ExoGym: Sim-to-Real Learning of Exoskeleton Control via Musculoskeletal Modeling and Reinforcement Learning

Zhimin Hou, Juncheng Zhou, Ivan Lopez-Sanchez, Jin Sen Huang, Xianlian Zhou, Hao Su†, *IEEE Senior Member*

***Abstract*—Despite significant progress in sim-to-real robotic control, there remains a lack of simulation platforms for learning exoskeleton control policies based on human musculoskeletal systems. Existing research in musculoskeletal modeling and control has primarily focused on replicating human motion, with limited exploration of how the human musculoskeletal system adapts to assistive forces from exoskeletons, or how sim-to-real discrepancies can be addressed for effective policy deployment. To bridge this gap, we introduce ExoGym, a simulation platform that integrates a full-body human musculoskeletal model with redundant musculature and a lower limb exoskeleton, both controlled by reinforcement learning (RL)-based policies to replicate reference motion. The exoskeleton control policy is trained through interaction with this adaptively controlled human model, enabling co-adaptive behavior. To support learning under partial observability, we formulate a long short-term memory (LSTM)-based feature representation tailored for exoskeleton control. Additionally, we apply feature alignment and domain randomization to bridge the sim-to-real gap and enable zero-shot sim-to-real transfer. Using ExoGym, we train a hip exoskeleton control policy and successfully deploy it on a physical device. Simulation results show that the RL-based exoskeleton control policy reduces biological joint torque by approximately 15.4% and muscle activation by 15.9% compared to unassisted walking. Real-world experiments across three subjects and three walking speeds validate the robustness and generalization of the trained control policy.**

***Index Terms*—Sim-to-real Learning, Exoskeleton Control, Musculoskeletal modeling, Reinforcement learning**

I. Introduction

Human motion is influenced by various anatomical factors, including bone geometry, muscle conditions, fatigue, habits, and even emotions [1]. Developing exoskeleton controllers presents significant challenges due to the complexity and variability inherent in human biomechanics. Learning-based approaches, such as reinforcement learning (RL) [2] and supervise learning [3], have shown promise in enabling flexible and adaptive control strategies. However, these methods for exoskeleton control typically require extensive human experiments or data collection from human users, which are time consuming and costly. Alternatively, neuromechanical models that simulate the neuro-musculo-skeletal dynamics of the human body provide a viable means of generating human behaviors [4]. Learning exoskeleton control policies in simulation provides a promising solution to address the challenges associated with time-intensive and high-cost physical training experiments [5]. *However, two challenges have limited the wide application of sim-to-real learning for exoskeleton control.*

Most existing simulation platforms focused on modeling and control of the human musculoskeletal system to replicate complex human behaviors [1], [7], [8]. However, most of them did not simulate the physical human-robot interactions and obtain biomechanical adaptation, limiting their ability to support the development of exoskeleton controllers. Accurately modeling the human musculoskeletal system relies on a rigid body skeleton system and a muscle-tendon actuation system [9]. OpenSim is a widely used physics-based simulation platform for human biomechanical modeling [4]. Recently, physics engines, such as MuJoCo [8] and DART [1], were explored to improve the computational efficiency of musculoskeletal modeling, which facilitates their applications in learning-based methods. Several full-body musculoskeletal models were developed and proven to replicate human locomotion behaviors [1], [7]. However, controlling a human musculoskeletal system remains challenging due to the complexity of muscle synergies, making it a high-dimensional and over-actuated problem [1]. To address this challenge, RL-based methods have shown great promise in directly mapping human states to muscle activations of the human musculoskeletal system [7]. Human-like behaviors were successfully replicated by designing reward functions for RL-based methods [10], [11]. The sample efficiency of RL-based methods was improved by reducing the dimensionality of the action space rather than directly generating muscle activations [7], [12]. A two-level RL-based method was proposed to obtain the desired skeletal actuation torque and then use a neural network to derive muscle activations from the actuation torque [1]. *Nevertheless, in addition to producing human motion, developing exoskeleton controllers also requires the biomechanical adaptation of the human musculoskeletal system to external forces, which remains unexplored in existing simulation platforms.*

Sim-to-real discrepancies significantly hinder the effectiveness of deploying exoskeleton control policy on physical exoskeletons and human users. Learning exoskeleton control policies using RL depends on the formulation of state space, action space, and reward function [13]. Several RL-based controllers were learned from human-exoskeleton interactions. An RL-based controller was developed to minimize the interaction force between the human and the exoskeleton [7].

Z. Hou, J. Zhou, I. Lopez-Sanchez, J. Huang, and H. Su are with Lab of Biomechatronics and Intelligent Robotics, Tandon School of Engineering, New York University, USA.

X. Zhou is with the Department of Biomedical Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA.

Corresponding author: hao.su@nyu.edu

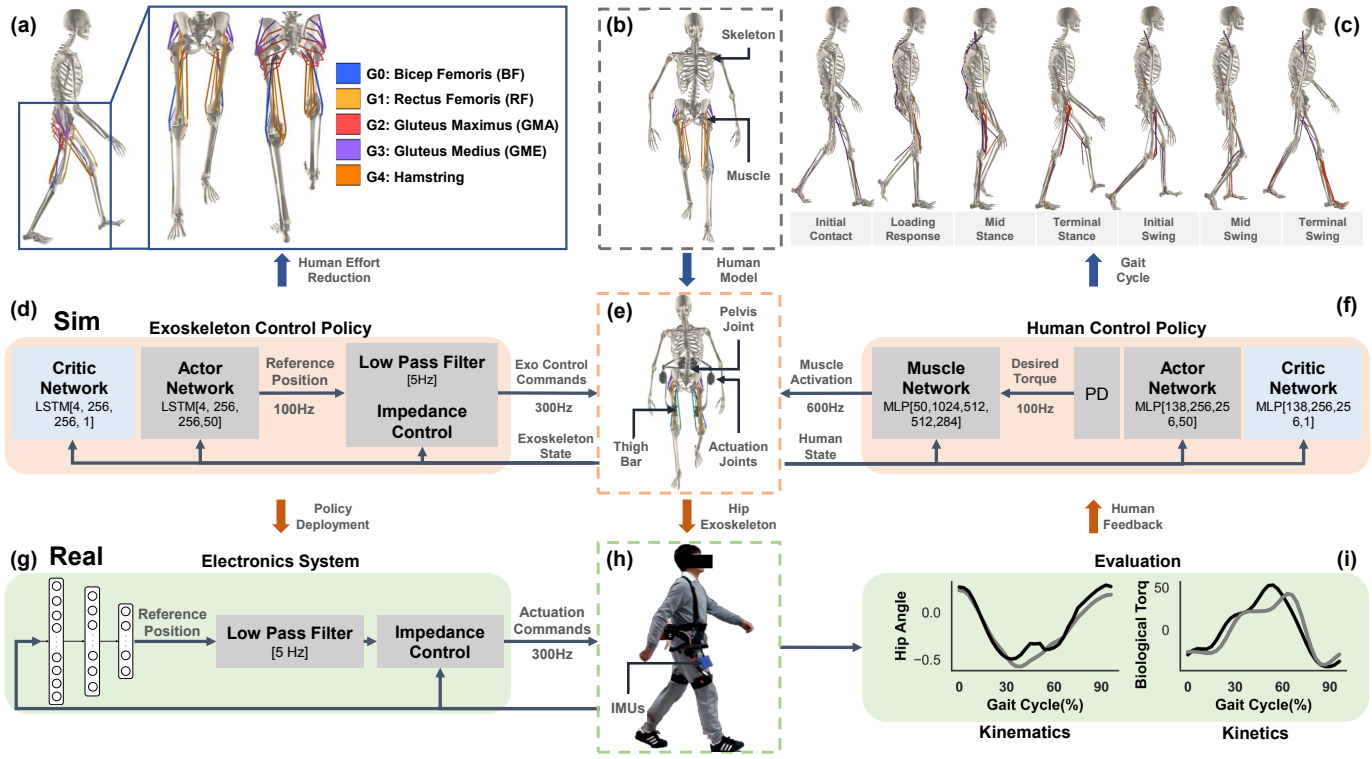


Fig. 1. Our ExoGym consists of the modelling of fully-body musculoskeletal system ((a)-(c)) and two control policies learning through the physical interaction with it ((d)-(f)). A RL-based human control policy can be trained to both replicate the reference locomotion and adapt to external forces. Different from the coupled training scheme in [5], the learned human control policy with the the biomechanical adaptation capability can be reused to develop or learn the exoskeleton control policy separately. Furthermore, in contrast to [5], [6], an exoskeleton control policy is learned using LSTM-based feature representation to facilitate the policy deployment on physical exoskeleton (d). Finally, the actor network with learned parameters can be deployed on the customized electronic system and QDD-driven hip exoskeleton.

