VALERIAN: Invariant Feature Learning for IMU Sensor-based Human Activity Recognition in the Wild

Yujiao Hao
McMaster University
Hamilton, Ontario, Canada
haoy21@mcmaster.ca

Boyu Wang
Western Ontario University
London, Ontario, Canada
bwang@csd.uwo.ca

Rong Zheng
McMaster University
Hamilton, Ontario, Canada
rzheng@mcmaster.ca

ABSTRACT

Deep neural network models for IMU sensor-based human activity recognition (HAR) that are trained from controlled, well-curated datasets suffer from poor generalizability in practical deployments. However, data collected from naturalistic settings often contains significant label noise. In this work, we examine two in-the-wild HAR datasets and DivideMix, a state-of-the-art learning with noise labels (LNL) method to understand the extent and impacts of noisy labels in training data. Our empirical analysis reveals that the substantial domain gaps among diverse subjects cause LNL methods to violate a key underlying assumption, namely, neural networks tend to fit simpler (and thus clean) data in early training epochs. Motivated by the insights, we design VALERIAN, an invariant feature learning method for in-the-wild wearable sensor-based HAR. By training a multi-task model with separate task-specific layers for each subject, VALERIAN allows noisy labels to be dealt with individually while benefiting from shared feature representation across subjects. We evaluated VALERIAN on four datasets, two collected in a controlled environment and two in the wild. Experimental results show that VALERIAN significantly outperforms baseline approaches. VALERIAN can correct 75% − 93% of label errors in the source domains. When only 10-second clean labeled data per class is available from a new target subject, even with 40% label noise in training data, it achieves ∼ 83% test accuracy.

Code is available at: https://github.com/YujiaoHao/VALERIAN.git

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing**; • **Computing methodologies** → **Neural networks**; Supervised learning.

KEYWORDS

IMU, multi-task learning, human activity recognition, domain adaptation, learning with noisy labels

ACM Reference Format:
Yujiao Hao, Boyu Wang, and Rong Zheng. 2023. VALERIAN: Invariant Feature Learning for IMU Sensor-based Human Activity Recognition in the Wild. In *International Conference on Internet-of-Things Design and Implementation (IoTDI '23), May 9–12, 2023, San Antonio, TX, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3576842.3582390

1 INTRODUCTION

Inertial measurement unit (IMU) sensor-based human activity recognition (HAR) has gained a lot of interest recently due to its pervasiveness in smartphones and smartwatch devices [25, 26, 37, 46]. With the increasing adoption of deep neural network models in HAR tasks, there is a need to acquire a large amount of well-curated and labeled sensory data to train such models. Unfortunately, the majority of public HAR datasets are from controlled settings where subjects are asked to perform prescribed activities in lab environments. They typically contain a small collection of subjects and activity types over a limited period of time. For example, PAMAP2 [28], a popular dataset for HAR, only includes eight subjects with 59.67 minutes of measurements per subject. Furthermore, data collected from controlled settings often have very different characteristics from those of freestyle motions in naturalistic environments [35].

Collecting IMU sensor data in the wild faces its own set of challenges. Arguably, the biggest difficulty is to label such data accurately [42]. Recalls from one's memory are known to be notoriously unreliable [27]. Labeling wearable data by observing signal patterns requires extensive domain knowledge and experience since sensor readings are impacted by not only activity types but also subject characteristics, on-body positions and sensor orientations. A mainstream method to label such data is to resort to another human-interpretable modality such as visual or audio recordings and determine the labels manually post hoc. Unfortunately, labels obtained this way are still error-prone due to mis-synchronization across different modalities, human errors or missing data (e.g, occlusion in vision data). As the first contribution of the work, *we examine two datasets collected in naturalistic settings to understand the extent and characteristics of noisy labels.*

Learning with noisy labels (LNL) has long been investigated in the machine learning community with many effective methods being proposed for computer vision tasks. Due to the lack of reliable ground truth in real-world noisy data, studies are mainly conducted by adding artificial noise to clean labeled datasets [32]. Through an empirical study, we find that DivideMix, one state-of-the-art LNL method fails to achieve good accuracy and sometimes cannot converge at all. In-depth analysis reveals that the root cause is the violation of a key underlying assumption in LNL methods, i.e., models fit simpler (and thus clean) data in early training epochs. With substantial subject diversity, it is difficult to distinguish wrongly labeled data from correct ones from a different subject whose data

IoTDI '23, May 9–12, 2023, San Antonio, TX, USA

ACM ISBN 979-8-4007-0037-8/23/05…$15.00
https://doi.org/10.1145/3576842.3582390

follows a different distribution (also known as *domain gaps*). Therefore, the second contribution of the work is *to unravel the interplay between subject domain gaps and LNL for HAR tasks.*

The insights from the empirical study motivate our third contribution, namely, the design of VALERIAN, an inVariant feAture LEarning foR In-the-wild domain AdaptatioN method of IMU-based HAR. Its core component is a one-step domain invariant feature learner that tackles label noises and learns the shared feature representation among multiple subjects simultaneously. VALERIAN uses self-supervised pretraining to learn robust features that are independent of label quality. The pretrained parameters are used to initialize the shared feature encoder of a multi-task learning model, where each noisy labeled subject in the training set is considered as a separate task. The network consists of shared feature encoder and subject-dependent task-specific layers that are trained iteratively with noisy labeled data. To combat noisy labels, early-learning regularization (ELR) [22] is adopted by introducing a loss term reflecting the temporal ensemble of past predictions. VALERIAN can be applied in two ways: 1) *label correction*, i.e., to clean the labels of noisy labeled datasets so that accurate HAR models can be developed, and 2) *domain adaption*, i.e., to adapt the trained model to an unseen subject. Specifically, VALERIAN can predict activity labels of each subject in the training set using a respective task-specific layer. To achieve higher accuracy, we assume a small number of correctly labeled data is available from a new subject. The data is used to update a task-specific layer to allow fast adaption of the trained model to the subject.

We evaluate the performance of VALERIAN using two controlled datasets with different levels and distributions of labeling noises, and two in-the-wild datasets. Noises are introduced to investigate the impact of the amount of label noise on model performance. VALERIAN consistently outperforms baseline approaches across all settings. In label correction, VALERIAN can correct up to 93% wrongly labeled samples. In domain adaptation, even with 40% label noise in training data, it achieves an ∼ 83% test accuracy with only 10 seconds of correctly labeled data per class. A similar evaluation on a true in-the-wild dataset with noisy labels achieves an over 20% improvement in the F1-Score compared to baseline methods.

The rest of the paper is organized as follows. Section 2 describes the motivation of this work. In Section 3, we introduce the VALERIAN method and the details of each component. In Section 4, we present the implementation details and performance evaluation of VALERIAN. Section 5 describes the related work and how they differ from ours. Finally, we conclude the paper in Section 6 with a summary and a discussion of future research directions.

2 MOTIVATION

To understand the characteristics of in-the-wild HAR datasets and to gain insights into why mainstream LNL methods tend to fail on such tasks, we inspect two datasets and the behavior of a state-of-the-art (SOTA) LNL algorithm in this section.

2.1 Characteristics of in-the-wild HAR datasets

In this work, a HAR dataset is considered to be in the wild (or collected in naturalistic settings) if the activities of subjects are not precisely scripted. As a result, experimenters do not know exactly what activities shall be performed at what time. The ExtraSensory dataset is one such example [36], where IMU data were collected from users' smartphone devices as they went about their daily activities. Activity labels were initially self-reported. Further curation was done by researchers who utilized information from other sensing modalities to automatically correct some labels. A detailed description of the curation procedure in ExtraSensory can be found in [35]. As another example, the Realworld dataset [34] contains data collected from fifteen subjects performing activities such as climbing stairs down and up, jumping, lying, standing, sitting, running/jogging and walking. Although in most cases, subjects were asked to perform a certain activity, during climbing up/downstairs outside trials, the variations of terrains are not controlled by the experimenters and thus un-prescribed activities may occur.

Fig. 1 illustrates the percentage of clean and mislabeled data in both datasets. For RealWorld, we inspect the video recording of climbing up and climbing down trials, note down the start and end times, and the type of activities. We find that there are periods when the subjects actually walk on flat ground (7% of the time) or stand still (3% of the time), which were mislabeled as climbing up or down in the dataset. For ExtraSensory, when comparing the self-reported and curated labels, we find that 34.5% are unchanged, 39.2% are corrected in the curation process and 26.3% are marked as invalid since the phones were not with the subjects during data collection. Moreover, upon closer inspection of curated data in ExtraSensory, we find the data labels are still noisy. For example, in Fig. 2, the left plot corresponds to accelerometer measurements labeled as standing while the right one is labeled as walking. However, one can easily observe the "signature" periodical pattern associated with walk cycles in the left plot rather than in the right plot – an indication of mislabeling even after auto-curation.

