EXOSKELETONS

User preference optimization for control of ankle exoskeletons using sample efficient active learning

Ung Hee Lee[1,2,3], Varun S. Shetty[1,2], Patrick W. Franks[3], Jie Tan[4], Georgios Evangelopoulos[3], Sehoon Ha[4,5], Elliott J. Rouse[1,2,3]*

One challenge to achieving widespread success of augmentative exoskeletons is accurately adjusting the controller to provide cooperative assistance with their wearer. Often, the controller parameters are "tuned" to optimize a physiological or biomechanical objective. However, these approaches are resource intensive, while typically only enabling optimization of a single objective. In reality, the exoskeleton user experience is likely derived from many factors, including comfort, fatigue, and stability, among others. This work introduces an approach to conveniently tune the four parameters of an exoskeleton controller to maximize user preference. Our overarching strategy is to leverage the wearer to internally balance the experiential factors of wearing the system. We used an evolutionary algorithm to recommend potential parameters, which were ranked by a neural network that was pretrained with previously collected user preference data. The controller parameters that had the highest preference ranking were provided to the exoskeleton, and the wearer responded with real-time feedback as a forced-choice comparison. Our approach was able to converge on controller parameters preferred by the wearer with an accuracy of 88% on average when compared with randomly generated parameters. User-preferred settings stabilized in 43 ± 7 queries. This work demonstrates that user preference can be leveraged to tune a partial-assist ankle exoskeleton in real time using a simple, intuitive interface, highlighting the potential for translating lower-limb wearable technologies into our daily lives.

INTRODUCTION

Wearable robots, including exoskeletons and powered prostheses, hold the potential to change human mobility (*1*–*4*). Millions of people suffer from mobility deficits driven by lost mechanical effort from the human neuromotor system, often stemming from factors such as aging, weakened muscles, stroke, or limb loss (*5*, *6*). These individuals are likely to walk slower, fatigue more easily, and fall more often, which can lead to secondary conditions including obesity and depression (*6*–*9*). Robotic assistive technology could potentially address these impairments by assisting the user's gait, thereby offloading the demands on the neuromotor system (*2*, *10*, *11*). Although advancements in actuation, computation, and machine learning have enabled the development of a plethora of wearable robots, these technologies have not become commonplace partly because of the challenges in designing and tuning their control strategies (*12*–*14*).

An important challenge in the control of wearable robots is adjusting their behavior to meet the goals of a specific, individual user. Modern control strategies typically include a set of parameters that define instructions that mathematically describe the provided assistance. These parameters are adjusted to optimize the assistance for an individual user or a particular activity (*13*) often using one of two methods: expert-based tuning to mimic natural human locomotion (*15*) or automatic tuning based on metabolic rate or other physiological objectives (*16*, *17*). Although these approaches can be successful, they are also limited in that they typically focus on a single objective (biomechanics, speed, or metabolic rate) (*15*, *16*, *18*). In reality, there are many factors that may influence the user's experience, and these single objective–based tuning approaches cannot capture these varied aspects (*19*). Multi-objective optimization for tuning control parameters has been investigated (*20*, *21*), but it is not yet clear how to effectively choose or design these objectives.

In the development of assistive technologies, user preference has always held an informal role; however, recently, researchers have begun to formalize its use in the design and tuning of these systems (*19*, *22*–*25*). User preference is a promising objective because it enables the wearer to adjust settings by synthesizing the multifactorial nature of their experience. That is, when selecting their preferred assistance settings, users may simultaneously consider multiple physiological or biomechanical objectives, including comfort, balance, fatigue, stability, and exertion, among others. Previous research has begun to investigate the role of user preference in wearable robotic systems and has shown that user-preferred assistance settings are reliable and unique (*19*, *24*). In addition, previous work has leveraged user-driven tuning approaches to identify user preference for controlling exoskeletons and prostheses (*19*, *22*, *26*). However, because these approaches rely only on the user to explore a low-dimensional controller parameter space (the set of available tuning parameters), they may be impractical to use outside the laboratory and challenging to scale to higher-dimensional parameter spaces.

To address limitations associated with user-driven tuning, automatic tuning methods that implement active learning of user preference have been explored. In this work, we use "automatic" to denote the algorithm searching a parameter space instead of the user directly searching the space of potential control strategies. Active learning is a branch of machine learning in which learning

[1]Department of Mechanical Engineering, University of Michigan, 2350 Hayward, Ann Arbor, MI 48109, USA. [2]Department of Robotics, University of Michigan, 2505 Hayward, Ann Arbor, MI 48109, USA. [3]X, the Moonshot Factory, 100 Mayfield Ave., Mountain View, CA 94043, USA. [4]Robotics at Google, 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA. [5]Georgia Institute of Technology, 85 Fifth Street NW, Atlanta, GA 30308, USA.
*Corresponding author. Email: ejrouse@umich.edu

algorithms work with the user to more efficiently explore the parameter space. The learning algorithm selects a subset of data from the larger pool of unlabeled data within the parameter space and queries the user to label the data. Using this method, the algorithm effectively chooses a subset of data to be labeled and used as training data, which ultimately reduces the number of labels needed overall to train the algorithm (*27*). This is especially attractive for cases where labels are costly and time-consuming [human-robot interaction (*28*)] to obtain. Active learning of user preference has shown promising results in a number of different robotic domains (*29*–*33*). For applications in exoskeletons, Bayesian optimization with Gaussian processes has been used to adjust parameters on the basis of user preference (*23*). These works primarily used complete-assist exoskeletons (e.g., Atalante, Wandercraft) where the robot drives the full motion of the wearer (*23*, *34*, *35*). Thus, it is unclear how previous approaches would apply to devices that work in tandem with their users (partial-assist systems), including exoskeletons and prostheses.

The objective of this work is to develop an optimization strategy for tuning higher-dimensional control strategies while requiring minimal effort and cognitive burden from the wearer. Our overarching strategy was to use the wearer's perception to weigh the internal, experiential factors of wearing the exoskeleton. To this end, we developed a sample-efficient, active learning strategy–based controller that optimizes user preference while the exoskeleton wearer is in the loop; we defined sample efficiency to describe minimizing the number of user queries before the approach reaches the optimum. Our contributions are twofold: a sample-efficient optimization algorithm for obtaining user-preferred controller settings in a four-dimensional parameter space and a real-time framework for allowing users to determine preference at will, simplifying the form of feedback through pairwise comparisons. In this work, we first describe our algorithmic approach, RankCMA-ES, using a neural network and evolutionary strategy for modeling and optimizing preference. Then, we validate the proposed algorithm both in simulation and in experiments with human participants. The intent of our work is to present a system for users to tune controllers automatically and conveniently on the basis of their preferences. We believe that approaches that enable preference-based automatic tuning are a key step in the translation of wearable technologies out of laboratories and, ultimately, into our daily lives.

RESULTS

The proposed approach used an active learning paradigm, where a learning algorithm interactively queries the user for feedback to identify their preferred control parameters (torque assistance settings for bilateral ankle exoskeletons). Our overarching objective was to design a system that iteratively converges to the wearer's preferred parameters while minimizing the queries to the user. Our approach comprises two components; the first component models the latent preference function of the user, and the second component is an optimization strategy that efficiently queries the user on the basis of the learned preference function while generating better candidate solutions. To model the preference function, we used neural networks [RankNet (*36*)] to estimate preferred settings given a set of control parameter candidates. Ultimately, the model provides an estimation of a score for each parameter candidate, which is then used

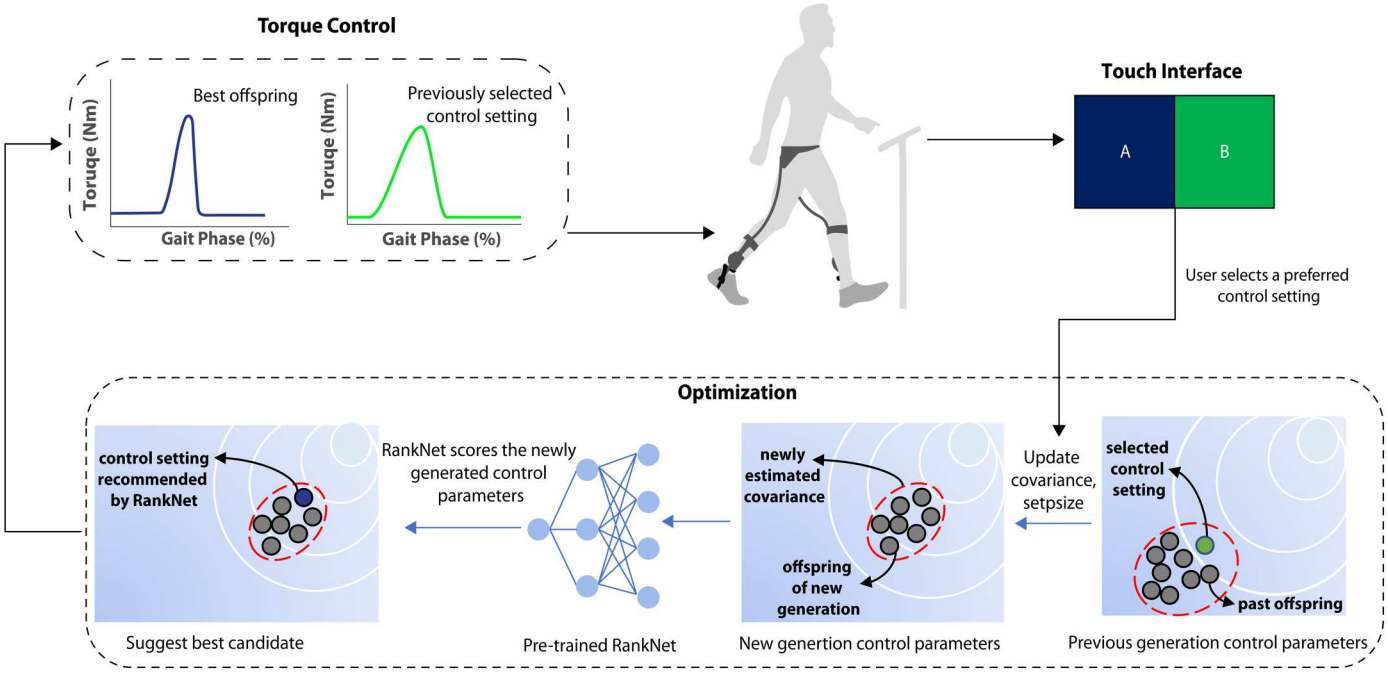

Fig. 1. Overview of our control framework. A method that optimizes exoskeleton torque profile on the basis of user's pairwise preference feedback. For each loop of the optimization, the user experiences two exoskeleton control parameters: the best offspring (blue) and previously selected control parameter (green). Then, the user provides their preference by selecting their preferred parameters via a touch screen interface. On the basis of the selection, a new set of control parameters was generated by the evolutionary strategy (CMA-ES), where these parameters were ranked by RankNet. The highest-ranked parameter (the best offspring) along with the previously selected control parameters were presented to the user as a new pairwise comparison. This process iterated until the approach reached the termination condition.

