nature portfolio

https://doi.org/10.1038/s41586-024-07382-4

Supplementary information

Experiment-free exoskeleton assistance via learning in simulation

In the format provided by the authors and unedited

Supplementary Text 1 | Automatic recognition of gait phase shown by internal representation of the controller

To understand the neural network controller's ability to generate continuous torque assistance during both versatile walking and running without relying on explicit gait phase estimation or activity classification, we analyzed the internal state representation of the learned controller using T-distribution Stochastic Neighbor Embedding (T-SNE)[1]. We visualized the controller's knowledge representation by performing a dimensionality reduction of the activations of each layer through T-SNE (Supplementary Fig. 4). The controller's internal states, which included input joint angles and joint angular velocities and the output joint angles from three subjects' data were used to generate T-distributed stochastic neighbor embeddings. These embeddings were labeled based on the commonly recognized gait phases. This process was performed for both walking and running at varying speeds while using the same trained controller. Each dot in Supplementary Fig. 4 represents a time step. All the dots under each of the three speeds were collected from about 30 seconds of treadmill walking. The heel strike phase accounts for an average of 10% of the gait phase, mid stance phase 24%, late stance phase 18%, swing phase 35%, and other phases 13% of a gait cycle. Other Phases refer to the Foot Flat phase (between Heel Strike and Mid Stance) and the Push-off phase (between Mid Stance and Late Stance). Since these two phases are relatively short, we classify them into Other Phases for brevity. The resulting T-SNEs clearly show that the output joint angles corresponding to similar gait phases are clustered together based on similarity. Internal states clustered in accordance with high-level gait phase features (such as heel strike, stance, and swing phase) for varying walking and running speeds. These observations highlight the ability of the trained controller to not only generate continuous assistive torque without explicit gait phase or activity classification but also adapt to varying speeds during walking and running.

Supplementary Text 2 | Benchmarks for reducing the metabolic rate with robotic assistive devices

In recent years, various benchmarks have been developed to evaluate the performance of robotic assistive devices and to continuously push the boundaries of the capability of these devices. These benchmarks help to ensure that assistive robots are becoming increasingly advanced and effective at assisting individuals with various needs. Measuring the impact on the user's energy expenditure is one of the common ways to validate the effectiveness of exoskeletons on healthy subjects. This measurement provides a quantitative way to assess the level of assistance provided by the exoskeleton in reducing the energy cost of a specific task, such as walking or running. Comparing the metabolic rate of assisted locomotion to that of locomotion without the device provides a stringent evaluation to accurately evaluate the assistive effects of a robotic assistive device. If the metabolic rate is lower with the device's assistance turned on compared to the baseline condition with no device, it can be concluded that the device provides a net positive effect on energy expenditure. There has been a lot of effort put into creating tethered and portable robotic devices to help reduce the energy expenditure required for movement. Tethered exoskeletons allow for the actuation system to be located off the wearer, which can result in a lighter and more comfortable wearable device. Untethered systems, on the other hand, can be worn and used independently without the need for a tether, enabling greater mobility and independence in outdoor environments. Single-joint assistance is more efficient for untethered applications as it reduces the weight of the device, leading to lower metabolic costs. The hip joints are ideal targets for assistance as they contribute the most to positive mechanical work[2,3]. Currently by using a hip exoskeleton, a person can reduce the maximum amount of energy during treadmill walking by 16.6 to 24.9 %[4] (Supplementary Tables 4 and 5). From this study, the normalized metabolic cost relative to the maximum torque is 0.10 ± 0.03 W/Kg/Nm for walking, 0.35 ± 0.12 W/Kg/Nm for running, and 0.23 ± 0.03 W/Kg/Nm for stair climbing.

Supplementary Text 3 | Control policy network training using proximal policy optimization to improve efficiency

Reinforcement learning (RL) involves an agent that interacts with an environment and learns to transit from one state to another by choosing the right actions to achieve assigned goals through trial and error. Based on how the agent evaluates its current situation and how the agent chooses an action, reinforcement learning can be divided into two main approaches, namely, value-based, and policy-based methods. Value-based methods primarily revolve around identifying the implications of being in a state using a value function. Value functions output an expected reward from a given state. A variant of the value function can take into account actions from a given state and therefore can return an expected reward for choosing an action from a given state. Therefore, the agent can get information about its current state without waiting till the end for the final results. The policy in this approach is not explicitly learned and is derived from the value function which simply chooses actions that maximize the return. Policy-based methods explicitly focus on learning the optimal policy thus guiding the agent in choosing the right action in each state. There is no intermediate step in calculating the expected reward for each state-action pair. In this method, policies estimate the probabilities of choosing each action in each state.

Recently, approaches combining both policy and value-based methods have been proposed to leverage the advantages of each individual method. Actor-Critic is an example of such a method, used in our proposed framework, where the actor follows a policy-based approach, and the critic follows a value-based approach. In actor-critic methods, the critic leverages approximation and simulation to learn a value function and guide the optimization of the actor's policy for choosing optimal actions. Lastly, value function and policy are approximated using neural networks. These methods have generalized convergence guarantees and offer faster convergence compared to their individual counterparts.

To ensure data efficiency and stability during the training, the network weights were optimized using the state-of-the-art policy gradient algorithm known as Proximal Policy Optimization (PPO), which is a widely used algorithm for training reinforcement learning agents. As a type of policy gradient algorithm, it updates the parameters of a policy (i.e., the agent's decision-making strategy) in order to maximize the expected total reward. PPO uses a "clip" function to prevent the policy from changing too much from one iteration to the next, which helps stabilize training and prevents the algorithm from getting stuck in poor local optima. Overall, PPO is considered a "state-of-the-art" algorithm for training RL agents, and it has been used to achieve strong results in a wide range of tasks and environments[5]. PPO is known for its stability and high sample efficiency, with a trust region constraint to ensure that the new policy does not deviate too much from the old one, which is defined by:

$$\mathrm{r_t}(\theta) = \frac{\pi_\theta(\mathrm{a_t}|\mathrm{s_t})}{\pi_{\theta\mathrm{old}}(\mathrm{a_t}|\mathrm{s_t})} \tag{1}$$

This probability ratio, also known as the likelihood ratio, compares the probability of the current policy (π_θ) to the probability of the old policy ($\pi_{\theta old}$). A large value of this ratio indicates a significant change in the updated policy compared to the old one. This is used in reinforcement learning algorithms to determine the update step for the policy. PPO prevents excessively large policy updates by using a clip function to restrain the ratio between the old and new policy $r_t(\theta)$ within a prescribed range $[1-\epsilon, 1+\epsilon]$ where ϵ is chosen to be 0.2 in this work. Therefore, the surrogate objective becomes $clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t$ where $\hat{A}_t$ represents the goodness of a certain action. Finally, the objective function was set as the expected value of the lower bound of the unclipped objective function:

$$J^{CLIP}(\theta) = \hat{E}_t\left[\min\left(r_t(\theta)\hat{A}_t\,, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\,\right)\right] \tag{2}$$

The exoskeleton control policy can be updated by maximizing the clipped discounted total reward with a gradient ascent.

Supplementary Text 4 | Controller training environment setup and computer hardware specifications

We used DART (Dynamic Animation and Robotics Toolkit)[6] to simulate the human musculoskeletal model, physical hip exoskeleton, and human-robot interaction in a virtual environment. DART is an open-source library for rigid-body and contact dynamics simulation, which was created by the Graphics Lab and Humanoid Robotics Lab at Georgia Institute of Technology and has been extensively used in the machine learning and robotics research community. DART consists of algorithms and data structures for kinematic and dynamic computation for rigid bodies and animation. DART is widely used because of its high accuracy and stability in forward and hybrid dynamics due to its use of Featherstone's Articulated Body Algorithm using Lie groups and generalized coordinates in geometric notations to represent rigid bodies. Lastly, DART encapsulates a multibody dynamic simulation and a myriad of kinetic tools that facilitate the quick development of robot control approaches. During training, the time integration frequency for the environment was set to 600 Hz and the control frequency (for both exoskeleton and human) was set to 100 Hz. The training and testing are performed on a desktop computer with an Intel® Xeon® CPU E5-1660 v3 @ 3.00GHz × 16. A generic GPU (RTX3090, NVIDIA) with CUDA cores was used to expedite the training which was completed in ~ 8 hours. PyTorch is used to implement neural networks and the PPO method for the learning process. The network weights were initialized using the Xavier uniform initialization method. In total about 20 million samples were generated on the fly during the training. The policy and value networks of the motion imitation network and interaction network are updated at a learning rate of 10^{-4}. The max iteration is set to 120,000. The learning rate of the muscle coordination network is also set to 10^{-4}. Neural network parameters are described in Supplementary Table 1. Hyperparameter settings for training using PPO are shown in Supplementary Table 2.

