ResidualAssist: Residual Reinforcement Learning in Simulation for Leg-Decoupled Exoskeleton Control in Walking, Running, and Asymmetric Gait

Abstract— **Learning exoskeleton assistive torque directly through reinforcement learning remains challenging because assistance is highly phase-sensitive, whereas noisy biomechanical rewards provide indirect guidance for torque timing and amplitude. Here, we propose ResidualAssist, a structured residual reinforcement learning framework for hip exoskeleton control. Rather than learning the entire assistive torque profile from scratch, ResidualAssist combines a simple but interpretable velocity-based torque prior with a reinforcement learning policy that learns state-dependent residual torque corrections. A leg-decoupled policy structure generates bilateral assistive torque from each leg's state and is evaluated under a unilateral-load-induced asymmetric gait condition. In simulation, the learned exoskeleton policy combined with human adaptation training can reduce RMS biological hip moment and RMS hip muscle-group activation by 13.2% and 10.2% relative to the no-exoskeleton baseline. The policy is then deployed on a physical exoskeleton, where the measured torque profiles showed high consistency ($R \geq 0.92$) with the simulated profiles at 1.25 m/s walking speed. Our controller is further evaluated with six subjects treadmill walking (0.75–1.8 m/s), running (2.5 m/s), and an asymmetric-gait condition under a 2 kg unilateral load, in which the exoskeleton assistance maintains a positive power ratio approximately 90% on both legs.**

I. Introduction

Lower-limb exoskeletons have emerged as a promising technology for augmenting and restoring human mobility, including gait assistance, rehabilitation, and load carriage. Although data-driven methods have been increasingly used in lower-limb exoskeleton research for gait phase estimation [1], [2], locomotion mode recognition [3]–[5], and prediction of biomechanical quantities such as joint moments and ground reaction forces [6], many of these approaches are still formulated as supervised learning problems. In such settings, models are trained to map sensor measurements to pre-labeled kinematic or biomechanical targets. For torque-level exoskeleton control, this paradigm still faces limitations. First, supervised learning requires large amounts of labeled human-exoskeleton data, which are time-consuming to collect and may raise safety concerns when exploring diverse assistance strategies. Second, supervised models are typically tied to the training distribution and generalize poorly to unseen gait patterns, or locomotion conditions. Although classical reinforcement learning method has been explored for exoskeleton control [7], it optimizes only a small set of parameters within a predefined assistive torque profile structure.

These limitations of supervised learning motivate simulation-based deep reinforcement learning (DRL), which enables policy learning from large-scale interaction data and optimization of reward functions defined on biomechanical quantities accessible in simulation. Some studies [8]–[11] have leveraged musculoskeletal simulation environments to build digital humans, allowing access to biological signals. Building upon these platforms, deep reinforcement learning has been increasingly adopted in exoskeleton torque generation. Kim et al. [12] use rigid body simulation with DDPG for smooth locomotion transitions. Zhou et al. [13] use AMP-PPO also in rigid body simulation to train a CRNN for hip-torque prediction. These rigid body simulation use joint actuation rather than muscle-driven dynamics. Consequently, they do not provide muscle activation signals and represent human motion using simplified joint-level dynamics. Luo et al. [14] present policy distillation for sensor-robust control but validate only in simulation. Yuan et al. [15] introduce a four-stage curriculum that trains a musculoskeletal human actor and a hip exoskeleton actor and deploy it to a hip exoskeleton for biomechanical and metabolic validation [16]. Exo-Plore [17] develops a human-aligned musculoskeletal simulation method, but it just optimizes the gain and delay parameters of a predefined delayed output-feedback controller [18], restricting assistance to a fixed torque-generation structure. Barati et al. [19] train a torque controller using a reflex-based musculoskeletal model and deploy it to a hip exoskeleton but they model the exoskeleton as an ideal torque source, without explicitly modeling physical human–exoskeleton coupling. Luo et al. [20] learn a versatile hip-exoskeleton control policy in simulation and deploy it on real hardware but it lacks detailed quantitative analysis of the sim-to-real gap. Park et al. [21] develop a deep policy-distillation framework for hip exoskeleton control, but they have to train a privileged teacher policy using full musculoskeletal and device states, requiring an additional policy-distillation stage for IMU-based deployment.

Despite recent progress, two challenges remain in reinforcement-learning-based torque control. The first challenge is in optimization process during learning. The low-level fixed-gain proportional-derivative (PD) controller in [20] cannot directly adapt its PD parameters to changes in user dynamics or locomotion state. Direct torque learning policies in [12], [15], [19], [21] offer more flexibility. But learning the complete assistive torque profile requires the policy to simultaneously discover assistance timing and magnitude from scratch. This broad search is difficult to

