2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

User-Tailored Learning to Forecast Walking Modes for Exosuits

Gabriele Abbate[1], Enrica Tricomi[2], Nathalie Gierden[2], Alessandro Giusti[1], Lorenzo Masia[2], Antonio Paolillo[1]

***Abstract*— Assistive robotic devices, like soft lower-limb exoskeletons or exosuits, are widely spreading with the promise of helping people in everyday life. To make such systems adaptive to the variety of users wearing them, it is desirable to endow exosuits with advanced perception systems. However, exosuits have little sensory equipment because they need to be light and easy to wear. This paper presents a perception module based on machine learning that aims at estimating 3 walking modes (i.e., ascending or descending stairs and walking on level ground) of users wearing an exosuit. We tackle this perception problem using only inertial data from two sensors. Our approach provides an estimate for both future and past timesteps that supports control and enables a self-labeling procedure for online model adaptation. Indeed, we show that our estimate can label data acquired online and refine the model for new users. A thorough analysis carried out on real-life datasets shows the effectiveness of our user-tailored perception module. Finally, we integrate our system with the exosuit in a closed-loop controller, validating its performance in an online single-subject experiment.**

Fig. 1. We equip an easy-to-wear and light exosuit with a user-tailored perception model estimating the current and intended walking gait (ascending or descending stairs and walking on level ground) using only IMU readings.

I. INTRODUCTION

Soft exosuits are rapidly expanding as rehabilitation and assistive devices for their lightness, ease of use, and effectiveness in improving walking endurance [1]–[4]. To fulfill their goal, exosuits must have advanced perception skills, e.g. to enable adaptive behavior and cope with the variety of users wearing them [5]. Furthermore, it is desirable to provide exosuits' controllers with predictive capabilities to enable real-time adjustments to users' intention and improve responsiveness [6] and user experience [7].

However, light exosuits have inherently scarce sensory equipment, which makes the development of perception modules a challenge. To this end, Artificial Intelligence (AI) and Machine Learning (ML) offer suitable tools to extract meaningful information from scarce, noisy, and raw sensor data. ML is commonly used for robot perception, for example, in the domain of human-robot interaction [8], or rehabilitation robotics [9]. In assistive robotics, ML is used to detect walking modes (e.g., walking on level ground or ascending/descending stairs) to adapt the control of underactuated exosuits to the user's motion. However, many works rely on additional hardware which can either be complex to integrate into a lightweight device (e.g., load cell and pressure insole sensors [10], [11]) or have technical limitations (like the approaches using cameras [12] that suffer from occluded views and difficult light conditions). In this regard, Inertial Measurement Unit (IMU) sensors are a sensible option, since they are lightweight and can be effectively deployed to detect changes in walking patterns [13]. Even if their measurements are prone to angular drift [14], ML methods can be used on IMU orientation data to classify the walking modes, as presented in a previous work [15]. In this paper, we provide an ML method to classify 3 walking modes of users wearing an exosuit using only IMU measurements, see Fig. 1. With respect to Zhang et al. [15], we propose to: (i) reduce the number of sensors from 3 to 2, to make the system even lighter; (ii) instead of a Random Forest (RF) we rely on a Temporal Convolutional Network (TCN) architecture for its capability of handling raw temporal data, without the need for handcrafting relevant features; (iii), we extend the model to produce estimates not only at the current time but within a window extending a few seconds in the future and the past.

Another crucial challenge for assistive wearable devices is their adaptability to new users, which is key to their effectiveness [16]. This capability is difficult to implement with traditional model-based control methods since they rely on predefined dynamics that can vary significantly from one individual to another [17]. At the same time, ML-based approaches require a substantial amount of data to ensure robust performance across different users [18], [19]. To achieve this capability, our pre-trained perception model can be fine-tuned on a new user with data acquired *during* the operation of the device, without any explicit supervision. In particular, pseudo-labels are derived for a given timestep *in hindsight*,

This work was supported by Innosuisse - Swiss Innovation Agency, through the VRHEM project (100.533 IP-ICT), by the Swiss National Science Foundation (grant number 213074), by the Multidimension AI Project from the Carl-Zeiss Foundation (P2022-08-010), and by the Istituto nazionale per l'assicurazione contro gli infortuni sul lavoro (INAIL) under grant agreement PR23-RR-P1 FeatherEXO.

[1]Dalle Molle Institute for Artificial Intelligence (IDSIA), USI-SUPSI, Lugano, Switzerland name.surname@idsia.ch

[2]Munich Institute for Robotics and Machine Intelligence (MIRMI), Department of Computer Engineering, School of Computation, Information and Technology, Technical University of Munich (TUM), Munich, Germany.

exploiting data collected in the following few seconds. These labels are saved in a user-specific dataset and used to adapt the model to the user automatically. This approach is a form of *Self-Supervised Robot Learning*, which allows a robot to collect its training (or fine-tuning) data without external supervision. This paradigm was initially adopted in robotics for segmentation of traversable terrain [20]–[22], then applied to other tasks such as grasping [23], [24] and long-range sensing for navigation [25]–[27].

In summary, we present the following contributions:

- a TCN architecture to classify 3 walking modes.
- an approach to estimate walking modes within a time window including past, current, and future timesteps.
- a self-labeling procedure that allows the model to adapt to new users wearing the exosuit.

Experimental results on a multi-user dataset show that: (i) the TCN-based architecture outperforms previous work [15]; (ii) forecasts of future values of the target variable outperform repeating the prediction for the current timestep, showing that our prediction is solid and can be useful for control purposes; (iii) past estimates make our self-labeling procedure effective at adapting the model's performance to a new user without any external supervision. Finally, we integrate the classifier with the exosuit's closed-loop controller in an online experiment, validating its capability with a single subject.