Supplementary Text 5 | Robustness of controller shown by success rates under parameter perturbations

Transferring a learned controller from a simulation environment to physical robot hardware (sim2real) with minimal loss in accuracy and at the same time adapting the controller to individuals with varying characteristics is imperative and a unique challenge in human-machine systems. Without addressing sim2real challenges such as the discrepancy between model and reality, a controller trained in simulation is prone to failure when deployed in the real world. To enable successful sim-to-real transfer, our proposed framework leverages domain randomization techniques[5] during training where we randomized our hip exoskeleton parameters (e.g., mass, inertia, motor strength, control latency, etc.) to simulate the uncertainties of the real robot. Furthermore, we employed muscle strength randomization during training to account for individual variations where the maximum isometric forces of muscles were randomized. The range of the perturbed parameters is shown in Supplementary Table 3 in the supplementary materials. We evaluated the robustness of the controller towards uncertainties by applying multiple degrees of perturbations in the input state and the trained network parameters.

Through this parameter perturbation experiment, we demonstrate that our controller is robust despite the presence of kinematics perturbation through the simulation experiment. A 20% perturbation means that all the input signals to the exoskeleton control network (i.e., virtual human's joint angle or joint angular velocity) are randomly sampled within [80%, 120%] * nominal value (obtained from the public motion capture dataset) in the simulation. We conducted 200 simulation tests with randomized kinematic perturbation (Supplementary Fig. 5). A simulation test is considered successful if the center of mass of the human-exoskeleton system remains above 80% of its initial height for 10 seconds. If the center of mass falls below this value, it indicates that the virtual human in the simulation may have fallen or tripped, in which case the test is deemed a failure. We then count the number of successful and failed tests to calculate the success rate. When human gaits dramatically deviate from normal walking kinematics (e.g., 80% perturbation), the controller may not be able to provide effective assistance since the success rate is merely above 40% (Supplementary Fig. 5). We simulated up to 80% perturbation but, in reality, it is extremely unlikely that the measured joint angles and angular velocities from IMUs deviate this much away from the nominal values. From this simulation experiment, we demonstrated that our controller is robust to the presence of small kinematics perturbation.

Supplementary Text 6 | Detailed specifications of our lightweight and compliant exoskeleton

In this study, we developed a custom portable lightweight, and compliant exoskeleton for hip flexion and extension assistance (Supplementary Fig. 3). The textile components of the hip exoskeleton consisted of a waist belt, two thigh braces, and two inertial measurement unit (IMU) straps, as shown in Extended Data Fig. 5. The waist belt and thigh braces consisted of woven fabric. Two IMUs (LPMS-B2 Series 9-Axis Inertial Measurement Units, LP-Research Inc., Japan) were secured by elastic bands on the anterior part of each thigh. Two actuators (RMD-X8 1:9, Microactuator Technology Co.) were fixed in the exoskeleton to align with hip joints on both sides. The electronics including a printed circuit board (PCB) and a Li-Po battery (270 g, 22.2 V, 1800 mAh) were placed in the control box which was attached to the back of the waist by the waist belt to power the hip exoskeleton. Our design of a quasi-direct drive actuation-based exoskeleton demonstrates mechanical versatility by being lightweight (3.2 Kg overall mass), compliant (backdrivable), high torque (36 N·m), and having high control bandwidth. The electronic architecture of the exoskeleton facilitated high-level torque control, motor control, sensor signal conditioning, data communication, and power management, as shown in Fig. S3. The low-level controller embedded in the smart actuator could measure the motor motion, current, velocity, and position. The high-level microcontroller is a Teensy 4.1 (PJRC, USA) and implements continuous torque control. The microcontroller acquired joint angles and joint angular velocities from the wireless IMU sensors in real-time through a Raspberry Pi Zero W (Raspberry Pi Foundation, UK). A Bluetooth microcontroller (ItsyBitsy nRF52840 Express, Adafruit, USA) connected to the main controller acted as a transceiver to communicate with a desktop interface for real-time monitoring and data collection. Two IMUs are mounted on the anterior of both left and right thighs and they provide the motion information, including Euler angle, angular velocity, and acceleration at a frequency of 100 Hz. We estimated the hip joint angle based on the change in thigh segment angle relative to the standing position measurement. We performed a torque tracking experiment with our exoskeleton using the reference torque profile obtained from one representative subject during running (2.0 m s^{-1}). The measured torque was estimated from the motor current. The result showed that our robot achieved a small tracking error, with a root-mean-squared error (RMSE) of 0.83 Nm (4.92% of peak torque) (Supplementary Fig. 6). The results indicate that our actuator can accurately deliver the desired assistance torque, which is crucial to providing effective assistance to the wearer.

Supplementary Text 7 | Selection of error range for domain randomization

The error ranges for domain randomization are determined according to common settings in the machine learning field, e.g., [7, 8-10]. For the error range of the robot mass, robot inertia, and actuator peak torque, we randomly sampled within a range of [80%, 120%] of their actual values, which we measured or computed with high confidence. Additionally, we set a range of [50%, 150%] for the control delay in the training process because of more uncertainties in the hardware control response. It is common practice to introduce larger error ranges in simulation, such as 20%, 30%, or even 50%, as part of domain randomization[8-10] since they help to produce a controller that possesses strong robustness and adaptability. For the error range of the human, the maximum isometric force of all lower limb muscles is randomized within [50%, 100%] of their nominal values to account for the variability in muscle strength among individuals. In the study by Thelen et al.[7], it was found that a 30% loss in muscle strength caused over 40% reduction of ankle plantarflexion power output, comparable to 39-44% reductions measured in healthy old adults of approximately 70 years of age. To enhance the adaptability of the controller trained through simulation, we opted for a larger reduction of muscle strength (50%) to cover more variability.

Supplementary Method

Treadmill and stepmill metabolic rate testing protocol. To evaluate the effect on metabolic rate, we tested eight participants while walking at 1.25 m s^{-1} and running at 2.0 m s^{-1} on a level treadmill, stair climbing at 65 steps min^{-1} on a step mill (Supplementary Video 4). First, we measured the resting metabolic rate while participants stood still for 5 min. Then, the participants completed walking, running, or stair climbing without the exoskeleton. Then, the participants completed a warm-up and adaptation period consisting of five minutes of walking, running, or stair climbing with the assistance of the exoskeleton. Participants then completed a walking task at 1.25 m s^{-1}, a running task at 2.0 m s^{-1}, or a stair climbing task under the Assist On, Assist Off, and No Exo conditions. The no-exo conditions were conducted two times at the beginning and the end of the protocol and the average was taken to avoid an order effect. The order of all the other conditions was randomized. Each condition lasted five minutes, and participants were allowed at least a five-minute break between conditions.

