Semantic-Discriminative Mixup for Generalizable Sensor-based Cross-domain Activity Recognition

WANG LU, Beijing Key Lab. of Mobile Computing and Pervasive Devices, Inst. of Computing Tech., CAS, China
JINDONG WANG, Microsoft Research Asia, China
YIQIANG CHEN*, Beijing Key Lab. of Mobile Computing and Pervasive Devices, Inst. of Computing Tech., CAS, China
SINNO JIALIN PAN, Nanyang Technological University, Singapore
CHUNYU HU, Qilu University of Technology (Shandong Academy of Sciences), China
XIN QIN, Beijing Key Lab. of Mobile Computing and Pervasive Devices, Inst. of Computing Tech., CAS, China

It is expensive and time-consuming to collect sufficient labeled data to build human activity recognition (HAR) models. Training on existing data often makes the model biased towards the distribution of the training data, thus the model might perform terribly on test data with different distributions. Although existing efforts on transfer learning and domain adaptation try to solve the above problem, they still need access to unlabeled data on the target domain, which may not be possible in real scenarios. Few works pay attention to training a model that can generalize well to unseen target domains for HAR. In this paper, we propose a novel method called Semantic-Discriminative Mixup (SDMix) for generalizable cross-domain HAR. Firstly, we introduce semantic-aware Mixup that considers the activity semantic ranges to overcome the semantic inconsistency brought by domain differences. Secondly, we introduce the large margin loss to enhance the discrimination of Mixup to prevent misclassification brought by noisy virtual labels. Comprehensive generalization experiments on five public datasets demonstrate that our SDMix substantially outperforms the state-of-the-art approaches with **6**% average accuracy improvement on cross-person, cross-dataset, and cross-position HAR.

CCS Concepts: • **Human-centered computing → Ubiquitous computing**; • **Computing methodologies → Transfer learning**.

Additional Key Words and Phrases: Human Activity Recognition, Transfer Learning, Domain Generalization

ACM Reference Format:
Wang Lu, Jindong Wang, Yiqiang Chen, Sinno Jialin Pan, Chunyu Hu, and Xin Qin. 2022. Semantic-Discriminative Mixup for Generalizable Sensor-based Cross-domain Activity Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 2, Article 65 (June 2022), 19 pages. https://doi.org/10.1145/3534589

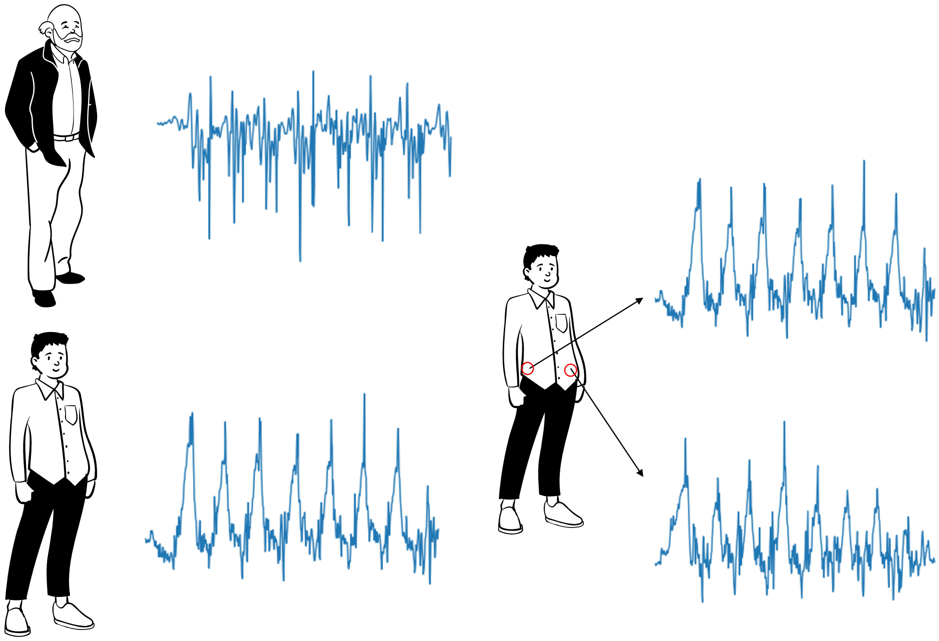
*Wang Lu, Yiqiang Chen, and Xin Qin are also with University of Chinese Academy of Sciences. Yiqiang Chen is also with Pengcheng Laboratory. Correspondence to: Jindong Wang and Yiqiang Chen.

Authors' addresses: Wang Lu, Beijing Key Lab. of Mobile Computing and Pervasive Devices, Inst. of Computing Tech., CAS, Beijing, China, luwang@ict.ac.cn; Jindong Wang, jindong.wang@microsoft.com, Microsoft Research Asia, Beijing, China; Yiqiang Chen, yqchen@ict.ac.cn, Beijing Key Lab. of Mobile Computing and Pervasive Devices, Inst. of Computing Tech., CAS, China; Sinno Jialin Pan, sinnopan@ntu.edu.sg, Nanyang Technological University, Singapore; Chunyu Hu, hcy@qlu.edu.cn, Qilu University of Technology (Shandong Academy of Sciences), China; Xin Qin, qinxin18b@ict.ac.cn, Beijing Key Lab. of Mobile Computing and Pervasive Devices, Inst. of Computing Tech., CAS, Beijing, China.

2474-9567/2022/6-ART65
https://doi.org/10.1145/3534589

1 INTRODUCTION

Sensor-based human activity recognition (HAR) aims to train machine learning models to recognize human activities based on different sensor data such as accelerometer and gyroscope. HAR has wide applications in many areas including senior care, rehabilitation, and personal fitness [10, 31]. Machine learning HAR models, especially deep learning-based models, often need large amounts of well-labeled data for training. However, it is expensive and time-consuming to collect or annotate massive labeled data in real applications. Even if there exist sufficient data for training, the performance of models may still deteriorate when applied to a new environment with different data distributions. For instance, a model trained on the activity data collected from the elderly cannot be directly used for adults because of the difference in their body size, activity patterns, and other characteristics.



(a) Different sensor readings on different subjects.

(b) Different sensor readings on different positions.

Fig. 1. Different sensor reading distributions on different subjects and different positions when walking.

As shown in Figure 1, the sensor data collected during walking follow different distributions. From Figure 1(a), we can see data of the adult is more stable than data of the elderly when walking. And Figure 1(b) demonstrates that there exist differences between data collected from two positions of one person at the same time. Therefore, directly applying a model trained from existing data to a new environment may suffer from dramatic deterioration caused by distribution shifts.

Domain adaptation (DA) [19, 36] is a popular technique to transfer the knowledge from a well-labeled source domain to a target domain while reducing their distribution discrepancies. Over the years, DA has been successfully applied to HAR by either aligning feature distributions via different distances [34] or adversarial training [5]. Nevertheless, DA needs access to the target data in training. This may be less realistic in modern applications where we often expect a trained HAR model to perform well in different situations such as different environments, people, and datasets. Recently, domain generalization (DG) [33] is gaining increasing attention. In contrast to DA, the goal of DG is to learn a model from one or several different but related domains that will generalize well on *unseen* target domains. For example, we expect a DG algorithm that trains on activity data from diverse ages to be able to generalize well to new persons of different ages. Existing literature on DG is mainly based on data augmentation [35], domain-invariant learning [8], or meta-learning [2]. Since HAR applications often require computation-restricted devices, we are interested in data augmentation-based DG methods for their simplicity and effectiveness.

Mixup [42] is a popular data augmentation method and has been applied to DG [35, 40]. Mixup enlarges the diversity of training data by interpolating between different domains. However, the generalization performance of Mixup-based DG can be undermined because of two challenges: *semantic inconsistency* and *discriminative slackness*. Firstly, the distributions of different activities are lying on different semantic spaces while direct Mixup will overlook their different semantic characteristics, resulting in semantic inconsistency between classes. Secondly, the interpolation of mixup can easily generate noisy data, which will decrease discriminations of the activity classifier.

In this paper, we propose **SDMix**, a *Semantic-Discriminative Mixup* approach for generalizable sensor-based HAR. To prevent semantic inconsistency, SDMix utilizes a semantic-aware Mixup technique to prevent classification boundaries from approaching the activity categories with large distribution shifts. We introduce activity semantic range for HAR that corresponds to the statistical values of activity shifts in the original space or feature space. Therefore, this technique can eliminate negative effects brought by different semantic ranges of different activities. Then, to handle discriminative slackness, SDMix introduces the large margin loss to replace the original cross-entropy loss. Through the large margin loss, each class is far away from each other, which means most virtual data points will locate between classes and induce less noise.

