

Deep Domain Adaptation, Pseudo-Labeling, and Shallow Network for Accurate and Fast Gait Prediction of Unlabeled Datasets

Jaeyoung Na¹, Hyunwoo Kim, Giuk Lee², *Member, IEEE*, and Woochul Nam¹, *Member, IEEE*

Abstract—Developing personalized gait phase prediction models is difficult because acquiring accurate gait phases requires expensive experiments. This problem can be addressed via semi-supervised domain adaptation (DA), which minimizes the discrepancy between the source and target subject features. However, classical DA models have a trade-off between accuracy and inference speed. Whereas deep DA models provide accurate prediction results with a slow inference speed, shallow DA models produce less accurate results with a fast inference speed. To achieve both high accuracy and fast inference, a dual-stage DA framework is proposed in this study. The first stage uses a deep network for precise DA. Then, a pseudo-gait-phase label of the target subject is obtained using the first-stage model. In the second stage, a shallow but fast network is trained using the pseudo-label. Because computation for DA is not conducted in the second stage, an accurate prediction can be accomplished even with the shallow network. Test results show that the proposed DA framework reduces the prediction error by 1.04% compared with a shallow DA model while maintaining its fast inference speed. The proposed DA framework can be used to provide fast personalized gait prediction models for real-time control systems such as wearable robots.

Index Terms—Domain adaptation, dual-stage architecture, gait phase, inference speed.

I. INTRODUCTION

HUMAN bipedal walking has been extensively studied owing to its importance as an essential motion in human life [1], [2], [3], [4]. There are two different approaches to consider gait phase: discrete and continuous. In the former, the gait cycle can be classified into the following discrete phases: initial contact, loading response, mid-stance, terminal stance, pre-swing, initial swing, mid-swing, and terminal swing. This discrete gait analysis method is typically used for abnormal

gait detection in medical diagnoses and athletic ability evaluation [5], [6]. However, this method is not suitable for assistive robots because it requires a time-varying assistive force. In the latter, the gait phase is represented by a continuous variable that increases from 0% to 100% at every gait cycle [7]. The gait phase variable is 0% when the heel contacts the ground, which is referred to as heel strike. Then, the phase variable increases linearly over time and becomes 100% when the next heel strike occurs on the same foot. At this moment, the value discontinuously decreases to 0% to represent the next gait cycle. This continuous gait phase is required for wearable assistive robots to synchronize the assistive force with the desired gait phase [7], [8], [9], [10].

In addition, gait phase prediction is difficult because subjects walk with their own gait patterns, which vary over time [11]. Conventional gait phase prediction models adopt heuristic rule-based approaches. To capture the gait cycle transition, deterministic gait events, such as the heel strike, are estimated using sensors mounted on the lower limbs. For example, *Ding et al.* [12] measured the thigh angle with inertial measurement unit (IMU) sensors to capture the instant of maximum hip flexion. Then, the stride time was estimated as the time interval between two consecutive maximum hip flexion events. The authors mentioned that their approach is valid only for tests on a treadmill, where stride time is almost constant, and that different approaches are needed to consider stride time variability. To address this problem related to rule-based models, researchers have introduced machine learning (ML) techniques for gait phase prediction [9], [13], [14], [15], [16], [17] because they are relatively robust to speed variations. Convolutional neural networks (CNNs) have been recently used to increase the gait phase prediction accuracy [9] as they are useful for extracting features from time-series signals from multiple sensors [18], [19], [20], [21], [22]. As gait patterns are considerably different among subjects, the prediction of the ML model is more accurate when it is trained with its gait motion dataset than when it is trained with other subjects' data. This personalized ML training requires sensor data and gait phase values for all target subjects. While lower limb motion data (i.e., joint angle data) can be easily acquired, measuring the true gait phase value requires costly and time-consuming experiments, as all subjects need to visit laboratories with high-cost equipment, such as treadmills with pressure sensors or visual motion capture systems.

Domain adaptation techniques can significantly reduce the effort required to acquire the true gait phase from different subjects. Hereafter, if gait phase ground truths for a subject

Manuscript received 4 July 2022; revised 24 December 2022 and 19 March 2023; accepted 25 March 2023. Date of publication 4 May 2023; date of current version 30 May 2023. This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT) under Grant 2021R1A4A3030268 and in part by the Research and Development Program for Forest Science Technology provided by the Korea Forest Service (Korea Forestry Promotion Institute) under Grant 2021364B10-2123-BD01. (*Corresponding author: Woochul Nam.*)

This work involved human subjects or animals in its research. The authors confirm that all human/animal subject research procedures and protocols are exempt from review board approval.

The authors are with the Department of Mechanical Engineering, Chung-Ang University, Seoul 06974, Republic of Korea (e-mail: jaeyng.na@gmail.com; vestplay@cau.ac.kr; giuklee@cau.ac.kr; wcnam@cau.ac.kr).

Digital Object Identifier 10.1109/TNSRE.2023.3272887

are available, the subject is referred to as a source subject (SS); if the true gait phase is unavailable, the subject is referred to as a target subject (TS). Semi-supervised domain adaptation (DA) can be used to train the ML model, which is optimized to a TS without its true gait phase; specifically, the model can be trained with the following data: the SS motion data, SS gait phase, and TS motion data. Semi-supervised learning can be realized by searching for subject-independent features.

DA models for time-series data can be divided into shallow and deep models. Shallow models utilize only a few parameters to generate subject-independent features. Although these models produce less accurate predictions, they provide fast inference, which is necessary for real-time applications, such as wearable robots. We previously developed a shallow DA model for gait analysis using a multilayer perceptron (MP) [23]. To increase the model accuracy, a time-domain feature set was used as the input variable instead of raw time-series data. The deep model utilizes a deep neural network to learn features from signal data. However, a deep network requires a long inference time, which hinders real-time control.

The novelty of the study is the development of a dual-stage DA model to address the trade-off between accuracy and inference speed. Specifically, in the first stage, a CNN-based deep network is used for precise DA. Considering the extraordinary pattern recognition ability of the deep CNN, the quality of the subject-independent feature obtained from the deep CNN is better than that of the shallow MP. Therefore, the deep DA model is expected to provide accurate gait phase predictions. Then, in the second stage, a shallow MP is trained with the TS gait phase that was predicted in the first stage. Note that a deep network is not used in the second stage; thus, the inference time for the second stage is very short. In addition, when the TS gait phase is predicted for a test, only the second stage is used. Thus, the TS gait phase can be accurately and rapidly predicted without a ground truth.

