User-Agnostic Adaptation of Human Locomotion Intent: Leveraging Teacher-Student-Learning and Ensemble Modeling

Mahdi Bonyani[a,*], Maryam Soleymani[a] and Chao Wang[b]

[a]Ph.D. Student, Bert S. Turner Department of Construction Management, Louisiana State University, USA
[b]Associate Professor and Graduate Program Advisor, Bert S. Turner Department of Construction Management, Louisiana State University, USA

ARTICLE INFO

Keywords:
Teacher-Student-Learning
Human Activity Recognition
Unsupervised User-Agnostic Adaptation
Wearable Robots
Human Intent Prediction
Ensemble Learning
Deep learning

ABSTRACT

Wearable robot control in complicated contexts requires an understanding of human locomotion intent and behaviors. However, signals from human-robot interfaces are typically reliant on the user, leading to poor performance of the model trained on training user when applied to new end-point users. This study aims to address this problem by developing a novel Teacher-Student-Learning (TSL) approach using Heterogeneous Ensemble Hypotheses (HEH) to achieve unsupervised user-agnostic adaptation. The motivation behind using the Teacher-Student-Learning architecture is to leverage the diverse features extracted by HEH while maintaining computational efficiency. HEH reduces the difference between labeled training users and unlabeled end-point users by incorporating multiple diverse feature generators to extract a wide range of features, thereby increasing classification accuracy and reducing variance. However, this approach sacrifices efficiency due to the ensemble nature of HEH. To address this trade-off, the knowledge from HEH is distilled into a single network using TSL, ensuring precision and efficiency. The proposed approach is evaluated on two publicly available human locomotion datasets and a 2D moon dataset. Experimental results demonstrate the effectiveness of the proposed method, outperforming all other methods. Tested with three datasets, the proposed method can classify end-point users' data with an average accuracy of 98.9%, 96.7%, and 96.9% while yielding a low processing time (1 ms). Compared to a benchmark method, the suggested strategy improves the average accuracy by 1.5% and 7.2% for categorizing the target users' locomotor modes and stabilizes the learning curves. The proposed method represents a significant contribution to the field of human activity recognition and human intent prediction for human-robot interaction systems, enhancing the efficiency and generalization capacity of these systems.

1. Introduction

Wearable robotic devices, such as exoskeletons [1, 2], powered prostheses [3, 4], as well as supernumerary robotic limbs [5], are predicted to help people restore their mobility, increase load carrying capacity, and improve energy economy. Wearable robots appear to have a bright future, however, creating an intelligent and adaptable controller for a wearable robot still presents difficulties. A wearable robot's controller typically consists of low, middle, and high-level controllers arranged in a hierarchical design [6]. The middle-level controller switches its control settings in response to a prediction of the planned locomotion mode made by the high-level controller. The assistive force curve [2], swing trajectories [7], and joint impedance parameters [8] are examples of possible control configurations. The middle-level controller sets the settings of the low-level controller, which is then responsible for activating the joint position control or torque control. It is possible to preprogram and tune the control parameters of various locomotor modes for a variety of individuals [1, 9, 2, 10]. How and when to transition between various locomotor modes (e.g., standing, walking, sprinting, and ascending stairs) is still a major challenge. In order to seamlessly transition between different locomotor modes, wearable robots must first promptly ascertain the wearer's intention [11, 12].

Human intent is often complex and varies widely, making it challenging to accurately anticipate and plan for. According to prior research, EMG and IMU signals offer a unique insight into how humans interact with robots, providing insight into how robots can effectively understand and respond to human intent [3, 13, 11, 14]. Numerous categorization techniques have been put forth to categorize the recorded human-robot communications and use that data to infer the human locomotion mode [11, 13, 15]. There is a statistical discrepancy between two datasets which can lead to poor performance for existing classification methods when compared with a new dataset [13, 16]. However, the distribution difference is typical for signals pertaining to human-robot interfaces. The signals of the human-robot interface are user-centric because various users have varying muscles, skin tones, as well as patterns of mobility. The user's physical condition (i.e., limb weariness and skin conductivity) as well as the sensor location can both affect the interface signals, even for the same individual. It is not possible for the current approaches to capture and identify significant signals of various users and train a new model for every new user in order to properly determine a human's intent to move [15, 17].

The use of user-independent sensors, including vision sensors, is one way to address the issue of user-dependency. Vision sensors have the ability to see their surroundings in advance, which allows them to reasonably forecast how people will move [18, 19, 20, 21]. Zhang et. al [22] have shown that a visual perception system trained on a single

*Corresponding author
ORCID(s):

healthy user's dataset can be used directly to precisely tune a powered prosthesis's control modes and help various above knee amputees walk on challenging Environments without the need for additional new user training. Moreover, these findings have confirmed that most visual perception systems don't require various user data. However, when a person wears the gadget and changes states, like from standing to walking, visual sensors might not pick up on it. Thus, to completely understand human intent, the signals from the human-robot interaction still appear to be required.

Every sensor selection involves certain trade-offs: although correctly interpreting interface signals to determine human intent is crucial, user dependence is an unavoidable issue. How may the interface signals' reliance on the user be reduced? The disparity in interface signal distribution among topics is a significant contributor to user reliance. Reducing the distribution disparity will also reduce user reliance. Domain adaptation is a popular technique for tackling this issue [23, 24, 25] as illustrated on Table 1. Data from the training user domain and data from the end-point user domain are the two types of data used in this technique. There is a disparity between the two domains; the training user domain data are labeled, while the end-point user domain data are unlabeled. Using labeled training user domain and unlabeled end-point user domain data, the classifier is trained by unsupervised domain adaption techniques. The learned classifier may be utilized to categorize the end-point user data with a reasonable degree of accuracy after training. Maximum classifier discrepancy (MCD) [24] and domain-adversarial neural networks (DANN) [23] are two common domain adaption techniques. Ganin et al. proposed DANN consists of a domain classifier, a label classifier, as well as a feature generator. The generator have competition with the domain classifier. The objective of the classifier is to differentiate between the characteristics derived from the data of the training users' domain as well as the data of the end-point users' domain. To trick the domain classifier into thinking it can't tell these characteristics apart, the feature generator looks for identical hidden features in both the end-point and the training user domain data. Furthermore, significant characteristics that the label classifier can reliably classify must be extracted via the feature generator. Although DANN performs well, it ignores feature classes. Global adaptation may occasionally fail because of inconsistent differences between training and end-point user data across distinct classes. Saito et al. [24] developed MCD, which comprises of two classifiers and a feature generator, as a solution to this problem. The feature generator, the two classifiers, and themselves are antagonistic. The generator is utilized to reduce the difference between the classifiers, while the classifiers are utilized to maximizing it. As a result, the classifiers are optimized to provide more accurate predictions, while the feature generator is optimized to generate features with the least variance. Due to its ability to align the features of two domains in each class, MCD is recognized to perform better than DANN after training.

Images are usually classified using the domain adaption techniques that are currently in use. Predicting human movement intent and categorizing signals from the human-robot interaction have received little attention in domain adaptation research. An unsupervised User-Agnostic adaption technique akin to MCD was suggested in our earlier research [26]. Convolutional neural networks (CNNs) can anticipate the end-point users' intention to move after it was trained using labeled data from the training user as well as unlabeled data from the end-point users. The MCD single feature generator may not be capable of capturing the full complexity of the data and all the hidden features from the input data [27, 28]. This thought experiment provides insight into the theoretical implications of this study. The aforementioned problem may be solved via ensemble learning [29, 30], albeit at the expense of efficiency, by extracting a variety of features and improving generalization ability. But with wearable robots, efficiency is also vital. Is it feasible to pick up many characteristics at once and yet be effective? In addition, supervised validation data of the intended users were needed in order to assess the neural network's efficiency during training and establish when to terminate it. If the destination subject does not have a labeled validation set, will the network overfit the training users' dataset?

The current work creates Teacher-Student-Learning (TSL) and Heterogeneous Ensemble Hypotheses (HEH), a novel unsupervised User-Agnostic adaptation technique, to address these problems. The Heterogeneous Ensemble Hypotheses [31], which relies on the usage of two intelligent agents as opposed to one, is the source of inspiration for the several distinct hypotheses that make up the proposed HEH. Evry hypothesis is made up of a classifier as well as a feature generator. We assume that by optimizing the difference between several feature generators, the HEH may learn a variety of features. Then, acquiring knowledge from the pseudo labels of HEH and TSL, which is only a single a classifier as well as a feature generator and is a student of HEH, may acquire knowledge from HEH. As a result, the TSL can both stay efficient and improve its capacity for generalization. Additionally, the overfitting issue is resolved in this article by using a lightweight model. Fig. 1 depicts the overall architecture of the suggested TSL.

In addition, the primary motivation lies in addressing the challenge of creating intelligent and adaptable controllers for wearable robots by accurately anticipating the wearer's locomotion intent. The existing approaches face limitations due to the complexity and variability of human intent, leading to user-dependent interfaces. The paper proposes a novel unsupervised User-Agnostic adaptation technique, leveraging TSL and HEH, to overcome these challenges. By integrating multiple theories and refining loss functions, the proposed method effectively forecasts the user's unlabeled locomotor modes while preventing overfitting without requiring labeled end-point user data. The contributions include providing a theoretical justification for learning features from multi-view data, restructuring the model topology

Table 1
Recent works in user-agnostic adaptation and their relation to the proposed method.

Work	Approach	Novelty
Ganin et al. [23]	Domain-Adversarial Neural Networks (DANN)	- Focuses on global adaptation, which may fail due to inconsistent differences between training and end-point user data across classes.
Saito et al. [24]	Maximum Classifier Discrepancy (MCD)	- Aligns features of two domains in each class, performing better than DANN - However, the single feature generator may not be able to capture the full complexity of the data and all the hidden features
Zhang et al. [26]	Unsupervised User-Agnostic adaptation using CNNs	- Requires supervised validation data of the intended users to assess the neural network's efficiency during training. - To overfit on the training users' dataset when the target subject does not have a labeled validation set.
Proposed method	TSL and HEH	- Provides a theoretical justification for learning a complete set of features using multi-view data and ensemble methods. - Integrates multiple theories to effectively predict unlabeled locomotor modes. - Restructures the model to prevent overfitting without labeled end-point user data and predic in real-time.

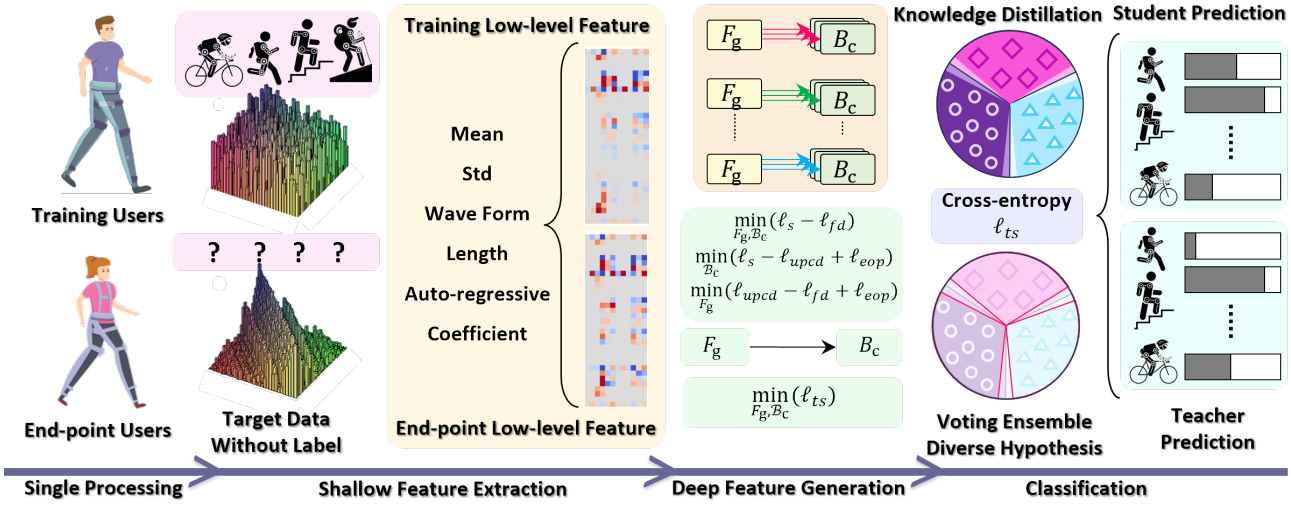

Figure 1: An overview of the TSL and HEH. The first human-robot interaction signals are used to extract low-level information like the standard deviation and mean. The HEH teacher network is the first to identify the end-point user characteristics. The goals of HEH's training are to reduce the classifier discrepancy's upper bound ℓ_{upcd}. To further improve accuracy, HEH also focuses on minimizing the training user error and entropy of output predictions that called ℓ_s and ℓ_{eop}, respectively. Additionally, HEH seeks to improve the accuracy of the classifier by maximizing the feature discrepancy ℓ_{fd}. The student network is trained using the teach predictions of HEH, which are effectively pseudo labels, following training. While end-point users' locomotor patterns may be reliably predicted by both TSL and HEH, the latter is significantly more effective.

for improved generalization, and achieving state-of-the-art accuracy levels in identifying locomotor intent and actions. Overall, the paper's contributions enhance clarity by offering a comprehensive solution to the user-agnostic adaptation of human locomotion intent, advancing the field of wearable robotics towards more intelligent and adaptable systems. The following are the main contributions of the current paper:

- Providing a theoretical justification for why a portion of features can be learned from multi-view data using the MCD single feature generator and how TSL and HEH are able to learn the entire set of features.
- Integrating several theories, refining the loss functions, and condensing the information to precisely and effectively forecast the target subject's unlabeled locomotor modes.
- Restructuring the model topology to prevent overfitting without the need for labeled end-point users data.
- Reaching cutting-edge accuracy levels (94.4% and 97.4%) in identifying the target subject's locomotor intent and actions on two public datasets.

The organizational structure of the paper is as follows: Section 2 introduces the Methodology, comprising subsections dedicated to 2.1 Theoretical Foundation and 2.2 Network Architecture. Following this, Section 3 presents the Result and Discussion, where the outcomes of the study are analyzed and interpreted.

2. Methodology

Before delving into the specifics of the proposed TSL and HEH for recognition human intent, In order to logically illustrate the drawbacks of the single model approach utilized in MCD and how the proposed method (HEH and TSL) overcome those drawbacks, the present part builds a thought experiment. Real data are typically "multi-view," as stated in [32, 27], meaning that they may have several hidden properties that may be used to categorize the input data. When there is a difference between the end-point users and

training user domain, **Proposition 1** shows that the single learner may not be capable of capturing the full complexity features of the multi-view data and all the hidden features from the input, which reduces the learner's capacity for generalization.