To sum up, our contributions are as follows:

(1) We explore Mixup for generalizable human activity recognition and empirically observe two limitations of Mixup: the semantic inconsistency and discriminative slackness, which undermine the performance of Mixup for domain generalization.
(2) We propose SDMix to handle the semantic inconsistency and discriminative slackness challenges in generalizable HAR by introducing semantic-aware Mixup and enhancing its discrimination using large margin loss. SDMix is conceptually simple and easy to implement.
(3) Extensive experiments demonstrate that our proposed SDMix significantly outperforms the state-of-the-art approaches (6% average accuracy improvement on cross-person, cross-dataset, and cross-position scenarios).

The remainder of this paper is organized as follows. Section 2 provides a brief review of the related work. Section 3 presents problem formulation and background. Then, Section 4 proposes SDMix. In Section 5, experimental evaluation and analysis on five public HAR datasets in three settings. Finally, we present the conclusions and the future work in Section 6.

2 RELATED WORK

2.1 Human Activity Recognition

Sensor-based HAR is receiving increasing attention over the years [31]. To alleviate the cost for data collection, there is growing interest in applying transfer learning and domain adaptation for HAR [5, 6, 34]. All existing domain adaptation and transfer learning-based HAR are assuming the availability of target domain data during training. Very little attention is paid to generalizable HAR, i.e., when the test distribution is unavailable for training. To our best knowledge, GILE [22] is the first work for generalizable HAR that disentangled domain-agnostic and domain-specific features in the latent feature space. However, GILE is based on a variational auto-encoder which is more complicated in real applications. Our method does not use generative models and it can perform well on simple CNNs based on Mixup data augmentation.

2.2 Domain Adaptation

Domain adaptation has developed for many years and detailed surveys can be found in [19, 36]. There is much prior work focusing on HAR with domain adaptation and a detailed survey can be found in [6]. For cross-domain activity recognition, Wang et al. [32] proposed a stratified transfer learning (STL) algorithm that learns the class-wise feature transformation. Wang et al. [34] designed a unified framework for source domain selection and activity transfer. Later, to select the most similar source domain to the target domain and perform accurately transfer activity, Qin et al. [24] proposed an adaptive spatial-temporal transfer learning (ASTTL) approach. Most recently, Lu et al. [15] tried to make full use of data structures and matched substructures of data via optimal transport. Although the above work can relieve distribution shifts, it cannot cope with situations when targets data is inaccessible.

2.3 Domain Generalization

Domain generalization (DG) tries to learn a model from one or several different but related domains that will generalize well on unseen testing domains. DG is different from leave-one-out-cross-validation (LOOCV) [37] since LOOCV is a model selection technique which selects models via parts of training data while DG is to enhance generalization on unseen targets. Existing domain generalization methods can be grouped into three categories [33]: data manipulation, representation learning, and learning strategy. Data manipulation contains two kinds of popular techniques: data augmentation [9, 18] which is mainly based on augmentation, randomization, transformation of input data, and data generation [14, 23] which generates diverse samples to help generalization. Representation learning also includes two kinds of representative techniques: domain-invariant representation learning [13, 25] which performs kernel, adversarial training, explicitly feature alignment between domains, or invariant risk minimization to learn domain-invariant representations and feature disentanglement [12, 41] which tries to disentangle the features into domain-shared or domain-specific parts for better generalization. Learning strategy mainly has three kinds of methods: ensemble learning [16] which relies on the power of ensemble to learn a unified and generalized predictive function, meta-learning [2, 30] which is based on the learning-to-learn mechanism to learn general knowledge by constructing meta-learning tasks to simulate domain shift, and gradient operation [11, 28] which tries to learn generalized representations by directly operating on gradients. For more details, please refer to [33]. Despite the popularity of DG in recent years, most of the methods are proposed for computer vision tasks that do not consider the application of HAR.

2.4 Mixup

Mixup [42] is a simple but effective technique for data augmentation. It extends the training distribution by incorporating the prior knowledge that linear interpolations of feature vectors should lead to linear interpolations of the associated targets. Although vanilla Mixup has shown its ability for domain generalization, many adapted versions are proposed for better performances. Recent work [38, 39] used the vanilla Mixup for domain adaptation without modifications. Wang et al. [35] mixed up samples across multiple source domains with two different sampling strategies to improve the generalization performance across different tasks. They perform well on computer vision while there is no Mixup method designed specifically for HAR.

3 PROBLEM FORMULATION AND PRELIMINARIES

We are given training data from several source domains. Our goal is to train a model that can perform well on an *unseen* target domain with these available source domain data. Since the target data have different distributions compared to the source domains, we expect that the trained model is generalizable. A common technique to enhance generalization capability is data augmentation and Mixup is a common way. Mixup may encounter two challenges: semantic inconsistency and discriminative slackness. To cope with these two challenges, SDMix is proposed which utilizes a semantic-aware Mixup technique to eliminate negative effects brought by different semantic ranges of different activities and introduces the large margin loss to reduce virtual noisy data.

3.1 Problem Formulation

Following the definition of generalizable cross-domain activity recognition from existing work [22], we are given S labeled source domains as the training dataset: $\mathcal{D}^{tr} = \{\mathcal{D}^i\}_{i=1}^S$. We use $P^i(\mathbf{x}, y)$ on $\mathcal{X} \times \mathcal{Y}$ to denote the joint distribution of one domain, where $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^m$ denotes the input and $y \in \mathcal{Y} = \{1, \cdots, C\}$ corresponds to output. m and C denote the input dimension and number of classes. Our goal is to learn a generalized model from $\mathcal{D}^{tr}$ to predict well on an unlabeled target domain, $\mathcal{D}^T$, which is unseen in training. In our problem, the training and test domains have the same input and output spaces but different distributions, i.e., $P^i(\mathbf{x}, y) \neq P^j(\mathbf{x}, y), \forall i, j \in \{1, 2, \cdots, S, T\}$. We aim to train a model h from $\mathcal{D}^{tr}$ to minimize the risk on $\mathcal{D}^T$: $\min_h \mathbb{E}_{(\mathbf{x},y)\sim P^T}[h(\mathbf{x}) \neq y]$.

3.2 Background

Mixup [42] is a popular approach for data augmentation. Given two random samples $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2)$, Mixup extends the training distribution by incorporating the prior knowledge: linear interpolations of feature vectors should lead to linear interpolations of the associated targets, formulated as [1]:

$$\begin{aligned} \tilde{\mathbf{x}} &= \lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2, \\ \tilde{y} &= \lambda y_1 + (1-\lambda) y_2, \end{aligned} \tag{1}$$

where $\lambda \sim Beta(\alpha, \alpha)$ and $\alpha \in (0, \infty)$ is a hyperparameter. As a powerful data augmentation technique, Mixup plays a vital role to enlarge the distribution diversity in existing domain generalization research [35, 40].

We are specifically interested in applying Mixup to generalizable HAR because of its simplicity. Unfortunately in HAR, Mixup faces the semantic inconsistency and discriminative slackness challenges that will dramatically undermine its performance.

4 SDMIX

In this paper, we propose **SDMix** (Semantic-Discriminative Mixup) to solve these two challenges in generalizable HAR. In the following, we will introduce the key components of SDMix.

4.1 Semantic-aware Mixup

In generalizable HAR, the distributions of sensor reading from two domains are different due to different lifestyles, habits, body shapes, and device positions. While Mixup tries to increase the diversity of distributions, we argue that the *activity semantic range* will largely impede Mixup.

Definition 1 (Activity semantic range). *The activity semantic range of a domain $\mathcal{D}$ refers to the maximum shift towards the activity center, denoted as:*

$$R_c = \max d(\mu_{\mathbf{x}|y=c}, \mathbf{x} : \mathbf{x} \in P(\mathbf{x}|y=c)), \tag{2}$$

where $\mu_{\mathbf{x}|y=c} = \mathbb{E}[\mathbf{x}|y=c]$ denotes the activity semantic center for class c and $d(\cdot,\cdot)$ is a certain distance function.

Definition 2 (Semantic inconsistency). *If data of two classes have different activity semantic ranges, i.e., $R_{c_1} \neq R_{c_2}$, then we say that there exists a semantic inconsistency between these two classes.*

[1] When performing Mixup, labels are often in the form of one-hot.

Different activities or even the same activity performed by different persons (domains) tend to have different activity semantic ranges. For instance, a person that walks in a parking lot may have larger activity semantic range than another person that walks on a treadmill with a similar speed in flat since walking in a parking lot is more unstable and contains more degree of freedom. Another example is that two persons both playing basketball may have different activity semantic ranges due to different habits and positions. The existing Mixup cannot consider the influence of activity semantic range from two different domains, resulting in *semantic inconsistency*. Activity semantic range varies between any two classes from any different domains. This situation cannot be avoided in both the feature space (Figure 7) and the original space. As shown in Figure 2(a), when the above situation occurs that two classes have different activity semantic ranges, the vanilla Mixup generates wrong classifications near the decision boundary. Note that vanilla Mixup utilizes the same λ for data and categories, which neglects the influence of activity semantic ranges.