to rank these candidates. We used an evolutionary algorithm, covariance matrix adaptation evolutionary strategy (CMA-ES), specifically 1 + λ CMA-ES (*37*, *38*), as an elicitation method to sample parameters for querying the user. Given the estimated score from RankNet, CMA-ES selects the best parameters among the population in each generation. We validated the proposed algorithm, which we denote as RankCMA-ES, in both simulation and robotic hardware with human participants (Fig. 1). The following sections outline the model, optimization algorithm, and validation methodologies.

Modeling human preference

We modeled human preferences of exoskeleton controller settings by leveraging a previously published dataset composed of multiple users (*19*). We used RankNet, which is a neural network model with a cost function that assigns probabilities of preference between two settings, originally developed to learn ranking problems (*36*). In our case, we are not necessarily interested in ranking the entire set of assistance preferences, but instead, we modeled the preference landscape. To elicit a user′s preference, we used a forced pairwise choice paradigm: asking wearers to choose which control parameters were preferred when given a choice between two candidates. See Materials and Methods for details on the architecture and training process of RankNet.

We modeled the user preference landscape across two controller dimensions for the ankle exoskeletons: peak torque timing and peak torque magnitude (Fig. 2, A and B). A single landscape was created and trained on the basis of previously collected user preference data. See the "Dataset" section in Materials and Methods for more detail on the creation of the dataset. Most of the users′ preferred settings in the dataset occurred past 55% of the gait cycle; thus, during optimization, RankNet sorts offspring primarily on the basis of peak torque timing and provides a higher score with the timing later in the gait cycle. Note that the total range of peak torque timing was from 30 to 60%.

Validation of optimization in simulation

We first analyzed the performance of RankCMA-ES in simulation by defining synthetic users′ preference profiles and simulated user responses on the basis of these profiles (see the "Experimental protocol: Simulation" section). We analyzed the performance of RankCMA-ES using leave-one-subject-out cross-validation and 1 + 1 CMA-ES (*39*) as a baseline. We repeated the cross-validation three times to account for stochasticity of the performance. The baseline was chosen to see the effect of the preference model, where the 1 + 1 strategy did not use the model information. In general, the performance of RankCMA-ES exceeded the baseline after 30 generations and reached the vicinity of ground-truth optimal score in 150 generations, whereas the baseline showed suboptimal convergence (Fig. 2C). This result implicitly demonstrates the sample efficiency of the proposed algorithm by leveraging the model information.

In addition, we studied the effect of the number of offspring (λ) on the performance of the algorithm. We found the effect of scheduling to be paramount; without scheduling, the algorithm failed to reach the global optimum (Fig. 2D). The details of the algorithm, including the scheduling, are described in the "Combining RankNet with CMA-ES" section.

Validation of optimization with human participants

To further validate the proposed optimization algorithm, we had participants choose between two torque profile settings while the algorithm learned their preference and suggested optimized parameters. Participants walked on a treadmill at a fixed speed of 1.2 m/s while wearing bilateral ankle exoskeletons (Dephy Inc.; Figs. 1 and 3D). They interacted with a touch screen interface that allowed the user to change between two torque profile options (Fig. 3, B and C). The user was instructed to try both options with no time limitation until they were certain which profile they preferred. Once the user decided on a preferred torque profile, they confirmed their selection via a button on the user interface (note S1 and fig. S1). After the user repeated the pairwise choice process 50 times (this number of generations was chosen on the basis of a pilot study), they entered a validation session without being made aware of the change in session type. For the validation session, the protocol was identical to the main optimization session, except one of the two presented torque profiles corresponded to the optimized profile [the last torque profile the user selected before entering the validation session (table S2)], whereas the other profile was obtained from a set of randomly generated control parameters. There were a total of 10 selections made during the validation session. The progression of preferred torque profiles for a representative participant for the optimization and validation sessions is depicted in Fig. 4. On average, the duration of the main session (optimization and validation) was 39 ± 18 min per trial across participants. A trial represents both the preference elicitation session and validation session. We have included the experimental duration for each participant in table S3.

We calculated the mean optimized torque profile across trials for each participant. Overall, users had distinctive optimized profiles over the range of permissible parameters (Fig. 5, A and B). For peak torque timing, the parameters ranged from 40 to 58% gait cycle (all participants except for two had timing between 53 and 58% gait cycle). For peak torque magnitude, the parameters were between 12 and 20 N m. The optimized rise time ranged from 20 to 33% gait cycle, whereas the optimized fall time ranged from 6 to 30% gait cycle. We observed that the optimized profiles covered a wide area within the allowable range of torque; however, peak torque timings were concentrated in a relatively narrow range across participants (Fig. 5B). In addition, to measure the variability of the optimal parameters within a user across trials, we calculated the mean of the SD of the parameters within users. The SDs of the optimal parameters were 7.0%, 3.3%, 1.3%, and 2.4 N·m for rise time, peak torque timing, fall time, and peak torque, respectively. In this work, all torque values are represented as the commanded torque from the controller; there was some minor discrepancy between the commanded torque and the estimated torque from the applied current. The mean absolute error between the commanded and actual torques was 0.56 ± 0.7 N·m across all trials and participants.

We quantified the ability of users to identify their preferred torque profile (validation accuracy) as the percentage that users selected the optimized torque profile instead of the random profile experienced during the 10 validation sessions. The minimum validation accuracy would likely be 50% on average, stemming from the choice between two alternatives. We analyzed the mean and SD of the validation accuracy across trials for each participant (Fig. 6A) and found that participants with lower accuracy had higher

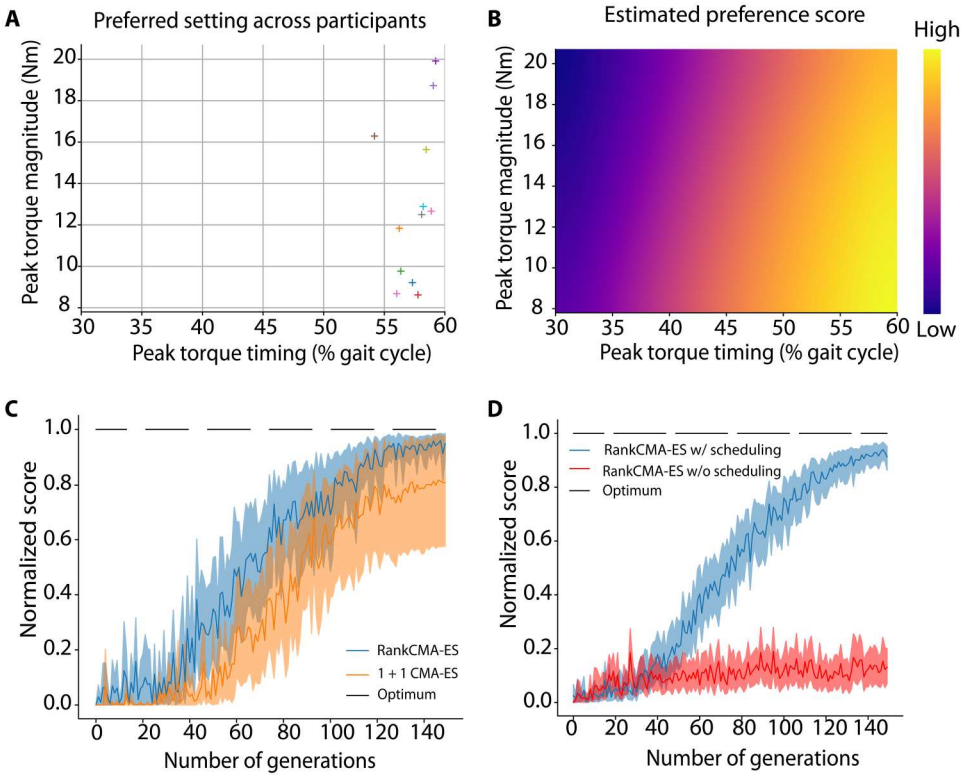

Fig. 2. Estimated preference score from RankNet. (**A**) Users′ preferred controller settings (peak torque timing and peak torque magnitude) from (*19*). Note that these are different users than the participants participating in this study. (**B**) Estimated preference score from RankNet, which we pretrained using the dataset in (*19*). In accordance with (A), scores are higher in the region where users′ settings are clustered. (**C**) Effect of RankNet in simulation. Comparison of optimization results between RankCMA-ES and 1 + 1 CMA-ES. RankCMA-ES (blue) outperforms baseline CMA-ES (orange). The dashed line shows the score of global optimum (upper bound). (**D**) Effect of lambda scheduling in simulation. RankCMA-ES with scheduling (blue) outperforms RankCMA-ES without the scheduling (red). The translucent bands represent 95% confidence intervals

variance. Overall, the mean and SD across all participants was 87.8 ± 10.6%, with a 95% confidence interval of [84%, 90%], which demonstrates that users were able to select optimized parameters over the randomly generated parameters about 9 of 10 times. In addition, we calculated the mean and SD of the validation accuracy across two participant groups, knowledgeable and naive, on the basis of their experience in the field of wearable robots (Fig. 6B). The accuracy for the knowledgeable participant group and naive participant group was 93.8 ± 8.8% and 80.0 ± 17.1%, respectively, with knowledgeable participants significantly outperforming the naive participants (*t* test, $P = 0.001$).