Proposition 1: Let's say we have two linear binary classifiers $\left(\mathcal{B}_{c1} \text{ and } \mathcal{B}_{c2}\right)$ and a feature generator $\left(F_g\right)$. Let the source and destination data be, respectively, S_x and T_x.S_x and T_x have labels $S_y \in \{0,1\}$ and $T_y \in \{0,1\}$, respectively. Let $\boldsymbol{o}_1, \boldsymbol{o}_2, \boldsymbol{o}_3$, and their three Concealed agnostic-feature be $\left\|\boldsymbol{o}_1\right\| = \left\|\boldsymbol{o}_2\right\| = \left\|\boldsymbol{o}_3\right\| = 1$ exist in every piece of S_x source data. By using supervised learning, it is possible to extract the linear combination of the hidden features, $\boldsymbol{c}_1\boldsymbol{o}_1 + \boldsymbol{c}_2\boldsymbol{o}_2 + \boldsymbol{c}_3\boldsymbol{o}_3$, from the feature generator F_g. These are the coefficients of various hidden features, denoted by $\left\{c_i \geq 0 \mid i \in \{1,2,3\}\right\}$. Since the classifier's activation function is set to ReLU, $\mathcal{B}\left(F_g(x)\right) > 0$. If $\mathcal{B}\left(F_g(x)\right) > 0$, the prediction is $\hat{y} = 1$, otherwise $\hat{y} = 0$.

Due to the differences between the training and end-point users' domains, only a portion of the end-point users' data (ratio $= r, r < 1$) has all three of the hidden characteristics $\left(\boldsymbol{o}_1, \boldsymbol{o}_2, \text{ and } \boldsymbol{o}_3\right)$. Assume that there is just all Concealed agnostic-feature $\left(\boldsymbol{o}_1, \boldsymbol{o}_2, \text{and } \boldsymbol{o}_3\right)$ in the $\frac{1-r}{3}$ end-point users' data. F_g may not be able to capture all the features of the end-point users' domain after training $\mathcal{B}_{c1}$ and $\mathcal{B}_{c2}$ to maximize their discrepancy. Also, this limitation arises as F_g concurrently attempts to minimize classifier discrepancy within the same domain. This implies that $\exists \boldsymbol{c} \in \left\{\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3\right\}, c = 0$. Since $\mathcal{B}_{c1}$ and $\mathcal{B}_{c2}$ are linear:

$$\begin{aligned}
&\mathcal{B}_{c1}\left(F_g(\boldsymbol{x})\right) = \mathcal{B}_{c1}\left(\boldsymbol{c}_1\boldsymbol{o}_1 + \boldsymbol{c}_2\boldsymbol{o}_2 + \boldsymbol{c}_3\boldsymbol{o}_3\right) = \\
&\boldsymbol{c}_1\mathcal{B}_{c1}\left(\boldsymbol{o}_1\right) + \boldsymbol{c}_2\mathcal{B}_{c1}\left(\boldsymbol{o}_2\right) + \boldsymbol{c}_3\mathcal{B}_{c1}\left(\boldsymbol{o}_3\right), \\
&\mathcal{B}_{c2}\left(F_g(\boldsymbol{x})\right) = \mathcal{B}_{c2}\left(\boldsymbol{c}_1\boldsymbol{o}_1 + \boldsymbol{c}_2\boldsymbol{o}_2 + \boldsymbol{c}_3\boldsymbol{o}_3\right) = \\
&\boldsymbol{c}_1\mathcal{B}_{c2}\left(\boldsymbol{o}_1\right) + \boldsymbol{c}_2\mathcal{B}_{c2}\left(\boldsymbol{o}_2\right) + \boldsymbol{c}_3\mathcal{B}_{c2}\left(\boldsymbol{o}_3\right)
\end{aligned} \tag{1}$$

Then, in the end-point users' domain, the classifier discrepancy is:

$$\begin{aligned}
\epsilon_{T_{F\,g}}\left(\mathcal{B}_{c1}, \mathcal{B}_{c2}\right) &= \boldsymbol{E}_{F_g(S_x)\sim T_{F\,g}}\left[\left|c_1\left(F_g\left(S_x\right)\right) - c_2\left(F_g\left(S_x\right)\right)\right|\right] \\
&= r\left|\mathcal{B}_{c1}\left(\boldsymbol{c}_1\boldsymbol{o}_1 + \boldsymbol{c}_2\boldsymbol{o}_2 + \boldsymbol{c}_3\boldsymbol{o}_3\right) - \mathcal{B}_{c2}\left(\boldsymbol{c}_1\boldsymbol{o}_1 + \boldsymbol{c}_2\boldsymbol{o}_2 + \boldsymbol{c}_3\boldsymbol{o}_3\right)\right| \\
&+ \frac{1-r}{3}\left[\left|\mathcal{B}_{c1}\left(\boldsymbol{c}_1\boldsymbol{o}_1\right) - \mathcal{B}_{c2}\left(\boldsymbol{c}_1\boldsymbol{o}_1\right)\right| + \left|\mathcal{B}_{c1}\left(\boldsymbol{c}_2\boldsymbol{o}_2\right) - \mathcal{B}_{c2}\left(\boldsymbol{c}_2\boldsymbol{o}_2\right)\right| + \left|\mathcal{B}_{c1}\left(\boldsymbol{c}_3\boldsymbol{o}_3\right) - \mathcal{B}_{c2}\left(\boldsymbol{c}_3\boldsymbol{o}_3\right)\right|\right] \\
&= \left(r + \frac{1-r}{3}\right)\left[\boldsymbol{c}_1\left|\mathcal{B}_{c1}\left(\boldsymbol{o}_1\right) - \mathcal{B}_{c2}\left(\boldsymbol{o}_1\right)\right| + \boldsymbol{c}_2\left|\mathcal{B}_{c1}\left(\boldsymbol{o}_2\right) - \mathcal{B}_{c2}\left(\boldsymbol{o}_2\right)\right| + \boldsymbol{c}_3\left|\mathcal{B}_{c1}\left(\boldsymbol{o}_3\right) - \mathcal{B}_{c2}\left(\boldsymbol{o}_3\right)\right|\right] \\
&\leq \frac{1+2r}{3}\left[\boldsymbol{c}_1\left|\mathcal{B}_{c1}\left(\boldsymbol{o}_1\right)\right| + \boldsymbol{c}_1\left|\mathcal{B}_{c2}\left(\boldsymbol{o}_1\right)\right| + \boldsymbol{c}_2\left|\mathcal{B}_{c1}\left(\boldsymbol{o}_2\right)\right| + \boldsymbol{c}_2\left|\mathcal{B}_{c2}\left(\boldsymbol{o}_2\right)\right| + \boldsymbol{c}_3\left|\mathcal{B}_{c1}\left(\boldsymbol{o}_3\right)\right| + \boldsymbol{c}_3\left|\mathcal{B}_{c2}\left(\boldsymbol{o}_3\right)\right|\right]
\end{aligned} \tag{2}$$

Thus, if S_x only has one feature, $\boldsymbol{o}_i$, then $F_g\left(S_x\right) = c_i\boldsymbol{o}_i, i \in \{1,2,3\}$. If and only if $\{\mathcal{B}_{c1}\left(\boldsymbol{o}_i\right)\}\mathcal{B}_{c2}\left(\boldsymbol{o}_i\right) \leq 0$ $\mid i \in \{1,2,3\}$, the maximum value is attained.

For each i in the range $\{1,2,3\}$, to maximize the $\epsilon_{T_{F\,g}}\left(\mathcal{B}_{c1}, \mathcal{B}_{c2}\right), \mathcal{B}_{c1}\left(\boldsymbol{o}_i\right)\mathcal{B}_{c2}\left(\boldsymbol{o}_i\right) \leq 0$. Since $j \in \{1,2\}$ and $\mathcal{B}_{cj}\left(\boldsymbol{o}_i\right) \geq 0$:

$$\exists \mathcal{B}_{cj}, \mathcal{B}_{cj}\left(\boldsymbol{o}_i\right) = 0, i \in \{1,2,3\}, j \in \{1,2\} \tag{3}$$

Due to the fact that $\mathcal{B}_{c1}$ and $\mathcal{B}_{c2}$ were taught to accurately recognize source data:

$$\begin{aligned}
\mathcal{B}_{c1}\left(F_g\left(S_x\right)\right) &= \mathcal{B}_{c1}\left(\boldsymbol{c}_1\boldsymbol{o}_1 + \boldsymbol{c}_2\boldsymbol{o}_2 + \boldsymbol{c}_3\boldsymbol{o}_3\right) \\
&= \boldsymbol{c}_1\mathcal{B}_{c1}\left(\boldsymbol{o}_1\right) + \boldsymbol{c}_2\mathcal{B}_{c1}\left(\boldsymbol{o}_2\right) + \boldsymbol{c}_3\mathcal{B}_{c1}\left(\boldsymbol{o}_3\right) > 0 \\
\mathcal{B}_{c2}\left(F_g\left(S_x\right)\right) &= \mathcal{B}_{c2}\left(\boldsymbol{c}_1\boldsymbol{o}_1 + \boldsymbol{c}_2\boldsymbol{o}_2 + \boldsymbol{c}_3\boldsymbol{o}_3\right) \\
&= \boldsymbol{c}_1\mathcal{B}_{c2}\left(\boldsymbol{o}_1\right) + \boldsymbol{c}_2\mathcal{B}_{c2}\left(\boldsymbol{o}_2\right) + \boldsymbol{c}_3\mathcal{B}_{c2}\left(\boldsymbol{o}_3\right) > 0
\end{aligned} \tag{4}$$

Integrating 3 and 4, it can be learned:

$$\begin{aligned}
&\#M_1 + \#M_2 \in \{3,4\}, \text{ for } i_1 \in M_1 \text{ and} \\
&i_2 \in M_2, \mathcal{B}_{c1}\left(\boldsymbol{o}_{i_1}\right) = \mathcal{B}_{c2}\left(\boldsymbol{o}_{i_2}\right) = 0
\end{aligned} \tag{5}$$

where $\#M_1$ and $\#M_2$ denote the number of elements in the corresponding sets, M_1 and M_2.

Without sacrificing generality, we may suppose that $\mathcal{B}_{c1}\left(\boldsymbol{o}_1\right) = 1, \mathcal{B}_{c1}\left(\boldsymbol{o}_2\right) = \mathcal{B}_{c1}\left(\boldsymbol{o}_3\right) = 0$ and $\mathcal{B}_{c2}\left(\boldsymbol{o}_1\right) = 0, \mathcal{B}_{c2}\left(\boldsymbol{o}_2\right) = \mathcal{B}_{c2}\left(\boldsymbol{o}_3\right) = 1$ if $\#M_1 + \#M_2 = 3$.

Owing to 1 and 4:

$$\begin{aligned}
&\mathcal{B}_{c1}\left(F_g\left(S_x\right)\right) = \boldsymbol{c}_1\mathcal{B}_{c1}\left(\boldsymbol{o}_1\right) + \boldsymbol{c}_2\mathcal{B}_{c1}\left(\boldsymbol{o}_2\right) + \boldsymbol{c}_3\mathcal{B}_{c1}\left(\boldsymbol{o}_3\right) \\
&= \boldsymbol{c}_1 > 0 \\
&\mathcal{B}_{c2}\left(F_g\left(S_x\right)\right) = \boldsymbol{c}_1\mathcal{B}_{c2}\left(\boldsymbol{o}_1\right) + \boldsymbol{c}_2\mathcal{B}_{c2}\left(\boldsymbol{o}_2\right) + \boldsymbol{c}_3\mathcal{B}_{c2}\left(\boldsymbol{o}_3\right) \\
&= \boldsymbol{c}_2 + \boldsymbol{c}_3 > 0
\end{aligned} \tag{6}$$

Training the F_g to reduce the classifier discrepancy is the final stage in the MCD process.

$$\begin{aligned}\min_{c_1,c_2,c_3} e_{T_{\mathrm{Fg}}}\left(\mathcal{B}_{c1},\mathcal{B}_{c2}\right)=&\\ \min_{c_1,c_2,c_3}\frac{1+2r}{3}&\left|c_1\mathcal{B}_{c1}\left(o_1\right)-c_1\mathcal{B}_{c2}\left(o_1\right)\right|\\ +&\left|c_2\mathcal{B}_{c1}\left(o_2\right)-c_2\mathcal{B}_{c2}\left(o_2\right)\right|\\ +&\left|c_3\mathcal{B}_{c1}\left(o_3\right)-c_3\mathcal{B}_{c2}\left(o_3\right)\right|\\ =&\min_{c_1,c_2,c_3}\frac{1+2r}{3}\left[c_1,c_2,c_3\right]\end{aligned}\tag{7}$$

The criterion of reducing the model discrepancy is met when $ci = 0$ for some $i \in 2,3$ because $c1 + c2 + c3 \geq c1 + c2$ and $c1 + c2 + c3 \geq c1 + c3$ due to the conditions $\{ci \geq 0 \mid i \in \{1,2,3\}\}$, and this is further supported by equations 6 and 7.

Given that if S_x only includes feature $o_i, i \in \{2,3\}$, then $\exists i \in \{2,3\}, k_i = 0, \mathcal{B}\left(F_g\left(S_x\right)\right) = 0$. Then, $1 - \frac{1-r}{3}$ is the target accuracy.

If a classifier can identify two features, then F_g not be able to capture all the features of the end-point users' domain, which is $\exists k \in \left\{c_1, c_2, c_3\right\}, c = 0$, as per the symmetry in the MCD training result for the single model. The target accuracy in this instance is $1 - \frac{1-r}{3}$.

Without sacrificing generality, we may suppose that $\mathcal{B}_{c1}\left(o_1\right) = 1, \mathcal{B}_{c1}\left(o_2\right) = \mathcal{B}_{c1}\left(o_3\right) = 0$ if $\#M_1 + \#M_2 = 2$. In contrast, $\mathcal{B}_{c2}\left(o_1\right) = \mathcal{B}_{c2}\left(o_3\right) = 0, \mathcal{B}_{c2}\left(o_2\right) = 1$.

Minimizing the classifier discrepancy in this instance is equivalent to:

$$\begin{aligned}\min_{c_1,c_2,c_3} e_{T_{\mathrm{Fg}}}\left(\mathcal{B}_{c1},\mathcal{B}_{c2}\right)=&\\ \min_{c_1,c_2,c_3}&\frac{1+2r}{3}\\ &\left|c_1\mathcal{B}_{c1}\left(o_1\right)-c_1\mathcal{B}_{c2}\left(o_1\right)\right|\\ +&\left|c_2\mathcal{B}_{c1}\left(o_2\right)-c_2\mathcal{B}_{c2}\left(o_2\right)\right|+\\ &\left|c_3\mathcal{B}_{c1}\left(o_3\right)-c_3\mathcal{B}_{c2}\left(o_3\right)\right|\\ =&\min_{c_1,c_2,c_3}\frac{1+2r}{3}\left[c_1+c_2\right]\end{aligned}\tag{8}$$

where the likelihood of each event is 50% and the value of c_3 can be either 1 or 0, as it has no effect on the outcome. If, however, S_x just has feature o_3, then $\mathcal{B}\left(F_g\left(S_x\right)\right) = 0$. The desired level of precision is $1 - \frac{1-r}{3}$. Based on the two scenarios mentioned above, there's a strong likelihood (p > 50%) that the single F_g will not be able to capture all the features. Furthermore, for the single learner in this paper, the end-point users' accuracy is $1 - \frac{1-r}{3}$.