Indirect calorimetry (VO2 Master Analyzer, VO2 Master Health Sensors Inc., Canada) was used to measure $\mathrm{O_2}$ consumption for each condition. We calculated the metabolic rate based on $\mathrm{O_2}$ data gathered from the last two minutes of each condition using the Brockway equation[11]. Respiratory output was calculated as:

$$\mathrm{EE} = 16.58 \cdot \mathrm{VO_2} + 4.51 \cdot \mathrm{VCO_2} \tag{3}$$

where EE is the respiratory response in W kg^{-1} and $\mathrm{VO_2}$ and $\mathrm{VCO_2}$ are volumetric flow rates in ml $(kg\cdot min)^{-1}$. Net metabolic cost for each condition was obtained by subtracting the standing metabolic rate from the metabolic rate of each condition. $\mathrm{VCO_2}$ is not measured but is derived as being equal to $\mathrm{VO_2} \cdot \mathrm{RQ}$, where RQ is the respiratory quotient value that is set at 0.85 (the range of RQ in human metabolism is approximately 0.7 to 1.0).

Treadmill torque profile testing protocol. Eight subjects participated in a treadmill testing protocol for slow walking (0.75 m s^{-1}), normal walking (1.25 m s^{-1}), fast walking (1.75 m s^{-1}), running (2 m s^{-1}) to validate the adaptability of the control policy in walking and running at different speeds. Participants walked or ran under each speed while wearing the exoskeleton with the assistance turned on. Each condition lasted one minute, and participants were allowed at least a five-minute break between conditions. The motor current was recorded by the low-level controller embedded in the actuator and used to calculate the output torque:

$$\tau = n \cdot k_t \cdot I \tag{4}$$

where n is the gear reducer ratio, k_t is the torque constant of the motor which equals 2.09 and I is the current of the motor.

Single-participant walk-run-stair transition testing protocol. In the single-subject study, the participant started walking at about 0.8 m s^{-1} and gradually increased the speed to run at about 2 m s^{-1}. Then the subject decelerated before reaching the stairs. Finally, the subject climbed 7 stairs at roughly a constant speed. The controller generated continuous assistive torque which transitioned between activities smoothly, thus demonstrating the controller's ability to handle continuous activity changes. The angular velocity from the thigh IMU and motor output torque were used to calculate the assistance power of the exoskeleton:

$$P = \tau \cdot \omega \tag{5}$$

where τ is the output torque of the actuator and ω is the angular velocity measured by IMU. The mechanical work delivered by the exoskeleton during each step was calculated by the integral of the assistance power with respect to time:

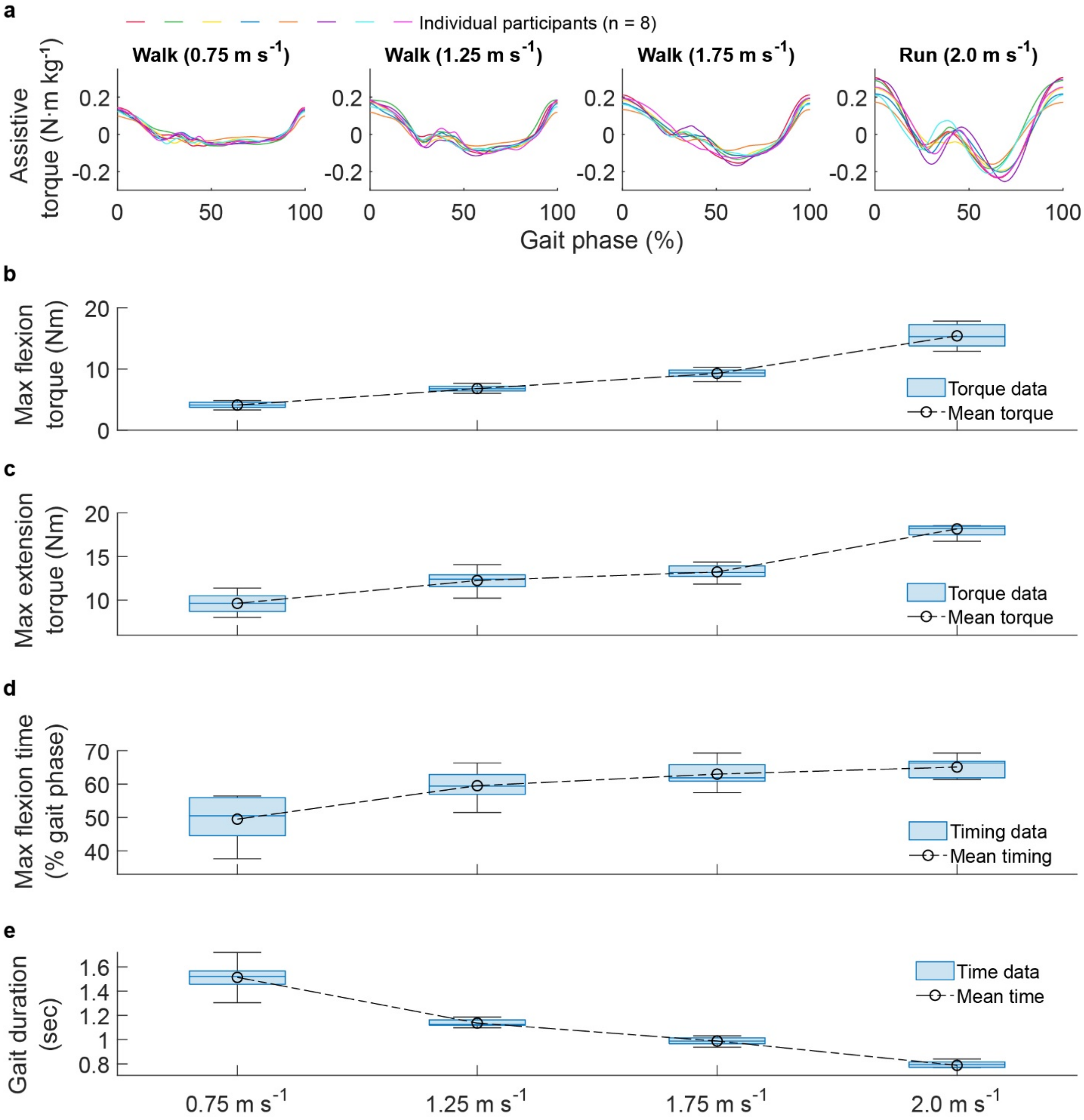
$$W = \int_{t_i}^{t_f} P \ dt \tag{6}$$

where P is the assistance power of the exoskeleton, t_i is the initial time, and t_f is the final time of each step.