From Fig. 1, we conclude ExtraSensory is much noisier than RealWorld since the former is crowdsourced data. What also distinguishes the two datasets is the distribution of label noises. Specifically, for RealWorld, most mislabeling happens in the climbing up/down trials when the ground labels are "walk on a flat ground" or standing. In contrast, in ExtraSensory, mislabeling exists almost between any two activities. To characterize the distribution of noisy labels, a noise transition matrix T is often used, where element T_{ij} corresponds to the probability of mislabeling a data sample with ground truth label i to label j [11]. When mislabels occur equally likely for all classes other than the true class, the associate noise pattern is called *symmetric noise.* Otherwise, if there is a dominant off-diagonal element in each row in T, the associate noise pattern is called *asymmetric noise.*

Table 1 shows the noise transition matrix of data in three locomotion classes and one location class in ExtraSensory by comparing their curated labels (row headings) and the original ones (column headings). As ExtraSensory is a multi-label dataset with many classes, only top-5 mutually exclusive labels are included in the table. We observe that with the exception of "running", noise transition probabilities of all classes are best modeled as symmetric noise.

Table 1: The noise transition matrix of ExtraSensory, based on its curated labels. For walking and standing, only top-4 mislabeling sources are shown due to space limits.

	walking	strolling	cleaning	cooking	eating
walking	75.28%	3.46%	3.46%	2.35%	1.67%
	running	exercise	go upstairs	go downstairs	
running	79.92%	19.66%	0.21%	0.21%	
	standing	cooking	cleaning	shower	dressing
standing	56.79%	8.47%	7.51%	5.35%	5.34%
	at home	at school	at work	at party	at gym
at home	96.71%	1.49%	1.27%	0.27%	0.26%

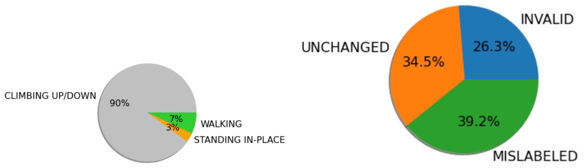


Figure 1: The statistics of two in-the-wild IMU-based HAR datasets. Left: Realworld, Right: ExtraSensory. A noticeable portion of the data labels in both datasets are noisy.

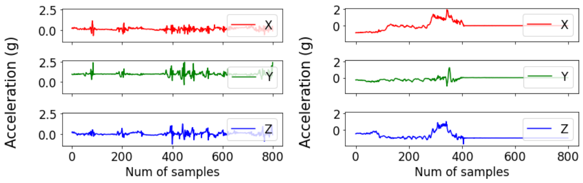
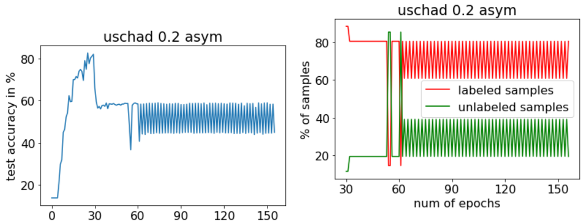


Figure 2: Accelerometer data in ExtraSensory with curated labels. Left: standing (subject id: FDAA70A1-42A3-4E3F-9AE3-3FDA412E03BF, row id: 4339), Right: walking (subject id: 2C32C23E-E30C-498A-8DD2-0EFB9150A02E, row id: 5454).

2.2 LNL can be Harmful to IMU-based HAR with noisy labels

Learning with noisy labels has long attracted attention with many deep learning-based methods proposed recently that primarily target computer vision tasks. According to [32], there are mainly four categories of LNL methods: robust architecture, robust regularization, robust loss design and sample selection. In this section, we use DivideMix [20], a representative sample selection based method to illustrate the behavior and deficiency of LNL. In Section 4, results from a robust regulation method are presented.

The basic idea of DivideMix is to first initialize a model with all training data for a few epochs (called *warm-up phase*). A Gaussian mixture with two modes is fitted to divide data samples based on their normalized losses into two partitions – those with lower losses (higher confidence) are considered clean labeled samples, and those with high losses are treated as unlabeled data. Semi-supervised learning is then applied to the mixed data. Subsequently, co-refinement of labeled data and co-guessing of the labels of unlabeled data is performed by two neural networks working together iteratively, to reduce biases.



(a) Test accuracy

(b) Division of clean and noisy labeled data

Figure 3: The performance of DivideMix on USCHAD in leave-one-subject-out experiments.

To study the behavior of DivideMix for HAR, we add artificial noise to clean labeled dataset. The USCHAD dataset [50] is selected as it contains carefully curated ground truth labels. This dataset consists of accelerometer and gyroscope measurements collected from fourteen participants performing ten types of locomotions in a controlled environment (i.e., walking forward, walking left, walking right, walking upstairs, walking downstairs, running forward, jumping up, sitting, standing and sleeping). Both symmetric and asymmetric noise patterns are considered, but due to space limits, only results from asymmetric noise are included. The transition matrix of asymmetric noise is defined by flipping pairs of the most confusing activities (see Fig. 8 for details). We adopt the DeepConvLSTM model architecture proposed in [26] as feature extractor for HAR tasks. The model contains 4 convolutional neural network (CNN) layers and 2 long short-term memory (LSTM) layers totalling ~296k trainable parameters.

Fig. 3 shows the behavior of DivideMix over training epochs in presence of 0.2 asymmetric labelling noise. In the experiments, 13 of 14 subjects are included in the training data and the remaining subject is used in testing. The warm-up phase ends at 30 epochs. As shown in Fig. 3a, test accuracy increases quickly during the warm-up phase indicating that the model can learn despite label noises. However, after the warm-up phase, the test accuracy drops drastically and fluctuates between 45% and 60% after 60 epochs. A closer look at the division between labeled and unlabeled data in the training set is in Fig. 3b. It reveals that despite only 20% of the data samples being labeled incorrectly, DivideMix gradually converges to split the data approximately 81-19 or 61-39. As a result, some clean labeled data is classified as unlabeled and fail to contribute as much to the training process.

To shed the light on why DivideMix fails in these experiments, further analysis is in order. First, we inspect the effect of memorization. Deep neural network models are known to have the propensity for fitting training data including outliers or mislabeled data. However, it has been empirically demonstrated that such a memorization phenomenon tends to happen at a late stage of training [2, 22]. In early training epochs, the model prioritizes learning simple patterns. To test if this hypothesis is true for HAR tasks, we show in Fig. 4 the breakdown of training samples among five

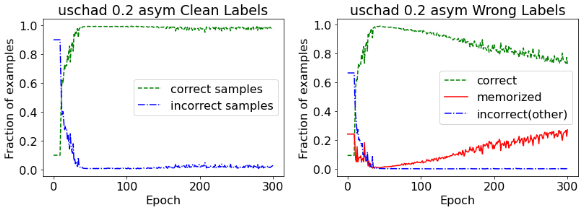


Figure 4: Results of the DivideMix model on USCHAD with 0.2 asymmetric noise. Left: the fraction of clean labeled samples that are predicted correctly (green) and incorrectly (blue). Right: the fraction of samples with wrong labels that are predicted correctly (green), memorized (red), and incorrectly as neither the true nor the labeled class (blue).

categories. Specifically, a data sample that is correctly labeled can be either correctly or wrongly predicted by the trained model up to the associated epoch. For a data sample that is wrongly labeled, three situations may arise: i) its prediction is the same as the ground truth label (*correct*), ii) its prediction is the same as the wrong label (*memorized*) or iii) otherwise, i.e., its prediction is neither the ground truth label nor the wrong label. From Fig. 4, even after a few epochs, memorization is non-negligible, especially in the case of asymmetric noise. When a noisy label is memorized, the model has high confidence in its *wrong* prediction.

We believe the root cause of early memorization and the consequent failure of DivideMix in HAR tasks is due to the large variability across subjects when performing the same activity. Subject diversity is a well-recognized problem in IMU-based HAR [6]. However, the problem is exacerbated when noisy labels are present. In Fig. 5, we show the normalized cross-entropy losses for Subject 2 – 14 in the training data and the division of clean and noisy labels for each subject in DivideMix after a 30-epoch warm-up period. Clearly, the normalized losses (Fig. 5(a)) no longer follow a two-component GMM. Instead, they are better modelled by a mixture of three or more components. Inspecting the division of labelled and unlabeled data for each subject by DivideMix, we find that some data presumed to be clean is in fact noisy (Fig. 5(b)) while a portion of presumably noisy data is in fact clean for each subject (false noisy in Fig. 5(c)). Some subject (e.g., Subject 14) appears to be penalized with a higher percentage of clean data being mislabeled as unlabeled by DivideMix. More than 10% of Subject 14's clean data is misclassified as noisy (due to high normalized losses).