II. Problem formulation and approach

A. Walking mode classification

We propose an ML-based solution that learns, from previous walking experiences, the user's past, current, and future behavior. Our approach is a classifier estimating three walking modes that humans use when moving in indoor and urbanized environments, i.e. Stair Ascent (SA), walking on the Level Ground (LG), and Stair Descent (SD). Such walking modes are represented, at the current time sample k, by the following discrete variable:

$$c_k \in \{\mathrm{SA}, \mathrm{LG}, \mathrm{SD}\}. \tag{1}$$

We aim to estimate, at each timestep k, the walking class in a *target window* ranging from N timesteps before to N after the current time. More in detail, the vector

$$\boldsymbol{y}_k = (c_{k-N}, \ldots, c_k, \ldots, c_{k+N})^\top \in \mathbb{R}^{2N+1} \tag{2}$$

is the target variable we want to estimate; $2N+1$ is the target window size. Past estimates (from $k-N$ to $k-1$) are relevant to assess the perception performance and fine-tune the process with Self-Supervised Learning (SSL) paradigms, as introduced in Sec. II-C. Past, current, and future estimates (from $k-N$ to $k+N$), instead, are useful for control, as explained in the context of our setup (Sec. III-B).

Our classifier is fed with the information about the user's motion provided by the minimal sensory equipment of light exosuits (e.g., thigh-mounted IMUs). For each timestep k, such information is collected in the feature vector $\boldsymbol{f}_k \in \mathbb{R}^f$.

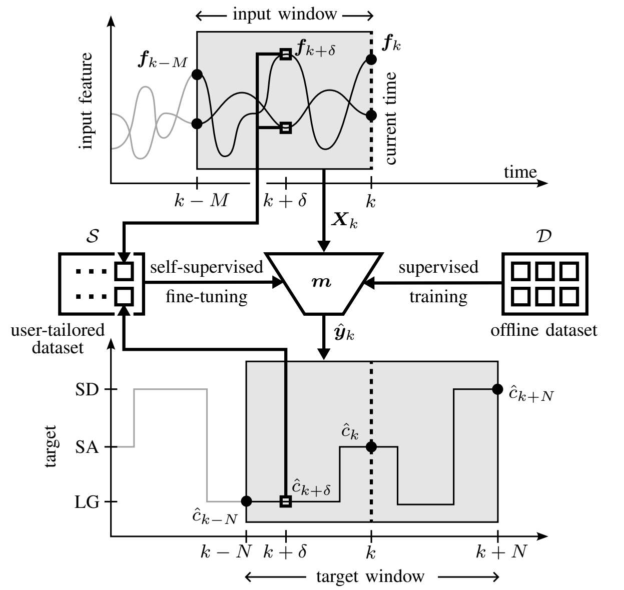

Fig. 2. Proposed approach: at each timestep k, the model $\boldsymbol{m}$ is fed with an input window of the last $M+1$ samples and outputs an estimate of walking modes in a target window including N timesteps ahead and behind; δ marks the timestep in the target window where $\boldsymbol{m}$ performs best. The estimate $\hat{c}_{k+\delta}$ is used to pseudo-label the corresponding input $\boldsymbol{f}_{k+\delta}$ and refine $\boldsymbol{m}$.

Our classifier uses as input the last $M+1$ samples organized in the input matrix $\boldsymbol{X}$:

$$\boldsymbol{X}_k = \left(\boldsymbol{f}_{k-M}^\top, \boldsymbol{f}_{k-M+1}^\top, \ldots, \boldsymbol{f}_k^\top\right)^\top \in \mathbb{R}^{f \times (M+1)} \tag{3}$$

being $M+1$ the size of the so defined *input window*.

Our classification model $\boldsymbol{m}$, for each timestep k, takes in input the matrix $\boldsymbol{X}_k$ and gives an estimate of the vector $\boldsymbol{y}_k$:

$$\hat{\boldsymbol{y}}_k = \boldsymbol{m}(\boldsymbol{X}_k | \mathcal{D}), \tag{4}$$

where the hat over the variable denotes the approximation of the true value due to the estimation process; $\mathcal{D}$ is the dataset used to train the classifier (see Sec. II-B). Optionally, the model $\boldsymbol{m}$ can be further refined by augmenting the training set with data acquired online and self-labeled following the procedure detailed in Sec. II-C.

B. Offline supervised training

To train and test the classifier (4), we organize data in sequences, each consisting of a series of the following pairs:

$$\mathcal{D} = \{\bar{c}_{i,j}, \boldsymbol{f}_{i,j}\}_{i=1,j=1}^{L_j,S} \tag{5}$$

where $\bar{c}$ denotes the label, i.e. the ground truth values of the walking class that are available during the data acquisition; i and j indicate the number of samples and sequences, respectively; S is the number of data sequences, while L_j is the length (expressed in timesteps) of the j-th sequence.

The training of the model (4) is performed offline by minimizing the distance between the model's output and the labels and using a training subset of data $\mathcal{D}$. Furthermore, offline performance analysis can be carried out on a testing subset of $\mathcal{D}$ (different from the training set). This analysis allows us to pick a $\delta \in [-N, N]$ such that $k+\delta$ marks the

Algorithm 1 User-tailored self-supervised fine-tuning

$\boldsymbol{m} \leftarrow \text{train}(\mathcal{D})$
$\delta \leftarrow \text{evaluate}(\boldsymbol{m})$
for a new sequence **do**
  $\mathcal{S} = \emptyset$
  while $k \leq H$ **do**
    $\boldsymbol{X}_k \leftarrow \text{update}(\boldsymbol{f}_k)$
    $\hat{\boldsymbol{y}}_k = \boldsymbol{m}(\boldsymbol{X}_k|\mathcal{D})$
    $\hat{c}_{k+\delta} \leftarrow \text{select}(\hat{\boldsymbol{y}}_k, \delta)$
    $\mathcal{S} \leftarrow \text{add}(\hat{c}_{k+\delta}, \boldsymbol{f}_{k+\delta})$
  end while
  $\boldsymbol{m} \leftarrow \text{retrain}(\mathcal{D} \cup \mathcal{S})$
end for

timestep in the target window where the classifier yields the best performance averaged across all samples. It is expected that $\delta < 0$, i.e. that the best performance is obtained within the first half of the target window, where the model can leverage input features before and after $k + \delta$, see Fig. 2.

