

Research Article

Joint moment estimation for hip exoskeleton control: A generalized moment feature generation method

Yuanwen Zhang^{a,b,1}, Jingfeng Xiong^{b,1}, Haolan Xian^b, Chuheng Chen^b, Xinxing Chen^b, Haipeng Liang^c, Chenglong Fu^b, Yuquan Leng^{a,b,*}

^a State Key Laboratory of Robotics and Systems, School of Biomedical Engineering, Harbin Institute of Technology, Shenzhen 518055, China

^b Shenzhen Key Laboratory of Biomimetic Robotics and Intelligent Systems, Guangdong Provincial Key Laboratory of Human-Augmentation and Rehabilitation Robotics, Department of Mechanical and Energy Engineering, Southern University of Science and Technology, Shenzhen 518055, China

^c School of Engineering and Materials Science, Queen Mary University of London, London, E1 4NS, UK

ARTICLE INFO

Article history:

Received 13 April 2025

Revised 19 May 2025

Accepted 27 May 2025

Available online 9 June 2025

Keywords:

Lower-limb exoskeleton

Hip joint moment estimation

Deep learning

ABSTRACT

Hip joint moments during walking are the key foundation for hip exoskeleton assistance control. Most recent studies have shown estimating hip joint moments instantaneously offers a lot of advantages compared to generating assistive torque profiles based on gait estimation, such as simple sensor requirements and adaptability to variable walking speeds. However, existing joint moment estimation methods still suffer from a lack of personalization, leading to estimation accuracy degradation for new users. To address the challenges, this paper proposes a hip joint moment estimation method based on generalized moment features (GMF). A GMF generator is constructed to learn GMF of the joint moment which is invariant to individual variations while remaining decodable into joint moments through a dedicated decoder. Utilizing this well-featured representation, a GRU-based neural network is used to predict GMF with joint kinematics data, which can easily be acquired by hip exoskeleton encoders. The proposed estimation method achieves a root mean square error of 0.1180 ± 0.0021 Nm/kg under 28 walking speed conditions on a treadmill dataset, improved by 6.5% compared to the model without body parameter fusion, and by 8.3% for the conventional fusion model with body parameter. Furthermore, the proposed method was employed on a hip exoskeleton with only encoder sensors and achieved an average 20.5% metabolic reduction ($p < 0.01$) for users compared to assist-off condition in level-ground walking.

© 2025 The Author(s). Published by Elsevier B.V. on behalf of Shandong University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Lower-limb exoskeletons are capable of assisting in human locomotion, thus reducing joint moments, metabolic expenditure, and mechanical loading of joints. This allows it to be applied to assist weight-bearing workers [1,2], elderly individuals in need of walking assistance [3,4], and patients with arthritis [5]. During walking, the hip joint contributes 40%–50% of the total positive joint power of the lower limb joints [6], which means that hip-assisted exoskeletons can bring substantial benefits. Furthermore, compared with other lower limb exoskeletons, hip exoskeletons have smaller metabolic penalties caused by distal-borne mass [7]. Therefore, hip exoskeletons are considered an important choice for walking assistance.

The moment of the hip joint is a fundamental basis for hip exoskeleton assistance control. Since calculating joint moments

with motion capture systems or force platforms is not feasible for the hip exoskeleton in the real world, it is necessary to utilize other sensors fixed on the exoskeleton system. Recent researchers have focused on determining the assistive torque of the exoskeleton based on the human motion information measured by these sensors, such as encoders on the motors or inertial measurement units (IMUs).

Currently, there are two main methods to generate the assistive torque of hip exoskeletons based on hip joint motion data: determining output torque based on assistive torque curve generation [8,9] and gait phase estimation [10,11], and calculating assistive torque according to instantaneous joint moment estimates, like EMG signals [12], joint angles [13], and ground reaction force [14]. The former method involves the following steps: (a) estimating the gait phase and the assistive torque profile for the entire gait cycle based on the user's historical walking data, and (b) generating the assistive torque based on the gait phase and the assistive torque profile. However, this method may exhibit delayed assistive torque when dealing with variable walking gaits, which can hinder the user's movements [15].

* Corresponding author.

E-mail address: lengyuquan@hit.edu.cn (Y. Leng).

¹ The two authors contributed equally to this work.

The other method generates assistive torque directly through a computational model by measuring the kinematic information of the human body. Compared to the first method, the second method calculates assistive torque independent of gait periodicity, allowing it to generate appropriate walking assistance in real-time even with changing gaits. Therefore, calculating assistive torque with instantaneous joint moment estimation has the potential to become an important computational method for future exoskeleton control.

Estimating the hip biomechanical joint moment accurately is the key foundation to calculate assistive torque instantaneously, but it remains a significant challenge to obtain the estimates based on measurable data from the wearable sensors on the human body. Although some researchers have calculated joint moments by establishing approximate dynamic models [16,17], these methods rely on complex sensor equipment such as ground reaction force sensors, which increases the system complexity, leading to the instability of the exoskeleton system. Besides, extra sensors can be uncomfortable for the users to wear and even hinder their movement.

As data-driven methods have shown remarkable effectiveness in biological measurement [18–20] and exoskeleton control fields [9,15], researchers have started focusing on measured motion information (such as hip joint position changes) by simple and lightweight sensors during human walking for the estimation of joint moment. Wang used the XGBoost method to achieve the estimation of the knee joint adduction moment in an experiment involving 106 participants by using only the information from two IMU sensors [21]. However, traditional machine learning methods have limited performance in improving the estimation accuracy of joint moment estimation, and researchers are turning to neural networks to further enhance the estimation accuracy of joint moment. Molinaro trained a TCN network that utilized expanded causal convolutional layers to encode temporal information and learn data features from input, achieving hip joint moment estimation based on data of simulated IMU and angle of hip joint [15,22]. Furthermore, Hossain designed an Ensemble network that utilized deep neural network layers with different structure types, improving the performance of human dynamics estimation. Through fusion modules and bagging techniques, the model achieved a more accurate estimation of lower-limb joint moment [23].

For the hip exoskeleton control, the joint moment estimation model should be able to adapt to new users so it can provide accurate joint moment estimation promptly. However, the existing methods have not yet achieved this goal, for the reasons that the influence of human parameters is underestimated or difficult to obtain an estimation model that integrates individual characteristics. The lack of personalization might lead to poor applicability to new users, where the assistance torque generated by the exoskeleton might not be suitable for the user's walking.

To address the above problems, we proposed a joint moment estimation neural network based on generalized moment feature (GMF) generation. A feature generator was designed to obtain GMF, which is more generalized than the joint moment, facilitating the estimator to learn the relationship between the kinematics and dynamics during human walking better. Meanwhile, the GMF can be decoded with body parameters into the joint moment, achieving personalizing joint moment estimation. Furthermore, a GRU-based estimator was designed to reduce model computation time with minor accuracy loss under the proposed GMF neural network framework.

The main contributions of this study are as follows:

- (1) Proposing a novel joint moment estimation neural network based on user-specific body parameters, which is conducive

to obtaining a superior individual-independent representation of biomechanical joint moments and achieving better estimation accuracy.

- (2) Conducting the proposed method on a hip exoskeleton to validate its assistance gain for users as well as showing its potential application value in scenarios relying solely on encoder sensors.

The structure of this paper will be described as follows. Section 2 describes the proposed joint moment estimation method based on the GMF generation. The experiments and results are presented in Sections 3 and 4, discussed in Section 5. The conclusion of this paper is written in Section 6.

2. Method

This section provides a detailed explanation of the proposed joint moment estimation method based on GMF generation. We will first describe the conventional form of using neural networks to estimate biomechanical joint moments. Then we introduce the concept of GMF, addressing the issues that arise from the conventional form. Based on this, we propose a neural network framework for generating GMF and estimating it using a GRU module, as well as a decoder for decoding it into the joint moment (Fig. 1 - Model training). Finally, we present the training strategy for the proposed neural network model and the deployment of estimating joint moment in the real world (Fig. 1 - Model deployment).

2.1. The basic model of joint moment estimation

This study aims to estimate the sagittal hip joint moments based on the joint kinematics during walking utilizing neural networks. As mentioned in [24], the interpretability of a model is crucial for generating models with superior accuracy and robustness. Therefore, two essential attributes are assumed that a neural network model should possess to estimate joint moment: (a) incorporating the relationship between kinematics and dynamics, and (b) ensuring the continuity of motion.

