physical data collection with the user. By combining the structure of imitation learning with the adaptability of reinforcement learning, they offer greater robustness and personalization without requiring extensive manual tuning or per-user calibration.

Despite these advances, challenges remain in developing practical control systems. Sample efficiency is critical for reinforcement learning approaches, as collecting real-world human interaction data is time-consuming. Sim-to-real transfer techniques, as explored by Song et al. [15], offer solutions by pre-training policies in simulation. Additionally, personalization remains essential due to individual differences in anatomy and movement patterns. Our work builds upon these advances by developing a custom pipeline that combines imitation learning with deep reinforcement learning specifically for ankle kinematics and human gait locomotion control. This approach aims to achieve both sample efficiency during training and adaptability during deployment, addressing key limitations of existing methods.

III. Methods

The objective of this study is to develop and evaluate an ankle prosthesis with one degree of freedom, utilizing reinforcement learning and imitation learning techniques to enable stable locomotion in the humanoid agent. The key challenge is simulating a humanoid agent with an ankle prosthesis, since the standard humanoid agent has full access to the state of all body joints, whereas the prosthetic ankle has only limited states to reference.

A. Environment Selection

The simulation environment chosen for this study is Mujoco (Multi-Joint dynamics with Contact) [16]. Mujoco is a physics engine specifically designed for simulating articulated structures in robotics and biomechanical applications. Mujoco enables high-fidelity modeling of dynamic interactions by providing fast and accurate numerical integration, contact handling, and optimization-based physics.

B. Humanoid Model Selection

The humanoid model employed is the torque-based humanoid from Loco-Mujoco [17]. Loco-Mujoco is a physics-based reinforcement learning framework designed for simulating locomotion in humanoid and legged robots. Loco-Mujoco provides realistic control mechanisms and allows for the application of torque-based actuation, which simplifies the complexity of muscle excitations while maintaining biologically plausible joint structures. The humanoid model used in this study consists of 36 state variables representing the agent's kinematic properties and 12 action outputs that govern joint torques. The detailed state and action representations are illustrated in Table II and III. Torque-driven control was chosen over muscle-driven models in this study because the primary focus is on the plausibility of gait kinematics rather than muscle activations. Additionally, muscle-driven models are computationally expensive, making them impractical for our current framework.

C. Baseline via Imitation Learning

Initially, we trained a multilayer perceptron (MLP) agent using imitation learning to closely replicate expert gait behavior. The agent was trained to minimize the mean squared error (MSE) between the agent's ankle torque predictions and the expert data provided by LocoMuJoCo's Variational Adversarial Imitation Learning (VAIL) model. The training of the MLP agent follows Fig. 1. A VAIL agent is trained first using the expert dataset, which teaches the model how to walk. This trained VAIL agent also provides dataset for training the MLP agent. A filter is applied to the output of the VAIL agent to keep only the selected states and the expert ankle action. The expert action serves as the ground truth for the MLP model, driving the ankle action to mimic expert movements. Upon training the individual agents, both VAIL and MLP agents are integrated into a unified system, as shown in Fig. 2. At each simulation step, the state space is provided to both agents. The VAIL agent outputs 12 actions as usual, but the MLP agent's output replaces the right ankle action of the VAIL agent. The resulting action space is then applied to update the simulation state for the next step.

D. Reinforcement Learning Implementation

Building upon the IL baseline, we developed a PPO (Proximal Policy Optimization) agent with incremental complexity to refine and adapt the humanoid model. Three iterations of PPO were tested:

- **Iteration 1 (Vanilla Policy Gradient):** Served as a preliminary model for discrete-action benchmarks; however, it performed poorly in continuous environments due to instability.
- **Iteration 2 (Basic PPO):** Included gradient clipping for stability improvements in continuous environments, showing promising results but struggled with sparse reward scenarios.

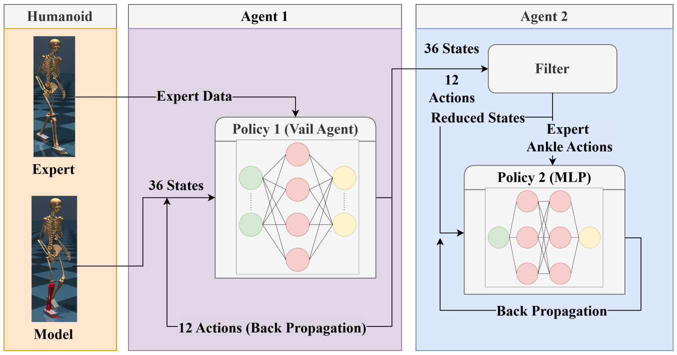

Fig. 1. Flowchart illustrating the two-agent system used for training the humanoid model with an ankle prosthesis. The VAIL agent (Agent 1) learns from expert data to control the humanoid, while the MLP-based prosthesis agent (Agent 2) learns to predict ankle actions based on filtered state information.

