ContrastSense: Domain-invariant Contrastive Learning for In-the-wild Wearable Sensing

GAOLE DAI, Nanyang Technological University, Singapore
HUATAO XU, The Hong Kong University of Science and Technology, Hong Kong
HYUNGJUN YOON, KAIST, Republic of Korea
MO LI*, The Hong Kong University of Science and Technology, Hong Kong
RUI TAN*, Nanyang Technological University, Singapore
SUNG-JU LEE, KAIST, Republic of Korea

Existing wearable sensing models often struggle with domain shifts and class label scarcity. Contrastive learning is a promising technique to address class label scarcity, which however captures domain-related features and suffers from low-quality negatives. To address both problems, we propose ContrastSense, a domain-invariant contrastive learning scheme for a realistic wearable sensing scenario where domain shifts and class label scarcity are presented simultaneously. To capture domain-invariant information, ContrastSense exploits unlabeled data and domain labels specifying user IDs or devices to minimize the discrepancy across domains. To improve the quality of negatives, time and domain labels are leveraged to select samples and refine negatives. In addition, ContrastSense designs a parameter-wise penalty to preserve domain-invariant knowledge during fine-tuning to further maintain model robustness. Extensive experiments show that ContrastSense outperforms the state-of-the-art baselines by 8.9% on human activity recognition with inertial measurement units and 5.6% on gesture recognition with electromyography when presented with domain shifts across users. Besides, when presented with different kinds of domain shifts across devices, on-body positions, and datasets, ContrastSense achieves consistent improvements compared with the best baselines.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing systems and tools**.

Additional Key Words and Phrases: Wearable Sensing, Contrastive Learning, Domain Generalization

ACM Reference Format:
Gaole Dai, Huatao Xu, Hyungjun Yoon, Mo Li, Rui Tan, and Sung-Ju Lee. 2024. ContrastSense: Domain-invariant Contrastive Learning for In-the-wild Wearable Sensing. In *Proceedings of Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*. ACM, New York, NY, USA, 32 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

The increasing prevalence of wearable sensing devices, such as smartwatches, smart glasses, activity trackers, augmented/virtual reality headsets, etc., has facilitated the ubiquitous collection of human data, giving rise to

*Corresponding authors

Authors' Contact Information: Gaole Dai, GAOLE001@e.ntu.edu.sg, Nanyang Technological University, Singapore; Huatao Xu, huatao@ust.hk, The Hong Kong University of Science and Technology, Hong Kong; Hyungjun Yoon, hyungjun.yoon@kaist.ac.kr, KAIST, Republic of Korea; Mo Li, lim@ust.hk, The Hong Kong University of Science and Technology, Hong Kong; Rui Tan, tanrui@ntu.edu.sg, Nanyang Technological University, Singapore; Sung-Ju Lee, profsj@kaist.ac.kr, KAIST, Republic of Korea.

Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., Vol. 8, No. 4, Article 0.

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

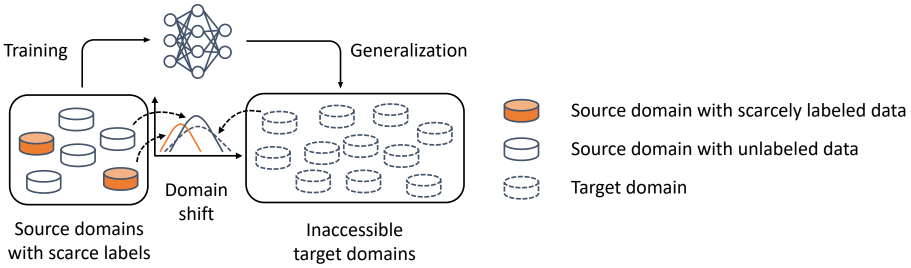

Training
Generalization
Domain shift
Source domains with scarce labels
Inaccessible target domains
Source domain with scarcely labeled data
Source domain with unlabeled data
Target domain

Fig. 1. In-the-wild wearable sensing with domain shifts and scarce class labels. Each cylinder refers to a domain, such as one user, device, on-body position when wearing the device.

numerous sensing tasks such as human activity recognition and tracking. These sensing tasks have significantly contributed to applications such as intelligent healthcare, smart home systems, and human-computer interaction [27, 32, 57, 68]. To unleash the full potential of wearable sensing data, deep learning techniques have been widely adopted, leading to notable advancements in performance [12, 29, 66].

The adoption of wearable sensing in real-world scenarios, however, still faces a major challenge since the wearable sensing data collected from different domains, i.e., different users, devices, on-body positions, or datasets, are heterogeneous [11]. Domain shifts, i.e., the data distribution difference across domains in such a case, would lead to performance degradation when applying learning models across domains [10, 25]. The problem is further challenged by the scarcity of class labels, as annotating wearable sensing data could be expensive and time-consuming [48, 52, 61]. As a result, class labels are generally only available in a few domains and their amount is limited, which leads to insufficient training of existing supervised learning models [39, 65].

As depicted in Fig. 1, this paper considers an in-the-wild scenario for wearable sensing: there exist many domains concerning different users, devices, on-body positions, or datasets. The data collection overhead and privacy concerns restrict access to some domains, which are called the *source domains*. In the source domains, most data are unlabeled and only few labeled data are available. On the other hand, all other domains are inaccessible during the learning phase, which are called the *target domains*. The objective is to transfer wearable sensing models trained from *source domains* to the *target domains* with high performance. This paper focuses on classification tasks with wearable sensors in this scenario.

Most existing deep-learning approaches are ill-suited for such an in-the-wild wearable sensing scenario. Recent domain adaptation techniques aim at adapting features from source domains to target domains [2, 8, 14, 63, 64]. Nevertheless, most existing solutions require labeled or unlabeled data from target domains, which are not accessible in the considered scenario in Fig. 1 due to privacy concerns [42]. Without requiring any target domain data, domain generalization learns domain-invariant features among source domains, which are assumed invariant among target domains as well [31, 35, 42]. However, most existing domain generalization approaches rely on abundant labeled data in the source domains for supervised model training, which is not available considering the expensive annotation process [65].

To handle class label scarcity, self-supervised learning methods have gained research attention [15, 65], as they can utilize unlabeled data to capture general representations. In particular, Contrastive Learning (CL) has achieved good performance in various fields [16, 38, 39]. In CL, data that are augmented from the same samples are positives, while data from different samples are negatives. The model would learn high-level features by discriminating positives from negatives. Later, the encoder is fine-tuned for downstream tasks with limited labeled data. However, applying CL directly to in-the-wild wearable sensing poses two challenges. The first challenge is caused by domain shifts. As CL typically ignores the domain shifts, the model often has degraded performance when applied to the target domains. The second challenge is the *adjacent negatives* problem. Considering the continuity of human activities or status, adjacent samples might share high similarities and be from the same class. However, in the original setting of CL, these adjacent samples are treated as negatives. As a result, their features are repealed from each other, leading to less effective features [58].

To address the above two challenges, we identify two unique opportunities inherent to wearable sensor data that can be effectively utilized. While class labels require significant annotation efforts, domain labels that specify the user ID or device type of the collected data and time labels that record when the data samples are collected can be easily obtained. The source of the data can be acquired during data collection with proper anonymity, whereas timestamps are commonly available along with sensor readings.

By exploiting both opportunities, this paper proposes a novel wearable sensing framework, ContrastSense, which trains models effectively to address the domain shifts and adjacent negatives problems. Three novel components are designed and integrated: (i) Domain labels are utilized to derive Contrastive Domain Loss (CDL) that measures the similarity of features from different domains. By maximizing CDL, the encoder is trained to minimize the discrepancy between domains thus extracting generalizable features. (ii) A novel contrastive loss, SInfo loss, is proposed with negative selection. The SInfo loss only selects samples outside a nearby time window using time labels, avoiding the inclusion of adjacent negatives from the same class. Additionally, easy-to-discriminate samples from different domains are excluded, minimizing the risk of learning domain-related features. (iii) During fine-tuning, a parameter-wise penalty is proposed to constrain the training of the encoder to maintain generalizability.

Extensive experiments on two kinds of sensing modalities and tasks are performed, including human activity recognition with inertial measurement units (IMU) and gesture recognition with electromyography (EMG). The results suggest that when presented with domain shifts across users, ContrastSense outperforms state-of-the-art models by 8.9% and 5.6% average F1 scores on the two tasks, respectively. Additionally, ContrastSense is evaluated on different kinds of settings and domain shifts, including devices, on-body positions, and datasets, and consistently outperforms the best baselines. When presented with multiple domain shifts simultaneously across different datasets, ContrastSense outperforms the best baseline by 9.0% and 4.3% average F1 scores on the two tasks, respectively. The main contributions of this paper are summarized as follows:

(1) This paper considers a novel and realistic wearable sensing scenario where domain shifts and class label scarcity are presented simultaneously. To the best of our knowledge, this is the first work that studies domain generalization with class label scarcity for in-the-wild wearable sensing applications.
(2) This paper proposes a general learning framework for different wearable sensors, ContrastSense, that leverages domain and time labels to achieve domain-invariant CL for generalizable features across domains with practical overhead.
(3) The framework is evaluated on different kinds of wearable sensors, tasks, and domains. The results suggest that ContrastSense outperforms the state-of-the-art baselines. The code is available at https://github.com/MaginaDai/ContrastSense-Public.

The rest of the paper is organized as follows. Section 2 presents related works. Section 3 defines the problem and motivates this study with experimental evidence. Section 4 presents the preliminary of CL and the framework of ContrastSense. Section 5 details the designs of ContrastSense. Section 6 presents the experiment results. Section 7 discusses the limitations and future works. Section 8 concludes this paper.

2 Related Works

2.1 Domain Adaptation and Generalization

Domain adaptation refers to the process of adapting models trained on the source domains to the target domains, using labeled or unlabeled data from the target domains for training [2, 14, 63, 64]. Specifically, CMUDA [2] conducts domain adversarial training on source and target domains to align their features. However, those target domain data are unavailable in our proposed scenario. Different from domain adaptation, domain generalization methods focus on learning domain-invariant features only using data from the source domains. To achieve this, data augmentations, domain-invariant learning, and meta-learning techniques have been extensively studied [30, 35, 41, 67]. However, most existing methods assume that the source domains contain sufficient high-quality labeled data, which may be unrealistic for in-the-wild wearable sensing [39, 48]. ContrastSense is able to extract domain-invariant features primarily with unlabeled data from the source domains, distinguishing it from existing domain generalization methods.

2.2 Self-supervised Learning

Self-supervised learning represents a category of training methods that generate labels for unlabeled data autonomously to overcome class label scarcity [33]. It mainly includes Generative Learning and Contrastive Learning (CL). Generative Learning focuses on reconstructing missing parts of the unlabeled data [33, 65]. For example, LIMU-BERT [65] designs a lightweight BERT-like model to reconstruct masked IMU data for temporal feature extraction. CL augments samples into positive-negative pairs, and the models are trained to distinguish positives from negatives. Several existing CL frameworks, such as SimCLR [3], MoCo [16], and SimSiam [4], have achieved state-of-the-art performance on various tasks. However, existing self-supervised learning methods generally do not account for domain shifts, which can lead to significant performance degradation when applied to unseen domains. This paper aims to address the limitations of CL and apply it to in-the-wild wearable sensing.