Fig. 6 shows the case when training DivideMix on data from one subject with 0.2 asymmetric labeling noise. We train the model using data from 4 trials of the subject and test with the remaining trial. To avoid overfitting, the network size of DeepConvLSTM is reduced by retaining only two CNN layers and one LSTM layer with a total of 56k trainable parameters. The distributions in Fig. 6 can indeed be modeled as 2-component GMM following the basic assumption of DivideMix and thus can be correctly handled by the method (results omitted for brevity). Comparing the results from Fig. 6 with Fig. 5(a) and Fig. 3, it is clear that DivideMix works reasonably well on data from a single subject but failed in the case of multiple subjects. Therefore, it is reasonable to conclude that the discrepancy is due to the domain gaps in multi-subject data.

Though our analysis focuses on DivideMix, other categories of LNL methods such as ELR [22] and CDR [43] make the same assumptions that high-confidence labels in early training stages are more trustworthy. Unfortunately, as evident from the empirical analysis in this section, such assumptions no longer hold in presence of diverse subject data in HAR tasks.

3 METHOD

Let the input and label spaces be $\mathcal{X}$ and $\mathcal{Y}$, respectively. Due to high subject diversity in HAR tasks, each subject in the training set is treated as a separate source domain in the joint space $\mathcal{X} \times \mathcal{Y}$. In the rest of the paper, we use "domain" and "subject" interchangeably. Let $\mathcal{D}_k = \{(x_n^k, \tilde{y}_n^k)\}_{n=1}^{N_k}$, where N_k is the number of data samples from subject k and $\tilde{y}$ denotes noisy labels. The source domains are denoted by $\mathcal{D}_s = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_K\}$, where K is the number of subjects. We further assume that a small collection of clean labeled samples can be obtained for an unseen subject t denoted by $\mathcal{D}_t = \{(x_n, y_n)\}_{n=1}^{M}$. The goal of *HAR from data in-the-wild* is to learn a model from $\mathcal{D}_s$ that can either be easily adapted to a new target domain given $\mathcal{D}_t$, or be used to correct the wrong labels in $\mathcal{D}_s$.

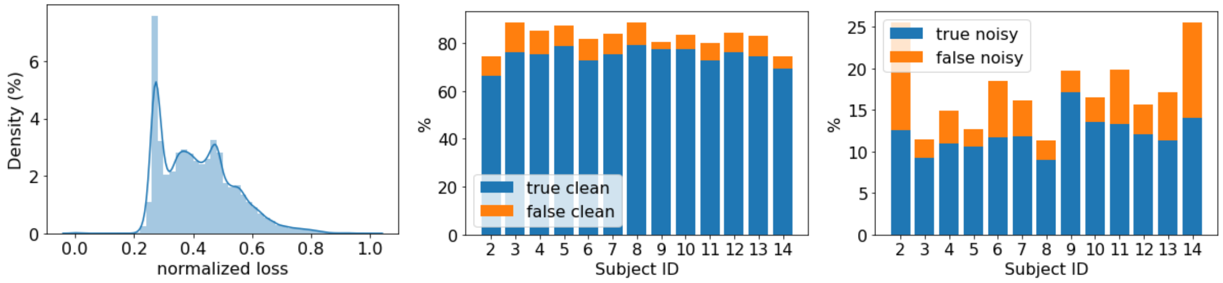
Motivated by the observations from Section 2, we propose VALERIAN, a one-step method that handles noisy labels and distribution gaps across multiple source domains simultaneously. Our solution is based on two key insights: i) unsupervised learning that aims to learn representations invariant to instance-level variations is not affected by noisy labels; and ii) within each source domain, clean data tends to exhibit simpler patterns (than wrongly labelled data), which can be learned in early training epochs. Moreover, we assume that in absence of noisy labels, there exist domain-invariant features across subjects in HAR tasks. This assumption has been empirically verified in prior work [12]. After model training, VALERIAN can be used in both cleaning noisy labels in the training data and enabling fast adaptation to a new subject from a small amount of clean labeled data (Fig. 7).

3.1 Solution overview

VALERIAN takes advantage of known techniques in machine learning but combines them in innovative ways. It has three key building blocks: i) self-supervised pre-training, ii) invariant feature learning from noisy labelled data, and iii) fast adaption to unseen subjects.

Self-supervised pre-training takes unlabeled data and performs data augmentation to pre-train feature extractor that captures structures of underlying distributions. Invariant feature learning in VALERIAN has two objectives: 1) to learn shared feature representations across domains and 2) to combat the memorization effect introduced by noisy labels. To do so, we adopt a multi-task learning model for domain invariant feature learning which was first proposed in [12]. The model consists of a shared feature extractor across multiple source domains and multiple task-specific output layers. To counter the effect of noisy labels, we introduce a regularization term similar to ELR in the loss function during training. Finally, for a new subject with a small amount of clean data, fast adaption is performed on one of the task-specific layers only.

Algorithm 1 summarizes the training procedure of VALERIAN. Next, we will provide the details of each building block.

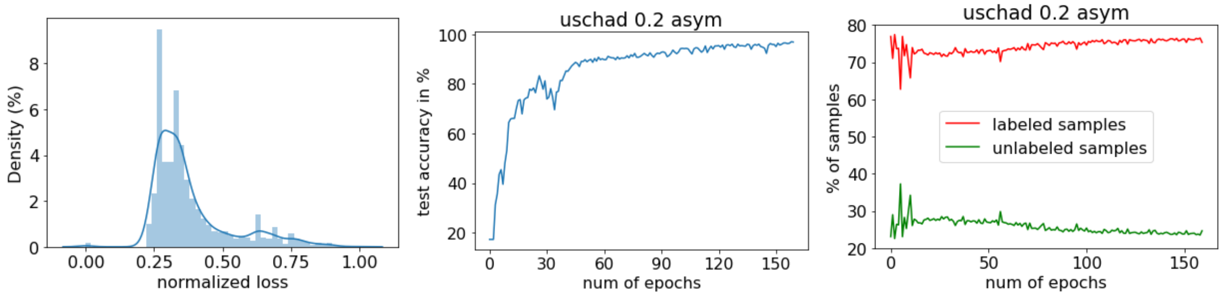


(a) Distribution of normalized losses

(b) Partition of data predicted as clean

(c) Partition of data predicted as noisy

Figure 5: Effects of subject diversity on early learning. Plots are generated on a model trained on Subject 2 − 14 in USCHAD with 0.2 asymmetric noise and after 30 epochs of warm-up training in DivideMix.



(a) Distribution of normalized losses

(b) Test accuracy

(c) Division of clean and noisy labeled data

Figure 6: Performance of DivideMix on a single subject (subject id: 2). (a) is generated after the warm-up phase, while (b) and (c) are generated on the full training process of a DivideMix model.

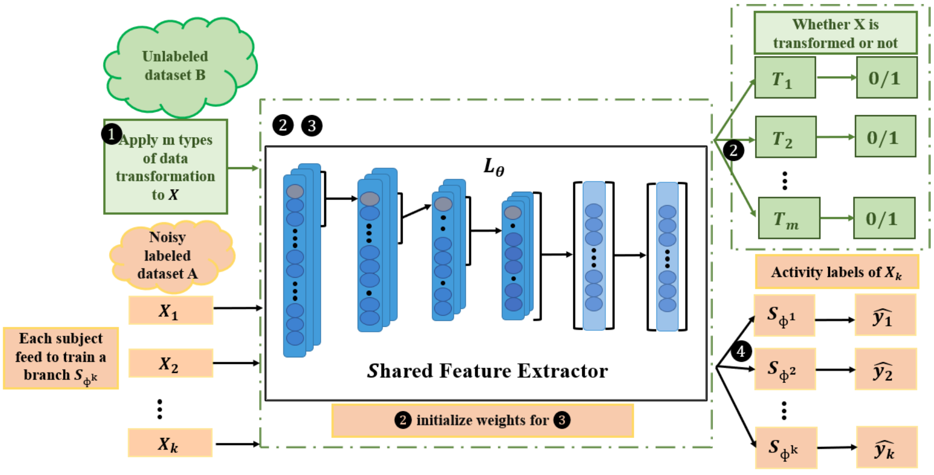


Figure 7: An overview of VALERIAN training procedure. Step 1: Data transformation. Step 2: self-supervised training. Step 3 and 4: alternating training.