C. User-tailored self-supervised fine-tuning

The a-posteriori estimates provided by our classifier represent a precious source of supervision for novel sensory data acquired online. Indeed, the estimate $\hat{c}_{k+\delta} \in \hat{\boldsymbol{y}}_k$ picked from the target window produced at time k can be taken as the pseudo-label for the corresponding input feature vector $\boldsymbol{f}_{k+\delta}$ and create a new dataset (see Fig. 2 for a visual reference). Assuming that the new sequence acquired online has H samples, such a new dataset $\mathcal{S}$ is composed of the pairs:

$$\mathcal{S} = \{\hat{c}_{k+\delta}, \boldsymbol{f}_{k+\delta}\}, \quad k = 1, \dots, H \tag{6}$$

which can be used to augment the training set $\mathcal{D}$ and retrain the model $\boldsymbol{m}$. Such a self-labeling mechanism is explained in detail in Algorithm 1. First, we train a model $\boldsymbol{m}$ on the available dataset and evaluate its performance over the target window to pick a suitable δ. Then, we consider data from a new user and for each timestep k, we use $\boldsymbol{m}$ to estimate the target window $\hat{\boldsymbol{y}}_k$. From $\hat{\boldsymbol{y}}_k$, we pick the estimate $\hat{c}_{k+\delta}$ to use as pseudo-label of the input features at the corresponding timestep, i.e. $\boldsymbol{f}_{k+\delta}$. Once the new sequence is fully labeled, we add it to our dataset and use it to fine-tune $\boldsymbol{m}$.

III. Experimental setup

A. Exosuit

We use a hip exosuit designed to assist hip flexion during the swing phase of walking using tendon-driven transmissions [12] (Fig. 3). The device weighs 2.7 kg and consists of a waist belt secured at the user's body and two textile thigh harnesses. The belt has one actuation stage for each leg, a power supply (RED POWER battery, 18.5 V, 3500 mAh, 25 C), and a control unit. Each actuation stage is driven by a compact flat brushless motor (T-Motor, AK60-6, 24 V, 6:1 planetary gear-head reduction, Cube Mars actuator, TMOTOR, Nanchang, Jiangxi, China) connected to a 35 mm diameter pulley. Assistive forces are transmitted to the user's thighs through artificial tendons made of Black Braided Kevlar Fiber (KT5703-06, maximum load capacity of 2.2 kN, Loma Linda, CA, USA). The tendons run from the actuation stages at the rear of the belt to the front, where they attach distally to the thigh harnesses via 3D-printed anchor points.

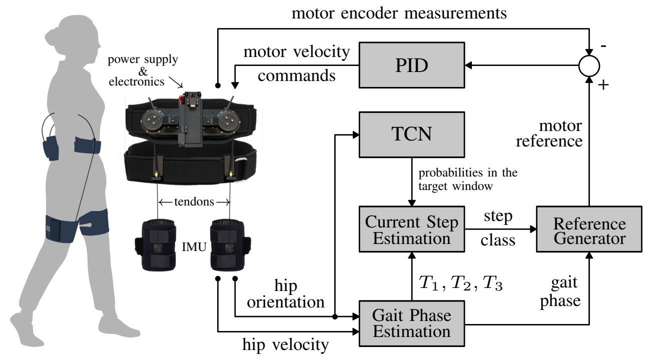

Fig. 3. Exosuit's main components and control architecture.

The real-time control system runs on an Arduino MKR 1010 WiFi (Arduino, Ivrea, Italy), which manages gait phase estimation and actuator control at 100 Hz. The perception model is deployed on a Jetson Orin Nano (NVIDIA, Santa Clara, CA, USA), where it performs inference of the walking condition. Hip joint kinematics are captured by two IMUs (Bosch, BNO055, Gerlingen, Germany), mounted laterally on the thigh harnesses, which communicate with the control unit via Bluetooth Low Energy modules (Feather nRF52 Bluefruit, Adafruit).

B. Closed-loop controller

We integrate the proposed perception module with the exosuit's controller described in a previous work [28] and skematically depicted in Fig. 3. During the walking, a *Gait Phase Estimator* uses the hip orientation and velocity provided by the IMUs to determine the start of the stance phase (T_1), the transition to the swing phase (T_2), and the end of the swing phase (T_3), taking into account that the swing phase is about 40% of the entire gait cycle [29]. When T_2 is detected, the *Current Step Estimation* block averages the class probabilities returned by the proposed walking mode classifier (implemented as a *TCN*, see Sec. III-D) in a time window that spans from T_1 (in the past) to T_3 (in the future).[1] The class yielding the highest value is used to classify the current step (as SA, LG, or SD). Such a step class, together with the gait phase also provided by the Gait Phase Estimation, is used by a *Reference Generator* block to modulate the amplitude of the motor reference position, applying different assistance gains to match the physiological change in kinematics [30]. A PID control loop takes the reference motor signal to compute the commands for the exosuit's actuation system. Note that during the stance phase (i.e., between T_1 and T_2), the motor cable is slightly released to allow the legs to extend, then it is actuated as explained above during the swing phase.

[1] If T_1 or T_3 fall outside the target window, we consider only the data included within it.