The rationality of the second attribute is supported by previous researches [25–27], which indicates that using an appropriate window to capture the input time series can improve the estimation accuracy of the joint estimation model. Regarding the first attribute, to determine the model inputs from candidate kinematic information, we conducted a series of preliminary experiments. Data comprising the angles, angular velocities, angular accelerations, and their combinations of the bilateral hip joints were selected as the inputs for each time step of the estimator. The results of these experiments demonstrated that as the dimensionality of the inputs increased, the accuracy of the joint moment estimation model gradually improved. Therefore, the general inputs of the joint moment estimation model can be described as follows: $\mathbf{X} = [\mathbf{x}_t, \mathbf{x}_{t-1}, \dots, \mathbf{x}_{t-\Delta T}] \in R^{\Delta T \times D}$, where ΔT represents the window size and D denotes the dimensionality of the joint kinematics data ($\Delta T = 100, D = 6$). A basic neural network model for estimating joint moments is described as F . For the estimated joint moment \hat{M} at time t , the process of estimation can be expressed as

$$\hat{M} = F(\mathbf{X}) \quad (1)$$

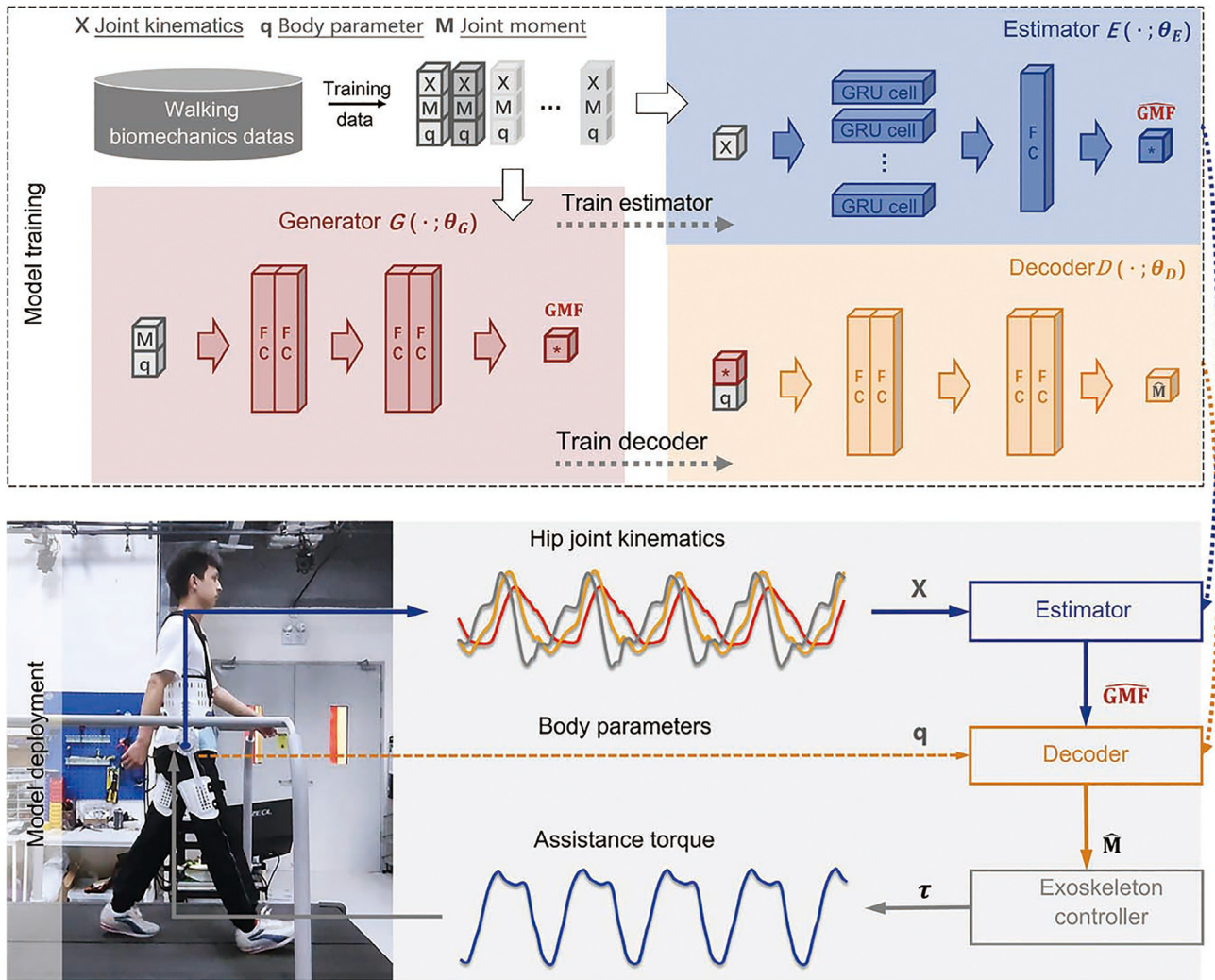


Fig. 1. Joint moment estimation for hip exoskeleton assistance based on GMF generation. During the training process, kinematics information (\mathbf{X}) is utilized as input for the estimator, while body parameters (\mathbf{q}) and hip joint moments (M) serve as inputs for the generator. The decoder takes body parameters and the GMF generated by the generator (represented by “*”) as inputs. The estimator and decoder are then employed to estimate hip joint moments (\hat{M}) to provide assistance torque during walking for new users (the generator is not needed).

2.2. GMF-based model

The conventional form of joint moment estimation networks does not incorporate personalized information of the subjects as the individual parameters are not included in (1). Therefore, to address the issue of decreased accuracy when estimating joint moments using only joint kinematics in a multi-user environment, we propose an novel joint moment estimation method based on GMF generation.

2.2.1. Inspiration of GMF generation

Firstly, we introduce the concept of GMF, which is an individual-independent representation from the joint moment that combines with the subject’s body parameters \mathbf{q} (mass m and height h are used in this paper, which means $\mathbf{q} = [m, h]$). Similar to the expression in (1), the joint moment estimation method based on GMF can be represented as follows:

$$GMF = G(\mathbf{q}, M) \quad (2)$$

$$\widehat{GMF} = E(\mathbf{X}) \quad (3)$$

$$\hat{M} = D(\widehat{GMF}) \quad (4)$$

where M denotes the ground truth of joint moment, G denotes the GMF generator, E denotes the GMF estimator and D denotes the GMF decoder, satisfying $D(E(\cdot)) \in F(\cdot)$.

Equations of (2), (3), and (4) show the core framework of GMF-based model to achieve a better estimation: estimating the GMF instead of directly estimating the joint moment, so that the decrease in estimation accuracy caused by individual differences can be reduced. In other words, GMF-based model improves the generalization performance of the joint moment estimation across multiple users.

Based on the these inspiration, we design a method for generating GMF and apply it to the training of the joint moment estimation model. Fig. 1 illustrates the process of generating GMF. The joint moment estimation method based on GMF leverages both the joint moment and body parameters. It utilizes an adaptive neural network to extract GMF that is independent of individuals. This feature is then used to train the estimator. Additionally, a decoder is trained to decode the estimated GMF into the joint moment. It is worth noting that, compared with inputting body parameters into the network directly, this method expresses the role of body parameters in the network model more specifically, which means it can achieve better optimization from training.

2.2.2. GMF generator

A feature generator G is used to generate the GMF. More specifically, this generator utilizes a neural network to combine the joint moment of subjects with their body parameters and finally generates the desired GMF.

The main objective of GMF generation is to discover a better estimation target to achieve a more accurate estimation by the estimator E as described by (3). Therefore, unlike typical deep learning tasks, in this method, the labels learned by the estimator and the GMF generator are mutually the output values of each other during the training process. This structure guides the GMF generator to generate training labels, namely the proposed GMF, from the joint moment and body parameters of subjects, which can enhance the accuracy of the estimator.

Meanwhile, it is essential that the GMF generated by the generator should be able to be decoded into the joint moment. Using θ_G to represent the network parameters of the feature generator G , the optimal generator parameters θ_G^* in the above process can be mathematically represented as

$$\theta_G^* = \underset{\theta_G}{\operatorname{argmin}} \|E(\mathbf{X}) - G(\mathbf{q}, M; \theta_G)\|_2 \quad (5)$$

$$\text{s.t. } G(\theta_G) \in \mathcal{G}$$

where \mathcal{G} represents the function space to which G belongs, ensuring the GMF generated through G can be decoded into joint moment.

The network structure of the training phase (as shown in Fig. 1) provides a specific illustration of the role of the GMF generator within the overall network structure. The inputs of the generator include joint moment and body parameters, and its output is the GMF. The GMF Generator serves two important functions: (a) generating the optimal GMF to improve the estimation accuracy of the estimator, enhancing the model's performance across different users; (b) ensuring the decoder can accurately decode the generated GMF into the joint moment. To implement a model with the proposed method, as an example, we employ a fully connected network with five hidden layers to construct the GMF generator. Each hidden layer consists of 32 nodes, and a LeakyReLU activation function is applied after each hidden layer to achieve non-linearity.

2.2.3. GMF estimator

In exoskeleton systems, recurrent neural networks (RNN) can better leverage their rapid computation ability [28] to estimate joint moments. Existing recurrent neural network structures include RNN, LSTM, GRU [29]. Due to the challenges of RNN in handling long-term dependencies, vanishing gradients, and exploding gradients, they have been gradually replaced by LSTM and GRU. Furthermore, GRU has a more simplified structure and fewer parameters compared to LSTM, resulting in lower memory usage and faster computation efficiency. In many cases, there is no significant difference in estimation accuracy between the two models [30]. Therefore, the estimator in this study utilizes a neural network based on the GRU structure.