3.2 Self-supervised pre-training

In [14], the authors find that a ResNet pre-trained on ImageNet datasets appears to work consistently better than random initialized ones as a feature extraction network for LNL image classification tasks. Inspired by this, we pre-train a feature extractor network by removing the labels in HAR datasets. It is thus natural to consider feature learners that require no label information, such as contrastive learning [7] or self-supervised learning. Self-supervised

Algorithm 1 Invariant feature learning for in-the-wild domain adaptation

```
Require: Source domains $\mathcal{D}_s = \{D_k\}_{k=1}^K$, learning rate $\gamma$, hyperparameters $\alpha, \beta, \lambda, \mu$
Ensure: VALERIAN model with parameter $\theta$ and $\phi$
Initialize $\theta$ with self-supervised pretrain
Random initialize $\phi = \{\phi^1, \phi^2, ..., \phi^K\}$
Initialize ensemble predictions $t \leftarrow 0_{[n \times C]}$
repeat
    Sample tasks $T = \{T_1, T_2, ..., T_K\}$ over $\mathcal{D}_s$
    //Update $\phi^k$ with fixed $\theta$
    for $k$ is 1 to $K$ do
        Freeze parameters of $\phi$ except $\phi^k$
        for each minibatch B in $T_k$ do
            for $(x_i, \tilde{y}_i)$ in B do
                $p_i \leftarrow S_{\phi^k}(L_\theta(x_i))$
                $t_i \leftarrow \beta t_i + (1 - \beta) p_i$
            end for
        end for
        $\mathcal{L}oss \leftarrow \mathcal{L}_{CE}(T_k, \theta; \phi^k) + \mu|\phi^k|_1 + \frac{\lambda}{|B|} \sum \log(1 - \langle p_i, t_i \rangle)$
        $\phi^k \leftarrow \phi^k - \gamma \nabla_{\phi^k} \mathcal{L}oss(T_k, \theta; \phi^k)$
    end for
    //Update $\theta$ with fixed $\phi$
    for each minibatch B in $T$ do
        $B' = Mixup(B, \alpha)$
        for $(x_i, \tilde{y}_i)$ in B' do
            $p_i \leftarrow S_\phi(L_\theta(x_i))$
            $t_i \leftarrow \beta t_i + (1 - \beta) p_i$
        end for
    end for
    $\mathcal{L}oss \leftarrow \mathcal{L}_{CE}(T, \phi; \theta) + \mu|\phi|_1 + \frac{\lambda}{|B|} \sum \log(1 - \langle p_i, t_i \rangle)$
    $\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}oss(T, \phi; \theta)$
until convergence
```

learning is a machine learning method that learns semantic features from unlabeled data with customized tasks [9]. As there is no ground truth label, to leverage of this technique, data augmentation techniques and auxiliary tasks need to be introduced. In [30], Saeed *et al.* introduce various data transformations and train a multi-task model to classify the type of transformation applied. The features extracted from the IMU data embed information regarding natural human motion while the transformed ones introduce different degrees of distortion. Trained to classify the type of transformation applied, a neural feature extractor learns to represent human motion more accurately and obtains more meaningful discriminative features. We adopt the same idea and apply the following transformations:

(1) *Noised*: it adds random Gaussian noise to the original data samples.
(2) *Scaled*: this transformation changes the magnitude of data samples within a sliding window by multiplying with a random scalar.
(3) *Rotated*: this transformation mimics different sensor orientations by multiplying the original data with a rotation matrix of randomly generated axis-angle.
(4) *Negated*: this transformation negates samples within a time window, resulting in a vertical or a horizontal flip of the original input signal.
(5) *Reversed*: it reverses the data along the time-axis, resulting in a complete mirror image of the original input.
(6) *Permuted*: sensor signals are randomly sliced and swapped within a data window.
(7) *Time-Warped*: it mimics the change of motion frequency by locally stretching or warping a time series through a smooth distortion of time intervals.
(8) *Channel-Shuffled*: it randomly shuffles sensor data in axial dimensions.

One or several of these transformations (called *pretext tasks*) are applied to each data window of each sensor separately (accelerometer and gyroscope). Each head of the multitask learning model corresponds to a binary classifier. By learning whether a certain type of transformation has been applied to the original data samples, the feature extractor portion of the network captures high-level semantic information that is invariant to these transformations and thus beneficial to downstream tasks.

3.3 Domain invariant feature learning

Self-supervised learning alone is insufficient to handle domain gaps among subjects. Moreover, data labels are necessary to fine tune model parameters for downstream tasks. To generalize well to unseen subjects, we utilize the invariant feature learning framework (IFLF) from [12] but modify it to work with LNL. It consists of three components:

Alternating training. An IFLF model is a multi-task model trained with tasks sampled from all source domains. Each subject has its individual task-specific layer S_{ϕ^k} but shares a common feature extractor network L_θ. If the model is trained by simply iterating among tasks sampled from $\mathcal{D}_1$ to $\mathcal{D}_K$, catastrophic forgetting may occur[17], namely, a model forgets previously learned tasks, and can only work properly on newly learned tasks. To avoid catastrophic forgetting, the alternating training strategy is employed from [18], to update L_θ and S_{ϕ^k} separately. In each training epoch, we first freeze the parameters of the feature extractor network, and update the parameters of each task-specific layer with its respective data; then, we freeze the parameters of all task-specific layers and update the invariant feature extractor using all data from the previous step.

Feature extractor. By the merit of multi-task learning, L_θ generalizes well across domains through the shared representations among related tasks [29]. For HAR tasks, we use DeepConvLSTM [26] as the backbone network. It includes four CNN layers and two LSTM layers.

The objective function ℓ_L works on multiple source domains to learn a domain invariant feature representation that clusters the features by their labels. It is defined as follows:

$$\ell_L = \sum_{k=1}^{K} \mathcal{L}_{CE}(T_k, \phi^k; \theta), \tag{1}$$

where $\mathcal{L}_{CE}$ is the categorical cross-entropy loss function calculated on each T_k with given θ and ϕ, defined as $\mathcal{L}_{CE} = -\sum_{i=1}^{C} \tilde{y}_i log(p_i)$

on data from each task k. We call such a multi-task model *basic multi-task learning model* (BMTL). To further boost the quality of extracted features, we use self-supervised pre-train as described in Section 3.2 to initialize the model parameter θ.

Task-specific networks. Generally, if the shared feature generalizes well across all source domains, it also works well on the target domain. L_θ needs to have sufficient capacity to explore the entire latent space $\mathcal{Z}$ and extract domain invariant features. In contrast, a task-specific network S_ϕ^k should be as simple as possible with fewer learnable parameters to allow fast adaptation with target domain data. In the implementation, a lightweight task-specific layer S_{ϕ^k} includes a fully connected layer with a softmax activation function. The task-specific objective function is defined as the sum of a categorical cross-entropy loss and an ℓ_1-norm regularization term as follows,

$$\ell_{S^k} = \mathcal{L}_{CE}(T_k, \theta; \phi^k) + \mu|\phi^k|_1, k = 1, 2, \ldots, K, \tag{2}$$

where μ is a hyper-parameter to control the sparsity of S_ϕ^k. The regularization term imposes sparsity on the task-specific layers and helps mitigate overfitting.

3.4 Learning with noisy labels

With the multitask learning model introduced previously, we can get the best of both worlds: shared network parameters for feature extraction for all subjects and subject-dependent output layers. As a result, the underlying assumption of dominant LNL methods is that in early training epochs, each subject-dependent model tends to incur low losses (higher confidence) on clean data and large losses on mislabeled data are likely to hold. To handle noisy labels, in principle, we can incorporate any existing LNL method in the invariant feature learning framework. However, we find that DivideMix has high training costs due to its use of two networks in co-teaching and co-refinement. When combined with invariant feature learning, its complexity grows linearly with the number of source domains. Therefore, in VALERIAN, we use ELR to counter memorization effects by forcing model predictions to be close to their temporal ensemble. An ELR loss is defined as :

$$\mathcal{L}_{elr} = \frac{1}{|B|} \sum_{i=1}^{|B|} \log\left(1 - \langle p_i, t_i \rangle\right), \tag{3}$$

where p_i is the model output of input sample x_i, and $t_i = \beta t_i + (1 - \beta) p_i$ is the temporal ensemble controlled by hsyper-parameter β. (3) maximizes the inner product of p_i and t_i, and the logarithm in $\mathcal{L}_{elr}$ inverts the exponential function implicit in the softmax function in p_i.

MixUP [49] is a simple yet effective data augmentation technique in improving model generalization capabilities [44]. In HAR tasks, we can mix up data samples from the same activity class but different subjects. To apply Mixup data augmentation, each data sample of a mini-batch is interpolated with another sample randomly chosen from a different source domain but belongs to the same class. Specifically, for a pair of samples $(x_1, \tilde{y}) \in \mathcal{D}_i$ and $(x_2, \tilde{y}) \in \mathcal{D}_j$, the mixed data sample $(x', \tilde{y})$ is computed by:

$$a \sim Beta(\alpha, \alpha), \tag{4}$$

$$a' = max(a, 1 - a), \tag{5}$$

$$x' = a' x_1 + (1 - a') x_2 \tag{6}$$

where a is the MixUp factor sampled from a $Beta$ distribution controlled by hyper-parameter α. Finally, the total losses in (1) and (2) are updated as:

$$\mathcal{L}oss_L = \ell_L + \mu|\phi|_1 + \lambda \mathcal{L}_{elr}, \tag{7}$$

$$\mathcal{L}oss_{S^k} = \ell_{S^k} + \lambda \mathcal{L}_{elr}, k = 1, 2, \ldots, K, \tag{8}$$

where λ is a hyper-parameter to control the importance of ELR. It is worth noting that the loss is calculated differently in the alternating training procedure as L_θ includes all source domains while ϕ^k only concerns the data of the kth subject. MixUp augmentation is only used in updating the feature extraction layers (L_θ).