TABLE I
COMPARISON WITH STATE-OF-THE-ART SIM-TO-REAL PLATFORM

Methods	Human		Exoskeleton	
	Modeling	Control	Control	Deployment
[4], [8]	√	–	–	–
[1], [12], [10], [9]	√	√	–	–
[13], [6], [7]	√	√	√	–
[5], Our ExoGym	√	√	√	√

Another RL policy for lower-limb exoskeleton control was learned to reduce the biological torque by providing assistance to hip joints [6]. However, the effectiveness of these RL-based controllers was only demonstrated in simulation [7], [13], [6]. The deployment of the exoskeleton control policy on physical exoskeletons was not systematically validated. *Our prior work [5] has demonstrated the effectiveness of deploying policy on a physical exoskeleton, however, the coupled training scheme of motion imitation, exoskeleton control, and muscle coordination networks does not support the reuse of learned human control policy. Moreover, how to bridge the sim-to-real gap for effective zero-shot policy deployment has not been thoroughly investigated.*

This article makes two main contributions to the advancement of sim-to-real learning and the deployment of exoskeleton control, compared to the state-of-the-art methods summarized in Table I. First, we present ExoGym, a simulation platform that simulates the dynamics of a human musculoskeletal system by incorporating a full-body musculoskeletal model, an exoskeleton model, and a human-exoskeleton interaction model. We train a human control policy using RL to replicate reference locomotion and achieve biomechanical adaptation for the development of exoskeleton control policies. verify, evaluate, and validate different controllers. Second, we learn an exoskeleton control policy using ExoGym and successfully deploy it on the physical exoskeleton. To facilitate policy learning and deployment using partial observations, we propose a long short-term memory (LSTM) based feature representation. We bridged the sim-to-real gap for zero-shot policy transfer by combining feature alignment and domain randomization.

II. OVERVIEW OF OUR EXOGYM

The overall structure of our proposed ExoGym is illustrated in Fig. 1. It comprises modeling of the human-exoskeleton interaction system and learning two control policies to drive the human system and the exoskeleton. The human-exoskeleton interaction system consists of a full-body human musculoskeletal model, an exoskeleton model, and human-exoskeleton interaction model. The dynamics of the human-exoskeleton interaction system are simulated on DART [1]. On top of [1], [5], a fully-body human musculoskeletal model, consisting of a skeleton with $n_h = 50$ degrees of freedom (DoFs) and $n_m = 284$ muscle tendon units (Fig. 1(a) and Fig. 1(b)), was developed for lower-limb locomotion. The height of the human musculoskeletal model is 170cm, and the weight is 72kg. A human control policy will be trained using RL to output muscle activation for replicating the given reference motion and human, such as walking on the level ground (see Fig. 1(c)). The human musculoskeletal model with learned human control policy can be utilized to develop the controller for different lower-limb exoskeletons (Fig. 1(f)).

A hip exoskeleton for flexion and extension with $n_r = 2$ DoFs is used as an example to demonstrate the workflow of our proposed ExoGym (Fig. 1(e)). $n_r \ll n_h$ is the dimension of actuation joints for the exoskeleton. The constraints between the human musculoskeletal model and the exoskeleton model are depicted in Fig. 1(e). The base of the exoskeleton model is fixed to the pelvis joint of the human musculoskeletal model, allowing no relative movement. The exoskeleton joints are precisely aligned with the human hip joints, and assistive forces are applied to the lower limbs via a thigh bar. The ball joint constraints were added between the exoskeleton model and the human musculoskeletal model.

Five muscle groups that may be affected by the hip exoskeleton are illustrated in Fig. 1(a). The exoskeleton control policy can be learned by RL to reduce human muscle effort based on the feedback from the human musculoskeletal model (Fig. 1(d)). Different from the human control policy learning, the learned exoskeleton control policy needs to be deployed on the physical exoskeleton (Fig. 1(h)). Furthermore, discrepancies between the simulated and physical dynamics will affect the effectiveness of zero-shot policy transfer. In order to reduce the influence of the transmission system for sim-to-real transfer, we utilize QDD actuators to drive the exoskeleton joints directly. Furthermore, an electronic system was developed to reproduce the learned exoskeleton control policy for our hip exoskeleton. The sim-to-real gap was addressed from two aspects to demonstrate how to use our ExoGym (Section V).

III. Human Musculoskeletal System Model and Control

A. Human Musculoskeletal System Modeling

The human musculoskeletal system consists of a rigid humanoid skeleton model coupled with computational muscle-tendon models.

1) Skeleton Modeling: The dynamics of the human musculoskeletal model are formulated as follows:

$$\boldsymbol{\mathcal{M}}(\boldsymbol{q}^h)\ddot{\boldsymbol{q}^h} + \boldsymbol{\mathcal{C}}(\boldsymbol{q}^h, \dot{\boldsymbol{q}^h})\dot{\boldsymbol{q}^h} = \boldsymbol{J}_m^T \boldsymbol{F}_m(\boldsymbol{a}) + \boldsymbol{J}_c^T \boldsymbol{F}_c + \boldsymbol{\tau}_{ext} \quad (1)$$

where $\boldsymbol{q}^h \in \mathbb{R}^{n_h}$ and $\dot{\boldsymbol{q}^h} \in \mathbb{R}^{n_h}$ are human joint angles and velocities, respectively. $\boldsymbol{\mathcal{M}}(\cdot)$ is the mass matrix, $\boldsymbol{\mathcal{C}}(\cdot,\cdot)$ is the Coriolis and gravitational forces. $\boldsymbol{F}_m \in \mathbb{R}^{n_h}$ and $\boldsymbol{F}_c \in \mathbb{R}^{n_h}$ are the muscle and constraint forces. $\boldsymbol{J}_m$ and $\boldsymbol{J}_c$ are Jacobian matrices, which map muscle forces and constraint forces to the human joint space. $\boldsymbol{a} \in \mathbb{R}^{n_m} \in [0,1]^{n_m}$ is muscle activation. $\boldsymbol{\tau}_{ext} \in \mathbb{R}^{n_h}$ is the external torque, including the applied force by the exoskeleton and the interaction force with the environment.

2) Muscle Modeling: According to the Hill-type model [1], the $i-$th muscle-tendon unit is formulated to derive the muscle force $F_m(a_i; l_i, v_i)$ from activation as $F_{max} \cdot [a_i \cdot f_l(l_i) \cdot f_v(v_i) + F_p(l_i)]$. l_i and v_i represent the normalized muscle length and the rate of muscle changes, respectively. $F_p(l_i)$ is the passive force developed by the muscle. F_{max} is maximum isometric muscle force. Therefore, the muscle force can be derived as follows:

$$\boldsymbol{F}_m(\boldsymbol{a}) = \boldsymbol{F}_m^{max} \cdot [\boldsymbol{a} \cdot \boldsymbol{F}_l + \boldsymbol{F}_m^p] = \boldsymbol{A} \cdot \boldsymbol{a} + \boldsymbol{b} \quad (2)$$

where $\boldsymbol{F}_l$ is the function affected by muscle length and the rates of muscle changes.

B. Human Musculoskeletal System Control

A parameterized RL policy $\boldsymbol{\pi}_{\boldsymbol{\phi}}(\boldsymbol{a}|\boldsymbol{s}^h; \boldsymbol{\phi})$ was defined for musculoskeletal system control to derive muscle activation $\boldsymbol{a}$ according to the human state $\boldsymbol{s}^h$. To mitigate the challenges of learning high-dimensional action, the policy is decoupled as follows:

$$\boldsymbol{\pi}_{\boldsymbol{\phi}}(\boldsymbol{a}|\boldsymbol{s}^h; \boldsymbol{\phi}) = \boldsymbol{\pi}_m(\boldsymbol{a}|\boldsymbol{u}^h, \boldsymbol{s}^h; \boldsymbol{\phi}_m) \circ \boldsymbol{\pi}_s(\boldsymbol{u}^h|\boldsymbol{s}^h; \boldsymbol{\phi}_h) \quad (3)$$

where $\boldsymbol{\pi}_s(\boldsymbol{u}^h|\boldsymbol{s}^h; \boldsymbol{\phi}_h)$ is a skeleton-related policy to output a latent action $\boldsymbol{u}^h$ with relatively lower action dimensions that original muscle activations. $\boldsymbol{\pi}_m(\boldsymbol{a}|\boldsymbol{u}^h, \boldsymbol{s}^h; \boldsymbol{\phi}_m)$ is the muscle-related policy that maps the latent action to the muscle activations. Similarly to [1], the latent action is the actuation torque $\boldsymbol{\tau_q}$ of each human joint. A neural network is developed for $\boldsymbol{\pi}_m(\cdot|\cdot)$ to derive muscle activations $\boldsymbol{a}$ from the output joint actuation torque by solving a linear regression objective.