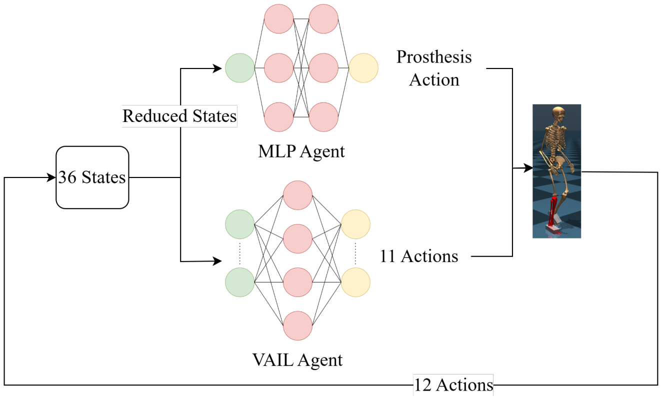

Fig. 2. Diagram showing the evaluation of the system, demonstrating the mapping from 36 state inputs into 12 action outputs by combining the action space of the two agents implemented.

- **Iteration 3 (Enhanced PPO):** Incorporated Generalized Advantage Estimation (GAE), randomized batch sampling, and policy reuse for more stable and effective updates.

Our final PPO implementation used a reward structure encouraging continuous locomotion:

$$\text{Reward} = \begin{cases} 1, & \text{per successful step} \\ -100, & \text{if fallen} \end{cases} \tag{1}$$

A built-in `_has_fallen()` function monitored gait stability, penalizing the agent if it exceeded physiological thresholds for pelvic orientation and height.

IV. Experiment

A. Imitation Learning Experiment

The overall objective of this experiment was to quantitatively and qualitatively compare the prosthetic ankle kinematics (joint angles) and kinetics (torques) between the expert agent and the MLP imitation learning agent. The experiment was conducted as follows:

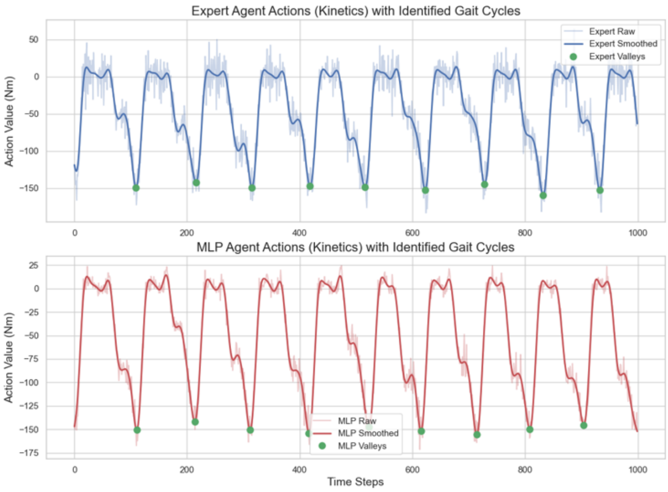

Fig. 3. Comparison of prosthetic ankle torque actions between Expert and MLP agents. The top plot shows the Expert agent's torque actions with identified valleys marking individual gait cycles. The bottom plot displays the corresponding torque actions of the MLP imitation learning agent. Raw data (lighter lines), smoothed curves (darker lines), and identified valleys (green dots) are presented.

1) Conducted two independent simulation rollouts of the expert agent and the MLP imitation learning agent, both resulting in large datasets of kinematic and kinetic information.
2) Applied a low-pass Butterworth filter to filter the generated data, reducing noise, and facilitating analysis.
3) Identified gait cycles by locating local minima (valleys) in plots of kinetic data (torque). These local minima signify transitions between adjacent gait cycles.
4) Split data into individual gait cycles based on intervals between discovered valleys.
5) Selected five consecutive gait cycles from both datasets for close inspection.
6) The mean curve and standard deviation (STD) shading were computed for the selected cycles from both expert and imitation learning datasets to display similarity and variability.
7) Calculated the mean absolute error (MAE) between the respective gait cycles of the MLP imitation learning agent and the expert agent to provide quantitative performance measurement.