Only a portion of the characteristics may be learned by the lone student. Ensemble many learners, on the other hand, could learn every characteristic. We shall show in **Proposition** 2 that feature generators can learn all features by optimizing feature diversity.

Proposition 2: Let us consider three feature generators F_{g_1}, F_{g_2}, and F_{g_3}. The linear combination of three concealed agnosticfeature, $c_{i1}o_1 + c_{i2}o_2 + c_{i3}o_3$, may be learned by each feature generator. $\left\|o_1\right\| = \left\|o_2\right\| = \left\|o_3\right\| = 1$. Following the optimization of feature diversity (Manhattan distance):

$$\ell_{fd} = \boldsymbol{E}_S\left[\sum_{i=1}^{3}\left|F_{g_i}(S) - \boldsymbol{E}_{i\in[1,3]}F_{g_i}(S)\right|\right]\tag{9}$$

The group of three students will discover all hidden features:

$$\exists i, o_{ij} > 0, \forall j \in \{1,2,3\}, \quad i \in \{1,2,3\}\tag{10}$$

Proof: **Proposition 1** states that feature discrepancy is:

$$\begin{aligned}\ell_{fd} =& \lambda_s \boldsymbol{E}_{S_x}\left[\sum_{i=1}^{3}\left|F_{g_i}\left(S_x\right) - \boldsymbol{E}_{i\in[1,3]}F_{g_i}\left(S_x\right)\right|\right]\\ &+ \lambda_t \boldsymbol{E}_{T_x}\left[\sum_{i=1}^{3}\left|F_{g_i}\left(T_x\right) - \boldsymbol{E}_{i\in[1,3]}F_{g_i}\left(T_x\right)\right|\right]\\ =&\left(\lambda_s + \lambda_t r\right)\sum_{i=1}^{3}\left|\sum_{j=1}^{3}\left(c_{ij} - \boldsymbol{E}_{i\in[1,3]}c_{ij}\right)o_j\right|\\ &+ \lambda_t\frac{1-r}{3}\sum_{i=1}^{3}\sum_{j=1}^{3}\left|\left(c_{ij} - \boldsymbol{E}_{i\in[1,3]}c_{ij}\right)o_j\right|\end{aligned}\tag{11}$$

where λ_s and λ_t denote the ratios, respectively, of the sample numbers of the end-point user and training user data to the total sample numbers.

The Manhattan distance is calculated as follows:

$$\left|\sum_{i=1}^{3} c_i o_i\right| = \sum_{i=1}^{3}\left|c_i\left\|o_i\right\|\right| = \sum_{i=1}^{3}\left|c_i\right|\tag{12}$$

Hence:

$$\begin{aligned}\ell_{fd} =& \left(\lambda_s + \lambda_t r\right)\sum_{i=1}^{3}\sum_{j=1}^{3}\left|c_{ij} - \boldsymbol{E}_{i\in[1,3]}c_{ij}\right|\\ &+ \lambda_t\frac{1-r}{3}\sum_{i=1}^{3}\sum_{j=1}^{3}\left|c_{ij} - \boldsymbol{E}_{i\in[1,3]}c_{ij}\right|\\ =&\left(\lambda_s + \lambda_t\frac{1+2r}{3}\right)\sum_{i=1}^{3}\sum_{j=1}^{3}\left|c_{ij} - \boldsymbol{E}_{i\in[1,3]}c_{ij}\right|\end{aligned}\tag{13}$$

Without loss of generality, we can assume $c_{1j} \leq c_{2j} \leq c_{3j}$. Then:

$$\begin{aligned}\sum_{i=1}^{3}\left|c_{ij}-E_{i\in[1,3]}c_{ij}\right| &= E_{i\in[1,3]}c_{ij}-c_{1j}+c_{3j}-E_{i\in[1,3]}c_{ij}+\\ &\left|c_{2j}-E_{i\in[1,3]}c_{ij}\right|\\ &= c_{3j}-c_{1j}+\left|c_{2j}-\frac{c_{1j}+c_{2j}+c_{3j}}{3}\right|\\ &= c_{3j}-c_{1j}+\left|\frac{2}{3}\left(c_{2j}-\frac{c_{1j}+c_{3j}}{2}\right)\right|\\ &\leq \frac{4}{3}\left(c_{3j}-c_{1j}\right)\end{aligned} \tag{14}$$

where the maximum value is achieved when $\boldsymbol{c}_{2j}=\boldsymbol{c}_{3j}$ or $\boldsymbol{c}_{2j}=\boldsymbol{c}_{1j}$ and $\boldsymbol{c}_{1j}=0$.

Maximizing the feature discrepancy will therefore result in $\min\{\boldsymbol{c}_{ij} \mid i\in\{1,2,3\}\}=0$ as well as $\max\{\boldsymbol{c}_{ij} \mid i\in\{1,2,3\}\}>0$. For each j in the interval $j\in\{1,2,3\}$, argmax $i\in\{1,2,3\}\boldsymbol{c}_{ij},\boldsymbol{c}_{ij}>0$.. We shall thus discover all three concealed agnostic-feature.

Every feature generator ought to pick up at least two concealed agnostic-features, in Eq. 6. As a result, $F_{\mathrm{g}}\left(S_x\right)=\boldsymbol{c}_{il}\boldsymbol{o}_l+\boldsymbol{c}_{im}\boldsymbol{o}_m, j\neq m$, and for any S_x that only has one $\boldsymbol{o}_l, \exists i=\mathrm{argmin}_{i\in\{1,2,3\}}, \boldsymbol{c}_{ij}=0 \mid l\neq j$.

According to Eq. 2, this paper shown $\mathcal{B}_{\mathrm{ci1}}\left(\boldsymbol{o}_m\right)\times\mathcal{B}_{\mathrm{ci2}}\left(\boldsymbol{o}_m\right)\leq 0$ due to the $\mathcal{B}_{\mathrm{c}}$ are trained to enhance the discrepancy between classifiers, as demonstrated in Eq. 2. Consequently, $\mathcal{B}_{\mathrm{cin}}\boldsymbol{o}_m=0, \exists n\in\{1,2\}$. Eq. 4 states that the training user data must be correctly classified:

$$\begin{aligned}&\mathcal{B}_{\mathrm{cin}}\left(F_{\mathrm{g}_i}\left(S_x\right)\right)>0\\ &\mathcal{B}_{\mathrm{cin}}\left(\boldsymbol{c}_{il}\boldsymbol{o}_l+\boldsymbol{c}_{\mathrm{im}}\boldsymbol{o}_m\right)>0\\ &\boldsymbol{c}_{il}\mathcal{B}_{\mathrm{cin}}\boldsymbol{o}_l+0>0\\ &\mathcal{B}_{\mathrm{cin}}\boldsymbol{o}_l>0\end{aligned} \tag{15}$$

In summary $\mathcal{B}_{\mathrm{cin}}\left(F_{\mathrm{g}_i}\left(S_x\right)\right)=\boldsymbol{c}_{il}\mathcal{B}_{\mathrm{cin}}\left(\boldsymbol{o}_l\right)>0$ for each S_x that includes feature $\boldsymbol{o}_l$, where $\exists i\in\{1,2,3\}$ and $n\in\{1,2\}$.

Consequently, in this theoretical experiment, the ensemble learners' end-point users accuracy is 100%.

Proposition 3: Distillation of Knowledge Assume that the teacher ensemble learners are $\{F_{\{\mathrm{g}_i\}},\mathcal{B}_{\mathrm{cin}} \mid n\in\{1,2\}, i\in\{1,2,3\}\}$ as illustrated in **Proposition 2**. This demonstrates the power of pseudo labels in enabling a student model with Distillation knowledge when utilized just one F_{g_s} and one $\mathcal{B}_{\mathrm{cs}}$ to classify the intent of all end-point users $F_{\mathrm{g}_s}(S)=\boldsymbol{c}_{s1}\boldsymbol{o}_1+\boldsymbol{c}_{s2}\boldsymbol{o}_2+\boldsymbol{c}_s\boldsymbol{o}_3>0$, indicating $\left\{\mathcal{B}_{\mathrm{cs}}\left(\boldsymbol{o}_j\right)>0 \mid j\in\{1,2,3\}\right\}$ and $\left\{\boldsymbol{c}_{sj}>0 \mid j\in\{1,2,3\}\right\}$.

Proof. The soft label is then used by the student learner to train and identify S_x that solely contains feature $\boldsymbol{o}_j.\mathcal{B}_{\mathrm{cin}}\left(F_{\mathrm{g}_i}\left(S_x\right)\right)>0$, as demonstrated in **Proposition 2** for $n\in\{1,2\}$ and $\exists i\in\{1,2,3\}$. Next, the soft label $\bar{y}\left(S_x\right)>0$, which is the ensemble learners' average, is shown. Once the cross entropy between $\bar{y}\left(S_x\right)$ and $\mathcal{B}_{\mathrm{cs}}\left(F_{\mathrm{g}_s}\left(S_x\right)\right)$ has been minimized, $\mathcal{B}_{\mathrm{cs}}\left(F_{\mathrm{g}_s}\left(S_x\right)\right)>0$. Given that single feature $\boldsymbol{o}_j$ is present in S_x

$$\begin{aligned}&\mathcal{B}_{\mathrm{cs}}\left(F_{\mathrm{g}_s}\left(S_x\right)\right)=\mathcal{B}_{\mathrm{cs}}\left(\boldsymbol{c}_{sj}\boldsymbol{o}_j\right)=\boldsymbol{c}_{sj}\mathcal{B}_{\mathrm{cs}}\left(\boldsymbol{o}_j\right)>0\\ &\rightarrow \boldsymbol{c}_{sj}>0 \text{ and } \mathcal{B}_{\mathrm{cs}}\left(\boldsymbol{o}_j\right)>0, j\in\{1,2,3\}\end{aligned} \tag{16}$$

In a theoretical setting, the aforementioned theorems highlight the drawbacks of the solitary learner as well as the benefits of the TSL and HEH. The TSL and HEH strengths and specifics of its implementation will be covered in more detail in the sections that follow.

2.1. Theoretical Foundation

This paper proposes TSL and HEH, which has the potential to learn a variety of characteristics and reduce the upper limit of the error in forecasting end-point user intent. A subset of features can only be learned by a single learner, as explained before. Furthermore, using an Neural Network (NN) does not ensure that the global optimum will be found because of the huge number of parameters in the network. In addition, the optimization process of the unsupervised user-agnostic adaptation is complex and challenging due to the loss function characterized by non-convex. The problem could be mitigated by grouping many students. However, the main justification for creating HEH is that learning integration is only beneficial if the learners provide unique predictions.

Proposition 4. The average hypothesis $\overline{\boldsymbol{\mu}}$ is considered for all the individual hypotheses $\boldsymbol{\mu}_{\boldsymbol{c}}$, and denote the hypothesis errors $\boldsymbol{\mu}_{\boldsymbol{c}}$, errors $\overline{\boldsymbol{\mu}}$, as well as the discrepancy between $\boldsymbol{\mu}_{\boldsymbol{c}}$ and $\overline{\boldsymbol{\mu}}$ as $\varepsilon\left(\boldsymbol{\mu}_{\boldsymbol{c}}\right), \varepsilon(\overline{\boldsymbol{\mu}})$, and $\ell\left(\boldsymbol{\mu}_{\boldsymbol{c}}\right)$ respectively. The $\varepsilon(\overline{\boldsymbol{\mu}})$ is calculated as follows:

$$\varepsilon(\overline{\boldsymbol{\mu}})=\mathbf{e}_{\boldsymbol{c}}\varepsilon\left(\boldsymbol{\mu}_{\boldsymbol{c}}\right)-\mathbf{e}_{\boldsymbol{c}}\ell\left(\boldsymbol{\mu}_{\boldsymbol{c}}\right) \tag{17}$$

Input data and label are represented by S and y. e stands for the expected. Additionally,

$$\begin{aligned}&\varepsilon(\overline{\boldsymbol{\mu}})=\mathbf{e}_S(\overline{\boldsymbol{\mu}}(S)-y)^2,\\ &\varepsilon\left(\boldsymbol{\mu}_{\boldsymbol{c}}\right)=\mathbf{e}_S\left(\boldsymbol{\mu}_{\boldsymbol{c}}(S)-y\right)^2,\\ &\ell\left(\boldsymbol{\mu}_{\boldsymbol{c}}\right)=\mathbf{e}_S\left(\boldsymbol{\mu}_{\boldsymbol{c}}(S)-\overline{\boldsymbol{\mu}}(S)\right)^2\end{aligned} \tag{18}$$

Proof.

$$\begin{aligned}\mathbf{e}_{\boldsymbol{c}}\ell\left(\boldsymbol{\mu}_k\right) &= \mathbf{e}_{\boldsymbol{c}}\mathbf{e}_S\left(\boldsymbol{\mu}_{\boldsymbol{c}}(S)-\overline{\boldsymbol{\mu}}(S)\right)^2\\ &= \mathbf{e}_{\boldsymbol{c}}\mathbf{e}_S\left[\boldsymbol{\mu}_{\boldsymbol{c}}(S)-y-\left(\boldsymbol{\mu}_{\boldsymbol{c}}(S)-y\right)\right]^2\\ &= \mathbf{e}_{\boldsymbol{c}}\mathbf{e}_S\left[\left(\boldsymbol{\mu}_{\boldsymbol{c}}(S)-y\right)^2+\left(\boldsymbol{\mu}_{\boldsymbol{c}}(S)-y\right)^2\right]\\ &- 2\mathbf{e}_{\boldsymbol{c}}\mathbf{e}_S\left[\left(\boldsymbol{\mu}_{\boldsymbol{c}}(S)-y\right)\left(\boldsymbol{\mu}_{\boldsymbol{c}}(S)-y\right)\right]\\ &= \mathbf{e}_{\boldsymbol{c}}\mathbf{e}_S\left[\left(\boldsymbol{\mu}_{\boldsymbol{c}}(S)-y\right)^2-\left(\boldsymbol{\mu}_{\boldsymbol{c}}(S)-y\right)^2\right]\\ &= \mathbf{e}_{\boldsymbol{c}}\mathrm{e}_S\left(\boldsymbol{\mu}_{\boldsymbol{c}}(S)-y\right)^2-\mathbf{e}_S\left(\boldsymbol{\mu}_{\boldsymbol{c}}(S)-y\right)^2\\ &= \mathbf{e}_{\boldsymbol{c}}\varepsilon\left(\boldsymbol{\mu}_{\boldsymbol{c}}\right)-\varepsilon(\boldsymbol{\mu}).\end{aligned} \tag{19}$$