The human musculoskeletal system control was simplified to learn the policy $\boldsymbol{\pi}_s(\boldsymbol{u}^h|\boldsymbol{s}^h; \boldsymbol{\phi}_h)$ using continuous control RL methods. The episodic control problem can be formulated as a *Markov Decision Process* (MDP) $< \mathcal{S}_h, \mathcal{U}_h, \mathcal{R}_h, \mathcal{T}_h, \mathcal{P}_h^0, \gamma_h >$. $\boldsymbol{s}^h \in \mathcal{S}_h$ is the state of the human musculoskeletal model. $\boldsymbol{u}^h \in \mathcal{U}_h$ is the action taken by the RL policy. $\mathcal{R}_h : \mathcal{S}_h \times \mathcal{U}_h \rightarrow r(\boldsymbol{s}^h, \boldsymbol{a}^h) \in \mathbb{R}$ is the reward function for each time step. $\mathcal{T}_h : \mathcal{S}_h \times \mathcal{A}_h \rightarrow \mathcal{S}_h$ is the environment transition function. $\mathcal{P}_h^0$ is an initial state distribution and $\gamma_h \in (0,1)$ is the discount factor. The parameter of the RL policy is updated by maximizing the expected cumulative reward, as follows:

$$\boldsymbol{\phi}_h^* = \underset{\boldsymbol{\phi}_h}{\operatorname{argmax}}\, \mathbb{E}_{(\boldsymbol{s}_t^h, \boldsymbol{u}_t^h)}[\sum\nolimits_{t=0}^{\infty} \gamma_h r(\boldsymbol{s}_t^h, \boldsymbol{u}_t^h)] \quad (4)$$

where $\boldsymbol{s}_0^h \sim \mathcal{P}_h^0$, and action $\boldsymbol{u}_t^h \sim \boldsymbol{\pi}_{\boldsymbol{\phi}}(\boldsymbol{u}_t^h|\boldsymbol{s}_t^h; \boldsymbol{\phi}_h)$ is sampled at time step $\boldsymbol{t}^h$ and the muscle activation $\boldsymbol{a}$ is derived from $\boldsymbol{\pi}_m(\boldsymbol{a}|\boldsymbol{u}^h, \boldsymbol{s}^h; \boldsymbol{\phi}_m)$ to drive the human musculoskeletal model. The next state is sampled by $\boldsymbol{s}_{t+1}^h \sim \mathcal{T}_h(\boldsymbol{u}_t^h|\boldsymbol{s}_t^h; \boldsymbol{\phi}_h)$ and the reward r_t is calculated. Human motion trajectory during one episode can be collected as $\boldsymbol{\Omega}^h = \{(\boldsymbol{s}_0^h, \boldsymbol{u}_0^h, r_0^h, \boldsymbol{a}_0), \cdots, (\boldsymbol{s}_T^h, \boldsymbol{u}_T^h, r_T^h, \boldsymbol{a}_T)\}$.

The objective of human musculoskeletal system control is to adapt the behaviors according to the assistance force provided by the exoskeleton. Different from previous studies [1] [14], the human state $\boldsymbol{s}^h$ is defined as $\boldsymbol{s}^h = (\boldsymbol{s}_q^h, \boldsymbol{\tau}_{exo})$. $\boldsymbol{s}_q^h = (\boldsymbol{p}, \boldsymbol{v}, \kappa)$ is kinematics data including position $\boldsymbol{p}$ and velocity $\boldsymbol{v}$ of each bone node. $\kappa \in [0,1]$ is a phase variable to align with the reference motion. $\boldsymbol{\tau}_{exo}$ is the external force provided by the exoskeleton. $\boldsymbol{u}_t^h$ is the reference joint angle, and a PD controller is employed to obtain the actuation torque $\boldsymbol{\tau}_q^h$, as follows:

$$\boldsymbol{\tau_q}^h = \boldsymbol{\mathcal{K}}_p^h(\boldsymbol{u}_t^h - \boldsymbol{q}^h) - \boldsymbol{\mathcal{K}}_d^h \dot{\boldsymbol{q}^h} \quad (5)$$

where $\boldsymbol{\mathcal{K}}_p^h \in \mathbb{R}^{n_h \times n_h}$ and $\boldsymbol{\mathcal{K}}_d^h \in \mathbb{R}^{n_h \times n_h}$ are predefined diagonal parameter matrices. r_h is defined to replicate the given reference motion, as follows:

$$r_h = (\alpha_h^q r_q + \alpha_h^v r_v) \cdot r_{ee} + \alpha_h^c r_h^c \quad (6)$$

where r_q, r_v, and r_{ee} are designed to encourage the imitation of the reference joint angle, velocity, and end-effector pose. r_h^c is the desired constraints for the human musculoskeletal model should satisfy, such as encouraging steady walking. α_h^q, α_h^v, and α_h^c are corresponding weighting coefficients.

IV. Learning Exoskeleton Control Policy in Simulation

Any continuous control RL method can be formulated to learn the exoskeleton control policy, similarly to the formulation described in Section III-B. However, the state used for RL-based exoskeleton control policy learning should consist of the human state, which is only partially observable in the physical exoskeleton. Therefore, the episodic exoskeleton control problem is formulated as *Partial Observable Markov Decision Process* (POMDP) $< \mathcal{S}_e, \mathcal{O}_e, \mathcal{U}_e, \mathcal{R}_e, \mathcal{T}_e, \mathcal{P}_e^0, \gamma_e >$. $\boldsymbol{s}^e \in \mathcal{S}_e$ is the full state of the RL exoskeleton agent, such as the human states. The parameterized exoskeleton control policy is defined as $\boldsymbol{\pi}_\psi(\boldsymbol{u}^e|\boldsymbol{o}^e;\boldsymbol{\psi})$. The observation $\boldsymbol{o}^e \in \mathcal{O}_e$ is measurable and accessible for physical exoskeleton control in the real world. $\boldsymbol{u}^e \in \mathcal{U}_e$ is the output action taken by the RL policy for exoskeleton control. $\mathcal{R}_e : \mathcal{S}_e \times \mathcal{U}_e \rightarrow r_e(\boldsymbol{s}^e, \boldsymbol{u}^e) \in \mathbb{R}$ is the reward function. γ_e is the discount factor. $\mathcal{T}_h : \mathcal{S}_h \times \mathcal{U}_h \rightarrow \mathcal{S}_h$ is the environment transition function. $\mathcal{P}_e^0$ is an initial state distribution. The exoskeleton control trajectory during one episode can be collected as $\boldsymbol{\Omega}^e = \{(\boldsymbol{o}_0^e, \boldsymbol{u}_0^e, r_0^e), \cdots, (\boldsymbol{o}_T^e, \boldsymbol{u}_T^e, r_T^e)\}$.

LSTM is employed to utilize the history information for dealing with the POMDP formulation. Several lower-level controllers have demonstrated superior sample efficiency performance in learning interactive control using RL [15]. Instead of obtaining the actuation torque, the impedance controller is developed for exoskeleton control. $\boldsymbol{u}^e$ is the output reference joint angle sampled from $\boldsymbol{\pi}_{\boldsymbol{\psi}}(\boldsymbol{u}^e|\boldsymbol{o}^e;\boldsymbol{\psi})$. The actuator control command $\boldsymbol{\tau}_{\boldsymbol{q}}^e$ is calculated as follows:

$$\boldsymbol{\tau}_{\boldsymbol{q}}^e = \boldsymbol{\mathcal{K}}_p^e(\boldsymbol{u}^e - \boldsymbol{q}^e) - \boldsymbol{\mathcal{K}}_d^e \dot{\boldsymbol{q}^e} \tag{7}$$

where $\boldsymbol{\mathcal{K}}_p^e \in \mathbb{R}^{n_r \times n_r}$ and $\boldsymbol{\mathcal{K}}_d^e \in \mathbb{R}^{n_r \times n_r}$ are predefined diagonal parameter matrices. $\boldsymbol{q}^e \in \mathbb{R}^{n_r}$ and $\dot{\boldsymbol{q}^e} \in \mathbb{R}^{n_r}$ denote the hip joint angles and velocities, respectively. Additionally, the assistive torque applied to the physical exoskeleton is clipped by the maximal peak torque $[-\boldsymbol{\tau}_{max}, \boldsymbol{\tau}_{max}]$ of the actuators.