To investigate the underlying factors that drove misselection, we analyzed how the difference between the random and optimized torque profiles affected the validation accuracy for all participants. To quantify this difference, we calculated the root mean square error (RMSE) between optimized and randomized profiles. Because RMSE is a continuous variable, we grouped the range of errors into three groups—RMSE of 0 to 3, 3 to 6, and 6 to 9—and calculated the average and SD of each group′s validation accuracy. To calculate the validation accuracy, the number of times a user selected the optimized profile was divided by the total number of user selections per RMSE group to account for the stochasticity of the sampling process for the randomized profiles. Using an analysis of variance (ANOVA), we were able to observe that RMSE had a statistically significant effect on the validation accuracy. Specifically, the difference between RMSE group 0 to 3 and group 3 to 6 and the difference between the RMSE group 0 to 3 and group 6 to 9 were significantly different (Fig. 6, C and D).

To study users′ adaptation to exoskeleton assistance, we examined the effects of the number of trials on the validation accuracy and preferred control parameters. To this end, we fit linear mixed effects models for validation accuracy and each control parameter. We found that there were no significant effects ($P > 0.05$) of the number of trials on validation accuracy or optimized control parameters. See figs. S3 and S4 for more details.

Last, to investigate the sample efficiency of our approach in the human participant study, we calculated the number of selections that users made until they selected the parameters that were used for the validation session (table S4). On average, users made 43 selections before continually selecting the same parameter sets for the remainder of the preference elicitation session. That is, on average, users kept selecting the same parameter sets from selections 43 to 50 before these settings were chosen for the validation session. Across all trials (14 participants with three trials per participant), five trials did not converge; in other words, for five trials, participants kept changing the preferred parameters until the 50th selection.

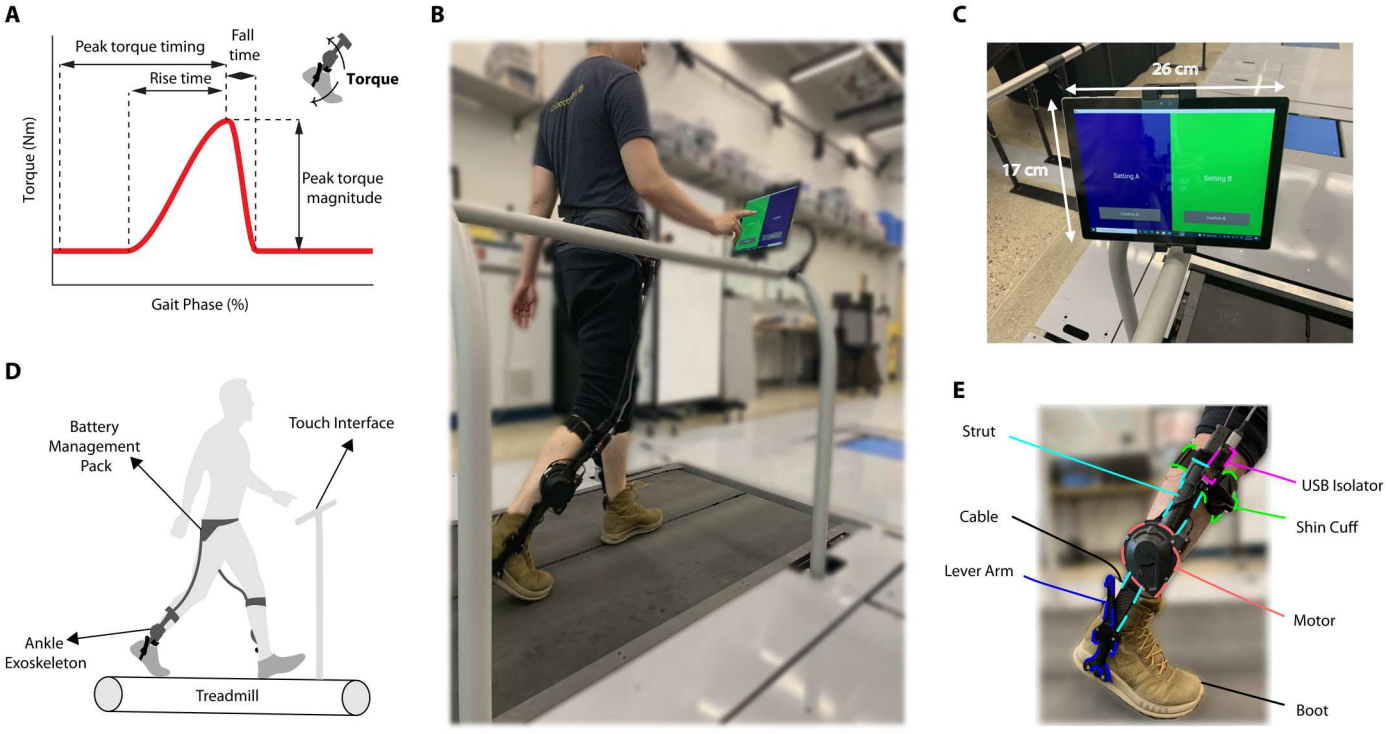

Fig. 3. Torque assistance profile and experimental setup. (**A**) The shape of the torque profile for controlling the exoskeleton. Four settings—rise time, peak torque timing, fall time, and peak torque magnitude—determine the shape of the assistance profile. (**B**) A real-world representation of the setup. (**C**) The placement and dimensions of the tablet where the GUI was presented to users. (**D**) The experimental setup for the preference optimization. (**E**) A bilateral ankle exoskeleton system (Dephy ExoBoot) used in the experiment. The exoskeleton provides unidirectional plantarflexion assistance through a brushless motor and belt transmission during push-off.

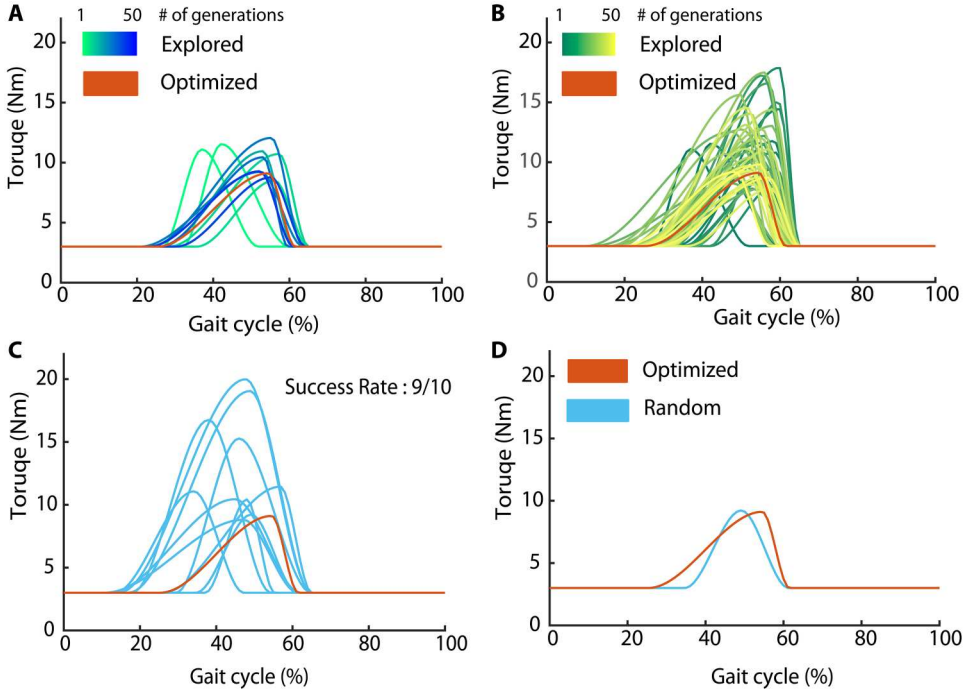

Fig. 4. Progression of torque profiles for one representative participant (number 7, trial 3). (**A**) Selected torque profiles across number of generations. The red line depicts the optimized profile (the profile selected by the participant at the end of the optimization). (**B**) All (selected and nonselected) torque profiles presented in one preference elicitation session (50 generations). (**C**) Torque profiles presented during the validation session. The light blue lines represent randomly generated torque profile, and the orange line represents the optimized torque profile. (**D**) Torque profiles when user selected the randomized profile over the optimized. For this particular participant, the user selected the randomized (light blue) over optimized (orange) once out of the 10 selections during the validation session.

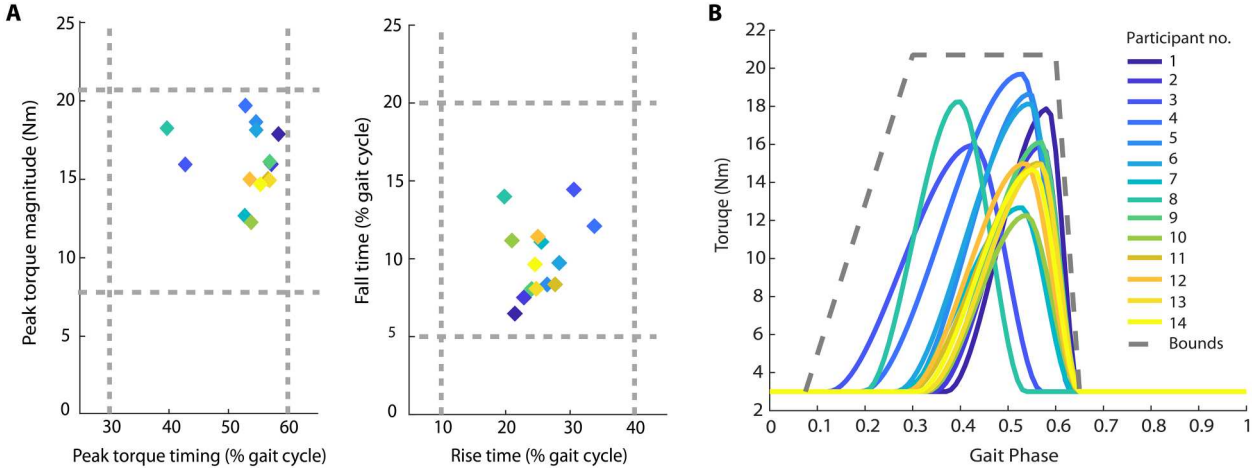

Fig. 5. Optimized control parameters for users in this study averaged across trials. (**A**) The left plot shows the control parameters of peak torque timing and magnitude, whereas the right plot depicts the optimized control parameters of rise time and fall time. The gray lines depict the bounds of each control parameter. Different colors represent different participants, which correspond to the participant number in (B). (**B**) Average torque profiles of all participants. The gray lines depict the bounds of the profile imposed by the control parameters (outside of these bounds, only a bias torque of 3 N·m was applied to the exoskeletons).