Fig. 3 illustrates the details of how the data is processed. Two separate rollouts are conducted for both the expert and MLP agent, generating kinematics and dynamics data for analysis. After passing through a low-pass filter, the valleys in the data are identified, showing in green dots in the graph. Following that, these separate gait cycles are combined and plotted for further analysis.

B. Reinforcement Learning Experiment

The objective of this experiment was to develop a reinforcement learning (RL) policy capable of generating joint torques and kinematic patterns that enabled a humanoid agent to walk—without relying on any pretrained controller. Once we qualitatively observed successful locomotion behavior, we proceeded with quantitative analysis to validate the performance of the learned policy.

We began by implementing Proximal Policy Optimization (PPO) from OpenAI's Spinning Up library (Schulman et al., 2017), training it across three progressively complex environments to build a robust locomotion policy. These environments included a balance control task, a two-torque linkage walking game, and finally, a full-body humanoid locomotion simulation in MuJoCo.

In the initial balance task, a vanilla policy gradient approach resulted in unstable parameter updates and suboptimal behavior in continuous control. Introducing PPO's clipped objective significantly improved training stability and led to more promising results in downstream tasks.

We conducted rollouts using observation spaces of 36, 22, and 16 dimensions, and found that the policy trained with 22 features produced the most consistent gait. Qualitative assessment via rollout video confirmed that the agent successfully achieved stable walking behavior.

V. Results

A. Imitation Learning Results

The MLP agent was able to accurately replicate the kinematics and kinetics of the ankle conducted by the VAIL agent under controlled conditions, indicating its capability as a baseline model. Fig. 6 provides a close comparison of the 36-state MLP agent, the 22-state MLP agent, and the 22-state MLP agent with altered prosthetic properties (lower mass and inertia of the right foot). Both 36-state and 22-state MLP agents were very close to the expert agent in both kinematics and kinetics, which demonstrates that the reduction of the input state space from 36 states to 22 states did not have any significant negative impact on the overall performance of the predicted action values. Furthermore, the close alignment between the standard 22-state agent and the 22-state agent with prosthetic properties demonstrates the effectiveness and robustness of MLP-based imitation learning under various prosthetic conditions. Table I presents the quantitative performance metrics of the Raw VAIL agent compared to various MLP imitation learning agents with differing input state dimensions. The metrics used are the average steps an agent can walk before falling and the mean MAE in Nm between the expert and MLP agents. The results indicate that agents trained with 36 and 22 states (including the 22-state prosthesis-adjusted agent) maintained relatively high performance, closely matching the expert in terms of both average steps per rollout and mean absolute error (MAE) in torque predictions.

B. Reinforcement Learning (RL) Results

We have preliminary qualitative results indicating that Proximal Policy Optimization (PPO) reinforcement learning is effective in mimicking natural human locomotion. Through a series of iterative experiments, we transitioned from vanilla

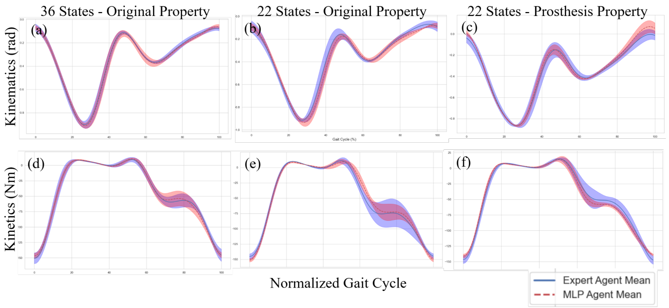

Fig. 4. Comparison of ankle joint kinematics (top row) and kinetics (bottom row) between Expert agent (solid blue line) and MLP agent (dashed red line) across different conditions: (a, d) 36-state MLP agent, (b, e) 22-state MLP agent, and (c, f) 22-state MLP agent with adjusted prosthetic properties (reduced mass and inertia).

TABLE I
PERFORMANCE COMPARISON OF RAW VAIL AGENT AND VARIOUS MLP AGENTS WITH DIFFERENT INPUT DIMENSIONS AND PROSTHETIC PROPERTIES.