Algorithm 1 User-Agnostic Adaptation using TSL and HEH

procedure USER-AGNOSTIC-ADAPTATION
  Input:
    Source domain dataset $\mathcal{D}_S = \{(x_i, y_i)\}_{i=1}^{N_S}$ with N_S labeled examples
    Target domain dataset $\mathcal{D}_T = \{x_j\}_{j=1}^{N_T}$ with N_T unlabeled examples
  Output: Adapted model $\mathcal{B}_c^s$ capable of classifying target domain data

  Initialization:
    n_{F_g} feature generators $\{F_g^i\}_{i=1}^{n_{F_g}}$ for HEH
    $n_{\mathcal{B}_c}$ classifiers $\{\mathcal{B}_c^k\}_{k=1}^{n_{\mathcal{B}_c}=n_{F_g}\times n_{\mathcal{B}_c/F_g}}$ with $n_{\mathcal{B}_c/F_g}$ classifiers per generator
    Single classifier $\mathcal{B}_c^s$ for final decision
    Optional domain classifiers $\{D^m\}_{m=1}^{n_D}$ for adaptation

  for each F_g^i in $\{F_g^i\}_{i=1}^{n_{F_g}}$ **do**
    Train F_g^i using ℓ_s loss on $\mathcal{D}_S$ to extract features
    Extract diverse features $\mathcal{F}_i = \{F_g^i(x) \mid x \in \mathcal{D}_S \cup \mathcal{D}_T\}$
  end for
  Combine extracted feature sets $\{\mathcal{F}_i\}_{i=1}^{n_{F_g}}$ using ensemble: $\mathcal{F}_{ens} = \{(F_g^1(x), ..., F_g^{n_{F_g}}(x)) \mid x \in \mathcal{D}_S \cup \mathcal{D}_T\}$

  Apply Teacher-Student Learning:
  Teacher Phase:
  for each $(F_g^i, \mathcal{B}_c^k)$ pair **do**
    Train using $\ell_s, \ell_{upcd}, \ell_{fd}, \ell_{eop}$ losses
    Use ensemble $\{(F_g^i, \mathcal{B}_c^k)\}$ to predict pseudo-labels $\mathcal{D}_T^{pseudo} = \{(x_j, y_j^{pseudo})\}_{j=1}^{N_T}$
  end for
  Student Phase:
  Train $\mathcal{B}_c^s$ on $\mathcal{D}_T^{pseudo}$ using ℓ_{ts} loss to match teacher predictions
  Optional: Use $\{D^m\}$ and adversarial loss to further adapt $\mathcal{B}_c^s$

  Evaluate $\mathcal{B}_c^s$ on target domain data and benchmark datasets
  Output: The adapted efficient model $\mathcal{B}_c^s$
end procedure

Therefore, $\varepsilon(\bar{\mu}) = \mathbf{e}_c \varepsilon\left(\mu_c\right) - \mathbf{e}_c \ell\left(\mu_c\right)$.

If the disparity between the hypotheses is not zero, **Proposition 4** shows that the average hypothesis's can provide more accurate and more reliable predictions than individual hypotheses. If the gap between the hypotheses widens and the accuracy of each individual hypothesis increases, the average hypothesis' accuracy will also jump. Because HEH may enhance the disparity between distinct hypotheses, it can raise the accuracy of anticipating the subject's purpose.

The second rationale for suggesting HEH is that it could reduce the inaccuracy of the $\varepsilon\left(\boldsymbol{\mu_c}\right)$ in the end-point users. Also, the estimating the disparity between hypotheses is easy and does not need data labels, estimating the inaccuracy of $\varepsilon\left(\boldsymbol{\mu_c}\right)$ in the end-point users' domain-where endpoint users labels are unavailable - is more difficult. By calculating the upper bound of each $\varepsilon\left(\boldsymbol{\mu_c}\right)$ at the end-point users' domain, the current study tackles this problem.

Proposition 5. Assume that $\boldsymbol{\mu}$ is a hypothesis made up of a classifier $\mathcal{B}_{\mathrm{c}}$ and a feature generator F_{g} such that $\boldsymbol{\mu} = \mathcal{B}_{\mathrm{c}} \circ F_{\mathrm{g}}$. Let $S_{F_{\mathrm{g}}}$ and $T_{F_{\mathrm{g}}}$ be the changed end-point and training users domains using F_{g}, and let $\mathcal{B}_{\mathrm{s}}$ be the classifier space. The upper bound of the end-point users intent classification error $\varepsilon_{T_{F_{\mathrm{g}}}}\left(\mathcal{B}_{\mathrm{C}}\right)$ is:

$$\begin{aligned} &\varepsilon_{T_{F\,\mathrm{g}}}\left(\mathcal{B}_{\mathrm{c}}\right) \leq \lambda + \varepsilon_{S_{F\,\mathrm{g}}}\left(\mathcal{B}_{\mathrm{c}}\right) + d_{\mathcal{B}_{\mathrm{c}}\Delta\mathcal{B}_{\mathrm{c}}}\left(S_{F_{\mathrm{g}}}, T_{F_{\mathrm{g}}}\right)/2, \\ &\lambda = \min_{\mathcal{B}_{\mathrm{c}}\in\mathcal{B}_{\mathrm{s}}}\left(\varepsilon_{S_{F\,\mathrm{g}}}\left(\mathcal{B}_{\mathrm{c}}\right) + \epsilon_{T_g}\left(\mathcal{B}_{\mathrm{c}}\right)\right), \end{aligned} \tag{20}$$

$$\begin{aligned} &d_{\mathcal{B}_{\mathrm{c}}\Delta\mathcal{B}_{\mathrm{c}}}\left(S_{F_{\mathrm{g}}}, T_{F_{\mathrm{g}}}\right) = \\ &2 \sup_{\mathcal{B}_{\mathrm{c}1}\mathcal{B}_{\mathrm{c}2}\in C}\left|\varepsilon_{T_{F\,\mathrm{g}}}\left(\mathcal{B}_{\mathrm{c}1}, \mathcal{B}_{\mathrm{c}2}\right) - \varepsilon_{S_{F_{\mathrm{g}}}}\left(\mathcal{B}_{\mathrm{c}1}, \mathcal{B}_{\mathrm{c}2}\right)\right| \end{aligned} \tag{21}$$

$$\begin{aligned} &\varepsilon_D\left(\mathcal{B}_{\mathrm{c}}\right) = \mathrm{e}_{F_{\mathrm{g}}(S)\sim D}\left[\left|\mathcal{B}_{\mathrm{c}}\left(F_{\mathrm{g}}(S)\right) - y\right|\right], \\ &\varepsilon_D\left(\mathcal{B}_{\mathrm{c}1}, \mathcal{B}_{\mathrm{c}2}\right) = \mathrm{e}_{F_{\mathrm{g}}(S)\sim D}\left[\left|\mathcal{B}_{\mathrm{c}1}\left(F_{\mathrm{g}}(S)\right) - \mathcal{B}_{\mathrm{c}1}\left(F_{\mathrm{g}}(S)\right)\right|\right], \end{aligned} \tag{22}$$

where the classification errors of $\mathcal{B}_{\mathrm{c}}$ as well as two classifiers discrepancy, $\mathcal{B}_{\mathrm{c}1}$ and $\mathcal{B}_{\mathrm{c}2}$, in the domain $D\left(D = S_{F_{\mathrm{g}}}\right.$ or $\left.T_{F_{\mathrm{g}}}\right)$, are denoted by the variables $\varepsilon_D\left(\mathcal{B}_{\mathrm{c}}\right)$ and $\varepsilon_D\left(\mathcal{B}_{\mathrm{c}1}, \mathcal{B}_{\mathrm{c}2}\right)$, respectively. The $\mathcal{B}_{\mathrm{c}}\Delta\mathcal{B}_{\mathrm{c}}$-distance is shown by $d_{\mathcal{B}_{\mathrm{c}}\Delta\mathcal{B}_{\mathrm{c}}}\left(S_{F_{\mathrm{g}}}, T_{F_{\mathrm{g}}}\right)$, where λ stands for the integrated error in the converted end-point and training user users domain, sup for the upper limit, as well as e for the expectation.

Proof: A triangle inequality for classification error was demonstrated by [33, 34] for any classification functions $\mathcal{B}_{\mathrm{c}1}, \mathcal{B}_{\mathrm{c}2}$, and $\mathcal{B}_{\mathrm{c}3}$, where $\varepsilon\left(c_1, c_2\right) \leq \varepsilon\left(c_1, c_3\right) + \varepsilon\left(c_2, c_3\right)$. Therefore, based on Eq. 23, where $c^* = \mathrm{argmin}_{\mathcal{B}_{\mathrm{c}}\in\mathcal{B}_{\mathrm{s}}}\left(\varepsilon_{S_{F\,\mathrm{g}}}\left(\mathcal{B}_{\mathrm{c}}\right) + \varepsilon_{S_{T_{\mathrm{g}}}}\left(\mathcal{B}_{\mathrm{c}}\right)\right)$.

Propositions 4 and **5** show that by decreasing the training user error, the integrated error, the $\mathcal{B}_{\mathrm{c}}\Delta\mathcal{B}_{\mathrm{c}}$-distance as well as increasing the discrepancy of hypotheses, HEH not only reduces the error upper bound, but also ensures more accurate predictions in the end-point user domain. It is possible to derive the matching loss functions that might reduce the end-point users' error. This section will provide a detailed introduction to a number of potential loss functions and the theoretical underpinnings that support them.

$$
\begin{aligned}
\varepsilon_{T_{F\ g}}\left(\mathcal{B}_{\mathrm{c}}\right) &\leq \varepsilon_{T_{F\ g}}\left(\mathcal{B}_{\mathrm{c}}{}^{*}\right)+\varepsilon_{T_{F\ g}}\left(\mathcal{B}_{\mathrm{c}}, \mathcal{B}_{\mathrm{c}}{}^{*}\right) \\
&\leq \varepsilon_{T_{F\ g}}\left(\mathcal{B}_{\mathrm{c}}{}^{*}\right)+\varepsilon_{S_{F\ g}}\left(\mathcal{B}_{\mathrm{c}}, \mathcal{B}_{\mathrm{c}}{}^{*}\right)+\left|\varepsilon_{T_{F\ g}}\left(\mathcal{B}_{\mathrm{c}}, \mathcal{B}_{\mathrm{c}}{}^{*}\right)-\varepsilon_{S_{F\ g}}\left(\mathcal{B}_{\mathrm{c}}, \mathcal{B}_{\mathrm{c}}{}^{*}\right)\right| \\
&\leq \varepsilon_{T_{F\ g}}\left(\mathcal{B}_{\mathrm{c}}{}^{*}\right)+\varepsilon_{S_{F\ g}}\left(\mathcal{B}_{\mathrm{c}}\right)+\varepsilon_{S_{F\ g}}\left(\mathcal{B}_{\mathrm{c}}{}^{*}\right)+\left|\varepsilon_{T_{F\ g}}\left(\mathcal{B}_{\mathrm{s}}, \mathcal{B}_{\mathrm{c}}{}^{*}\right)-\varepsilon_{S_{F\ g}}\left(\mathcal{B}_{\mathrm{c}}, \mathcal{B}_{\mathrm{c}}{}^{*}\right)\right| \\
&\leq \varepsilon_{S_{F\ g}}\left(\mathcal{B}_{\mathrm{c}}\right)+\lambda+\left|\varepsilon_{T_{F\ g}}\left(\mathcal{B}_{\mathrm{c}}, \mathcal{B}_{\mathrm{c}}{}^{*}\right)-\varepsilon_{S_{F\ g}}\left(\mathcal{B}_{\mathrm{c}}, \mathcal{B}_{\mathrm{c}}{}^{*}\right)\right| \\
&\leq \varepsilon_{S_{F\ g}}\left(\mathcal{B}_{\mathrm{c}}\right)+\lambda+\sup\left|\varepsilon_{T_{F\ g}, \mathcal{B}_{\mathrm{c}2}}\left(\mathcal{B}_{\mathrm{c}1}, \mathcal{B}_{\mathrm{c}2}\right)-\varepsilon_{S_{F\ g}}\left(\mathcal{B}_{\mathrm{c}1}, \mathcal{B}_{\mathrm{c}2}\right)\right| \\
&= \varepsilon_{S_{F\ g}}\left(\mathcal{B}_{\mathrm{c}}\right)+\lambda+d_{\mathcal{B}_{\mathrm{c}} \Delta \mathcal{B}_{\mathrm{c}}}\left(S_{F_{\mathrm{g}}}, T_{F_{\mathrm{g}}}\right)/2,
\end{aligned}
\tag{23}
$$

2.1.1. Training User Error

The end-point users' error and the training users error have a positive correlation, according to **Proposition 5**. Reducing the crossentropy between the labels S_y of the training user data S_x and the classification results is a straightforward way to reduce the error on the training user domain:

$$
\ell_{\mathrm{s}}=\mathrm{e}_{\mathrm{S} \sim D_S} \sum_{n=1}^{N}-I\left[n=S_y\right] \log P_n\left(y \mid S_x\right) \tag{24}
$$

using the expectation operator denoted by e. The expected probability is illustrated by P_n for class n. An indicator function, denoted as $I\left[n=S_y\right]$, can be represented as follows:

$$
I\left[n=S_y\right]=\begin{cases}1, & \text{if } n=S_y \\ 0, & \text{otherwise}\end{cases} \tag{25}
$$

This is a piecewise function that outputs 1 when n is equal to S_y, and 0 otherwise.

2.1.2. Integrated Error Minimization

The classifier space capacity as well as the bias between the training user and end-point users domains determine the minimal combined error λ. It is unlikely to construct a HEH that processes well in both the training user and end-point users domains if the minimal integrated error λ is really big, according to $\mathcal{B}_{\mathrm{s}}$ [35]. If the classifier space has a large enough capacity, λ is often small. Since the classifiers in this study are neural networks with big enough parameters to construct a high-capacity classifier space, it may be assumed that λ is modest.