In this study, only one GRU block is used as an implementation case, with 16 hidden neurons, and a fully connected network with a size of 16×1 is used to output the predicted target (Fig. 2). Specifically, at each time step, the GRU receives two different modalities: the hidden information from the previous time step's GRU hidden layer and the inputs \mathbf{X} . It adjusts the information flow through update gates and reset gates, and finally obtains the output based on these two gates. Thus, given the hip joint sagittal plane angle signal at time t as \mathbf{X}_t , the hidden inputs from the previous time step as \mathbf{H}_{t-1} , the estimated GMF at the current time step \widehat{GMF} can be obtained as follows:

$$\mathbf{H}_t = \text{GUR}(\mathbf{X}_t, \mathbf{H}_{t-1}) \quad (6)$$

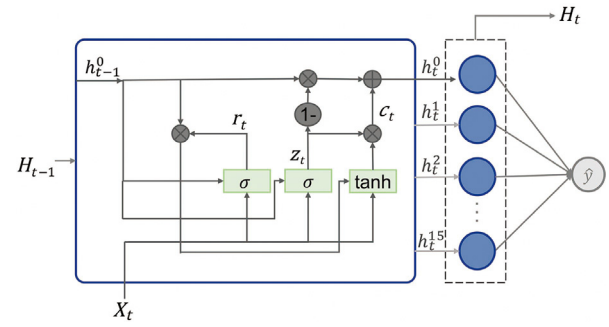


Fig. 2. The fast estimator based on GRU, the computation process of one output within the GRU unit, denoted as h^0 , is shown in the solid box. The computation process for the remaining 15 hidden layer nodes, inputs ($h_{t-1}^1 - h_{t-1}^{15}$), and outputs ($h_t^1 - h_t^{15}$), follows the same process.

$$\widehat{GMF} = \text{FC}(\mathbf{H}_t) \quad (7)$$

Writing the network parameters (\mathbf{W} and \mathbf{b}) as θ_E , like (5), the optimal parameters θ_E^* are solved as

$$\theta_E^* = \underset{\theta_E}{\operatorname{argmin}} \|E(\mathbf{X}; \theta_E) - G(\mathbf{q}, M; \theta_G)\|_2 \quad (8)$$

2.2.4. GMF decoder

The GMF decoder is used to decode the generated GMF into joint moment. We employ the same network structure as the GMF generator for constructing the GMF decoder. To improve the accuracy of the output, the body parameters are included as part of its inputs. Using θ_D to represent the network parameters of the GMF decoder, the optimal parameters θ_D^* of the GMF decoder in the above process can be obtained from:

$$\theta_D^* = \underset{\theta_D}{\operatorname{argmin}} \|D(\mathbf{q}, GMF; \theta_D) - M\|_2 \quad (9)$$

2.3. Model training

The proposed method consists of two parts: GMF estimation and GMF decoding, which involves the training of three models: estimator, generator, and decoder. In the first part, the GMF generator is jointly trained with the estimator to obtain better individual-independent representation (GMF) and its estimation model (estimator). Meanwhile, in the second part, the GMF generator is trained together with the decoder to ensure the decodability of the generalized GMF and to obtain a model capable of decoding the GMF into the joint moment.

Based on (5), (8), and (9), two loss functions L_1, L_2 are constructed for the network training. The backpropagation process is shown in Fig. 1 - Model training with solid lines. L_1 represents the loss between the results of the GMF estimator and the GMF generator, while L_2 represents the loss between the outputs of the GMF decoder and ground-truth values of the joint moment. The loss functions are defined as follows:

$$J_{mse}(y_i, \hat{y}_i) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (10)$$

$$L_1 = J_{mse}(F(\mathbf{X}), G(\mathbf{q}, M)) \quad (11)$$

$$L_2 = J_{mse}(D(G(\mathbf{q}, M), \mathbf{q}), M) \quad (12)$$

It is worth noting that the optimization of the parameters of the GMF generator is also included in L_2 . By selecting the weighting coefficients w_1 and w_2 of the two loss functions, it

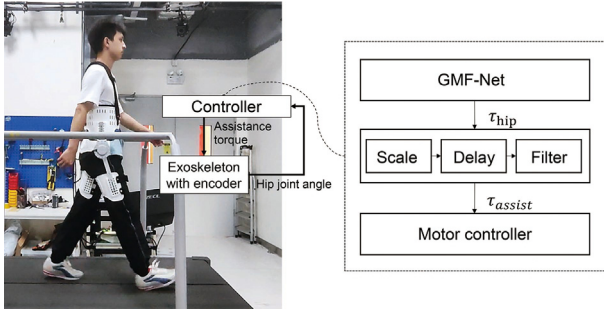


Fig. 3. Exoskeleton and controller used for validating the proposed method. Motors with encoder provide the controller hip joint angles as inputs and generate assistance torque to assist users. The controller is a three-level structure, where the high-level estimates hip joint moments of the user, the mid-level processes hip joint moments to obtain the desired assistance torque and the low-level FOC generates motor torque stably.

can be ensured that the GMF, which is generated by the feature generator and optimized by L_1 , can be decoded into the joint moment through the GMF decoder. Therefore, the condition in (5) is satisfied. The overall loss function of the joint moment estimation network is given by

$$Loss = w_1 * L_1 + w_2 * L_2 \quad (13)$$

2.4. Model deployment

The trained GMF estimator and decoder will be used in the estimation process for the final joint moment estimation. As shown in Fig. 1 - Model deployment, using the joint kinematics as input, the GMF is obtained by the fast GMF estimator, which is then combined with user body parameters to be decoded into the predicted joint moment:

$$\hat{M} = D(E(\mathbf{X}), \mathbf{q}) \quad (14)$$

Subsequently, the estimated joint moments are converted into assistive torques by the mid-level and low-level controllers. In this study, we implement a three-step middle-level exoskeleton controller [15,31] to provide users assistance during walking.

A hip exoskeleton prototype was designed to implement assistance experience for users. As shown in Fig. 3, the assistance torque is obtained by scaling, delaying, and filtering the hip moment estimates. Specifically, τ_{assist} at time t is calculated by

$$\tau_{assist,t} = \text{Low-pass Filter}(k \cdot \hat{M}_{t-\delta}) \quad (15)$$

where k is the scale factor and δ is the delay factor.

3. Experiments

There were two designed experiment parts in this study: the first part was to validate the GMF-based model by a dataset, which is aimed at evaluating the accuracy and generalization of the proposed method, and the second part was to apply the trained model on a hip exoskeleton to validate its performance (assistance gain for users) when used as an exoskeleton high-level controller.

This work involved human subjects in its research. Approval of all ethical and experiment procedures and protocols was granted by Medical Ethics Committee of Southern University of Science and Technology [No. 20240022], under Application 2024-02-24. All participants provided written informed consent to participate in the study.

3.1. Implementation details

We conducted model training on a device with a CPU (Intel, Core i7-12700F) and an GPU (Nvidia, GeForce RTX 3060Ti). The Adam optimizer [32] was employed. In all experiments, we performed five repetitions with different random initializations to ensure robustness. Detailed parameters are described in each section before the corresponding experiment results. The model deployment (exoskeleton controller) was implemented on a Mini-PC (Intel, NUC11PH).

3.2. Validation on dataset

3.2.1. Dataset and training-testing data partitioning

The publicly available dataset used in this article is sourced from [33]. The treadmill data of 20 healthy subjects (12 males and 8 females, age 21.20 ± 3.04 yr, height 1.70 ± 0.07 m, weight 67.76 ± 11.64 kg) were used for model training and testing. Specifically, the data of each subject included hip joint sagittal plane angle positions and flexion/extension joint moment at 28 different treadmill speeds. Data collection of each subject was performed through 7 experiments, with each experiment consisting of four different speeds maintained for 30 s each. The speeds for the first experiment were 0.5 m/s, 1.2 m/s, 1.55 m/s, and 0.85 m/s. In subsequent experiments, each speed increased by increments of 0.05 m/s. Hip joint angular velocity and angular acceleration were computed corresponding to the data collection frequency, and the hip joint moments used were smoothed using butterworth filtering.

To demonstrate the effectiveness of the proposed method, the dataset was partitioned based on inter-subject differences to create a maximum separation between training and test data, resembling a multi-user scenario. The T-SNE method was utilized to divide the hip joint angle data samples from the subjects into two categories, as shown in Fig. 4, with 10 subjects in each category. Consequently, for the experiments in this study, the training and validation sets were composed of data from the first 10 subjects (in an 8:2 ratio, independent of each other), while the data from the remaining 10 subjects were used as the test set to assess the model's performance.

3.2.2. Models for accuracy validation

We evaluated the performance of the proposed method by computing the RMSE of the predicted joint moment on the test dataset. To better demonstrate the effectiveness of the proposed method, we compared our method with two baseline methods:

- **Model without body parameters** - means directly estimating moment by inputting joint kinematics into the estimator without body parameters.
- **Fusion model without GMF** - means implementing a conventional fusion model where two networks separately take joint kinematics and body parameters as inputs to obtain two hidden features, and these two features are then stacked and passed through a fully connected network to estimate joint moment.

To ensure comparability, the architecture of the part inputting joint kinematics in two baseline neural networks is the same with the GMF estimator, and the optimizer (Adam) parameters and initial learning rate ($lr = 0.001$) are the same. In our method, we set the parameters $w_1 = 1$ and $w_2 = 0.05$ in (13). The baseline models are represented as

$$\hat{M}_{b1} = E_{b1}(\mathbf{X}) \quad (16)$$

$$\hat{M}_{b2} = FC_{b2}(E_{b2}(\mathbf{X}), \mathbf{q}) \quad (17)$$

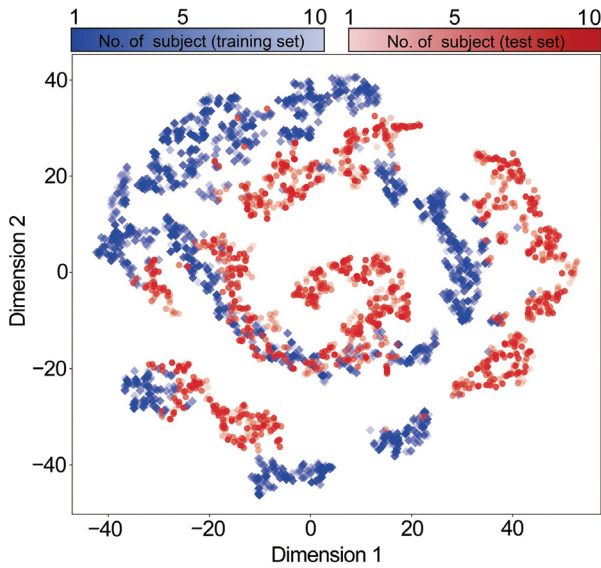


Fig. 4. Visualization of the input features distribution (joint kinematics) of the training set (in blue) and the test set data (in red) using the T-SNE method. The varying shades of color represent data from 10 individual subjects in each set. Under the data partitioning method that maximizes individual differences, the intra-subject distribution of input features within the same set (intra-class distance) appears to be similar, while the distribution of input features between different sets (inter-class distance) exhibits significant differences.