The remainder of this paper is organized as follows: Section II presents previous works on gait phase prediction and domain adaptation. Section III describes the dataset and the proposed model. Section IV presents the feature distributions and prediction errors. Finally, the paper is concluded in Section V.

II. RELATED WORKS

A. CNN-Based Gait Phase Prediction

CNN models have been widely used to predict discrete and continuous gait phases [6], [9], [25], [26], [27], [28], [29], [30]. Su et al. [25] trained various machine learning models (i.e., CNN, k-nearest neighbor, decision tree, naïve Bayesian, and linear discriminant analysis) to classify five discrete gait phases using nine IMU sensor signals. The CNN architecture was also used for continuous gait phase prediction under various ground conditions: flat ground, uphill, downhill, and ascending/descending stairs [9]. Their CNN model provided satisfactory results even when the ground conditions varied over time. Martinez-Hernandez et al. [26] attached three IMU sensors to the lower limbs to predict three different walking activities (i.e., walking on level ground and ascending/descending ramps) and eight discrete gait phases. A CNN model and first-order Markov chain were used to predict the current and next gait phases, respectively. Arshad et al. [28] classified stance and swing phases using a

single IMU sensor mounted on the waist. The performance of 16 different classifiers, including a CNN and recurrent neural network, were compared. Wang et al. [30] measured gait data from pressure array sensors on the foot and from IMU sensors on the thigh and shank to discern four discrete gait phases. Their results showed that the CNN model outperformed the hidden Markov and k-nearest neighbor models. Additionally, the accuracy was higher when the IMU signals were used than when the foot pressure signal was used.

B. Semi-Supervised DA for Gait Estimation

Semi-supervised DA is useful for inter-subject gait analysis [6], [26] because the gait pattern varies across subjects [7]. Guo et al. [6] used a multi-source DA to detect gait abnormalities from motion capture and electromyography data. This study showed that the classification accuracy of normal and abnormal gait was improved by using a DA called maximum cross-domain classifier discrepancy. This DA approach alternately maximizes and minimizes the cross-domain discrepancy. Because this DA technique induces a class-wise domain shift reduction, it is inapplicable to gait phase regression. Thus, this study modified a domain adversarial neural network (DANN), which can be modified to conduct regression tasks. Mu et al. [31] proposed a model to classify four discrete gait states to cope with the sensor-shift issue. They used both the original DANN and the multi-source DANN to consider the sensor-shift without data annotations. Although the original DANN significantly increased the classification accuracy, the effects of the multi-source DA on accuracy were not considerable. This gait classification framework only focused on accuracy and did not consider inference time, and thus it would be challenging to execute their model in embedded systems. Choi et al. [23] modified the DANN to predict the continuous gait phase. They also proposed a method for selecting the best SS (among several subjects) by calculating time-invariant correlations between embedding vectors. However, their network for DA is a very shallow MP, and thus the efficiency of DA may be limited. To address this problem, DA is conducted by a deep CNN in this study. Then, a shallow CNN is adopted for fast inference in embedded systems.

III. METHODS AND MATERIALS

A. Dataset

A public dataset of gait motions was used in this study [32]. Two IMU sensors attached to the tibia bone, as shown in Fig. 1(a), were used for gait phase prediction. Example plots of the angle, angular velocity, and gait phase are shown in Fig. 1(b). This dataset provides gait data for three different walking modes, namely, slow, fast, and comfortable walking speeds. In the last mode, the subjects walked at self-selected comfortable speeds, which varied between 1.1-1.67 m/s. The comfortable speed mode was considered in this study because it represents walking in real life. Subjects 02-06 in the dataset were used in this study and will be referred to as Subjects A-E hereon. The angle and angular velocity of both limbs were used as the prediction model input.

B. Gait Phase Prediction Model

Although the previous DA gait phase model [23] improved the prediction accuracy, the improvement was limited because

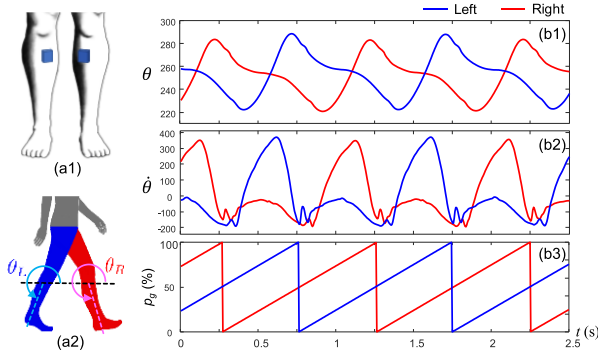


Fig. 1. Dataset. (a1) and (a2) Location and shank angle sign of the IMU sensors, respectively; (b1) and (b2) Angle and angular velocity measured from the IMU sensors on the shanks, respectively. (b3) Continuous gait phase.

it used a shallow MP network for the DA, as shown in Fig. 2(a). This issue can be addressed by adopting a deep neural network (e.g., a deep CNN), which is an efficient architecture for pattern recognition. However, a deep network requires a relatively long inference time, which can be an obstacle for real-time applications.

In this study, a novel framework was developed to achieve precise DA and fast inference. This new framework comprises two stages, as shown in Fig. 2(b). The first-stage model performs precise DA with a deep neural network; hereafter, this model is referred to as deep domain adaptation (DDA). The second-stage model is composed of a few layers for fast inference; hereafter, this model is referred to as shallow and fast inference (SFI). Once the training is completed, DDA is not executed. Thus, fast inference is possible during the test.

Before describing the new framework, it should be noted that two variables are required to consider the continuous gait phase. Because the continuous gait phase variable increases from 0% to 100% during a gait cycle, the phase changes discontinuously from 100% to 0% at every heel strike. This discontinuous variation leads to a very large error close to the heel strike, even for slight differences. Suppose that the prediction lags the true gait by only 2%. When the true phase is 1%, the model will predict a 99% phase. Although this is a very small delay, the phase error value is significant. To prevent this, gait phase P_g can be replaced with two variables (i.e., x and y) [7], [23] as follows:

$$P_g = \frac{100}{2\pi}\theta = \frac{100}{2\pi}\arctan2(y, x) \quad (1)$$

where x and y are defined as $\cos\theta$ and $\sin\theta$, respectively. In contrast to P_g , x and y continuously change in every gait phase; thus, x and y were used for error evaluation.