3.5 Applications of VALERIAN

3.5.1 Label correction. After learning the domain invariant features from a noisy training set, VALERIAN is capable of relabeling the training samples close to their ground truth activities. For a (noisy labeled) data sample $(X, \tilde{y})$ from subject k, the prediction $\hat{y}^k$ of task-specific layer S_{ϕ^k} is taken as the new label for the sample.

3.5.2 Fast adaptation to new subjects. Since the network parameters in task-specific layers are already sparse, for a new subject, one can either initiate a new task-specific layer from scratch or randomly select a S_{ϕ^k} to update its trained parameters. A small amount of clean data is taken from $\mathcal{D}_t$ to train the task-specific layer.

4 PERFORMANCE EVALUATION

4.1 Datasets

We consider four publicly available datasets to cover a wide variety of device types, data collection protocols, and activity classes for recognition. Because the evaluation of machine learning models requires the availability of clean ground truth labels, the first two datasets, USCHAD and WISDM [41] were collected under controlled laboratory environments. To simulate labelling errors, symmetric or asymmetric noise is injected into the labels with different noise transition matrices. WISDM contains a large number of subjects. Raw accelerometer and gyroscope data were collected from a smartphone in each participant's pants pocket at a rate of 20Hz. There are a total of 51 test subjects performing seven locomotion activities (i.e., walking, jogging, stairs, sitting, standing, kicking a soccer ball, playing tennis) for three minutes per trial to achieve equal class distribution.

The third and fourth datasets, ExtraSensory and RealWorld, allow us to gauge VALERIAN's ability to handle real in-the-wild data. In ExtraSensory, crowdsourced mobile phone data are collected from 60 subjects during daily living activities. In the evaluation, we only consider six locomotion-related activities, namely, walking, running, cycling, sitting, standing and lying down. In the absence of ground truth labels, we take instead the curated data labels as ground truth. However, as discussed in Section 2, the curated data remains to be noisy. Moreover, ExtraSensory also suffers from severe class imbalance and missing class issues (only nine out of 60 subjects have data from all six classes in the dataset).

4.2 Baseline methods

Five baseline models have been implemented for comparison.

- *Single-task learning model (STL)*: STL is trained from scratch solely on the clean data from a target domain (a new subject). As the number of clean data increases, it is expected STL's performance to improve since there is no label noise.
- *Basic multi-task learning model (BMTL)*: Similar to VALERIAN, BMTL is a multi-task learning approach trained with noisy source domains and adapted to a target domain with a small number of clean labels. However, unlike VALERIAN, BMTL does not perform self-supervised pre-training and treats all training data as if it were clean.
- *Subject-independent model with cross-entropy losses (SI)*: It pools all but test subjects' data to train a single subject-independent model and treats all training data as clean.
- *Subject-independent model with ELR (SI-ELR)*: It is a subject-independent model trained by pooling all but test subjects' data together. Unlike SI, it utilizes ELR to combat noisy labels. Additionally, we denote by *SI-ELR-best* the best-performing model (based on clean validation data) saved after the training epochs. Note, in practice, we cannot decide when to stop training to obtain SI-ELR-best with truly noisy data, and thus its results are presented for reference only.
- *Butterfly [21]*: It is a joint LNL and domain adaptation method, which treats all but test subjects' data as a single source domain. It takes all unlabeled data samples from a target domain together with noisy labeled source domain data to train a model. Butterfly maintains four deep networks simultaneously, two for adaptations (i.e., noisy-to-clean, labeled-to-unlabeled, and source-to-target domains) and the remaining two for classification in the target domain.

STL, BMTL and VALERIAN are all supervised domain adaptation methods and utilize some data from the target domain. In contrast, SI and SI-ELR do not require any target domain data. Butterfly on the other hand includes unlabeled target domain data during training and thus no transfer learning is done at inference time using labeled target domain data.

4.3 Implementation and evaluation procedure

Data preparation. A standard IMU data pre-processing procedure is implemented for the experiments, including interpolation, low-pass filtering, normalization, and data segmentation. A Butterworth low-pass filter [5] with a cut-off frequency of 10Hz is employed to remove high-frequency noise from interpolated data. After low-pass filtering, we normalize the data and then segment it into sliding windows with a fixed length of 2 seconds with an 80% overlap between adjacent windows.

Implementation. The implementation of the feature extractor follows DeepConvLSTM in all models. It includes four layers of 1D CNN and two LSTM layers with 128 hidden units and a dropout rate of 0.25 to prevent over-fitting [33]. The CNN layers have 64 channels with kernel size 5 and stride 1.

For STL, the models are trained with a RMSProp optimizer [3] at a learning rate of 10^{-3} and a decay factor of $p = 0.9$. The maximum iteration number is set to be 500. The SI models are trained with 200 epochs only, as the memorization effect will gradually degrade the model performance in latter training epochs. Butterfly and ELR are trained using hyper-parameters as specified in the original papers. VALERIAN utilizes DeepConvLSTM in L_θ while the number of S_{ϕ^k} branches is determined by the number of subjects in the training data. Each S_{ϕ^k} may have a different output shape depending on the number of classes in the dataset for the corresponding subject. VALERIAN is trained with an Adam [16] optimizer at a learning rate of 10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, with hyper-parameters $\mu = 0.4$, $\alpha = 0.2$, $\beta = 0.7$, and $\lambda = 3$. The batch size is set to 64 and the number of training epochs is 300 without early stopping. The hyper-parameters and the optimizer used in each model are consistent across all datasets.

Evaluation process. In evaluating the two use cases of VALERIAN, we present the performance of label correction only on the two controlled datasets with artificially added noise as their ground truth labels are available. For domain adaptation to an unseen subject, results are presented on all four datasets.

Artificially injecting noise to clean labeled data is commonly used in evaluating LNL methods. For the controlled datasets, we consider two noise patterns with four levels each, namely, symmetric noise with $\tau = \{0.1, 0.2, 0.4, 0.6\}$ and asymmetric noise with $\tau = \{0.1, 0.2, 0.3, 0.4\}$. The noise transition matrices for asymmetric ones are then defined according to Fig. 8. From Section 2, we have seen that LNL with asymmetric noises is generally harder than that with symmetric noises. For example, when $\tau = 0.4$ and the number of classes $C = 10$ under asymmetric noise, roughly 60% of data in each class is correctly labeled while the remaining 40% is labeled to another class. As a result, the percentage difference between correctly and wrongly labeled data is only 20%. In contrast, in the symmetric noise cases, the percentage gap is $60 - \frac{40}{9} \approx 55.6\%$ (since the percentage of the wrongly labeled class is $\frac{40}{9}$). Therefore, for asymmetric noise, the maximum τ is set to 0.4 but in the case of symmetric noise, the maximum τ is set to to 0.6. In the experiments, to better simulate real-world noise patterns, the noise transition matrices of asymmetric noise are defined by setting the probability of the most similar class of each activity to τ, as shown in Fig. 8[1].

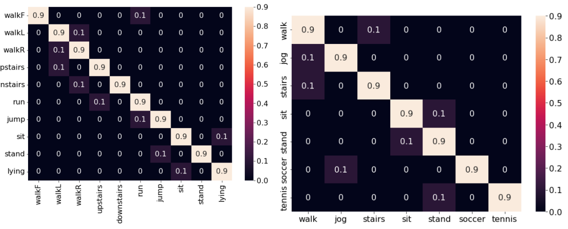


Figure 8: Noise transition matrix T with asymmetric noise for USCHAD (Left) and WISDM (Right), $\tau = 0.1$.

[1] The most similar class is determined by the confusion matrix of a model trained on clean data.

Leave-one-subject-out evaluation is conducted on all four datasets. In the experiments, we randomly select one subject as the target domain at a time, until all subjects are chosen. In Butterfly, we evaluate it in a way described in the original paper [21] and take 75% of unlabeled target samples for training and the remaining for testing. Experiments are repeated five times for each parameter setting, and the average test accuracy and its standard deviation are reported.

4.4 Results

4.4.1 Label Correction for Source Domain. In this experiment, we evaluate the label correction accuracy and the overall accuracy of source domain data using the two controlled datasets with artificially injected symmetric and asymmetric noise. Here, we consider a wrongly label sample as a positive sample and thus recall is defined as the ratio between the number of correctly predicted samples that were previously wrongly labeled and the total number of wrongly labeled samples.