2.3 Contrastive Learning for Wearable Sensing

CL has also been leveraged to address class label scarcity in wearable sensing [15, 21, 24, 39]. For example, Cosmo [39] captures common and complementary features from different modalities via contrastive fusion learning. CPCHAR [15] conducts contrastive pretraining on the unlabeled data by predicting the nearby features. Additionally, some works focus on negative selection in CL for wearable sensing [21, 58]. Notably, ColloSSL [21] selects negative devices and samples based on a sampling algorithm to calculate a multi-view contrastive loss. However, these approaches still encounter significant challenges, primarily due to domain shifts and the selection of adjacent negatives, when applied to in-the-wild wearable sensing.

2.4 Summary of Existing Works

As summarized in Table 1, existing domain adaptation and generalization approaches like [2, 41, 67] can extract generalized features from labeled data. However, the class label scarcity may cause the sampled distribution to deviate from the actual domain distribution, resulting in less representative domain-invariant features. Self-supervised learning approaches like CPCHAR [15] or LIMU-BERT [65] can mitigate the impact of class label scarcity. Nonetheless, assuming the unlabeled data is independent and identically distributed (i.i.d.), they overlook the domain shifts and therefore cannot extract generalizable features across domains. Besides, many approaches require access to target domain data during training, which is impractical in contexts where data privacy is a concern. To the best of our knowledge, the proposed framework, ContrastSense, is the first work that studies domain generalization with class label scarcity for in-the-wild wearable sensing applications.

3 Motivation

This section formally defines the problem and then presents the limitations of existing domain-invariant and self-supervised learning with experimental evidence. The Heterogeneity Human Activity Recognition (HHAR) dataset [50] is used, which consists of data collected from nine users and three types of smartphones.

Table 1. Limitations of existing works and the contributions of ContrastSense.

Methods	Address Domain Shifts	Address Class Label Scarcity	Target Domain Inaccessibility	Evaluated across Sensing Modalities
LIMU-BERT [65]	✗	✓	✗	✗
MoCoHAR [59]	✗	✓	✗	✗
CPCHAR [15]	✗	✓	✓	✗
ClusterHAR [58]	✗	✓	✓	✗
ColloSSL [21]	✗	✓	✓	✗
ConSSL [28]	✗	✓	✓	✗
FMUDA/CMUDA [2]	✓	✗	✗	✗
GILE [41]	✓	✗	✓	✗
CALDA [63]	✓	✗	✗	✓
Mixup [67]	✓	✗	✓	✓
ContrastSense	✓	✓	✓	✓

3.1 Problem Statement

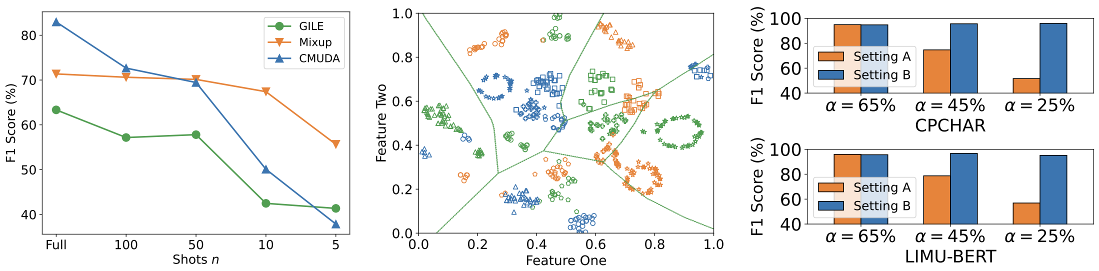
Among a large amount of domains, only some domains are accessible for training due to the data collection overhead and privacy concerns. Such accessible domains form the source domains set $\mathcal{D}_S$, whereas the other unavailable domains form the target domains set $\mathcal{D}_T$. The percentage of the domains in $\mathcal{D}_S$ among all domains is α. Given the expense for labeling data, only some domains within $\mathcal{D}_S$ are labeled for training, forming the *labeled source domains* set $\mathcal{D}_{LS}$. The percentage of the domains in $\mathcal{D}_{LS}$ among $\mathcal{D}_S$ is β. $\mathcal{D}_{LS}$ contains only n shots of labeled samples, e.g., 10 shots in total. The term n shots refers to that there are n samples for each class. The remaining domains within $\mathcal{D}_S$ contain unlabeled data and form *unlabeled source domains* set $\mathcal{D}_{US}$. The objective is to effectively utilize both the labeled and unlabeled data in $\mathcal{D}_{LS}$ and $\mathcal{D}_{US}$ to train sensing models that could generalize to $\mathcal{D}_T$ when the data from $\mathcal{D}_T$ are not accessible for training.

3.2 Impact of Class Label Scarcity

The IMU data collected from different users, on-body positions, or devices exhibit domain shifts. Variations in motion speeds and strength levels among users, such as the slower running pace of the elderly when compared with the young, contribute to such domain shifts. The use of different devices and different wearing positions may also result in varying levels of accuracy, sensitivity, and selection rate, leading to distribution discrepancies [11, 50]. In this section, we focus on data heterogeneity across users as an example.

While domain adaptation and generalization approaches may alleviate the impact of domain shifts [35, 72], most of them suffer from class label scarcity of in-the-wild wearable sensing. To investigate the impact of class label scarcity, we perform an experiment with setting A, where $\alpha = 25\%$ domains (users) are randomly selected as $\mathcal{D}_S$ and $\beta = 50\%$ source domains are labeled. Three domain adaptation and generalization methods are evaluated, i.e., CMUDA [2], GILE [41], and Mixup [67]. Please see Section 6.2 for a detailed introduction to the methods.

Fig. 2(a) shows that, compared with training with fully labeled data, the performance of all three models significantly drops when n is small. The performance of GILE and CMUDA degrades more notably as both were designed with the assumption that the source domains have sufficient labels, and they learn the domain-invariant features in a supervised or semi-supervised way. The performance of Mixup drops by 14% when n is smaller than 50 since class label scarcity makes the data synthesizing less effective. The above results show the significant impact of the scarcity of class labels on existing domain adaptation and generalization approaches.



(a) Impact of class label scarcity.

(b) Visualization of domain shifts.

(c) Impact of domain shifts.

Fig. 2. Motivation study. (a) The performances of the three models drop with the absence of fully labeled data for supervision. (b) Features from three users are visualized in different colors. Besides, features from different classes are presented in different shapes. User 1 (green) is labeled while users 2 (blue) and 3 (orange) are unlabeled. The green lines are the decision boundary of the classifier trained with the labeled data of user 1. (c) The performances of CPCHAR [15] and LIMU-BERT [65] degrade when presented with a larger domain shift.

3.3 Impact of Domain Shifts

Though existing self-supervised learning can handle class label scarcity, they might capture the domain-related features during training, leading to performance degradation when presented with domain shifts. The hidden feature from the auto-regressor of a CL method, CPCHAR [15], is visualized in the 2D plane using t-distributed stochastic neighbor embedding (t-SNE) [54]. Fig. 2(b) plots the features of three users with different colors, where users 1 and 2 are from source domains and user 3 is from target domains. While the classifier can accurately classify the features of user 1 (green color), it fails to classify the features of user 2 and user 3 due to the domain shifts across different users. Therefore, the distribution shifts among different domains may impair the performance of self-supervised learning when transferred from the source to the target domains.

To further investigate the impact of domain shifts on self-supervised learning, two different experiment settings are examined: (i) In setting A, $\alpha\%$ is varied and n is fixed to 50, which is to present the label scarcity and domain shifts simultaneously; (ii) In setting B, unlabeled data and 50 shots of labeled data are randomly extracted from all domains (including $\mathcal{D}_S$ and $\mathcal{D}_T$) for fine-tuning, which only present the models with label scarcity. The number of unlabeled data is the same under the two settings. Fig. 2(c) shows the performance of two state-of-the-art self-supervised learning methods, CPCHAR [15] and LIMU-BERT [65] with the above two settings. When presented with limited shots but no domain shifts in setting B, CPCHAR and LIMU-BERT achieve high performance since they have access to both source and target domain data. This indicates their ability to handle label scarcity. However, both learning methods suffer from large performance drops in setting A as α decreases due to the impact of domain shifts. We have also conducted experiments with $n = 10$ and $n = 100$, and similar trends have been observed. For example, when $n = 10$ ($n = 100$) in setting A, the F1 scores of CPCHAR decrease from 80.83% (95.25%) when $\alpha = 65\%$ to 54.19% (53.37%) when $\alpha = 25\%$. The results show that domain shifts significantly affect the model performance. Specifically, when fewer domains are available for training, more performance degradation is observed when applied to target domains.

The above experimental evidence shows the limitations of domain adaptation and generalization methods on handling label scarcity and the limitations of existing self-supervised learning methods on handling domain shifts, which motivates our work, ContrastSense.

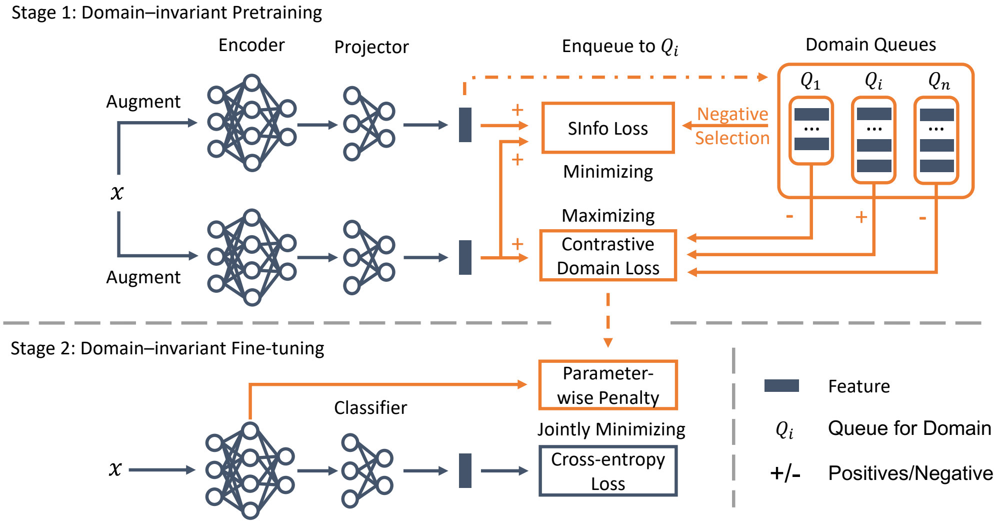


Fig. 3. ContrastSense framework. The left part (blue color) is the general CL procedure while the right part (orange color), including Contrastive Domain Loss, SInfo Loss with Negative Selection, and Parameter-wise Penalty, presents the key components in ContrastSense.