The prediction error for the gait phase was calculated using the normalized root mean square error (NRMSE), which can

be obtained as

$$\begin{aligned} \text{NRMSE}(\%) &= \sqrt{\frac{\frac{1}{n} \sum_{i=1}^n (x_{L,i} - \hat{x}_{L,i})^2 + (y_{L,i} - \hat{y}_{L,i})^2 + (x_{R,i} - \hat{x}_{R,i})^2 + (y_{R,i} - \hat{y}_{R,i})^2}{4}}{2}} \\ &\times 100, \end{aligned} \quad (2)$$

where $x_{L,i}$, $y_{L,i}$, $x_{R,i}$, and $y_{R,i}$ are the left and right predicted gait phase values, respectively; $\hat{x}_{L,i}$, $\hat{y}_{L,i}$, $\hat{x}_{R,i}$, and $\hat{y}_{R,i}$ are the left and right ground truths, respectively; n is the number of test samples. The error was normalized to 2 because the range of $x_{L,i}$, $x_{R,i}$, $y_{L,i}$, and $y_{R,i}$ is 2 (i.e., between -1 and 1).

1) *Stage I: Deep Neural Network for Domain Adaptation (Pseudo-Label Generation)*: As shown in Fig. 3, the DDA in Stage I is composed of a CNN feature extractor, FC mapping network, FC regression network, and FC domain discriminator. The feature extractor was designed using a CNN to utilize its powerful pattern-recognition capability. The mapping network is required for feature mapping into different spaces where the SS and TS features are similar. The regression network is trained only with source data because the gait phase is available only for the SS. Because the mapping network is trained such that its output for the TS is similar to that for the SS, the regression network, which is trained with the SS only, is expected to accurately predict the TS gait phase.

While the mapping network generates indiscriminative features in both domains, the domain discriminator is trained to classify whether the input data are SS or TS signals. A gradient reversal layer (GRL) between the mapping network and domain discriminator inverts the gradient sign for adversarial training [24]. The original DANN used binary cross-entropy loss for the domain classifier. However, in this study, a least square GAN loss is introduced because it leads to stable adversarial training [33].

Four signals (i.e., the shank angle and angular velocity of each limb) with a 0.5-s duration and 200-Hz sampling frequency were transformed into a 100×4 tensor and fed to the feature extractor. The architectures of the feature extractor, regression network, and domain discriminator are presented in Table I. Batch normalization was used in all layers (except the output layers), and leakyReLU was used as the activation function.

The loss function L_r of the regression network is defined by the mean square error (MSE) as follows (3), shown at the bottom of the page, n_s is the number of samples for the SS. Note that the loss for the TS is not included in (3) because the TS gait phase is not available. The loss L_d of the subject discriminator is defined as:

$$L_d = \frac{1}{n_s + n_t} \sum_{i=1}^{n_s+n_t} (p_i - z_i)^2 \quad (4)$$

where n_t is the number of samples for the TS and p_i is the probability that the input sample belongs to the SS. z_d equals

$$L_r = \frac{1}{n_s} \sum_{i=1}^{n_s} \frac{(x_{L,i} - \hat{x}_{L,i})^2 + (y_{L,i} - \hat{y}_{L,i})^2 + (x_{R,i} - \hat{x}_{R,i})^2 + (y_{R,i} - \hat{y}_{R,i})^2}{4} \quad (3)$$