Fig. 9 shows the recall rates of different approaches. It can be observed that VALERIAN outperforms all baseline methods by a large margin in all cases and can correct as high as 93% of labelling errors when the noise level is 10%. At high noise levels, e.g., 40% asymmetric noise, its performance dropped to around 75%. SI and SI-ELR have similar performance, both upper bounded by BMTL as they ignore the domain gap among training subjects. Interestingly, Butterfly performs the worst. This can be attributed to the fact that Butterfly treats data from different subjects as a single domain.

Fig. 10 shows the training accuracy for different methods. Benefiting from high recall rates of noisy data, VALERIAN achieves the best training accuracy among all. Note that the prediction errors in this setting include both mis-prediction of wrongly labeled data (i.e., memorization) and that of correctly labeled data in the training data, which can be due to the inherent limitation of the model architecture and uncorrected noisy labels.

4.4.2 Domain Adaptation with Clean Labeled Target Domain. First, we present the evaluation results on controlled datasets where clean data from unseen subjects is available. As the case of symmetric noise is simpler, we only present results from asymmetric noise due to space limit.

Overall performance. Fig. 11 and 12 show the results on USCHAD and WISDM with asymmetric noise of different levels, respectively. From these figures, we observe that VALERIAN works well and its performance is quite stable across different noise levels and types of noise in both datasets. As STL is trained entirely on clean data from $\mathcal{D}_t$, its performance is not impacted by noise patterns and levels. As more clean data become available, the performance of STL serves as an upper bound of LNL models. From the figures, we see that with 20 shots, VALERIAN has comparable or slightly worse performance than STL. However, with a smaller number of target domain data, VALERIAN learns more efficiently. For example, with five shots, the average accuracy of VALERIAN for UHSCHAD and WISDM across all noise levels and patterns are 84.35 and 83.87, respectively, which are superior than BMTL (78.71 and 78.46) and STL (75.26 and 77.20). As the noise level increases, the accuracy of VALERIAN degrades slightly as expected. However, even with 40% symmetric noise, it can achieve an average accuracy of 81.99% for USCHAD for 5-shot learning, amounting to less than 3% reduction compared to the case with 10% symmetric noise. Similar observations can be made for asymmetric noise and WISDM.

Comparison with other LNL methods. Table 2 summary the results of SI-ELR and Butterfly. For comparison, we also include the results of SI to further demonstrate that LNL methods can be harmful if applied naively. SI, SI-ELR and VALERIAN are tested with 5-shot learning while Butterfly is a unsupervised domain adaptation method, which already sees unlabeled target data during model training.

From Table 2, it is clear that none of the three methods performs well in HAR with noisy labels. The vanilla SI model does not explicitly handle subject divergence nor label noises. Its performance degrades as the noise ratio τ increases. In comparison, SI-ELR ignores subject divergence and deals with noisy labels using a regularization term. Though designed to handle label noise, SI-ELR-best has worse performance than SI when the asymmetric noise level is greater than 10%. The results are consistent with our observations with DivideMix in Section 2 and reveal that subject diversity harms ELR's ability to combat label noises. SI-ELR fares moderately better for asymmetric noise. However, with 40% noise, SI-ELR-best is 7% worse than SI and 25% worse than VALERIAN in USCHAD.

Butterfly on average has worse accuracy than SI and SI-ELR-best and performs poorly as the noise level increases in both datasets. This is in part due to the fact that Butterfly uses unlabeled target domain data at training time while SI and SI-ELR-best benefit from transfer learning with a few shots of clean labeled data at inference time. However, the difference in accessing target domain labels does not justify the large variance in Bufferfly's test accuracy on USCHAD as shown in Table 2. As an example, with 0.3 asymmetric noise, its highest test accuracy is 70.11% when subject 7 is in the test set, whereas its lowest accuracy is 13.8% for test subject 8. We believe that the poor performance of Butterfly is because it treats different subjects in the training set as a single domain.

Ablation Study. To see how each component contributes to the final performance of VALERIAN, an ablation study was conducted on the USCHAD dataset with 5-shot learning and 0.4 asymmetric noise. Similar results could be expected for other noise settings or datasets. As shown in Table. 3, the domain invariant feature learner plays the most important role in VALERIAN. Without IFLF, VALERIAN degrades to an ELR model and fails to deal with subject divergence. Moreover, in absence of a dedicated meta-learning strategy, it is insufficient to update parameters of the whole model by only a few clean labeled data samples. As a result, a large standard deviation in test accuracy is observed. MixUp contributes a ~ 6% accuracy to the overall solution, empirically demonstrating its usefulness in improving model generalization in HAR tasks with noisy labels. Inclusion of ELR in VALERIAN leads to ~ 7% improvement. Recall the poor performance of ELR alone in Table 2. The results speak unequivocally for the need to combine LNL and meta-learning to handle subject diversity. Lastly, we find that self-supervised pre-train contributes ~ 4% test accuracy.

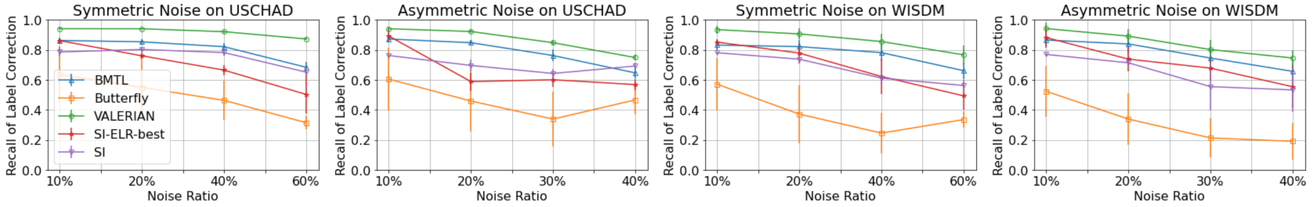


Figure 9: The recall rates of different methods on noisy training data.

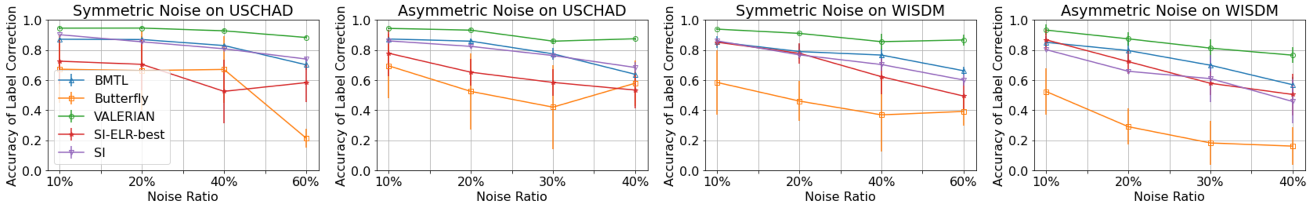


Figure 10: The training accuracy of different methods on noisy training data.

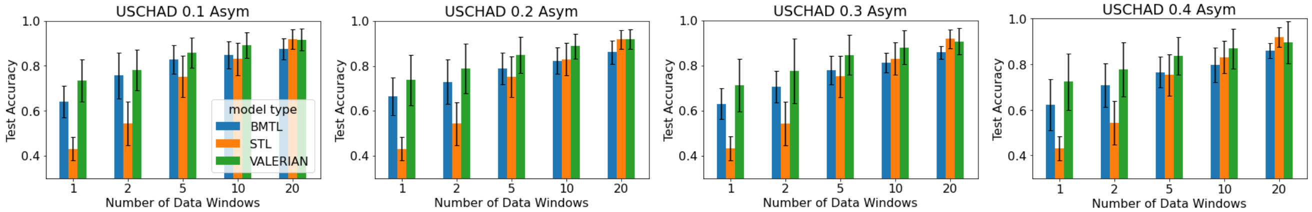


Figure 11: Evaluation on USCHAD with different levels of asymmetric noise and different numbers of data windows per activity class from $\mathcal{D}_t$. The test accuracy and standard deviation are averaged across all subjects in leave-one-out experiment.

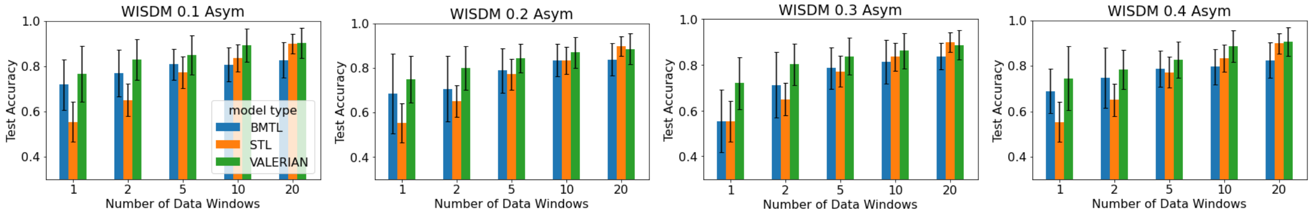


Figure 12: Evaluation on WISDM with different level of asymmetric noise and different numbers of data windows per activity class from $\mathcal{D}_t$. The test accuracy and standard deviation are averaged across all subjects in leave-one-out experiment.