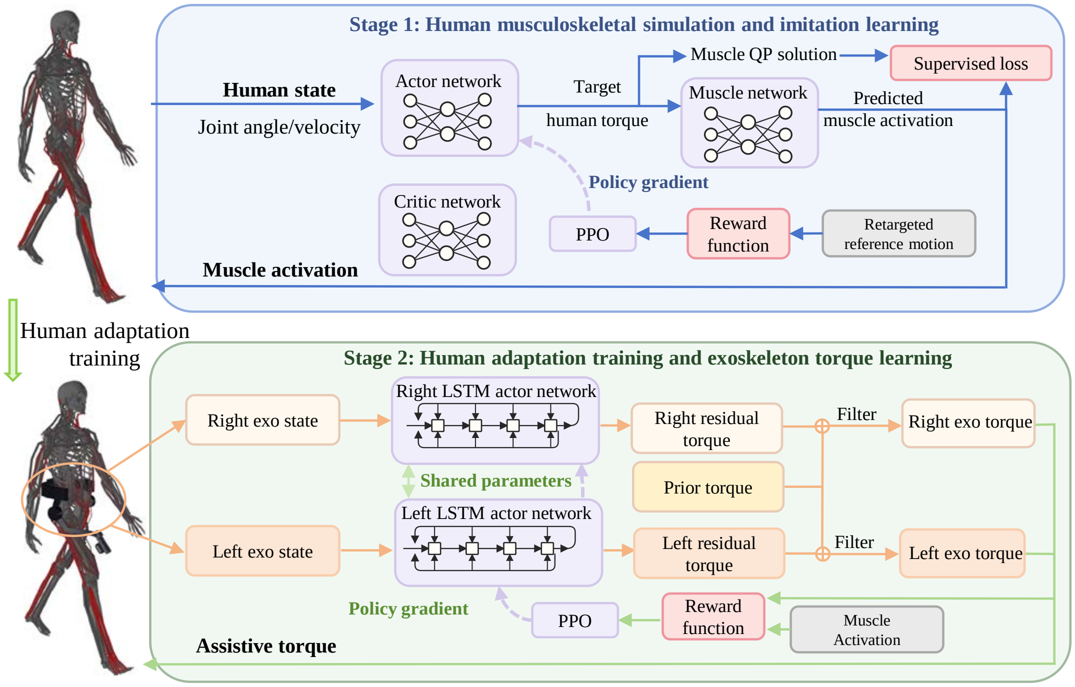


Fig. 1. ResidualAssist combines a velocity-based nominal torque with learned residual torque corrections and applies the same actor separately to each leg using only that leg's local state. Stage 1 trains the musculoskeletal human controller via imitation learning. Stage 2 jointly updates the human controller for adaptation and trains the exoskeleton policy. The velocity-based prior provides nominal assistance, while PPO learns residual torque corrections. The left and right legs share actor network parameters but use only each leg's local state as actor input.

TABLE I
COMPARISON OF REPRESENTATIVE DRL-BASED HIP EXOSKELETON CONTROL METHODS.

Method	Muscle model	Prior-guided torque learning	Quantified sim-to-real validation	Asymmetric gait evaluation
Kim et al. [12]	×	×	×	×
Zhou et al. [13]	×	×	×	×
Exo-Plore [17]	✓	×	×	×
Luo et al. [20]	✓	×	✓	×
Yuan et al. [15]	✓	×	✓	×
Park et al. [21]	✓	×	✓	×
Barati et al. [19]	✓	×	✓	×
Ours	✓	✓	✓	✓

optimize because noisy biomechanical rewards, such as biological joint moments and muscle activations, provide weak and high-variance learning signals. The second challenge is bilateral assistance robustness. When assistive torque depend on both-leg states, a disturbance on one side may alter the assistance delivered to the other side. Table I summarizes seven representative simulation-based hip exoskeleton control methods. None of them investigates the prior-guided torque learning and asymmetric gait control problem. Motivated by these challenges, this work addresses two research questions: How can biomechanical prior knowledge reduce the difficulty of learning useful exoskeleton torque from scratch? Can each leg of a bilateral exoskeleton be controlled using only information from that same leg? To answer these questions, we develop a simulation-based hip-exoskeleton residual learning method built upon a muscle-actuated human controller. The exoskeleton controller combines a prior torque with a residual policy and uses a leg-decoupled actor to generate bilateral assistance from each leg's local state.

The contributions of this work are twofold. First, instead of learning the complete assistive torque profile from scratch, we use a simple and effective velocity-based nominal torque and train the reinforcement learning policy to learn only residual torque corrections. This formulation reduces the exploration space and avoids the low-assistance local optimum. Second, rather than using the states of both legs to jointly generate bilateral torque, we apply the same actor separately to each leg using only that leg's local state. This removes direct dependence on the opposite leg from torque generation while retaining shared actor parameters between the two legs.

II. TWO STAGE FORMULATION AND SIMULATION MODELING

This section formalizes the two-stage training procedure illustrated in Fig. 1 and describes our simulation models used to implement it. We first define the roles of the human and exoskeleton controllers in the two stages and then introduce the musculoskeletal and exoskeleton modeling.

A. Two Stage Problem Formulation

Let $H_{\phi,\eta}$ denote the human controller, including the high-level motion policy and the low-level muscle network, and let π_θ^E and V_ψ^E denote the exoskeleton actor and critic, respectively. The training procedure separates human locomotion acquisition from exoskeleton assistance learning. Unlike the simultaneous human–exoskeleton training in [20], our two-stage procedure establishes a pretrained human controller before introducing exoskeleton assistance. This design enables pre-deployment evaluation of the exoskeleton controller using biomechanical metrics, such as simulated biological hip moments and muscle activations.

In Stage 1, following prior work on muscle-actuated control [22], we adopt a two-level architecture to train the $H_{\phi,\eta}$. A high-level actor–critic policy is optimized using PPO to generate target joint commands that track retargeted reference motions, which are converted into desired joint torques by a proportional–derivative (PD) controller. A low-level muscle network maps these torques to muscle activations and is supervised using quadratic programming (QP) solutions that enforce joint-torque consistency.

In Stage 2, the pretrained human controller interacts with the initialized exoskeleton actor and critic in the musculoskeletal simulation. In the real world, the human response to assistance is not fixed. Users continuously adjust their gait timing, joint kinematics, and muscle coordination in response to the applied torque. Therefore, the exoskeleton policy π_θ^E is optimized, while the human controller $H_{\phi,\eta}$ is updated to adapt to assistance under reference motion and target speed constraints.

B. Musculoskeletal and Exoskeleton Modeling

The musculoskeletal model used in this work is adopted from [22], which includes a tree structure of rigid bones connected by 8 revolute joints and 14 ball-and-socket joints. The model also includes 284 musculotendon units corresponding to skeletal muscles that contribute to joint motion in the skeleton. A Hill-type muscle model [8] is used to formulate muscle contraction dynamics. In Stage 2, the exoskeleton link masses and inertial properties are imported from the device URDF, and the links are rigidly attached to the human body segments. Their contributions are therefore included in the coupled human–exoskeleton multibody dynamics. The modeled exoskeleton has a mass of 2.72 kg, excluding the electronics, wearable, and battery.

III. Human Adaptation Training and Exoskeleton Policy Torque Learning

Building upon the pretrained musculoskeletal human controller, this section presents the proposed exoskeleton learning framework. The method consists of two key components: a structured residual torque-learning formulation for continuous assistance optimization and a parameter-shared leg-decoupled control policy for bilateral control.