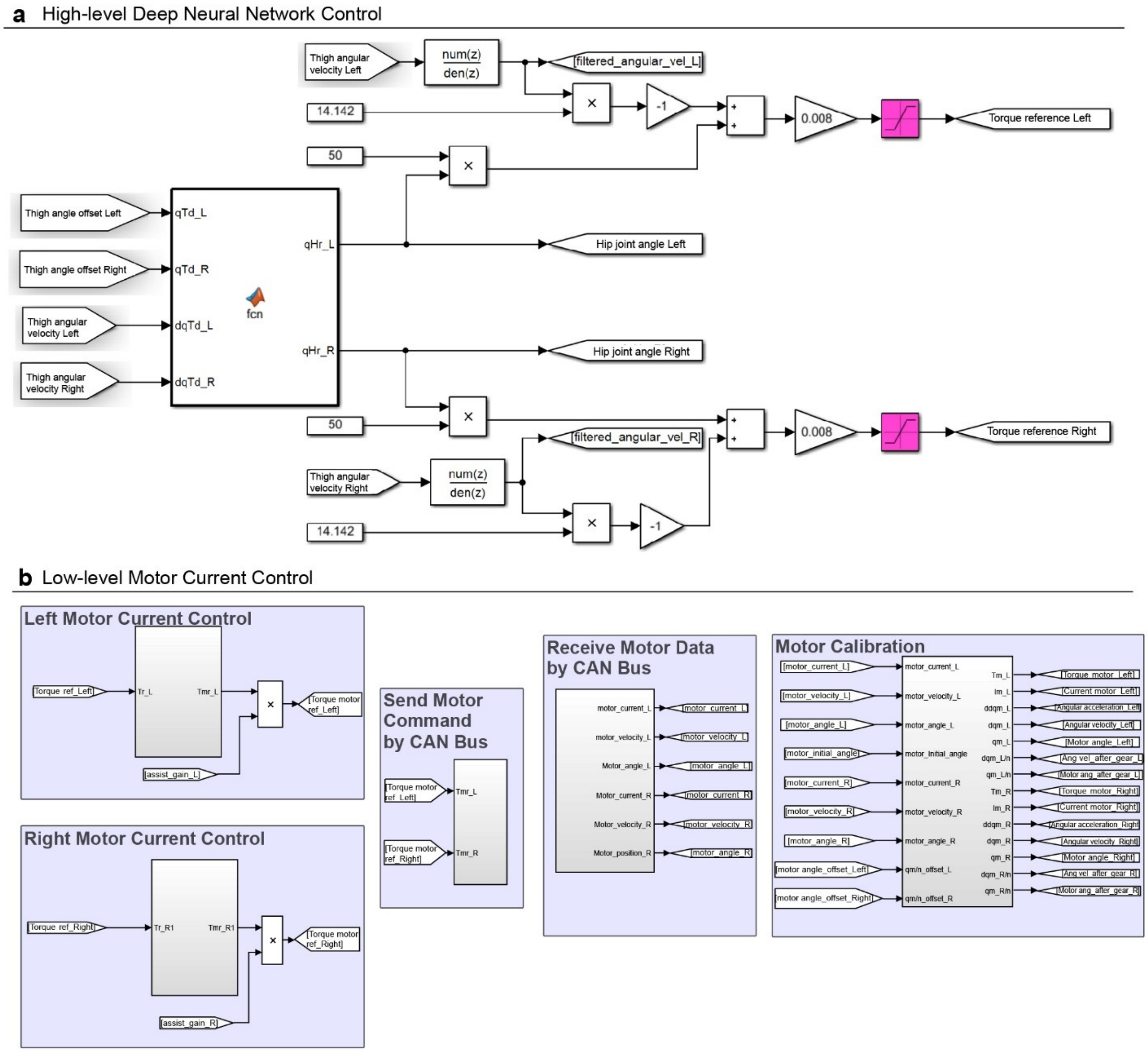
Participants. We recruited participants both with and without previous experience with wearing exoskeletons. The number of participants was based on similar studies in the field[12,13] , and no participants were excluded. Participants were not blinded to the condition they were tested in, and all participants reported no previous musculoskeletal injuries or diseases. Written informed consent was obtained from all participants, and consent for publication of identifiable images was also obtained.

Weight penalty. The weight penalty for an exoskeleton is not only determined by the total mass of the device but also by the distribution of mass on the wearer's body. The metabolic rate was estimated by multiplying the added mass of each component of the exosuit by coefficients found in the literature for the effects of added mass during walking[14] and running[15-17]. Research has shown that the metabolic rate during walking is closely related to the moment of inertia of the legs and that a load located distally (further away from the center of gravity) has a greater impact than a load located proximally (closer to the center of gravity)[14]. Therefore, an exoskeleton that carries a significant portion of its weight on the waist would have a lower weight penalty than one that carries most of its weight on the thighs. This is why the exoskeleton in this study, which has a total weight of 3.2 kg, with 81 % carried at the waist and 19% on the thighs, has a low weight penalty. This information can be found in Supplementary Table 6-7.

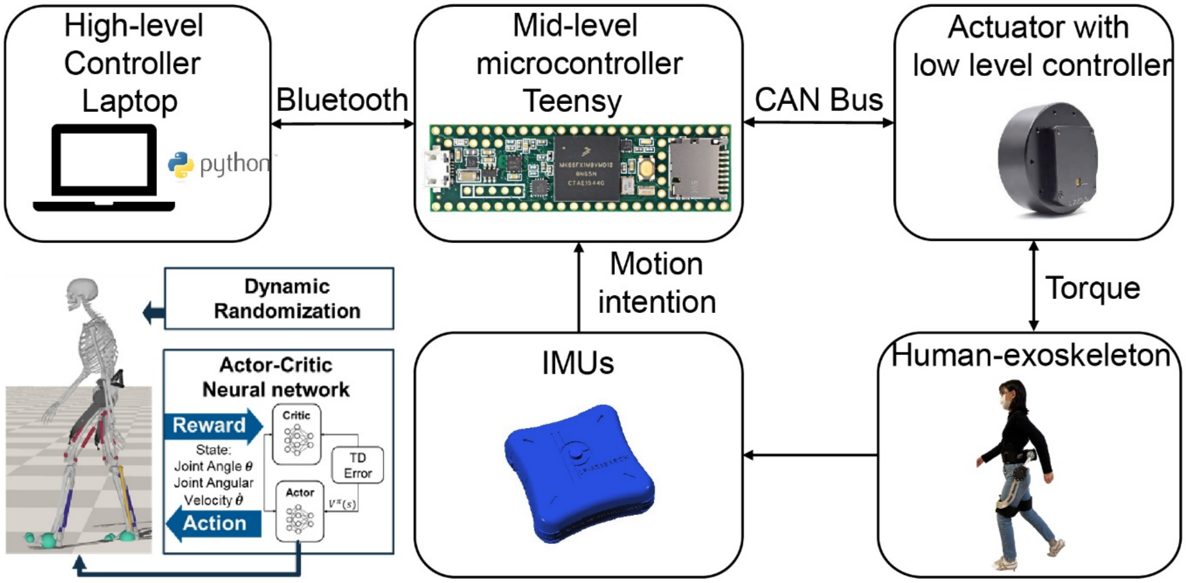
Statistical analysis. We organized the data and conducted statistical analyses (paired *t*-test) in MATLAB (MathWorks, Natick, MA, USA). Mean ± Standard Error of the Mean (SEM) of the net metabolic rate was calculated and reported for all tasks and conditions.



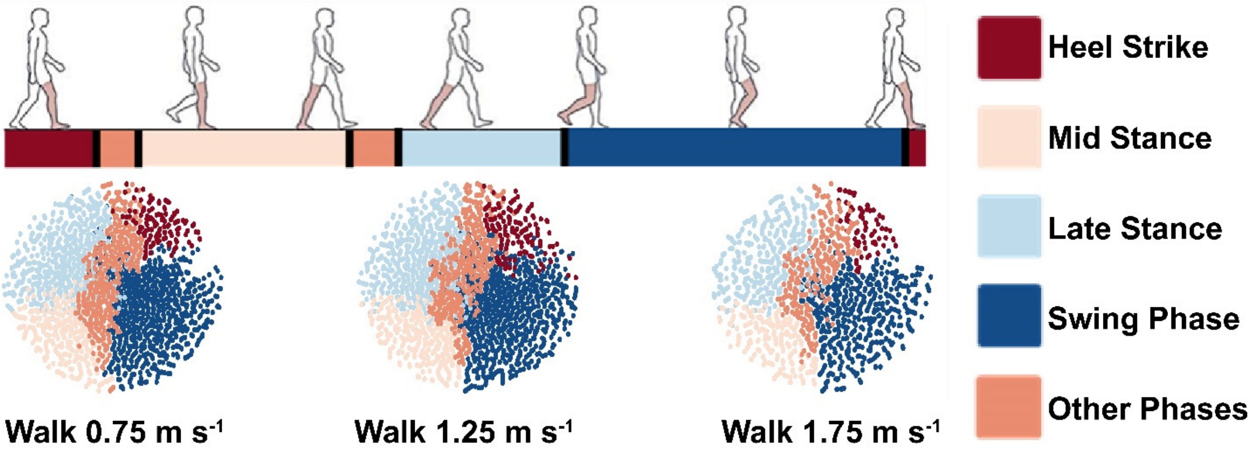
Supplementary Fig. 1 | Assistive torque is generalizable and adaptive to inter-participant kinematic differences at six different speeds during walking and running experiments. a, Assistive torque profile normalized by participant's body mass for walking at 0.75 m s^{-1}, 1.25 m s^{-1} and 1.75 m s^{-1} and running at 2 m s^{-1}. Each line represents the assistance profile of one of the 8 participants averaged across approximately 30 strides. **b**, Maximum flexion torque increased with the locomotion speed. **c**, Timing of the maximum flexion torque increased with the locomotion speed. **d**, Maximum extension torque increased with the walking speed and reduced at running. **e**, Stride time decreased with the locomotion speed. In box plot (b,c,d,e), center lines represent the median, box limits delineate 25th and 75th percentile, and whiskers reflect maximum and minimum values (*n* = 8; individual participants).