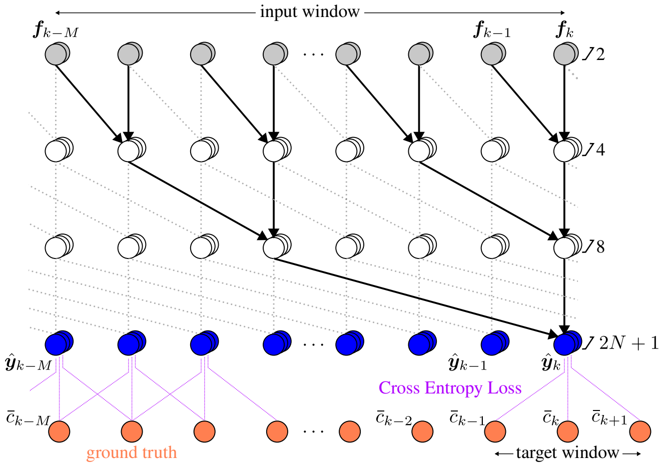

Fig. 4. A simplified architecture of our TCN model with the kernel size set to 2 and $N = 1$. Two hidden layers output 4 and 8 feature channels, respectively. The dotted gray lines represent dilated causal convolutions. Black arrows highlight those producing $\hat{\boldsymbol{y}}_k$ (rightmost blue circle). This is the target window estimate resulting from the two-channel input window (gray circles on top) containing $\boldsymbol{f}$ from time k back to $k - M$.

C. Dataset collection

The data collection campaign involved five young, healthy participants, with an average age of 25 ± 3 years, height of 173.9 ± 12.0 cm, and weight of 66.6 ± 16.0 kg. Research procedures were performed according to the Declaration of Helsinki and approved by the Ethical Committee of Heidelberg University (resolution S-313/2020).

Each participant completed 6 walking sequences, totaling 30 sequences across all users. Every sequence contains instances of all three walking modes (LG, SA, and SD). In half of the sequences, we kept the exosuit off, while in the other half, we actuated it to provide constant assistance, regardless of the walking mode. This is done to make data more representative of a case where a perception model is used in a closed-loop fashion to control the exosuit.

For each condition (i.e., exosuit on and off): (i) the first sequence consists in ascending 10 consecutive stair ramps (10 steps each), walking $140\,\mathrm{m}$ on level ground, and descending the 10 stair ramps, (ii) the second sequence consists in descending the 10 stair ramps, walking $40\,\mathrm{m}$ on level ground, and ascending the 10 stair ramps, (iii) the last sequence, involves descending 8 stair ramps, walking $70\,\mathrm{m}$ on level ground, ascending 2 ramps, descending 2 ramps, walking another $70\,\mathrm{m}$, and finally ascending 8 stair ramps.

We randomized the mentioned sequences and conditions across subjects to avoid order effects. Along the path, we recorded the left and right thighs' sagittal plane angles using IMUs. These IMU orientations constitute the feature vector $\boldsymbol{f}_k$. At the same time, we recorded the ground truth signal through an external manual switch used to segment the path (i.e. LG, SA, and SD) by the user, which corresponds to $\bar{c}_k$.

D. Classification architectures

We use a TCN [31] to implement our approach described in Sec. II. TCNs are a family of architectures used to solve classification problems with sequential data, as an alternative to Recurrent Neural Network (RNN). We opt for a TCN architecture instead of RNN, e.g. the Long Short-Term Memory, as they perform better on many sequence-based tasks [31]. Furthermore, their low memory footprint and computational complexity allow the deployment for real-time control even onboard low-powered devices [32]. Finally, we chose TCNs also because they can be re-trained quickly—a key aspect for our SSL mechanism, especially when compared to other architectures such as Transformer (TF) [33], as detailed below.

TCNs perform causal convolutions, i.e., the output at a given time is convolved only with elements from earlier times in the previous layer. In this way, no information can leak from the future into the past. These convolutions are also increasingly dilated by a factor that determines the spacing between filter elements. The dilation factor allows for capturing long-range dependencies without increasing the network's depth. Tuning the dilation factor and the convolution kernel sizes enables a flexible configuration of the model receptive field (i.e. the input window size defined in Sec. II).

We adapt an existing Pytorch implementation [34]. A simplification of our architecture is depicted in Fig. 4. We concatenate 3 residual blocks with increasing feature channels to apply a dilated causal convolution. The dilation factors are set to 1, 2, and 4, respectively, and the kernel size to 5, meaning that the input window size is 57 samples. As the input features are collected at $30\,\mathrm{Hz}$, the result is a receptive field of about $2\,\mathrm{s}$. Our model's inputs are two IMU orientation signals as described in III-C. For each input sample, the model outputs a target window of length $2N+1$, as explained in Sec. II. We set $N = 60$, i.e. a target window estimating $2\,\mathrm{s}$ in the past and predicting $2\,\mathrm{s}$ in the future. Each window element contains unnormalized class scores for each of the 3 walking modes. Such scores are used, together with the labels, to compute the cross-entropy loss while training the model. At inference, the softmax function is applied to produce probability scores for each class; the highest one is used to classify the sample and produce the vector $\hat{\boldsymbol{y}}_k$.

For comparison purposes, we also implement a RF classifier inspired by previous work [15], and a model based on the encoder–decoder TF [33]. For the RF, although we consider only 2 input signals (instead of 6), we compute the same handcrafted features. In particular, we consider an input window of 60 samples and we obtain 6 features from each signal, respectively: first, last, minimum, maximum, average, and standard deviation values. Note that this classifier can only produce the output at the current time (i.e., it can provide an estimate of c_k in (2), but not the full vector $\hat{\boldsymbol{y}}_k$) and receives a handcrafted feature set that summarizes a window of past data. Further details of the RF implementation are in Sec. IV. The TF model is a drop-in replacement of the TCN model, being trained to process the same input and produce the same output. The TF implementation roughly shares the same number of parameters as the TCN, but is slower in the training process. In fact, the TCN can process the entire

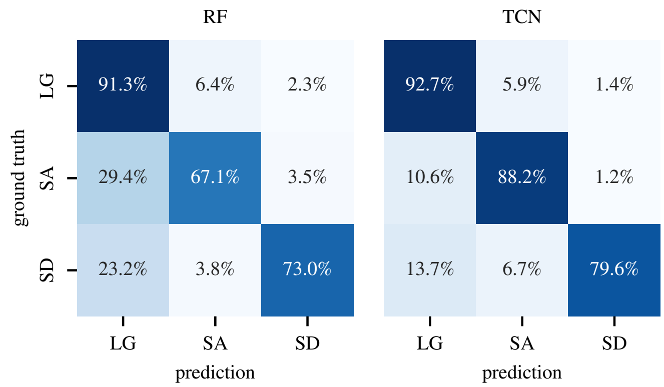

Fig. 5. Confusion matrix for the RF (left) and TCN approach (right).