4.4.3 Domain Adaptation on Noisy Labeled Target. Next, we compare the performance of VALERIAN, BMTL and STL on two noisy labeled datasets: ExtraSensory and RealWorld, which are in the wild datasets. Considering the data imbalance and class missing issue, we take F1-Score rather than accuracy as metrics to evaluate model performance here. Note that since the ground truth labels from curated data are noisy, the quantitative results need to be taken with a grain of salt. To generate t-SNE plots, we randomly selected one subject from each dataset and cleaned its labels manually.

Fig. 13(a) shows the F1-Score of the three models with gradually increasing the number of data windows on ExraSensory . Compared to results with the two controlled datasets, all methods show their worst performance. This can be attributed to the noisy target domain labels during fast adaption or learning STL model. The large standard deviation in STL results even with 20-shots indicates either label noise in target domain data or noise in ground truth or both. In fact, for many of the subjects, the training and validation set is not i.i.d due to data noise, resulting in a validation accuracy jumping back and forth between training epochs. However, VALERIAN still outperforms the other two methods in all cases tested. To see if VALERIAN can indeed learn good features from noisy data, we show in Fig. 13(c) the t-distributed stochastic neighbor embedding (t-SNE) plot of the outputs of its feature extraction network. Clearly, the classes are well separated. This is in contrast

Table 2: Results of methods that are not designed for few-shot learning on UHCHAD and WISDM, when 5 clean labeled samples per class are available from the target domain. Report with test accuracy in (%).

Method/Noise ratio	10%	20%	30%	40%
SI	74.26 ± 16.07	73.44 ± 13.57	68.66 ± 13.56	65.35 ± 14.31
SI-ELR-best	77.84 ± 15.41	70.48 ± 17.38	69.71 ± 20.43	58.49 ± 13.17
Butterfly	65.14 ± 15.25	52.44 ± 25.51	41.96 ± 28.15	37.89 ± 15.06
VALERIAN	85.71 ± 6.65	84.88 ± 8.11	84.81 ± 8.82	83.68 ± 8.18

(a) Results on USCHAD dataset with four levels of artificially added asymmetric noise patterns in data labels.

Method/Noise ratio	10%	20%	30%	40%
SI	58.66 ± 17.36	56.81 ± 15.29	53.10 ± 13.73	48.38 ± 11.36
SI-ELR-best	66.15 ± 11.46	65.23 ± 8.85	58.59 ± 11.91	55.10 ± 9.66
Butterfly	57.98 ± 14.66	36.75 ± 13.23	24.47 ± 15.36	14.30 ± 1.43
VALERIAN	84.85 ± 8.73	84.41 ± 6.53	83.71 ± 8.01	82.63 ± 7.98

(b) Results on WISDM dataset with four levels of artificially added asymmetric noise patterns in data labels.

Table 3: Ablation study of VALERIAN on USCHAD with 0.4 asymmetric noise.

Method	**Test Accuracy**
VALERIAN	83.68 ± 8.18
VALERIAN w/o ELR	76.55 ± 6.69
VALERIAN w/o self-supervised pre-train	79.28 ± 7.37
VALERIAN w/o MixUp	77.69 ± 2.34
VALERIAN w/o IFLF	60.18 ± 10.35

with overlapping among classes in 13(b), which shows the t-SNE plot of the outputs from the feature extraction network in BMTL.

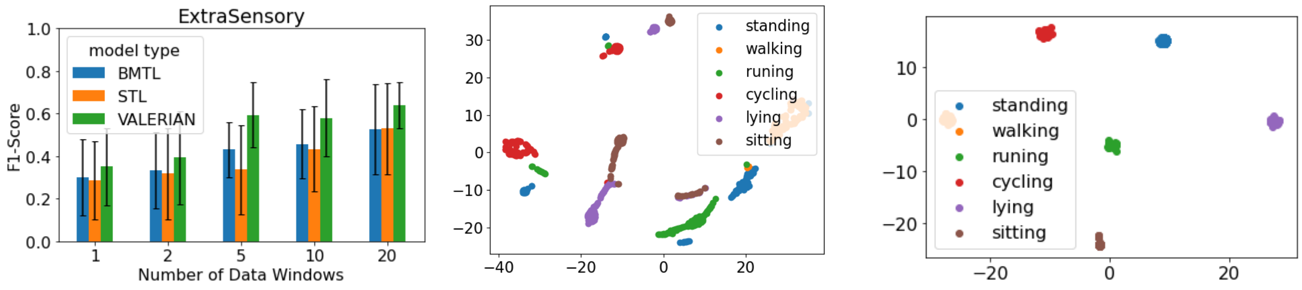
Similar observations can be made for RealWorld. With BMTL, the t-SNE plot in Figure 14(b) shows closeness (and thus likely mislabeling) between running and walking, standing and climbing up, climbing down and walking activities. In comparison, clusters generated by VALERIAN are better separated.

5 RELATED WORK

Learning with noisy labels. LNL has been investigated in computer vision and audio signal processing for over a decade [10, 32]. Existing methods can be categorized into three groups. First, contrastive learning-based LNL methods [45, 47] add regularization terms to the loss function to obtain a well-clustered feature structure. Second, curriculum learning [4, 23] or teacher-student networks such as MentorNet [15] trains a neural network to guide a student network by assigning weights to samples. Since the pioneer co-teaching work [11], the use of two networks together gains popularity in LNL and has been adopted in several recent papers including DivideMix [20], ELR+ [22], co-regularization [40]). Instead of training a model that works on the noisy labeled samples, another line of work aims to select clean labeled samples out of noisy ones [24, 51]. Despite all the advancements in LNL, none of the afore-mentioned work considers domain gaps between source and target domains (also known as domain shifts).

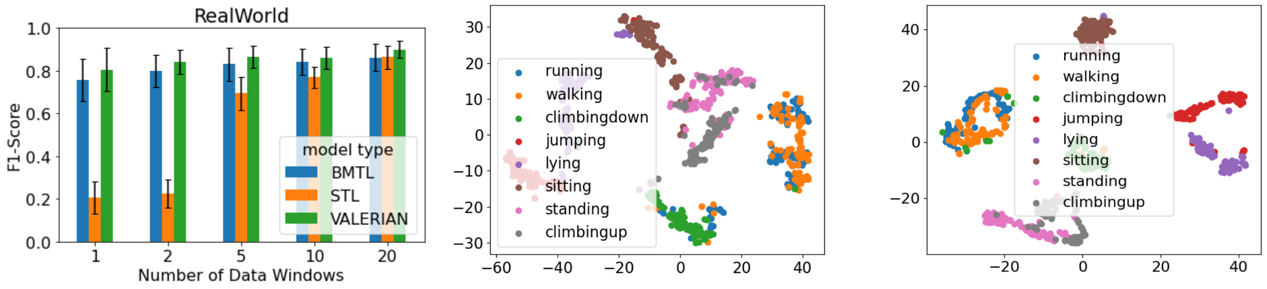
Weakly-supervised learning in sensor-based HAR. There are some works in mobile computing that deal with weakly-supervised learning problems related to sensor-based HAR [13, 38, 39]. Wang *et al.* in [38, 39] define weakly-supervised learning as detecting the start and end of an activity of interest in a given time-series data sequence, similar to the sound event detection problem[1, 8, 19]. Unlike our problem, the goal is to crop the data of interest from a noisy sequence for training so that a machine learning model can gain a better discriminative power. For instance, consider a collected *climbing up* IMU data trial with two activities: climbing upstairs and walking on the flat ground. Wang *et al.* treat walking as a background activity and try to detect the onset and offset timestamps of climbing upstairs events. In contrast, in this work, we treat the data within such a trial as a mixture of *climbing up* and noisy labeled *walking* activities. Apart from the different ways of treating label noises, existing works still require further steps to handle subject diversity within the training process to generalize well to new unseen subjects.

Joint LNL and domain adaptation. A few works consider LNL together with domain shifts. Shu *et al.* in [31] considered noise either in data or label of a single source domain and perform weakly-supervised model training to adapt to a target domain. In [21, 48] researchers propose one-step solutions to LNL and unsupervised domain adaptation. However, these methods have been applied to image classification tasks, where there is only a single source domain. Thus, the authors only consider the domain shift between one source domain and one target domain. In contrast, in our work, we need to take into account domain shifts amongst multiple source domains, namely, different human subjects. As discussed in Section 2, subject diversity in training data prevents conventional LNL methods from working effectively since early learning can inadvertently memorize noisy data.