In this paper, we propose *semantic-aware Mixup* to reduce the influence of different activity semantic ranges from different domains. The samples generated by semantic-aware Mixup can be formulated as:

$$\begin{aligned} \tilde{\mathbf{x}} &= \lambda \mathbf{x}_1^i + (1-\lambda)\mathbf{x}_2^j, \\ \tilde{y} &= t y_1^i + (1-t) y_2^j, \\ \lambda &\sim Beta(\alpha, \alpha), \end{aligned} \tag{3}$$

where $(\mathbf{x}_1^i, y_1^i)$, $(\mathbf{x}_2^j, y_2^j)$ denote samples from domain i and j. And t is the *activity semantic factor*, computed as:

$$t = \frac{\lambda \times R_{c_1}^i}{\lambda \times R_{c_1}^i + (1-\lambda) \times R_{c_2}^j}, \tag{4}$$

where $R_{c_1}^i$ and $R_{c_2}^j$ are the activity semantic ranges for the class c_1 and c_2 of domains $\mathcal{D}^i$ and $\mathcal{D}^j$, respectively. Note that c_1 and c_2 may or may not be the same, indicating that this works for both the same or different classes. We only apply t for y, which is equivalent to calibrating mixed $\tilde{y}$ with semantic range for a mixed $\tilde{\mathbf{x}}$.

What does Eq. (4) do? We illustrate its effects in Figure 2, where the larger R makes the corresponding y play a more important role. In Figure 2(a), vanilla Mixup pays no attention to the activity semantic range and treats $\mathbf{x}_i$ equally. Therefore, its decision boundary is biased towards the class with larger activity semantic range. While in Figure 2(c), the boundary is balanced by considering the effects of different activity semantic ranges. In Figure 2(b) and Figure 2(d), a toy example where simple models learned from simulation data following Gaussian distributions is to demonstrate that SDMix can learn better decision boundaries.

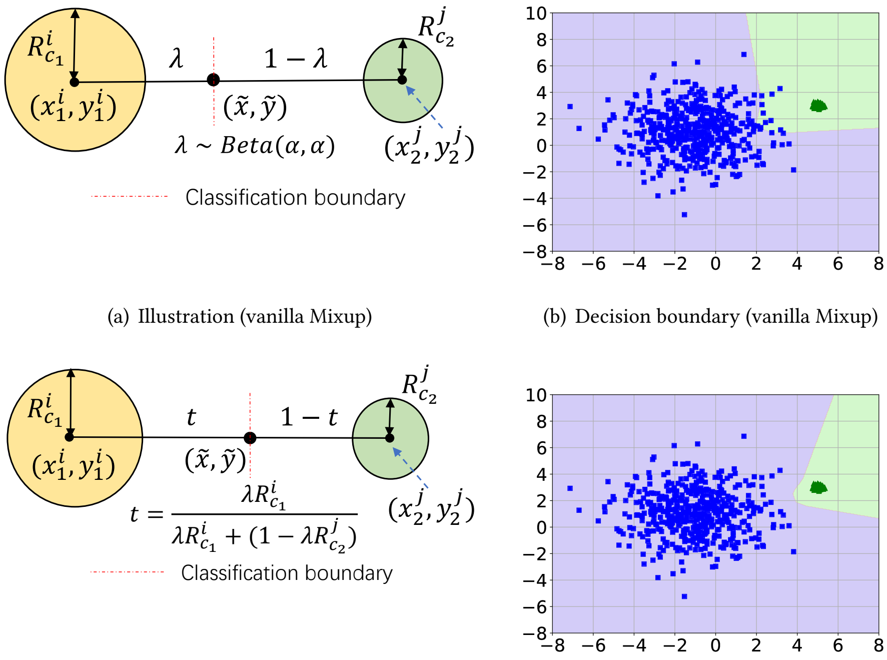
We present two alternatives to compute the value of activity semantic range R_c^i by either using the largest distance from samples to centroid, or the mean of all distances:

$$R_c^i = \max_{\mathbf{x} \in \mathcal{D}_c^i} d(\mathbf{x}, \mu_c^i), \tag{5a}$$

$$R_c^i = \frac{\sum_{\mathbf{x} \in \mathcal{D}_c^i} d(\mathbf{x}, \mu_c^i)}{|\mathcal{D}_c^i|}, \tag{5b}$$

where $\mathcal{D}_c^i$ represents the samples associated with class c from domain $\mathcal{D}^i$ and $|\cdot|$ is the cardinality. For computation, $d(\cdot,\cdot)$ can be ℓ_1, ℓ_2-distances or cosine distance. We choose different computing functions to achieve better performance. μ_c^i denotes the c-th class center of domain $\mathcal{D}^i$:

$$\mu_c^i = \frac{1}{|\mathcal{D}_c^i|} \sum_{\mathbf{x} \in \mathcal{D}_c^i} \mathbf{x}. \tag{6}$$



(a) Illustration (vanilla Mixup)

(b) Decision boundary (vanilla Mixup)

(c) Illustration (ours)

(d) Decision boundary (ours)

Fig. 2. The semantic inconsistency of Mixup. Different colors correspond to different classes. (a) Vanilla Mixup pays no attention to activity semantic range. (b) Decision boundary for vanilla Mixup leads to semantic inconsistency. (c) Our method considers the effects of activity semantic range. (d) Our semantic-aware Mixup mitigates such an issue.

4.2 Enhancing Discrimination of Mixup

Other than activity semantic range that describes the activity shift in HAR, we also notice another limitation of Mixup: the *virtual noisy activity instance* will influence the discrimination of Mixup.

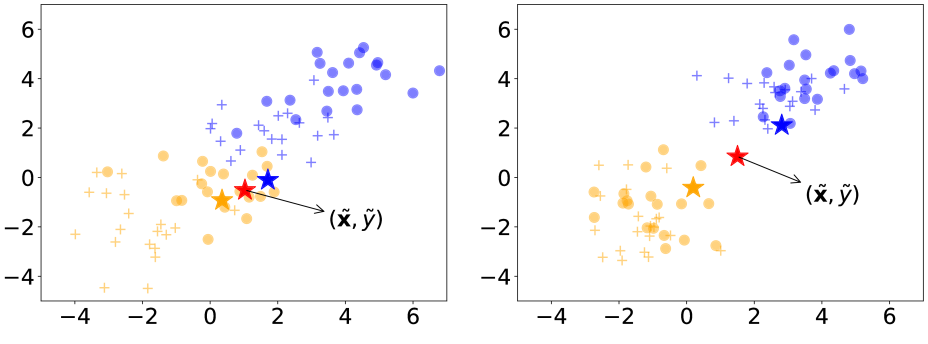
Definition 3 (Virtual noisy activity instance). *The interpolation* $(\tilde{\mathbf{x}}, \tilde{y})$ *generated by the Mixup of* $(\mathbf{x}_1, y_1)$ *and* $(\mathbf{x}_2, y_2)$ *is a virtual noisy activity instance, if*

$$||h(\tilde{\mathbf{x}}) - \tilde{y}||_1 \geq \epsilon, \tag{7}$$

where h *denotes the classification function and* $\epsilon > 0$ *is a threshold.*

Definition 4 (Discriminative slackness). *When trained the model with virtual noisy activity instances via Vanilla Mixup, the model may fail to discriminate different classes. We say that the model meets the discriminative slackness problem.*

Specially speaking, when $\mathbf{x}_1$ is close to the ideal decision boundary of class y_2 (or $\mathbf{x}_2$ to y_1), $\tilde{\mathbf{x}}$ could be extremely close to the decision boundary of y_1 or y_2 instead of their interpolation $\tilde{y}$. And from the point of location, $\tilde{\mathbf{x}}$ may belong to $\hat{y}$ which is different from y. For instance, due to different lifestyles, the running data of one person may have distributions similar to the walking data of another person. Thus, his data is easily misclassified. This causes *discriminative slackness* of Mixup, as shown in Figure 3(a).