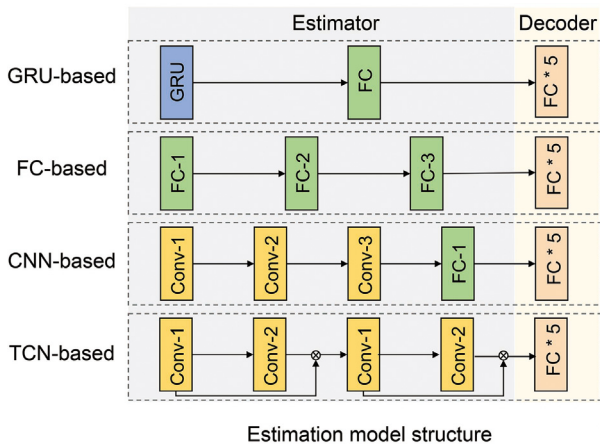


Fig. 5. Four joint moment estimation models consisting of estimators with different neural network structures, including the proposed GRU-based method for fast estimation and three mainstream model architectures.

3.2.3. Generalization validation

More neural network structures were used to verify the proposed GMF framework. To ensure the generalization of the proposed GMF, three existing mainstream network structures (FCNN [19], CNN [27], TCN [15]) were further used to observe the changes in joint moment estimation accuracy after introducing GMF as shown in Fig. 5, following the experimental process outlined in 3.2.2.

- **The FCNN** - was composed of three fully connected layers. The first layer contained 32 nodes, followed by two hidden layers with 16 nodes each.
- **The CNN** - consisted of three convolutional layers, each with a 3×1 kernel size and padding of 1. The layers progressively processed features with 32, 64, and 32 channels, respectively. The extracted features were then passed through an

adaptive average pooling layer with an output size of 1, followed by a fully connected layer with 32 and 1 neurons.

- **The TCN** - employed two residual blocks, each containing two convolutional layers. The first convolutional layer within each block had a 3×1 kernel size, 32 channels, padding of 1, and a dilation of 1. The second convolutional layer maintained the same kernel size and number of channels but applied a dilation of 2 with padding of 2.

To ensure that the variables are consistent, all these models utilized the same GMF generator and decoder, which are trained with the GRU-based estimator in 3.2.2. Specifically, we replaced the GRU-based estimator with the three different network structures. Following the training process shown in Fig. 1, a series of new training processes were conducted. During the training process, only the parameters of the new estimator were updated while the parameters of the generator and decoder were fixed (keeping the generated GMF unchanged).

Previous studies such as [23,26,27] have demonstrated that increasing the number of network parameters has a positive impact on the estimation accuracy of joint moment estimation. Therefore, to ensure comparability of the experimental results, we adjusted the network parameter number of the three models to make them as close as possible to the GRU-based model in terms of validation accuracy when conducting the first baseline as (16).

3.3. Assistance for human walking

The poor generalized model might lead to the controller instability, which reduces the exoskeleton performance and even results in a fall for the user [15]. Therefore, real-world validation is critical for instantaneous joint moment estimation methods used for exoskeleton control. We validated the proposed method on an exoskeleton and implemented walking experiments with users wearing it.

Considering the validation effect, we adopted a rigid exoskeleton structure, ensuring the full torque transfer from the motors to the user as possible. The candidate coefficients of (15) were set as $k = 0.15$ and $\delta = 0.3$ s, which were determined through preliminary experiments.

To quantify the gain for reducing efforts when users walk with the proposed high-level controller, comparison experiments were conducted. Five participants (age 23.4 ± 1.4 y, height 1.7 ± 0.2 m, mass 60.6 ± 1.4 kg) were recruited to complete metabolic trials under three conditions: No Exo, Assist off, and Assist On. The K5 portable metabolic system (COSMED, Rome, Italy) was used to record the experiment data. Specifically, A subject took part in 3 trials with their metabolic cost being collected: a trial without the hip exoskeleton including neutral standing for 3 min and walking at 1.25 m/s for 5 min, a trial with the no-powered exoskeleton for 5 min walking, and a trial with the powered exoskeleton for 5 min walking. At least a 10 min break between trials to allow subjects to return to normal metabolic levels. Before each trial, the participant was required to fast for a minimum of 2 h to reduce metabolic bias from food intake. The instantaneous metabolic cost E was computed based on $\dot{V}O_2$ and $\dot{V}CO_2$ (collected by K5) as

$$E = \frac{16.48 * \dot{V}O_2 + 4.48 * \dot{V}CO_2}{60 * m} \quad (18)$$

where m is the body mass of the subject, and the net metabolism rate E_{net} was obtained by

$$E_{net} = E_{walking} - E_{static} \quad (19)$$

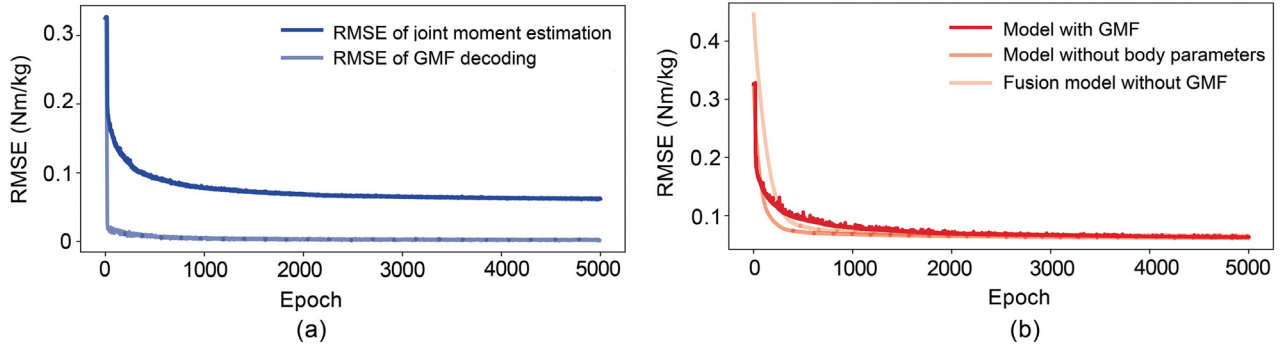


Fig. 6. (a) Trend of the accuracy of the joint moment estimation (dark blue) and accuracy of GMF decoding (light blue) of training data during the training process of the proposed estimation model based on GMF. Both values of them continuously improve and converge to the optimum over the 5000 epochs. (b) The trend of estimation accuracy on the validation set for our method (dark red) compared to two other contrastive methods (light red). The three methods approximate the best accuracy achieved under the experimental conditions set.

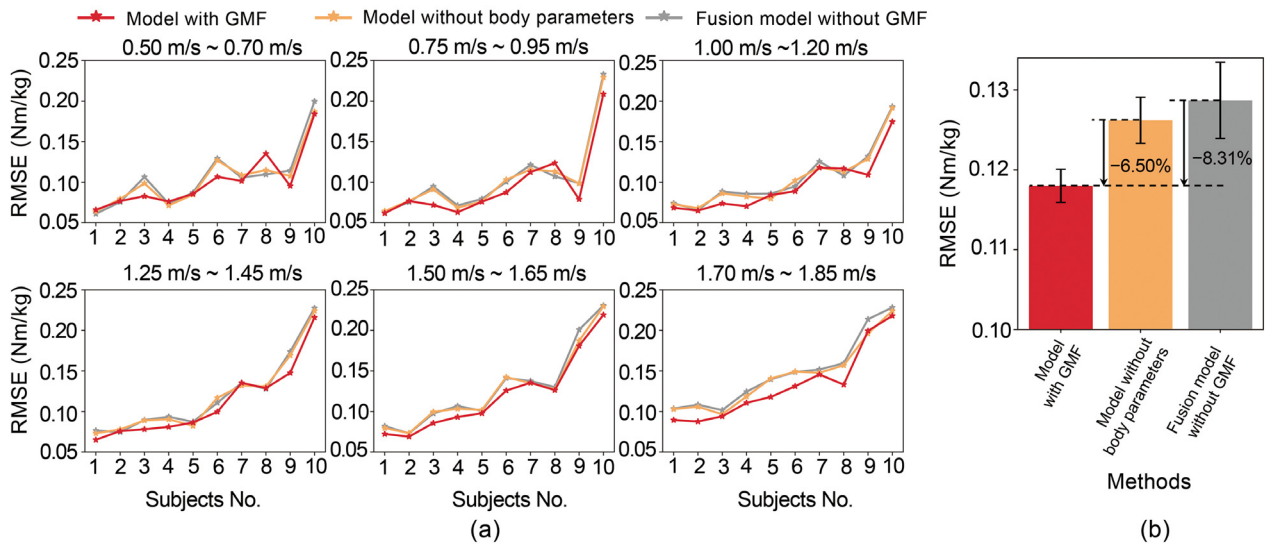


Fig. 7. (a) Comparison of the accuracy of joint moment estimation for all subjects at different speed ranges using the proposed method, along with the baseline methods. The proposed method improves the estimation accuracy of the model in most cases, especially at faster speeds. (b) Comparison of the overall average and variance of the test results for the three methods.

4. Results

In this section, the results of the experiments are presented with detailed statistical analysis. The convergence of the proposed model during training is illustrated in Section 4.1. Sections 4.2 and 4.3 demonstrate the accuracy improvement brought by the proposed GMF method and the generalization performance of GMF. In Section 4.4, the tested assistance gain for users with the proposed method is given.