A. Velocity-based Prior Torque

A central design choice in the proposed method is the use of joint angular velocity as a biomechanical prior for exoskeleton torque assistance. From a biomechanical perspective, hip-joint mechanical power is determined by the alignment between joint torque and angular velocity. A nominal assistive torque aligned with the instantaneous joint velocity therefore generates nonnegative mechanical power. In particular, for a positive velocity gain k_v, setting $\tau_t = k_v \dot{q}_t$ gives $P_t = k_v \dot{q}_t^2 \geq 0$. This relationship allows the torque direction and magnitude to vary continuously with the joint motion, without requiring explicit gait-phase segmentation

Algorithm 1 Human adaptation training and leg-decoupled residual exoskeleton policy

Require: Pretrained human–muscle model $H_{\phi,\eta}$; leg-local actor π_θ^E; critic V_ψ^E; velocity torque gain k_v; residual torque gain k_c; torque filter $\mathcal{F}(\cdot)$
Ensure: Deployable exoskeleton policy π_θ^E
1: Initialize θ and ψ; load $H_{\phi,\eta}$
2: **for** $k = 1, \ldots, I_{\text{exo}}$ **do**
3: Initialize $\mathcal{D}_E, \mathcal{D}_H \leftarrow \emptyset$; reset the parallel environments
4: **for** $t = 0, \ldots, T-1$ **do**
5: Obtain the human command and muscle activation from $H_{\phi,\eta}(s_t^H)$
6: **for** $i \in \{L, R\}$ **do**
7: Form the leg-local observation: $o_t^i = [\hat{q}_t^i, \dot{\hat{q}}_t^i, \hat{\tau}_{t-1}^i]$
8: Sample the residual command using the shared actor: $\Delta\tau_t^i \sim k_c \pi_\theta^E(\cdot \mid o_t^i)$
9: Compose the assistive torque: $\tau_t^i = \mathcal{F}\Big[\mathrm{Clip}(k_v \dot{q}_{f,t}^i + \Delta\tau_t^i)\Big]$
10: **end for**
11: Form the bilateral critic state: $s_t^E = [o_t^L, o_t^R]$
12: Apply (τ_t^L, τ_t^R) and step the human–exoskeleton simulation
13: Compute the exoskeleton reward r_t^E and the constrained human reward r_t^H
14: Store the joint exoskeleton transition in $\mathcal{D}_E$ and the human transition in $\mathcal{D}_H$
15: **end for**
16: Estimate $\hat{A}_t^E$ and $\hat{R}_t^E$ using $V_\psi^E(s_t^E)$
17: Update the shared actor θ using the PPO clipped surrogate objective
18: Update the critic ψ using the bilateral returns
19: Update $H_{\phi,\eta}$ using $\mathcal{D}_H$ under the constrained human objective
20: **end for**
21: **return** π_θ^E

or a predefined torque trajectory. This velocity-aligned assistance can be interpreted as a virtual negative-damping mechanism [23], [24]. Ramella et al. [25] use virtual negative damping as the complete assistance profile and only tune its gain through Bayesian optimization. In contrast, we use a velocity-based term as a nominal prior, while recurrent PPO learns state-dependent residual corrections to the final torque profile.

From a learning and optimization perspective, the velocity-based torque prior provides a nominal assistance pattern, so the policy does not need to discover the complete torque waveform from scratch. Instead, the residual policy learns corrections that refine the timing and amplitude of this nominal pattern. This structured formulation avoids the low-assistance solution observed in direct torque learning under the same training budget. A detailed explanation can be found in Fig. 4 in the experiment section.

Based on this formulation, we define the velocity-based prior torque as

$$\tau_t^{\text{prior}} = k_v \dot{q}_{f,t}^i \tag{1}$$

where $k_v > 0$ is a fixed velocity gain. $\dot{q}_{f,t}^i$ denotes the hip angular velocity low-pass filtered using a second-order Butterworth filter with a cutoff frequency of 10 Hz.

B. Residual Torque Prediction using Reinforcement Learning

Instead of learning the complete assistive torque profile from scratch, we use reinforcement learning to learn residual torque corrections to a velocity-based nominal torque. The objective of this stage is to generate assistive torque that reduces biological joint torque and muscle activation while preserving natural human kinematics.

a) Problem Formulation: The proposed approach decomposes exoskeleton torque into a biomechanically meaningful prior component and a learned residual compensation term. The final assistive torque is obtained by adding a learned residual torque,

$$\tau_t^{\text{exo}} = \mathcal{F}\big(\tau_t^{\text{prior}} + \Delta\tau_t\big), \tag{2}$$

where $\Delta\tau_t$ is predicted by the reinforcement learning policy and $\mathcal{F}(\cdot)$ denotes a low-pass filter, which is implemented as a second-order low-pass Butterworth filter with a cutoff frequency of $f_{\text{cutoff}} = 5\,\text{Hz}$.

b) Policy Observation and Action: Different from commonly used leg-coupled structure, which takes the concatenated states of both legs and jointly outputs the two residual torque commands, we adopt a leg-decoupled design. The observation for leg $i \in \{L, R\}$ is defined as

$$o_t^i = [\hat{q}_t^i, \dot{\hat{q}}_t^i, \hat{\tau}_{t-1}^i], \tag{3}$$

where $\hat{q}_t^i$ and $\dot{\hat{q}}_t^i$ denote the mean of hip angle and angular velocity in the current step and past two steps. $\hat{\tau}_{t-1}^i$ is the normalized assistive torque (0-1) applied at the previous control step. The actor is implemented as a two-layer LSTM with 256 hidden units and shared parameters across legs. For each leg, the policy outputs an action u_t^i, which is scaled to obtain the residual torque,

$$\Delta\tau_t^i = k_c\, u_t^i. \tag{4}$$

The final assistive torque for each leg is computed by combining the velocity-based prior torque with the learned residual torque, followed by clipping to the actuator limit $[-\tau_{\max}, \tau_{\max}]$ and low-pass filtering,

$$\tau_t^i = \mathcal{F}[Clip(\big(k_v \dot{q}_{f,t}^i + \Delta\tau_t^i\big), -\tau_{max}, \tau_{max}]. \tag{5}$$