(a) Discrimination (vanilla Mixup)

(b) Discrimination (ours)

Fig. 3. Discriminative slackness of Mixup. The generated data in (a) can easily be misclassified. Different colors correspond to different classes, and different shapes correspond to different domains. Stars in blue or yellow are selected data points for mixing. The red star is the virtual data point. *Best viewed in color.*

In this paper, we introduce *large margin loss* to enhance the discrimination of Mixup, which is formulated as:

$$\ell_y(\mathbf{x}_k, y_k) = \mathscr{A}_{c \neq y_k} \max\{0, \gamma + u_{h,\mathbf{x}_k,\{c,y_k\}} \mathrm{sign}(h_c(\mathbf{x}_k) - h_{y_k}(\mathbf{x}_k)\}, \tag{8}$$

where ℓ_y is the large margin loss, $\mathscr{A}$ is an aggregation operator for multi-class setting, and $\mathrm{sign}(\cdot)$ adjusts the polarity of the distance. $h_c : \mathcal{X} \longrightarrow \mathbb{R}$ or h_{y_k} is a function that generates a prediction score for classifying the input vector $\mathbf{x} \in \mathcal{X}$ to class c or y_k. γ is the distance to the boundary which we expect. $u_{h,\mathbf{x},\{c_1,c_2\}}$ is the distance of a point $\mathbf{x}$ to the decision boundary of class c_1 and c_2, which is computed as:

$$u_{h,\mathbf{x},\{c_1,c_2\}} = \min_{\delta} ||\delta||_p, \quad s.t.\ h_{c_1}(\mathbf{x} + \delta) = h_{c_2}(\mathbf{x} + \delta), \tag{9}$$

where $|| \cdot ||_p$ is l_p norm. As shown in [7], define $q = \frac{p}{p-1}$, then Eq. (8) can be computed as:

$$\ell_y(\mathbf{x}_k, y_k) = \mathscr{A}_{c \neq y_k} \max\left\{0, \gamma + \frac{h_c(\mathbf{x}_k) - h_{y_k}(\mathbf{x}_k)}{\left\|\nabla_{\mathbf{x}} h_c(\mathbf{x}_k) - \nabla_{\mathbf{x}} h_{y_k}(\mathbf{x}_k)\right\|_q}\right\}. \tag{10}$$

We illustrate the effect of large margin in Figure 3. In Figure 3(a), two classes with shape circle are far away from each other, but yellow circle data points are close to blue plus points. Direct adoption of the vanilla Mixup with the blue star and the yellow star will generate the red star, which is a virtual noisy data point (according to the location of data points). This means that a small class margin may induce terrible virtual labels when performing Mixup. In Figure 3(b), we can see large margin can alleviate this issue.

4.3 SDMix for HAR

Similar to the vanilla Mixup, our proposed SDMix remains conceptually simple and can be easily trained in an end-to-end manner. More importantly, our SDMix considers the semantics for different domains and also reduces noisy data when performing Mixup. Therefore, it makes the model perform better on unseen target data.

Figure 4 shows the network architecture of the proposed method. Mixed data goes through two blocks and each block contains one convolution layer and one pooling layer acting as feature extractor G_f. We view 1D time series as 2D data with the height being 1, following Fig. 2. in [31]. Then, it goes through one fully-connected layer which serves as the classification layer G_y. Semantics are fused via weighted label Mixup. In Figure 4, we only

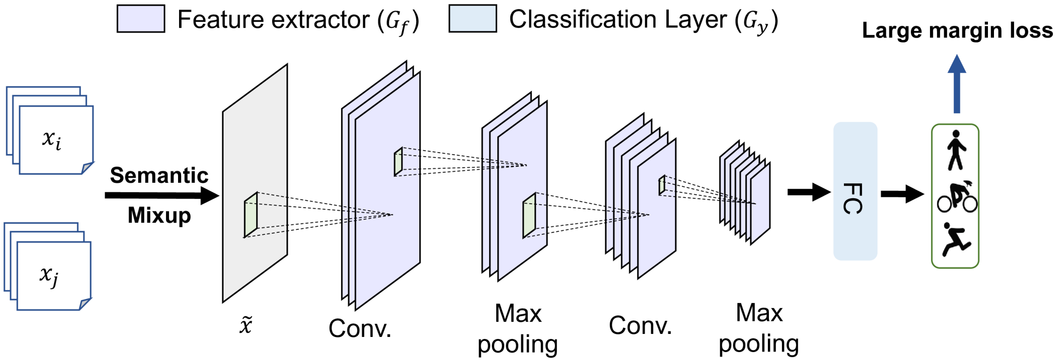


Fig. 4. The network architecture of the proposed method.

show conducting Mixup in the original space. We also can perform Mixup on the outputs of the feature extractor which can bring better results in some circumstances. In addition, we do not show BN layers in Figure 4.

The loss of SDMix can be expressed as:

$$\ell_{\text{SDMix}}(\mathbf{x}_1, y_1; \mathbf{x}_2, y_2) = \ell_y(\text{Mix}_\lambda(\mathbf{x}_1, \mathbf{x}_2), \text{Mix}_\lambda(y_1, y_2)), \tag{11}$$

where $\text{Mix}_\lambda(\cdot, \cdot)$ represents the semantic-aware mix operation computed by Eq. (3). z is the outputs of the feature extractor. ℓ_y is the large margin loss.

Therefore, our learning objective is formulated as:

$$\min \mathbb{E}_{P_1,P_2\sim P_{\mathcal{D}}}\mathbb{E}_{(\mathbf{x}_1,y_1)\sim P_1,(\mathbf{x}_2,y_2)\sim P_2}\mathbb{E}_{\lambda\sim Beta(\alpha,\alpha)}\left[l_{\text{SDMix}}(\mathbf{x}_1, y_1; \mathbf{x}_2, y_2)\right], \tag{12}$$

where $P_{\mathcal{D}}$ is the distribution of domains and P_1, P_2 are distributions of data in selected domains. Note that such an objective is optimized in a mini-batch style in deep networks and the details will be introduced in experiments. Specifically, we choose two source domains randomly each time and choose two samples from these two selected source domains respectively to perform semantic-aware Mixup. Semantic factors are updated every several iterations. For fairness (the same number of optimization iterations), the model only utilizes interpolated examples and does not utilize the original training examples. We believe that our method can achieve better performance with both original data and virtual data generated via mixing.

5 EXPERIMENTS

We evaluate the proposed method by constructing different generalization scenarios on five public datasets.

5.1 Datasets

The statistical information of the five public datasets is shown in Table 1, which includes sensor types, sampling frequency, and activity classes. Now, we briefly introduce them in the following.

The UniMib SHAR dataset (**SHAR**) [17] records 9 types of activities of daily living and 8 types of falls. 30 subjects (aged from 18 to 60) conduct the 17 fine-grained activities with data collected by an acceleration sensor embedded in an Android phone. 17 activities include 9 types of activities of daily living (StandingUpFL, LyingDownFS, StandingUpFS, Running, SittingDown, GoingDownS, GoingUpS, Walking, Null) and 8 types of falls (FallingBackSC, FallingBack, FallingWithPS, FallingForw, FallingLeft, FallingRight, HittingObstacle, Syncope).

UCI daily and sports dataset (**DSADS**) [3] consists of 19 activities collected from 8 subjects wearing body-worn sensors on 5 body parts. 19 activities include sitting, standing, lying on back and on right side, ascending and descending stairs, standing in an elevator still, moving around in an elevator, walking in a parking lot, walking on a treadmill with a speed of 4 km/h, running on a treadmill with a speed of 8 km/h, exercising on a stepper, exercising on a cross trainer, cycling on an exercise bike in horizontal and vertical positions, rowing, jumping, and playing basketball. Each subject wears three sensors: accelerometer, gyroscope, and magnetometer.

PAMAP2 physical activity monitoring dataset (**PAMAP2**) [26] contains data of 18 different physical activities, performed by 9 subjects wearing 3 sensors. 18 activities include lying, sitting, standing, walking, running, cycling, Nordic walking, watching TV, computer work, car driving, ascending stairs, descending stairs, vacuum cleaning, ironing, folding laundry, house cleaning, playing soccer, rope jumping, and other (transient activities). The sampling frequency is 100Hz and the data dimension is 27 (3 inertial measurement units).

USC-SIPI human activity dataset (**USC-HAD**) [43] is composed of 14 subjects (7 male, 7 female, aged from 21 to 49) executing 12 activities with a sensor tied on the front right hip. The data dimension is 6 and the sample rate is 100Hz. 12 activities include Walking Forward, Walking Left, Walking Right, Walking Upstairs, Walking Downstairs, Running Forward, Jumping Up, Sitting, Standing, Sleeping, Elevator Up, and Elevator Down.

Human activity recognition using smartphones dataset (**UCI-HAR**) [1] is collected by 30 subjects performing 6 daily living activities with a waist-mounted smartphone. 6 activities include walking, sitting, lying, standing, walking upstairs, walking downstairs. The data dimension is 6 and the sample rate is 50Hz.

Table 1. Statistical information of five datasets.

Dataset	#Subject	#Sensor	#Class	#Sample
SHAR	30	1	17	236,919
DSADS	8	3	19	1,140,000
UCI-HAR	30	6	6	1,310,000
PAMAP2	9	3	18	3,850,505
USC-HAD	14	2	12	5,441,000

5.2 Experimental Setup

We construct three types of experiments to thoroughly evaluate all methods for generalizable HAR: (1) Cross-person HAR similar to existing research [22], (2) Cross-dataset HAR, which is our construction that is more challenging than cross-person HAR since cross-dataset setting implies that not only the persons are different, but also the sensor types and positions, and (3) Cross-position HAR, which utilizes data from different positions with same persons. In all three settings, we adopt the sliding window technique [4] with 50% overlap to construct training samples following common practice in HAR [31]. The inputs should be cut into individual inputs according to the sampling rate and each new input serve as one instance.

We give a more detailed introduction of three settings in the following.