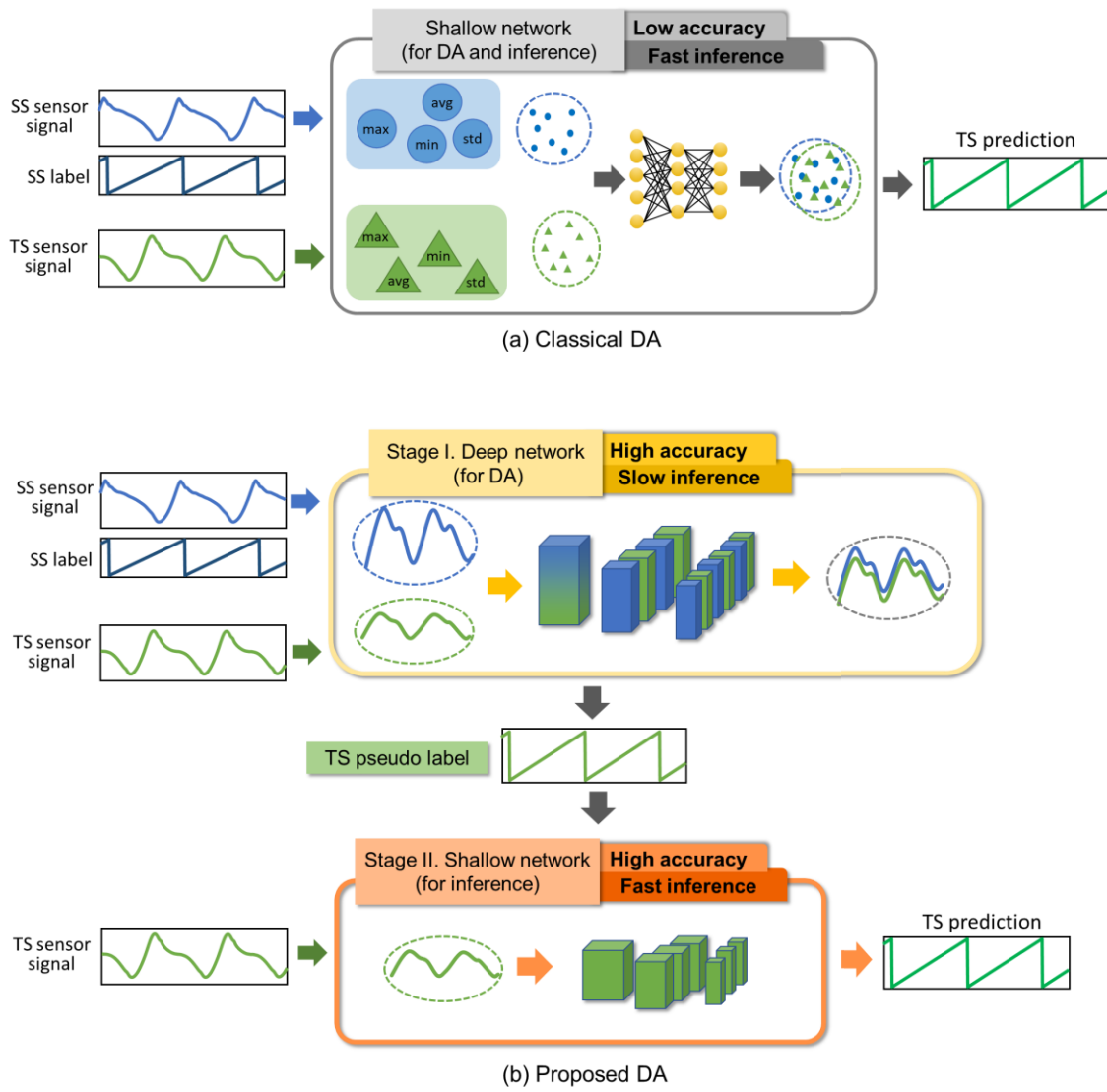


Fig. 2. DA frameworks. (a) Classical DA model (SDA) and (b) the proposed framework (DDA+SFI).

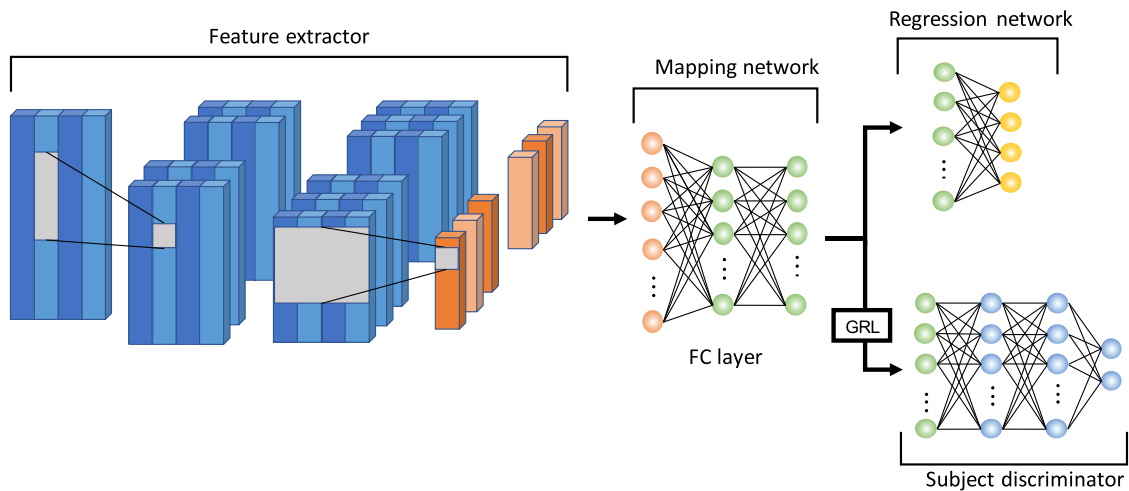


Fig. 3. DDA architecture (Stage I).

unity if the input sample belongs to the SS, z_d equals zero if the input sample belongs to the TS.

The regression network is responsible for predicting the gait phase; thus, its weights are trained to reduce L_r . The subject

TABLE I
DDA ARCHITECTURE

	Layer	Number of filters (Stride)	Filter size	Size of the output tensor
Feature extractor	Conv-1	10 (1)	21×1	80×4×10
	Conv-2	20 (1)	21×1	60×4×20
	Conv-3	30 (1)	21×4	40×1×30
Mapping network	Flattening layer	-	-	1200×1
	Hidden layer	-	-	256×1
	Output layer	-	-	256×1
Regression network	Input layer	-	-	256×1
	Output layer	-	-	4×1
Subject discriminator	Input layer	-	-	256×1
	Hidden layer-1	-	-	256×1
	Hidden layer-2	-	-	256×1
	Output layer	-	-	2×1

discriminator must determine whether the input sample is SS or TS data. To this end, the weights of the subject discriminator are trained to reduce L_d . The output of the feature extractor requires two different characteristics that are desirable for gait prediction. Meanwhile, the output corresponding to the SS and TS data must be similar. To achieve these goals simultaneously, the weights of the feature extractor are trained to minimize L_r and maximize L_d . While the subject discriminator is trained to decrease L_d the feature extractor must be trained to increase L_d . This adversarial goal of the feature extractor and subject discriminator can be achieved using a GRL because it changes the gradient signs during backpropagation. The parameter λ_p of the GRL determines the balance between L_r minimization and L_d maximization. For stable training, λ_p gradually increases over the epoch from 0 to 1. Details of this adversarial training are given in [24]. It is worth noting that the weight values of the DDA are not determined as the values of the final epoch (i.e., 1,000 epochs). Instead, the final values of the weights were determined as the values of a specific epoch in which the prediction accuracy of the SS data is a minimum.

Stage II: Shallow network for inference

If the DDA is directly used for a test, the inference speed will be slow because of its deep architecture. Therefore, a shallow model (SFI) was created and trained separately. The TS gait phase values predicted by the DDA were used to train the SFI. Hereafter, the prediction result for the TS is referred to as the pseudo-label. Note that the SFI does not contain any networks for DA; thus, it can be constructed with a few layers.

The SFI is composed of a shallow CNN feature extractor and an FC regression network. The SFI input is a 200 ms

time-series signal on angle and angular velocity. The SFI architectures are listed in Table II. Batch normalization was applied to all layers (except the output layers), and leakyReLU was used as the activation function. The SFI was trained using an ADAM optimizer with a learning rate of 0.005 and 2000 epochs.