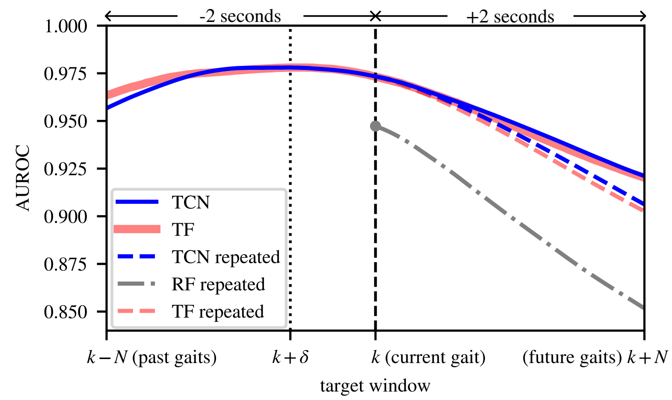

Fig. 6. Models' performance for each timestep in the target window: AUROC of the TCN (solid blue) and TF (solid red); TCN, TF and RF repeating the estimate at k in the future (dashed blue, red and gray dash-dotted lines, respectively). The dotted black line at timestep $k+\delta$ corresponds to the peak of performance for the TCN model.

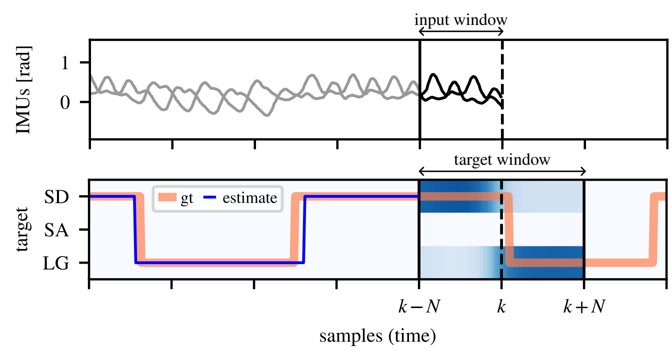

Fig. 7. TCN inference for a 14 s time-frame: IMUs signals composing the feature vector, with the input window (top plot, gray lines) and prediction of the model compared against the ground truth (bottom plot, blue and orange lines respectively). In the target window, the probabilities of each walking class are depicted with blue shades (the darker, the higher).

training dataset 10 times faster than the TF (1 s per epoch on a consumer-grade GPU instead of 10 s).

E. Evaluation metrics

We use the Area Under the Receiver Operating Curve (AUROC) to evaluate the performance of our classifiers. In particular, we use the multi-class AUROC using the one-vs-rest approach, computing the AUROC of each class against the rest and averaging the result across all classes. The AUROC does not depend on thresholds. It ranges between 0.5 for a non-informative classifier and 1.0 for an ideal one. We also report the confusion matrix, which summarizes predicted versus ground truth values and highlights misclassifications of the model.

IV. Results

A. Estimation at the current time

We evaluate the performance of our TCN approach at estimating which walking mode (i.e., LG, SA or SD) is performed by a user wearing the exosuit. The performances are compared against an RF approach inspired by [15], which can only produce an estimate for the current time k, as explained in Sec. III-D. For this comparison, we consider only the estimate that our TCN produces for time k, instead of the entire target window. Also, since users may exhibit distinct walking patterns, it is valuable to analyze each individual separately, while also evaluating the model's capacity to generalize to previously unseen users. To this end, we apply a leave-one-user-out cross-validation procedure for both models. In each validation fold, the data of one user is excluded from training, and the AUROC is computed on that user's data. The averaged AUROC across all users is 0.925 for the RF and 0.962 for the TCN, confirming the superior performance and generalization capability of the proposed model. Figure 5 further illustrates this by showing the aggregated confusion matrices: the TCN notably reduces the misclassifications of the RF, particularly for the SA and SD classes. To directly compare the models on a per-user basis, we use the Wilcoxon signed-rank test. This non-parametric paired test is appropriate because it does not assume normality of the AUROC distributions and evaluates whether the differences observed for the same users are systematic. The results confirm that the TCN yields significantly higher AUROC values than the RF ($p = 0.03125$).

B. Estimation in a time window

Our approach can provide estimation both in advance and back in time. In practice, we use our framework to predict the future walking modes that will happen in a time window of 2 s and the past modes that just occurred in the last 2 s.

For this validation, we compute the AUROC averaged among all users as done in Sec. IV-A but for all the instants in the target time window, see Fig. 6. The plot shows that $\delta = -10$ marks the timestep in the target window at which the classifier performs best, confirming our speculation from Sec. II-B. For reference, we also report the performance of the TF based model described in Sec. III-D. As expected, all models lose performance as predictions move forward from k, i.e. as they forecast further in the future. However, performance values remain high (AUROC greater than 0.9) for both models in the entire window. Furthermore, in estimating the future, the proposed approach outperforms three trivial baselines built by repeating the prediction at time

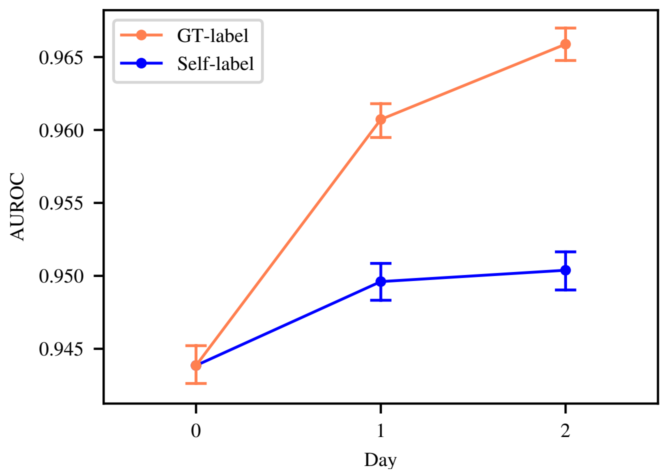

Fig. 8. Models' performance over time: incrementally adding labeled data to the training set improves the AUROC (orange line) even using our self-labeling procedure (blue). Error bars denote 95% confidence intervals.