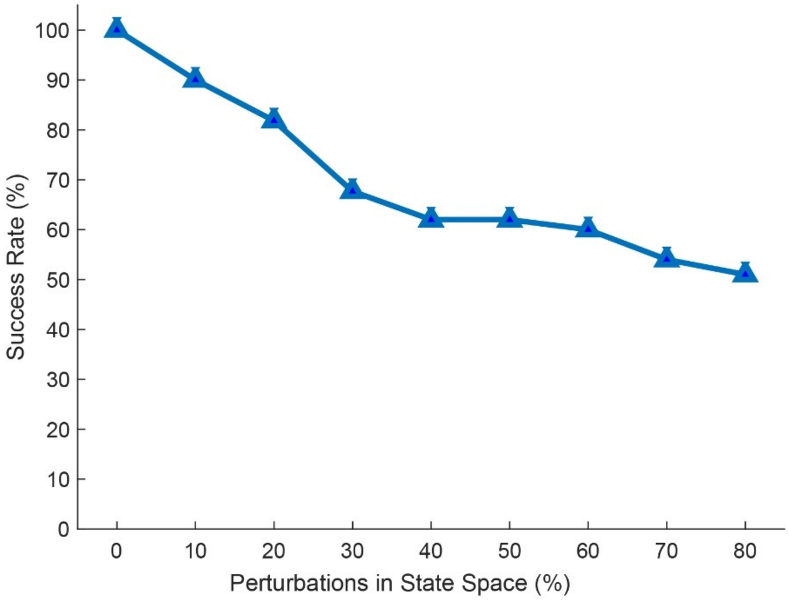
Supplementary Fig. 2 | Exoskeleton control diagram. a,**b**, The exoskeleton was electrically connected to a target PC computer with a Simulink Real-Time system (MathWorks, Natick, USA) running both a high-level neural network control (a) and a low-level motor control module (b).


High-level Controller Laptop
python
Bluetooth
Mid-level microcontroller Teensy
CAN Bus
Actuator with low level controller
Motion intention
Torque
Dynamic Randomization
Actor-Critic Neural network
Reward
Critic
TD Error
State:
Joint Angle θ
Joint Angular Velocity $\dot{\theta}$
Actor
$V^{\pi}(s)$
Action
IMUs
Human-exoskeleton

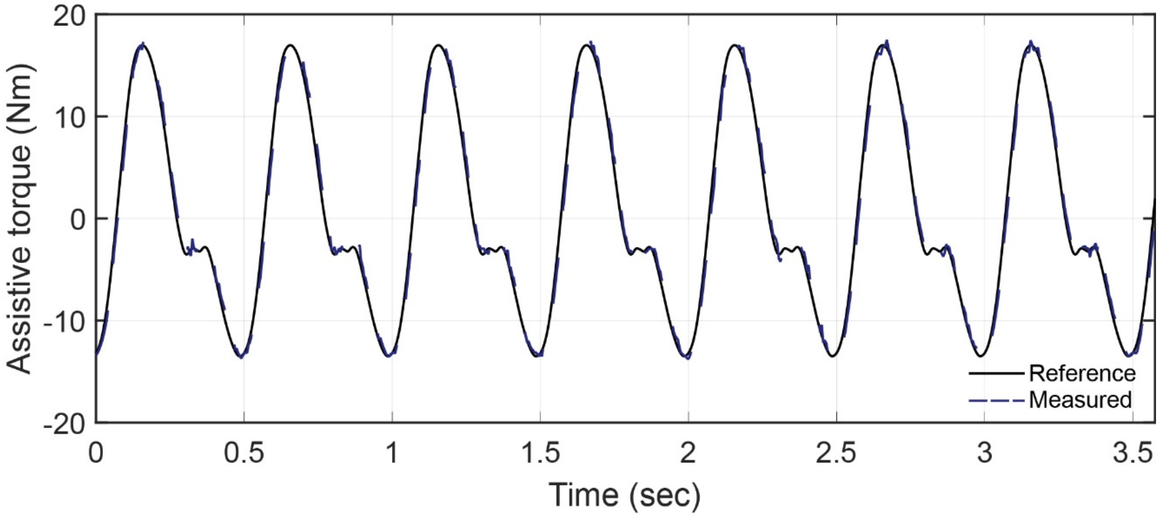
Supplementary Fig. 3 | Exoskeleton electronic hardware architecture. The electronic architecture of the exoskeleton facilitated high-level torque control, motor control, sensor signal conditioning, data communication, and power management. The low-level controller embedded in the smart actuator could measure the motor current, velocity, and position. The high-level microcontroller was a Teensy 4.1 (PJRC, USA) and implemented continuous torque control. The microcontroller acquired joint angle and joint angular velocity from the wireless IMU sensors in real time. A Bluetooth microcontroller (ItsyBitsy nRF52840 Express, Adafruit, USA) connected to the main controller acted as a transceiver to communicate with a desktop interface for real-time monitoring and data collection.



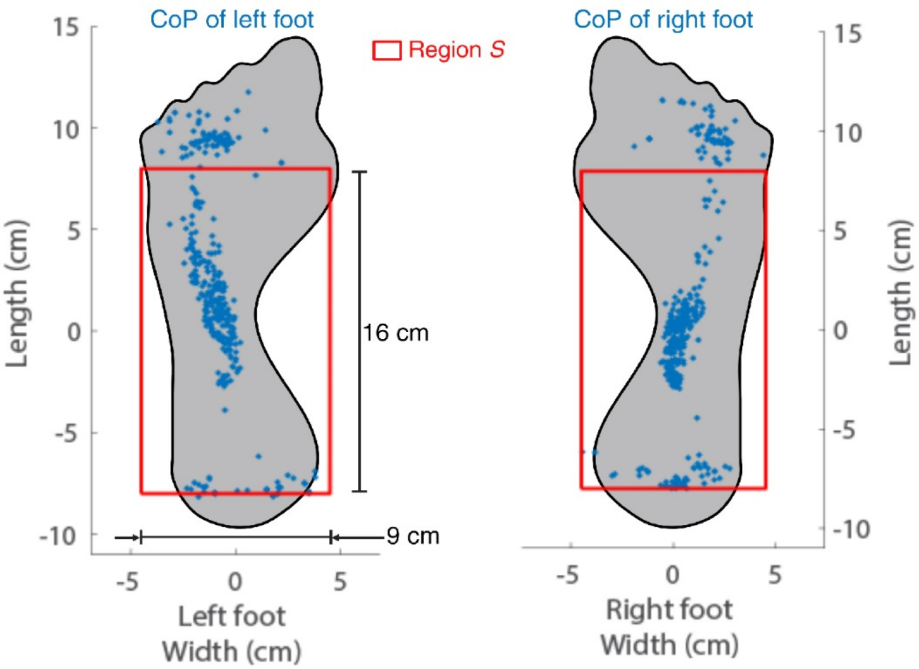
Supplementary Fig. 4 | Representative neural network-based exoskeleton controller's internal knowledge representation analysis of real-time control experiments. T-SNE analysis shows the internal state of the controller, with clusters corresponding to gait phases in walking and running. Each dot represents one internal state (exoskeleton joint angle, angular velocity, and the output of the neural network) and is colored according to the current activity. Color clusters form, indicating that controller states corresponding to the same activities are implicitly grouped together. The neural network-based exoskeleton controller automatically developed localized clusters of states representing distinct events of interest during gait cycles, highlighting there is an underlying structure to the states that were automatically grouped by activity.