C. Reward Function Design

a) Human Adaptation Training Reward Function: To prevent the simulated human from exploiting the reward through unrealistic gait changes, the stage 1 imitation objective is retained as an anchoring term, together with velocity-tracking constraints. This constrained formulation allows the human controller to adapt its gait and muscle coordination to assistance while maintaining reference-motion and target-velocity tracking. The human reward in stage 2 is defined as

$$r_{\text{H}} = \alpha_{\text{imit}}^h r_{\text{imit}} + \alpha_{\text{vel}}^h r_{\text{vel}} + \alpha_{\text{me}}^h r_{\text{me}}, \tag{6}$$

where r_{imit} encourages the controller to remain close to the reference natural gait, r_{vel} maintains the target locomotion speed at 1.25 m/s of the reference motion. r_{me} is a muscle effort proxy reward that encourages a reduction in the simulated effort of the selected hip-muscle groups relative to the matched no-exoskeleton condition. The muscle effort proxy reward r_{me} is defined as:

$$\begin{aligned} r_{\text{me}} &= \text{clip}\left(1 - \frac{E_t^{\text{with-exo}}}{E^{\text{no-exo}}}, -1, 1\right), \\ E_t^{\text{with-exo}} &= \sum_{m \in \mathcal{M}_{\text{hip}}} M_m a_{m,t}. \end{aligned} \tag{7}$$

where $E_t^{\text{with-exo}}$ denotes the mass-weighted muscle effort proxy at time step t, M_m is the modeled mass of muscle m, and $a_{m,t} \in [0, 1]$ is its activation at time step t. The fixed scalar baseline $E^{\text{no-exo}}$ is computed as the mean muscle effort proxy of all samples collected during a separate no-exoskeleton rollout of the human controller under the same reference motion and target speed settings. The selected hip-muscle set $\mathcal{M}_{\text{hip}}$ includes the Biceps Femoris, Rectus Femoris, Gluteus Maximus, Semitendinosus, and Gluteus Medius.

b) Exoskeleton Reward Function: The exoskeleton policy is optimized to reduce simulated muscular effort while avoiding assistance that resists the user's motion. Its reward is defined as

$$r_{\text{E}} = \alpha_{\text{me}} r_{\text{me}} + \alpha_{\text{power}} r_{\text{power}} + \alpha_{\text{smooth}} r_{\text{smooth}} - \alpha_{\text{sat}} r_{\text{sat}}, \tag{8}$$

The muscle effort proxy reward r_{me} is the same as that used in human adaptation reward. The power reward r_{power} encourages the exoskeleton to deliver positive mechanical power:

$$r_{\text{power}} = \tanh\left(\frac{P_{\text{exo}}}{P_{\text{ref}}}\right), \qquad P_{\text{ref}} = 20\ \text{W}, \tag{9}$$

To encourage gait periodic torque while discouraging saturated and abruptly varying torque profiles, we define the smoothness reward r_{smooth} using a harmonic residual at the reference gait frequency and saturation penalty r_{sat}. The harmonic residual measures the deviation of the output torque from the discrete harmonic-oscillator relation at the reference gait frequency. For each leg $i \in \{L, R\}$, the harmonic residual is computed as

$$s_{\text{harm},t}^i = \tau_t^i - 2\tau_{t-1}^i + \tau_{t-2}^i + \omega_g^2\left(\tau_{t-1}^i - \bar{\tau}_t^i\right), \tag{10}$$

where τ_t^i is the applied assistive torque, and $\bar{\tau}_t^i$ is its exponential moving average with a one-gait-cycle time scale. The discrete gait frequency is $\omega_g = 2\pi/N_g$, where N_g is

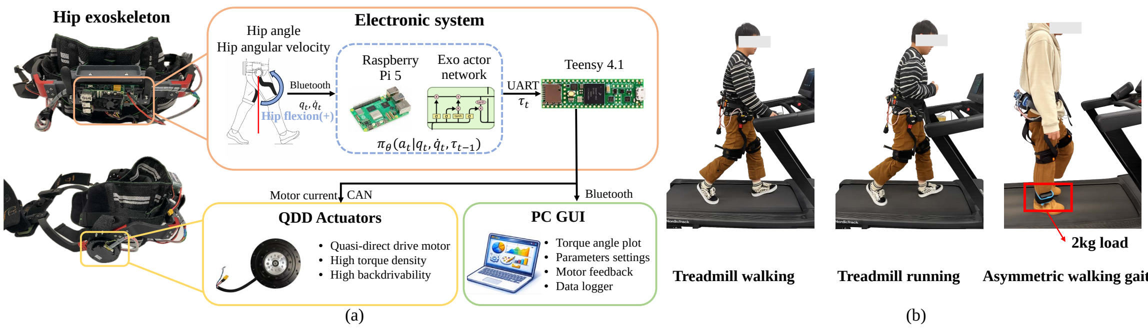


Fig. 2. (a) Experimental setups for policy deployment. Customized electronic system for policy deployment on quasi-direct driven hip exoskeletons. The forward propagation of trained actor network with optimized parameters is run on Raspberry Pi 5 to generate the motor commands. The lower-level controller is run on Teensy 4.1 to send the current commands to actuators via CAN. (b) Experiments demonstrating policy deployment on hip exoskeletons for treadmill walking, running and asymmetric walking gait.