k from $k+1$ up to $k+N$ for the TCN, TF and RF models respectively. This confirms that the model captures hidden patterns in the data that predict the user's future behavior (that might be due to users' subtle adjustment motions before the stairs begin) and produces meaningful predictions, rather than learning to predict a constant value for future timesteps. Finally, the plot shows that the choice between the TCN and the TF models is arbitrary, since their performance is similar. However, we select the TCN for our implementation since it is more efficient to train, as explained in Sec.III-D. This is relevant for our plans of fine-tuning the model onboard the exosuit device to quickly adapt to new users, leveraging our self-supervised pipeline.

Figure 7 shows a time frame of the TCN running over a sequence. The top plot displays the IMU orientations, highlighting the input window in black. The bottom plot shows the ground truth in orange for each timestep, along with the model prediction for the current time k in solid blue. In addition, a blue colormap displays the 3 probabilities returned by the model for each timestep in the target window. Darker blue corresponds to a higher probability for the corresponding class (on the y-axis). The blue solid line and the target window closely match the ground truth values. Such a plot qualitatively confirms the classifier's good performance.

The performance of our TCN in estimating the walking mode in a time window is qualitatively shown in the video accompanying the paper.

C. Self-supervised learning

We simulate a situation where a new user wears the exosuit over three days, and each day new walking sequences are recorded. Additional data can be used to refine the model and improve the performance, as explained in Sec. II-C. We take the dataset recorded as described in Sec. III-C, containing walking sequences from 5 users. Each user is kept out in turn as the *new user*, then the following protocol is applied:

- at day 0, we train a model using the data of 6 sequences collected by each of the 4 users. We compute the performance on 2 sequences from the new user;
- at day 1, we augment the training set by adding the 2 sequences for the new user from day 0. Testing is performed on 2 new sequences from the same user.
- at day 2, we increase again the training set with the 2 sequences for the new user considered in day 1. Testing is performed on 2 new sequences from the same user.

For each user, we perform 6 simulation runs, one for each permutation of the available sequences (i.e. 6 per user), taken in pairs. In total, we simulate 30 runs. The resulting performances are averaged across all the runs for each simulated day. Furthermore, we run the experiment twice, changing the way we label new data added to the training set each day. In the second run, we use the estimates obtained by the model trained the previous day (leveraging the self-labeling procedure of Sec. II-C). The results are shown in Fig. 8: the addition of novel data from a user improves performance on new data collected by the same user; as expected, the improvement is larger when the new data is labeled with ground truth. However, performance also improves when the proposed self-labeling procedure is adopted, which requires no external supervision and can be performed automatically.

D. Closed-loop assistive robotics experiment

We run an online experiment to qualitatively assess our classifier's performance and demonstrate its practical use in closed-loop within the exosuit's controller. The system was tested on a 28-year-old female participant walking at self-selected speed along a sequence including all three modes (SA, LG, SD). The sequence consisted of 91 ascending stairs, $98\,\mathrm{m}$ on level ground, and 72 descending stairs. During the trial, IMU signals, the motor reference position command, final classifications for both legs, and ground truth were recorded. Ground truth was obtained from a manual switch pressed by the participant when changing modes. We deployed a model trained on the dataset described in Sec. III-C, expanded with as little as $10\,\mathrm{min}$ of data previously collected with the new testing user. The results are qualitatively presented in Fig. 9, demonstrating the performance of our approach in a real-life scenario. The motor reference amplitudes increased during stair ascent and decreased during stair descent according to the perception module classification, which closely matches the ground truth walking mode.

V. Conclusions and Future Work

We have proposed a machine learning approach to estimate the walking modes of a user wearing a lightweight exosuit. The approach consists of a classifier implemented using a temporal convolutional network, taking the input of two IMUs and estimating the walking modes in a time window, which includes past and future time steps. A thorough analysis has shown the performance of our approach at improving previous implementations, forecasting future walking modes, and computing past estimates to enable our effective self-labeling procedure. An online proof of concept integrating