4 From Contrastive Learning to ContrastSense

4.1 Contrastive Learning and Its Limitations

To deal with the class label scarcity problem, this paper adopts Contrastive Learning (CL), which can extract high-level features from unlabeled data. CL involves pretraining and fine-tuning stages, as shown in Fig. 3 (the left part). During pretraining, data augmentations are applied to transform each data sample into two views, which are positives to each other, while other views from different data samples are negatives. Two branches of encoders and projectors extract features from these views, which are then used to discriminate between positives and negatives. To optimize the process, InfoNCE loss [38] is widely adopted:

$$L_{InfoNCE} = -\sum_{k=1}^{K} \log \frac{\exp(z_k \cdot z_{q+}/\tau)}{\sum_{q=1}^{2K} I_{[q\neq k]} \exp(z_k \cdot z_q/\tau)}, \tag{1}$$

where K is the batch size. There are $2K$ features in total since each sample is augmented to two views and then encoded to features. The z_{q+} and z_k are positive features, while the other $2K-2$ features are negatives to z_k. The $I_{[q\neq k]}$ is an indicator function, which is equal to 1 only when $q \neq k$. Otherwise, it is equal to 0. The τ is a temperature coefficient. Through minimizing $L_{InfoNCE}$, the model learns to pull z_{q+} and z_k closer and push the negatives away in the feature space. If the negatives are from categories that are different from the positives, the model would learn the difference between categories, which is beneficial to activity recognition.

During fine-tuning, the high-level features learned by positives-negatives discrimination are specialized with a classifier for wearable sensing tasks. The cross-entropy loss can be adopted to optimize the process. CL has achieved superior performance in many fields, such as computer vision and natural language processing [3, 16, 43, 46], and has been applied to wearable sensing [15, 21, 39, 61]. Despite its effectiveness, however, CL may suffer from certain limitations when applied to wearable sensing.

Limitations of Contrastive Learning in Wearable Sensing. First, CL may learn domain-specific features from sensor data, which could undermine its generalizability [69]. When negatives are sampled from multiple source domains $\mathcal{D}_S$, the encoder may learn to use domain shifts to distinguish positives from negatives of other domains during pretraining. For instance, data collected from different users with distinct walking frequencies may exhibit significant domain discrepancies. Consequently, the model could rely on these domain-specific characteristics to improve positives-negatives classification. However, such knowledge might not be applicable to target domains without such characteristics. Even if the pretrained model is robust to domain shifts with some designs, they might forget the knowledge during fine-tuning. The reason is that the scarcely labeled data are from limited labeled source domains $\mathcal{D}_{LS}$. When they are used to fine-tune the model, they would cause the model to overfit to $\mathcal{D}_{LS}$, and again, learn some domain-specific features.

Second, CL randomly samples sensor data and considers all views augmented from different data samples as negatives, which could include some negatives that are adjacent to the positives and share a high similarity. This inclusion of adjacent negatives might hinder the model from learning high-quality features. For example, when a data sample x_t at time t is sampled along with some adjacent samples, such as x_{t-1} and x_{t+1}, they are treated as negatives to x_t in CL. However, in the context of wearable sensing, human activities and status are consecutive and can persist for short or long periods [1, 60], making adjacent frames more likely to belong to the same class. As a result, their features should be clustered rather than pushed away for better downstream task performance [5]. Therefore, it is essential to exclude these adjacent negatives from the CL process.

4.2 Overview of ContrastSense

To simultaneously handle class label scarcity and domain shifts, ContrastSense includes three key designs as depicted in Fig. 3 (the right part) to overcome the limitations of CL and make CL domain-invariant.

1. **Contrastive Domain Loss (CDL).** Based on the domain labels, CDL treats features from the same (different) domains as positives (negatives). In this way, models are driven by CDL to discriminate features by domains and extract the discrepancies between domains. CDL is then integrated with CL and maximized to extract domain-invariant features. Moreover, to improve the effectiveness of features, CDL utilizes samples from *domain queues*, which is a data structure storing a large number of features from each domain with acceptable memory usage (Section 5.1).
2. **SInfo Loss with Negative Selection.** SInfo loss exploits domain and time labels and carefully selects negative samples from domain queues to improve the robustness of features and avoid adjacent negatives. Time labels are leveraged to select non-nearby samples as negatives. Besides, SInfo loss discards some that are easy to discriminate to avoid capturing domain-related features using domain labels (Section 5.2).
3. **Parameter-wise Penalty.** A parameter-wise penalty is incorporated into the fine-tuning based on the importance of each parameter, which serves as a constraint to fine-tune the model. In this way, more important parameters to domain-invariant features would be tuned less to preserve the domain-invariant knowledge (Section 5.3).

The above three components are integrated into the two stages in Fig. 3. During the domain-invariant pretraining, SInfo loss is minimized to extract high-level features by discriminating positives and selected negatives, whereas CDL is maximized to drive those features domain-invariant. During the domain-invariant fine-tuning, the parameter-wise penalty is incorporated to keep the generalizability of the encoder.

5 Detailed Design of ContrastSense

5.1 Contrastive Domain Loss

Contrastive Domain Loss (CDL) drives the model to learn domain-invariant features with the unlabeled data from $\mathcal{D}_S$ in a contrastive way. Specifically, the wearable sensing data samples from the same domain are defined as positives while data samples from different domains are negatives to each other. The similarity across positive features is maximized while the similarity across negative features is minimized. CDL is defined as:

$$L_{CDL} = -\sum_{k\in K} \frac{1}{|Q_i(x_k)|} \sum_{d\in Q_i(x_k)} \log \frac{\exp(z_k \cdot z_d/\tau)}{\sum_{q\in Q} I_{[q\neq k]} \exp(z_k \cdot z_q/\tau)}, \tag{2}$$

where $Q_i(x_k)$ refers to a collection of features that are from the same domain as data sample x_k, and the size of the collection is $|Q_i(x_k)|$. The Q is a collection of features from all source domains. The z_k is the feature of data sample x_k. The dot product between z_k and its positive feature z_d calculates their similarity. By clustering the positives with each other while pushing the negatives away, the model would learn the domain-related features, which is, however, contradictory to our goal. Therefore, CDL is maximized to learn domain-invariant features.

While existing supervised domain generalization methods [22, 35, 41, 51] merely utilize scarcely labeled data in D_{LS}, CDL learns domain-invariant features from unlabeled data in $\mathcal{D}_S$. Compared with few shots of labeled data, a large amount of unlabeled data from $\mathcal{D}_{LS}$ could provide more accurate sample distributions on $\mathcal{D}_{LS}$. In addition, CDL incorporates data from more domains, i.e., $\mathcal{D}_{US}$, which enables the domain-invariant features to be more representative. Existing works [9, 41, 72] require a domain classifier and the number of domains as prior knowledge in order to determine the architectures of the domain classifier. But when new domains are available, their networks need to be changed. CDL does not require a domain classifier, which simplifies the model design. Besides, it can be extended to new domains without changing the networks.

Domain Queues. To construct the feature collection $Q_i(x_k)$ for data x_k, we may simply sample a batch of data and collect data based on their domain labels. A large batch size K may provide more realistic domain distributions. However, it uses a large memory space as K goes large. ContrastSense employs a set of domain queues using domain labels, which not only provide a more memory-efficient way of constructing feature collections but also better capture more realistic domain distributions.

The domain queues set Q includes a set of domain-wise sub-queues: $Q = \{Q_n, n = 1, 2, \cdots, |Q|\}$, where $|Q|$ is the number of the sub-queues. Notably, $|Q|$ might be smaller than $|\mathcal{D}_S|$ due to the randomness of data sampling. Each time a batch of features is extracted, features from the $i-$th domain are enqueued to Q_i, and an equal number of features at the front of the domain queues are dequeued. The number of features stored in the domain queues is maintained constant at M, which is four times larger than K. The domain queues serve two primary objectives. Firstly, utilizing more samples from the domain queues for CDL than a data batch could provide a more accurate domain distribution. Secondly, the domain queues prevent encoding samples from scratch, optimizing memory usage during the pretraining phase. An experimental analysis of the memory usage is provided in Section 6.4. However, the challenge arises as the encoders undergo constant updates, potentially causing the features stored in the domain queues to become outdated compared to the features in the current batch. To mitigate this, ContrastSense employs a momentum update mechanism [16], which is described in Section 5.4.

5.2 SInfo Loss with Negative Selection

To extract discriminative high-level features, a novel contrastive loss, SInfo loss L_{SInfo}, is proposed to discriminate positives and selected negatives from the domain queues:

$$L_{SInfo} = -\sum_{k\in K} \log \frac{\exp(z_k \cdot z_{q+}/\tau)}{\sum_{q\in Q^s} I_{[q\neq k]} \exp(z_k \cdot z_q/\tau)}, \tag{3}$$

where $Q^s = \{Q^s_n, n = 1, \cdots, |Q|\}$ represents the domain queues after the negative selection. The domain queues retain an extensive collection of features, which are suitable for use as negatives. Compared with merely employing negatives from a data batch in Eq. (1), utilizing more negatives from the domain queues challenges the ability of the model to identify the positive for z_k, thus enhancing the quality of the learned high-level features [3, 16].

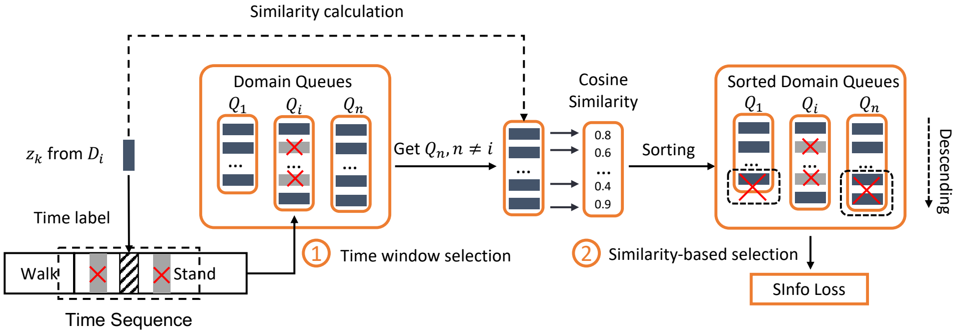


Fig. 4. Negative selection for SInfo loss in ContrastSense. The time window selection picks non-nearby samples based on time labels, while the similarity-based selection discards some samples from different domains using domain labels.

Moreover, SInfo loss has a dedicated procedure for selecting features in the domain queues as negatives, rather than indiscriminately using all features. This negative selection process prevents low-quality negatives from interfering with the contrast process, leading to more effective and robust features.

Heuristics for Negative Selection. ContrastSense employs two key heuristics to select negatives. First, to avoid using adjacent samples, time labels that indicate the time proximity between samples are employed to identify adjacent ones. Their features are then excluded from the negatives. Second, for samples from different domains, only a subset that is hard to discriminate from the positive is chosen. Negatives that are easy to distinguish from the positives have limited contribution to the loss and gradient during model updates [47], but may inadvertently teach the model some domain-specific knowledge. By omitting such samples using domain labels, the model could learn more robust features. Incorporating both heuristics, the negative selection for SInfo loss unfolds in two phases: time window selection followed by similarity-based selection, as depicted in Fig. 4.