- **Cross-Person**: We conduct experiments on SHAR, DSADS, PAMAP2, and USC-HAD for this setting. We divide each dataset into 4 domains, e.g., 8 people are evenly divided into 4 domains where each domain consists of data from 2 people and there is no overlap between domains. For SHAR, we follow GILE to choose four persons. For the other datasets, we divide all persons into 4 groups. For PAMAP2, we choose 12 classes to ensure that each class has a certain number of data.
- **Cross-Dataset**: We conduct experiments on DSADS, PAMAP2, USC-HAD, and UCI-HAR for this setting. We do not split domains; instead, each dataset can be treated as one domain and we take the common classes from all datasets. To perform the Cross-Dataset setting, we first need to unify $\mathcal{X}$ and $\mathcal{Y}$. For $\mathcal{X}$, two sensors that belong to the almost same position are selected and each sample contains six dimensions. Then,

data is down-sampled and the sampling rate is 25Hz. For $\mathcal{Y}$, six common classes are selected, including walking, walking upstairs, walking downstairs, sitting, standing, and lying.

- **Cross-Position**: We conduct experiments on DSADS for this setting. 3 sensors are located in five parts of a body in DSADS. Each body part corresponds to a different domain. Each sample contains three sensors with nine dimensions.

Table 2. Detailed information on cross-person, cross-dataset, and cross-position experiments.

Cross-Person					
Dataset	#Domain	#Sensor	#Class	#Domain Sample	#Total
SHAR	4	1	17	(57,984;88,033;45,904;44,998)	236,919
DSADS	4	3	19	(285,000)×4	1,140,000
PAMAP2	4	3	12	(592,600; 622,200; 620,000; 623,400)	2,458,200
USC-HAD	4	2	12	(1,401,400;1,478,000;1,522,800;1,038,800)	5,441,000
Cross-Dataset					
Dataset	#Domain	#Sensor	#Class	#Domain Sample	#Total
	4	2	6	(672,000;810,550;514,950;470,850)	2,468,350
Cross-Position					
Dataset	#Domain	#Sensor	#Class	#Domain Sample	#Total
DSADS	5	3	19	(1,140,000)*5	5,700,000

Table 2 shows the information on cross-person, cross-dataset and cross-position scenarios. It is worth noting that since SHAR dataset contains much fewer samples than others, we replace it with UCI-HAR dataset [1] for cross-dataset experiments. The reason we did not run cross-person experiments on UCI-HAR is that the results on this dataset are all near 100%. In total, we constructed 16, 4, and 5 tasks for cross-person, cross-dataset, and cross-position HAR, respectively. We use 0, 1, 2, 3 to denote the four divided domains for cross-person experiments. We use 0, 1, 2, 3, 4 to denote the five divided domains for cross-position experiments.

Baselines. The latest method for generalizable HAR is GILE (Generalizable Independent Latent Excitation) [22]. Since there are no other methods designed for HAR except GILE, we turn to comparing with several popular domain generalization methods. DeepAll combines all source data together and trains a model according to ERM. DANN [8] and Coral [29] are two traditional DG methods, and we extend them to fit the domain generalization setting. Mixup [42], GroupDRO [27], RSC [11], and ANDMask [20] are four state-of-the-art universal domain generalization methods which can be directly used for HAR. Except GILE, we reproduced all other methods with the same network architecture in Pytorch [21] for fairness. GILE is based on VAE, and we directly use their released code. Table 3 shows information on the input size and kernel size of the models in all settings and the final dimension of input size represents the window size. We do not compare other Mixup-based DG methods since they are built for computer vision tasks that are not applicable to HAR problems.

To obtain the final model, we split data of the source domains into the training splits and validation splits. The training splits are utilized for training the model while the validation splits are utilized to guide the selection of the model. For all benchmarks, we select the best model via validation accuracy. In practice, we leave 80% of source domain data as training splits while the rest data are for validation. For testing, we evaluate the selected models on all data of the target domain. For our architecture, the model contains two blocks, and each has one convolution layer, one pool layer, and one batch normalization layer. A single fully-connected layer serves as

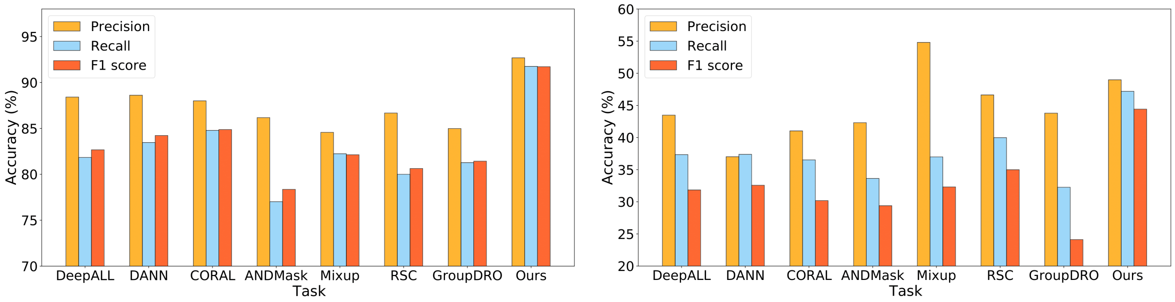
Table 3. Information on the architectures of the models.

Setting	Dataset	Input	Kernel Size
Cross-Person	SHAR	(3,1,151)	(1,9)
	DSADS	(45,1,125)	(1,9)
	PAMAP2	(27,1,200)	(1,9)
	USC-HAD	(6,1,200)	(1,6)
Cross-position	-	(9,1,125)	(1,9)
Cross-dataset	-	(6,1,50)	(1,6)

the classifier. In each step, each domain selects 32 samples. The maximum training epoch is set to 150. For our method, we search best hyperparameters within $[0.1, 0.2, 0.5, 1, 10]$ for α, $[1, 2, 5]$ for top c in large margin loss, and $[10, 100, 10000, 100000]$ for γ. For all methods except GILE[2], the Adam optimizer with a learning rate 10^{-2} and weight decay 5×10^{-4} is used. We use outputs of the feature extractor to compute radii and utilize different computing ways for different datasets. For fair study, we extensively tune hyperparameters for each method and report their average results of three trials. Code for SDMix will be available at http://transferlearning.xyz.

5.3 Experimental Results

The results of cross-person, cross-dataset, and cross-position HAR are shown in Table 4, 5, and 6 respectively. On average, our proposed SDMix substantially outperforms the second-best methods: **6.0%** for cross-person, **10.33%** for cross-dataset HAR, and **2.34%** for cross-position HAR. Moreover, from Figure 5, we can see that our method also achieves the best F1 score compared with other state-of-the-art methods whether in a balanced or unbalanced situation. These indicate that our method is effective for generalizable cross-domain HAR applications.



(a) Cross-person: nearly balanced classification

(b) Cross-dataset: imbalanced classification

Fig. 5. Precision, recall, and F1 score.Fig. 5(a) contains results from the first task in DSADS where data of the first two persons are used for testing and data of the rest persons are used for training. Each class has an equal number of instances (120). In Fig. 5(b), data in USC-HAD are used for testing while data in other datasets are used for training. Classes in USC-HAD have 2545, 2290, 3608, 2048, 1904, 3744 instances respectively.

We observe more insightful conclusions. (1) *When are Mixup-based methods most effective?* When the dataset volume is rather small with more classes, i.e., on difficult tasks. This can be verified on SHAR and DSADS datasets that have fewer samples but more classes. In these datasets, Mixup-based methods dramatically increase the

[2] The source code for GILE is available at https://github.com/Hangwei12358/cross-person-HAR

Table 4. Classification accuracy (± std. error) for *cross-person* HAR. The **bold** and underline items denote the best and second-best results, respectively.