2.1.3. Distance Minimization

The $\mathcal{B}_{\mathrm{c}}\Delta\mathcal{B}_{\mathrm{c}}$-distance is the last item in Eq. 19 after the training user error $\varepsilon_{S_{F\ g}}\left(\mathcal{B}_{\mathrm{c}}\right)$ and the combination error λ are minimized. The difference between the disagreement of $\mathcal{B}_{\mathrm{c}1}$ and $\mathcal{B}_{\mathrm{c}2}$ on the converted training user domain and that in the end-point users' domain is measured by the $\mathcal{B}_{\mathrm{c}}\Delta\mathcal{B}_{\mathrm{c}}$-distance in Eq. 19. Following optimization, the classifier domain may shift towards the location characterized by the lowest training user error. This shift occurs during the initial training of F_{g} and $\mathcal{B}_{\mathrm{c}}$ using the cross-entropy loss ℓ_s described in equation Eq. 20. The objective is to minimize the training user error throughout this process. The disagreement between two random $\mathcal{B}_{\mathrm{c}}$ on the training user's domain in the optimized classifier space $\mathcal{B}_{\mathrm{s}}$ will be modest, implying a small $\varepsilon_{S_{\mathrm{Fg}}}\left(\mathcal{B}_{\mathrm{c}1}, \mathcal{B}_{\mathrm{c}2}\right)$. Consequently, the $\mathcal{B}_{\mathrm{c}}\Delta\mathcal{B}_{\mathrm{c}}$-distance demonstrates that a minor modification to the classifier in the training user domain can lead to a significant modification in the classifier in the end-point users' domain. The disagreement upper bound of $\mathcal{B}_{\mathrm{c}1}$ and $\mathcal{B}_{\mathrm{c}2}$ at the end-point users should be minimized to decrease the $\mathcal{B}_{\mathrm{c}}\Delta\mathcal{B}_{\mathrm{c}}$ - distance:

$$
\min_{T_{F\ g}} \sup_{\mathcal{B}_{\mathrm{c}1}, \mathcal{B}_{\mathrm{c}2} \in \mathcal{B}_{\mathrm{c}}} \mathrm{e}_{F_{\mathrm{g}}(S) \sim T_{F\ g}}\left[\mid \mathcal{B}_{\mathrm{c}1}\left(F_{\mathrm{g}}(S)-\mathcal{B}_{\mathrm{c}2}\left(F_{\mathrm{g}}(S)\right.\right.\right] \tag{26}
$$

Although the classifiers' disagreement may be quantified, its upper bound is not immediately quantifiable. We transform equation Eq. 21 into a min-max problem to determine the upper bound. Two distinct classifiers, $\mathcal{B}_{\mathrm{c}1}$ and $\mathcal{B}_{\mathrm{c}2}$, can alter the discrepancy, and a feature generator, F_{g}, can alter the converted end-point user domain, $T_{F_{\mathrm{g}}}$. The classifiers $\mathcal{B}_{\mathrm{c}1}$ and $\mathcal{B}_{\mathrm{c}2}$ have been learned to identify the discrepancy upper bound of hypotheses as well as to maximize the discrepancy between each other. To minimize the classifier discrepancy, F_{g} is learned to reduce its upper bound. This step aims to enhance the model's performance by reducing discrepancy in its classifier. The entire procedure may be finished by:

$$
\min_{F_{\mathrm{g}}} \max_{\mathcal{B}_{\mathrm{c}1}, \mathcal{B}_{\mathrm{c}2}} \mathrm{e}_{F_{\mathrm{g}}(S) \sim T_{F\ g}}\left[\mid \mathcal{B}_{\mathrm{c}1}\left(F_{\mathrm{g}}(S)-\mathcal{B}_{\mathrm{c}2}\left(F_{\mathrm{g}}(S)\right.\right.\right] \tag{27}
$$

$\mathcal{B}_{\mathrm{c}1}, \mathcal{B}_{\mathrm{c}2}$, and F_{g} in (22), respectively, are insufficient to train the hypotheses in this study. The rationale is that in order to further reduce the goal error, the current study incorporates a number of distinct hypotheses, including several feature generators F_{g_i} and classifiers $\mathcal{B}_{c_k}$. The decrease of classifier discrepancy ℓ_{upcd} after adding more classifiers should be:

$$
\begin{aligned}
&\ell_{upcd} = \sum_{i=1}^{n_{F_g}} \sum_{j=1}^{n_{\mathcal{B}_c}} \ell_{upcd}^{ij}, \\
&\ell_{upcd}^{ij} = \mathrm{e}_{T_x \sim D_T} \left[\sum_{n=1}^{N} \left(\left| P_n^{ij}\left(T_y \mid T_x\right) - P_n^{i}\left(T_y \mid T_x\right) \right| \right) / N \right], \\
&P_n^{ij}\left(T_y \mid T_x\right) = \mathcal{B}_{c_k}\left(F_{g_i}\left(T_x\right)\right)[n], k = n_{\mathcal{B}_c}(i-1) + j, \\
&\bar{P}_n^{i}\left(T_y \mid T_x\right) = \mathrm{e}_{j \in \left[1, n_{\mathcal{B}_c}\right]} \left[P_n^{ij}\left(T_y \mid T_x\right) \right]
\end{aligned}
\tag{28}
$$

Hence, provided a hypothesis F_{g_i} called generator as well as $\mathcal{B}_{c_k}$ called classifier, ℓ_{upcd}^{ij} denotes the loss of classifier disagreement. $P_n^{ij}\left(T_y \mid T_x\right)$ denotes class n predictions in the case when F_{g_i} and $\mathcal{B}_{c_k}$ make up the hypothesis and T_x is the input.The initial end-point user domain is indicated by D_T, and the average prediction of many hypotheses with F_{g_i} as their generator is $\bar{P}_n^{i}\left(T_y \mid T_x\right)$

2.1.4. Feature Discrepancy Maximization

Proposition 2 shows that in addition to the classifier discrepancy, maximizing the feature discrepancy is also required to discover the whole set of hidden features. Furthermore, as **Proposition 4** demonstrates, the correctness of each hypothesis as well as the hypothesis variety can reduce the end-point users' error of the integrated hypotheses. Therefore, in order to maximize the feature discrepancy, the current work constructs a loss ℓ_{fd} :

$$
\ell_{fd} = \mathbf{e}_S \left[\sum_{i=1}^{n_{F_g}} \left| F_{g_i}(S) - \mathrm{e}_{i \in [1, n_{F_g}} F_{g_i}(S) \right| \right] \tag{29}
$$

2.1.5. Classifier Confidence Maximization

Using the cluster assumption, HEH can help to reduce the end-point users' error in their hypotheses. In unsupervised learning, where the data labels are undisclosed, the application of clustering techniques enables the analysis of the unlabeled dataset, revealing its intrinsic characteristics. Similarly, the unsupervised domain adaptation may benefit from the clustering method. By assuming the samples aligns with clusters, it can be segmented into distinct groups. The expectation is that data within each cluster should share the same category classification [36]. Consequently, high-density zones shouldn't be crossed by the decision border. The conditional entropy loss can be minimized in order to achieve this goal [37]:

$$
\begin{aligned}
&\ell_{eop} = -\sum_{j=1}^{n_{\mathcal{B}_c}} \sum_{i=1}^{n_{F_g}} \mathrm{e}_{T_x \sim D_{end-point}} \left[P^{ij}\left(T_y \mid T_x\right)^T \ln P^{ij}\left(T_y \mid T_x\right) \right], \\
&P^{ij}\left(T_y \mid T_x\right) = \mathcal{B}_{c_k}\left(F_{g_i}\left(T_x\right)\right), k = n_{\mathcal{B}_c}(i-1) + j
\end{aligned}
\tag{30}
$$

A few benefits come with reducing conditional entropy. Initially, the domain where the end-point users' data is dense may not be included in the optimized hypothesis space and the hypothesis space may be further restricted. Consequently, since altering the decision boundary in low-density areas won't result in a significant classifier disagreement in the end-point user domain, the $\mathcal{B}_c\Delta\mathcal{B}_{c^-}$ distance $d_{\mathcal{B}_c\Delta\mathcal{B}_c}\left(S_{F_g}, T_{F_g}\right)$ may be reduced. Then, when the $\mathcal{B}_c\Delta\mathcal{B}_c$-distance decreases, the upper bound of the goal error will drop as well. Second, the end-point users' data density information may be used by the decision boundary to limit its options, which might result in a suitable prior probability distribution and improve performance.

2.1.6. Feature Generator Optimization

The optimization process of the feature generator of the proposed model (TSL & HEH) involves a sophisticated multi-step approach aimed at enhancing the model's ability to adapt to end-point user's data by leveraging labeled training users' data. This process is crucial for user-agnostic adaptation, particularly in applications like classifying locomotion intent and activities without direct labeling of the end-point user's data. The optimization process is precisely designed to minimize overfitting, maximize feature discrepancy, and ensure efficient and accurate prediction of the end-point user's locomotion modes. The optimization process of the feature generator is divided into three main steps, each targeting specific aspects of the model's performance:

Minimization of Source Classification Error and Maximization of Feature Discrepancy: Initially, the feature generators and classifiers are trained to minimize the training users classification error (L_s) while maximizing the feature discrepancy ($L_{d,g}$). This step aims to ensure that the model can accurately classify the labeled training users' data while encouraging the feature generators to produce diverse representations that capture different aspects of the data. The optimization formula for this step is given by:

$$
\min_{g_i, c_k} L_s - \xi_g L_{d,g}, \tag{31}
$$

where g_i represents the feature generators, c_k the classifiers, and ξ_g is a constant weight for the loss function $L_{d,g}$.

Maximization of Classifier Discrepancy and Minimization of Entropy Loss: After the initial training, the classifiers are further optimized to maximize the classifier discrepancy ($L_{d,c}$) and minimize the entropy loss (L_e). This phase freezes the feature generators to focus on enhancing the classifiers' ability to distinguish between different classes confidently. The optimization formula for this phase is:

$$
\min_{c_k} L_s - \xi_c L_{d,c} + \xi_e L_e, \tag{32}
$$

where ξ_c and ξ_e are constant weights for the loss functions $L_{d,c}$ and L_e, respectively.

Minimization of the Upper Bound of Classifier Discrepancy: In the final step, the focus shifts back to the feature generators, which are now trained to minimize the upper

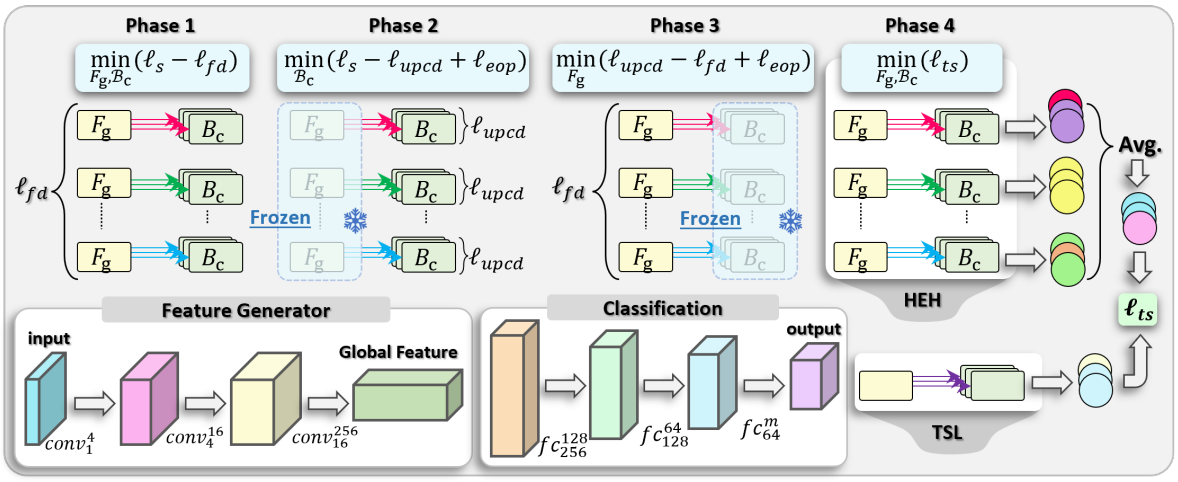

Figure 2: The suggested HEH and TSL and HEH's network design and training procedures are shown in $\mathcal{B}_c$ and F_g, which stand for a classifier and a feature generator, respectively. The notation in the convolutional layer uses a superscript to represent the input feature channels and a subscript to denote the output feature channels. Subsections 2.2 and 2.3 cover training procedures and loss functions.

bound of the classifier discrepancy (thus decreasing the $C\Delta C$-distance) and maximize the feature discrepancy. This step aims to refine the feature generators' ability to transform the end-point user's data into feature spaces that align well with the training users data, facilitating accurate classification of the end-point users data. The optimization formula for this step is:

$$\min_{g_i} L_{d,c} - L_{d,g} + \xi_e L_e, \tag{33}$$

where g_i represents the feature generators.

The optimization process of the feature generator in the proposed model (TSL & HEH) is a carefully structured approach that leverages the strengths of ensemble learning and knowledge distillation. By iteratively focusing on minimizing training users error, maximizing feature and classifier discrepancies, and minimizing entropy loss, the framework aims to achieve high accuracy in unsupervised user-agnostic adaptation tasks. This multi-step optimization process ensures that the model can effectively learn from labeled training data and generalize to unlabeled end-point users data, achieving state-of-the-art accuracy levels on public datasets.

2.2. Network Architecture

A neural network typically involves a delicate balance between its generalization and fitting capabilities. With more parameters and the ability to learn deeper features, a big network may overfit the training dataset, particularly in situations where the earlystop time is not determined by a end-point user validation set. Although a small neural network could match the training dataset more precisely than a big network, a lightweight neural network might be superior at generalization. The ensemble approach is used in this study to address this problem. Because it can help poor learners generate accurate predictions, the ensemble technique is appealing [38]. The ensemble method's generalization capacity can outperform that of a strong learner even when individual learner is poor. Furthermore, even a single learner can acquire information from the ensemble learner following the distillation of knowledge. Because of this feature, we are able to create a lightweight neural network (see Fig. 2) that increases efficiency and prevents overfitting.

Here, a convolutional neural network serves as the feature generator. In the first two convolutional layers, the kernel size and stride are (1×1) and 1, respectively. The capabilities of a multi-layer perceptron (MLP) in extracting deeper characteristics can be likened to those exhibited by these two layers. The third layer's kernel size is identical to the input image's size. This layer performs the convolution of each feature, resulting in a comprehensive global feature vector. To normalize signal amplitudes across diverse sensors and channels, batch normalization is implemented after each conv layer and preceding the activation function (ReLu6).

In this article, the classifier is constructed using three fully connected layers. The global feature is mapped to vector size 64 by the top two MLP using a ReLu6 and batch normalization. These hidden features are mapped to the input image's classification scores in the final MLP.

In the HEH, there are 25 classifiers and 5 feature generators. To accomplish the min-max training goal, five classifiers share each generator. The input data is mapped to five distinct distributions using five generators. Voting on the categorization findings of each hypothesis determines the human purpose category based on the highest probability. Effective TSL is trained using the mean predictions of the HEH as pseudo-labels. The efficient TSL comprises a single F_g and single B_c.

2.2.1. Training steps

In subsection 2.2, the theoretical foundations of loss functions were presented. How to integrating train the networks based on these functions is explained in the current section.

Phase 1. The F_{g} and $\mathcal{B}_{\mathrm{c}}$ are initially trained to reduce the training user classification error, according to the analysis presented in subsection 2.2. In order to increase the feature discrepancy, the feature generators are additionally optimized:

$$\min_{F_{\mathrm{g}i},\mathcal{B}_{c_k}} \ell_s - \xi_{F_{\mathrm{g}}} \ell_{fd}, i \in \left[1, n_{F_{\mathrm{g}}}\right], k \in \left[1, n_{F_{\mathrm{g}}} \cdot n_{\mathcal{B}_{\mathrm{c}}}\right] \quad (34)$$

where $\xi_{F_{\mathrm{g}}}$ is the loss function ℓ_{fd} constant weight.