Time Window Selection. To select negatives for feature z_k from domain D_i, its time label is compared with the time labels of features in the domain queue Q_i. A time window is applied to identify nearby samples. The features of those samples within the window are excluded from the selection process:

$$Q^t = \{z_q | t(z_q) \notin [t(z_k) - T/2, t(z_k) + T/2], \forall z_q \in Q\}, \tag{4}$$

where T is the time window length. The $t(\cdot)$ is to get the time label for z_q. The Q^t is the sampled domain queues. In some real-life scenarios, sudden changes in activities or status may occur, where the previous same-class assumption does not hold. To address this, a shorter time window can be selected for sequences with more frequent changes.

Similarity-based Selection. Subsequently, the similarities between the features from different domains and z_k are calculated, based on which the top r% most similar features are selected for contrast.

$$Q_n^s = \{z_q | rank(sim(z_q, z_k)) > r, z_q \in Q_n^t\}, n \neq i, \tag{5}$$

where $sim(z_q, z_k)$ refers to the cosine similarity between z_q and z_k. The $rank(\cdot)$ returns the ranking of z_q. For example, it returns 0.99 if z_q is the most similar feature among 100 features. The Q_n^t is the $n-$th domain queue after the time window selection. The top r% of similar features are selected for positives-negatives discrimination driven by SInfo loss, promoting the acquisition of effective and domain-invariant features. While the ranking process needs to be performed per domain queue for each positive sample, the training time only increases slightly from 1.57s to 1.60s per epoch, since the ranking for the queues can be conducted parallelly.

5.3 Parameter-wise Penalty

After the domain-invariant pretraining, the encoder f is fine-tuned using scarcely labeled wearable sensing data from $\mathcal{D}_{LS}$, along with one classifier to specialize the high-level features for downstream tasks. During fine-tuning, preserving domain-invariant knowledge is crucial for generalizability across diverse domains. To illustrate, when the model is fine-tuned for human activity recognition, the model can hardly generalize to the users in the wild if it overfits the users from $\mathcal{D}_{LS}$. To preserve the domain-invariant knowledge, the parameter-wise penalty is proposed to determine the extent to which each parameter in the encoder can be adjusted during fine-tuning. It assigns a penalty based on the criticality of parameters for preserving domain-invariant features. Parameters that are more critical for maintaining domain invariance receive higher penalties when adjusted.

To estimate the level of importance of each parameter on domain-invariant feature extraction, the fisher information matrix F is derived from the pre-trained encoder f:

$$F(i) = \left(\frac{\partial L_{CDL}}{\partial \theta_f(i)}\right)^2, \tag{6}$$

where $F(i)$ is one element in F for parameter $\theta_f(i)$. If $\theta_f(i)$ influences the quality of domain-invariant features to a large extent, then the first-order derivative of CDL to $\theta_f(i)$ would be large. When $\theta_f(i)$ is tuned, a larger $F(i)$ would lead to a larger penalty:

$$L_{penalty} = \sum_i F(i) \cdot (\theta_{f'}(i) - \theta_f(i))^2, \tag{7}$$

where $\theta_{f'}$ refers to the updated parameters in the encoder. The $L_{penalty}$ estimates the deviation of the fine-tuned encoder to the pretrained encoder and assign parameter-wise penalty based on their importance rather than a consistent penalty to all parameters. Therefore, parameters that are important to domain alignment have smaller flexibility to be tuned. In this way, ContrastSense specializes the high-level features for downstream wearable sensing tasks and at the same time maintains the domain-invariant knowledge in the encoder.

In contrast to the elastic weight consolidation approach that protects all previous task-related knowledge during continual learning [26], the parameter-wise penalty in ContrastSense only preserves the domain-invariant knowledge learned by CDL. This is because the high-level knowledge learned with SInfo loss is to discriminate positives from negatives, which should be refined for downstream classification tasks rather than preserved.

5.4 Domain-invariant Contrastive Learning

As depicted in Fig. 3, the three components are integrated into the ContrastSense framework for domain-invariant pretraining and fine-tuning, in order to obtain wearable sensing models for in-the-wild adoption.

5.4.1 Domain-invariant Pretraining.

Based on CDL and SInfo Loss, the loss L_{pt} for domain-invariant pretraining is derived as follows:

$$L_{pt} = L_{SInfo} - \lambda_1 L_{CDL}, \tag{8}$$

where λ_1 is a weight coefficient. The L_{SInfo} is minimized to extract discriminative features by differentiating between positives and negatives sampled from the domain queues, whereas L_{CDL} is maximized to facilitate domain-invariant feature extraction.

Data Augmentations. In each batch, unlabeled data x sampled from $\mathcal{D}_{\mathcal{S}}$ is augmented into x_k and x_{q+}. Data augmentations in [48] are adopted for IMU data, including rotation, negating, flipping, scaling, time warping, and adding noise. The same data augmentations are applied to EMG data, with the exclusion of the rotation augmentation, as it is not suitable for EMG signals.

Feature Extraction. Two branches of feature encoders f and f_m extract high-level representations from x_k and x_{q+}, which are subsequently projected to another feature space for calculating L_{pt} by projectors g and g_m [3]. The obtained features can thus be represented by $z_k = g \cdot f(x_k)$ and $z_{q+} = g_m \cdot f_m(x_{q+})$. The Euclidean norm of z_k and z_{q+} are normalized to one. The features output by g_m is enqueued to the domain queues Q, while the features output by g serves as queries to the queues for their negatives to derive L_{CDL} and L_{SInfo}.

Momentum update. The features in previous batches output by f_m and p_m are stored in the domain queues for CDL and SInfo calculation, which, however, could be inconsistent with the features in the current batches due to model update. To ensure feature consistency in the domain queues, ContrastSense adopts a standard technique in CL, momentum update [16], to update the encoder f_m and projector p_m. While the encoder f and projector p are updated via gradient descent, the momentum update for f_m and p_m is given by: $\theta_{F_m} \leftarrow m \cdot \theta_{F_m} + (1-m) \cdot \theta_F$, where $F = \{f, g\}$. The θ_{F_m} refers to the parameters of f_m or g_m, and θ_F represents the parameters of f or g. The m is the momentum ratio that is close to 1, which slows down the update speed of f_m and g_m and mitigates the feature inconsistency. The selection of m is further discussed in Section 6.4.

The design of the encoders for IMU in ContrastSense follows that of DeepSense [66], but in addition, the residual connection between layers [17] and the self-attention layers [55] are included to extract high-level features. The encoder in [6] is adopted to extract features from EMG, which consists of two convolutional layers. The projectors are Multi-layer Perceptions (MLP) with three linear layers.

5.4.2 Domain-invariant Fine-tuning. After the pretraining, the encoder f would be saved along with its fisher information matrix. Then the domain-invariant fine-tuning is evoked to specialize the encoder for wearable sensing tasks along with one classifier. The classifier has one GRU layer followed by two linear layers. The loss L_{ft} for domain-invariant fine-tuning is described as follows:

$$L_{ft} = L_{clf} + \lambda_2 L_{penalty}, \tag{9}$$

where L_{clf} is the cross-entropy loss, which is calculated based on the output of the classifier. The λ_2 is a weight coefficient. During the domain-invariant fine-tuning, the loss L_{clf} can specialize the high-level features for the downstream tasks, and $L_{penalty}$ assigns a parameter-wise penalty to preserve the domain-invariant knowledge. After the fine-tuning, the encoder and the classifier are saved for inference on the data in the target domains.

6 Experiments

6.1 Experiment Setup

To present the effectiveness of ContrastSense, two kinds of modalities and tasks are selected, i.e., human activity recognition (HAR) with IMU and gesture recognition (GR) with EMG. Different cross-domain scenarios over users, devices, on-body positions, and datasets are adopted to train and evaluate ContrastSense. The considered cases includes *cross-user*, *cross-device*, *cross-position*, *cross-user-device*, *cross-user-position*, and *cross-dataset* scenarios. In the cross-user (cross-device or cross-position) scenario, different users (devices or positions) are considered as different domains. In the *cross-user-device* and *cross-user-position* scenarios, one user with different devices or different on-body positions are considered as different domains. In the *cross-dataset* scenario, the datasets are treated as domains, in which cases multiple domain shifts are presented simultaneously. A quantitative analysis of the degrees of domain shifts in each scenario is provided in Appendix A.

We randomly select $\alpha\%$ domains as the training set, 15% domains for validation, and the rest domains for testing. The n shots of labels from $\beta\%$ domains in the training set are used for fine-tuning. For example, in the cross-user scenario, models are trained on the unlabeled data from $\alpha\%$ of users and n shots of labeled data from $\alpha \times \beta\%$ of users and then tested on the users in the test set. To provide a thorough assessment of the model performance, a wide range of experiments are conducted under varied α, β, and n, which present the models with varying levels of domain shifts and class label scarcity. Five different random splits are generated, and the average results are reported.

6.2 Human Activity Recognition with Inertial Measurement Units

6.2.1 Datasets. ContrastSense is evaluated with four public datasets on HAR with IMU, i.e., HHAR, MotionSense, Shoaib, and HASC-PAC2016 datasets. MotionSense and HASC-PAC2016 are abbreviated as Motion and HASC, respectively. Being widely adopted by existing works [65, 70, 71], they cover a wide range of users, devices, and on-body positions, and to the best of our knowledge, the HASC dataset is the largest real-world dataset for HAR. The following introduces the details of each dataset:

Motion dataset [37]. It contains 24 users with different genders, ages, heights, and weights. One iPhone 6s is put into the front pockets of users to collect accelerometers and gyroscopes data for six activities at a 50Hz sampling rate. This dataset mainly involves user heterogeneity.

HHAR dataset [50]. It collects IMU sensing data for six activities from nine users and three types of smartphones. Data from accelerometers and gyroscopes are sampled at the highest frequency permitted by the devices rather than a fixed one. This dataset mainly involves user and device heterogeneity.

Shoaib dataset [49]. It collects IMU data from five different body positions on ten subjects, including the wrist, upper arm, belt, and left and right pockets. Samsung Galaxy smartphones are used to collect seven different activities at a 50Hz frequency. This dataset mainly involves user and on-body position heterogeneity.

HASC dataset [19]. It contains real-life motion data from 64 people and 18 devices across 5 years. Users perform unconstrained activities consecutively, and the wearing positions of devices could be changed during the test. Devices including smartphones, smartwatches, and commercial wearable sensors are included. Six activities are considered in our study and the data numbers of those activities are highly imbalanced. This dataset is the largest and has the highest scale of heterogeneity across users, devices, and on-body positions.

All four datasets include walking, upstairs, and downstairs activities. In addition, the Motion dataset contains jogging, sitting, and standing activities; the HHAR dataset collects data for biking, sitting, and standing; the Shoaib dataset includes jogging, sitting, standing, and biking activities; the HASC dataset has jumping, jogging, and staying activities. The accelerometers and gyroscopes are used for HAR, and the window length is set to 200 without overlapping. Data with different frequencies are resampled to 50Hz.