	Src	Tar	DeepAll	DANN	CORAL	ANDMask	GroupDRO	RSC	Mixup	GILE	SDMix
SHAR	1,2,3	0	57.98±5.90	55.99±1.82	57.90±1.65	58.33±5.21	54.60±1.74	57.81±3.12	58.68±1.91	49.57±3.57	**65.80**±0.70
	0,2,3	1	57.29±3.26	50.20±3.54	**59.92**±1.60	48.89±2.75	58.72±5.20	55.06±6.17	55.69±3.89	48.02±4.02	59.86±2.40
	0,1,3	2	69.96±2.20	63.92±3.72	69.41±1.65	66.34±2.85	68.09±2.63	67.87±2.08	69.96±4.17	65.44±5.44	**74.45**±0.77
	0,1,2	3	40.16±1.23	42.06±2.46	40.60±0.33	42.39±3.13	42.17±0.56	40.27±1.34	41.83±1.23	39.27±3.27	**47.99**±0.67
	AVG	-	56.35±1.78	53.04±2.16	56.96±0.57	53.99±1.93	55.89±1.33	55.25±3.17	56.53±2.21	50.54±3.04	**62.03**±0.47
DSADS	1,2,3	0	83.26±4.22	88.09±4.84	90.51±2.00	85.17±1.05	91.77±1.16	84.21±1.93	88.08±2.12	79.67±1.67	**94.20**±1.04
	0,2,3	1	77.79±0.95	79.37±2.57	83.58±5.20	77.73±1.94	84.30±1.58	78.79±2.47	80.95±1.65	75.00±0.00	**91.07**±0.54
	0,1,3	2	84.68±3.01	82.48±3.14	83.04±4.84	83.32±5.38	82.06±8.51	81.48±4.46	88.00±2.52	77.00±1.00	**93.61**±0.80
	0,1,2	3	74.74±3.78	76.05±6.18	75.45±1.37	78.17±3.87	78.48±0.23	77.12±1.94	84.20±1.96	67.00±1.00	**87.00**±1.39
	AVG	-	80.12±1.08	81.50±2.84	83.15±1.32	81.10±3.03	84.15±2.15	80.40±1.88	85.31±1.76	74.65±0.15	**91.47**±0.27
PAMAP2	1,2,3	0	89.02±0.60	85.78±3.60	82.82±6.85	87.54±0.80	83.42±2.08	87.14±2.29	80.24±10.41	83.33±0.33	**89.14**±0.51
	0,2,3	1	75.80±1.84	75.86±1.61	77.43±1.44	77.97±1.53	74.95±2.37	77.31±5.95	78.77±3.81	68.67±0.67	**84.99**±0.29
	0,1,3	2	51.83±6.77	49.47±10.89	48.14±7.20	45.36±7.94	50.59±8.20	55.11±3.40	55.53±4.37	44.00±2.00	**63.11**±2.27
	0,1,2	3	82.98±3.06	84.64±3.41	84.47±3.62	84.01±3.45	82.66±1.97	83.64±5.30	86.63±0.68	76.67±0.67	**88.77**±0.22
	AVG	-	74.91±1.56	73.94±2.44	73.21±3.18	73.72±1.12	72.90±2.15	75.80±2.67	75.29±3.28	68.25±1.00	**81.50**±0.53
USC-HAD	1,2,3	0	79.90±3.80	80.54±1.39	79.02±3.11	79.46±0.94	81.57±1.45	80.05±2.57	79.70±2.66	78.67±0.67	**84.09**±0.66
	0,2,3	1	59.04±2.50	61.36±3.48	59.87±3.89	60.01±4.69	58.96±3.45	57.04±2.60	57.43±4.14	**63.00**±1.00	62.86±0.74
	0,1,3	2	72.57±1.27	75.90±0.45	74.41±1.64	73.74±1.19	71.59±1.61	74.25±0.86	73.83±1.19	**77.00**±1.00	76.46±0.97
	0,1,2	3	67.15±1.29	69.33±4.06	60.42±6.70	65.58±0.54	59.47±2.44	66.99±1.86	59.91±2.98	61.67±0.67	**71.90**±0.05
	AVG	-	69.67±1.38	71.78±1.97	68.43±1.81	69.70±1.02	67.90±0.51	69.58±0.19	67.72±1.60	70.08±0.08	**73.83**±0.45
	AVG all	-	70.26	70.07	70.44	69.63	70.21	70.26	71.21	65.88	**77.21**

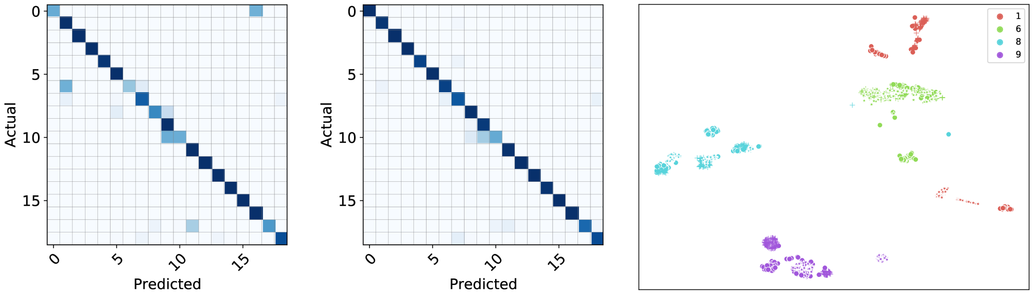
Table 5. Classification accuracy for *cross-dataset* HAR. The **bold** and underline items are the best and second-best results.

Source	Target	DeepAll	DANN	CORAL	ANDMask	GroupDRO	RSC	Mixup	SDMix
DSADS, USC, PAMAP2	UCI-HAR	46.06	39.10	44.44	43.22	33.20	45.28	40.24	**46.41**
USC, UCI-HAR, PAMAP2	DSADS	29.73	39.46	26.35	41.66	51.41	33.10	37.35	**52.66**
DSADS, USC, UCI-HAR	PAMAP2	43.84	36.61	32.93	40.17	33.80	45.94	23.12	**53.65**
DSADS, UCI-HAR, PAMAP2	USC	45.33	41.82	29.58	33.83	36.74	39.70	47.39	**53.54**
AVG	-	41.24	39.25	33.32	39.72	38.79	41.01	37.03	**51.57**

Table 6. Classification accuracy for *cross-position* HAR. The **bold** and underline items are the best and second-best results.

Source	Target	DeepALL	DANN	CORAL	ANDMask	GroupDRO	RSC	Mixup	SDMix
1,2,3,4	0	41.52	45.45	33.22	47.51	27.12	46.56	**48.77**	47.50
0,2,3,4	1	26.73	25.36	25.18	31.06	26.66	27.37	34.19	**36.10**
0,1,3,4	2	35.81	38.06	25.81	39.17	24.34	35.93	37.49	**42.53**
0,1,2,4	3	21.45	28.89	22.32	30.22	18.39	27.04	29.50	**34.52**
0,1,2,3	4	27.28	25.05	20.64	29.90	24.82	29.82	29.95	**30.93**
AVG	-	30.56	32.56	25.43	35.57	24.27	33.34	35.98	**38.32**

diversity of data distributions that benefits generalization. On the larger PAMAP2 and USC-HAD datasets that involve more diversity, the vanilla Mixup only slightly surpasses DeepAll or even worse. (2) *When is SDMix less effective?* When the training domains are much larger than the test domains, i.e., the training domains are already containing diverse[3] data, where SDMix may bring smaller improvements. In our statistics, source domains $1 \sim 3$ in PAMAP2 have more samples that make it enough to learn models directly using DeepAll. The same goes for cross-dataset HAR when UCIHAR is the target. In addition, when activity semantic range is hard to compute, SDMix only slightly surpasses other methods or even worse. For example, there exist little information and many categories in the cross-position setting, which makes it hard to estimate activity semantic range. SDMix only has a slight improvement compared with other methods and even performs worse on the first task. (3) *What influences stability in HAR generalization?* Algorithms, random states, data splits, and so on. Whichever algorithm is used, there exist fluctuations among the results of three trials. Even with DeepAll, results are also different. From Table 4, we can see that our method achieves the smallest STD compared to other state-of-the-art methods except GILE[4]. The results demonstrate that our method is still reliable and stable with three trials compared to other state-of-the-art methods. (4) *What influences generalization in HAR?* Dataset quantity, diversity, and the cross-domain distribution discrepancy. There is no doubt that the generalization ability of a model will increase when the dataset becomes larger and diverse (rf. Table 4). More importantly, we see a significant performance drop from cross-person to cross-position HAR, indicating the importance of cross-domain discrepancies [5]. The cross-dataset and position scenario is more challenging than cross-person since the sensor devices, positions, subjects are all different. In this scenario, our SDMix achieves the best performance by harnessing their diversity. Note that there is still room for improvement in this setting.