For hip exoskeleton control, $\boldsymbol{o}^e = (\boldsymbol{q}^e, \dot{\boldsymbol{q}}^e, \overline{\boldsymbol{\tau}_{\boldsymbol{q}}^e})$ consists of the kinematic data measured from wearable sensors, such as inertial measurement units (IMUs) illustrated in Fig. 1. $\overline{\boldsymbol{\tau}_{\boldsymbol{q}}^e}$ is the normalized previous actuation torque. r_e is defined to reduce human effort as follows:

$$r_e = \alpha_e^h r_h + \alpha_e^m r_m + \alpha_e^c r_e^c \tag{8}$$

where r_h is the component to encourage human walking performance, defined in (6). r_m is the component to encourage the reduction of biological torque or muscle activation of the human musculoskeletal system. r_e^c is defined to smooth the actuator control command. $\alpha_e^h, \alpha_e^m, \alpha_e^c$ are coefficients to balance the weights of each component.

Algorithm 1 Training Scheme

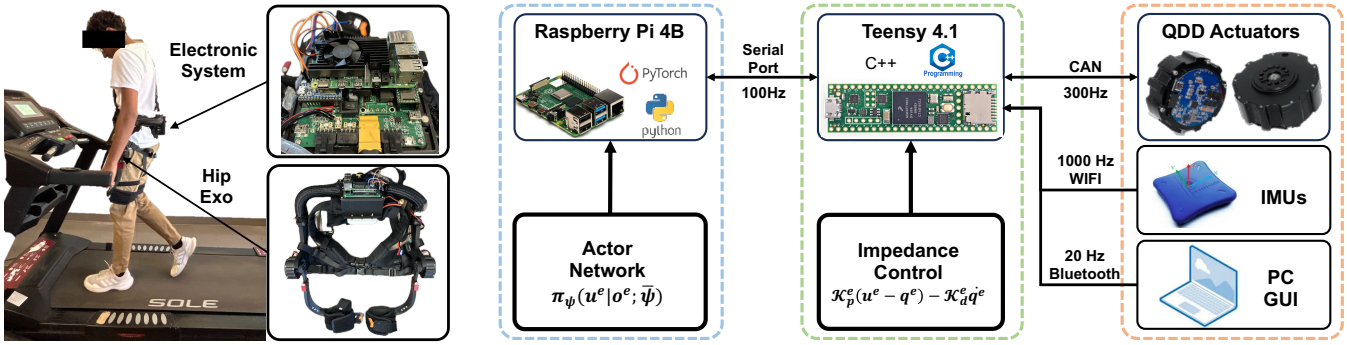
Learning Human Control Policy by PPO
Input: initial human control policy parameter $\boldsymbol{\phi}_h^0$
Input: initial muscle network parameter $\boldsymbol{\phi}_m^0$
for Each Iteration $i \in 1, 2, \cdots, I_{human}$ **do**
  $\boldsymbol{\phi}_m^i \leftarrow \boldsymbol{\phi}_m^{i-1}$, $\boldsymbol{\phi}_h^i \leftarrow \boldsymbol{\phi}_h^{i-1}$, $\mathcal{D}_i^h \leftarrow \varnothing$
  for Each human agent $j \in 1, 2, \cdots N_{env}$ **do**
    Collect trajectories $\{\boldsymbol{\Omega}_j^h\}$ from policy $\boldsymbol{\pi}(\cdot|\cdot;\boldsymbol{\phi}^i)$
    Compute estimated advantages $\{\widehat{\boldsymbol{A}}_j^h\}$
    $\mathcal{D}_i^h \leftarrow \mathcal{D}_i^h + \{\widehat{\boldsymbol{A}}_j^h\}$
  end for
  Update parameters $\boldsymbol{\phi}_h^i$ for L_h epochs using minibatch data with size $\boldsymbol{B}_{human}$ sampled from $\mathcal{D}_i^h$
  for Each muscle network update step **do**
    Update parameter $\boldsymbol{\phi}_m^i$ via gradient descent
  end for
end for
Learning Exoskeleton Control Policy by PPO
Input: optimal human control policy parameter $\boldsymbol{\phi}_h^*$
Input: optimal muscle network parameter $\boldsymbol{\phi}_m^*$
Input: exoskeleton policy network initial parameter $\boldsymbol{\psi}^0$
for Each Iteration $i \in 1, 2, \cdots, I_{exo}$ **do**
  $\boldsymbol{\psi}^i \leftarrow \boldsymbol{\psi}^{i-1}$, $\mathcal{D}_i^e \leftarrow \varnothing$
  for Each exoskeleton agent $j \in 1, 2, \cdots N_{env}$ **do**
    Collect trajectories $\{\boldsymbol{\Omega}_j^e\}$ from policy $\boldsymbol{\pi}(\cdot|\cdot;\boldsymbol{\psi}^i)$
    Compute estimated advantages $\{\widehat{\boldsymbol{A}}_j^e\}$
    $\mathcal{D}_i^e \leftarrow \mathcal{D}_i^e + \{\widehat{\boldsymbol{A}}_j^e\}$
  end for
  Update parameters $\boldsymbol{\psi}^i$ for L_e epochs using minibatch data with size $\boldsymbol{B}_{exo}$ sampled from $\mathcal{D}_i^e$
end for

V. Close Sim-to-real Gap for Exoskeleton Control Policy Deployment

Both the human control policy and the exoskeleton control policy can be learned by any continuous RL methods, such as SAC [16] and PPO [17]. The training procedures of each policy using PPO are illustrated in Algorithm 1, with the network architecture and hyperparameter provided in Appendix VIII-A and VIII-B. Once the exoskeleton control policy satisfies the termination criterion, the actor network with learned parameters $\bar{\boldsymbol{\psi}}$ is deployed on the customized electronic system by closing the sim-to-real gaps.

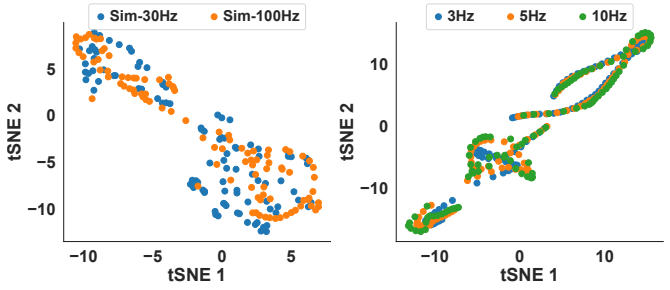
A. Feature Alignment

The discrepancy between the observation for exoskeleton control policy in simulation and the physical exoskeleton may largely affect the effectiveness of zero-shot policy transfer. In simulation, the simulation frequency f_{sim}^{env} of our proposed ExoGym represents the human reaction time, while the control frequency f_{sim}^{ctl} used for policy training can be customized for different reference motions. The feature representation of the observation collected from our proposed ExoGym using different control frequencies is visualized by t-SNE, and the 2D features are illustrated in Fig. 3(a). The control frequency f_{real}^{ctl} of running policies on the physical exoskeleton is determined by



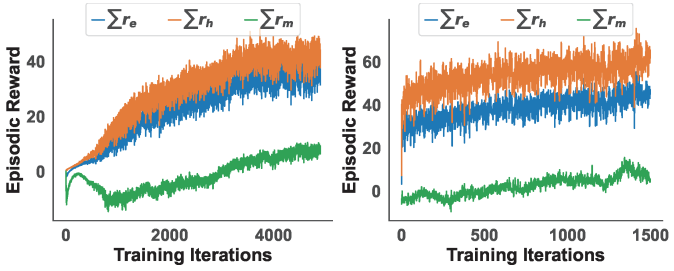
(a) (b)

Fig. 2. Details of deploying learned policy on our customized electronic system and QDD-driven hip exoskeleton. (a) Illustration of a subject wearing the hip exoskeleton while walking on a treadmill. (b) Communication architecture used to deploy the exoskeleton control policy on our customized electronic system.



(a) (b)

Fig. 3. Feature representation based on t-SNE to compare the observations for exoskeleton control policy learning. (a) Observations collected in simulation under the control frequency of 30Hz and 100Hz. (b) Observations collected in simulation using the low-pass filter with three different cutoff frequency: 3Hz, 5Hz, and 10Hz.