2) *Shallow Domain Adaptation*: For performance comparison, an earlier DA model proposed by [23] was used. Because this earlier model conducts DA with few layers, hereafter, this model is referred to as shallow DA (SDA). SDA comprises a mapping network, regression network, and domain discriminator. The mapping network consists of a 15-node hidden layer and a 20-node output layer. The regression network simply connects the last layer of the mapping network and the output $x_{L,i}$, $y_{L,i}$, $x_{R,i}$, and $x_{R,i}$ without hidden layers. The domain classifier has one hidden layer with 20 nodes, and the output is a sigmoid value. The tanh activation function is applied to the mapping network and regression network. For the domain classifier, the leakyReLU function was used. Batch normalization was applied to all layers except the output layer. Note that no feature extractor is used for the SDA, while the DDA contains a deep CNN feature extractor. While the time-series angle and angular velocity were used as the DDA input, the time-domain features of the angle and its velocity were used as the SDA input. Specifically, 12 features were calculated from the sliding window signals: the last value, maximum, minimum, average, and standard deviation of the (left/right) shank angles, as well as the last (left/right) angular velocity value. An adaptive window technique [23] was used to consider variations in the stride time. The sliding window length was determined as 30% of the previous stride time.

TABLE II
SFI ARCHITECTURE

	Layer	Number of filters (Stride)	Filter size	Size of the output tensor
Feature extractor	Conv-1	5 (1)	5×1	30×4×5
	Conv-2	5 (1)	5×4	20×1×5
	Flattening layer	-	-	100×1
Regression network	Hidden layer	-	-	16×1
	Output layer	-	-	4×1

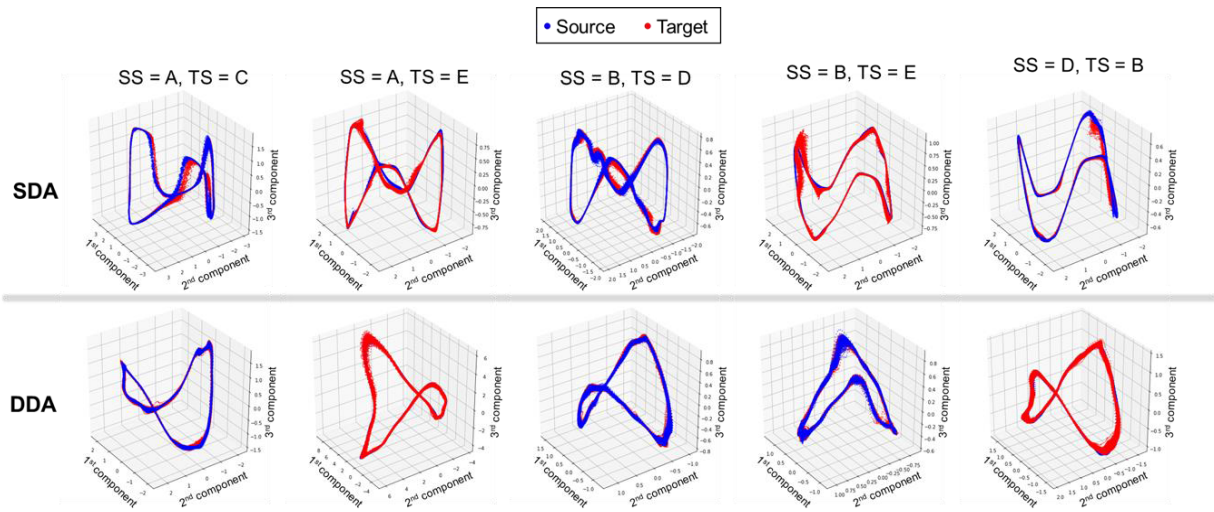


Fig. 4. PCA plots of domain-adapted features.

IV. RESULTS

Among the five subjects in the public dataset [32] used in this study, one was used as the SS and another was used as the TS; thus, 20 subject combinations could be used for validation. The performance of the proposed framework was evaluated using the feature distribution and prediction error.

A. Domain-Adapted Feature Distribution

As the output of the mapping network is the feature obtained by the DA, it was used for DA quality evaluation. If DA is accomplished, the output features of the SS should be similar to those of the TS. To effectively visualize the feature distribution, a principal component analysis (PCA) and t-SNE plot were used, as shown in Figs. 4 and 5, respectively.

For both the PCA and t-SNE plots, the SS and TS distributions were similar for SDA and DDA, suggesting that DA was successfully accomplished. In addition, the distribution similarity increased when DDA was used compared to when SDA was applied. While the SDA has some dissimilar regions, the SS and TS distributions overlap in every region for the DDA, as shown in Figs. 4 and 5. This difference suggests that the deep network in DDA is more efficient for DA than SDA.

B. Prediction Error

The prediction error for the TS gait phase was measured by the NRMSE. The NRMSE of the four different models, No-DA, SDA, DDA, and SFI, are listed in Table III (Bolds represent the outperformed value for each case). Note that the No-DA model predicts the TS gait phase without DA. The DDA error was calculated with the pseudo-label in Stage I, and the SFI error was obtained with the output of Stage II. Comparing the errors of the No-DA and SDA models, the regression accuracy improved considerably with the latter. It is worth noting that the NRMSE of the DDA was even smaller than the SDA error. For example, when SS = B and TS = A, the SDA error is 4.519%, whereas the DDA error is only 2.622%. There are many other cases in which the error is significantly reduced when the DDA is used. These results suggest that the proposed deep network achieved DA more effectively than the previous shallow network (SDA). The errors of the SFI and DDA models tend to be similar, suggesting that the prediction accuracy degradation (owing to architecture reduction) is negligible. Additionally, Table IV presents the NRMSE of the intra-subject study for comparison. The network structure used in the intra-subject study is the same as the SFI structure provided in Table II. However, it is