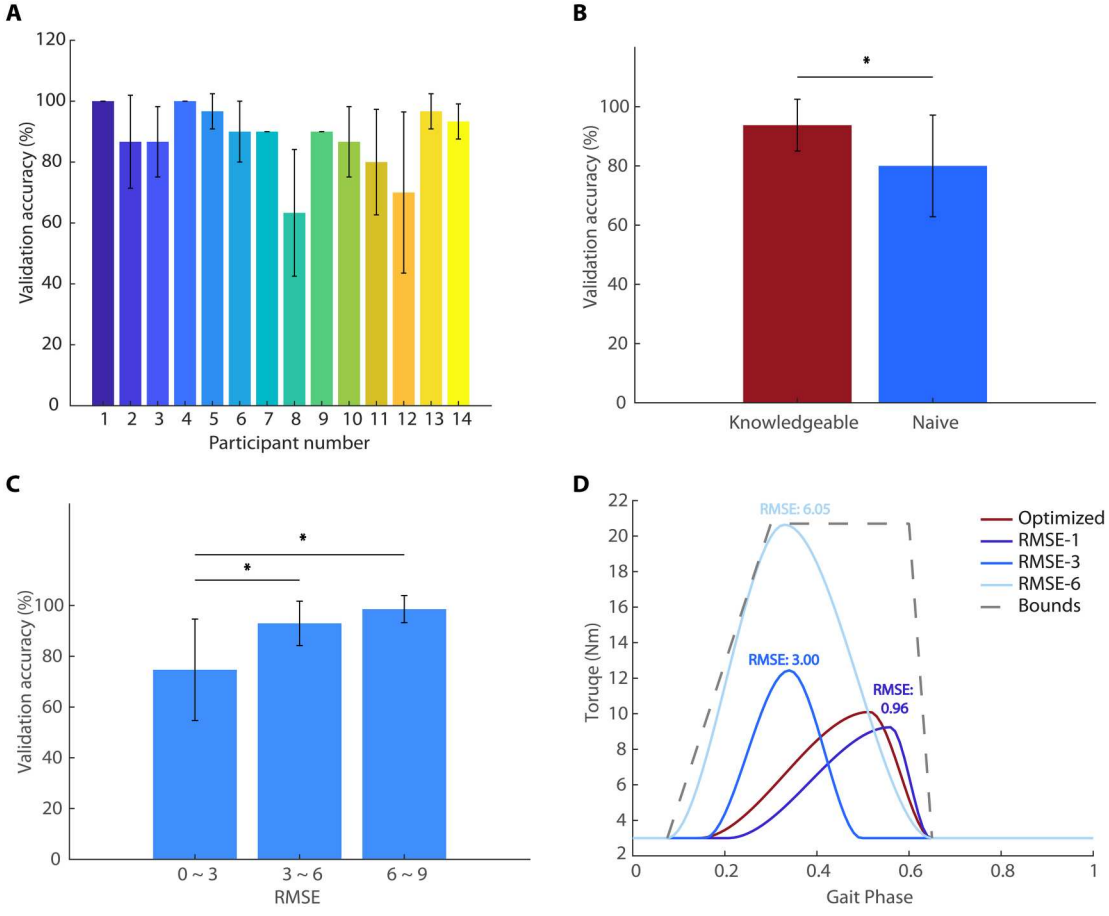

Fig. 6. Validation accuracy across trials for all participants; comparison based on prior knowledge and relation between validation accuracy and RMSE. (**A**) Mean and SD of validation accuracy across trials for all participants. (**B**) Mean and SD of validation accuracy across knowledgeable and naive participant groups with respect to their experience wearing exoskeletons. (**C**) Mean and SD of validation accuracy across RMSE groups. RMSE was measured by comparing the randomized and optimized profiles in torque-gait phase space during the validation sessions, and the validation accuracy was calculated in three RMSE groups. (**D**) An example of torque profiles that represent each RMSE group. An asterisk denotes a statistically significant difference ($P < 0.05$).

DISCUSSION

In this work, we present a sample-efficient, active learning strategy for optimizing control parameters on the basis of user preference while the human is in the loop. By explicitly relying on the learned preference function to lower the number of user queries, our method reduces the overall number of samples necessary for optimization of a new user's unknown preference function compared with an approach with no preference model (baseline CMA-ES). We chose to optimize user preference because of its ability to encompass multiple objectives perceived by the wearer. Preferred settings were obtained using pairwise feedback when comparing two different controller settings. We validated the performance in both simulation and human participant experiments with a robotic ankle exoskeleton. The average experimental validation accuracy, as captured by the user's ability to distinguish the optimized control parameters from randomized settings, was 88%. This accuracy indicates that users were able to reliably distinguish and select their preferred settings and that the algorithm was able to identify users' preferred control parameters well. We also studied whether previous experience with wearable robotics increases the validation accuracy and observed that knowledgeable users had significantly higher accuracy.

The main innovation of our work is twofold: First, we developed a sample-efficient algorithm, which demonstrated repeatable performance with high accuracy. We designed an algorithm by combining RankNet with CMA-ES. The use of CMA-ES in assistive exoskeletons has demonstrated its success in optimizing metabolic rate during use (*16*, *40*, *41*); however, it was unknown whether CMA-ES could successfully optimize human preference. Compared with other works that used preference-based optimization of exoskeleton control, we achieved a higher validation accuracy [83.3% on a single trial across users (*34*)]. The optimization approach in (*34*) collected two types of feedback (pairwise and coactive), which included five options, to optimize their algorithm. However, only pairwise feedback was used to quantify the performance of their approach. This indicates that our work is on par with or better than existing preference optimization. Second, we intentionally designed a system that can be used in the real world. The only extraneous equipment needed includes a simple user interface to collect feedback, which could reside on a tablet, smartphone, or smartwatch, and thus eliminates the need for additional equipment or supervision from experts. Our approach was implemented in a multidimensional controller space while still requiring the simplest form of user feedback (binary choice), thereby minimizing the cognitive burden from the wearer. Recent previous work (*19*) required users to track past history, which may be infeasible to directly search higher-dimensional controller spaces. Our approach allows the user to control the pace of tuning. For example, exoskeleton users may wish to adjust their settings as they perform different activities or stay on the same settings as long as they desire. We believe that these contributions culminate in an approach that fosters advancement in exoskeleton control and translation of exoskeleton hardware.

One of the key ideas in our approach is the difference in the dimensionality of the previous data used for pretraining and the dimensionality of the optimized controller. That is, we trained a preference model in a two-dimensional controller parameter space and used these data to solve a higher (four)–dimensional optimization problem for new users. Our results highlight two advantages: first, use of preexisting preference data that were collected in a different modality (nonpairwise feedback) (*19*). This potential benefit allows for data reuse, merging, and augmentation and thereby reduces the costs of human testing and data collection. The second advantage is extensibility to optimizing in higher dimensions. Similar to this study, we can reuse the preference data that we collected for the four-parameter controller space and apply them to a higher-dimensional controller parametrization. However, note that the unmodeled parameters from (*19*) (rise time and fall time) potentially have a lower effect on the overall shape of the applied assistance.

The validation accuracy of 87.8 ± 10.6% shows that the proposed algorithm was able to optimize the control parameters on the basis of user preferences. During the validation sessions, users were able to perceive the differences between optimized and randomized control parameters and repeatedly identify their preferred settings. The blinded validation test indirectly validates the performance of the optimization, but it does not provide information on how close the optimized settings were to the global optimum in the controller parameter space. To mitigate this shortfall, we further analyzed how validation accuracy increases as the randomized parameters deviated from the optimized parameters. The statistically significant effect of RMSE on validation accuracy demonstrated that as the deviation of randomized and optimized parameters increased, users were better able to identify the differences. In addition, certain errors in discerning parameters are inevitable because of limits on human ability to perceive the differences. The just noticeable difference, which is a minimum change in stimuli humans can reliably perceive (*42*), is used to quantify these perceptual abilities. Previous studies showed that human participants could identify changes between 3.6 to 6.8% of stride period in actuation timing using identical exoskeletons (*43*). This may indicate that when randomized parameters were within about 3% range of optimized parameters, users would have not been able to distinguish between these two assistance settings.

We further analyzed the validation accuracy by dividing users into two groups on the basis of whether they had prior experience or knowledge of exoskeletons (knowledgeable versus naive). We observed that knowledgeable users performed significantly better than naive users. This may indicate that familiarity with exoskeletons contributes to sensing and identifying preferred control settings. A similar study was conducted previously, where the precision in identifying preferred settings was compared between these two groups; although experienced users had better precision, there was no statistical difference found in (*19*). This insignificance disagrees with the results of our study and highlights an area for future study.

One limitation of this approach is that the preference model is pretrained, learned in advance on a dataset of different users, and fixed during the optimization procedure. Although the model (RankNet) is not updated on the basis of the current user, the algorithm, which is based on CMA-ES, still optimizes and adapts to a new user. The pretrained preference model learns a single, aggregate function across multiple users, without conditioning on specific user information. This can be challenging if a new user is out of distribution from the model training set. For example, participant 8 showed a low optimized peak torque timing (38%) when compared with other participants (about 55%). In general, RankNet

recommended settings that had a higher peak torque timing during the optimization process (Fig. 2B). This may have hindered the performance of the algorithm for this participant, converging to a suboptimal setting, which, in turn, resulted in the lowest validation accuracy among all participants (Fig. 6A).