4.1. Convergence of neural network

The convergence of the proposed neural network trained with 10 subjects from dataset is illustrated in Fig. 6(a), represented by the root mean square error (RMSE):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (20)$$

where \hat{y}_i denotes the estimated value by the neural network, y_i is the ground truth value, and n is the number of samples. During the training process, both the joint moment estimation accuracy and the decoding accuracy of the decoder consistently improved

without significant fluctuations. The RMSE of the joint moment estimation on the training dataset, averaged over five repetitions, was 0.0630 ± 0.0014 Nm/kg, while the RMSE of GMF decoding was 0.0016 ± 0.0017 Nm/kg.

4.2. Estimation accuracy of models

Fig. 7 reports the RMSE results of our method compared to the baseline methods. Overall, our method achieves an RMSE of 0.1180 ± 0.0021 Nm/kg on the test dataset. The mean results show a significant improvement compared to the two baseline methods (6.50% vs. model without body parameters and 8.31% vs. fusion model without GMF). The experimental results demonstrate that, except for subject 6, whose average joint moment estimation error of all walking speed increased by 0.97% compared to model without body parameters and 2.57% compared to fusion model without GMF, the joint moment estimations for other subjects are more accurate. The second baseline method, which directly fuses body parameters (\mathbf{q}) and joint kinematics (\mathbf{X}) through the network, does not achieve improved joint moment estimation accuracy compared to model without body parameters with only inputting joint kinematics. In fact, its performance slightly degrades, suggesting that conventional fusion models struggle to effectively integrate body parameters.

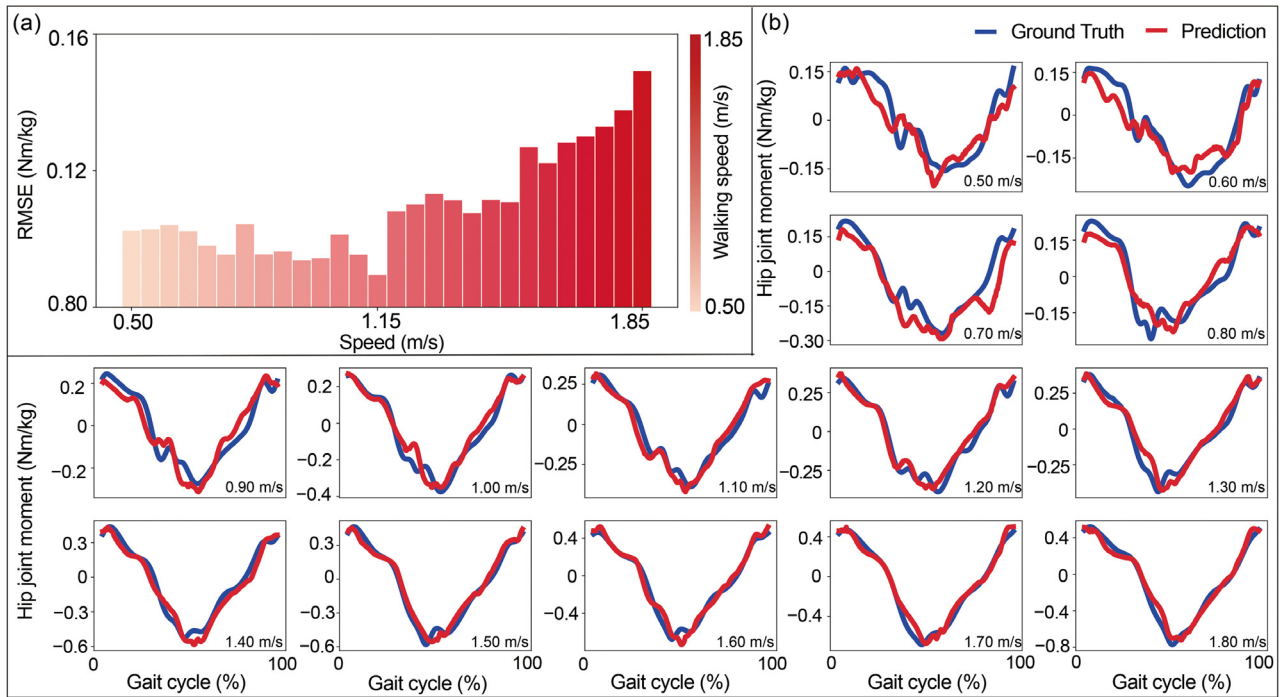


Fig. 8. (a) Joint moment estimation error of the proposed method at 28 different walking speeds. The results show that within the provided range of walking speeds, the joint moment estimation error exhibits a “V” shaped pattern, with the minimum estimation error occurring at 1.15 m/s. (b) Samples of the predicted curve and the ground truth curve of a complete gait cycle for one subject at 14 walking speeds, spaced at intervals of 0.1 m/s.

Table 1

R_{tv} of proposed method compared to baseline methods.

Index	Model with GMF	Model without body parameters	Fusion model without GMF
R_{tv}	0.8907	1.1142	1.1023
Rate of Increase	-	20.06%	19.20%

As it is difficult to achieve the optimal validation set accuracy with limited time, in addition to using the RMSE, we used the rate of increase in the RMSE on the test set relative to the RMSE on the validation set, denoted as R_{tv} , which represents the degree of overfitting [34,35] of the optimal model, obtained from:

$$R_{tv} = \frac{RMSE_{test} - RMSE_{validation}}{RMSE_{validation}} \quad (21)$$

This equation implies that the R_{tv} of a model with better generalization performance should be smaller (with consistent accuracy on the validation dataset).

Fig. 6(b) shows the variation of validation accuracy during the training process for our method and the two baseline methods. All three models achieve similar accuracy on the validation set (around 0.0632 Nm/kg). Based on the statistical analysis (Table 1), our method reduces R_{tv} by 20.06% and 19.20%. While RMSE intuitively demonstrates the effectiveness of our method, the R_{tv} for result evaluation reveals the underlying reason for the accuracy improvement of our method: the mapping of GMF and joint kinematics is more accurate, reducing the estimation inaccuracy of the model in new subject environments.

Complete hip joint moment curves for one gait cycle of a subject at 14 different walking speeds is shown in Fig. 8(b) for qualitative comparison between the ground truth and predicted values. Furthermore, we analyzed the variation of model estimation error across 28 walking speeds ranging from 0.5 m/s to 1.85 m/s, as depicted in Fig. 8(a). The estimation error exhibits a decreasing and then increasing trend. For low walking speeds,

this could be attributed to the greater variability in gait patterns during slow walking [36]. On the other hand, the increase in estimation error for fast walking speeds is likely driven by the higher magnitude of joint moments, as shown in Fig. 8(b).

4.3. Generalization of GMF

Table 2 presents the performance of GMF in the generalization experiment. Within the scope of this experiment, all four estimators showed an improvement in joint moment estimation accuracy after introducing GMF (compared to the method without body parameters: 7.96% in FCNN, 8.12% in CNN, 8.41% in TCN, and compared to the method fusion body parameters but without GMF: 11.44% in FCNN, 10.94% in CNN, 9.13% in TCN). Since the GMF generator and decoder used in the experiment were from a pre-trained model and with no parameters change, the results further demonstrate that the generated GMF is a more individual-independent representation of joint moment and contributes to the accuracy of predicting joint moment based on joint kinematics.

4.4. Reducing metabolism rate of users during walking

Fig. 9 illustrates the assistance performance of the exoskeleton deployed with proposed GMF-based controller, including the assistive torques during three phases: transitioning from standing to walking, steady-state walking at 1.25 m/s, and transitioning from walking to standing. Benefiting from the powerful representational capabilities of neural networks, the GMF model controller can achieve stable assistance during process transitions. The peak torque was about 7.81 Nm for the support phase and 4.29 Nm for the swing phase, respectively. The peak amplitude Difference between the two phases shows the proposed method achieves expected assistance torque as it is widely known the hip joint requires greater moment in the support phase than it in the swing phase during walking. Fig. 9 also shows a more detailed

Table 2
Estimation accuracy performance with different estimators in the generalization experiment.

Estimator	Model without body parameters	Fusion model without GMF	GMF-based model (Ours)
GRU	0.1262 ± 0.0021	0.1287 ± 0.0048	0.1180 ± 0.0030
FCNN	0.1270 ± 0.0029	0.1324 ± 0.0048	0.1169 ± 0.0018
CNN	0.1231 ± 0.0009	0.1270 ± 0.0122	0.1131 ± 0.0009
TCN	0.1260 ± 0.0011	0.1270 ± 0.0037	0.1154 ± 0.0008
Average	0.1256 ± 0.0015	0.1288 ± 0.0022	0.1159 ± 0.0018