Also, in order to maximize the $\mathcal{B}_{\mathrm{c}}\Delta\mathcal{B}_{\mathrm{c}}$-distance, the classifier discrepancy should be increased to determine the discrepancy upper bound. Additionally, to leverage the cluster assumption and enhance classifier confidence, it is imperative to minimize entropy loss. The feature generators F_{g_i} are frozen throughout this phase. The training of $\mathcal{B}_{\mathrm{c}_k}$ involves using the training user error, entropy, as well as classifier discrepancy loss.

$$\min_{\mathcal{B}_{\mathrm{c}_k}} \ell_S - \xi_{\mathcal{B}_{\mathrm{c}}} \ell_{upcd} + \xi_{eop} \ell_{eop}, k \in \left[1, n_{F_{\mathrm{g}}} \cdot n_{\mathcal{B}_{\mathrm{c}}}\right] \quad (35)$$

where ℓ_{upcd} and ℓ_s 's constant weights are represented by $\xi_{\mathcal{B}_{\mathrm{c}}}$ and ξ_{eop}, respectively.

By optimizing the classifiers, we can determine the maximum classifier discrepancy, thus providing an improved model. The training of feature generators focuses on minimizing the discrepancy upper bound, subsequently reducing the $\mathcal{B}_{\mathrm{c}}\Delta\mathcal{B}_{\mathrm{c}}$ - distance once the upper bound is established. Furthermore, the average hypothesis's error is minimized by optimizing the feature discrepancy. It is also important to minimize the entropy loss to separate the characteristics of various classes into distinct clusters. Classifiers $\mathcal{B}_{\mathrm{c}_k}$ are frozen in this stage. Also, F_{g_i} are trained to maximize feature discrepancy, decrease upper bound classifier discrepancy, and raise classification confidence by modifying the end-point users' domain to match the training user domain:

$$\min_{F_{\mathrm{g_{eop}}}} \ell_{\mathrm{upcd}} - \ell_{fd} + \xi_{\mathrm{eop}} \ell_{eop}, i \in \left[1, n_{F_{\mathrm{g}}}\right] \quad (36)$$

To reach a fixed epoch, these three processes are repeated numerous mini-batches at a time.

The pseudo labels for each end-point user of data may be obtained by the HEH model after training. Cross-entropy reduction between predictions of the proposed method (TSL and HEH) and pseudo-labels guides the training of the TSL architecture.

$$\ell_{ts} = \boldsymbol{E}_{T_x \sim D_T} \sum_{n=1}^{N} -P_n\left(y_{\mathrm{teacher}} \mid T_x\right) \log P_n\left(y_{\mathrm{student}} \mid T_x\right) \quad (37)$$

where y_{teacher} and y_{student} represent the teacher's (HEH) and student's predictions of T_x, respectively.

Table 2
Configuration of the proposed method on different datasets.

Dataset	DSADS and ENABL3S	Moon
#Generators	5	5
#Classifiers	25	25
Generator architecture	CNN	ANN
Global feature Size	256	32
$\xi_{F_{\mathrm{g}}}$	5	3
$\xi_{\mathcal{B}_{\mathrm{c}}}$	5	3
ξ_{eop}	0.01	1
Epochs	100	50
Optimizer	Adam	Adam
Learning rate	2×10^{-4}	1×10^{-3}
Batch size	256	200

2.2.2. Experimental Setup

In line with other studies [24], we began by carrying out a pre-experiment to categorize 2D interweaving moon spots. Fig. 3 displays the training user samples, which are a lower as well as an upper moon. Each of the two moons, designated 0 and 1, has 1600 samples based on the Gaussian noise N(0,0.06). The training user data is translated and rotated to create the end-point users' data. Fig. 3 displays the translations and the rotation angles. Fig. 3 illustrates the performance visualized by drawing the binary classifier's decision border. The network size was further reduced due to the dataset only containing two features. Table 2 provides the network design as well as hyper-parameters utilized on the moon experiment.

Two publicly available datasets are used in this article to assess how well the suggested strategies work. Northwestern University's Encyclopedia of Able-bodied Bilateral Lower Limb Locomotor Signals (ENABL3S) is one dataset [39]. Also, the Bilken University-provided Daily and Sports Activities Data Set (DSADS) is used to evaluate the proposed method [40]. The creators of these two datasets have previously segmented and filtered the signals; their publications [40, 39] provide the specific processing techniques used. Ten healthy users' signals were recorded from experiments for ENABL3S [39]. The locomotion modes that the users were requested to walk on varied terrains as well as inversions of standing (St), stair ascent (SA), ramp ascent (RA), level ground walking (LW), ramp descent (RD), as well as stair descent (SD). Filtered signals of joint angle, IMU, as well as EMG are included in ENABL3S [39]. A sliding window with a width of 300 ms divided the filtered signals into segments. Human aim is not always obvious, and it can be challenging to pinpoint exactly when it manifests. Thus, prior studies used gait events, such as heel-contact and toe-off, to determine the duration of transitioning between various locomotor modes [41, 11, 22]. The segmented signals

can be used to forecast the intention of movement because each segment selected 300 ms before the gait occurrences.

Next, the segmented samples were used to extract the low level features. The waveform length, mean absolute value, number of zero-crossing points, number of changing slope signs, as well as coefficients of a sixth-order autoregressive model are the ten superficial properties of EMG signals. From each channel of the joint angle and IMU data, six characteristics were extracted: the standard deviation, mean, minimum, maximum, the beginning value, as well as the end value. In this research, the features of the IMU and EMG are molded into a 33×12 picture with single channel.

Eight healthy participants were asked to complete 19 distinct tasks in DSADS, such as jumping, sitting, standing, running, as well as riding a bike [40]. Five 9-axis IMUs' worth of signals are included in DSADS [40]. A 5-second window was used to segment the recorded signals. The segmented signals' derived characteristics are identical to the IMU signals' in ENABL3S. A 45×6 single-channel picture is created by reshaping the extracted features. DSADS cannot be utilized to anticipate human intent since it lacks a transition between various motion types.

There are 9,000 and 22,000 signal segments in the DSADS and ENABL3S datasets, respectively. Each user's data were shuffled at random and split into a test set (30%) and a training set (70%). The absence of a validation set explained by the assumption that the end-point users dataset is unlabeled, and the inability to acquire a labeled end-point users validation set for the purpose of fine-tuning the neural network or figuring out the early-stop time. Each experiment included a leave-one-subject-out test, in which an end-point user was chosen and the remaining individuals served as training user. The labeled training user dataset as well as the unlabeled end-point user dataset were used to train the suggested HEH. To assess the effectiveness of user-agnostic adaptation, the HEH was assessed using the test set following a predetermined number of epochs of training. Once an experiment was completed, a new user was chosen as the test user for the following experiment, and so on until all users were transversed.

Experiments were conducted to compare the suggested HEH and TSL with various state-of-the-art methods, such as CNN (convolutional neural network), SVM (support vector machines), ANN (artificial neural networks), DFA [42], MCD [24], MMD [43], CORAL [44], and DANN [23]. Only the source training datasets were used for training and testing LDA, SVM, ANN, and CNN. This paper considers these four models as baseline models. The DANN, CNN, MMD, DFA, MCD, CORAL, HEH, as well as the proposed method utilized the same architecture for label classifier and feature generator. DANN contains a domain classifier. The training technique and loss functions of the benchmark method utilized in the previous study were the same as those of MCD, but the benchmark method's network size was substantially bigger.

3. Result and Discussion

In this section, we present our findings organized into subsections focusing on specific datasets, statistical test, and feature diversity. The subsections provide detailed analyses of the Moon, ENABL3S, and DSADS datasets, likely discussing patterns, trends, and unique characteristics observed within each dataset. Furthermore, we explore the statistical test and diversity of features across these datasets, examining their impact on the overall analysis and discussing implications for future research or applications.

3.1. Moon Dataset

It was discovered that the TSL and HEH, which consist of just one classifier and one feature generator, can effectively learn from HEH and outperform other approaches on moon dataset with various translations and rotation degrees. The suggested TSL and HEH may more precisely identify and illustrate in Fig 3 the decision boundary that categorizes the moon dataset. The proposed TSL and HEH outperform both of ANN and DANN with P-value<0.002 in terms of performance, with 97.9% accuracy that was achieved after 50 epochs of training. The P-value indicating the probability of the null hypothesis being true, is computed using One-way ANOVA with a post hoc test. If the P-value is less than 0.05, the difference is considered significant. In addition, the proposed method (TSL and HEH) outperforms MCD in terms of classification accuracy by 4.8%, the difference is not statistically significant (P-value = 0.07). Compared to the other three approaches, TSL and HEH's classification accuracy standard deviation is at least 1% lower, suggesting that TSL and HEH is a more stable approach.

The suggested TSL and HEH have steeper learning curves than the alternative approaches. After around thirty epochs of training, all networks reach a plateau. Compared to MCD, the suggested TSL and HEH's plateau is higher and more stable. One explanation might be because the HEH's various generators and classifiers reduce the variation in classification accuracy, allowing the TSL and HEH to learn from the reliable pseudo-labels provided by the HEH. All networks, with the exception of TSL and HEH, reach a peak accuracy before crossing the plateau, but after additional training, the accuracy declines. This event highlights the significance of the validation set in assisting with the early-stop time determination as well as the stability of the TSL and HEH. The network's performance following specific epochs might be significantly poorer than its maximum performance if the validation set is absent. As seen in 3, the suggested TSL and HEH reduces this issue and attains a greater stability.

3.2. ENABL3S Dataset

The suggested HEH outperformed all existing approaches on ENABL3S [39], predicting end-point users' locomotor intent more correctly (mean= 95.6%) and steadily due to standard deviation (std) equal to 1.6 as shown in Table 2. Nonetheless, the HEH inference time of up to 5.8 ms can be the reason for the identification findings' temporal

Table 3
Inference time and accuracy of categorizing and recognizing the intent locomotion modes for the training and end-point users of DSADS [40] as well as ENABL3S [39] and utilizing state-of-the-art approaches. The standard deviation is indicated by Std. The inference time shows how long it takes to identify each human intent online.

Dataset	ENABL3S					DSADS				
Users	Training		End-point			Training		End-point		
Models	Mean(%)	Std(%)	Mean(%)	Std(%)	Time(ms)	Mean(%)	Std(%)	Mean(%)	Std(%)	Time(ms)
SVM	89.7	1.3	78.6	6.8	0.2	96.6	1.0	81.8	6.6	0.2
LDA	92.2	0.2	85.3	3.9	0.2	98.4	0.1	92.2	3.3	0.2
CNN	96.9	0.3	88.5	5.4	1.0	99.1	0.4	90.2	6.1	1.0
ANN	93.8	0.4	87.2	3.4	0.3	98.2	0.5	83.7	9.5	0.3
MCD [24]	95.2	1.2	93.8	1.8	1.0	98.9	0.5	95.5	4.5	1.0
CORAL [44]	96.5	0.4	91.6	3.4	1.0	99.1	0.2	91.8	4.3	1.0
MMD [43]	96.3	0.6	92.9	2.2	1.0	99.2	0.2	95.6	3.3	1.0
DFA [42]	94.9	1.7	91.7	3.0	1.0	98.6	0.7	92.4	3.3	1.0
DANN [23]	96.4	0.3	88.7	4.8	1.0	99.3	0.2	91.3	5.2	1.0
HEH (Ours)	96.5	0.7	95.6	1.6	5.8	99.3	0.4	98.2	4.3	5.2
TSL & HEH (Ours)	**97.4**	0.7	**96.7**	1.6	**1.0**	**99.7**	0.4	**98.9**	4.1	**1.0**

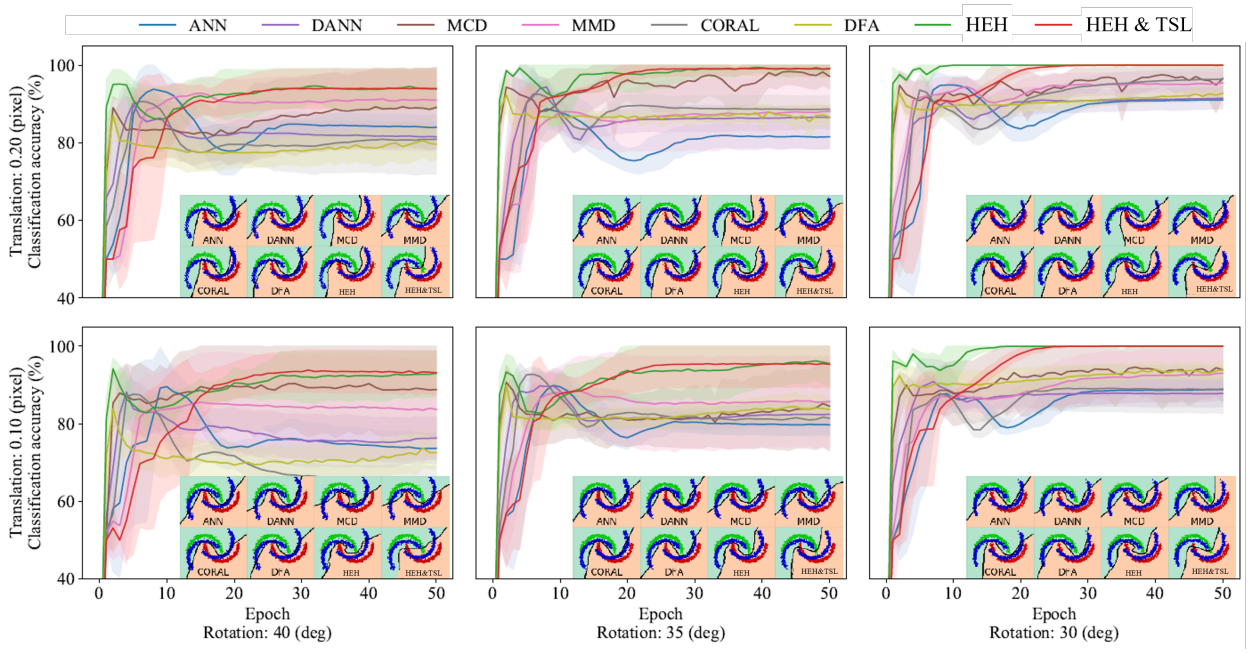

Figure 3: Decision boundaries and learning curves of several techniques used in the training phase to categorize. The training user data for two distinct classes are represented by the green and red dots. The desired data, shown by blue dots, should fall into two groups but isn't given a title. The average performance in five trials is shown by the line in term of accuracy, and the shaded area by the standard deviation.