TABLE II

TRAINING HYPERPARAMETERS

Parameter	Value
Simulation Frequency	$f_{sim} = 600\ Hz$
Torque Control Frequency	$f_{\tau} = 100\ Hz$
Number of Environments	$N = 32$
Number of Stage 1 Iterations	$I_{human} = 10000$
Number of Stage 2 Iterations	$I_{exo} = 5000$
Number of Epochs	$L = 10$
Transition per Iteration	$T = 2048$
Minibatch Size	$B = 256$
Discount	$\gamma_e = 0.99$
Clip Rate	$\epsilon = 0.2$
Learning Rate	$\alpha = 1 \times 10^{-4}$
Torque Scale Gain	$k_v = 3.54, k_c = 10$
Max Torque Limits	$\tau_{max} = 15\ N{\cdot}m$
Reward Weights (Eq. 9)	$\alpha^h_{imit} = 1.0, \alpha^h_{vel} = 0.3,\ \alpha^h_{me} = 0.4$
Reward Weights (Eq. 11)	$\alpha_{me} = 0.63,\ \alpha_{power} = 0.27$ $\alpha_{smooth} = 0.10,\ \alpha_{sat} = 0.27$

the number of control steps per reference gait cycle. The smoothness reward and the saturation penalty are defined as:

$$r_{\text{smooth}} = \exp\left(-\lambda\sqrt{\frac{1}{2}\sum_{i\in\{L,R\}}\left(s^i_{\text{harm},t}\right)^2}\right),$$
$$r_{\text{sat}} = \frac{1}{2}\sum_{i\in\{L,R\}}\max\left(0, \frac{|\tau^i_t| - \tau_{\text{sat}}}{\tau_{\max} - \tau_{\text{sat}}}\right), \tag{11}$$

where the weight λ is set to 5 and τ_{sat} is set to 12 in the experiment. In r_{sat}, τ^i_t is the assistive torque before clipping.

IV. EXPERIMENTS

A. Experiment Setups

a) Simulation Training Setups and Hyperparameters: The proposed method was trained on a PC running Ubuntu 24.04 with an NVIDIA GeForce RTX 5080 GPU. Each training stage required approximately 8 hours. The training hyperparameters are listed in TABLE II, and the exoskeleton policy network contains approximately 1.6 million trainable parameters. The assistive torque command is updated at 100 Hz. The velocity gain is set to $k_v = 3.54$ and the actuator limit is set to $\tau_{\max} = 15$ N·m. The exoskeleton policy was trained only using the 1.25 m/s walking reference motion. The same policy and fixed gains were used without retraining or retuning for all deployment conditions.

b) Deployment Process and Experiment Protocol: The developed hip exoskeleton and its customized electronic system are illustrated in Fig. 2 (a). The mass of the exoskeleton is 3.8 kg including the battery and electronics. Human hip joint angles and angular velocities are transmitted to a Raspberry Pi 5 at a sampling rate of 100 Hz and used as inputs to the actor network to generate the assistive hip torque. The filtered torque command is sent to a Teensy 4.1 microcontroller, which converts it into current commands for the QDD actuators at 100 Hz. In addition, a PC-based graphical user interface (GUI) developed using PyQt5 is used to visualize experimental data. All experimental procedures were approved by the Institutional Review Board (IRB) of the authors' institution.

The protocol of deploying the trained exoskeleton control policy is depicted in Fig. 2 (b). The experimental protocol consisted of two locomotion conditions: treadmill walking-running and asymmetric gait condition. For treadmill walking-running, subjects walk at three speeds and run at one speed (0.75 m/s, 1.25 m/s, 1.8 m/s for walking and 2.5 m/s for running), with each speed maintained steady for at least 20 seconds. The treadmill speed was continuously varied across conditions without resting intervals between consecutive speeds or motion modes, enabling the evaluation of controller performance under continuously varying locomotion speeds. For asymmetric gait condition, subjects walked at 1.25 m/s with a 2 kg unilateral ankle weight attached to right leg to simulate unilateral-load-induced asymmetric gait, inducing deliberate asymmetry in hip kinematics and gait timing between legs.

Six healthy adults (n = 6) participated in this experiment. The participants' demographics were as follows: age, 23.3$\pm$2.10 years; weight, 77.88$\pm$17.20 kg; height, 177.7$\pm$5.05 cm (mean $\pm$ standard deviation). Throughout all experimental conditions, hip joint kinematics and exoskeleton interaction data were collected, including hip joint angles, hip angular velocities, exoskeleton torque, and exoskeleton power. All simulation and experimental hip angle cycles are segmented to begin at peak hip flexion, without heel-strike detection.

B. Simulation Validation and Assistive Performance

We first conduct ablation studies to validate the contribution of each component in the proposed framework, and then evaluate the overall assistive performance in simulation.

We compare four conditions: (1) a no exoskeleton baseline (No Assist), (2) an exoskeleton-worn condition with assistance off (Assist Off), (3) active assistance with the pretrained human controller frozen (Assist On + Frozen Human), and (4) active assistance with the human adaptation training together with the exoskeleton policy (Assist On + Adapted Human). Fig. 3 illustrates that the learned assistive torque exhibits an emergent timing shift relative to the baseline biological hip moment. This timing shift is not prescribed explicitly and coincides with a larger reduction in biological hip moment when the simulated human controller is allowed to adapt. Fig. 4 compares the training performance between learning from scratch policy and the proposed residual policy. In the evaluated training run, learning from scratch policy tends to converge to a low-assistance local optimum, sacrificing human-effort reduction to avoid unfavorable torque generation. The velocity-based prior prevents this conservative solution by providing a positive-power nominal assistance pattern, allowing residual learning to optimize timing and amplitude corrections while maintaining substantial assistive output.

Then we compare the hip biological moment and hip muscle group activation of four conditions in simulation. As shown in Fig. 5 (a), relative to the No Assist baseline, the adaptation condition reduces peak and RMS biological moment by 21.9% and 13.2%, respectively. In comparison, assistance with a frozen human controller reduces these metrics by only 10.9% and 6.0%. Wearing the exoskeleton with assistance disabled provides no unloading benefit, slightly increasing peak and RMS moment by 0.5% and 2.9%. A similar trend is observed for hip muscle-group activation in Fig. 5 (b). The adaptation condition reduces peak activation by 19.5% and RMS activation by 10.2%, whereas the frozen-human condition reduces peak activation by 9.1% but increases RMS activation by 2.0%. The Assist Off condition increases RMS activation by 3.2%. These results show that human adaptation substantially increases the simulated unloading achieved with exoskeleton assistance. Allowing the human controller to adjust its gait and muscle coordination in response to the learned assistance produces larger reductions in biological hip moment and muscle activation than keeping the human controller frozen. The reported reductions are derived from musculoskeletal simulation and should be interpreted as indicators of assistive potential rather than direct physiological measurements during hardware deployment. But these comparisons illustrate a practical advantage of musculoskeletal simulation: candidate exoskeleton controllers can be screened before hardware deployment using simulated biological joint moments and muscle activation as complementary indicators of assistive potential.