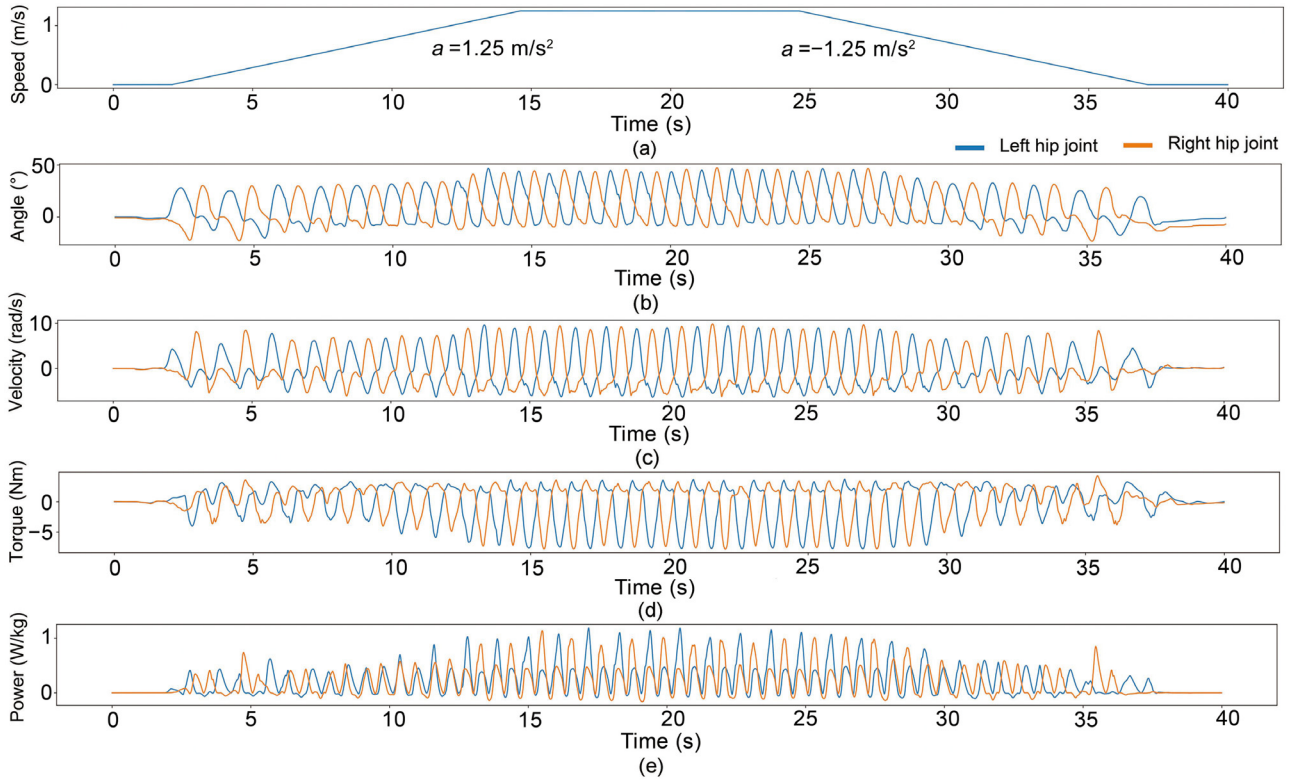


Fig. 9. Controller performance in variable-speed walking trial. (a) Walking speed varied from 0 m/s to 1.25 m/s, maintained for 10 s, and then decelerated to 0 m/s. (b)–(e) Motor angle, angular velocity, assistance torque, and power trajectories on hip joint with the walking speeds corresponding to (a). The positive direction of the vertical axis represents the flexion of the leg during walking, while the negative direction represents the extension.

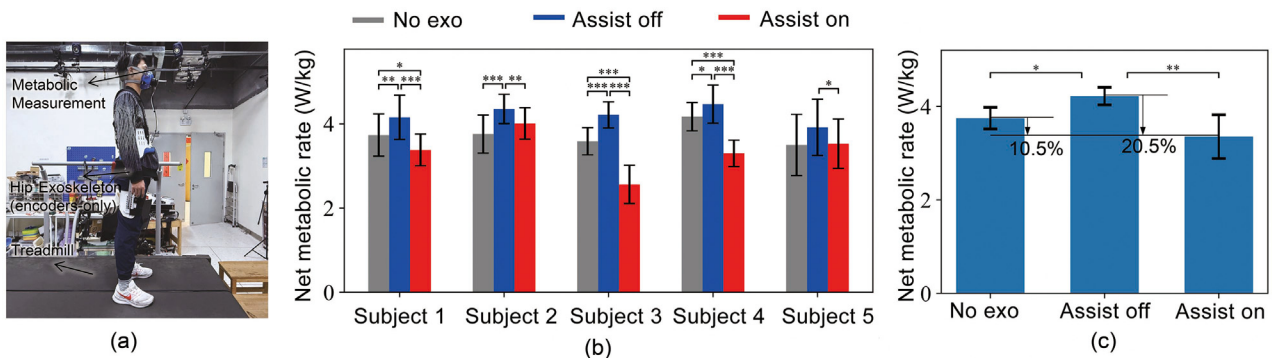


Fig. 10. (a) Experiment environment of the metabolic cost test. (b) Net metabolic rate for five subjects (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$). All subjects received the assistance gain compared to Assist Off. (c) Average net metabolic rate for all subjects.

manifestation of the hip joint angle, angular velocity, and power during assistive walking. All of these data were obtained from the motor encoders of hip exoskeleton.

The metabolic cost test results are presented in Fig. 10. During the experiments, the exoskeleton, utilizing the proposed joint moment estimation method, delivers assistive torque to users,

effectively reducing their metabolic rate. The mean net metabolic cost reduction achieves 10.5% compared to normal walking (No Exo), while the reduction achieves 20.5% compared to the Assist Off model.

Multiple comparison tests were performed to evaluate the metabolic cost of users during walking with the exoskeleton, as

the results are shown in Fig. 10(b) and (c). Specifically, an analysis of variance (ANOVA) was conducted to compare the metabolic cost of users across three conditions: No Exo, Assist Off, and Assist On. Let k denote the number of groups and N the total number of observations. The test statistic is defined as

$$F = \frac{MS_{between}}{MS_{within}} = \frac{\frac{1}{k-1} \sum_{i=1}^k n_i (\bar{x}_i - \bar{x})^2}{\frac{1}{N-k} \sum_{i=1}^k \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)^2} \quad (22)$$

where \bar{x}_i is the sample mean of group i , \bar{x} is the overall mean, n_i is the sample size of group i . The corresponding p -value is calculated from the cumulative distribution function of the F-distribution with degrees of freedom $df_1 = k - 1$ and $df_2 = N - k$:

$$p = P(F_{k-1, N-k} \geq F_{obs}) \quad (23)$$

where F_{obs} is the observed value of the F-statistic. To control the family-wise error rate during post hoc comparisons, we applied the Bonferroni correction. If m independent comparisons are conducted, the corrected significance threshold is

$$\alpha' = \frac{\alpha}{m} \quad (24)$$

A notable result is that 2 participants (Subject 3 and Subject 4) in the validation experiments achieved better assistance gain because the participants were most comfortable with the exoskeleton size and had undergone longer walking training for using the exoskeleton, which has been proven to be effective in increasing a subject's metabolic cost reduction [37].

5. Discussion

This study utilizes a GMF generation method with a GRU-based network to achieve a better estimation of the hip joint moment and its application on the exoskeleton, which benefits from the consideration of inter-user variability in joint moment estimation for exoskeleton assistance. The GMF, which is obtained through an adaptive neural network, leads to significant improvements in model accuracy and has the potential to become an advanced method for future joint moment estimation. The metabolism cost test result showed that our methods achieve the satisfactory results of assistance gain.

5.1. Balance between accuracy and computation efficiency

This paper proposed a GMF-based neural network framework, aiming to improve estimation accuracy with limited computation sources. To achieve the accuracy superior to the state-of-the-art (SOTA) model, neural networks with large-scale parameters are conventionally required. Our previous research has stated the relationship between accuracy and computation time [27], in which the estimation accuracy of the neural network approaches saturation as the number of network parameters increases. For a real-time exoskeleton control system, efficient computation translates to faster response speed, higher estimation precision, and longer endurance time. This is one of the primary motivations of this study: to explore more efficient estimation methods rather than relying solely on increasing the number of neural network parameters.

It is clear that better estimation accuracy would benefit the exoskeleton control based on instantaneous joint moment estimates. However, there is an uncertain question of whether a better prediction accuracy provides a significant increase in assistance gain. From the results in Table 2, FCNN, CNN and TCN models have a higher estimation accuracy than the GRU model. However, on the one hand, the other three models have a lower control frequency, which may lead to a poor performance when they were applied on the hip exoskeleton. On the other hand,

the multiple comparisons test (with Bonferroni correction) results show that the difference in estimation accuracy among the four models is not completely significant. Therefore, the GRU model is selected as the GMF estimator for the exoskeleton control system in this study.

5.2. Transferability of using other input sensors

Compared to most of the existing methods for estimating joint moment (using sensor information such as IMU as model input), the GRU-based estimator used in this paper takes joint kinematics data as inputs instead of raw sensor data. Because the joint moment estimation had been considered a model identification problem that requires establishing an approximate model with dynamic characteristics rather than a completely uninterpretable black-box model. This also means that the model controller in this study uses sensor data in a more humanly understandable way, trying to ensure the reliability of the result [38]. Our exoskeleton experiments were able to implement a deep learning-based assistance control model utilizing only motor encoders and got successful results, which we believe it was benefited from this more rational approach to data processing.

Besides, for other potential practical applications, model training can be performed using angles obtained from sensors or using highly correlated data like IMU data of sensors as inputs directly, without requiring changes to the overall model structure. However, additional challenges may be posed in these ways, such as sensor accuracy drift and data transmission loss [39].

5.3. Better potential estimation accuracy

The other important inputs used by the model are human body parameters, which are composed of height and weight in this paper. These two parameters are valid but might not be the optimal attributes for obtaining GMF. Future research could explore more suitable body parameters to achieve better GMF generation and more accurate estimations in the problem of joint moment estimation based on joint kinematics.

Furthermore, since the proposed method originates from the research on assistive hip exoskeleton control, the estimator model proposed in this paper uses a GRU-based network structure. However, for joint moment estimation tasks that do not require timeliness, the optimal network structure for the estimator may not necessarily be based on recurrent neural networks. The joint moment estimation method we propose can be extended to the field of biomechanics, as demonstrated in the generalization experiment, where it also performs well on mainstream neural networks besides the GRU-based network. For systems with low real-time requirements, future researchers can use more appropriate network structures within the framework of this method to obtain higher-precision joint moment estimation models.

For practical applications, it is possible to further improve the estimation accuracy by using more advanced hyperparameters tuning methods, such as Bayesian optimization [40], to find the optimal hyperparameters for the GMF modules.