Supplementary Fig. 5 | High success rate of our controller in handling uncertainties and disturbances in the network input. The input states were changed in the range from 0 % to 80 % to introduce perturbations. The success rate (in terms of whether the virtual human model can maintain balance in the simulation) decreased from 100 % to 50 % when perturbations in states increased from 0 % to 80 %. This result highlights the overall robustness of our controller to disturbances.



Supplementary Fig. 6 | Torque tracking performance of our hip exoskeleton. Our exoskeleton achieved high torque tracking accuracy with an assistance torque profile for one representative participant running at 2 m s^{-1}, with root mean squared error (RMSE) of 0.83 Nm (4.92% of peak torque). This demonstrates that our exoskeleton is able to accurately deliver the desired torque to the user.



Supplementary Fig. 7 | The body center of pressure (CoP) positions in 10 seconds of walking. Each point in the figure represents the CoP of each foot at different time steps during 10 seconds of walking. These CoP points only exist when the foot is in contact with the ground, and by definition they will always stay within the foot boundary during the foot contact period. A red rectangle area with a width of 9 cm and length of 16 cm is overlayed on each foot. These two rectangle areas denote the region S which is used in the calculation of the balance reward in (9) mainly for the sake of balance stability. In the reward function (6) for the motion imitation network, we assigned a much higher weight ($w^p = 0.75$) for the motion imitation reward than the weight ($w^{cop} = 0.2$) of the CoP-based balance reward. Due to this setting, the virtual human prioritizes tracking the reference motion, and the CoP position is only encouraged to stay within the region S. In fact, the CoP position may leave the region S briefly during each gait cycle as shown in the figure, and this is also to prevent the formation of a stiff or static gait.

Supplementary Table 1 | Neural network parameters

Neural network	Input dimension	Number of hidden layers	Number of neurons	Output dimension
Motion Imitation Network (Actor)	(50,2)	2	[256, 256]	(50,1)
Motion Imitation Network (Critic)	(50,2)	2	[256, 256]	(1,1)
Muscle Coordination Network	(50,1)	3	[512, 256, 256]	(208, 1)
Exoskeleton Control Network (Actor)	(18,1)	2	[128, 64]	(2,1)
Exoskeleton Control Network (Critic)	(18,1)	2	[128, 64]	(1,1)

Interaction	Translation Stiffness	Translation Damping	Rotation Stiffness	Rotation Damping
Human pelvis and exoskeleton waist belt	$k^y = 8000$	$c^y = 10$	N/A	N/A
Human legs and exoskeleton leg straps	$k^x = 1500,$ $k^y = 500,$ $k^z = 1500$	$c^x = 10,$ $c^y = 1,$ $c^z = 10$	$a^x = 10,$ $a^y = 3,$ $a^z = 10$	$\beta^x = 1,$ $\beta^y = 0.1,$ $\beta^z = 1$

Supplementary Table 2 | Hyperparameters for control policy training

Hyper-parameter	**Value**	**Hyper-parameter**	**Value**
Discount Factor	0.99	**Epochs**	10
Learning Rate	10^{-4}	**Clip Threshold**	0.2
Batch Size	128	**Memory Buffer**	2048
Epochs (Muscle coordination network)	3	**Max Iterations**	120,000

Supplementary Table 3 | Parameters spaces for the domain randomizations

Dynamic parameters	Training range
Mass of exoskeleton	[0.8, 1,2]*default value
Inertia of exoskeleton	[0.8, 1,2]*default value
Motor strength of exoskeleton	[0.8, 1.2]*default value
Control latency of the exoskeleton	[0.5, 1,5]*default value
Friction coefficient of the environment	[0.9, 1,6]*default value
Muscle isometric forces	[0.5, 1,5]*default value

Supplementary Table 4 | Benchmark with state-of-the-art portable hip exoskeletons

Ref.	Assistance	Peak torque/fo rce	Device mass (kg)	Activities	Comparison conditions	Metabolic reduction (%)
Seo16[18]	Flexion & Extension	9.8 Nm	2.8	Walking	Assist On vs. No Exo	13.2
Y Lee17[19]	Flexion & Extension	12 Nm	2.6	Walking	Assist On vs. No Exo	13.2
H Lee17[20]	Flexion & Extension	N/A	2.8	Walking	Assist On vs. No Exo	7
Nasiri18[21]	Unpowered	N/A	1.8	Running	Assist On vs. No Exo	8
D Kim18[22]	Flexion & Extension	12 Nm	2.8	Stair climbing	Assist On vs. No Exo	10.16
J Kim18[23]	Extension	300 N	4.7	Walking & Running	Assist On vs. No Exo	2.7 (walk) 3.9 (run)
J Kim19[12]	Extension	N/A	5	Walking & Running	Assist On vs. No Exo	9.3 (walk) 4 (run)
Panizzolo19[24]	Extension	0.05 Nm kg^{-1}	0.65	Walking	Assist On vs. No Exo	3.3
Cao20[25]	Flexion & Extension	150 N	4	Walking	Assist On vs. No Exo	13.05
Gordon22[26]	Flexion & Extension	15 Nm	6.8	Walking	Assist On vs. No Exo	10
Zhang22[27]	Flexion	12 Nm	2.2	Walking	Assist On vs. Assist Off	10
J Kim22[28]	Flexion	N/A	2.31	Walking	Assist On vs. No Exo	7.2
Ishmael21[29]	Flexion & Extension	45 Nm	2.45	Walking (above-knee amputation)	Assist On vs. No Exo	15.6
This work	**Flexion & Extension**	**14 Nm, 18 Nm & 16 Nm**	**3.2**	**Walking, Running & Stair climbing**	Assist On vs. No Exo	**24.3 (walk) 13.1 (run) 15.4 (stair)**

Supplementary Table 5 | Benchmark with state-of-the-art lower-limb exoskeletons

Joint	Activity	Device mass (kg)	Tethered /portable	Assistance	Comparison conditions	Metabolic reduction (%)	Note	Ref.
Ankle	Walking	2.7	Portable	Plantarflexion	Assist On vs. No Exo	23		Slade22 [30]
	Walking	2.7	Portable	Plantarflexion	Assist On vs. No Exo	17	Outdoor	Slade22 [30]
	Walking	4	Portable	Plantarflexion	Assist On vs. No Exo	8	23 kg load	Mooney14 [31]
	Walking	3.6	Portable	Plantarflexion	Assist On vs. No Exo	10		Mooney14 [32]
	Walking	3.6	Portable	Plantarflexion	Assist On vs. No Exo	11		Mooney16 [33]
	Walking	13.5	Tethered	Plantarflexion	Assist On vs. Assist Off	30		Franks21 [34]
	Walking	0.88	Tethered	Plantarflexion	Assist On vs. Assist Off	39		Poggensee21 [35]
	Walking	0.83	Tethered	Plantarflexion	Assist On vs. Assist Off	24.2		Zhang17 [13]
	Running	0.88	Tethered	Plantarflexion & Dorsiflexion	Assist On vs. No Exo	14.6		Witte20 [36]
	Running	1.1	Tethered	Plantarflexion	Assist On vs. Assist Off	24.8		Miller22 [37]
Hip	Walking	2.8	Portable	Flexion & Extension	Assist On vs. No Exo	13.2		Seo16 [18]
	Walking	2.8	Portable	Flexion & Extension	Assist On vs. No Exo	7	Elderly	H Lee17 [20]
	Walking	2.6	Portable	Flexion & Extension	Assist On vs. No Exo	13.2		Y Lee17 [19]
	Walking	2.4	Portable	Flexion & Extension	Assist On vs. No Exo	15.5	5% slope	Seo17 [38]
	Stair climbing	2.8	Portable	Flexion & Extension	Assist On vs. No Exo	10.16	Elderly	D Kim18 [22]
	Running	4.7	Portable	Extension	Assist On vs. No Exo	3.9		J Kim18 [23]
	Walking & Running	5	Portable	Extension	Assist On vs. No Exo	9.3 & 4		J Kim19 [12]
	Walking	2.1	Portable	Flexion & Extension	Assist On vs. No Exo	19.8		Lim19 [4]
	Walking	4	Portable	Flexion & Extension	Assist On vs. No Exo	13.05		Cao20 [25]
	Walking	6.8	Portable	Flexion & Extension	Assist On vs. No Exo	10		Gordon22 [26]
	Walking	2.31	Portable	Flexion	Assist On vs. No Exo	7.2		J Kim22 [28]
	Walking	2.2	Portable	Flexion	Assist On vs. Assist Off	10		Zhang22 [27]
	Walking	13.5	Tethered	Flexion & Extension	Assist On vs. Assist Off	26		Franks21[34]