6.2.2 Baselines. The performance of ContrastSense is compared with eight state-of-the-art domain adaptation and generalization and self-supervised learning approaches designed for HAR:

FMUDA and CMUDA [2] are two semi-supervised domain adaptation methods for HAR. FMUDA matches features by minimizing maximum mean discrepancy across different domains. CMUDA adopts the domain adversarial neural networks to align features.

Mixup [67] is a classific data augmentation approach for domain generalization. It generates artificial data by mixing data from different distributions in a linear way.

GILE [41] is one domain generalization approach that learns domain-invariant and domain-specific features with an Independent Excitation mechanism for cross-user HAR.

LIMU-BERT [65] is one generative learning method on IMU sensing with scarce labels, which designs a lightweight BERT-like model with a self-attention mechanism to extract high-level temporal features. It is abbreviated as LIMU.

CPCHAR [15] adopts Contrastive Learning for HAR to deal with class label scarcity. During pretraining, it learns the temporal structure in data by predicting nearby features.

ColloSSL [21] leverages unlabeled IMU data from multiple devices for contrastive learning. It proposes a device selection module and a contrastive sampling algorithm to select devices and data samples to calculate a novel loss, Multi-view Contrastive Loss.

ClusterCLHAR [58] proposes a HAR framework using Contrastive Learning, which selects negatives by unsupervised clustering methods based on SimCLR [3].

Table 2. Performance comparison on cross-user experiment with $n = 10$, $\alpha = 25\%$, and $\beta = 40\%$ for HAR with IMU.

Method	HHAR		Motion		Shoaib		HASC		Average		Weighted	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
FMUDA	50.96	48.49	66.21	62.61	50.21	48.96	37.69	25.45	51.27	46.38	49.88	45.37
CMUDA	48.85	46.48	65.18	61.54	50.04	48.47	35.12	23.39	49.80	44.87	48.50	43.96
Mixup	58.16	55.57	64.80	61.54	72.73	72.03	35.28	26.59	57.74	53.93	58.82	55.35
GILE	49.11	42.51	58.55	57.16	62.32	59.82	46.90	25.65	54.22	46.29	54.58	47.74
ColloSSL	–	–	–	–	52.80	52.26	–	–	–	–	–	–
LIMU	65.81	60.40	63.69	60.93	71.48	71.11	**54.44**	13.79	63.86	51.57	64.78	53.25
CPCHAR	56.33	53.34	63.46	61.90	72.15	71.49	33.70	27.26	56.41	53.49	57.56	54.77
ClusterCLHAR	47.20	42.37	67.90	65.02	70.23	69.56	39.84	26.12	56.29	50.76	56.55	51.36
ContrastSense	**68.35**	**66.43**	**73.26**	**71.82**	**78.46**	**78.23**	44.70	**34.92**	**66.19**	**62.85**	**67.03**	**63.92**

Note that there is no existing work that specifically targets CL for domain generalization on HAR. We selected the most relevant works as baselines for ContrastSense. Some CL methods, such as SimCLRHAR [53] and MoCoHAR [59], are not included for comparison since ClusterCLHAR already outperforms them as reported in [58]. FMUDA and CMUDA conduct domain-invariant learning on unlabeled data from $\mathcal{D}_T$, which are inaccessible in our scenarios. So we instead train the two models with the unlabeled data from $\mathcal{D}_S$. ColloSSL [21] requires at least three synchronized devices worn by the participants thus it is only evaluated on Shoaib dataset in the cross-user scenario. The accuracy and F1 score on each dataset are reported along with the mean accuracy and F1 score across the four datasets for result comparison. In addition, considering the difference between datasets, the weighted average accuracy and F1 score are also reported. The weight $w(i)$ for the i-th dataset is obtained via: $w(i) = n_{test}(i)/\sum_j n_{test}(j)$, where $n_{test}(i)$ represents the number of test samples for the i-th dataset.

6.2.3 Implementation Details. The Adam optimizer is adopted to train the model along with the cosine annealing learning rate decay [34]. The initial learning rates are set to be 1e-4 and 5e-4 during pretraining and fine-tuning, respectively. The hyperparameters for IMU is set as follows: $T = 120$, $\lambda_1 = 0.7$, $\lambda_2 = 50$, $m = 0.999$ and $M = 1024$. The selection ratio r in the similarity-based selection is set to 0.5, 0.8, 0.8, and 0.7 for HHAR, Motion, Shoaib, and HASC datasets, due to the difference in the number of domains and data quality of datasets. More discussions on the hyperparameters are presented in Appendix B.

6.2.4 Result Comparison. We evaluate the performance of ContrastSense in cross-user experiments with four different datasets. We set the initial number of shots n to 10, the percentage of source domain α to 25%, and the percentage of labeled source domain β to 40%. Later, we also investigate the impact of varied n, α, and β.

Table 2 shows the results of ContrastSense in comparison to the baselines. With limited source domains and scarce labels, ContrastSense achieves an average improvement of 8.9% in F1 score and 2.3% in accuracy over the four datasets. Notably, the improvement in F1 scores is more substantial with more users in the dataset. Specifically, on HHAR and Shoaib datasets, the F1 scores are improved by 6.0% (runner-up LIMU) and 6.2% (runner-up Mixup) while on Motion and HASC datasets, the F1 scores are improved by 6.8% (runner-up ClusterCLHAR) and 7.7% (runner-up CPCHAR). These results demonstrate that ContrastSense can better handle class label scarcity and domain diversity across users than the baselines (a statistical test is provided in Appendix C). Note that on HASC dataset, LIMU achieves slightly higher accuracy than ContrastSense but a much lower F1 score. The reason is that the data in HASC dataset is highly imbalanced, where some classes have much more data than others. LIMU may overfit to classes with sufficient data, whereas ContrastSense shows better robustness. Overall, ContrastSense achieves better performance on cross-user scenarios than all eight baselines.

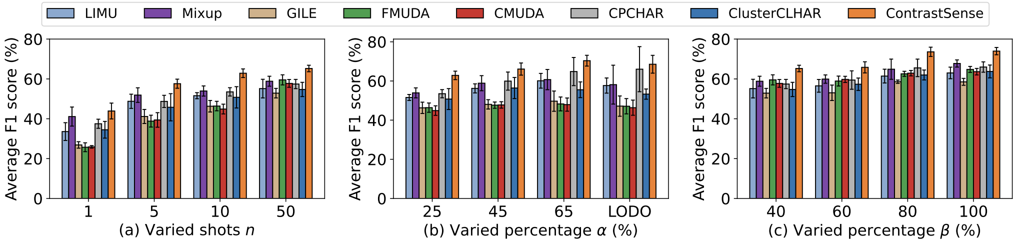


Fig. 5. Performance comparison with varied settings on the cross-user experiment for HAR with IMU. The average F1 score on the four datasets is presented. The error bar represents the standard deviation. ColloSSL is not included since it can only be applied to the Shoaib dataset.

Different Shots of Labels. We vary the label shots number n, which influences the degree of class label scarcity. In Fig. 5(a), with more labels provided, the performances of all models improve since more labels provide more accurate sampled distributions for supervision. When n is 1, 5, 10, 50, ContrastSense outperforms the best baseline by 2.8%, 5.7%, 8.9%, and 5.7% on average in terms of F1 scores, respectively.

Different Percentages of Source Domains. We further vary the percentages of source domains α, which affects the degree of domain shifts to be experienced. We select four different settings, $\alpha = 25\%, \alpha = 45\%$, $\alpha = 65\%$, and Leave-One-Domain-Out (LODO) settings. In Fig. 5(b), When α is 25%, 45%, and 65%, ContrastSense outperforms the best baseline by 8.9%, 6.1%, and 5.5% F1 score on average, respectively. In the LODO setting, ContrastSense outperforms the best baseline by 2.5% F1 score. Besides, we notice that in the LODO setting, ContrastSense does not outperform LIMU and CPCHAR on the HHAR and Motion datasets. This can be attributed to the relatively mild domain shifts presented in both datasets under the LODO setting. However, ContrastSense outperforms them in various settings, which demonstrates its robustness and generalizability.

Different Percentages of Labeled Source Domains. We further assess the impact of varied percentages of labeled source domains β, which influences the degree of domain shifts and the overfitting problem during fine-tuning. Fig. 5(c) shows that ContrastSense on average outperforms the best baseline by 5.7%, 5.9%, 8.0%, and 6.2% in terms of F1 score over the four datasets when β is 40%, 60%, 80%, and 100%, respectively. The results suggest that ContrastSense may learn domain-invariant features that are more robust by utilizing the unlabeled source domains. In contrast, domain-invariant features learned by the baselines could be less representative and less robust on the target domains, if only labeled data from the labeled source domains $\mathcal{D}_{LS}$ is used. The statistical tests are conducted on each setting based on the F1-score in Fig.5, and there is over 99.9% confidence to conclude that ContrastSense outperforms those baselines in different settings (please see Appendix C). Besides, a detailed results comparison on each dataset in varied settings is presented in Appendix D.

6.2.5 Ablation Study. An ablation study is conducted to validate the effectiveness of each design component in ContrastSense, the result of which is presented in Table 3. By pretraining the model with a classic CL loss, InfoNCE loss, 54.91% average F1 score is achieved. InfoNCE loss allows the model to extract useful information from the unlabeled data. In contrast to InfoNCE loss, which uses all views augmented from different samples, SInfo loss selects some negatives for contrast. Its performance is 5.1% better than that of InfoNCE loss in terms of average F1 score. Such a boost can be attributed to the negative selection strategy, which omits some adjacent samples and easily discernible ones from diverse domains. On the other hand, combining CDL with InfoNCE loss escalates the performance, reflecting a 5.7% increase on the average F1 score. CDL drives the model to extract domain-invariant information among $\mathcal{D}_S$ to enhance its generalizability without any activity labels. Moreover, a synergistic effect is observed when CDL is jointly optimized with SInfo loss, resulting in a 1.4% F1 score

Table 3. Ablation Study on HAR with IMU. It is based on the cross-user experiments with $n = 10$, $\alpha = 25\%$, and $\beta = 40\%$

Design*				HHAR		Motion		Shoaib		HASC		Average		Weighted	
InfoNCE	SInfo	CDL	PwP	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
	✓	✓	✓	**68.35**	**66.43**	**73.26**	**71.82**	**78.46**	**78.23**	**44.70**	**34.92**	**66.19**	**62.85**	**67.03**	**63.92**
	✓	✓		68.27	66.33	72.34	70.64	76.74	76.51	43.85	34.19	65.30	61.92	66.08	62.95
✓		✓		65.73	62.88	72.30	69.47	76.54	76.22	44.50	33.76	64.77	60.58	65.47	61.64
	✓			64.66	60.79	71.02	69.34	76.25	76.02	43.77	33.85	63.92	60.00	64.71	61.01
✓				54.94	51.17	69.98	68.13	71.19	70.20	37.83	30.13	58.48	54.91	58.80	55.36

* PwP refers to the parameter-wise penalty.

improvement on average compared with when just paired with InfoNCE loss. While SInfo loss and CDL both enhance the robustness of the model, the results indicate that they are complementary, as they address distinct challenges in the CL process. Last but not least, during the domain-invariant fine-tuning stage, the average F1 score is further improved by 1.2% with the PwP. When training with the PwP, the mean F1 score is 62.85 and the standard deviation is 2.17. When training without PwP, the mean F1 score is 61.92 and the standard deviation is 2.88. The Wilcoxon signed-rank test shows the test statistic is 154.0 and the p-value is 0.035, which means there is more than 95% confidence to conclude that the PwP is effective for improving performance. In addition to the results in Table 3, a more detailed analysis of the effectiveness of CDL and the negative selection is provided in Appendix E.