5.4. Limitations

The proposed method still has several limitations that need to be further addressed in future research.

5.4.1. Burden of wearing exoskeleton

Fig. 10 has shown a 12.5% metabolic cost increase when subjects wearing the hip exoskeleton without power compared to normal walking. Although this study focused on achieving accurate joint moment estimation, there is a clear need for exoskeleton structures with low wearing burden to enhance the assistance gain for users. Meanwhile, as the results of the experiment more conclusively verified that training benefits the exoskeleton user to reach a higher assistance gain, future studies may need to consider how to control this impact in experiments.

5.4.2. Accuracy of the decoder

To decode the GMF, we use a fully connected neural network with the same structure as the feature generator, and make the loss of decoder back-propagate to the generator with a weighting coefficient to ensure the decodability of the generated GMF. While this method achieves high accuracy (decoding RMSE less than 0.003 Nm/kg) in decoding the GMF into the joint moment, the decoding error can only approximate zero rather than reaching it due to the use of a separate neural network. Additionally, to ensure accurate GMF decoding, the decoder may become overly complex, leading to increased computational time. This complexity transforms what should be a straightforward inverse transformation into a more intricate and less reliable process. The concept of reversible neural networks [41–43], which have lightweight memory consumption and can preserve input information features, has been proposed. When applied to our method, it has the potential to achieve zero error in GMF decoding with fewer parameters. We may further explore the application of such networks in our method in the future.

5.4.3. Exoskeleton control with encoder-only sensors

Inspired by the idea that any sensor capable of measuring joint kinematics can be used to estimate joint moments, we implemented exoskeleton control using motor encoders alone. However, encoder-only sensors still face significant limitations in exoskeleton applications, primarily due to the known issue of coupling between motor motion and human movement.

Our experiments show that encoder-only sensors can provide stable and accurate estimations of joint moments when the assistance time is appropriately calibrated and comfortable for the user. However, determining the correct assistance timing requires pre-experimentation to identify the optimal delay time factor. Furthermore, if the assistance time is not accurately adjusted, the assistive ratio can significantly amplify the negative effects, leading to suboptimal performance. To address this issue in the future, more specific methods should be developed, such as the use of Series Elastic Actuators (SEA) [11], abnormal assistance detection techniques, and other potential solutions.

5.5. Multi-terrain adaptability and challenges

While this study successfully addresses inter-user variability in hip exoskeleton control and demonstrates effective assistance under levelground walking, terrain adaptability, such as walking on inclines or stairs, remains an unsolved challenge. Attempts to train models using public datasets containing multi-terrain data yielded insufficient estimation accuracy for safe control, likely due to variations in biomechanics and limited data quality. Existing studies, such as [44], have conducted experiments on ramps and stairs, but their assistance effectiveness was still mainly observed in level or inclined walking, and relied on task-specific retraining with exoskeleton-worn data, raising concerns of overfitting and poor generalization.

Given the challenges of multi-terrain adaptability, future efforts are expected to focus on scalable data collection and efficient model adaptation techniques.

Table A.1

Candidate Hyperparameters of the GMF generator and the decoder.

Model	Hyperparameter	Candidate values	Selected value
Generator	Number of hidden layers	3, 4, 5	5
	Nodes per layer	16, 32, 64	32
Decoder	Number of hidden layers	3, 4, 5	5
	Nodes per layer	16, 32, 64	32

6. Conclusion

To address the issues of lack of personalization joint moment estimation neural network for hip exoskeleton assistance, this study proposes an estimation method based on GMF generation. The method utilizes a generator to adaptively generate GMF, which serves as a more individual-independent representation compared to the biomechanical joint moment. A GRU-based neural network is then employed to estimate the generated GMF. The GMF improves the estimation accuracy of the estimator while could be decoded into joint moments accurately. Experimental results demonstrate enhanced estimation accuracy and strong generalization capability of the proposed method. Furthermore, real-world exoskeleton tests using encoder-only sensors validate its effectiveness in delivering assistive benefits to users. The contributions of this work advance the feasibility of real-time joint moment estimation and its practical application in exoskeleton control.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We sincerely thank all the students in HARLab for their invaluable support of this work, as well as all the participants in our experiments for their time and effort.

This work was supported by National Key R&D Program of China (2024YFC3082800), National Natural Science Foundation of China (52175272), Guangdong Basic and Applied Basic Research Foundation (2024B1515020008 and 2023B1515130007), and Shenzhen Science and Technology Program (KCFZ20230731093401004, RCYX20231211090345058 and JCYJ20220530114809021).

Appendix A. Candidate hyperparameters of the neural networks

All candidate hyperparameters for the GMF generator and decoder are summarized in Table A.1. The final selected hyperparameters were determined based on the best validation accuracy. During experimentation, both the generator and decoder exhibited a consistent trend: as the number of hidden layers and the number of nodes per layer increased, the validation accuracy improved. However, performance gains saturated beyond a certain point. To ensure a balance between model complexity and generalization capability, we selected the smallest architecture that achieved obvious validation accuracy benefits for the estimator.

The candidate hyperparameters for the four estimators are also listed in Table A.2. In contrast to the GMF generator and decoder, the estimators in this study showed a greater tendency to overfit. Consequently, the models achieving the highest validation accuracy often had excessive complexity and poor generalization. Therefore, the final selected estimator models were chosen not

Table A.2
Candidate and selected hyperparameters for each estimator.

Model	Hyperparameter	Candidate Values	Selected
GRU	Number of hidden layers	1, 2, 4	1
	Nodes per layer	16, 32, 64, 128	16
FCNN	Number of hidden layers	3, 4, 5	3
	Nodes per layer	16, 32, 64, 128	32
CNN	Channel of each layer	(16,32,16), (32,32,32), (32,64,32), (32,64,128,64,32)	(32,64,32)
	Kernel size	$\{3 \times 1, 5 \times 1\}$	3×1
TCN	Number of residual blocks	2, 3, 4	2
	Channels per layer	32, 50, 64	32
	Kernel size	$\{3 \times 1, 5 \times 1\}$	3×1

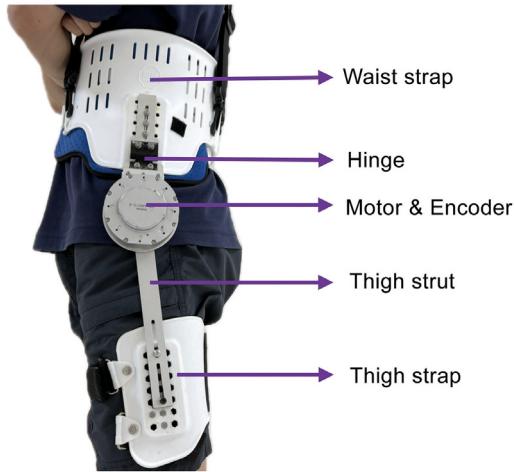


Fig. B.1. Structural components of the hip exoskeleton. Flexible plastic and soft fabric are used for the waist and thigh straps to enhance comfort and compliance, whereas the thigh strut and hinge are made of aluminum alloy to provide mechanical rigidity and durability.

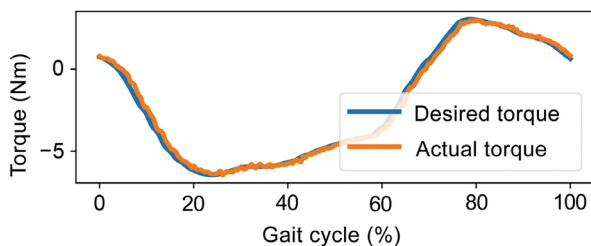


Fig. B.2. Motor torque tracking performance during one gait cycle. The figure shows the desired control torque with blue line and the actual tracked torque with orange line, where the tracking error observed between the two signals is primarily attributed to a control time delay.

solely based on accuracy but also on their robustness against overfitting.

During experiment, all models were trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 128.

Appendix B. Structure of the hip exoskeleton and motor tracking performance

To ensure effective utilization of the joint moment-based exoskeleton controller, the device is designed with a rigid mechanical structure. This rigidity guarantees efficient transmission of torque from the actuators to the user's limbs, minimizing energy losses due to structural compliance. As illustrated in Fig. B.1, the exoskeleton comprises a rigid framework including a waist strap, thigh strap, and a structural thigh brace, all of which securely interface with the user's body. The system integrates both active

and passive joints, consisting of electric motors and mechanical hinges, to deliver controlled assistance for flexion/extension movements and the passive joints allow for natural movement for abduction/adduction. The motors are equipped with encoders to provide real-time feedback on joint angles and angular velocities, enabling precise control of the exoskeleton's assistance.

By computing the tracking error between the desired and actual motor torque, we observed a maximum error of 0.15 Nm, which is less than 2% of the peak assistive torque. This result indicates that the motor is capable of accurately following the desired torque trajectory during the assistance process (see Fig. B.2).

Data availability

All data needed to evaluate the conclusions are presented in the manuscript and/or the Supplementary Materials. Additional data that support the findings of this study are available from the corresponding author upon reasonable request.