Agent	**Avg. Steps per Rollout**	**Mean MAE (Nm)**
Raw VAIL Agent	56,211.2	0.00
MLP - 36 States	41,799.6	9.91
MLP - 22 States	22,526.7	10.27
MLP - 22 States (Prosthesis)	43,087.8	10.17

policy gradient methods—limited by unstable updates in continuous action spaces—to enhanced PPO implementations that incorporated features such as generalized advantage estimation (GAE), policy clipping, and batch randomization. These improvements led to more stable learning dynamics and enabled our agent to achieve sustained upright walking. In particular, a reward structure focused on survival and penalizing falls proved effective for training a walking policy without imitation, allowing the agent to autonomously learn gait-like behavior that qualitatively resembles human locomotion.

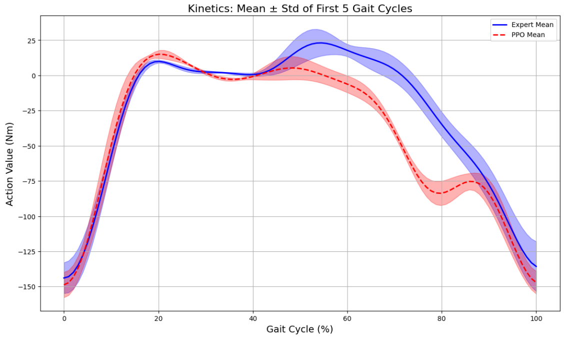

Fig. 5. Mean torque profiles (±1 standard deviation) are shown across the first 5 gait cycles, normalized to percent of the gait cycle. The expert agent (blue) exhibits smoother and more consistent torque control, while the PPO policy (red) approximates the overall pattern but with greater variability and reduced peak torque during mid-stance and push-off phases.

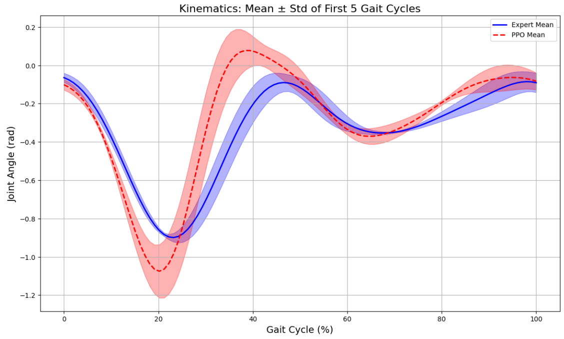

Fig. 6. Mean joint angle trajectories (±1 standard deviation) across the first 5 gait cycles, normalized to percent of the gait cycle. The expert (blue) displays smoother and more consistent joint motion, while the PPO-trained agent (red) captures the general trend but shows greater deviation during mid-stance and late stance phases, indicating less precise control of joint positioning.

VI. DISCUSSION, LIMITATIONS, CONCLUSIONS, AND FUTURE

The performance is limited when training an ankle actuator with only imitation learning as the agent cannot adapt to unseen situations such as states not present in the expert dataset. This is due to the agent not learning to adapt to various environmental circumstances during training and only learning how to copy the expert. But this can serve as a baseline to finetune with reinforcement learning given the adequate reward function and model to transition from IL to RL. Models trained with only reinforcement learning do not converge to stable human walking. It's challenging to train an ankle torque actuator from scratch only using environmental rewards, as this doesn't give the agent any baseline to follow. Additionally, training an ankle torque actuator from scratch only using rewards to imitate an expert is also difficult, as the agent will be able to walk a few steps but is unable to deal with compounding errors in its actions. Combining the best of both approaches, we can train the agent using only IL until the agent can walk a few steps, and then add environmental rewards such as a penalty for falling or tracking the stability of the center of mass to teach the agent to adapt while using the expert model as a baseline. Our best results have come from this, specifically introducing the environmental reward of not falling after a few hundred epochs, or when the agent can walk a few steps.

Another takeaway is the ankle actuator performance in terms of time before falling doesn't correlate with the performance of the agent it was trained on, as we see the best and most generalizable ankle actuator was trained on the agent trained in 180 epochs. This means there should be a balance between having perfect data and data with more variety (exploration vs exploitation in reinforcement learning), and that data can have a significant impact on the model performance.