6.3 Gesture Recognition with Electromyography

6.3.1 Datasets. ContrastSense is evaluated with three public datasets on GR with EMG, i.e., MyoArmBand [6], NinaPro DB4 datasets [40], and NinaPro DB5 datasets [40], which are abbreviated as Myo, DB4, DB5. To the best of our knowledge, the Myo dataset is the largest public EMG dataset. The following introduces each dataset:

Myo dataset [6]. It collects the data of 40 subjects with a commercial EMG sensor, the Myo Armband. The armband is worn by users on the forearm with a sampling rate of 200Hz, and 7 gestures are included, i.e., Neutral, Hand Close/Open, Wrist Extension/Flexion, and Ulnar/Radial Deviation.

NinaPro DB4 [40]. It collects the EMG data of 4 females and 6 males using 12 Cometa electrodes. The sampling rate is 2kHz. The first 8 electrodes are equally spaced around the forearm, which are used in our experiments.

NinaPro DB5 [40]. It contains the EMG data of 10 righthand subjects. Two Myo Armbands are deployed around the elbow, one on the radio humeral joint and one closer to the hand.

Following [6], the window length of each sample is set to 52, and there is no overlapping between samples. Besides, the EMG data in DB4 is down-sampled to 200Hz. Notably, NinaPro DB4 differs significantly from NinaPro DB5 in terms of data collection procedures, devices used, and participants.

6.3.2 Baselines. The performance of ContrastSense is compared with three state-of-the-art baselines for GR with EMG that are most related, i.e., Mixup, CALDA, and ConSSL. Other baselines in Section 6.2 are not used since they are proposed specifically for IMU. As Mixup has been introduced in Section 6.2, the latter two are briefly described as follows:

CALDA [63] propose a contrastive adversarial learning method for domain adaption, in which adversarial learning aligns the domain distributions and the contrastive loss clusters samples with the same labels.

ConSSL [28] designs a domain adaptation framework with contrastive learning to overcome the domain shifts across subjects. For each sample in the source domain, it selects the sample next to it as the positive and some non-adjacent ones as negatives, and then the model is fine-tuned on the target domain.

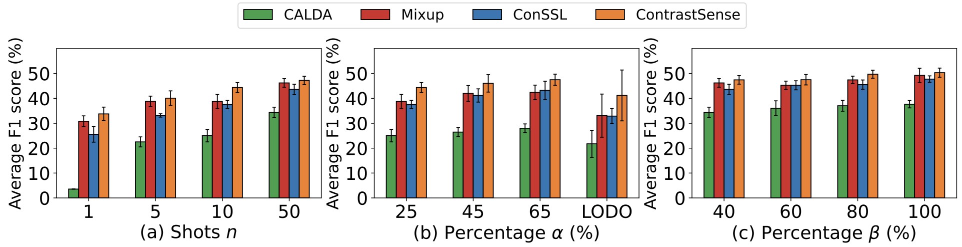


Fig. 6. Performance Comparison with varied settings on GR with EMG. (a) Performance with varied shots n. (b) Performance with varied percentages α. (c) Performance with varied percentages β.

Table 4. Performance comparison on cross-user experiment with $n = 10$, $\alpha = 25\%$, and $\beta = 40\%$ for GR with EMG.

Method	Myo		DB4		DB5		Average		Weighted	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
CALDA	35.76	33.27	27.06	22.47	25.21	19.24	29.34	24.99	32.40	28.94
Mixup	55.04	52.98	33.66	30.42	36.99	32.87	41.90	38.75	48.08	45.46
ConSSL	60.82	59.66	26.90	22.80	32.89	30.17	40.21	37.54	49.91	47.94
ContrastSense	**62.01**	**61.19**	**36.45**	**33.83**	**40.52**	**38.03**	**46.32**	**44.35**	**53.71**	**52.28**

Since the target domain data is not accessible in the proposed scenario, CALDA is only trained in the source domains, and ConSSL is fine-tuned on the labeled data in the source domains.

6.3.3 Implementation Details. The learning rate is set to 1e-3 for pretraining and fine-tuning. The T is 40, r is 0.8, and λ_2 is 4e3. The λ_1 is 0.7, 0.4, and 0.1 for Myo, DB4, DB5, respectively, considering their large difference in the number of users. The rest of the hyperparameters are kept aligned with those used for IMU sensors.

6.3.4 Result Comparison. Table 4 shows the performance of ContrastSense and baselines in the cross-user scenario on GR with EMG. On the Myo dataset, ContrastSense outperforms the best baseline by 1.2% average accuracy and 1.5% average F1 score. On the DB4 dataset, the average accuracy and F1 score improvement over the best baseline are 2.8% and 3.2%, respectively. As for the DB5 dataset, it outperforms the best baseline by 3.3% average accuracy and 4.7% average F1 score. The average performance of ContrastSense reaches 46.32% accuracy and 44.35% F1 scores, which outperforms the best baselines by 4.4% and 5.6%, respectively. The results indicate the superior performance of ContrastSense compared with the baselines on GR with EMG.

Different Shots of Labels. Fig. 6(a) presents the results with varied shots of labeled data when $\alpha = 25\%$ and $\beta = 40\%$. When n is 1, 5, 10, 50, the performance of ContrastSense outperforms the best baselines by 2.9%, 1.3%, 5.6%, and 1.0%, which shows the effectiveness of ContrastSense in handling class label scarcity.

Different Percentages of Source Domains. Fig. 6(b) presents the results with varied α when $\beta = 40\%$ and $n = 10$. When α is 25%, 45%, and 65%, ContrastSense outperforms the best baseline by 5.6%, 4.0%, and 4.3% on average in terms of F1 score. In the LODO setting, ContrastSense outperforms the best baseline by 8.1% F1 score. We observe some abnormal performance decrease comparing the results in the LODO setting with the results when $\alpha = 65\%$. The reason might be the data of the test user(s) randomly selected in some of the five splits are quite different from the data used for training. The results show the generalizability of ContrastSense to handle different amounts of unlabeled data from varied size of source domains.

Table 5. Ablation Study on GR with EMG. It is based on the cross-user experiments with $n = 10$, $\alpha = 25\%$, and $\beta = 40\%$

	Design*			Myo		DB4		DB5		Average		Weighted	
InfoNCE	SInfo	CDL	PwP	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
	✓	✓	✓	**62.01**	**61.19**	**36.45**	**33.83**	**40.52**	**38.03**	**46.32**	**44.35**	**53.71**	**52.28**
	✓	✓		61.40	60.68	36.25	33.51	40.28	37.96	45.98	44.05	53.24	51.88
✓		✓		59.06	57.92	35.00	32.86	38.62	36.86	44.27	42.55	51.21	49.78
	✓			60.40	58.99	34.80	33.08	40.08	37.24	45.09	43.10	52.29	50.58
✓				58.06	56.87	34.10	31.92	37.64	35.13	43.27	41.31	50.23	48.64

[*] PwP refers to the parameter-wise penalty.

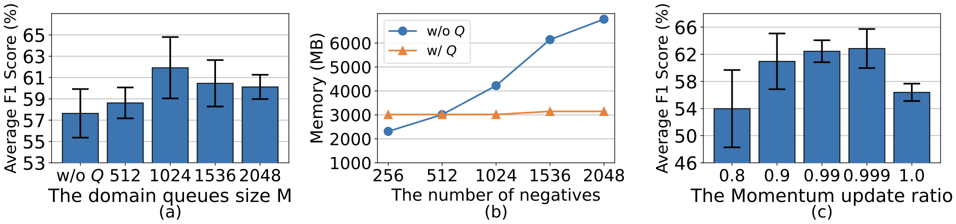


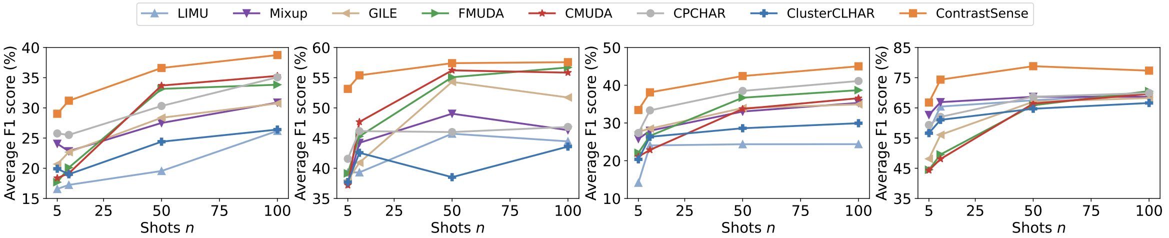
Fig. 7. Effectiveness of the domain queues. (a) Performance with varied domain queues size. (b) Memory usage with and without domain queues with varied numbers of negatives. (c) Performance with varied momentum update ratio.

Different Percentages of Labeled Source Domains. Fig. 6(c) presents the results with varied percentages β when $\alpha = 25\%$ and $n = 50$. When β is 40%, 60%, 80%, and 100%, the average F1 score achieved by ContrastSense are 1.2%, 2.2%, 2.3%, and 1.1% higher than the baselines, respectively. Statistical tests show that ContrastSense consistently outperforms those baselines in different settings on GR with EMG (please see Appendix C).

6.3.5 Ablation Study. The ablation study presented in Table 5 justifies the effectiveness of each design in ContrastSense for GR with EMG. While InfoNCE loss for CL achieves 41.31% F1 scores, it does not account for adjacent negatives and domain shifts. In contrast, SInfo loss addresses these limitations and yields a boost of 1.8% average F1 score with negative selection. When CDL is integrated with InfoNCE loss, the performance is further elevated by a 1.2% average F1 score. When both SInfo loss and CDL are included for the domain-invariant pretraining, the performance improves by 2.7% in terms of average F1 score, compared with using InfoNCE loss. When the PwP is further integrated, the mean F1 score is improved from 44.05 (standard deviation=1.41) to 44.35 (standard deviation=1.99). The Wilcoxon signed-rank test shows that there is over 80% confidence degree to conclude that the PwP is effective (the t-value is 78.0, the p-value is 0.165). Compared with the ablation study on IMU datasets, the improvements of the design components are smaller, which might be due to the difference in modalities and datasets. These results show that ContrastSense is effective for different kinds of modalities and applications.