References

- [1] D.J. Hyun, H. Park, T. Ha, S. Park, K. Jung, Biomechanical design of an agile, electricity-powered lower-limb exoskeleton for weight-bearing assistance, *Robot. Auton. Syst.* 95 (2017) 181–195.
- [2] W. Cao, C. Chen, D. Wang, X. Wu, L. Chen, T. Xu, J. Liu, A lower limb exoskeleton with rigid and soft structure for loaded walking assistance, *IEEE Robot. Autom. Lett.* 7 (1) (2022) 454–461.
- [3] A. Kapsalyamov, P.K. Jamwal, S. Hussain, M.H. Ghayesh, State of the art lower limb robotic exoskeletons for elderly assistance, *IEEE Access* 7 (2019) 95075–95086.
- [4] B. Kalita, J. Narayan, S.K. Dwivedy, Development of active lower limb robotic-based orthosis and exoskeleton devices: A systematic review, *Int. J. Soc. Robot.* 13 (4) (2021) 775–793.
- [5] R.L. Medrano, E.J. Rouse, G.C. Thomas, Biological joint loading and exoskeleton design, *IEEE Trans. Med. Robot. Bionics* 3 (3) (2021) 847–851.
- [6] D.J. Farris, G.S. Sawicki, The mechanics and energetics of human walking and running: A joint level perspective, *J. R. Soc. Interface* 9 (66) (2012) 110–118.
- [7] J. Kim, G. Lee, R. Heimgartner, D. Arumukhom Revi, N. Karavas, D. Nathanson, I. Galiana, A. Eckert-Erdheim, P. Murphy, D. Perry, N. Menard, D.K. Choe, P. Malcolm, C.J. Walsh, Reducing the metabolic rate of walking and running with a versatile, portable exosuit, *Science* 365 (6454) (2019) 668–672.
- [8] T. Xue, Z. Wang, T. Zhang, M. Zhang, Adaptive oscillator-based robust control for flexible hip assistive exoskeleton, *IEEE Robot. Autom. Lett.* 4 (4) (2019) 3318–3323.
- [9] Y. Qian, Y. Wang, C. Chen, J. Xiong, Y. Leng, H. Yu, C. Fu, Predictive locomotion mode recognition and accurate gait phase estimation for hip exoskeleton on various terrains, *IEEE Robot. Autom. Lett.* 7 (3) (2022) 6439–6446.
- [10] K. Seo, K. Kim, Y.J. Park, J.-K. Cho, J. Lee, B. Choi, B. Lim, Y. Lee, Y. Shim, Adaptive oscillator-based control for active lower-limb exoskeleton and its metabolic impact, in: 2018 IEEE International Conference on Robotics and Automation, ICRA, IEEE, Brisbane, QLD, 2018, pp. 6752–6758.
- [11] Y. Qian, H. Yu, C. Fu, Adaptive oscillator-based assistive torque control for gait asymmetry correction with a nSEA-driven hip exoskeleton, *IEEE Trans. Neural Syst. Rehabil. Eng.* 30 (2022) 2906–2915.

- [12] J. Zhang, C.C. Cheah, S.H. Collins, Experimental comparison of torque control methods on an ankle exoskeleton during human walking, in: 2015 IEEE International Conference on Robotics and Automation, ICRA, IEEE, Seattle, WA, USA, 2015, pp. 5584–5589.
- [13] B. Lim, J. Lee, J. Jang, K. Kim, Y.J. Park, K. Seo, Y. Shim, Delayed output feedback control for gait assistance with a robotic hip exoskeleton, *IEEE Trans. Robot.* 35 (4) (2019) 1055–1062.
- [14] G.M. Gasparri, J. Luque, Z.F. Lerner, Proportional joint-moment control for instantaneously adaptive ankle exoskeleton assistance, *IEEE Trans. Neural Syst. Rehabil. Eng.* 27 (4) (2019) 751–759.
- [15] D.D. Molinaro, I. Kang, A.J. Young, Estimating human joint moments unifies exoskeleton control, reducing user effort, *Sci. Robot.* 9 (88) (2024) eadi8852.
- [16] T.S. Buchanan, D.G. Lloyd, K. Manal, T.F. Besier, Estimation of muscle forces and joint moments using a forward-inverse dynamics model, *Med. Sci. Sport. Exerc.* 37 (11) (2005) 1911–1916.
- [17] S.S.P.A. Bishe, T. Nguyen, Y. Fang, Z.F. Lerner, Adaptive ankle exoskeleton control: Validation across diverse walking conditions, *IEEE Trans. Med. Robot. Bionics* 3 (3) (2021) 801–812.
- [18] R.B.D. Joshi, D. Joshi, MoveNet: A deep neural network for joint profile prediction across variable walking speeds and slopes, *IEEE Trans. Instrum. Meas.* 70 (2021) 1–11.
- [19] M. Mundt, W. Thomsen, T. Witter, A. Koeppe, S. David, F. Bamer, W. Potthast, B. Markert, Prediction of lower limb joint angles and moments during gait using artificial neural networks, *Med. Biol. Eng. Comput.* 58 (1) (2020) 211–225.
- [20] J. Zhang, Y. Zhao, T. Bao, Z. Li, K. Qian, A.F. Frangi, S.Q. Xie, Z.-Q. Zhang, Boosting personalised musculoskeletal modelling with physics-informed knowledge transfer, *IEEE Trans. Instrum. Meas.* (2022) 1–1.
- [21] C. Wang, P.P.K. Chan, B.M.F. Lam, S. Wang, J.H. Zhang, Z.Y.S. Chan, R.H.M. Chan, K.K.W. Ho, R.T.H. Cheung, Real-time estimation of knee adduction moment for gait retraining in patients with knee osteoarthritis, *IEEE Trans. Neural Syst. Rehabil. Eng.* 28 (4) (2020) 888–894.
- [22] D.D. Molinaro, I. Kang, J. Camargo, M.C. Gombolay, A.J. Young, Subject-independent, biological hip moment estimation during multimodal over-ground ambulation using deep learning, *IEEE Trans. Med. Robot. Bionics* 4 (1) (2022) 219–229.
- [23] M.S.B. Hossain, Z. Guo, H. Choi, Estimation of lower extremity joint moments and 3D ground reaction forces using IMU sensors in multiple walking conditions: A deep learning approach, *IEEE J. Biomed. Heal. Inform.* 27 (6) (2023) 2829–2840.
- [24] Y. Zhang, P. Tino, A. Leonardis, K. Tang, A survey on neural network interpretability, *IEEE Trans. Emerg. Top. Comput. Intell.* 5 (5) (2021) 726–742.
- [25] M. Eslamy, M. Rastgaar, Multi-joint leg moment estimation during walking using thigh or shank angles, *IEEE Trans. Neural Syst. Rehabil. Eng.* 31 (2023) 1108–1118.
- [26] D.D. Molinaro, E.O. Park, A.J. Young, Anticipation and delayed estimation of sagittal plane human hip moments using deep learning and a robotic hip exoskeleton, in: 2023 IEEE International Conference on Robotics and Automation, ICRA, IEEE, London, United Kingdom, 2023, pp. 12679–12685.
- [27] Y. Zhang, J. Xiong, Y. Qian, X. Chen, Y. Guo, C. Fu, Y. Leng, Predict hip joint moment using CNN for hip exoskeleton control, in: International Conference on Intelligent Robotics and Applications, Springer, 2023, pp. 200–209.
- [28] P.T. Yamak, L. Yujian, P.K. Gadosey, A comparison between arima, lstm, and gru for time series forecasting, in: Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence, 2019, pp. 49–55.
- [29] K. Cho, B. van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, 2014, arXiv:1409.1259.
- [30] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, arXiv:1412.3555.
- [31] I. Kang, H. Hsu, A. Young, The effect of hip assistance levels on human energetic cost using robotic hip exoskeletons, *IEEE Robot. Autom. Lett.* 4 (2) (2019) 430–437.
- [32] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
- [33] J. Camargo, A. Ramanathan, W. Flanagan, A. Young, A comprehensive, open-source dataset of lower limb biomechanics in multiple conditions of stairs, ramps, and level-ground ambulation and transitions, *J. Biomech.* 119 (2021) 110320.
- [34] D.M. Hawkins, The problem of overfitting, *J. Chem. Inf. Comput. Sci.* 44 (1) (2004) 1–12.
- [35] O.A. Montesinos López, A. Montesinos López, J. Crossa, Overfitting, model tuning, and evaluation of prediction performance, in: Multivariate Statistical Machine Learning Methods for Genomic Prediction, Springer, 2022, pp. 109–139.
- [36] H.G. Kang, J.B. Dingwell, Separating the effects of age and walking speed on gait variability, *Gait Posture* 27 (4) (2008) 572–577.
- [37] K.L. Poggensee, S.H. Collins, How adaptation, training, and customization contribute to benefits from exoskeleton assistance, *Sci. Robot.* 6 (58) (2021) eabf1078.
- [38] P. Kang, J. Li, S. Jiang, P.B. Shull, Reduce system redundancy and optimize sensor disposition for EMG–IMU multimodal fusion human–machine interfaces with XAI, *IEEE Trans. Instrum. Meas.* 72 (2022) 1–9.
- [39] K. Zhu, D. Li, J. Li, P.B. Shull, Real-time IMU-based kinematics in the presence of wireless data drop, *IEEE J. Biomed. Heal. Inform.* (2024).
- [40] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, S.-H. Deng, Hyperparameter optimization for machine learning models based on Bayesian optimization, *J. Electron. Sci. Technol.* 17 (1) (2019) 26–40.
- [41] J.-H. Jacobsen, A. Smeulders, E. Oyallon, I-RevNet: Deep invertible networks, 2018, arXiv:1802.07088.
- [42] A.N. Gomez, M. Ren, R. Urtasun, R.B. Grosse, The reversible residual network: Backpropagation without storing activations, *Adv. Neural Inf. Process. Syst.* 30 (2017).
- [43] L. Dinh, D. Krueger, Y. Bengio, NICE: Non-linear independent components estimation, 2015, arXiv:1410.8516.
- [44] D.D. Molinaro, K.L. Scherpereel, E.B. Schonhaut, G. Evangelopoulos, M.K. Shepherd, A.J. Young, Task-agnostic exoskeleton control via biological joint moment estimation, *Nature* 635 (8038) (2024) 337–344.