One way to address the limitation of the fixed preference model is to update or newly train RankNet on the basis of new users' preference feedback; however, the challenge would be to carefully choose the frequency of model updates because training necessitates abundant data (60,000 samples in this study), and updating too frequently may lead to forgetting (*44*). Alternatively, we can devise a neural network architecture that considers various user models and adapt the latent user variable using meta-learning during trials, which has proven effective for many robot learning scenarios (*45*, *46*).

In the human participant study, we used a fixed number of queries (50 generations) as a stopping criterion for the optimization. We chose a fixed number of iterations to avoid premature convergence while balancing the overall walking time to prevent fatigue. To investigate the sample efficiency of the convergence, we calculated the number of selections that users made until participants selected the parameters that were used continuously for the validation session. In future studies, thresholds for hyperparameters of CMA-ES (step size and sum of weighted differences of samples) can also be used to explore convergence (*47*).

An important area of future study is to understand the biomechanical or physiological changes that drive the user's preferred controller settings. A previous study using a custom, variable-stiffness ankle prosthesis indicated that kinematic symmetry may be optimized at the user-preferred stiffness, whereas neither metabolic rate nor body weight was supported (*26*). Although a definitive understading of factors that shape user preferences remains an unexplored area, we compared how the optimized torque profile from this study compares with the profiles obtained when optimizing for metabolic rate. In our study, the interparticipant means of control parameters were 27.9%, 10.0%, 53.3%, and 16.1 N·m, whereas the interparticipant means for metabolic rate optimization were 23.2%, 12.3%, 50.3%, and 53.6 N·m for rise time, fall time, peak torque timing, and peak torque magnitude, respectively. Both studies had similar walking speeds [ours: 1.20 m/s and (*16*): 1.25 m/s]. Overall, the parameters are similar, with the exception of torque magnitude. This deviation likely stems from the difference in the limits [ours: 7.8 to 20.7 N m and (*16*): 2 N m to 1 N m/kg]. On the basis of the similarity of profiles between two parameter sets, metabolic rate may be one of the factors that contribute to user-preferred settings.

The motivation of this work was to provide a framework for automatically tuning exoskeleton control parameters that leveraged user preference. Our algorithm enabled users to tune the controller in real time using simple, pairwise feedback. Our work lays a foundation for automated tuning systems for exoskeletons predicated on maximizing user experience. We believe that leveraging preference to control wearable robotic systems is important for these emerging technologies to reach their potential.

MATERIALS AND METHODS

Modeling user preference

We used pairwise comparisons to elicit preference feedback and used RankNet (*36*) to model preferences of exoskeleton assistance. Previous work has demonstrated that a pairwise feedback system, which asks users to choose between two options, is more reliable than numerical scores (Likert or 1 to 10 scales) (*23*, *48*, *49*). Given a true preference function $f: R^d \rightarrow R$ and a pair of settings x_a, x_b, a user prefers setting a over b ($a \succ b$) if $f(x_a) \succ f(x_b)$, where we use $\succ$ to denote the preferred setting from a pair. In this work, we used a fully connected neural network to model the true function, which provides continuous estimation of preference given a controller setting. To learn an estimate $\hat{f}$ of the preference function, we minimize the cross-entropy loss (*29*, *50*, *51*)

$$L_{\text{rank}}(\hat{f}) = -\sum \delta_a \log \hat{P}[x_a \succ x_b] + \delta_b \log \hat{P}[x_b \succ x_a] \quad (1)$$

where $\hat{P}$ denotes the probability of preferring a setting using a softmax function

$$\hat{P}[x_a \succ x_b] = \frac{\exp \hat{f}(x_a)}{\exp \hat{f}(x_a) + \exp \hat{f}(x_b)} \quad (2)$$

and δ is a function in {0, 1} depending on the true preference

$$\begin{aligned} \delta_a = 1, \delta_b = 0 \text{ if } x_a \succ x_b \\ \delta_a = 0, \delta_b = 1 \text{ if } x_b \succ x_a \end{aligned} \quad (3)$$

Preference optimization with a learned model and user feedback

Dataset

We created a dataset by using a previous study from our group that identified preferred assistance settings across multiple users (*19*). The assistance control parameters were peak torque timing and peak torque, which were used to shape the torque assistance profile of bilateral ankle exoskeletons (*16*). We used a total of 12 naive participants' data with three speed conditions (1, 1.2, and 1.4 m/s) where each condition had eight trials. We converted participants' preferred controller settings into pairwise preference data. Each sample had two sets of parameters and corresponding labels indicating the user's preference. For example, if $x_a \succ x_b$, the settings are (x_a, x_b) and the label is (1, 0). We created a synthetic user preference function because we do not have access to the underlying true preference landscape, and we assumed a convex shape for the function. We then calculated a mean and covariance matrix of preferred settings across all (3 speed conditions × 8 trials = 24) trials for each participant to fit two-dimensional Gaussian functions. Because only the relative order between settings matters, the actual value of the preference score (the Gaussian function value for given a setting) was scaled to their preference. We uniformly sampled the settings in the two-parameter space defined by the bounds in (*19*) for a total of 60,000 samples across all participants. Then, we used the output of the fitted Gaussian function to generate pairwise labels for all sampled settings.

Model architecture and training

We used a multilayer, feed-forward, fully connected neural network to model the preference function $\hat{f}$, which maps input (controller settings) to a score (latent user score). We divided the dataset into a training and validation set with an 8:2 ratio across all participants'

data and trained RankNet, using the architecture in Fig. 7, on the training set to minimize the cross-entropy loss (1). We used a stochastic gradient-based optimizer, ADAM (*52*), with L1 and L2 regularization to prevent overfitting and trained for 10 epochs. To find the optimal architecture and hyperparameters, we used Bayesian optimization (*53*) to identify the parameters in relatively few iterations compared with a grid search. The score function to maximize the optimization was the estimated preference score, provided by the model output trained on the validation set. After tuning, the number of hidden layers, neurons per layer, type of activation function, regularization rate for L1 and L2, and batch size were chosen as three, 220, tanh, 10^{-4}, and 16, respectively.

Preference optimization algorithm

One of the key elements of our approach is using a learned preference model to improve the sample efficiency of the optimization cycles by reducing the number of user queries while using a simple form of feedback from a user (pairwise comparison). We used a trained RankNet model on a dataset of user preference data to rank the offspring suggestions of CMA-ES. The modeling component of our algorithm is inspired by Akrour *et al.* (*54*); however, our approach differs in using RankNet instead of RankSVM (*55*). We chose RankNet over RankSVM because of the superior performance in ranking problems when multiple user data are introduced, which stems from the loss function in RankNet being pairwise differentiable (*56*, *57*). Intuitively, the optimization loop has two components: first, a user-independent preference model, provided by RankNet, that acts as an approximation of generic properties of the preference landscape, and second, a user-dependent, black-box optimizer, provided by CMA-ES, that guides the search for user-specific preferences. The optimizer provides suggestions based on explicit user feedback, which are then relatively scored using an estimate of how preference is shaped across users.

We first describe how basic evolutionary optimization, using CMA-ES, is framed for the preference optimization problem and then describe how we bring a learned preference model into the optimization loop. We refer to this combined algorithm as RankCMA-ES.

CMA-ES

Given that the mapping of controller settings to user preference is unknown but can be sampled, we can cast this problem as a black-box optimization problem using CMA-ES (*37*). In our setup, a sample is a set of candidate controller settings, and the unknown function is the preference that would be attributed to that sample.

First, a population of samples was generated from a multivariate normal distribution. Second, the user compared the best candidate sample with the previously selected sample to decide which they preferred between these two samples. Third, if the new candidate sample was preferred over the previously selected sample, then the shape and size of the normal distribution were updated through covariance matrix C and step size σ adaptation, respectively. Compared with standard evolutionary strategy, the mean of the population′s distribution was updated on the basis of the covariance matrix to increase the likelihood of the selection (*37*).

We used 1 + λ CMA-ES (*38*), which is a variant of traditional μ, λ CMA-ES (*37*), where μ and λ denote the number of parents and offspring, respectively. "," signifies that a new generation is selected