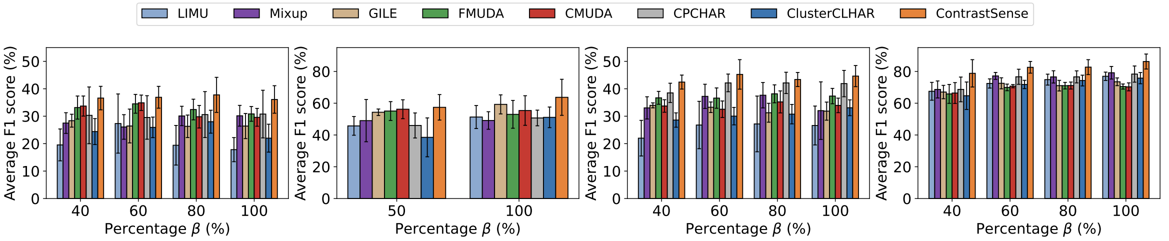
6.4 Effectiveness of the Domain Queues

The domain queues in ContrastSense not only supply more features for CDL and SInfo loss but also save memory usage during training. The effectiveness of domain queues is further discussed in this section with the IMU datasets, and similar results can be obtained with the EMG datasets.



(a) Cross-device scenario (b) Cross-position scenario (c) Cross-user-device scenario (d) Cross-user-position scenario

Fig. 8. Performance comparison with varied shots of labels n in different cross-domain scenarios.



(a) Cross-device scenario (b) Cross-position scenario (c) Cross-user-device scenario (d) Cross-user-position scenario

Fig. 9. Performance comparison with varied percentages β in different cross-domain scenarios.

Fig. 7(a) shows that compared with training without the domain queues (only the features in the current batch are used), training with the domain queues when $M = 1024$ achieves a 5.2% F1 score improvement, thereby substantiating the efficacy of the domain queues design. Besides, a larger domain queues size could improve the quality of the domain-invariant features, since more samples from source domains may provide a more realistic sampled domain distribution, allowing the encoder to align features from source domains more accurately. However, when the domain queues contain too many features, it may be challenging for the model to conduct positive-negative classification, resulting in the observed performance degradation in Fig. 7(a).

Fig. 7(b) shows that with a larger number of negatives for contrast, CL experiences a substantial surge in memory usage when domain queues are absent. In contrast, the integration of domain queues leads to a marginal increment in memory usage. The reason is that the utilization of domain queues empowers the model to directly access the stored features. In contrast, the model trained without the use of domain queues needs to encode negative samples to features from scratch, which uses much more memory than the domain queues.

However, the features from previous batches that are stored in the domain queues could be different from the current features, which may hinder the model from learning high-quality features. To mitigate this issue, ContrastSense employs the momentum update [16], the impact of which is presented in Fig. 7(c). When f_m and g_m are updated with a higher momentum update ratio, the stored features in the domain queues are more consistent, thus achieving better positive-negatives discrimination during the domain-invariant pretraining. When the ratio is 1.0, f_m and g_m are not updated, leading to suboptimal performance.

6.5 Overcoming Different Kinds of Domain shifts

The ability of ContrastSense to handle different cross-domain scenarios is further evaluated in this section. In the cross-device and cross-user-device scenarios, the HASC dataset is used for results comparison. Other datasets are excluded because they collect data from few devices. Cross-position and cross-user-position experiments are conducted on the Shoaib dataset since the other datasets do not provide on-body position labels or only include

Table 6. Performance comparison on cross-dataset experiment with $n = 50$ for HAR with IMU. For example, when the HHAR dataset is the target, the other three datasets are used for training, and the models are evaluated on the HHAR dataset.

	Target Dataset								Average		Weighted	
Method	HHAR		Motion		Shoaib		HASC					
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
FMUDA	57.97	46.90	69.27	54.22	65.74	59.60	55.63	39.99	62.15	50.18	61.76	50.51
CMUDA	55.61	45.23	68.60	47.77	54.37	50.31	41.58	32.84	55.04	44.04	55.44	44.53
Mixup	57.97	42.74	47.94	28.92	64.46	54.89	42.18	28.09	53.11	38.66	54.78	40.89
GILE	58.47	43.21	63.92	42.25	60.91	49.61	32.18	28.98	53.87	41.01	54.46	41.91
LIMU	45.77	26.33	62.75	36.55	64.00	58.19	19.40	19.47	47.98	35.14	48.52	36.25
CPCHAR	55.90	35.77	10.50	6.54	64.29	55.29	57.03	33.29	46.93	32.73	50.88	36.13
ClusterCLHAR	35.10	23.41	11.81	7.31	45.53	39.77	33.77	21.77	31.55	23.07	29.89	21.55
ContrastSense	**71.81**	**64.96**	**75.70**	**57.85**	**76.70**	**72.59**	**73.87**	**41.22**	**74.52**	**59.16**	**74.68**	**57.10**

limited positions. The cross-device-position experiment is not conducted since Shoaib only includes one device. As there are only five on-body positions in the Shoaib dataset, we randomly select two positions (domains) for training, one for evaluation, and the rest for testing in the cross-position scenario. In all scenarios except for the cross-dataset scenario, the percentage α is fixed at 25% to align the experiment settings with the proposed scenario in Fig. 1.

In the cross-dataset scenarios, the models are required to overcome domain shifts from users, devices, positions, and collection procedures. For the IMU datasets, we focused on four common classes among the datasets: *static*, *walking*, *downstairs*, and *upstairs*. For the EMG datasets, we considered five classes: *neural*, *hand close*, *hand open*, *wrist extension*, and *wrist flexion*. Considering the limited number of datasets, one dataset is selected as the target dataset, and the rest are used for training.

6.5.1 Cross-Device Scenario. Fig. 8(a) and Fig. 9(a) compare the performance of ContrastSense and baselines in the cross-device scenario. When $n = 5, 10, 50$, and 100, the F1 score improvements of ContrastSense over the best baseline are 3.3%, 5.7%, 2.9%, and 3.5%, respectively. When $\beta = 40\%, 60\%, 80\%$ and 100%, ContrastSense achieves higher F1 scores than the best baseline by 2.9%, 2.0%, 5.3%, and 5.3%, respectively. The results suggest that ContrastSense more effectively utilizes the scarce labels from limited $\mathcal{D}_S$ to generalize across different devices.

6.5.2 Cross-Position Scenario. Fig. 8(b) shows that when $n = 5, 10, 50$ and 100, ContrastSense outperforms the best baseline by 11.6%, 7.7%, 1.2%, and 0.9% in terms of F1 scores, respectively. Besides, Fig. 9(b) shows the F1 score gained with ContrastSense in comparison to the best baseline increases from 1.2% when $n = 50$ and $\beta = 50\%$ to 4.4% when $n = 50$ and $\beta = 100\%$. Overall, the results suggest that ContrastSense can better deal with the domain shifts across on-body positions with scarce labels and limited source domains.

6.5.3 Cross-User-Device Scenario. Fig. 8(c) and Fig. 9(c) present the performance of ContrastSense and baselines with varied numbers of label shots n and percentages β, when presented with domain shifts caused by users and devices. When $n = 5, 10, 50$, and 100, ContrastSense outperforms the best baseline by 5.8%, 4.8%, 3.9%, and 3.9% on average in terms of F1 score. When $\beta = 40\%, 60\%, 80\%$, and 100% with n fixed to 50, the performances of ContrastSense over the best baseline are 3.9%, 3.1%, 1.2%, and 2.7%, respectively. The results show that ContrastSense could handle domain shifts from users and devices simultaneously.

6.5.4 Cross-User-Position Scenario. The results of cross-user-position scenario are presented in Fig. 8(d) and Fig. 9(d). When $n = 5, 10, 50$, and 100, ContrastSense achieves 4.1%, 7.5%, 10.1%, and 6.8% improvements on average in

Table 7. Performance comparison on cross-dataset experiment with $n = 50$ for GR with EMG.

Method	Target dataset Myo		DB4		DB5		Average		Weighted	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
CALDA	28.03	19.46	28.43	27.62	27.74	24.68	28.07	23.92	28.06	21.90
Mixup	**40.03**	36.67	25.17	17.58	35.62	31.21	33.61	28.49	36.43	32.09
ConSSL	21.63	16.20	18.15	8.97	32.70	30.95	24.16	18.71	22.81	17.27
ContrastSense	39.24	**38.02**	**30.39**	**28.56**	**35.75**	**31.23**	**35.16**	**32.78**	**37.01**	**35.15**

Table 8. Inference Overhead on IMU sensor data

Metric	FMUDA	CMUDA	CPCHAR	GILE	ColloSSL	Mixup	ClusterHAR	LIMU	ContrastSense
Inference time (ms)	3	4	455	11	4	6	7	40	21
CPU usage (%)	11	13	46	8	18	16	15	24	18
Memory usage (%)	0.83	0.84	0.93	0.95	0.79	0.83	0.84	0.90	1.48

terms of F1 score compared with the best baseline. When $\beta = 40\%, 60\%, 80\%$, and 100% with $n = 50$, ContrastSense outperforms the best baseline by 10.1%, 5.4%, 6.1%, and 7.1% on average in terms of F1 score. ContrastSense could learn high-quality domain-invariant features when presented with different kinds of domain shifts.

6.5.5 Cross-Dataset Scenario. The results of the cross-dataset scenario are presented in Table 6 and Table 7. ContrastSense demonstrated a notable average accuracy improvement of 12.3% and an average F1 score improvement of 9.0% on the IMU datasets. For the EMG datasets, the improvements of ContrastSense over the best baselines are 1.6% in terms of accuracy and 4.3% in terms of F1 score. We acknowledge that while ContrastSense achieves better results, the performance on EMG datasets suggests that there is a large room for improvement. The reason might be that the large dataset gap between different EMG datasets makes the domain-invariant features extracted from the source EMG datasets less applicable to the target EMG dataset. Future work will focus on further exploring these cross-dataset challenges.

6.6 Computational Overhead of ContrastSense

The on-device inference overhead of ContrastSense is further analyzed in this section, using one Samsung Galaxy S8 equipped with an Octa-core CPU and 4 GB RAM. The average inference time on ten samples, the average CPU usage, and the total memory usage during operation on the mobile phone are included as the evaluation metric.

As shown in Table 8 and Table 9, the results demonstrate that ContrastSense achieves comparable inference times of 21ms and 8ms for IMU and EMG data, respectively, matching the performance of baseline methods. While the memory usage of ContrastSense is slightly higher, primarily due to its use of larger intermediate features and a more complex model structure (e.g., the DeepSense model [66]), it remains within the affordable range (only 1.48% for IMU data and 1.00% for EMG data) for modern mobile devices. Moreover, this overhead can be mitigated through techniques such as model quantization and distillation [62]. Overall, the overhead is affordable considering the performance gains of ContrastSense.

Table 9. Inference Overhead on EMG sensor data

Metric	CALDA	Mixup	ConSSL	ContrastSense
Inference time (ms)	10	4	43	8
CPU usage (%)	15	13	13	14
Memory usage (%)	0.83	0.13	0.80	1.00

7 Discussions

We comprehensively evaluate ContrastSense across various datasets and settings, demonstrating its ability to address domain shifts and class label scarcity. In this section, we further explore the practical adoption of ContrastSense and its applicability to other modalities, and outline potential future directions.