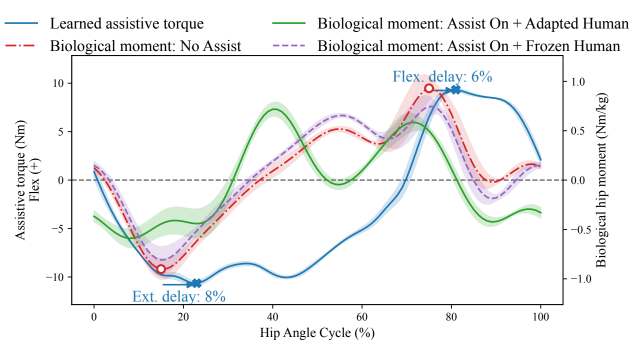
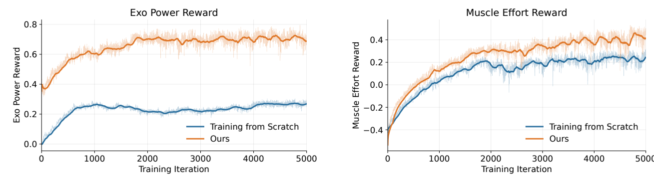
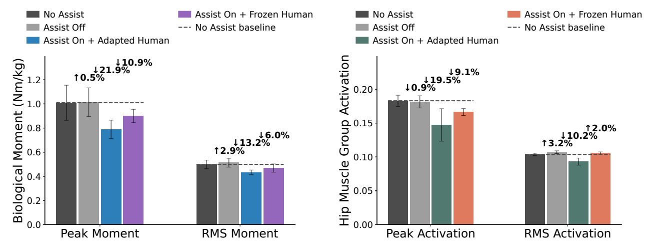


Fig. 3. The learned assistive torque exhibits an emergent timing shift, while human adaptation further reduces the simulated biological hip moment. The learned assistive torque is compared with the No Assist biological moment and with biological moments under assistance with either an adapted or frozen human controller (Left leg). Without prescribing an explicit timing offset, the extension and flexion torque peaks occur approximately 8% and 6% later than the corresponding baseline biological moment peaks, respectively.



(a) (b)

Fig. 4. The velocity-based prior helps residual learning avoid the low assistance solution reached by direct torque learning under the same training budget. Residual learning maintains higher steady state exoskeleton power and muscle effort rewards than learning the complete assistive torque profile from scratch. (a) Exoskeleton power reward and (b) Exoskeleton muscle effort proxy reward over 5,000 training iterations.



(a) (b)

Fig. 5. Exoskeleton assistance combined with human adaptation produces the largest simulated reductions in biological hip moment and hip muscle group activation. Relative to the No Assist baseline, Assist On + Adapted Human reduces RMS biological hip moment by 13.2% and RMS hip muscle group activation by 10.2%. Compared with Assist On + Frozen Human, allowing the human controller to adapt leads to substantially greater simulated unloading. The dashed line indicates the No Assist baseline. (a) Peak and RMS biological hip moment. (b) Peak and RMS hip muscle group activation.

C. Sim-to-real Deployment Results

Fig.6 presents the exoskeleton assistive torque and corresponding positive power across six subjects (S1–S6) during walking (0.75 m/s, 1.25 m/s, and 1.8 m/s) and running (2.5 m/s). As walking speed increases, the magnitude of peak assistive torque and positive power generally increases, reflecting the higher mechanical demands at faster gait speeds. The inter-subject variability appears mainly as amplitude and subtle timing differences rather than profile shape, reflecting individual gait patterns across subjects. Even at the same treadmill speed, subjects exhibit slightly different hip kinematics (stride length, range of motion, velocity profile). These kinematic differences naturally translate into subject-specific torque amplitudes while preserving the overall phase

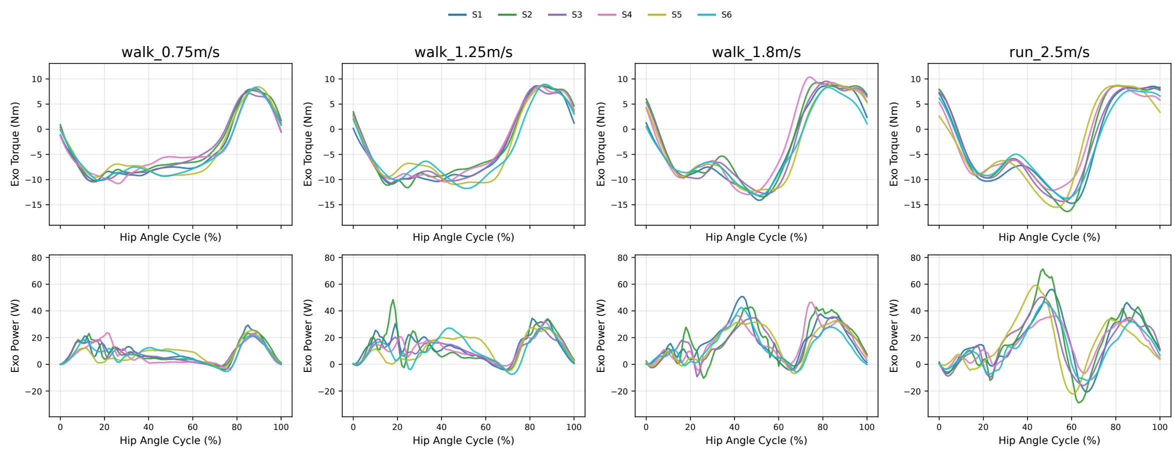


Fig. 6. The same ResidualAssist controller operates across the tested walking and running speeds without retuning. Assistive torque and positive power generally increase with locomotion speed across six subjects, while the overall cyclic assistance pattern is preserved from 0.75 m/s walking to 2.5 m/s running. Assistive hip torque (top row) and corresponding mechanical power (bottom row) across six subjects under different locomotion speeds (walking at 0.75, 1.25, and 1.8 m/s, and running at 2.5 m/s) are shown.