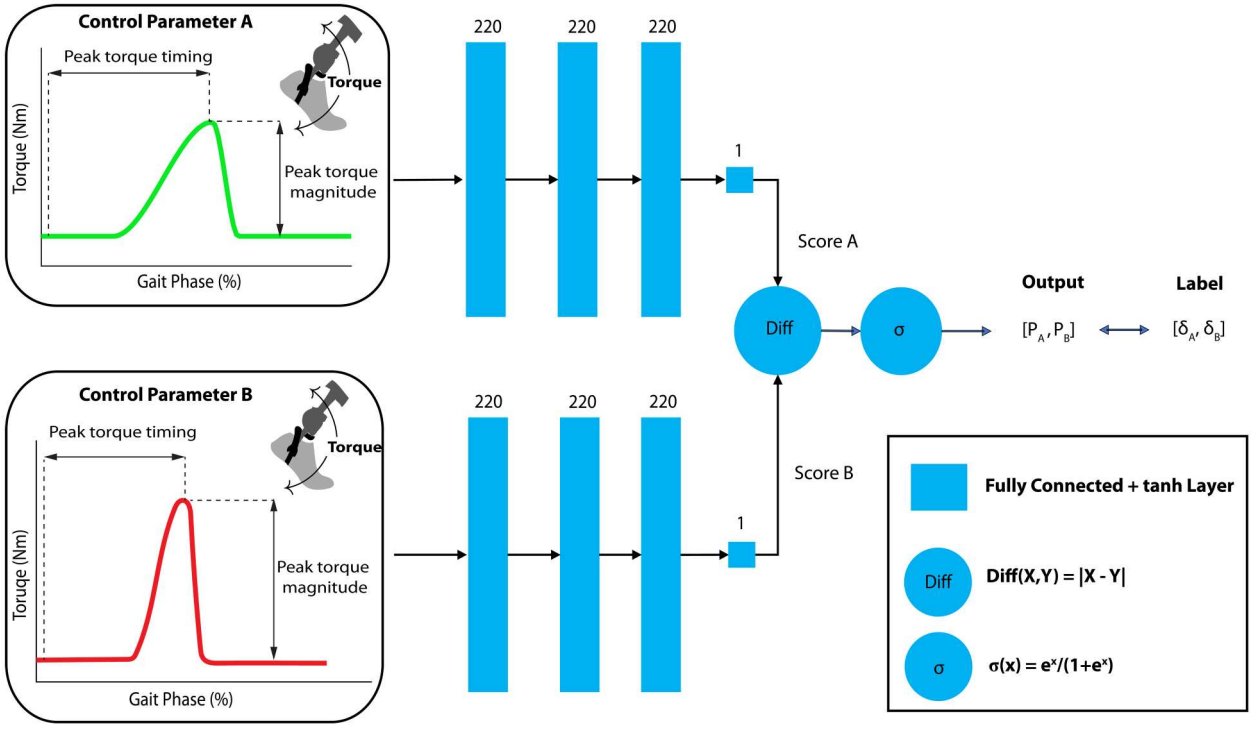

Fig. 7. RankNet architecture. During training, the neural network takes two sets of inputs (controller parameters) and minimizes the cross-entropy loss between the output and ground truth preference label. Each control parameter set is passed through the feed-forward network and converted to an estimated score. The difference of the scores from the two settings is calculated and converted to a probability of preferring one setting over the other.

from only children/offspring, whereas "+" denotes that a new generation is selected from a pool of children and parents. 1 + λ CMA-ES uses a size step adaptation based on a success rule as opposed to path length control. The success rule is defined as the rate of successful offspring λ_{succ}/λ (offspring that have a better score than the parent). Although the 1 + λ strategy may be prone to converging to suboptimal local minima compared with traditional CMA-ES, it converges faster, or equivalently, with fewer function evaluations (*38*, *58*), which makes it a more efficient algorithm for expensive function evaluations involving human participants. We used a λ of eight, which was selected on the basis of a pilot study, and for other hyperparameters of CMA-ES (the strategy parameters), we used values from (*47*) that are known to be suitable for real-world applications.

Combining RankNet with CMA-ES

The optimization algorithm using CMA-ES and the trained preference model through RankNet is outlined in Algorithm 1. The contributions are the following: First, the offspring in each generation are sorted using model inference. The specific offspring with the maximum estimated model score is passed to the user, instead of eliciting users' preference for all offspring candidates. This substantially reduces the number of queries required by users in each generation and overall by $\lambda - 1$ and $N \cdot (\lambda - 1)$, respectively, where N is the total number of generations and λ is the number of offspring per generation. Second, the user's pairwise comparisons are directly used for updating the optimized parameters. Specifically, the user provides their preference between the previously selected best parent and current generation's best offspring for updating the covariance matrix of the CMA-ES. Therefore, only one user response per generation is required to complete the optimization cycle. Third, λ is decreased as the optimization progresses. In this study, we used a scheduling scheme where λ is reduced by half every N iterations. This eventually reduces our optimization to the 1 + 1 CMA-ES version toward the end of the process. The purpose of the scheduling was twofold: (i) It was used to balance convergence speed with optimization accuracy. Because of the uncertainty and noise in modeling human preference, as the parent approaches closer to the optimal control parameters, it may lack the granularity of closely reaching the optimal points compared with 1 + 1 CMA-ES. In this way, we can achieve fast convergence while not sacrificing accuracy by initially leveraging the model (when accuracy matters less) and gradually relying more on the user as the optimization approaches the optimal parameters. (ii) The scheduling was used to account for potential changes in users' preference in the early stage of the trials. We initially used a large λ to have larger search space, and as the experiment progressed, we gradually reduced λ because we expected that user preferences would potentially stabilize. Fourth, the model input can be a subspace of the optimization parameter space. In this study, RankNet estimated a preference score using peak torque timing and peak torque, two of the four controller parameters in the optimization space (details in the "Ankle exoskeleton control" section). This allowed us to use an existing preference dataset and thus reduced the need for costly human participant time, data collection, and model building.

Study design

Experimental protocol: Simulation

We analyzed the performance of RankCMA-ES in simulation (Fig. 2). The purpose of the simulation was to accurately measure the performance of the algorithm in synthetic users' preference function; in human participant testing, we do not truly know the optimum of the user. Optimization performance was measured by the number of generations before convergence and how close the final optimization was to the global optimum. We first defined a synthetic preference profile using four-dimensional Gaussian for 12 synthetic users where the peak of each Gaussian represented preferred parameters of a synthetic user. Among preferred parameters, two parameters (peak torque timing and peak torque) were chosen from the dataset described in the Dataset section, meaning that the other two parameters (fall time and rise time) were synthetically generated (sampled from a constrained normal distribution given the bounds of each parameter). We synthetically created fall time and rise time because they were fixed and not tuned in the dataset we used (*19*). Second, we performed leave-one-subject-out cross-validation trials to train RankNet and test the optimization. The cross-validation was repeated three times (total $12 \times 3 = 36$ simulation runs) to account for stochasticity of the algorithm. For the optimization, we fixed the number of generations to 150 and the hyperparameters for RankNet and CMA-ES across all validations. The number of generations was selected on the basis of a pilot study for a single synthetic participant where the number was sufficient for convergence. All function evaluations (model outputs) were normalized in a [0, 1] range by making the Gaussian for each participant have a maximum value of 1.

We conducted ablation studies to further analyze RankCMA-ES. To study the effect of RankNet in optimization performance, we compared the proposed algorithm with 1 + 1 CMA-ES, which involved the same number of user queries per generation. In addition, we investigated the effect of scheduling for λ, decreasing the number of offspring as number of generation increases, by comparing the performance of the optimization with and without scheduling. We started with λ = 8 and reduced it by half every 10th generation until λ = 1. The initial value and reduction frequency were determined by grid search to maximize the performance while other hyperparameters were fixed. The procedure for validation of both ablation studies was identical to that in the above paragraph.

Experimental protocol: Human participant testing

The experimental protocol consisted of two practice sessions and three optimization trials, where each optimization trial was composed of preference elicitation and a validation session (Fig. 8). All sessions were completed over the course of one study day. During the first practice session, participants donned the ankle exoskeleton and walked on the treadmill while the exoskeleton was powered off. This session was included to ensure that the users adjusted to walking with the exoskeletons on the treadmill. During the second practice session, participants walked on the treadmill with the exoskeleton powered on and providing torque assistance. The users were asked to practice interacting with the tablet computer [graphical user interface (GUI)] to change the torque settings on the exoskeleton and to choose their preference until they were comfortable with the overall interface. For the main optimization session, the users chose between two different torque profiles for 50 generations followed immediately by a validation session, which consisted of selections between the optimal torque setting (the user's last preferred setting) and a randomized torque profile. The users were given a break from walking every 10 min and between every session. The breaks were designed to minimize users' fatigue, which may have affected their preference selections.

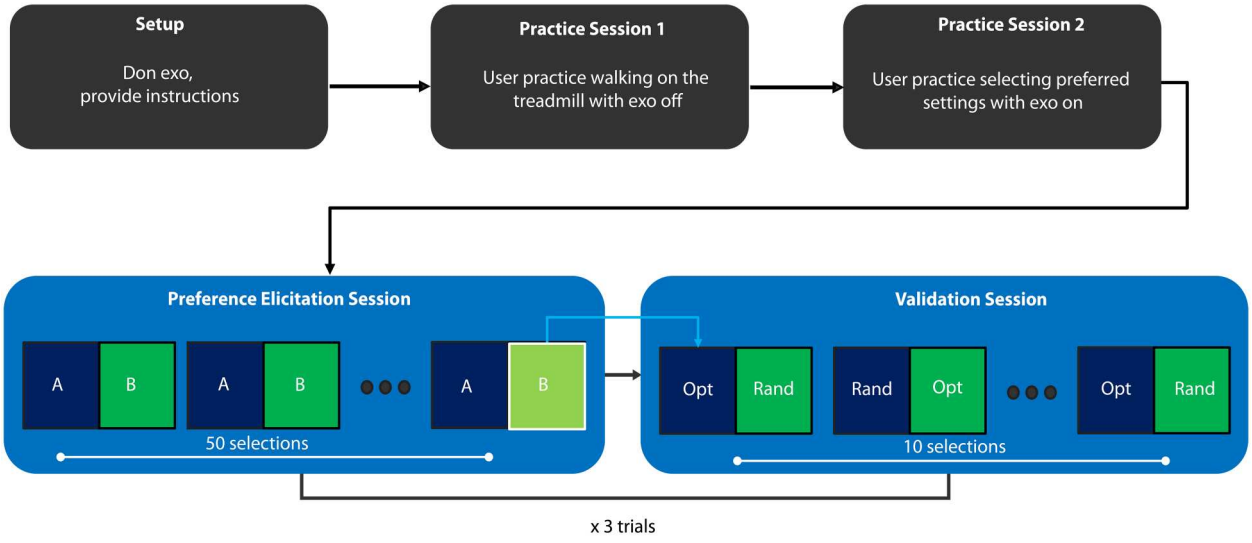

Fig. 8. Experimental protocol for human participant testing. The protocol consisted of the setup, two practice sessions, and three optimization trials that included a preference elicitation and validation session. The practice sessions were designed such that users become familiar with the instruments and preference elicitation process. During the preference elicitation session, users provided their preference between two settings (A and B). The last selected setting was used as the optimization output and passed on to the validation session. Users were not aware of the validation session when comparing between the randomly generated and the optimized control parameters. The sequence of settings in each selection was randomized to prevent users from memorizing the settings.

Experimental duration was limited to less than 6 hours per day and to 4 hours per session to refrain from strenuous exercise.