Hip-Knee	Inclined walking, level walking	9.98	Portable	Flexion & Extension	Assist On vs. No Exo	8.8%		Seo18 [39]
Knee	Walking	4.85	Portable	Flexion & Extension	Assist On vs. No Exo	-0.07	12 % slope	Choi19 [40]
	Walking	8.4	Portable	Flexion & Extension	Assist On vs. No Exo	4.2	15 % slope, 18.1 kg load	McLean19 [41]
	Walking	2.7	Portable	Extension	Assist On vs. Assist Off	0.6	Unilateral 15 % slope	D Lee20 [42]
	Walking	1.1	Portable	Flexion	Assist On vs. No Exo	2.5		Shepertycky21[43]
	Walking	13.5	Tethered	Flexion & Extension	Assist On vs. Assist Off	13		Franks21[34]
Hip	**Walking, Running & Stair climbing**	**3.2**	Portable	**Flexion & Extension**	Assist On vs. No Exo	**24.3, 13.1 & 15.4**		**This study**

Supplementary Table 6 | Exoskeleton mass breakdown and weight penalty

	Bilateral added mass m (kg)		**Metabolic penalty coefficients β (W/kg)**	
			Walking (1.5 m s^{-1})	**Running (2.5 m s^{-1})**
Waist	Actuator x 2	1.420		
	Waist belt	0.252		
	Control box	0.401		
	Battery & electronics	0.548		
	Waist subtotal	**2.621**	4.005[44]	5.737[45]
Thighs	Thigh strap x 2	0.193		
	Thigh frame x 2	0.388		
	Thigh IMU x 2	0.024		
	Bilateral thighs subtotal	**0.605**	6.674[44]	12.09[46]
Shanks		0	6.763[44]	29.57[46]
Feet		0	17.80[44]	42.26[47]
	Total weight of hip exo	**3.226**		
Expected metabolic penalty (W) $\sum m \times \beta$			**14.53**	**22.35**

Supplementary Table 7 | Metabolic penalty test results

Participant	Walking (Watt/kg)		Running (Watt/kg)		Stair climbing (Watt/kg)	
	Assist Off	No Exo	Assist Off	No Exo	Assist Off	No Exo
1	2.33	2.29	8.40	7.85	6.43	5.81
2	2.90	2.76	5.32	4.90	5.24	4.25
3	3.47	3.23	5.81	5.21	6.09	5.73
4	4.39	4.12	11.45	11.32	6.53	6.29
5	2.84	2.76	7.50	7.07	5.83	5.14
6	3.83	3.63				
7	1.95	1.77				
8	2.83	2.73				
9			8.75	7.67	6.05	5.60
10			11.07	10.70	5.54	5.20
11			11.56	11.36	7.03	6.38
Mean	3.07	2.91	8.73	8.26	6.09	5.55
SEM	0.28	0.26	0.88	0.92	0.20	0.24
Penalty (%)	**5.42**		**7.49**		**10.28**	

Supplementary Table 8 | Summary of notation

Notation	Description
q	Generalized coordinates
$M(q)$	Mass matrix
$c(q, \dot{q})$	Coriolis and gravitational forces
f_m	Muscle force
f_c	Constraint force
J_m, J_c	Jacobian matrices that map forces to generalized coordinates
f_{ext}	External force
f	Muscle tension
l	Muscle length
a	Level of muscle activation
f_l	Force-length function
f_v	Force-velocity function
f_p	Passive force
π^*	Expected cumulative rewards
r	Reward function
w	Weights of reward
k_p, k_v	Gains of proportional and derivative (PD) control
u	Action (assistive torque) that the agent takes

Supplementary References

1 Hinton, G. E. & Roweis, S. Stochastic neighbor embedding. *Advances in Neural Information Processing Systems* **15** (2002).

2 Sawicki, G. S., Lewis, C. L. & Ferris, D. P. It pays to have a spring in your step. *Exercise and Sport Sciences Reviews* **37**, 130 (2009).

3 Farris, D. J. & Sawicki, G. S. The mechanics and energetics of human walking and running: a joint level perspective. *Journal of The Royal Society Interface* **9**, 110-118 (2012).

4 Lim, B. *et al.* Delayed output feedback control for gait assistance with a robotic hip exoskeleton. *IEEE Transactions on Robotics* **35**, 1055-1062 (2019).

5 Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

6 Lee, J., X. Grey, M., Ha, S., Kunz, T., Jain, S., Ye, Y., S. Srinivasa, S., Stilman, M. and Karen Liu, C., 2018. DART: Dynamic animation and robotics toolkit. *The Journal of Open Source Software*, *3*(22), p.500.

7 Thelen, D. G. Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults. *Journal of Biomechanical Engineering*. **125**, 70-77 (2003).

8 Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V. and Hutter, M., 2020. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, **5**(47), p.eabc5986.

9 Tan, J., Zhang, T., Coumans, E., Iscen, A., Bai, Y., Hafner, D., Bohez, S. and Vanhoucke, V., 2018. Sim-to-real: Learning agile locomotion for quadruped robots. arXiv preprint arXiv:1804.10332.

10 Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W. and Abbeel, P., 2017, September. Domain randomization for transferring deep neural networks from simulation to the real world. In 2017 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 23-30). IEEE.

11 Brockway, J. Derivation of formulae used to calculate energy expenditure in man. *Human nutrition. Clinical Nutrition* **41**, 463-471 (1987).

12 Kim, J. *et al.* Reducing the metabolic rate of walking and running with a versatile, portable exosuit. *Science* **365**, 668-672 (2019).

13 Zhang, J. *et al.* Human-in-the-loop optimization of exoskeleton assistance during walking. *Science* **356**, 1280-1284 (2017).

14 Browning, R. C., Modica, J. R., Kram, R. & Goswami, A. The effects of adding mass to the legs on the energetics and biomechanics of walking. *Medicine & Science in Sports & Exercise* **39**, 515-525 (2007).

15 Sovero, S., Talele, N., Smith, C., Cox, N., Swift, T. and Byl, K., 2016, October. Initial data and theory for a high specific-power ankle exoskeleton device. In *International Symposium on Experimental Robotics* (pp. 355-364). Cham: Springer International Publishing.

16 Franz, J. R., Wierzbinski, C. M. & Kram, R. Metabolic cost of running barefoot versus shod: is lighter better? *Medicine & Science in Sports & Exercise* **44**, 1519-1525 (2012).

17 Cureton, K. *et al.* Effect of experimental alterations in excess weight on aerobic capacity and distance running performance. *Medicine and Science in Sports* **10**, 194-199 (1978).

18 Seo, K., Lee, J., Lee, Y., Ha, T. and Shim, Y., 2016, May. Fully autonomous hip exoskeleton saves metabolic cost of walking. In *2016 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 4628-4635). IEEE.

19 Lee, Y., Roh, S.G., Lee, M., Choi, B., Lee, J., Kim, J., Choi, H., Shim, Y. and Kim, Y.J., 2017, September. A flexible exoskeleton for hip assistance. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1058-1063). IEEE.

20 Lee, H.-J. *et al.* A wearable hip assist robot can improve gait function and cardiopulmonary metabolic efficiency in elderly adults. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **25**, 1549-1557 (2017).

21 Nasiri, R., Ahmadi, A. & Ahmadabadi, M. N. Reducing the energy cost of human running using an unpowered exoskeleton. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **26**, 2026-2032 (2018).