(a) F1-Score. **(b) t-SNE plot of features in BMTL.** **(c) t-SNE plot of features in VALERIAN.**

Figure 13: Evaluation on ExtraSensory with different the number of data windows per activity class from $\mathcal{D}_t$. The mean and standard deviation F1-Scores are averages across all subjects in leave-one-out experiment. t-SNE are generated on a random subject (id:4FC32141-E888-4BFF-8804-12559A491D8C) with data from all six classes.



(a) F1-Score. **(b) t-SNE plot of features in BMTL.** **(c) t-SNE plot of features in VALERIAN.**

Figure 14: Evaluation on RealWorld with different the number of data windows per activity class from $\mathcal{D}_t$. The mean and standard deviation F1-Scores are averages across all subjects in leave-one-out experiment. t-SNE are generated on a random subject (id:3) with data from all eight classes.

6 CONCLUSION

In this paper, we proposed VALERIAN, a domain invariant feature learning approach for IMU sensor-based HAR in the wild. An extensive experimental study demonstrated its superior performance over baseline methods for different levels of noise and noise patterns, and in two use scenarios. The key takeaway from this work is two-fold: 1) the effects of subject diversity and label noises intertwine in the learning behaviour of LNL models and can lead to catastrophic memorization of wrongly labelled data, and 2) it is important to design domain adaptation strategies to explicitly handle subject diversity in conjunction with LNL for better generalization in HAR.

It is plausible to apply VALERIAN to other sensor data modalities as long as there exist significant subject divergence and performance drop due to label noises. Components of VALERIAN (e.g, self-supervised learning, early loss regularization) can be replaced by other more advanced approaches though the framework remains applicable. Also orthogonal to the proposed approach are unsupervised domain adaption methods and domain generalization methods. One limitation of VALERIAN is the need of correctly labeled samples from a target domain for domain adaptation to achieve a higher inference accuracy. One interesting area of further investigation is to perform domain adaptation with noisy labeled target only. Finally, we believe significant efforts should be made to build in-the-wild datasets and benchmarks for IMU-based HAR.

REFERENCES

[1] Sharath Adavanne, Haytham Fayek, and Vladimir Tourbabin. 2019. Sound event classification and detection with weakly labeled data. (2019).

[2] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *International conference on machine learning*. PMLR, 233–242.

[3] Yoshua Bengio. 2015. Rmsprop and equilibrated adaptive learning rates for nonconvex optimization. *corr abs/1502.04390* (2015).

[4] Stefan Braun, Daniel Neil, and Shih-Chii Liu. 2017. A curriculum learning method for improved noise robustness in automatic speech recognition. In *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 548–552.

[5] Stephen Butterworth et al. 1930. On the theory of filter amplifiers. *Wireless Engineer* 7, 6 (1930), 536–541.

[6] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. 2021. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM Computing Surveys (CSUR)* 54, 4 (2021), 1–40.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[8] Heinrich Dinkel, Mengyue Wu, and Kai Yu. 2021. Towards duration robust weakly supervised sound event detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021), 887–900.

[9] Carl Doersch and Andrew Zisserman. 2017. Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision*. 2051–2060.

[10] Bo Han, Quanming Yao, Tongliang Liu, Gang Niu, Ivor W Tsang, James T Kwok, and Masashi Sugiyama. 2020. A survey of label-noise representation learning: Past, present and future. *arXiv preprint arXiv:2011.04406* (2020).

[11] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems* 31 (2018).

[12] Yujiao Hao, Rong Zheng, and Boyu Wang. 2021. Invariant Feature Learning for Sensor-based Human Activity Recognition. *IEEE Transactions on Mobile Computing* (2021).

[13] Jun He, Qian Zhang, Liqun Wang, and Ling Pei. 2018. Weakly supervised human activity recognition from wearable sensors by recurrent attention learning. *IEEE Sensors Journal* 19, 6 (2018), 2287–2297.

[14] Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. 2020. Beyond synthetic noise: Deep learning on controlled noisy labels. In *International Conference on Machine Learning*. PMLR, 4804–4815.

[15] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*. PMLR, 2304–2313.

[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[17] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114, 13 (2017), 3521–3526.

[18] Abhishek Kumar and Hal Daume III. 2012. Learning task grouping and overlap in multi-task learning. *arXiv preprint arXiv:1206.6417* (2012).

[19] Anurag Kumar and Bhiksha Raj. 2016. Audio event detection using weakly labeled data. In *Proceedings of the 24th ACM international conference on Multimedia*. 1038–1047.

[20] Junnan Li, Richard Socher, and Steven CH Hoi. 2019. DivideMix: Learning with Noisy Labels as Semi-supervised Learning. In *International Conference on Learning Representations*.

[21] F. Liu, J. Lu, B. Han, G. Niu, G. Zhang, and M. Sugiyama. 2019. Butterfly: A Panacea for All Difficulties in Wildly Unsupervised Domain Adaptation. In *NeurIPS LTS Workshop*.

[22] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. 2020. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems* 33 (2020), 20331–20342.

[23] Yueming Lyu and Ivor W Tsang. 2019. Curriculum loss: Robust learning and generalization against label corruption. *arXiv preprint arXiv:1905.10045* (2019).

[24] Curtis Northcutt, Lu Jiang, and Isaac Chuang. 2021. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research* 70 (2021), 1373–1411.

[25] Henry Friday Nweke, Ying Wah Teh, Mohammed Ali Al-Garadi, and Uzoma Rita Alo. 2018. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Systems with Applications* 105 (2018), 233–261.

[26] Francisco Ordóñez and Daniel Roggen. 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* 16, 1 (2016), 115.

[27] Steve Ramirez, Xu Liu, Pei-Ann Lin, Junghyup Suh, Michele Pignatelli, Roger L Redondo, Tomás J Ryan, and Susumu Tonegawa. 2013. Creating a false memory in the hippocampus. *Science* 341, 6144 (2013), 387–391.

[28] 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.

[29] Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017).

[30] Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. 2019. Multi-task self-supervised learning for human activity detection. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 2 (2019), 1–30.

[31] Yang Shu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. 2019. Transferable curriculum for weakly-supervised domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4951–4958.

[32] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[33] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[34] Timo Sztyler. 2019. *Sensor-based human activity recognition: Overcoming issues in a real world setting*. Ph. D. Dissertation. Mannheim, Germany. http://ub-madoc.bib.uni-mannheim.de/49914/.

[35] Yonatan Vaizman, Katherine Ellis, and Gert Lanckriet. 2017. Recognizing detailed human context in the wild from smartphones and smartwatches. *IEEE pervasive computing* 16, 4 (2017), 62–74.

[36] Yonatan Vaizman, Katherine Ellis, and Gert Lanckriet. 2017. Recognizing detailed human context in the wild from smartphones and smartwatches. *IEEE pervasive computing* 16, 4 (2017), 62–74.

[37] 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.

[38] Kun Wang, Jun He, and Lei Zhang. 2019. Attention-based convolutional neural network for weakly labeled human activities' recognition with wearable sensors. *IEEE Sensors Journal* 19, 17 (2019), 7598–7604.

[39] Kun Wang, Jun He, and Lei Zhang. 2021. Sequential weakly labeled multiactivity localization and recognition on wearable sensors using recurrent attention networks. *IEEE Transactions on Human-Machine Systems* 51, 4 (2021), 355–364.

[40] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. 2020. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13726–13735.

[41] Gary M Weiss, Kenichi Yoneda, and Thaier Hayajneh. 2019. Smartphone and Smartwatch-Based Biometrics Using Activities of Daily Living. *IEEE Access* 7 (2019), 133190–133202.

[42] Kieran Woodward, Eiman Kanjo, Andreas Oikonomou, and Alan Chamberlain. 2020. LabelSens: enabling real-time sensor data labelling at the point of collection using an artificial intelligence-based approach. *Personal and Ubiquitous Computing* 24, 5 (2020), 709–722.

[43] Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. 2020. Robust early-learning: Hindering the memorization of noisy labels. In *International conference on learning representations*.

[44] 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.

[45] Mouxing Yang, Yunfan Li, Zhenyu Huang, Zitao Liu, Peng Hu, and Xi Peng. 2021. Partially view-aligned representation learning with noise-robust contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1134–1143.

[46] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek Abdelzaher. 2017. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the 26th International Conference on World Wide Web*. 351–360.

[47] Li Yi, Sheng Liu, Qi She, A Ian McLeod, and Boyu Wang. 2022. On Learning Contrastive Representations for Learning with Noisy Labels. *arXiv preprint arXiv:2203.01785* (2022).

[48] Xiyu Yu, Tongliang Liu, Mingming Gong, Kun Zhang, Kayhan Batmanghelich, and Dacheng Tao. 2020. Label-noise robust domain adaptation. In *International Conference on Machine Learning*. PMLR, 10913–10924.

[49] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017).

[50] 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.

[51] Zizhao Zhang, Han Zhang, Sercan O Arik, Honglak Lee, and Tomas Pfister. 2020. Distilling effective supervision from severe label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9294–9303.