(a) Vanilla Mixup (b) Our method (c) Visualization

Fig. 6. Confusion matrices and Visualization on DSADS dataset for target 0 in the Cross-Person setting. Fig. 6(a) and Fig. 6(b) are confusion matrices of vanilla Mixup and our method. Deeper colors denote larger values. Fig. 6(c) is the visualization of the t-SNE embeddings with features learned by a simple model. Each class is denoted by a color and each shape corresponds to a domain. *Best viewed in color and zoom in.*

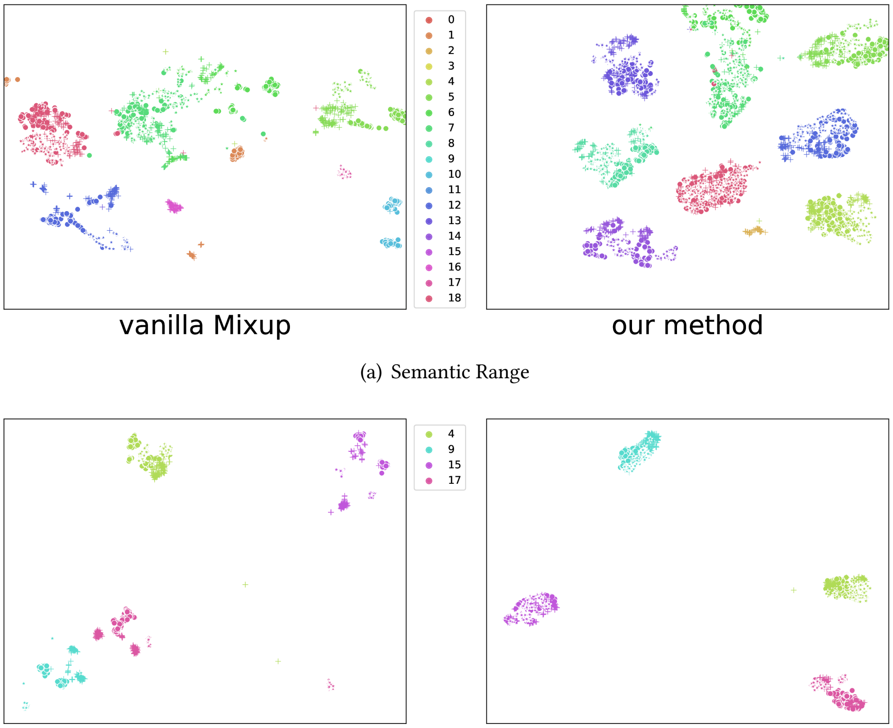
5.4 Analysis

Why is SDMix effective? Do semantic inconsistency and discriminative slackness really exist? To analyze the rationale behind the effectiveness of SDMix, we plot the confusion matrices for the vanilla Mixup and ours for target 0 on DSADS dataset in Figure 6. While most classes are correctly classified, we are more interested in the

[3] Diversity is relative. Few instances may be enough for simple situations (e.g. iid) while it may need more instances for difficult cases.
[4] GILE utilizes a different network architecture from other methods. Moreover, GILE selects the best model according to the target, which is unrealistic in reality and may be contrary to the principle of domain generalization.

misclassified classes: class 8 (walking in a parking lot) and class 9 (walking on a treadmill with a speed of 4km/h in flat); class 1 (standing) and class 6 (standing in an elevator still). Firstly, as shown in Figure 6(a), some samples in class 8 are misclassified as class 9 (Please see the row labeled 8). It is caused by different activity semantic ranges since walking in a parking lot has more variance than walking on a treadmill. Our method can correctly classify them. Secondly, some samples in class 6 are misclassified as class 1 (Please see the row labeled 6) since standing and standing in an elevator still are two similar activities with small differences that weaken the discrimination of Mixup. Again, our method can mitigate these two issues to achieve the best results as shown in Figure 6(b). As shown in Figure 6(c), class 8 in color 8 with shape circle has a larger semantic range than class 9 while class 1 almost compasses class 6 which illustrates they are close to each other. Figure 6(c) describes the phenomena while confusion matrices are the results brought by the phenomena, indicating the claim mentioned above.

In Figure 7(a), different classes (points with different colors) have different semantic ranges, which demonstrates that semantic inconsistency exists in reality and we should use semantic-aware Mixup. From Figure 7(b), we can see that data points in color 17 with shape + are close to data points in color 9, which proves the existence of the second issue. And from the right figure, we can see our method mitigates this issue. Figure 7 describes



(a) Semantic Range

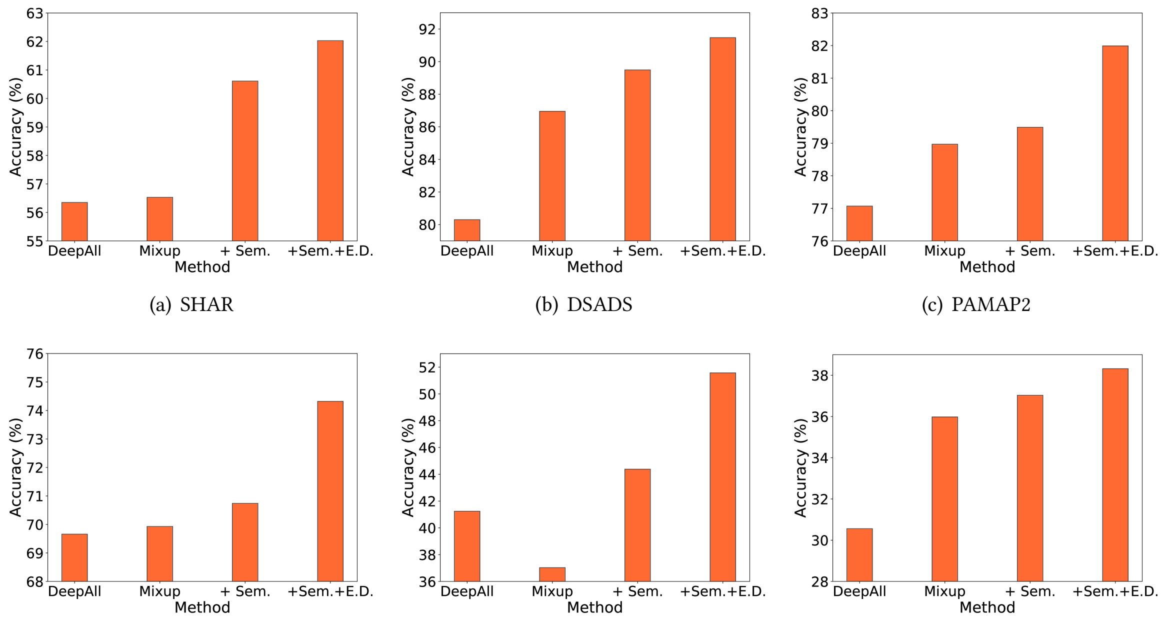
(b) Discrimination

Fig. 7. Visualization of the t-SNE embeddings of DSADS dataset for target 3 in the Cross-Person setting. Each class is denoted by a color and each shape corresponds to a domain. *Best viewed in color and zoom in.*

the phenomena and the improvements on the fourth task of DSADS in Cross-person setting also suggests that paying attention to semantic range and discrimination can bring better performance.[5]

5.5 Ablation Study

We perform ablation study in Figure 8. Figure 8(a) shows Mixup has a better performance than DeepAll while our Semantic-aware Mixup further improves accuracy on SHAR in the Cross-Person setting. Moreover, with large margin loss to enhance discrimination, there exists another improvement compared with semantic-aware Mixup, which demonstrates that our method achieves the best average accuracy and the two components of our SDMix are both effective. Results of ablation study on DSADS, PAMAP2, and USC-HAD in the Cross-Person setting and results in the Cross-Position setting (Figure 8(b), Figure 8(c), Figure 8(d), and Figure 8(f) respectively) all demonstrate that increasing diversity of data can bring better performance and two components of our SDMix are both effective. In Figure 8(e), Vanilla Mixup performs worse than DeepAll while ours achieve better results, which demonstrates that it is better to pay attention to semantic inconsistency and discrimination slackness when performing Mixup.



(a) SHAR

(b) DSADS

(c) PAMAP2

(d) USC-HAD

(e) Cross-Dataset

(f) Cross-Position

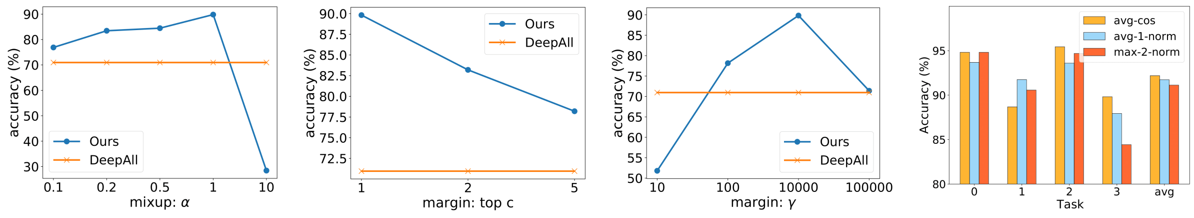
Fig. 8. Ablation study. Fig. 8(a)-8(d) are in the Cross-Person setting. Fig. 8(e) is in the Cross-Dataset setting while Fig. 8(f) is in the Cross-Position setting. 'Sem.' and 'E.D.' denotes semantic and enhancing discrimination, respectively.

5.6 Parameter Sensitivity Analysis

We evaluate the parameter sensitivity of SDMix in Figure 9. There are mainly four hyperparameters in our method: the Beta distribution in Mixup (α), the aggregation class number in Eq. (10) (top c), the required distance

[5] In contrast, paying no attention to semantic range and discrimination may suffer from some problems and have worse performance.

to boundaries in Eq. (10) (γ), and alternatives for computing activity semantic range in Eq. (5a) or Eq. (5b). From Figure 9(a)-9(c), we can see that the results with parameters around the highest points are all better than DeepAll. Figure 9(d) demonstrates that different approaches of computing activity semantic ranges obtain different results and we should choose the right one for better results through validation. Please note that our method achieves the best performance no matter which approach of computing is selected. In a nutshell, It demonstrates that SDMix is effective and robust that can be easily applied to real applications.