The input features to the model is a crucial consideration, in regards to ideal inputs and outputs for the ankle actuator. It could be that reducing the input features to just the lower body could improve the model. The full state space of the humanoid model is used as the feature input for the 36 state models, and this means even upper body joints such as shoulders and wrist positions are considered in the training of the model. This may be unnecessary, as we can see that similar gait cycles are achieved across the 36 state and reduced 22 state models, with only slightly higher variance. We may be able to tune and reduce the dimensions of the input feature space to only the most significant features, using techniques such as principal component analysis.

We also discovered that simpler models perform better in the context of learning human locomotion in physics informed simulations. Simple feedforward neural networks are more capable of effectively learning and modeling human locomotion than more intensive architectures such as Transformer models.

Based on these results, we can conclude that our proposed combined imitation learning and reinforcement learning approach to learning human locomotion in physics informed simulation has potential in prosthesis and exoskeletal control applications. Our framework of creating a baseline human locomotion model using imitation learning, which is then fine tuned for specific environment and user contexts using reinforcement learning, successfully outputs a humanoid locomotion model with similar gait cycle behavior and kinetics to a perfect humanoid locomotion model. Future work should look into testing our framework on humanoid models with prosthetic replacements to determine their effectiveness as controllers for these devices, and eventually deploy these models to real world hardware.

Acknowledgment

The authors thank Dr. Inseung Kang, Assistant Professor in the Department of Mechanical Engineering at Carnegie Mellon University, and the Director of the MetaMobility Lab at CMU. Dr. Kang provided invaluable guidance throughout the course of the project, sharing his expert insight in the fields of robotics, biomechanics, and deep learning methods.

References

[1] C.S. Molina and J.B. Faulk. *Lower Extremity Amputation*. StatPearls Publishing, Treasure Island, FL, 2022. Updated 2022 Aug 22.

[2] Centers for Disease Control and Prevention. Mobility limitation among persons aged ¿40 years with and without diagnosed diabetes and lower extremity disease — united states, 1999–2002. *Morbidity and Mortality Weekly Report*, 54(46):1183–1186, 2005.

[3] Ding Wang, Ning Gao, Derong Liu, Jinna Li, and Frank L. Lewis. Recent progress in reinforcement learning and adaptive dynamic programming for advanced control applications. *IEEE/CAA Journal of Automatica Sinica*, 11(1):18–36, 2024.

[4] Ryan R. Posh, Jonathan A. Tittle, David J. Kelly, James P. Schmiedeler, and Patrick M. Wensing. Hybrid volitional control of a robotic transtibial prosthesis using a phase variable impedance controller. *arxiv*, 2023.

[5] L. Huang, J. Zheng, Y. Gao, et al. A lower limb exoskeleton adaptive control method based on model-free reinforcement learning and improved dynamic movement primitives. *Journal of Intelligent & Robotic Systems*, 111:24, 2025.

[6] Michael R Tucker, Jeremy Olivier, Anna Pagel, Hannes Bleuler, Mohamed Bouri, Olivier Lambercy, José del R Millán, Robert Riener, Heike Vallery, and Roger Gassert. Control strategies for active lower extremity prosthetics and orthotics: a review. *Journal of neuroengineering and rehabilitation*, 12(1):1–30, 2015.

[7] Martha Manchola, Lisa R Mayag, Marcela Munera, Carlos A Garcia, and Julian D Colorado. Model-based strategy for gait rehabilitation using a robotic ankle exoskeleton. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(5):972–981, 2019.

[8] Richard B Woodward and Levi J Hargrove. Real-time out-of-distribution detection in lower-limb prostheses. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(11):2229–2237, 2019.

[9] Aaron J Young, Ann M Simon, and Levi J Hargrove. Intent recognition in a powered lower limb prosthesis using time history information. *Annals of biomedical engineering*, 42(3):631–641, 2014.

[10] Juanjuan Li, Binghan Zhong, Carly Pieringer, Mengjia Liang, Hualei Zeng, Seyha Balashov, Peter Brown, Krystyna Kim, Andy Fok, Lisa Lu, et al. Human-in-the-loop optimization of exoskeleton assistance during walking. *Science*, 356(6344):1280–1284, 2019.

[11] Yue Wen, Jennie Si, Xiangyang Gao, Sharon Huang, and He Helen Huang. Reinforcement learning control of a powered prosthetic leg with deep neural network state representation. *IEEE Transactions on Control Systems Technology*, 28(5):2007–2017, 2019.