Participants

We recruited 14 able-bodied participants (10 male and 4 female; average age, 26.8 years; height, 173 cm; weight, 69.3 kg; table S1). Eight participants had prior experience with robotic exoskeletons or were researchers in the field of wearable robotics (experienced participants), and six participants had no prior experience with exoskeletons (naive participants). Experience information was self-reported from a questionnaire provided before the experiment. All participants provided informed consent to a protocol approved by the University of Michigan Institutional Review Board.

Ankle exoskeleton system

Participants donned powered bilateral ankle exoskeletons (ExoBoot, Dephy Inc., Maynard, MA) that provided unidirectional actuation with a peak plantar-flexion torque of ~30 N m (Fig. 3E). Actuation was applied via brushless motors (T-motor U8 KV 100, Nanchang, Jiangxi, China) by applying tension on the belt that is fixed to a lever arm attached to the sole of the exoskeleton boot. The belt produced a nonlinear transmission (4 ~ 17:1) over the range of ankle angle motion. The torque was reacted via shin cuffs at the shank and carbon fiber plates embedded in the boots. The exoskeleton was equipped with onboard sensors including ankle joint and motor encoders, inertial measurement unit (IMU), motor current, and voltage sensors. Each side of the exoskeleton (including boots) had a mass of 1.9 kg, and peripherals included a battery (25-V LiPo) and a single-board computer (Raspberry Pi 3B+, Cambridge, UK) for controlling the actuator system, all of which weighed 1.2 kg. The single-board computer, which included higher-level controller logic, communicated with the exoskeleton embedded system [FlexSEA (*59*)] of the actuator via USB. A Python application programming interface (API) was used to command actuation and read sensors from the system. The main controller script on the single-board computer iterated at 200 Hz, communicating with a closed-loop controller within the motor drive, which closed internal loops at 1 and 10 kHz for position and current control, respectively (*60*).

Ankle exoskeleton control

The exoskeleton provided torque as a function of gait phase. Commanded ankle torque (τ_{ankle}) was converted to a desired motor current, i, using an ankle transmission ratio N [7:1 ~ 17:1, varies as function of ankle angle (*19*)] and motor torque constant K_t [0.14 N·m/A (*60*)]

$$i = \frac{\tau_{ankle}}{N \cdot K_t} \tag{4}$$

where current and torque constants were represented in the q-axis reference frame. The assistance torque profile was shaped using four parameters: rise time, peak torque timing, fall time, and peak torque magnitude, where the polynomial was defined by interpolation of the parameters (Fig. 3A). The methodology of shaping was originally developed by Zhang *et al.* (*16*) and was chosen for its ability to render appropriate assistance functions and agreement with prior studies.

We used a finite-state-machine controller API for the Dephy ExoBoot, developed by Shepherd *et al.* (*61*). The controller had four states that reflected human ankle joint kinematics. During state 1 (late toe-off), the belt was loosened using position control to reach a desired slack to minimize resistance in swing-phase dorsiflexion and allow for ground clearance. In state 2 (swing), the belt maintained a desired slack as needed to exert zero torque. In state 3 (heel strike), the belt was spooled into the actuator to be taut and ready to apply torque. In state 4 (toe-off), the assistance torque profile was applied using closed-loop current control, using the profile shape generated by the four controller parameters. When

the controller was in state 4, the assistance profile had a bias torque of 3 N·m before rise time and after fall time. The bias torque was applied so that the belt was taut to be in a state ready for applying assistance.

The controller parameters were sampled from the CMA-ES optimization while remaining within predefined bounds. The bounds of the four control parameters were informed by previous work on ankle exoskeletons (*16*, *19*). The bounds for the rise time, peak torque timing, and fall time were from 10 to 40%, 30 to 60%, and 5 to 20% of the gait cycle, respectively, and the bound for the peak torque magnitude was from 7.8 to 20.7 N m.

Statistical analysis

To quantify the success of the optimization in human participant testing, we calculated validation accuracy as the percentage of times the user chose the optimized control parameters over the randomly generated parameters during the validation session. During the main optimization sessions, we recorded the control parameters during all generations, including that of the validation session. First, we analyzed the validation accuracy by calculating the mean and SD across validation sessions per participant. To calculate the confidence interval of the validation accuracy, we modeled users′ selections as a linear mixed-effects model with a binomial distribution and the logit function as the link function. The models included validation accuracy as the response variable and participant as a random effect for intercept.

In addition, we compared the validation accuracy between naive and knowledgeable participants across all validation sessions using a two-tailed t test. The significance level was defined at $\alpha = 0.05$.

We hypothesized that as the randomized profile deviated further from optimized profile, users would be able to better distinguish the profiles than when the randomized and optimized profiles were more similar to one another. To evaluate this hypothesis, we used an ANOVA to measure the effect of RMSE between the randomized and optimized profile on validation accuracy where validation accuracy was the response variable, RMSE was a fixed factor, and participant was a random factor. To determine the statistical difference between the groups, we performed post hoc comparisons using Tukey′s post hoc test criterion. The significance level was defined at $\alpha = 0.05$. All statistical calculations were performed using MATLAB (MathWorks, Natick, MA). One of the trials for participant 10 was excluded from all the analyses because of unanticipated distractions that occurred during the experiment.

Supplementary Materials

This PDF file includes:
Materials and Methods
Figs. S1 to S4
Tables S1 to S4
References (*62*–*64*)

Other Supplementary Material for this manuscript includes the following:
MDAR Reproducibility Checklist

REFERENCES AND NOTES

1. A. J. Young, D. P. Ferris, State of the art and future directions for lower limb robotic exoskeletons. *IEEE Trans. Neural Syst. Rehabil. Eng.* **25**, 171–182 (2016).
2. A. F. Azocar, L. M. Mooney, J.-F. Duval, A. M. Simon, L. J. Hargrove, E. J. Rouse, Design and clinical implementation of an open-source bionic leg. *Nat. Biomed. Eng.* **4**, 941–953 (2020).
3. G. S. Sawicki, O. N. Beck, I. Kang, A. J. Young, The exoskeleton expansion: Improving walking and running economy. *J. Neuroeng. Rehabil.* **17**, 25 (2020).
4. R. W. Jackson, S. H. Collins, An experimental comparison of the relative benefits of work and torque assistance in ankle exoskeletons. *J. Appl. Physiol.* **119**, 541–557 (2015).
5. J. Ye, Y. Nakashima, B. Zhang, Y. Kobayashi, M. G. Fujie, Functional electrical stimulation based on a pelvis support robot for gait rehabilitation of hemiplegic patients after stroke, in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (IEEE, 2014), pp. 3098–3101.
6. D. C. Johnson, D. L. Damiano, M. F. Abel, The evolution of gait in childhood and adolescent cerebral palsy. *J. Pediatr. Orthop.* **17**, 392–396 (1997).
7. A. Boonstra, V. Fidler, W. Eisma, Walking speed of normal subjects and amputees. *Prosthet. Orthot. Int.* **17**, 78–82 (1993).
8. R. Waters, J. Perry, D. Antonelli, H. Hislop, Energy cost of walking of amputees: The influence of level of amputation. *J. Bone Joint Surg. Am.* **58**, 42–46 (1976).
9. N. Vanicek, S. Strike, L. McNaughton, R. Polman, Gait patterns in transtibial amputee fallers vs. non-fallers: Biomechanical differences during level walking. *Gait Posture* **29**, 415–420 (2009).
10. L. Gabert, S. Hood, M. Tran, M. Cempini, T. Lenzi, A compact, lightweight robotic ankle-foot prosthesis: Featuring a powered polycentric design. *IEEE Robot. Autom. Mag.* **27**, 87–102 (2020).
11. K. Seo, J. Lee, Y. Lee, T. Ha, Y. Shim, Fully autonomous hip exoskeleton saves metabolic cost of walking, in *2016 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2016), pp. 4628–4635.
12. G. Lv, H. Zhu, R. D. Gregg, On the design and control of highly backdrivable lower-limb exoskeletons: A discussion of past and ongoing work. *IEEE Control Syst.* **38**, 88–113 (2018).
13. M. R. Tucker, J. Olivier, A. Pagel, H. Bleuler, M. Bouri, O. Lambercy, J. D. R. Millán, R. Riener, H. Vallery, R. Gassert, Control strategies for active lower extremity prosthetics and orthotics: A review. *J. Neuroeng. Rehabil.* **12**, 1 (2015).
14. A. M. Dollar, H. Herr, Lower extremity exoskeletons and active orthoses: Challenges and state-of-the-art. *IEEE Transact. Robot.* **24**, 144–158 (2008).
15. J. R. Koller, D. H. Gates, D. P. Ferris, C. D. Remy, 'Body-in-the-loop' optimization of assistive robotic devices: A validation study, in *Robotics: Science and Systems* (RSS, 2016), pp. 1–10.
16. J. Zhang, P. Fiers, K. A. Witte, R. W. Jackson, K. L. Poggensee, C. G. Atkeson, S. H. Collins, Human-in-the-loop optimization of exoskeleton assistance during walking. *Science* **356**, 1280–1284 (2017).
17. Y. Ding, M. Kim, S. Kuindersma, C. J. Walsh, Human-in-the-loop optimization of hip assistance with a soft exosuit during walking. *Sci. Robot.* **3**, eaar5438 (2018).
18. S. Song, S. H. Collins, Optimizing exoskeleton assistance for faster self-selected walking. *IEEE Trans. Neural Syst. Rehabil. Eng.* **29**, 786–795 (2021).
19. K. Ingraham, C. Remy, E. Rouse, The role of user preference in the customized control of robotic exoskeletons. *Sci. Rob.* **7**, eabj3487 (2022).
20. S. Kumar, A. Mohammadi, D. Quintero, S. Rezazadeh, N. Gans, R. D. Gregg, Extremum seeking control for model-free auto-tuning of powered prosthetic legs. *IEEE Trans. Control Syst. Technol.* **28**, 2120–2135 (2020).
21. K. Stewart, C. Diduch, J. Sensinger, Assistive exoskeleton control with user-tuned multi-objective optimization, in *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)* (IEEE, 2019), pp. 554–558.
22. V. S. Shetty, U. H. Lee, K. A. Ingraham, E. J. Rouse, A data driven approach for predicting preferred ankle stiffness of a quasi-passive prosthesis. *IEEE Robot. Autom. Lett.* **7**, 3467–3474 (2022).
23. M. Tucker, E. Novoseller, C. Kann, Y. Sui, Y. Yue, J. W. Burdick, A. D. Ames, Preferencebased learning for exoskeleton gait optimization, in *2020 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2020), pp. 2351–2357.
24. M. K. Shepherd, A. M. Simon, J. Zisk, L. J. Hargrove, Patient-preferred prosthetic ankle-foot alignment for ramps and level-ground walking. *IEEE Trans. Neural Syst. Rehabil. Eng.* **29**, 52–59 (2020).
25. K. A. Ingraham, M. Tucker, A. D. Ames, E. J. Rouse, M. K. Shepherd, Leveraging user preference in the design and evaluation of lower-limb exoskeletons and prostheses. *Curr. Opin. Biomed. Eng.* **28**, 100487 (2023).
26. T. R. Clites, M. K. Shepherd, K. A. Ingraham, L. Wontorcik, E. J. Rouse, Understanding patient preference in prosthetic ankle stiffness. *J. Neuroeng. Rehabil.* **18**, 128 (2021).
27. B. Settles, "Active learning literature survey," (Tech. Rep. TR1648, University of Wisconsin-Madison, 2009).
28. C. Wirth, R. Akrour, G. Neumann, J. Fürnkranz, A survey of preference-based reinforcement learning methods. *J. Mach. Learn. Res.* **18**, 1–46 (2017).