(a) (b)

Fig. 4. Episodic exoskeleton reward, human walking reward, human effort reduction reward during policies training in simulation. (a) Exoskeleton control policy and human control policy were trained simultaneously. (b) Exoskeleton control policy and human control policy were trained separately.

the electronic system. The discrepancies illustrated in Fig. 3(a) recommend that the same control frequency should be selected for simulation according to the actual control frequency of the electronic system. In order to improve the stability of policy learning, the uncertainties in collected observations should be filtered using a low-pass filter $\mathcal{F}^o_{LFP}$. Furthermore, another low-pass filter $\mathcal{F}^{\boldsymbol{u}}_{LFP}$ is employed to smooth the actuator control commands $\boldsymbol{\tau}^e_{\boldsymbol{q}}$ or the output reference joint angles from the actor network. The feature representation of the observation processed by different low-pass filters is visualized using t-SNE. The 2D features illustrated in Fig. 3(b) suggest that identical low-pass filters $\mathcal{F}^o_{LFP}$ and $\mathcal{F}^{\boldsymbol{u}}_{LFP}$ should be chosen for both simulation and physical exoskeleton control.

B. Domain Randomization

Human biomechanical and kinematic responses are largely affected by the mass, inertia, friction coefficient, and isometric force parameters of the human musculoskeletal model (Section III-A). Domain randomization is applied for the human musculoskeletal model to facilitate the human control policy and to enable the exoskeleton control policy to generalize across different human users [18]. As illustrated in Fig. 1(b) and Fig. 1(c), QDD actuators are employed for our exoskeleton based on the assumption that QDD actuators are well-aligned with the hip joints of human users. With the direct actuation configuration, the dynamics of both actuators and exoskeleton can be neglected in the impedance controller, as derived in (7). Therefore, the domain randomization is not applied to the exoskeleton model.

VI. Validation

A. Setups for Simulation and Experiment

1) Setups for Policies Training in Simulation: Our proposed ExoGym is performed on a PC (LEGION) equipped with a GEFORCE RTX 4080 GPU. Human musculoskeletal model is conducted at a simulation frequency of f^{env}_{sim} =600 Hz, while the human control policy and exoskeleton control policy operate at a control frequency of $f^{ctl}_{sim} = 100$ Hz. A second-order low-pass Butterworth filter with a cutoff frequency of $f_{cutoff} = 10$Hz is applied to observations and actuator control commands. $N_{env} = 16$ simulation environments are run in parallel to improve the efficiency of data collection. Reference motions for human walking at various speeds on level ground, collected from an open-source dataset [19], are employed to train the human control policy.

2) Setups of Customized Electronic System for Policy Deployment: The hip exoskeleton and the electronic system were developed to validate the effectiveness of our proposed ExoGym and the learned exoskeleton control policy using it (see Fig. 2(a)). QDD actuators, Sig Motors (6010-H) with a gear ratio of 9.67 : 1, are employed to drive the hip exoskeleton. Fig. 2(a) illustrates the customized electronic system used for policy deployment. Fig. 2(b) shows the communication architecture of deploying the policy on our hip exoskeleton. The actor network with the learned parameters $\overline{\psi}$ runs on a Raspberry Pi 4B. At each control step, operating at a control frequency of 100Hz, the input to the actor network is the observation $\boldsymbol{o}^e$, which includes the human kinematics captured using IMUs (LPMS-B2, ALUBI) via WIFI at a sampling

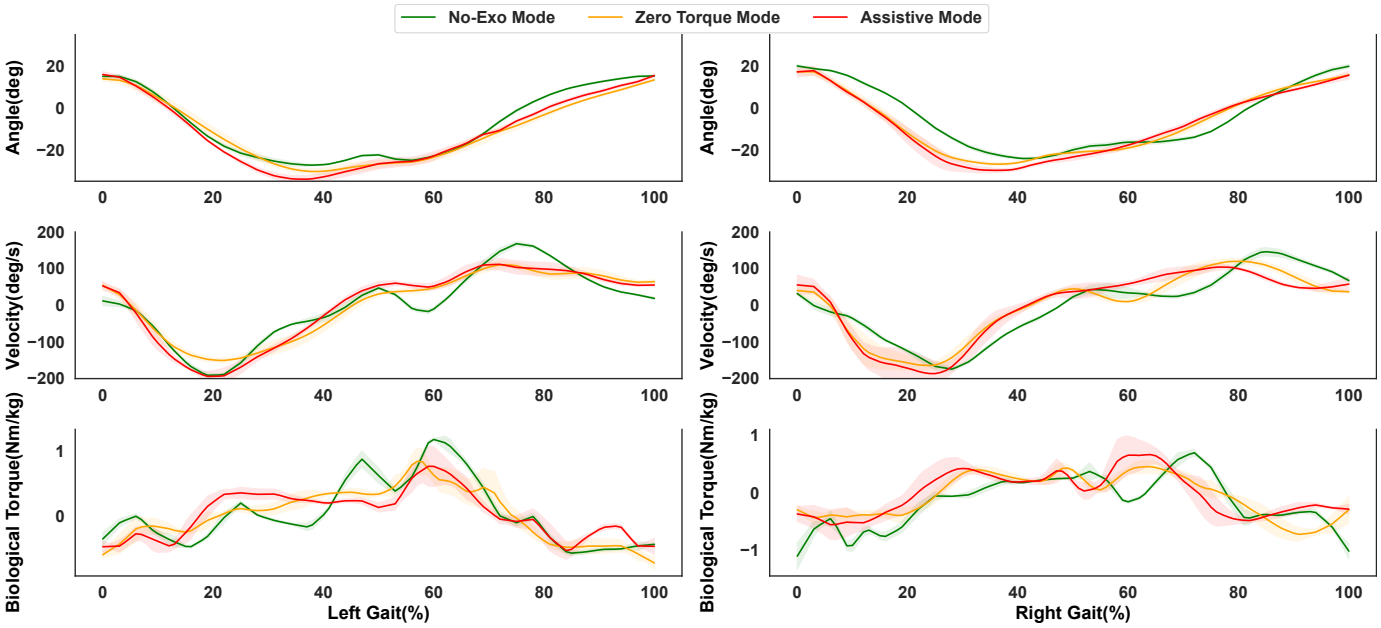


Fig. 5. Performance of hip joint angle, velocity, and biological torque over five gait cycles during the evaluation of human control policy in simulation. The green lines indicate that the learned human control policy under *No-Exo* mode can successfully replicate both kinematic and kinetic performance of human users. The orange lines demonstrate that the human control policy learned under *Assistive* mode can adapt to the changed external forces during the evaluation under *Zero-torque* mode. The red lines demonstrate that the exoskeleton control policy learned in *Assistive* mode can assist hip flexion and reduce the average biological torque by 15.4%, compared to the *No-Exo* mode.

frequency of 1000Hz. The output of the actor network, the reference joint angles $\boldsymbol{u}^e$, is calculated on the Raspberry Pi 4B and then transmitted to the microcontroller Teensy 4.1 via the serial port at 100 Hz. The actuator control commands $\boldsymbol{\tau}_{\boldsymbol{q}}^e$ are calculated by the impedance controller on the Teensy 4.1 and then transmitted to QDD actuators via CAN communication at a frequency of 300 Hz. Additionally, a user interface is displayed on a PC via Bluetooth at a frequency of 20 Hz.

B. Validation in Simulation

1) Protocol of Human Control Policy Learning and Exoskeleton Control Policy Learning in Simulation: The human control policy was trained using PPO, with the hyperparameters and implementation details summarized in Table II. The PD parameters used in (5) are set as $\boldsymbol{\mathcal{K}}_p^h = \mathbf{diag}(300, \cdots, 300)$ and $\boldsymbol{\mathcal{K}}_d^h = \sqrt{2 \cdot \boldsymbol{\mathcal{K}}_p^h}$. After 5000 training iterations, the human control policy is performed to evaluate walking performance and biomechanical adaptation during a $10s$ walking. Furthermore, the adaptation achieved by the human control policy was validated by the human musculoskeletal system under three assistive modes. In the *No-Exo mode*, the human control policy is performed to replicate level-ground walking and to collect muscle activation without wearing the hip exoskeleton. In the *Assistive mode*, an exoskeleton control policy is trained by PPO using the reward function defined in Appendix VIII-B and the hyperparameters provided in Table III. The impedance parameters are set as $\boldsymbol{\mathcal{K}}_p^e = \mathbf{diag}(50, \cdots, 50)$ and $\boldsymbol{\mathcal{K}}_d^e = 0.5 \cdot \sqrt{\boldsymbol{\mathcal{K}}_p^e}$. In the *Zero-Torque mode*, zero actuation torque from the exoskeleton control policy is applied on the human control policy to validate its adaptation capability.