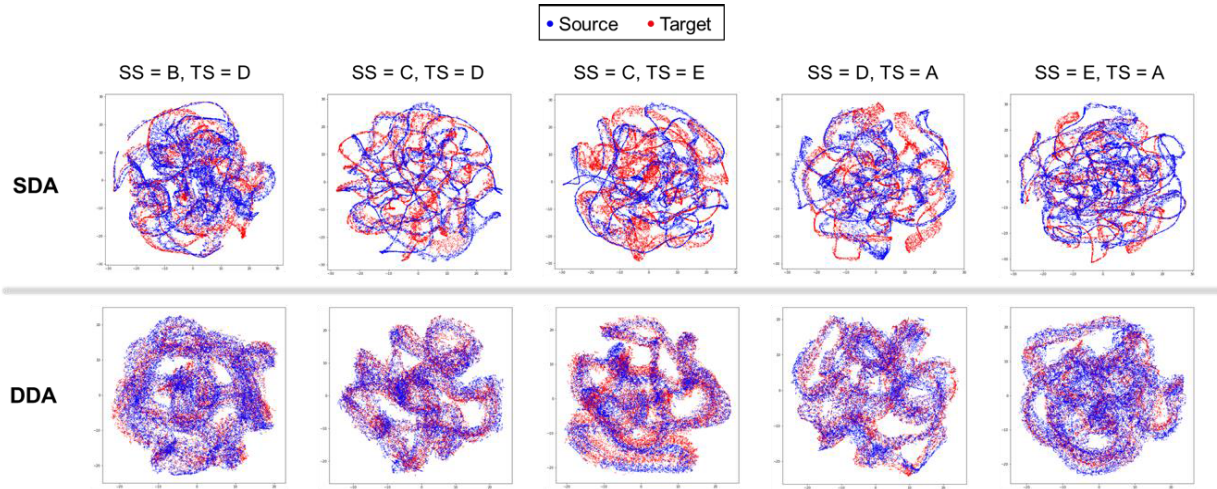


Fig. 5. t-SNE plots of domain-adapted features.

TABLE III
NRSME (%) OF TS

		Target subject				
		A	B	C	D	E
A	No DA		4.970	5.274	5.573	7.656
	SDA		3.591	4.747	4.966	4.351
	DDA		2.165	3.070	1.855	3.437
	SFI		2.209	3.174	1.789	3.233
B	No DA	7.205		3.544	4.756	5.894
	SDA	4.519		2.529	3.015	2.852
	DDA	2.622		2.363	2.108	2.085
	SFI	2.822		2.078	2.059	1.814
C	No DA	5.805	2.694		2.651	3.580
	SDA	4.913	2.624		2.415	3.252
	DDA	2.377	2.202		2.248	2.710
	SFI	2.334	2.068		2.261	2.583
D	No DA	5.038	3.788	4.439		4.004
	SDA	4.273	2.728	2.778		2.660
	DDA	1.717	2.212	2.461		3.286
	SFI	2.070	2.250	2.393		3.172
E	No DA	6.085	3.715	2.176	2.994	
	SDA	4.099	2.611	2.266	2.441	
	DDA	2.881	2.171	2.033	3.299	
	SFI	2.996	2.043	1.896	3.276	

worth noting that the SFI model and intra-subject model are different. To obtain the SFI prediction, domain adaptation was conducted by DDA. Then, the DDA prediction result was used to train the SFI model. Meanwhile, the intra-subject model was simply trained with the true gait phase. The error of the intra-subject study is smaller than that of the inter-subject models. However, the difference in the error was significantly reduced when DDA (or SFI) was used.

Note that the results in Table III were obtained by using DANN. To verify whether DANN is an effective DA framework, another representative DA model (i.e., adversarial discriminative DA [34]) was tested with the same dataset.

TABLE IV
NRSME (%) OF INTRA-SUBJECT ESTIMATION

Subject	A	B	C	D	E
NRMSE	1.408	1.800	1.778	1.146	1.489

TABLE V
NRSME (%) OF ADVERSARIAL DISCRIMINATIVE DA

		Target subject				
		A	B	C	D	E
Source subject	A		4.172	4.491	2.591	6.313
	B	4.721		2.304	3.095	3.324
	C	5.007	2.624		3.417	3.362
	D	3.185	2.752	2.948		4.702
	E	6.671	2.971	2.944	4.509	

The original adversarial discriminative DA was modified such that its CNN structure was similar to that of the DDA. The error of the adversarial discriminative DA is provided in Table V. The error of the adversarial discriminative DA is lower than that of the No-DA model except in four cases. However, the error is higher than the error obtained with DDA, which is trained with DANN. This suggests that DANN has a powerful DA capability for the continuous gait phase.

To further investigate the NRMSE values, various statistical values (i.e., average, standard deviation, median, minimum, maximum, and significance level) were calculated and compared, as shown in Fig. 6. The following characteristics were observed in the statistical analysis. First, the average NRMSE of the DDA was smaller than that of the SDA, suggesting that the DDA outperformed the SDA. Furthermore, two-sided paired t-tests verified that the NRMSE exhibited significant differences between the SDA and DDA. Second, the maximum and standard deviation of the NRMSE for the DDA were significantly smaller than those of the SDA, implying that the DDA guarantees an accurate prediction, regardless of the SS. Third, the standard deviation, minimum, and maximum of

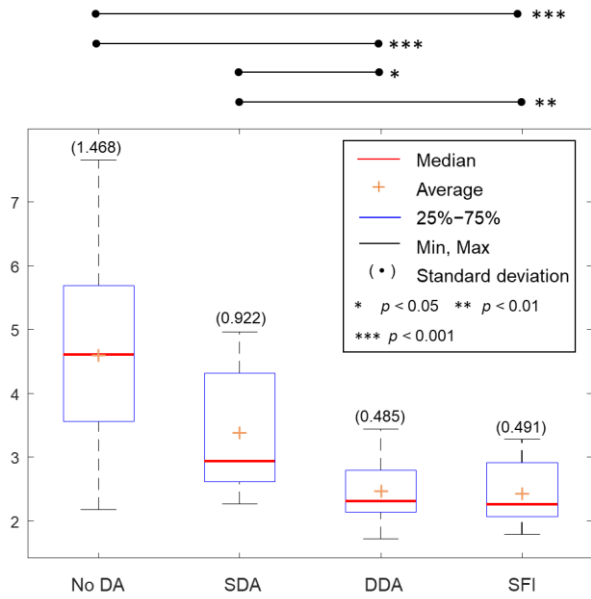


Fig. 6. Statistical values of the NRMSE.

the SFI were similar to those of the DDA, suggesting that the prediction performance can be maintained after model reduction. Some researchers may claim that the SFI error should be larger than that of DDA at all times because SFI is trained with the pseudo-label obtained from DDA. However, this counterintuitive result was obtained because the SFI prediction is larger than the pseudo-gait phase in some time intervals, and the SFI prediction is smaller than the pseudo-value in other intervals. Consider a situation where the true phase is larger than the pseudo-value and the SFI prediction. In this case, if the SFI prediction is larger than the pseudo-value, as shown in Fig. 7(a), the SFI value is closer to the true value than the pseudo-value. Thus, the SFI error is smaller than that of DDA. If the SFI prediction is smaller than the pseudo-value, as shown in Fig. 7(b), the SFI error is larger than that of DDA. As a result, the subject-wise average of the DDA error is very similar to that of the SFI error, as shown in Fig. 6. Moreover, the p -value of DDA and SFI is larger than 0.05, which also confirms that the error difference is not appreciable.