(a) α (b) Top c (c) γ (d) Alternatives to compute R_c

Fig. 9. Parameter sensitivity analysis (DSADS in the Cross-Person setting)

.

5.7 Extensibility

To demonstrate that our method is still a remark method in the common setting where training data and testing data share the same distribution, we conduct experiments on DSADS. We split DSADS into two parts, 20% for testing and the rest for training. When utilizing all rest data for training, the baseline accuracy is almost 100%.

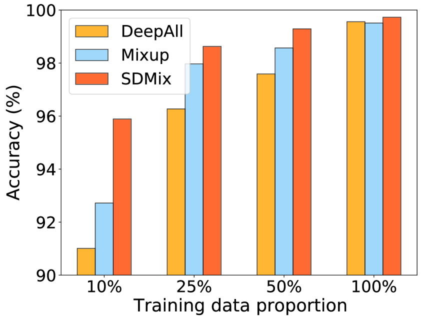


Fig. 10. Results on DSADS when training and testing data have the same distribution.

To better show performance differences between our method and other state-of-the-art methods, we utilize 10%, 25%, 50%, 100% of rest data for training. To select the best model during training, 20% of data serve as validation data while the rest data serve as the real training data. Therefore, when utilizing 10% of rest data for training, only 6.4% of all data are for training. The results are shown in Figure 10 where we can see that our method achieves the best performance compared to ERM and Mixup. Our method has improvements with 4.88%, 2.36%, 1.7%, 0.17% compared to ERM while it has improvements with 3.17%, 0.66%, 0.72%, 0.22% compared to Mixup using 10%, 25%, 50%, 100% training data respectively. As we can see, all methods achieve better performance with more training data used and our method achieves the best performance no matter how many training data are used. The results demonstrate that our method is still a remark method in a common setting.

6 CONCLUSION

In this paper, we proposed SDMix for generalizable sensor-based human activity recognition. SDMix solves two critical challenges: semantic inconsistency and discriminative slackness. Specifically, SDMix took the activity semantic range into consideration to learn more flexible interpolations. Moreover, SDMix introduced the large margin loss to Mixup that enhances the discrimination. Extensive experiments on cross-person, cross-dataset, and cross-position scenarios demonstrated the effectiveness of our method. In the future, we plan to extend SDMix to more challenging HAR settings including cross-category and incremental learning.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Plan of China No.2021YFC2501202, Natural Science Foundation of China (No. 61972383, No. 61902377, No. 61902379, No.62002187), Beijing Municipal Science & Technology Commission No.Z211100002121171.

REFERENCES

[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L Reyes-Ortiz. 2012. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *International workshop on ambient assisted living*. Springer, 216–223.

[2] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. 2018. Metareg: Towards domain generalization using meta-regularization. *Advances in neural information processing systems* 31 (2018).

[3] Billur Barshan and Murat Cihan Yüksek. 2014. Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units. *Comput. J.* 57, 11 (2014), 1649–1667.

[4] Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)* 46, 3 (2014), 1–33.

[5] Youngjae Chang, Akhil Mathur, Anton Isopoussu, Junehwa Song, and Fahim Kawsar. 2020. A systematic study of unsupervised domain adaptation for robust human-activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–30.

[6] Diane Cook, Kyle D Feuz, and Narayanan C Krishnan. 2013. Transfer learning for activity recognition: A survey. *Knowledge and information systems* 36, 3 (2013), 537–556.

[7] Gamaleldin Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio. 2018. Large Margin Deep Networks for Classification. In *Advances in Neural Information Processing Systems*, Vol. 31. 842–852.

[8] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research* 17, 1 (2016), 2096–2030.

[9] Vikas Garg, Adam Tauman Kalai, Katrina Ligett, and Steven Wu. 2021. Learn to expect the unexpected: Probably approximately correct domain generalization. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 3574–3582.

[10] Negar Golestani and Mahta Moghaddam. 2020. Human activity recognition using magnetic induction-based motion signals and deep recurrent neural networks. *Nature communications* 11, 1 (2020), 1–11.

[11] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. 2020. Self-challenging improves cross-domain generalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 124–140.

[12] Maximilian Ilse, Jakub M Tomczak, Christos Louizos, and Max Welling. 2020. Diva: Domain invariant variational autoencoders. In *Medical Imaging with Deep Learning*. PMLR, 322–348.

[13] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. 2018. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 624–639.

[14] Alexander H Liu, Yen-Cheng Liu, Yu-Ying Yeh, and Yu-Chiang Frank Wang. 2018. A unified feature disentangler for multi-domain image translation and manipulation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 2595–2604.

[15] Wang Lu, Yiqiang Chen, Jindong Wang, and Xin Qin. 2021. Cross-domain activity recognition via substructural optimal transport. *Neurocomputing* 454 (2021), 65–75.

[16] Massimiliano Mancini, Samuel Rota Bulo, Barbara Caputo, and Elisa Ricci. 2018. Best sources forward: domain generalization through source-specific nets. In *2018 25th IEEE international conference on image processing (ICIP)*. IEEE, 1353–1357.

[17] Daniela Micucci, Marco Mobilio, and Paolo Napoletano. 2017. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Applied Sciences* 7, 10 (2017), 1101.

[18] Narges Honarvar Nazari and Adriana Kovashka. 2020. Domain generalization using shape representation. In *European Conference on Computer Vision*. Springer, 666–670.

[19] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2009), 1345–1359.

[20] Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. 2021. Learning explanations that are hard to vary. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=hb1sDDSLbV

[21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, Vol. 32. 8026–8037.

[22] Hangwei Qian, Sinno Jialin Pan, and Chunyan Miao. 2021. Latent Independent Excitation for Generalizable Sensor-based Cross-Person Activity Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 11921–11929.

[23] Fengchun Qiao, Long Zhao, and Xi Peng. 2020. Learning to learn single domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12556–12565.

[24] Xin Qin, Yiqiang Chen, Jindong Wang, and Chaohui Yu. 2019. Cross-dataset activity recognition via adaptive spatial-temporal transfer learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 4 (2019), 1–25.

[25] Mohammad Mahfujur Rahman, Clinton Fookes, Mahsa Baktashmotlagh, and Sridha Sridharan. 2020. Correlation-aware adversarial domain adaptation and generalization. *Pattern Recognition* 100 (2020), 107124.

[26] Attila Reiss and Didier Stricker. 2012. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th international symposium on wearable computers*. IEEE, 108–109.

[27] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. 2020. Distributionally Robust Neural Networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=ryxGuJrFvS

[28] Yuge Shi, Jeffrey Seely, Philip H. S. Torr, N. Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. 2021. Gradient Matching for Domain Generalization. *CoRR* abs/2104.09937 (2021). arXiv:2104.09937 https://arxiv.org/abs/2104.09937

[29] Baochen Sun and Kate Saenko. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*. Springer, 443–450.

[30] Bailin Wang, Mirella Lapata, and Ivan Titov. 2021. Meta-learning for domain generalization in semantic parsing. In *NAACL*.

[31] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. 2019. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters* 119 (2019), 3–11.

[32] Jindong Wang, Yiqiang Chen, Lisha Hu, Xiaohui Peng, and S Yu Philip. 2018. Stratified transfer learning for cross-domain activity recognition. In *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 1–10.

[33] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, and Tao Qin. 2021. Generalizing to Unseen Domains: A Survey on Domain Generalization. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. 4627–4635. Survey Track.

[34] Jindong Wang, Vincent W Zheng, Yiqiang Chen, and Meiyu Huang. 2018. Deep transfer learning for cross-domain activity recognition. In *proceedings of the 3rd International Conference on Crowd Science and Engineering*. 1–8.

[35] Yufei Wang, Haoliang Li, and Alex C Kot. 2020. Heterogeneous domain generalization via domain mixup. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3622–3626.

[36] Garrett Wilson and Diane J Cook. 2020. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)* 11, 5 (2020), 1–46.

[37] Tzu-Tsung Wong. 2015. Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition* 48, 9 (2015), 2839–2846.

[38] Yuan Wu, Diana Inkpen, and Ahmed El-Roby. 2020. Dual mixup regularized learning for adversarial domain adaptation. In *European Conference on Computer Vision*. Springer, 540–555.

[39] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. 2020. Adversarial domain adaptation with domain mixup. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 6502–6509.

[40] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. 2021. A Fourier-based Framework for Domain Generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14383–14392.

[41] Zheng Xu, Wen Li, Li Niu, and Dong Xu. 2014. Exploiting low-rank structure from latent domains for domain generalization. In *European Conference on Computer Vision*. Springer, 628–643.

[42] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond Empirical Risk Minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=r1Ddp1-Rb

[43] Mi Zhang and Alexander A Sawchuk. 2012. USC-HAD: a daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proceedings of the 2012 ACM conference on ubiquitous computing*. 1036–1043.