delay. The inference time might be reduced by the TSL and HEH to 1 ms following knowledge distillation. The TSL and HEH's classification accuracy jumped to 96.4%, however the jump was not significant (P-value = 0.08). The improvement demonstrates the benefit of using an appropriate network design for assessing human locomotion intent, even though it is statistically significant. The suggested HEH further increases classification accuracy using the same classifier as well as generator architecture; this improvement is 1.8% more than with MCD, however the difference is not statistically notable (P-value =0.12). Additionally, the HEH method's performance is 2.3% greater than the benchmark methods, and this difference is statistically significant (P-value= 0.04). Even though TSL and HEH only use one classifier and one feature generator, it nevertheless outperforms MCD and the benchmark technique in terms of classification accuracy by 2.9% and 3.4%, respectively. Also, the difference is statistically significant (P-value=0.029). Thus, the performance of recognizing the locomotion intent

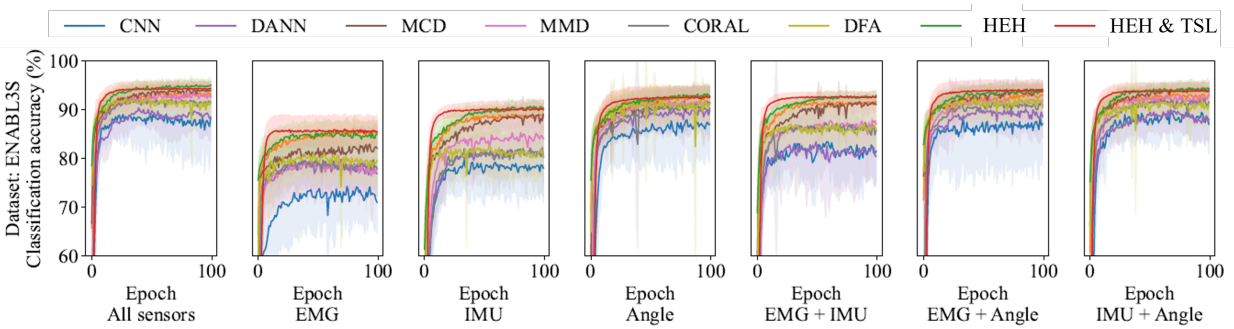

Figure 4: learning curves for several techniques used in the training process to categorize the end-point users set of the ENABL3S dataset. Ten separate leave-one-user-out trials were used to calculate the average classification accuracy. The shady area and the line show the standard deviation and mean of these results, respectively.

of end-point users may be improved by both refining the network design and mixing different hypotheses. Knowledge distillation has the potential to improve network efficiency while maintaining high performance, which is advantageous for real-time locomotion intent identification.

The proposed method (HEH and TSL), developed for predicting the locomotion intent of end-point users, exhibit significant improvement compared to non-adapted approach such as ANN, SVM, CNN, and LDA (P-value=0.002), as shown in Table 2. This suggests the effectiveness of the proposed approach in successfully transferring learned features from the training user domain to the end-point users' domain. The performance of end-point users using TSL, HEH, and MCD significantly surpasses that of DANN (P-value=0.004). This observation implies that the domain classifier might not operate effectively when domain biases are inconsistent across classes, and it faces difficulties in matching features belonging to each class between the two domains. Moreover, in comparison to non-adversarial domain adaptation methods such as CORAL and MMD, the proposed approach (TSL and HEH) along with MCD achieves higher classification accuracy (P-value=0.017). This underscores that matching features within each class yields better results than matching features globally.

Because the training user data is tagged, it makes sense that the training user intent recognition accuracy is better than the end-point users' result. The potential of the CNNs designed in this paper is validated by the significantly higher training user classification accuracy (P-value= 0.02) achieved with deep neural networks (e.g., MCD, DANN, CNN) compared with shallow classifiers (ANN, LDA, and SVM). The suggested convolutional neural network's reduced size and ability to calculate in parallel with a GPU have resulted in inference times for MCD, DANN, CNN, DFA, CORAL, MMD, and the proposed method (TSL and HEH) that are comparable to shallow classifiers.

The precision of anticipating the intention of end-point users to move varies depending on the type of sensor used. Utilizing all of the sensors results in a greater classification accuracy than utilizing only some of them, as Fig. 4 illustrates. Despite appearing to be the least accurate in identifying locomotor modes, EMG signals can nevertheless enhance classification accuracy if combined with additional sensors (such as angle and IMU sensors). Notably, the suggested method (TSL and HEH) outperforms alternative techniques for every possible combination of sensors, thus confirming the suggested TSL and HEH's efficacy.

While utilizing the validation data to finish the process of training early does not guarantee the highest accuracy, the overfitting issue appears to be minimal on ENABL3S. The suggested HEH approach starts to converge after 50 training epochs, exhibiting modest variance post-convergence. In contrast, other approaches display fewer stability during training, posing a risk of achieving lower accuracy after a specific number of epochs if ensemble diverse hypotheses are not incorporated.

3.3. DSADS Dataset

The suggested method (TSL and HEH) outperformed all existing approaches on DSADS [40] for diverse human activities like riding a bicycle, playing basketball, and jogging, predicting end-point users' locomotor intent more correctly (mean = 98.1%, std = 4.1%), as depicted in Fig. 5 and Table 3. This achievement is notably superior (P-value = 0.011) in comparison to the DFA methods employed in [42].

Remarkably, the MCD technique outperforms the DFA method in end-point user classification accuracy by 3.1%, and this increase is statistically significant (P-value = .037). This finding indicates that when the dataset is short, network design optimization is required to prevent overfitting. After training for around 20 epochs, as seen in Fig. 5, the DFA method technique initially reaches the maximum performance before beginning to overfit the training set. The ultimate performance of the DFA method is much less than the peak performance after 100 training epochs. Both MCD and the proposed method (TSL and HEH) become more stable during training as the neural networks' size is reduced, and they reach a jump after 50 training epochs. The proposed method (TSL and HEH) learning curves are more stable than MCD's, however, indicates that TSL and HEH improves the classification's stability.

Regarding DSADS, the end-point user categorization accuracy is not significantly impacted by the sensor location.

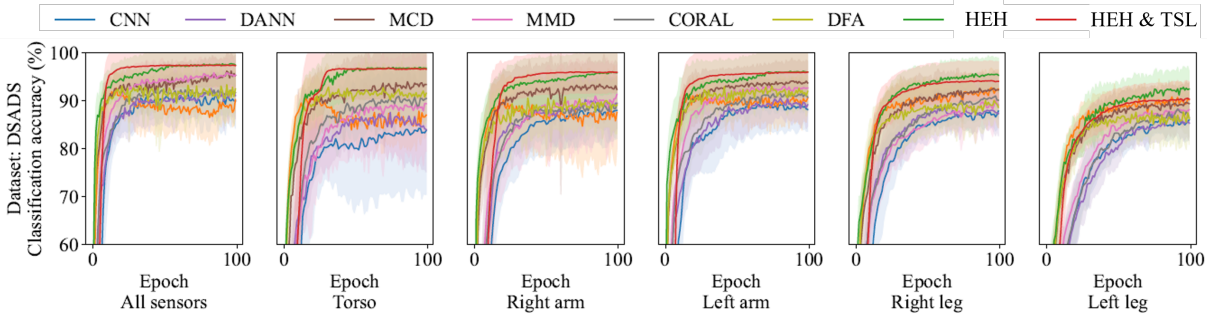

Figure 5: learning curves for several techniques used in the training process to categorize the DSADS dataset's end-point test set. The standard deviation and of mean the average classification accuracy across eight distinct leave-one-subject-out trials are shown by the shaded region as well as the line, respectively.

The individuals wore five IMUs on their various body parts: the torso (T), right arm (RA), left arm (LA), right leg (RL), and left leg (LL). The suggested TSL and HEH functions as well with a single sensor as it does with all the sensors combined, with the exception of the left leg's IMU. TSL and HEH performs better than any other approaches, regardless of the sensor used.

Surprisingly, LDA can still categorize end-point users individuals' locomotor modes with 92.2% accuracy. This accuracy is even better than that achieved by CNN, although the difference is not considered notably (P-value = 0.28). Although, this is an analytical technique, real-time computing may be done with great efficiency (inference time = 0.2 ms). Sometimes just training user data is reachable to train the classifier since the end-point users label as well as end-point users' data are unavailable. In these circumstances, LDA still appears to be a straightforward yet effective strategy for recognizing class in a small manifold space while ensuring a respectable degree of generalization. When utilizing the proposed method (TSL and HEH), provided the end-point users data are available, the accuracy achieved is substantially greater (P-value = 0.012) than with LDA, while the cost is still efficient at 1.0 ms.

3.4. Statistical Test

The results of one-way ANOVA with post hoc test comparing the proposed methods (TSL & HEH) with other methods on both the ENABL3S and DSADS datasets are presented in Table 4. The P-values indicate the significance of the differences in classification accuracy between the proposed methods and each of the other methods. The proposed methods show statistically significant differences compared to several baseline approaches on the ENABL3S dataset. Notably, when compared to SVM, LDA, CNN, and ANN, the proposed methods demonstrate highly significant differences ($P < 0.01$), suggesting superior performance in classifying locomotion intent for end-point users. However, the differences between the proposed methods and MCD, CORAL, MMD, and DFA are marginally significant ($0.01 \leq P < 0.05$), indicating a slightly weaker but still notable advantage.

Table 4
Comparison of one-way ANOVA with post hoc test results for the proposed methods (TSL & HEH) with other methods on ENABL3S and DSADS datasets.

Approach	P-value (ENABL3S)	P-value (DSADS)
SVM	0.002	0.011
LDA	0.002	0.012
CNN	0.002	0.021
ANN	0.002	0.003
MCD [24]	0.029	0.023
CORAL [44]	0.017	0.014
MMD [43]	0.017	0.026
DFA [42]	0.014	0.011
DANN [23]	0.004	0.006

On the DSADS dataset, the differences in classification accuracy between the proposed methods and the baseline approaches are generally less pronounced but still significant. The proposed methods outperform SVM, LDA, CNN, ANN, MCD, and DFA with statistically significant differences ($P < 0.05$), suggesting their effectiveness in predicting locomotion intent for diverse human activities. The differences between the proposed methods and CORAL, MMD, and DANN are marginally significant, indicating a moderate advantage of the proposed methods.

Finally, the ANOVA results underscore the effectiveness of the proposed methods (TSL & HEH) in improving classification accuracy for predicting human locomotion intent, particularly when compared to traditional and state-of-the-art approaches. These findings support the suitability of the proposed methods for real-world applications requiring accurate and user-agnostic adaptation of locomotion intent recognition systems.

3.5. Features diversity

This paper visualizes the concealed agnostic-feature adapted by various generators as well as the non-adapted features to additional confirmation whether the suggested method extracts varied features by t-SNE [45]. Every generator has the ability to enhance the balance between the training and end-point users' features, as seen in Fig. 6. The

training users' characteristics disperse deterrently from the end-point user features in the absence of adaptation. Following adaptation, the majority of the same class's training user and end-point user traits are found in the same area. Furthermore, the predicted feature distributions for various generators differ from one another, and every generator converts the feature into a distinct feature domain.

In order to categorize the locomotion of the end-point users and their desire to intent without labeling their data, this paper suggested the heterogeneous ensemble hypotheses and Teacher-Student-Learning. This paper sought to address the concerns about the absence of a validation set of the end-point users data, a network with large parameters may overfit the training user data as well as only a small subset of the features might be learned by a single generator. Therefore, multiple generators must be trained to capture all the desired features. By merging several classifiers and feature generators, improving loss functions, as well as light-weight the basic model, the suggested TSL and HEH achieved the aforementioned goals. Three distinct datasets were utilized to assess the proposed TSL and HEH, and it was contrasted with other techniques such as deep classifiers, shallow classifiers, and the state-of-the-art model. According to experimental results, TSL and HEH outperformed all other approaches with average classification accuracy of 98.9%, 96.7%, and 96.9% on the test set of datasets of the DSADS, ENABL3S, and moon. The suggested TSL and HEH method's viability has been confirmed by these findings. Furthermore, because it only used one classifier and one feature generator to learn features from the teacher network (HEH), the proposed method (TSL and HEH) maintained its efficiency (inference time = 1 ms). This supported the theory that a light-weight student model could acquire features from the pseudo labels of a teacher model.

These improvements were driven by a combination of factors, which are explored in detail below.

Initially, it is preferable to use two agents. As seen in Figs. 3, 5, and 4, the suggested HEH outperformed the MCD in end-point user classification accuracy by 5.0%, 2.9%, and 3.4%, respectively, while having identical feature generator and classifier network architectures. Theoretically, lowering the MCD loss functions might lower the end-point user error upper bound. Nevertheless, MCD might only discover a portion of the hidden characteristics and only uses one feature generator. Furthermore, using backpropagation to train feature generators and classifiers does not ensure that the global minimum of MCD loss functions will be discovered. A neural network may become stuck in the local minima for a number of reasons, including its huge parameter number and non-convex loss function. The limitations of multi-objective optimization are another factor. Eq. 20 has several elements, such as the training user error and the $\mathcal{B}_c\Delta\mathcal{B}_c$_distance between the training and end-point user domains. There may be a clash between these two goals. Obtaining the global optimum for each loss function may be less successful than optimizing numerous loss functions to strike a balance between them. As a result, even after training, the MCD can still make mistakes. Although HEH cannot completely resolve this issue, it can be somewhat resolved. Every hypothesis has the potential to be incorrect, but it is rare for many hypotheses to be incorrect at the same time. Reducing the classification error can be achieved by voting all of the hypothesis' findings. Additionally, the diverse-feature loss prevents the network from becoming stuck in the same local minima by forcing the many base learners to learn distinct parameters.

The proposed method's (TSL and HEH) feature generator and classifier have trainable parameters of around 1.63×10^6 and 4.21×10^4, respectively. The lack of a validation set led to a preset training epoch number as there was no basis for determining early stopping during training. The DFA method's end classification accuracy was much less accurate than its maximum performance. The training process curves for the MCD, HEH, as well as the proposed method stabilized when the network was lightened as shown in Fig. 4. The smaller feature generator is the single factor contributing to the steadier training process curves, given the loss functions of the DFA approach and MCD are identical. Notably, the ENABL3S dataset does not appear to overfit the DFA technique, suggesting that the learning curve may be stabilized by either expanding the dataset or shrinking the neural network size. In practice, obtaining a big data collection could be challenging, therefore choosing an appropriate network size is still crucial.

The class of the end-point users' data that is not labeled is difficult to guess. To infer their categories, it is therefore necessary to make use of some internal data features. Clustering can be used to identify groups of similar data points, providing insights into the underlying structure of the data. Using the cluster attribute for unsupervised user-agnostic adaptation makes sense. The cluster assumption is validated by the retrieved characteristics. The samples in the equivalent class are related to each other than the points in other classes (different colors), as Fig. 6 illustrates. various types of points have various clearances. Consequently, it makes sense to encourage the Classification line to shift far from the dense regions by using an entropy loss function. During the first training period, points from various courses may mix together. The loss entropy function can be detrimental in this situation. The suggested HEH method's feature generators, however, have a variety of loss functions, such as minimizing classifier discrepancy and training users' classification errors. Entropy loss has a lower weight than the two loss functions mentioned above. As a result, during the neural network's first training phase, the two loss functions mentioned above will predominate. The characteristics will become distinct clusters after the disparity between the classifier and the training users errors are reduced. The clearances between various feature clusters increase with adaptation, as seen in Fig. 6. The entropy loss will become more significant in training as the features are transformed into distinct clusters, hence reducing the goal classification error.