Accuracy improvements via DA can also be observed in the gait phase prediction results over time, as shown in Fig. 8. When DA was not applied, the predicted phases differed considerably from the true phase. Although the difference was reduced by the SDA, the error was still noticeable. However, the difference between the true and predicted phases was very small for the DDA and SFI.

C. Inference Time

Because of its compact architecture, the SFI requires a very short inference time. For a quantitative comparison, the inference times of the SFI and DDA were measured. When a Jetson Xavier NX was used for the inference, the average computation time of the DDA was 8.34 ms on the CPU and 2.52 ms on the GPU. The inference time of the SFI was 1.25 ms on the CPU and 1.55 ms on the GPU. This difference in the inference time suggests that the SFI is more suitable

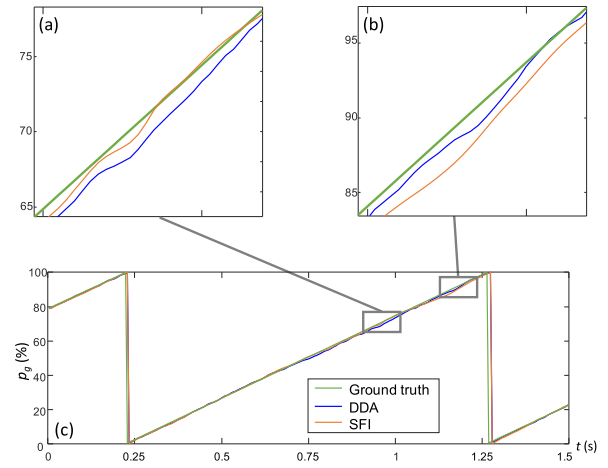


Fig. 7. Examples of gait phase profiles (SS = D, TS = A).

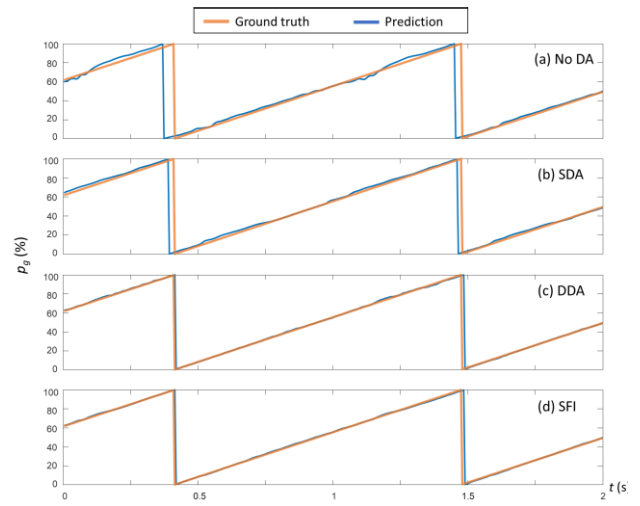


Fig. 8. Examples of gait phase profiles (SS = D, TS = A); (a)-(d) profiles for the No DA, SDA, DDA, and SFI models, respectively.

for real-time prediction in embedded systems. Note that the inference of SFI with CPU is slightly faster than that with GPU because the GPU mode requires some time to transfer the IMU signal to the GPU.

The number of weights was obtained because the inference speed strongly depends on the number of weights. The number of weights in the DDA is 496,594 and the SFI number is 2,901, suggesting that the number of weights is significantly reduced in Stage II. Note that an earlier CNN model for the gait phase [9] contains 61,266 weights considering that it is composed of two convolutional layers (with ten filters) and an FC network. The number of weights in the SFI is approximately 20 times smaller than in the previous model. Thus, SFI is a faster and more suitable model for embedded systems than the previous model.

V. CONCLUSION

The main contribution of this study is the development of a new DA framework for gait phase that is more accurate than the shallow DA model while maintaining a fast inference speed. Specifically, the DDA outperformed the SDA by reducing the gait phase prediction error from 3.38% to 2.47% on average. However, one disadvantage of the DDA is its slow

inference, which hinders its real-time application potential. The inference speed issue was addressed by developing a dual-stage framework. In the first stage, a gait phase pseudo-label is obtained from the DDA. In the second stage, a shallow network is trained using the obtained pseudo-label. By adopting a shallow architecture in the second stage, the inference speed can be significantly increased. Therefore, the new framework can be used for accurate and fast gait inference when the TS gait phase label is unavailable.

This study has some limitations. Although the proposed model can be optimized for a TS without the true TS gait phases, it still requires TS motion data for both Stages I and II. The model must also be individually trained for all TSs. Additionally, if the SS gait pattern is significantly different from the pattern, the DA performance may be unsatisfactory, resulting in a large prediction error.

Although the proposed DA framework focuses on inter-subject variation, it can also be used to consider other types of variations. For example, gait patterns change considerably with walking speed. After the gait motion and gait phase for a specific walking speed are measured, the gait phase prediction model can be optimized to slow and fast walking motions without their true gait phases by using the DDA. Gait motion is also affected by the ground conditions. For example, after measuring the motion data and true gait phase on flat ground, the gait phase on stairs and inclined/declined ground can be predicted using the DDA. In addition, the DDA can be used to predict the gait phase for walking along curved paths.