2) Simulation Results of Training Human Control Policy and Exoskeleton Control Policy: The episodic exoskeleton reward $\sum r_e$, human reward $\sum r_h$, and human effort reduction reward $\sum r_m$ during training are collected. Fig. 4(a) shows the increase

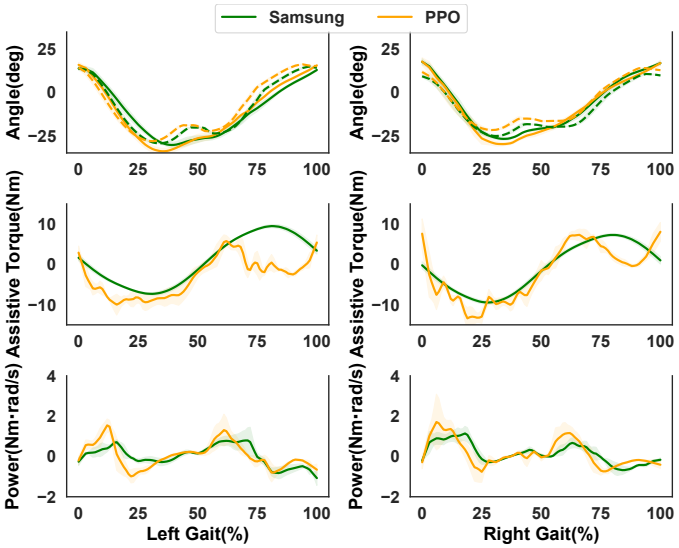
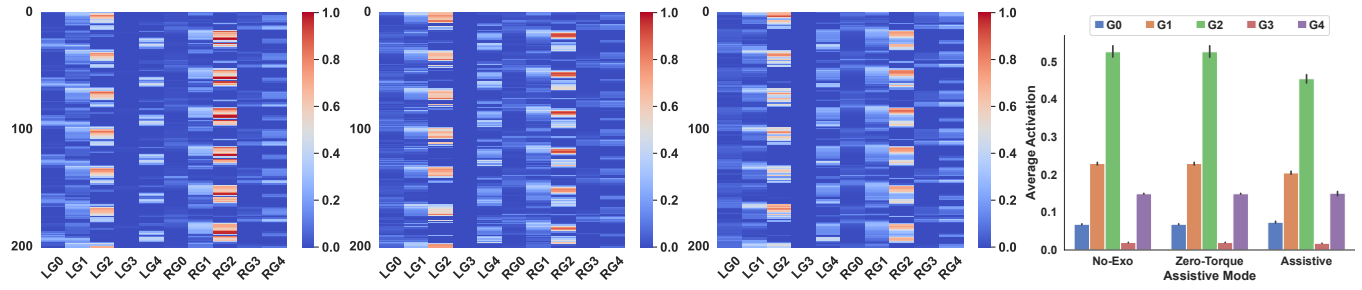


Fig. 6. Performance of hip joint angle, assistive torque, and assistive power over five gait cycles during the evaluation in simulation. Compared to fixed-timing assistance, the PPO-based exoskeleton control policy optimizes the timing of assistance and reduces the assistance for extension to enhance the walking performance.

of episodic rewards through simultaneously training the human control policy and exoskeleton control policy. The human musculoskeletal system does not fall after 1000 iterations of training, and both control policies converge after about 4000 iterations. The human control policy can modify the muscle activation according to the external force from the exoskeleton control policy. Afterwards, the learned human control policy is reused to actuate the human musculoskeletal system and to further optimize the exoskeleton control policy. The increase of episodic human effort reduction reward $\sum r_m$ in Fig. 4(b) demonstrates that exoskeleton control policy can be further optimized to reduce biological torque or muscle activation.

The hip joint angles, velocities, and biological torques over



(a) (b) (c) (d)

Fig. 7. Performance of muscle activation over 200 control steps during the evaluation in simulation. LG0, LG1, LG2, LG3, and LG4 represent five muscle groups of left limb, and RG0, RG1, RG2, RG3, and RG4 represent five muscle groups muscle of right limb, as shown in Fig. 1(a). (a) Heatmap of the muscle activation under *No-Exo* mode. (b) Heatmap of the muscle activation under *Zero-Torque* mode. (c) Heatmap of the muscle activation under *Assistive* mode. (d) Comparison of average muscle activation over five gait cycles under three assistive modes.

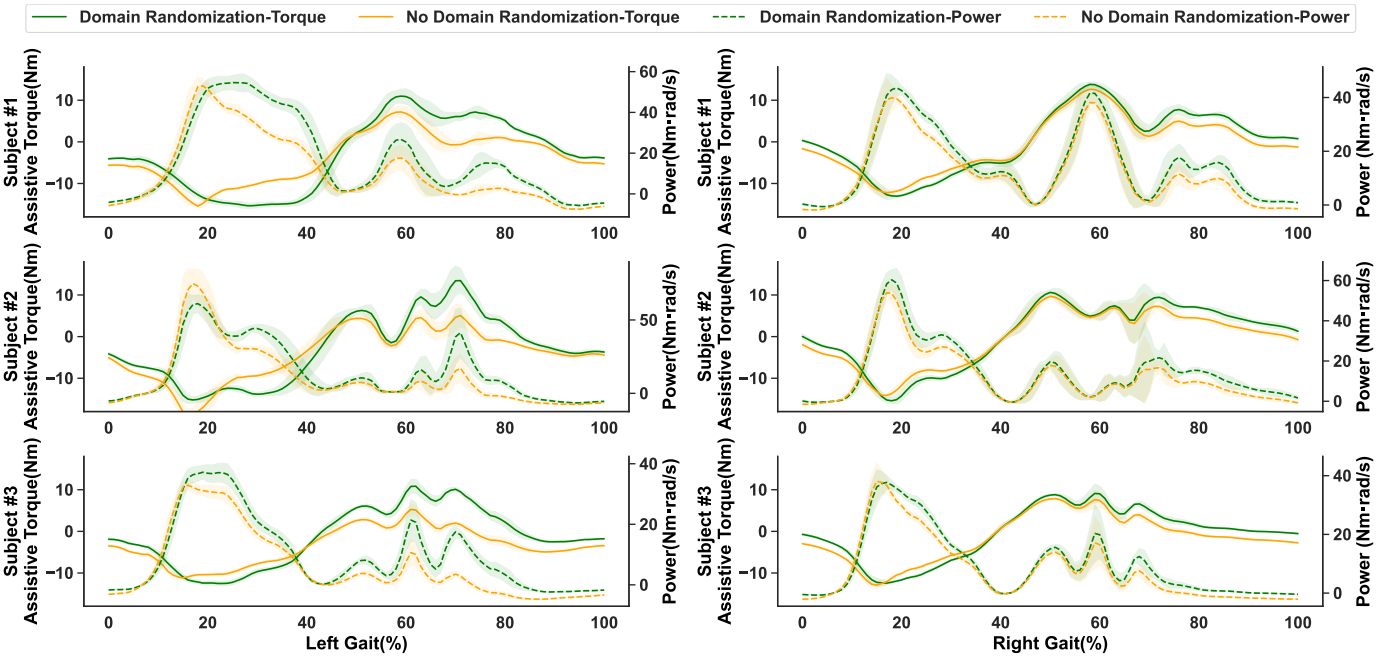


Fig. 8. Experimental results of deploying two exoskeleton control policies—one trained with domain randomization and one without—on three subjects walking at a speed of 1.25m/s. In the domain randomization setting, the parameters of the human musculoskeletal model were randomly scaled within the range [0.8, 1.2]. The policy trained with domain randomization produced greater positive power compared to the policy trained without it. The assistive torque and power demonstrate that domain randomization enhances the generalization capability across different subjects.

five gait cycles, visualized in Fig. 5, demonstrate that human control policy can replicate the reference kinematic and kinetic trajectories from the human dataset. In contrast to *No-Exo* mode and *Zero-Torque* mode, the exoskeleton control policy under *Assistive* mode improves walking performance, resulting in a larger maximal flexion angle and a smoother joint velocity. The differences between the *Zero-Torque* mode and *Assistive* mode highlight the capability of the PPO-based human control policy to adapt to external forces. Moreover, the biological torque of hip joints under *Assistive* mode has 15.4% compared to *No-Exo* mode (see Fig. 5(c)).