22 Kim, D.-S. *et al.* A wearable hip-assist robot reduces the cardiopulmonary metabolic energy expenditure during stair ascent in elderly adults: a pilot cross-sectional study. *BMC Geriatrics* **18**, 1-8 (2018).

23 Kim, J., Heimgartner, R., Lee, G., Karavas, N., Perry, D., Ryan, D.L., Eckert-Erdheim, A., Murphy, P., Choe, D.K., Galiana, I. and Walsh, C.J., 2018, May. Autonomous and portable soft exosuit for hip extension assistance with online walking and running detection algorithm. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 5473-5480). IEEE.

24 Panizzolo, F. A., Bolgiani, C., Di Liddo, L., Annese, E. & Marcolin, G. Reducing the energy cost of walking in older adults using a passive hip flexion device. *Journal of Neuroengineering and Rehabilitation* **16**, 1-9 (2019).

25 Cao, W., Chen, C., Hu, H., Fang, K. & Wu, X. Effect of hip assistance modes on metabolic cost of walking with a soft exoskeleton. *IEEE Transactions on Automation Science and Engineering* **18**, 426-436 (2020).

26 Gordon, D. F., McGreavy, C., Christou, A. & Vijayakumar, S. Human-in-the-Loop Optimization of Exoskeleton Assistance Via Online Simulation of Metabolic Cost. *IEEE Transactions on Robotics* **38**, 1410-1429 (2022).

27 Zhang, X. *et al.* Enhancing gait assistance control robustness of a hip exosuit by means of machine learning. *IEEE Robotics and Automation Letters* **7**, 7566-7573 (2022).

28 Kim, J. *et al.* Reducing the energy cost of walking with low assistance levels through optimized hip flexion assistance from a soft exosuit. *Scientific Reports* **12**, 11004 (2022).

29 Ishmael, M.K., Archangeli, D. and Lenzi, T., 2021. Powered hip exoskeleton improves walking economy in individuals with above-knee amputation. *Nature Medicine*, ***27***(10), pp.1783-1788.

30 Slade, P., Kochenderfer, M. J., Delp, S. L. & Collins, S. H. Personalizing exoskeleton assistance while walking in the real world. *Nature* **610**, 277-282 (2022).

31 Mooney, L. M., Rouse, E. J. & Herr, H. M. Autonomous exoskeleton reduces metabolic cost of human walking during load carriage. *Journal of Neuroengineering and Rehabilitation* **11**, 1-11 (2014).

32 Mooney, L. M., Rouse, E. J. & Herr, H. M. Autonomous exoskeleton reduces metabolic cost of human walking. *Journal of NeuroEngineering and Rehabilitation* **11**, 151, doi:10.1186/1743-0003-11-151 (2014).

33 Mooney, L. M. & Herr, H. M. Biomechanical walking mechanisms underlying the metabolic reduction caused by an autonomous exoskeleton. *Journal of Neuroengineering and Rehabilitation* **13**, 1-12 (2016).

34 Franks, P.W., Bryan, G.M., Martin, R.M., Reyes, R., Lakmazaheri, A.C. and Collins, S.H., 2021. Comparing optimized exoskeleton assistance of the hip, knee, and ankle in single and multi-joint configurations. *Wearable Technologies*, *2*, p.e16.

35 Poggensee, K.L. and Collins, S.H., 2021. How adaptation, training, and customization contribute to benefits from exoskeleton assistance. *Science Robotics*, *6*(58), p.eabf1078.

36 Witte, K.A., Fiers, P., Sheets-Singer, A.L. and Collins, S.H., 2020. Improving the energy economy of human running with powered and unpowered ankle exoskeleton assistance. *Science Robotics*, *5*(40), p.eaay9108.

37 Miller, D.E., Tan, G.R., Farina, E.M., Sheets-Singer, A.L. and Collins, S.H., 2022. Characterizing the relationship between peak assistance torque and metabolic cost reduction during running with ankle exoskeletons. *Journal of neuroengineering and rehabilitation*, *19*(1), p.46.

38 Seo, K., Lee, J. and Park, Y.J., 2017, July. Autonomous hip exoskeleton saves metabolic cost of walking uphill. In *2017 International Conference on Rehabilitation Robotics (ICORR)* (pp. 246-251). IEEE.

39 Seo, K., Kim, K., Park, Y.J., Cho, J.K., Lee, J., Choi, B., Lim, B., Lee, Y. and Shim, Y., 2018, May. Adaptive oscillator-based control for active lower-limb exoskeleton and its metabolic impact. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (pp. 6752-6758). IEEE.

40 Choi, B., Lee, Y., Lee, J., Lee, M., Lim, B., Park, Y.J., Kim, K., Kim, Y.J. and Shim, Y., 2019, November. Development of adjustable knee assist device for wearable robot based on linkage and rolling joint. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4043-4050). IEEE.

41 MacLean, M. K. & Ferris, D. P. Energetics of walking with a robotic knee exoskeleton. *Journal of Applied Biomechanics* **35**, 320-326 (2019).

42 Lee, D., Kwak, E. C., McLain, B. J., Kang, I. & Young, A. J. Effects of assistance during early stance phase using a robotic knee orthosis on energetics, muscle activity, and joint mechanics during

incline and decline walking. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **28**, 914-923 (2020).

43 Shepertycky, M., Burton, S., Dickson, A., Liu, Y.-F. & Li, Q. Removing energy with an exoskeleton reduces the metabolic cost of walking. *Science* **372**, 957-960 (2021).

44 Browning, R.C., Modica, J.R., Kram, R. and Goswami, A., 2007. The effects of adding mass to the legs on the energetics and biomechanics of walking. *Medicine & Science in Sports & Exercise*, *39*(3), pp.515-525.

45 Cureton, K.J., Sparling, P.B., Evans, B.W., Johnson, S.M., Kong, U.D. and Purvis, J.W., 1978. Effect of experimental alterations in excess weight on aerobic capacity and distance running performance. *Medicine and science in sports*, *10*(3), pp.194-199.

46 Franz, J.R., Wierzbinski, C.M. and Kram, R., 2012. Metabolic cost of running barefoot versus shod: is lighter better?. *Medicine & Science in Sports & Exercise*, *44*(8), pp.1519-1525.

47 Sovero, S., Talele, N., Smith, C., Cox, N., Swift, T. and Byl, K., 2016, October. Initial data and theory for a high specific-power ankle exoskeleton device. In *International Symposium on Experimental Robotics* (pp. 355-364). Cham: Springer International Publishing.

Supplementary Video Caption

Supplementary Video 1 | We present a control approach that learns assistive control strategies in simulation and can be transferred to a physical wearable robot to generate continuous assistance for multiple locomotion activities, including walking, running, and stair climbing.

Supplementary Video 2 | Human response to robots is slow and controller development typically requires 24-60 min human testing and is limited to single activity (mostly walking) control only. Our method leverages dynamics-aware and data-driven simulation. It requires no human testing and can be immediately deployed to a physical exoskeleton for multiple activities. We only need simulation once for 8 hours to learn the assistive control policy that is transferred to micro-controllers for real-time control of physical exoskeletons.

Supplementary Video 3 | The learned controller in simulation is transferred to a physical exoskeleton for real-time control that immediately improves mobility. Using control policies trained in simulation, the controller is versatile to assist multiple locomotion modes and leads to significant metabolic expenditure savings by 24.3 %, 13.1 %, and 15.4 % during walking, running, and stair climbing, respectively, compared with no exoskeleton conditions.

Supplementary Video 4 | The robot learns control strategies by simultaneous training of muscle-coordination neural network and robot controller neural network. It also learns multi-locomotion control by an activity imitation neural network. We bridge the sim-to-real gap by domain randomization of muscle models and robot parameters, and the control policy only requires one wearable sensor per leg to control the exoskeleton.

Supplementary Video 5 | The robot donning takes about 2 mins and doffing takes less than 1 min. The 3.2 kg low-profile hip exoskeleton does not affect the natural range of motion and thus can assist various movements for heterogeneous able-bodied individuals.