REFERENCES

- [1] M. Brandes, R. Schomaker, G. Möllenhoff, and D. Rosenbaum, "Quantity versus quality of gait and quality of life in patients with osteoarthritis," *Gait Posture*, vol. 28, no. 1, pp. 74–79, Jul. 2008.
- [2] D. H. Metz, "Mobility of older people and their quality of life," *Transp. Policy*, vol. 7, no. 2, pp. 149–152, Apr. 2000.
- [3] J. Taborri, E. Palermo, S. Rossi, and P. Cappa, "Gait partitioning methods: A systematic review," *Sensors*, vol. 16, no. 1, p. 66, Jan. 2016.
- [4] H. T. T. Vu et al., "A review of gait phase detection algorithms for lower limb prostheses," *Sensors*, vol. 20, no. 14, pp. 1–19, 2020.
- [5] A. Nieuwboer, R. Dom, W. De Weerd, K. Desloovere, S. Fieuws, and E. Broens-Kaucsik, "Abnormalities of the spatiotemporal characteristics of gait at the onset of freezing in Parkinson's disease," *Movement Disorders, Off. J. Movement Disorder Soc.*, vol. 16, no. 6, pp. 1066–1075, Nov. 2001.
- [6] Y. Guo, X. Gu, and G.-Z. Yang, "MCDCCD: Multi-source unsupervised domain adaptation for abnormal human gait detection," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 10, pp. 4017–4028, Oct. 2021.
- [7] I. Kang, P. Kunapuli, and A. J. Young, "Real-time neural network-based gait phase estimation using a robotic hip exoskeleton," *IEEE Trans. Med. Robot. Bionics*, vol. 2, no. 1, pp. 28–37, Feb. 2020.
- [8] K. Seo et al., "RNN-based on-line continuous gait phase estimation from shank-mounted IMUs to control ankle exoskeletons," in *Proc. IEEE 16th Int. Conf. Rehabil. Robot. (ICORR)*, Jun. 2019, pp. 809–815.
- [9] I. Kang, D. D. Molinaro, S. Duggal, Y. Chen, P. Kunapuli, and A. J. Young, "Real-time gait phase estimation for robotic hip exoskeleton control during multimodal locomotion," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3491–3497, Apr. 2021.
- [10] D. J. Villarreal, D. Quintero, and R. D. Gregg, "Piecewise and unified phase variables in the control of a powered prosthetic leg," in *Proc. IEEE Int. Conf. Rehab. Robot.*, Jul. 2017, pp. 1425–1430.
- [11] A. Mannini and A. M. Sabatini, "Walking speed estimation using foot-mounted inertial sensors: Comparing machine learning and strap-down integration methods," *Med. Eng. Phys.*, vol. 36, no. 10, pp. 1312–1321, Oct. 2014.
- [12] Y. Ding, I. Galiana, C. Sivi, F. A. Panizzolo, and C. Walsh, "IMU-based iterative control for hip extension assistance with a soft exosuit," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 3501–3508.
- [13] J. D. Farah, N. Baddour, and E. D. Lemaire, "Gait phase detection from thigh kinematics using machine learning techniques," in *Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA)*, May 2017, pp. 263–268.
- [14] S. Potluri, A. B. Chandran, C. Diedrich, and L. Schega, "Machine learning based human gait segmentation with wearable sensor platform," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2019, pp. 588–594.
- [15] A. Mannini, D. Trojaniello, A. Cereatti, and A. Sabatini, "A machine learning framework for gait classification using inertial sensors: Application to elderly, post-stroke and Huntington's disease patients," *Sensors*, vol. 16, no. 1, p. 134, Jan. 2016.
- [16] T. Miyake et al., "Heel-contact gait phase detection based on specific poses with muscle deformation," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2019, pp. 977–982.
- [17] R. L. Evans and D. K. Arvind, "Detection of gait phases using orient specks for mobile clinical gait analysis," in *Proc. 11th Int. Conf. Wearable Implant. Body Sensor Netw.*, Jun. 2014, pp. 149–154.
- [18] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mech. Syst. Signal Process.*, vol. 151, Apr. 2021, Art. no. 107398.
- [19] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 4277–4280.
- [20] M. Zihlmann, D. Perekrestenko, and M. Tschannen, "Convolutional recurrent neural networks for electrocardiogram classification," in *Proc. Comput. Cardiol. Conf. (CinC)*, Sep. 2017, pp. 1–4.
- [21] S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-time patient-specific ECG classification by 1-D convolutional neural networks," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 3, pp. 664–675, Mar. 2016.
- [22] O. Abdeljaber, O. Avci, M. S. Kiranyaz, B. Boashash, H. Sodano, and D. J. Inman, "1-D CNNs for structural damage detection: Verification on a structural health monitoring benchmark data," *Neurocomputing*, vol. 275, pp. 1308–1317, Jan. 2018.
- [23] W. Choi, W. Yang, J. Na, J. Park, G. Lee, and W. Nam, "Unsupervised gait phase estimation with domain-adversarial neural network and adaptive window," *IEEE J. Biomed. Health Informat.*, vol. 26, no. 7, pp. 3373–3384, Jul. 2022.
- [24] Y. Ganin et al., "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1–35, 2016.
- [25] B. Su, C. Smith, and E. G. Farewik, "Gait phase recognition using deep convolutional neural network with inertial measurement units," *Biosensors*, vol. 10, no. 9, p. 109, Aug. 2020.
- [26] U. Martinez-Hernandez, A. Rubio-Solis, and A. A. Dehghani-Sani, "Recognition of walking activity and prediction of gait periods with a CNN and first-order MC strategy," in *Proc. 7th IEEE Int. Conf. Biomed. Robot. Biomechatronics (Biorob)*, Aug. 2018, pp. 897–902.
- [27] N. S. Erdem, C. Ersoy, and C. Tunca, "Gait analysis using smart-watches," in *Proc. IEEE 30th Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC Workshops)*, Sep. 2019, pp. 1–6.
- [28] M. Z. Arshad, A. Jamsrandorj, J. Kim, and K.-R. Mun, "Gait events prediction using hybrid CNN-RNN-based deep learning models through a single waist-worn wearable sensor," *Sensors*, vol. 22, no. 21, p. 8226, Oct. 2022.
- [29] J. Hannink et al., "Sensor-based gait parameter extraction with deep convolutional neural networks," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 1, pp. 85–92, Jan. 2017.
- [30] F. L. Wang et al., "Recognition of the gait phase based on new deep learning algorithm using multisensor information fusion," *Sensors Mater*, vol. 31, no. 10, pp. 3041–3054, 2019.
- [31] F. Mu, X. Gu, Y. Guo, and B. Lo, "Unsupervised domain adaptation for position-independent IMU based gait analysis," in *Proc. IEEE Sensors*, Oct. 2020, pp. 1–4.
- [32] D. C. Miraldo, R. N. Watanabe, and M. Duarte, "An open data set of inertial, magnetic, foot-ground contact, and electromyographic signals from wearable sensors during walking," *Motor Control*, vol. 24, no. 4, pp. 558–570, Oct. 2020.
- [33] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2794–2802.
- [34] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7167–7176.