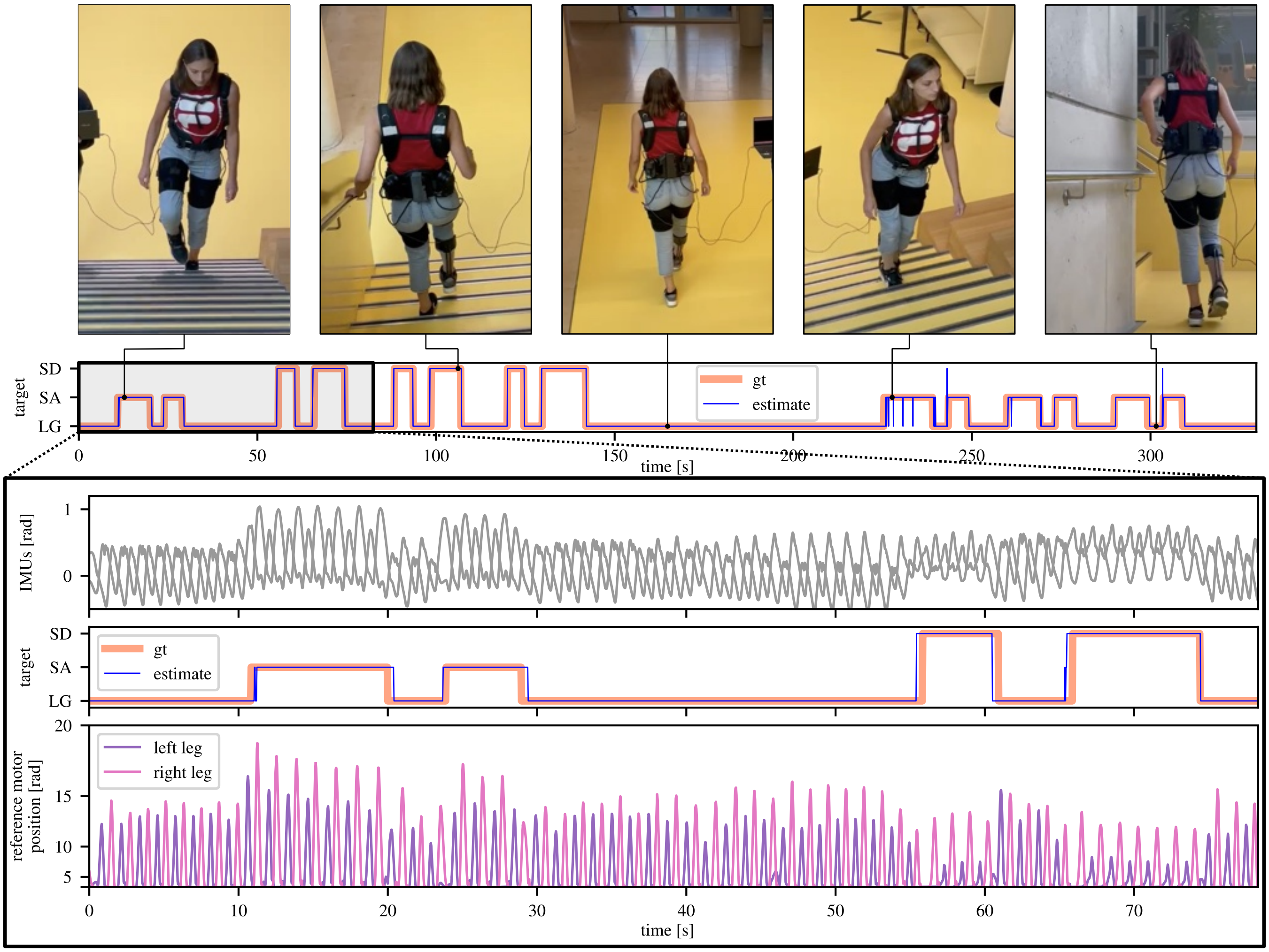

Fig. 9. Snapshots of an experiment running our perception model integrated in closed-loop with the exosuit's controller. The top plot shows the recorded ground truths for the walking mode (orange) and the model's classifications (blue). The call-out rectangle at the bottom shows in detail the first part of the experiment, in particular: input features signals (top plot), walking mode ground truth and estimates (middle), and reference motor positions (bottom). The purple and pink lines show that the amplitudes of the motor references are correctly modulated according to the current walking mode.

the classifier in closed-loop with the exosuit's controller has shown high performance with a single subject experiment. Overall, these results highlight the potential of user-tailored machine learning to enhance adaptability and usability in wearable assistive devices. However, the study is not without limitations, which in turn point toward promising avenues for future research. The current validation involved a relatively small pool of healthy subjects, which restricts generalization. Extending the framework to a larger and more diverse population, including users with mobility impairments, is a crucial next step. This would enable reporting quantitative measures such as inter-subject variability and prediction latency to better characterize real-time performance. Moreover, while we demonstrated reliable control of the exosuit, we did not yet quantify whether the provided assistance improves walking performance or comfort, an aspect that will require dedicated evaluation metrics. The set of walking modes considered was limited to level walking and stairs, but our system is readily extensible to additional modes (e.g., ramps), enabling evaluation in more complex locomotion tasks. Also, the model fine-tuning based on self-supervision has been tested only offline. Migrating the procedure fully onboard will facilitate field testing and user adaptation in real-world conditions. Future work will investigate confidence-based pseudo-label selection and corresponding retraining strategies to improve robustness and reproducibility in online adaptation. Finally, we plan to integrate lightweight auxiliary sensors (e.g., a barometer) for retrospective data labeling, as their long-term accuracy can improve training, even though their short-term noise limits their use for online walking mode prediction.

References

[1] M. Xiloyannis, R. Alicea, A.-M. Georgarakis, F. L. Haufe, P. Wolf, L. Masia, and R. Riener, "Soft robotic suits: State of the art, core technologies, and open challenges," *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1343–1362, 2021.

[2] J. Kim, B. T. Quinlivan, L.-A. Deprey, D. Arumukhom Revi, A. Eckert-Erdheim, P. Murphy, D. Orzel, and C. J. Walsh, "Reducing the energy cost of walking with low assistance levels through opti-

mized hip flexion assistance from a soft exosuit," *Scientific reports*, vol. 12, no. 1, p. 11004, 2022.

[3] M. K. Ishmael, D. Archangeli, and T. Lenzi, "Powered hip exoskeleton improves walking economy in individuals with above-knee amputation," *Nature Medicine*, vol. 27, no. 10, pp. 1783–1788, 2021.

[4] S. Hood, S. Creveling, L. Gabert, M. Tran, and T. Lenzi, "Powered knee and ankle prostheses enable natural ambulation on level ground and stairs for individuals with bilateral above-knee amputation: a case study," *Scientific Reports*, vol. 12, no. 1, p. 15465, 2022.

[5] P. Slade, M. J. Kochenderfer, S. L. Delp, and S. H. Collins, "Personalizing exoskeleton assistance while walking in the real world," *Nature*, vol. 610, no. 7931, pp. 277–282, 2022.

[6] A. Jayakumar, D. Jorge, J. Bermejo-García, R. Agujetas, and F. Romero, "Sensing and control strategies for a synergy-based, cable-driven exosuit via a modular test bench," *Sensors*, vol. 23, p. 4713, 2023.