The comparison of RL-based exoskeleton control policy and commonly used Samsung controller [20] is plotted in Fig. 6. The assistive timing of the Samsung controller relies on a predefined fixed delay time. The output reference joint angle from PPO, shown in Fig. 6(b), indicates that the timing of assistance for flexion and extension was optimized. The positive power, shown in Fig. 6(c), demonstrates that the exoskeleton control policy learned by PPO reduced the assistance for extension to enhance the walking performance. In addition, five groups of muscle activation affected by the assistance of hip exoskeleton (see Fig. 1(d)) are collected to validate the reduction in human effort. Heatmap of muscle activation of six gait cycles under three assistive modes is plotted in Fig. 7(a) to Fig. 7(c), respectively. Fig. 7(d) shows that the average muscle activation under *Assistive* mode over the latest five gait cycles has reduced 15.9%, compared to *No-Exo* mode. Furthermore, the results in Fig. 7 indicate that Rectus Femoris and Gluteus Maximus are mainly assisted by the hip exoskeleton.

C. Validation on Physical Exoskeleton

1) Protocol of Developing Exoskeleton Control Policy on Physical Exoskeleton: One female and two male able-bodied subjects (age: 27 ± 0.6years; height: 170 ± 12cm; weight: 68 ± 16kg; mean$\pm$standard) from our laboratory were recruited to validate the effectiveness and generalization capability of the zero-shot transferred exoskeleton control policy. Each subject walked at a fixed speed on the treadmill for five minutes. First, to evaluate generalization across different human users, the exoskeleton control policy trained on the human musculoskeletal model walking at a fixed speed of $1.25m/s$

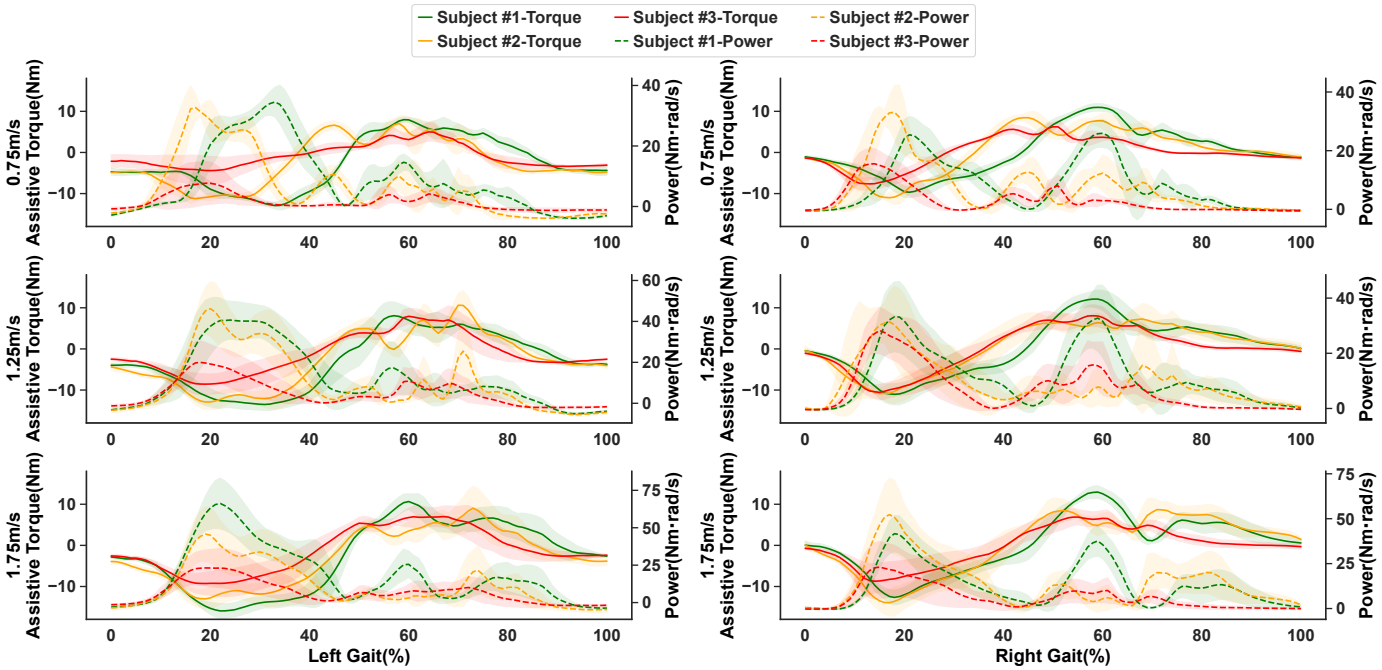
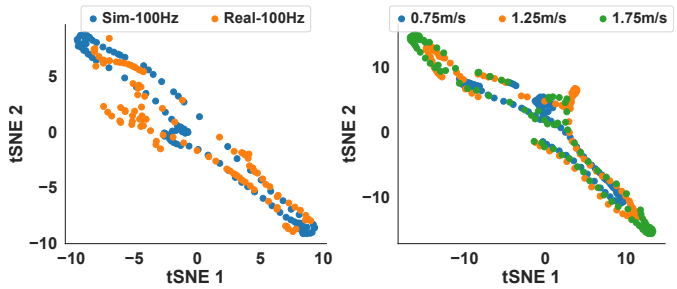


Fig. 9. Experimental results of deploying learned exoskeleton control policy on three subjects walking at speeds of $0.75m/s$, $1.25m/s$, and $1.75m/s$. The solid lines demonstrate that the exoskeleton control policy can adapt the assistive torque for each subject and each walking speed. The assistive power visualized by dashed lines demonstrate that the effectiveness and generalization capability of zero-shot transferred exoskeleton control policy.



(a) (b)

Fig. 10. Feature representation based on t-SNE to compare the observations for exoskeleton control policy development. (a) Observations collected from simulation and physical exoskeleton at 100Hz. (b) Observations collected from human user walking at three different speeds: 0.75m/s, 1.25m/s, and 1.75m/s.

with and without domain randomization was deployed on all three subjects walking at the same speed ($1.25m/s$) as in the simulation. Second, to evaluate generalization across various walking speeds, the exoskeleton control policy trained using the human musculoskeletal model walking at the speed of $1.25m/s$ was developed on three subjects walking at three different speeds: $0.75m/s$, $1.25m/s$, and $1.75m/s$.

2) Experimental Results of Deploying Exoskeleton Control Policy for Treadmill Walking: The assistive torque and power over five gait cycles, presented in Fig. 8, demonstrate that the learned exoskeleton control policy in simulation can be successfully deployed to provide effective assistance to three subjects with different heights, weights, and muscle properties. The use of LSTM-based feature representation in exoskeleton control policy learning helps reduce the dependence on explicit human state information. The discrepancies between the observations collected from our proposed ExoGym and physical exoskeleton at the same control frequency of 100Hz are visualized using t-SNE (see Fig. 10(a)). The small discrepancies benefit from the observation definition and feature alignment. The assistive torque and power indicate that the exoskeleton control policies trained using domain randomization have better adaptation capability than the policy trained without using domain randomization. The positive power shown in Fig. 9 indicates that the learned exoskeleton control policy can provide effective assistance to three different subjects across three walking speeds. Observations collected from a human user of walking at three different speeds are visualized using t-SNE (see Fig. 10(b)). Therefore, the generalization capability of the learned exoskeleton control policy across unseen walking speeds is attributed to the use of LSTM-based feature representation for processing observations and its integration with the impedance controller. The actuator control commands, calculated from (7), depend on both the joint velocity and modification from the RL policy.

VII. Discussion and Conclusion

The output action from the RL policy follows a Gaussian distribution and is clipped based on actuator limits to ensure the safety of zero-shot policy deployment [12]. Normalization of the observation and early termination were employed to accelerate convergence. While this study does not focus on achieving higher human walking performance or improving sample efficiency, future work may explore reward weight tuning and other methods to enhance the policy effectiveness [12]. Unlike previous works [5], [6], this study demonstrates that the human musculoskeletal model, driven by an RL-based human control policy, can adapt to the assistance provided by a hip exoskeleton. We also demonstrated that the proposed ExoGym and the learned human control policy can be effectively reused to develop and evaluate exoskeleton control policies.