29. P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, D. Amodei, Deep reinforcement learning from human preferences. *Adv. Neural Inf. Process. Syst.* **30**, 10.48550/arXiv.1706.03741, (2017).
30. D. Sadigh, A. D. Dragan, S. Sastry, S. A. Seshia, *Active Preference-based Learning of Reward Functions* (UC Berkeley, 2017).
31. E. Bıyık, N. Huynh, M. J. Kochenderfer, D. Sadigh, Active preference-based gaussian process regression for reward learning. arXiv:2005.02575 [cs.RO] (2020).
32. M. Tucker, N. Csomay-Shanklin, W.-L. Ma, A. D. Ames, Preference-based learning for user-guided hzd gait generation on bipedal walking robots, in *2021 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2021), pp. 2804–2810.
33. A. Erdogan, B. D. Argall, The effect of robotic wheelchair control paradigm and interface on user performance, effort and preference: An experimental assessment. *Robot. Auton. Syst.* **94**, 282–297 (2017).
34. M. Tucker, M. Cheng, E. Novoseller, R. Cheng, Y. Yue, J. W. Burdick, A. D. Ames, Human preference-based learning for high-dimensional optimization of exoskeleton walking gaits, in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2020), pp. 3423–3430.
35. K. Li, M. Tucker, E. Bıyık, E. Novoseller, J. W. Burdick, Y. Sui, D. Sadigh, Y. Yue, A. D. Ames, Roial: Region of interest active learning for characterizing exoskeleton gait preference landscapes, in *2021 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2021), pp. 3212–3218.
36. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, G. Hullender, Learning to rank using gradient descent, in *Proceedings of the 22nd International Conference on Machine Learning* (ICML, 2005), pp. 89–96.
37. N. Hansen, The CMA evolution strategy: A tutorial. arXiv:1604.00772 [cs.LG] (4 April 2016).
38. C. Igel, N. Hansen, S. Roth, Covariance matrix adaptation for multi-objective optimization. *Evol. Comput.* **15**, 1–28 (2007).
39. D. V. Arnold, N. Hansen, A (1+ 1)-cma-es for constrained optimisation, in *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation* (ACM, 2012), pp. 297–304.
40. G. Lv, H. Xing, J. Lin, R. D. Gregg, C. G. Atkeson, A task-invariant learning framework of lower-limb exoskeletons for assisting human locomotion, in *2020 American Control Conference (ACC)* (IEEE, 2020), pp. 569–576.
41. K. L. Poggensee, S. H. Collins, How adaptation, training, and customization contribute to benefits from exoskeleton assistance. *Sci. Robot.* **6**, eabf1078 (2021).
42. S. S. Stevens, Psychophysics: Introduction to its perceptual, neural and social prospects (Routledge, 2017).
43. X. Peng, Y. Acosta-Sojo, M. I. Wu, L. Stirling, Perception of powered ankle exoskeleton actuation timing during walking: A pilot study, in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)* (IEEE, 2021), pp. 4654–4657.
44. J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, R. Hadsell, Overcoming catastrophic forgetting in neural networks. *Proc. Natl. Acad. Sci. U.S.A.* **114**, 3521–3526 (2017).
45. C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in *International Conference on Machine Learning* (PMLR, 2017), pp. 1126–1135.
46. W. Yu, J. Tan, Y. Bai, E. Coumans, S. Ha, Learning fast adaptation with meta strategy optimization. *IEEE Robot. Autom. Lett.* **5**, 2950–2957 (2020).
47. N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.* **9**, 159–195 (2001).
48. C. Basu, Q. Yang, D. Hungerman, M. Sinahal, A. D. Draqan, Do you want your autonomous car to drive like you?, in *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (IEEE, 2017), pp. 417–425.
49. O. Chapelle, T. Joachims, F. Radlinski, Y. Yue, Large-scale validation and analysis of interleaved search evaluation. *ACM Trans. Inf. Syst.* **30**, 1–41 (2012).
50. E. Baum, F. Wilczek, Supervised learning of probability distributions by neural networks, in *Neural Information Processing Systems* (NIPS, 1987).
51. R. A. Bradley, M. E. Terry, Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika* **39**, 324–345 (1952).
52. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings* (ICLR, 2015), pp. 1–15.
53. E. Bakshy, L. Dworkin, B. Karrer, K. Kashin, B. Letham, A. Murthy, S. Singh, AE: A domain-agnostic platform for adaptive experimentation, in *Neural Information Processing Systems Workshops: Systems for ML* (NeurIPS, 2018).
54. R. Akrour, M. Schoenauer, M. Sebag, Preference-based policy learning, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (Springer, 2011), pp. 12–27.
55. T. Joachims, Optimizing search engines using clickthrough data, in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, 2002), pp. 133–142.
56. M.-F. Tsai, T.-Y. Liu, T. Qin, H.-H. Chen, W.-Y. Ma, Frank: A ranking method with fidelity loss, in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (ACM, 2007), pp. 383–390.
57. S. Parthasarathy, R. Lotfian, C. Busso, Ranking emotional attributes with deep neural networks, in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, 2017), pp. 4995–4999.
58. J. Wakunda, A. Zell, A new selection scheme for steady-state evolution strategies, in *Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation* (ACM, 2000), pp. 794–801.
59. J.-F. Duval, H. M. Herr, Flexsea: Flexible, scalable electronics architecture for wearable robotic applications, in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)* (IEEE, 2016), pp. 1236–1241.
60. U. H. Lee, C.-W. Pan, E. J. Rouse, Empirical characterization of a high-performance exterior-rotor type brushless dc motor and drive, in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2019), pp. 8018–8025.
61. M. K. Shepherd, D. D. Molinaro, G. S. Sawicki, A. J. Young, Deep learning enables exoboot control to augment variable-speed walking. *IEEE Robot. Autom. Lett.* **7**, 3571–3577 (2022).
62. M. Kim, H. Jeong, P. Kantharaju, D. Yoo, M. Jacobson, D. Shin, C. Han, J. L. Patton, Visual guidance can help with the use of a robotic exoskeleton during human walking. *Sci. Rep.* **12**, 3881 (2022).
63. K. E. Gordon, D. P. Ferris, Learning to walk with a robotic ankle exoskeleton. *J. Biomech.* **40**, 2636–2644 (2007).
64. S. Galle, P. Malcolm, W. Derave, D. De Clercq, Adaptation to walking with an exoskeleton that assists ankle extension. *Gait Posture* **38**, 495–499 (2013).

Acknowledgments: We thank M. Shepherd for providing the Dephy Exo controller Python API used in the project. We thank K. Ingraham for technical advice and H. Frame for proofreading the article. We also thank members of the team at X, the Moonshot Factory, particularly K. Zealand and A. Memo, for valuable insight throughout this project. Last, we thank C. Powell and the University of Michigan Consulting for Statistics, Computing and Analytics Research for the input on the statistical analyses. **Funding:** This work was supported by X, the Moonshot Factory, Robotics at Google, and D. Dan and Betty Khan Foundation. **Author contributions:** U.L. led the development of the approach with input from V.S.S., S.H., and E.J.R; collected all the experimental data; and conducted the computational and experimental work. V.S.S. assisted with computational and experimental work and developed the communication system. G.E. and J.T. advised the design of the machine learning algorithm. P.W.F. advised the experimental approach and biomechanical aspects of the study. S.H. directed the machine learning aspects of the project and advised the algorithm design and analysis. E.J.R. directed the project and oversaw the control framework, protocols, and biomechanical aspects. The manuscript was written by U.L, V.S.S., and E.J.R., with input from all authors. All authors approved the final version. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** Preference data collected during this study and the optimization code are available at https://doi.org/10.5061/dryad.p5hqbzktp. The dataset from our previous study that were used for training the RankNet model are available from DOI: 10.1126/scirobotics.abj3487.

Submitted 19 December 2022
Accepted 20 September 2023
Published 18 October 2023
10.1126/scirobotics.adg3705

Science Robotics

User preference optimization for control of ankle exoskeletons using sample efficient active learning

Ung Hee Lee, Varun S. Shetty, Patrick W. Franks, Jie Tan, Georgios Evangelopoulos, Sehoon Ha, and Elliott J. Rouse

Sci. Robot. **8** (83), eadg3705. DOI: 10.1126/scirobotics.adg3705

Science Robotics (ISSN 2470-9476) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005. The title *Science Robotics* is a registered trademark of AAAS.