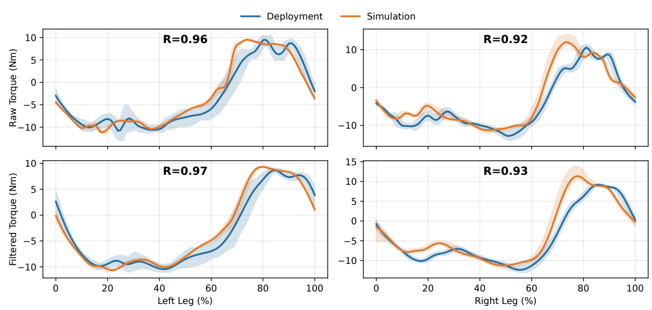


Fig. 7. The deployed assistive torque profiles show high consistency with their simulated counterparts at 1.25 m/s walking, with Concordance Correlation Coefficients of $R_{\text{left}} = 0.97$ for filtered left assistive torque and $R_{\text{right}} = 0.93$ for filtered right assistive torque. Raw (top) and filtered (bottom) torque profiles are shown for the left and right legs.

TABLE III

ASSISTIVE TORQUE AND POSITIVE POWER INCREASED WITH LOCOMOTION SPEED, WHILE THE POSITIVE POWER RATIO (PPR) REMAINED HIGH ACROSS THE TESTED CONDITIONS

Speed (m/s)	**RMS τ (Nm)**	**Peak τ (Nm)**	P^+ **(W)**	P^- **(W)**	**PPR (%)**
0.75	7.23 ± 0.27	10.82 ± 0.54	9.00 ± 0.90	-0.26 ± 0.11	97.1 ± 1.3
1.25	8.05 ± 0.15	12.00 ± 0.80	13.26 ± 1.05	-0.48 ± 0.15	96.5 ± 1.3
1.80	8.75 ± 0.44	14.26 ± 1.74	17.90 ± 2.25	-0.95 ± 0.40	95.0 ± 2.3
2.50	8.88 ± 0.17	15.07 ± 0.59	18.99 ± 1.50	-2.36 ± 0.46	89.0 ± 2.0

Note: Values are mean $\pm$ SD across six participants. For each participant, the left- and right-leg metrics were averaged before computing the group statistics. PPR denotes positive power ratio.

structure. Table III summarizes the mean and standard deviation of the power metrics (mean of two legs) across six subjects under four locomotion conditions. The mean positive power and mean negative power are obtained by summing the positive and negative portions of the power and dividing by the total number of sample points. The positive power ratio (PPR) represents positive mechanical power as a percentage of the total absolute mechanical power. The results show high positive power and positive power ratio. For example, at 1.25 m/s speed, the mean positive power is 26.52 W for two legs and positive power ratio is 96.5 %.

To quantify sim-to-real transfer consistency, we compute the Concordance Correlation Coefficient R between simulated and measured mean signals over the full hip angle cycle at 1.25 m/s walking speed. As shown in Fig. 7, the raw exoskeleton-torque profiles achieve $R \geq 0.92$ across both legs. Applying the same second-order Butterworth filter in simulation and deployment further raises the filtered torque to $R \geq 0.93$, confirming that the learned torque timing and amplitude are preserved during deployment.

D. Evaluation under Asymmetric Gait Conditions

As specified in our protocol, a 2 kg ankle weight is attached to the right leg to simulate unilateral-load-induced asymmetric gait. Fig. 8 compares the exoskeleton torque and power of leg-coupled and leg-decoupled structures under this condition. The recorded bilateral hip kinematics from the leg-decoupled deployment were replayed offline through the leg-coupled policy, ensuring that the torque and power differences reflected the controller structures rather than variations in gait kinematics between trials.

Under the tested unilateral-load condition, the leg-decoupled controller showed smaller inter-cycle variability in the torque and power profiles than the leg-coupled controller. It maintained a positive power ratio (PPR) approximately 90% on both legs, whereas the leg-coupled controller achieved lower PPR of 85% and 76% on the unloaded and loaded legs, respectively. By generating each torque command solely from the corresponding leg's local observation, the leg-decoupled structure removes direct contralateral-state dependence while allowing each leg to respond to its own kinematics. The observed PPR differences are consistent with less unintended negative-power assistance under the tested asymmetric-gait condition.

V. DISCUSSION AND CONCLUSION

Structured Residual Torque Learning. The training result shows that learning the complete assistive torque waveform from scratch settles at a conservative low-assistance solution under the tested training budget. In contrast, structured

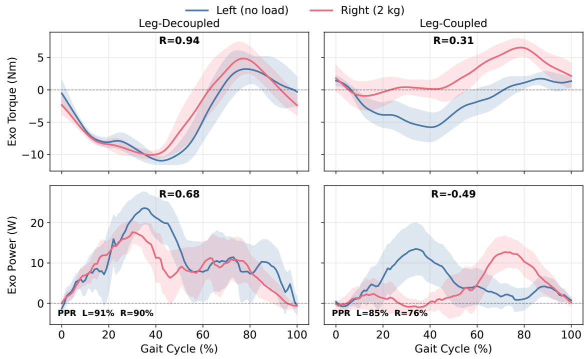


Fig. 8. The leg-decoupled controller maintains higher positive power ratios under the asymmetry gait condition. It achieves a positive power ratio approximately 90% on both legs, compared with 85% and 76% for the unloaded and loaded legs with the leg-coupled controller. (R: Concordance Correlation Coefficients)

residual learning maintains higher steady-state exoskeleton power and muscle effort rewards. These results suggest that the prior information provides a useful nominal assistance pattern that reduces the need for unconstrained torque exploration.