The effectiveness of our proposed ExoGym, integrating a human musculoskeletal model and RL-based human control policy, was demonstrated for hip exoskeleton control policy learning. The generalization capability of the learned exoskeleton controller was validated through zero-shot policy transfer

across three able-bodied subjects at three walking speeds. In future work, ExoGym can be extended to develop control policies for exoskeletons targeting other joints, such as knee and ankle joints. Additionally, while the exoskeleton model used in our simulation was identical to the physical exoskeleton, future work will explore cross-device policy transfer to assess robustness across different devices.

VIII. Appendix

A. Human Control Policy Learning

The reward of tracking human joint angle $\overline{\boldsymbol{q}_j^h}$, joint velocity $\overline{\dot{\boldsymbol{q}}_j^h}$, and end-effector position $\overline{\boldsymbol{x}_j^h}$ are defined as follows:

$$\begin{aligned} r_{\boldsymbol{q}} &= \exp(-\beta_{\boldsymbol{q}} \sum\nolimits_j ||\overline{\boldsymbol{q}_j^h} - \boldsymbol{q}_j^h||^2) \\ r_{\boldsymbol{v}} &= \exp(-\beta_{\boldsymbol{v}} \sum\nolimits_j ||\overline{\dot{\boldsymbol{q}}_j^h} - \dot{\boldsymbol{q}}_j^h||^2) \\ r_{\boldsymbol{ee}} &= \exp(-\beta_{\boldsymbol{ee}} \sum\nolimits_j ||\overline{\boldsymbol{x}_j^h} - \boldsymbol{x}_j^h||^2) \\ r_h^c &= \exp(-\beta_{\boldsymbol{com}} |\Delta \boldsymbol{x}_0^{com}|) \end{aligned} \tag{9}$$

where $\beta_{\boldsymbol{q}} = 2.0$, $\beta_{\boldsymbol{v}} = 0.1$, $\beta_{\boldsymbol{ee}} = 40.0$, and $\beta_{\boldsymbol{com}} = 500000$ are the selected sensitive coefficients. The weighting coefficients are set as $\alpha_h^q = 0.75$, $\alpha_h^v = 0.1$, and $\alpha_h^c = 0.1$.

B. Exoskeleton Control Policy Learning

The human effort reward r_m can be designed to reduce the biological torque. The constraint reward r_e^c is defined to smooth actuation control commands, as follows:

$$r_m = -\boldsymbol{a}_{hip}^3 + \exp(-\beta_{\boldsymbol{\tau}} \sum\nolimits_j ||\boldsymbol{\tau}_j^{hip}||^2) \tag{10}$$

where $\boldsymbol{a}_{hip}$ is the hip related activated muscles, depicted in Fig. 7(a). $\boldsymbol{\tau}_{hip} \in \mathbb{R}^2$ is the biological torque of hip joints. All weighting coefficients are given as $\alpha_e^h = 0.9$, $\alpha_e^m = 0.3$, $\alpha_e^c = 0.1$, and $\beta_{\boldsymbol{\tau}} = 0.008$.

TABLE II
Hyper-parameters for Training Human Control Policy using PPO

Parameter	Value
Actor Network	(256, 256, 50)
Critic Network	(256, 256, 1)
Muscle Network	(1024, 512, 512, 284)
Number of Iterations	$I_{human} = 5000$
Number of Epochs	$L_h = 10$
Horizon of Each Episode	$T = 2048$
Minibatch Size	$B_{human} = 128$
Discount	$\gamma_h = 0.99$
Clip rate	$\epsilon = 0.2$
Learning Rate	$\alpha_a = 0.0001, \alpha_c = 0.001$

TABLE III
Hyper-parameters for Training Exoskeleton Control Policy using PPO

Parameter	Value
LSTM Actor Networks	(256, 256, 50)
LSTM Critic Networks	(256, 256, 1)
Number of Iterations	$I_{exo} = 5000$
Number of Epochs	$L_e = 10$
Horizon of Each Episode	$T = 2048$
Minibatch Size	$B_{exo} = 128$
Discount	$\gamma_e = 0.99$
Clip rate	$\epsilon = 0.2$
Learning Rate	$\alpha_a = 0.0001, \alpha_c = 0.001$

References

[1] S. Lee, M. Park, K. Lee, and J. Lee, "Scalable muscle-actuated human simulation and control," *ACM Transactions On Graphics (TOG)*, vol. 38, no. 4, pp. 1–13, 2019.

[2] M. Li, Y. Wen, X. Gao, J. Si, and H. Huang, "Toward expedited impedance tuning of a robotic prosthesis for personalized gait assistance by reinforcement learning control," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 407–420, 2021.

[3] D. D. Molinaro, K. L. Scherpereel, E. B. Schonhaut, G. Evangelopoulos, M. K. Shepherd, and A. J. Young, "Task-agnostic exoskeleton control via biological joint moment estimation," *Nature*, vol. 635, no. 8038, pp. 337–344, 2024.

[4] A. Rajagopal, C. L. Dembia, M. S. DeMers, D. D. Delp, J. L. Hicks, and S. L. Delp, "Full-body musculoskeletal model for muscle-driven simulation of human gait," *IEEE transactions on biomedical engineering*, vol. 63, no. 10, pp. 2068–2079, 2016.

[5] S. Luo, M. Jiang, S. Zhang, J. Zhu, S. Yu, I. Dominguez Silva, T. Wang, E. Rouse, B. Zhou, H. Yuk, X. Zhou, and H. Su, "Experiment-free exoskeleton assistance via learning in simulation," *Nature*, vol. 630, no. 8016, pp. 353–359, 2024.

[6] S. Luo, G. Androwis, S. Adamovich, E. Nunez, H. Su, and X. Zhou, "Robust walking control of a lower limb rehabilitation exoskeleton coupled with a musculoskeletal model via deep reinforcement learning," *Journal of neuroengineering and rehabilitation*, vol. 20, no. 1, p. 34, 2023.

[7] C. Zuo, K. He, J. Shao, and Y. Sui, "Self model for embodied intelligence: Modeling full-body human musculoskeletal system and locomotion control with hierarchical low-dimensional representation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 13 062–13 069.

[8] V. Caggiano, H. Wang, G. Durandau, M. Sartori, and V. Kumar, "Myosuite: A contact-rich simulation suite for musculoskeletal motor control," in *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, ser. Proceedings of Machine Learning Research, vol. 168. PMLR, 23–24 Jun 2022, pp. 492–507.

[9] S. Song, Ł. Kidziński, X. B. Peng, C. Ong, J. Hicks, S. Levine, C. G. Atkeson, and S. L. Delp, "Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation," *Journal of neuroengineering and rehabilitation*, vol. 18, pp. 1–17, 2021.

[10] J. Weng, E. Hashemi, and A. Arami, "Natural walking with musculoskeletal models using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4156–4162, 2021.

[11] P. Schumacher, D. Haeufle, D. Büchler, S. Schmitt, and G. Martius, "DEP-RL: Embodied exploration for reinforcement learning in overactuated and musculoskeletal systems," in *The Eleventh International Conference on Learning Representations*, 2023.

[12] H.-J. Geiβ, F. Al-Hafez, A. Seyfarth, J. Peters, and D. Tateo, "Exciting action: Investigating efficient exploration for learning musculoskeletal humanoid locomotion," in *2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids)*. IEEE, 2024, pp. 205–212.

[13] L. Rose, M. C. Bazzocchi, and G. Nejat, "End-to-end deep reinforcement learning for exoskeleton control," in *2020 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE, 2020, pp. 4294–4301.

[14] S. Luo, G. Androwis, S. Adamovich, H. Su, E. Nunez, and X. Zhou, "Reinforcement learning and control of a lower extremity exoskeleton for squat assistance," *Frontiers in Robotics and AI*, vol. 8, p. 702845, 2021.

[15] R. Martín-Martín, M. A. Lee, R. Gardner, S. Savarese, J. Bohg, and A. Garg, "Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks," in *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2019, pp. 1010–1017.

[16] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. Pmlr, 2018, pp. 1861–1870.

[17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[18] G. Feng, H. Zhang, Z. Li, X. B. Peng, B. Basireddy, L. Yue, Z. Song, L. Yang, Y. Liu, K. Sreenath *et al.*, "Genloco: Generalized locomotion controllers for quadrupedal robots," in *Conference on Robot Learning*. PMLR, 2023, pp. 1893–1903.

[19] Carnegie Mellon University, "Cmu graphics lab motion capture database," http://mocap.cs.cmu.edu/, 2021, accessed: 10 November 2021.

[20] B. Lim, J. Lee, J. Jang, K. Kim, Y. J. Park, K. Seo, and Y. Shim, "Delayed output feedback control for gait assistance with a robotic hip exoskeleton," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 1055–1062, 2019.