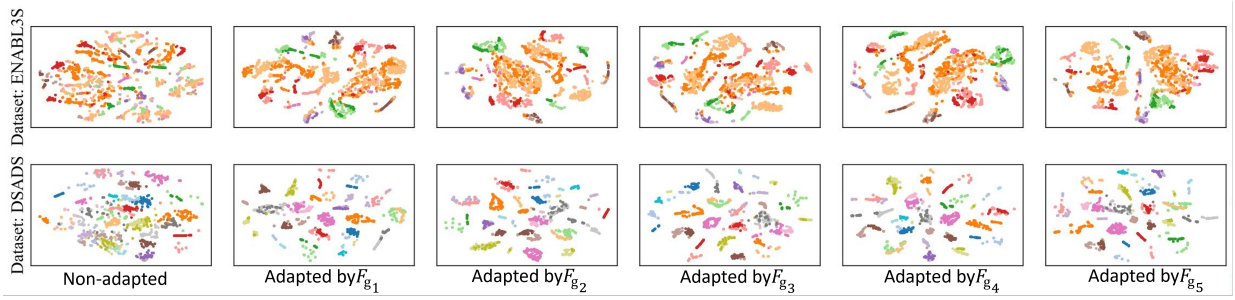

Figure 6: The display of the adapted features that were taken from five HEH feature generators and the non-adapted features projected using t-SNE. The end-point users' data is represented by the dark-colored dots, and the training users' data by the light-colored points. Points that share the same hue are members of the same class. Every feature is taken out of the training set. Features belonging to the same class concentrate in a comparable location after adaptation, and there is more clearing between various clusters than there was previously.

This paper has some shortcomings. First, training users' signals were not immediately recognized by the suggested HEH. Some shallow characteristics, including the standard deviation and the mean, were initially taken out of the training users signals in order to reduce the size of the training users' data and improve the algorithm's performance. Although these characteristics did decline the amount of data that needed to be processed, some crucial information can be lost. The deep CNN may also be used to extract additional features that are independent of the user for unsupervised user-agnostic adaption. Second, although labeling the end-point user data is not necessary for unsupervised User-Agnosticadaptation, it is still necessary to capture the end-point user data. The problem can grow even worse in particular cases where there is neither end-point users label nor end-point users' data available. Our only option in these circumstances is to train a stable classifier inside the training user domain. Figs. 5 ,4, and Table 3 demonstrate that even though the CNN was solely trained on training users' data, it was still able to achieve 88.5% and 90.2% mean performance on training users' data. Nevertheless, CNN's classification accuracy on the end-point users' data is significantly worse and fewer consistent than that of unsupervised domain adaptation techniques (e.g., the proposed method and MCD), and its generalization capacity remains inadequate. The bias between the training and end-point users domains is the cause of this constraint, and if the end-point users' data or end-point users' labels are unavailable, there is not much we can do. To further increase the performance on the end-point users' domain in this extreme case, it could be preferable to gather a small sample of the end-point users' data and create semi-supervised learning, data augmentation techniques, as well as few-shot learning.

4. Conclusions

Supervised learning (SL) requires a significant amount of data to accurately classify new signals. With humans, it is difficult to accurately capture and label data due to the complexity of the signals and the variation in people's behavior. Although, SL can correctly categorize the locomotion intent, but it is typically impractical for researchers and users alike to gather and label many signals in order to well learn the features of new user. The end-point users data label was not necessary for the unsupervised user-agnostic adaptation technique known as Heterogeneous Ensemble Hypotheses (HEH) and Teacher-Student-Learning (TSL), which was presented in this study. Following the labeled training users data and unlabeled end-point users data training, the proposed method (TSL and HEH) was able to classify the end-point users' modes and locomotor intent with high accuracy. The suggested TSL and HEH, in contrast to earlier techniques, could learn a variety of features and resolved the overfitting problem by adding the end-point users data cluster property, lightweight the base model, and training a variety of feature generators and classifiers. Two public datasets of human movement and a 2D moon dataset were used to empirically test and conceptually illustrate the performance of the TSL and HEH. According to experimental results, the HEH could categorize end-point users' data from the DSADS, ENABL3S, and moon data set with an average accuracy of 98.9%, 96.7%, and 96.9%, respectively. In contrast to the BM approach used in the earlier study, the suggested HEH reduced the inference time to 1 ms, harmonious its training process, and enhanced accuracy by 1.3% and 7.1%. The efficacy of the suggested TSL and HEH for unsupervised user-agnostic adaptation was confirmed by these outcomes. The suggested TSL and HEH may help wearable robots anticipate the end-point users' intended mobility without labeling the data, enhancing wearable robot intelligence and human-robot interaction.

In the future, we are considering the utilization and creation of datasets from different domains to investigate the generalization and reliability of our approach further. By diversifying the datasets used in our research, we aim to gain insights into how our methodology performs across various real-world situations and scenarios.

5. Disclosures

The authors declare no conflict of interest.

6. Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. 2222881. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] J. Zhang, P. Fiers, K. A. Witte, R. W. Jackson, K. L. Poggensee, C. G. Atkeson, S. H. Collins, Human-in-the-loop optimization of exoskeleton assistance during walking, Science 356 (6344) (2017) 1280–1284.

[2] Y. Ding, M. Kim, S. Kuindersma, C. J. Walsh, Human-in-the-loop optimization of hip assistance with a soft exosuit during walking, Science robotics 3 (15) (2018) eaar5438.

[3] T. R. Clites, M. J. Carty, J. B. Ullauri, M. E. Carney, L. M. Mooney, J.-F. Duval, S. S. Srinivasan, H. M. Herr, Proprioception from a neurally controlled lower-extremity prosthesis, Science Translational Medicine 10 (443) (2018) eaap8373.

[4] A. F. Azocar, L. M. Mooney, J.-F. Duval, A. M. Simon, L. J. Hargrove, E. J. Rouse, Design and clinical implementation of an open-source bionic leg, Nature Biomedical Engineering 4 (10) (2020) 941–953.

[5] M. Hao, J. Zhang, K. Chen, H. Asada, C. Fu, Supernumerary robotic limbs to assist human walking with load carriage, Journal of Mechanisms and Robotics 12 (6) (Dec. 2020).

[6] M. R. Tucker, J. Olivier, A. Pagel, H. Bleuler, M. Bouri, O. Lambercy, J. d. R. Millán, R. Riener, H. Vallery, R. Gassert, Control strategies for active lower extremity prosthetics and orthotics: A review, Journal of NeuroEngineering and Rehabilitation 12 (1) (2015) 1–29.

[7] J. Mendez, S. Hood, A. Gunnel, T. Lenzi, Powered knee and ankle prosthesis with indirect volitional swing control enables level-ground walking and crossing over obstacles, Science Robotics 5 (44) (2020) eaba6635.

[8] F. Sup, A. Bohara, M. Goldfarb, Design and control of a powered transfemoral prosthesis, The International Journal of Robotics Research 27 (2) (2008) 263–273.

[9] Y. Liu, X. Peng, Y. Tan, T. T. Oyemakinde, M. Wang, G. Li, X. Li, A novel unsupervised dynamic feature domain adaptation strategy for cross-individual myoelectric gesture recognition, Journal of Neural Engineering 20 (6) (2024) 066044.

[10] X. Wang, D. Ao, L. Li, Robust myoelectric pattern recognition methods for reducing users' calibration burden: challenges and future, Frontiers in Bioengineering and Biotechnology 12 (2024).

[11] D. Xu, Y. Feng, J. Mai, Q. Wang, Real-time on-board recognition of continuous locomotion modes for amputees with robotic transtibial prostheses, IEEE Transactions on Neural Systems and Rehabilitation Engineering 26 (10) (2018) 2015–2025.

[12] M. H. Zafar, E. F. Langås, F. Sanfilippo, Exploring the synergies between collaborative robotics, digital twins, augmentation, and industry 5.0 for smart manufacturing: A state-of-the-art review, Robotics and Computer-Integrated Manufacturing 89 (2024) 102769.

[13] B. Hu, E. Rouse, L. Hargrove, Fusion of bilateral lower-limb neuromechanical signals improves prediction of locomotor activities, Frontiers in Robotics and AI 5 (78) (2018) 1–16.

[14] X. Chen, K. Zhang, H. Liu, Y. Leng, C. Fu, A probability distribution model-based approach for foot placement prediction in the early swing phase with a wearable imu sensor, IEEE Transactions on Neural Systems and Rehabilitation Engineering (2021).

[15] B. Hu, A. M. Simon, L. Hargrove, Deep generative models with data augmentation to learn robust representations of movement intention for powered leg prostheses, IEEE Transactions on Medical Robotics and Bionics 1 (4) (2019) 267–278.

[16] J. Wu, C. Wu, F. Wang, L. Wang, Y. Wei, Improving visual grounding with multi-scale discrepancy information and centralized-transformer, Expert Systems with Applications 247 (2024) 123223.

[17] A. Windhausen, J. Heller, T. Hilken, D. Mahr, R. Di Palma, L. Quintens, Exploring the impact of augmented reality smart glasses on worker well-being in warehouse order picking, Computers in Human Behavior (2024) 108153.

[18] N. E. Krausz, T. Lenzi, L. J. Hargrove, Depth sensing for improved control of lower limb prostheses, IEEE Transactions on Biomedical Engineering 62 (11) (2015) 2576–2587.

[19] Y. Massalin, M. Abdrakhmanova, H. A. Varol, User-independent intent recognition for lower limb prostheses using depth sensing, IEEE Transactions on Biomedical Engineering 65 (8) (2018) 1759–1770.

[20] K. Zhang, C. Xiong, W. Zhang, H. Liu, D. Lai, Y. Rong, C. Fu, Environmental features recognition for lower limb prostheses toward predictive walking, IEEE Transactions on Neural Systems and Rehabilitation Engineering 27 (3) (2019) 465–476.

[21] B. Zhong, R. Luiz da Silva, M. Li, H. Huang, E. Lobaton, Environmental context prediction for lower limb prostheses with uncertainty quantification, IEEE Transactions on Automation Science and Engineering (2020) 1–13.

[22] K. Zhang, J. Luo, W. Xiao, W. Zhang, H. Liu, J. Zhu, Z. Lu, Y. Rong, C. W. de Silva, C. Fu, A subvision system for enhancing the environmental adaptability of the powered transfemoral prosthesis, IEEE Transactions on Cybernetics 51 (6) (2021) 3285–3297.

[23] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. March, V. Lempitsky, Domain-adversarial training of neural networks, Journal of Machine Learning Research 17 (59) (2016) 1–35.

[24] K. Saito, K. Watanabe, Y. Ushiku, T. Harada, Maximum classifier discrepancy for unsupervised domain adaptation, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Salt Lake City, UT, USA, 2018, pp. 3723–3732.

[25] J. Chen, J. Wang, C. W. de Silva, Mutual variational inference: An indirect variational inference approach for unsupervised domain adaptation, IEEE Transactions on Cybernetics (2021) 1–13.

[26] K. Zhang, J. Wang, C. W. de Silva, C. Fu, Unsupervised cross-subject adaptation for predicting human locomotion intent, IEEE Transactions on Neural Systems and Rehabilitation Engineering 28 (3) (2020) 646–657.

[27] Z. Allen-Zhu, Y. Li, Towards understanding ensemble, knowledge distillation and self-distillation in deep learning, arXiv:2012.09816 [cs, math, stat] (Jul. 2021).

[28] J. Zhao, X. Xie, X. Xu, S. Sun, Multi-view learning overview: Recent progress and new challenges, Information Fusion 38 (2017) 43–54.

[29] V. Bolón-Canedo, A. Alonso-Betanzos, Ensembles for feature selection: A review and future trends, Information Fusion 52 (2019) 1–12.

[30] F. Ali, S. El-Sappagh, S. M. R. Islam, D. Kwak, A. Ali, M. Imran, K.-S. Kwak, A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion, Information Fusion 63 (2020) 208–222.

[31] Z.-H. Zhou, Ensemble Learning, in: Z.-H. Zhou (Ed.), Machine Learning, Springer, Singapore, 2021, pp. 181–210.

[32] Q. Zhang, M. Lu, S. Yu, J. Xin, B. Chen, Discovering common information in multi-view data, Information Fusion (2024) 102400.

[33] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira, Analysis of representations for domain adaptation, in: B. Schölkopf, J. C. Platt, T. Hoffman (Eds.), Advances in Neural Information Processing Systems 19, MIT Press, 2007, pp. 137–144.

[34] S. Arena, E. Florian, F. Sgarbossa, E. Sølvsberg, I. Zennaro, A conceptual framework for machine learning algorithm selection for predictive maintenance, Engineering Applications of Artificial Intelligence 133 (2024) 108340.

[35] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, J. W. Vaughan, A theory of learning from different domains, Machine learning 79 (2010) 151–175.

[36] R. Shu, H. Bui, H. Narui, S. Ermon, A DIRT-T approach to unsupervised domain adaptation, 2018.

[37] Y. Grandvalet, Y. Bengio, Semi-supervised learning by entropy minimization, Advances in neural information processing systems 17

(2004) 529–536.
[38] Z.-H. Zhou, Ensemble Methods: Foundations and algorithms, CRC Press, 2012.
[39] B. Hu, E. Rouse, L. Hargrove, Benchmark datasets for bilateral lower-limb neuromechanical signals from wearable sensors during unassisted locomotion in able-bodied individuals, Frontiers in Robotics and AI 5 (14) (2018) 1–5.
[40] B. Barshan, M. C. Yüksek, Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units, Comput. J. 57 (2014) 1649–1667.
[41] H. Huang, F. Zhang, L. J. Hargrove, Z. Dou, D. R. Rogers, K. B. Englehart, Continuous locomotion-mode identification for prosthetic legs based on neuromuscular–mechanical fusion, IEEE Transactions on Biomedical Engineering 58 (10) (2011) 2867–2875.
[42] J. Wang, J. Chen, J. Lin, L. Sigal, C. W. de Silva, Discriminative feature alignment: Improving transferability of unsupervised domain adaptation by Gaussian-guided latent alignment, Pattern Recognition 116 (2021) 107943.
[43] M. Long, Z. CAO, J. Wang, P. S. Yu, Learning multiple tasks with multilinear relationship networks, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 1594–1603.
[44] B. Sun, K. Saenko, Deep CORAL: Correlation alignment for deep domain adaptation, in: G. Hua, H. Jégou (Eds.), Computer Vision – ECCV 2016 Workshops, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2016, pp. 443–450.
[45] L. van der Maaten, G. Hinton, Visualizing data using t-sne, Journal of Machine Learning Research 9 (Nov) (2008) 2579–2605.