[7] Y. Lu, Y. Aoustin, and V. Arakelian, "Optimization of design parameters and improvement of human comfort conditions in an upper-limb exosuit for assistance," *Multibody System Dynamics*, vol. 62, pp. 433–461, 2024.

[8] G. Abbate, A. Giusti, V. Schmuck, O. Celiktutan, and A. Paolillo, "Self-supervised prediction of the intention to interact with a service robot," *Robotics and Autonomous Systems*, vol. 171, p. 104568, 2024.

[9] M. Yip, S. Salcudean, K. Goldberg, K. Althoefer, A. Menciassi, J. D. Opfermann, A. Krieger, K. Swaminathan, C. J. Walsh, H. H. Huang, and I.-C. Lee, "Artificial intelligence meets medical robotics," *Science*, vol. 381, no. 6654, pp. 141–146, 2023.

[10] Y.-L. Park, B.-r. Chen, N. O. Pérez-Arancibia, D. Young, L. Stirling, R. J. Wood, E. C. Goldfield, and R. Nagpal, "Design and control of a bio-inspired soft wearable robotic device for ankle–foot rehabilitation," *Bioinspiration & biomimetics*, vol. 9, no. 1, p. 016007, 2014.

[11] X. Liu and Q. Wang, "Real-time locomotion mode recognition and assistive torque control for unilateral knee exoskeleton on different terrains," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 6, pp. 2722–2732, 2020.

[12] E. Tricomi, M. Mossini, F. Missiroli, N. Lotti, X. Zhang, M. Xiloyannis, L. Roveda, and L. Masia, "Environment-based assistance modulation for a hip exosuit via computer vision," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2550–2557, 2023.

[13] K. Yuan, Q. Wang, and L. Wang, "Fuzzy-logic-based terrain identification with multisensor fusion for transtibial amputees," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 618–630, 2014.

[14] C. Siviy, L. M. Baker, B. T. Quinlivan, F. Porciuncula, K. Swaminathan, L. N. Awad, and C. J. Walsh, "Opportunities and challenges in the development of exoskeletons for locomotor assistance," *Nature Biomedical Engineering*, vol. 7, no. 4, pp. 456–472, 2023.

[15] X. Zhang, E. Tricomi, F. Missiroli, N. Lotti, and L. Masia, "Real-time assistive control via IMU locomotion mode detection in a soft exosuit: An effective approach to enhance walking metabolic efficiency," *IEEE/ASME Transactions on Mechatronics*, vol. 29, no. 3, pp. 1797–1808, 2024.

[16] K. L. Poggensee and S. H. Collins, "How adaptation, training, and customization contribute to benefits from exoskeleton assistance," *Science Robotics*, vol. 6, no. 58, p. eabf1078, 2021.

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

[18] L. Rose, M. C. Bazzocchi, and G. Nejat, "A model-free deep reinforcement learning approach for control of exoskeleton gait patterns," *Robotica*, vol. 40, pp. 2189–2214, 2021.

[19] I. Kang, D. D. Molinaro, G. Choi, J. Camargo, and A. J. Young, "Subject-independent continuous locomotion mode classification for robotic hip exoskeleton applications," *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 10, pp. 3234–3242, 2022.

[20] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. R. Bradski, "Self-supervised monocular road detection in desert terrain," in *Robot. Sci. and Syst.*, 2006.

[21] D. Stavens and S. Thrun, "A self-supervised terrain roughness estimator for off-road autonomous driving," in *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2006, pp. 469–476.

[22] A. Lookingbill, J. Rogers, D. Lieb, J. Curry, and S. Thrun, "Reverse optical flow for self-supervised adaptive autonomous robot navigation," *International Journal of Computer Vision*, vol. 74, pp. 287–302, 2006.

[23] T. Mar, V. Tikhanoff, G. Metta, and L. Natale, "Self-supervised learning of grasp dependent tool affordances on the iCub humanoid robot," in *IEEE Int. Conf. Robot. and Autom.*, 2015, pp. 3200–3206.

[24] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. J. Robot. Res.*, vol. 37, no. 4-5, pp. 421–436, 2018.

[25] D. Gandhi, L. Pinto, and A. Gupta, "Learning to Fly by Crashing," in *IEEE/RSJ Int. Conf. Intelligent Robots Sys.*, 2017, pp. 3948–3955.

[26] R. Hadsell, P. Sermanet, J. Ben, A. Erkan, M. Scoffier, K. Kavukcuoglu, U. Muller, and Y. LeCun, "Learning long-range vision for autonomous off-road driving," *jfr*, vol. 26, no. 2, pp. 120–144, 2009.

[27] M. Nava, A. Paolillo, J. Guzzi, L. M. Gambardella, and A. Giusti, "Uncertainty-aware self-supervised learning of spatial perception tasks," *IEEE Robot. and Autom. Lett.*, vol. 6, no. 4, pp. 6693–6700, 2021.

[28] E. Tricomi, F. Missiroli, M. Xiloyannis, N. Lotti, X. Zhang, M. Stefanakis, M. Theisen, J. Bauer, C. Becker, and L. Masia, "Soft robotic shorts improve outdoor walking efficiency in older adults," *Nature Machine Intelligence*, 12 2023.

[29] D. A. Winter, *Biomechanics and Motor Control of Human Movement*. John Wiley & Sons, Inc., 2009.

[30] R. Riener, M. Rabuffetti, and C. Frigo, "Stair ascent and descent at different inclinations," *Gait & Posture*, vol. 15, no. 1, pp. 32–44, 2002.

[31] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[32] T. M. Ingolfsson, M. Hersche, X. Wang, N. Kobayashi, L. Cavigelli, and L. Benini, "EEG-TCNet: An accurate temporal convolutional network for embedded motor-imagery brain–machine interfaces," in *IEEE Int. Conf. on Systems, Man, and Cybernetics*, 2020, pp. 2958–2965.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[34] "Pythorch TCN," https://github.com/paul-krug/pytorch-tcn, accessed: 2025-02-27.