Leg-Decoupled Bilateral Control. Treating each leg independently under a shared policy removes contralateral sensor measurements from the actor input, thereby eliminating direct cross-leg dependence in torque generation. This structure maintains higher positive-power ratios under the tested unilateral-load-induced asymmetric gait and may also limit the disturbance of side-specific IMU sensor bias or drift on the opposite-leg torque command.

Limitations. The leg-decoupled controller was evaluated only under unilateral-load-induced asymmetric gait in healthy participants. Although this controlled loading protocol introduced bilateral kinematic asymmetry, it does not capture the neuromuscular deficits and compensatory strategies associated with pathological gait. Accordingly, the observed advantage of leg-decoupled control is limited to the tested loading condition, and its applicability to pathological gait asymmetry has not yet been evaluated.

REFERENCES

[1] K.-W. Park, J. Choi, and K. Kong, "Data-driven modeling for gait phase recognition in a wearable exoskeleton using estimated forces," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 3072–3086, 2023.

[2] S. Choi, C. Ko, and K. Kong, "Walking-speed-adaptive gait phase estimation for wearable robots," *Sensors*, vol. 23, no. 19, p. 8276, 2023.

[3] Y. Qian, Y. Wang, C. Chen, J. Xiong, Y. Leng, H. Yu, and C. Fu, "Predictive locomotion mode recognition and accurate gait phase estimation for hip exoskeleton on various terrains," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6439–6446, 2022.

[4] X. Wu, Y. Ma, X. Yong, C. Wang, Y. He, and N. Li, "Locomotion mode identification and gait phase estimation for exoskeletons during continuous multilocomotion tasks," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 13, no. 1, pp. 45–56, 2019.

[5] H. Shin, S. Kim, S. Kim, J. Lee, J. Kim, and K. Kim, "Multi-modal locomotion mode recognition in the real world for robotic hip complex exoskeletons," *IEEE Robotics and Automation Letters*, 2025.

[6] T. Tan, T. Van Wouwe, K. F. Werling, C. K. Liu, S. L. Delp, J. L. Hicks, and A. S. Chaudhari, "Gaitdynamics: a generative foundation model for analyzing human walking and running," *Nature Biomedical Engineering*, 2026. [Online]. Available: https://www.nature.com/articles/s41551-025-01565-8

[7] X. Tu, M. Li, M. Liu, J. Si, and H. H. Huang, "A data-driven reinforcement learning solution framework for optimal and adaptive personalization of a hip exoskeleton," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 10 610–10 616.

[8] S. L. Delp, F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, E. Guendelman, and D. G. Thelen, "Opensim: open-source software to create and analyze dynamic simulations of movement," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 11, pp. 1940–1950, 2007.

[9] C. L. Dembia, N. A. Bianco, A. Falisse, J. L. Hicks, and S. L. Delp, "Opensim moco: musculoskeletal optimal control," *PLOS Computational Biology*, vol. 16, no. 12, p. e1008493, 2020.

[10] V. Caggiano, H. Wang, G. Durandau, M. Sartori, and V. Kumar, "Myosuite–a contact-rich simulation suite for musculoskeletal motor control," *arXiv preprint arXiv:2205.13600*, 2022.

[11] 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.

[12] M. Kim, W.-J. Baek, and J. Park, "A deep reinforcement learning based end-to-end control framework for lower limb exoskeletons with smooth movement transitions," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025, pp. 12 005–12 012.

[13] Y. Zhou, C. Chen, J. Zhang, Y. Liu, R. Zhang, Z. Zhang, and X. Wu, "A simulation-based lower-limb exoskeleton control method," in *2025 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. Chengdu, China: IEEE, 2025, pp. 1453–1458.

[14] S. Luo, K. Feng, G. Androwis, S. Adamovich, E. Nunez, H. Su, and X. Zhou, "Robust neural network controller for a lower limb exoskeleton with minimal sensor configuration: a deep reinforcement learning approach with policy distillation," *Journal of Neuroengineering and Rehabilitation*, vol. 23, no. 1, p. 53, 2026.

[15] Y. Yuan, G. Androwis, and X. Zhou, "Smat: Staged multi-agent training for co-adaptive exoskeleton control," *IROS 2026, the IEEE/RSJ International Conference on Intelligent Robots and Systems, arXiv preprint arXiv:2603.07618*, 2026.

[16] Y. Yuan, J. Wolf, G. Androwis, and X. Zhou, "Staged multi-agent training (smat) for hip exoskeletons: Metabolic and biomechanical validation of a simulation-trained co-adaptive controller," *arXiv preprint arXiv:2608.00715*, 2026.

[17] G. Leem, J. Lee, J. Lee, S. Song, and J. Won, "Exo-plore: Exploring exoskeleton control space through human-aligned simulation," *arXiv preprint arXiv:2601.22550*, 2026.

[18] B. Lim, J. Jang, J. Lee, B. Choi, Y. Lee, and Y. Shim, "Delayed output feedback control for gait assistance and resistance using a robotic exoskeleton," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3521–3528, 2019.

[19] H. Barati, S. Kim, N. T. Xuan, J. Lee, and Y. J. Park, "End-to-end policy learning for hip exoskeleton via reinforcement learning and reflex-based musculoskeletal simulation," *IEEE Robotics and Automation Letters*, vol. 11, no. 7, pp. 8672–8679, July 2026.

[20] 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.

[21] I. Park, C. Song, and I. Kang, "Learning hip exoskeleton control policy via predictive neuromusculoskeletal simulation," *arXiv preprint arXiv:2603.04166*, 2026.

[22] 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.

[23] S. Heitmann, N. Ferns, and M. Breakspear, "Muscle co-contraction modulates damping and joint stability in a three-link biomechanical limb," *Frontiers in Neurorobotics*, vol. 5, p. 5, 2012.

[24] B. Koopman, E. H. Van Asseldonk, and H. Van Der Kooij, "Estimation of human hip and knee multi-joint dynamics using the lopes gait trainer," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 920–932, 2016.

[25] G. Ramella, A. Ijspeert, and M. Bouri, "Adaptive negative damping control for user-dependent multi-terrain walking assistance with a hip exoskeleton," in *2025 International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2025, pp. 1340–1346.