[12] Mathew Idzikowski, Madison Mitchell, Claudia Mancinelli, Suraj Nair, and Levi Hargrove. Imitation learning-based framework for learning 3d gait on a powered transfemoral prosthetic leg. *IEEE Transactions on Biomedical Engineering*, 67(12):3370–3380, 2020.

[13] Ruohan Chen, Jyothir Desai, Ramesh Raskar, and Greg Yang. Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators. *IEEE Transactions on Robotics*, 36(5):1510–1526, 2020.

[14] Tianyu Zhang, Huiqi Ma, Ashish Deshpande, and Luis Sentis. Personalized adaptive assistance policies for lower-limb exoskeletons using deep reinforcement learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 31:1598–1608, 2023.

[15] Simiao Song, John Zhai, Yilin Yang, Wanli Pang, Mingze Jia, Geraint Rees, and Neil Burgess. Sim-to-real transfer of robotic control with dynamics randomization. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 8583–8589, 2020.

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

[17] F. Al-Hafez, G. Zhao, J. Peters, and D. Tateo. Locomujoco: A comprehensive imitation learning benchmark for locomotion. *arXiv.org*, 2023. Accessed: Feb. 27, 2025.

Appendix

Table II lists all 36 state variables used in the humanoid model simulation, including positions and velocities of various joints. Table III details the 12 action outputs, each corresponding to controlled joint torques.

TABLE II
State Variables Used in Humanoid Model

Index	Description	Min	Max	Units
0	Position of Joint pelvis_ty	$-\infty$	∞	Position [m]
1	Position of Joint pelvis_tilt	$-\infty$	∞	Angle [rad]
2	Position of Joint pelvis_list	$-\infty$	∞	Angle [rad]
3	Position of Joint pelvis_rotation	$-\infty$	∞	Angle [rad]
4	Position of Joint hip_flexion_r	-0.787	0.787	Angle [rad]
5	Position of Joint hip_adduction_r	-0.524	0.524	Angle [rad]
6	Position of Joint hip_rotation_r	-2.0944	2.0944	Angle [rad]
7	Position of Joint knee_angle_r	-2.0944	0.174533	Angle [rad]
8	Position of Joint ankle_angle_r	-1.5708	1.5708	Angle [rad]
9	Position of Joint hip_flexion_l	-0.787	0.787	Angle [rad]
10	Position of Joint hip_adduction_l	-0.524	0.524	Angle [rad]
11	Position of Joint hip_rotation_l	-2.0944	2.0944	Angle [rad]
12	Position of Joint knee_angle_l	-2.0944	0.174533	Angle [rad]
13	Position of Joint ankle_angle_l	-1.0472	1.0472	Angle [rad]
14	Position of Joint lumbar_extension	-1.5708	0.377	Angle [rad]
15	Position of Joint lumbar_bending	-0.754	0.754	Angle [rad]
16	Position of Joint lumbar_rotation	-0.754	0.754	Angle [rad]
17	Velocity of Joint pelvis_tx	$-\infty$	∞	Velocity [m/s]
18	Velocity of Joint pelvis_tz	$-\infty$	∞	Velocity [m/s]
19	Velocity of Joint pelvis_ty	$-\infty$	∞	Velocity [m/s]
20	Velocity of Joint pelvis_tilt	$-\infty$	∞	Angular Velocity [rad/s]
21	Velocity of Joint pelvis_list	$-\infty$	∞	Angular Velocity [rad/s]
22	Velocity of Joint pelvis_rotation	$-\infty$	∞	Angular Velocity [rad/s]
23-35	Velocity of leg and lumbar joints	$-\infty$	∞	Angular Velocity [rad/s]

TABLE III
Normalized Action Outputs for Humanoid Model Control

Index	Action Name	Min Control	Max Control
0	mot_lumbar_ext	-1.0	1.0
1	mot_lumbar_bend	-1.0	1.0
2	mot_lumbar_rot	-1.0	1.0
3	mot_hip_flexion_r	-1.0	1.0
4	mot_hip_adduction_r	-1.0	1.0
5	mot_hip_rotation_r	-1.0	1.0
6	mot_knee_angle_r	-1.0	1.0
7	mot_ankle_angle_r	-1.0	1.0
8	mot_hip_flexion_l	-1.0	1.0
9	mot_hip_adduction_l	-1.0	1.0
10	mot_hip_rotation_l	-1.0	1.0
11	mot_knee_angle_l	-1.0	1.0
12	mot_ankle_angle_l	-1.0	1.0