7.1 Practical Adoption of ContrastSense for In-the-Wild Wearable Sensing

Real-life Use Scenarios. ContrastSense addresses the challenges of label scarcity and domain shifts in in-the-wild wearable sensing scenarios (Fig. 1), making it applicable for real-life scenarios, such as intelligent healthcare and human-computer interaction [20, 32]. For instance, wearable sensors on patients or the elderly can be utilized for services like fall detection and sleep staging, aiding in disease diagnosis, prevention, and intervention [20, 45]. In human-computer interaction, interactive game control requires accurate gesture recognition, which can be facilitated by hand-held devices or wearable sensors [32]. However, collecting labeled data for every patient or game player for model training is impractical due to the labeling overhead and data privacy concerns, limiting the amount of labeled data and the number of domains for training. In such cases, ContrastSense can be employed to extract domain-invariant features from the unlabeled data and limited labeled data, which enhances the feasibility of deploying pervasive computing solutions that blend into various real-world scenarios with less effort.

Enhancing the Real-life Deployability. While ContrastSense has achieved better performance compared with the baselines, its performance, particularly on the HASC dataset, is still below expectations. To enhance real-life deployability, future works may explore data augmentation techniques and deep learning-based data synthesis. Methods such as mixup and Generative Adversarial Networks [35, 56] can be leveraged to synthesize realistic data, thereby increasing data heterogeneity for improved domain alignment. Apart from the substantial domain shifts, the challenges related to the HASC dataset also include imbalanced categories and potential inaccuracies in labels collected through crowdsourcing. To handle those difficulties, methods like class re-sampling [18] and label correction [13] would be further investigated.

7.2 Applying ContrastSense to Other Kinds of Modalities

In the experiments, ContrastSense has been evaluated in the context of HAR with IMU and GR with EMG. While we are optimistic regarding its potential applicability to other time-series data or sensing modalities, certain assumptions inherent to ContrastSense may limit its extension to diverse modalities. For instance, the time window selection assumes adjacent samples within the window belong to the same class, which may not hold for tasks characterized by frequent class changes, such as speech recognition. To address this challenge, it is interesting to study using changing point detection algorithms (e.g., [23]) at class levels on the time sequences to automatically determine T. Furthermore, the exclusive consideration of time dimension characteristics might result in suboptimal performance for modalities with spatial dimensions, such as LiDAR and Wi-Fi. Exploring the definition of positives and negatives for CL based on spatial proximity [36] emerges as a potential direction for further investigation. Our future endeavors will focus on extending ContrastSense to diverse modalities and addressing these challenges to enhance its overall applicability.

8 Conclusion

To deal with domain shifts and class label scarcity simultaneously for in-the-wild wearable sensing, this paper proposes a novel domain-invariant CL framework, ContrastSense. ContrastSense can effectively utilize the unlabeled data from the source domains to extract high-level domain-invariant features with CDL and SInfo loss with negative selection. The domain-invariant encoder is fine-tuned with a parameter-wise penalty to preserve the domain-invariant knowledge learned with pretraining. Extensive experiments are conducted with different modalities, tasks, and domain heterogeneities. The results suggest that ContrastSense outperforms the state-of-the-art baselines in varied settings.

Acknowledgments

This research is supported in part by the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (MOE-T2EP20220-0004), and in part by Hong Kong GRF 16204224.

References

[1] Laura L Carstensen, Monisha Pasupathi, Ulrich Mayr, and John R Nesselroade. 2000. Emotional experience in everyday life across the adult life span. *Journal of personality and social psychology* 79, 4 (2000), 644.

[2] 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.

[3] 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.

[4] Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15750–15758.

[5] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. 2020. Debiased contrastive learning. *Advances in neural information processing systems* 33 (2020), 8765–8775.

[6] Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre Drouin, Alexandre Campeau-Lecours, Clément Gosselin, Kyrre Glette, François Laviolette, and Benoit Gosselin. 2019. Deep learning for electromyographic hand gesture signal classification using transfer learning. *IEEE transactions on neural systems and rehabilitation engineering* 27, 4 (2019), 760–771.

[7] Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research* 7 (2006), 1–30.

[8] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*. PMLR, 1180–1189.

[9] 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.

[10] Taesik Gong, Yewon Kim, Adiba Orzikulova, Yunxin Liu, Sung Ju Hwang, Jinwoo Shin, and Sung-Ju Lee. 2023. DAPPER: Label-Free Performance Estimation after Personalization for Heterogeneous Mobile Sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 7, 2 (2023), 1–27.

[11] Taesik Gong, Yeonsu Kim, Jinwoo Shin, and Sung-Ju Lee. 2019. Metasense: few-shot adaptation to untrained conditions in deep mobile sensing. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. 110–123.

[12] Yu Guan and Thomas Plötz. 2017. Ensembles of deep lstm learners for activity recognition using wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 2 (2017), 1–28.

[13] Yujiao Hao, Boyu Wang, and Rong Zheng. 2023. VALERIAN: Invariant Feature Learning for IMU Sensor-based Human Activity Recognition in the Wild. In *Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation*. 66–78.

[14] Yujiao Hao, Rong Zheng, and Boyu Wang. 2021. Invariant feature learning for sensor-based human activity recognition. *IEEE Transactions on Mobile Computing* 21, 11 (2021), 4013–4024.

[15] Harish Haresamudram, Irfan Essa, and Thomas Plötz. 2021. Contrastive predictive coding for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 2 (2021), 1–26.

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[18] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. 2016. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 5375–5384.

[19] Haruyuki Ichino, Katsuhiko Kaji, Ken Sakurada, Kei Hiroi, and Nobuo Kawaguchi. 2016. HASC-PAC2016: Large scale human pedestrian activity corpus and its baseline recognition. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct.* 705–714.

[20] Syed Anas Imtiaz. 2021. A systematic review of sensing technologies for wearable sleep staging. *Sensors* 21, 5 (2021), 1562.

[21] Yash Jain, Chi Ian Tang, Chulhong Min, Fahim Kawsar, and Akhil Mathur. 2022. ColloSSL: Collaborative self-supervised learning for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 1 (2022), 1–28.

[22] Seogkyu Jeon, Kibeom Hong, Pilhyeon Lee, Jewook Lee, and Hyeran Byun. 2021. Feature stylization and domain-aware contrastive learning for domain generalization. In *Proceedings of the 29th ACM International Conference on Multimedia.* 22–31.

[23] Sylvain Jung, Laurent Oudre, Charles Truong, Eric Dorveaux, Louis Gorintin, Nicolas Vayatis, and Damien Ricard. 2021. Adaptive change-point detection for studying human locomotion. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC).* IEEE, 2020–2024.

[24] Bulat Khaertdinov, Esam Ghaleb, and Stylianos Asteriadis. 2021. Contrastive self-supervised learning for sensor-based human activity recognition. In *2021 IEEE International Joint Conference on Biometrics (IJCB).* IEEE, 1–8.

[25] Md Abdullah Al Hafiz Khan, Nirmalya Roy, and Archan Misra. 2018. Scaling human activity recognition via deep learning-based domain adaptation. In *2018 IEEE international conference on pervasive computing and communications (PerCom).* IEEE, 1–9.

[26] 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.

[27] Alyssa Kubota, Tariq Iqbal, Julie A Shah, and Laurel D Riek. 2019. Activity recognition in manufacturing: The roles of motion capture and sEMG+ inertial wearables in detecting fine vs. gross motion. In *2019 International Conference on Robotics and Automation (ICRA).* IEEE, 6533–6539.

[28] Zhiping Lai, Xiaoyang Kang, Hongbo Wang, Xueze Zhang, Weiqi Zhang, and Fuhao Wang. 2022. Contrastive Domain Adaptation: A Self-Supervised Learning Framework for sEMG-Based Gesture Recognition. In *2022 IEEE International Joint Conference on Biometrics (IJCB).* 1–7.

[29] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. 2018. EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. *Journal of neural engineering* 15, 5 (2018), 056013.

[30] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. 2018. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[31] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. 2018. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 5400–5409.

[32] Yande Li, Taiqian Wang, Lian Li, Caihong Li, Yi Yang, Li Liu, et al. 2018. Hand gesture recognition and real-time game control based on a wearable band with 6-axis sensors. In *2018 international joint conference on neural networks (IJCNN).* IEEE, 1–6.

[33] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[34] Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* (2016).

[35] 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. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 2 (2022), 1–19.

[36] Wenjie Luo, Qun Song, Zhenyu Yan, Rui Tan, and Guosheng Lin. 2022. Indoor Smartphone SLAM with Learned Echoic Location Features. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems.* 489–503.

[37] Mohammad Malekzadeh, Richard G Clegg, Andrea Cavallaro, and Hamed Haddadi. 2019. Mobile sensor data anonymization. In *Proceedings of the international conference on internet of things design and implementation.* 49–58.

[38] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[39] Xiaomin Ouyang, Xian Shuai, Jiayu Zhou, Ivy Wang Shi, Zhiyuan Xie, Guoliang Xing, and Jianwei Huang. 2022. Cosmo: contrastive fusion learning with small data for multimodal human activity recognition. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking.* 324–337.

[40] Stefano Pizzolato, Luca Tagliapietra, Matteo Cognolato, Monica Reggiani, Henning Müller, and Manfredo Atzori. 2017. Comparison of six electromyography acquisition setups on hand movement classification tasks. *PloS one* 12, 10 (2017), e0186132.

[41] 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.

[42] Xin Qin, Jindong Wang, Yiqiang Chen, Wang Lu, and Xinlong Jiang. 2022. Domain Generalization for Activity Recognition via Adaptive Feature Fusion. *ACM Transactions on Intelligent Systems and Technology* 14, 1 (2022), 1–21.

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[44] Oona Rainio, Jarmo Teuho, and Riku Klén. 2024. Evaluation metrics and statistical tests for machine learning. *Scientific Reports* 14, 1 (2024), 6086.

[45] Anita Ramachandran and Anupama Karuppiah. 2020. A survey on recent advances in wearable fall detection systems. *BioMed research international* 2020, 1 (2020), 2167160.

[46] Nils Rethmeier and Isabelle Augenstein. 2023. A Primer on Contrastive Pretraining in Language Processing: Methods, Lessons Learned, and Perspectives. *Comput. Surveys* 55, 10 (2023), 1–17.

[47] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2020. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592* (2020).

[48] 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.

[49] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul JM Havinga. 2014. Fusion of smartphone motion sensors for physical activity recognition. *Sensors* 14, 6 (2014), 10146–10176.

[50] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. 2015. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM conference on embedded networked sensor systems*. 127–140.

[51] Qingyu Tan, Ruidan He, Lidong Bing, and Hwee Tou Ng. 2022. Domain Generalization for Text Classification with Memory-Based Supervised Contrastive Learning. In *Proceedings of the 29th International Conference on Computational Linguistics*. 6916–6926.

[52] Chi Ian Tang, Ignacio Perez-Pozuelo, Dimitris Spathis, Soren Brage, Nick Wareham, and Cecilia Mascolo. 2021. Selfhar: Improving human activity recognition through self-training with unlabeled data. *arXiv preprint arXiv:2102.06073* (2021).

[53] Chi Ian Tang, Ignacio Perez-Pozuelo, Dimitris Spathis, and Cecilia Mascolo. 2020. Exploring contrastive learning in human activity recognition for healthcare. *arXiv preprint arXiv:2011.11542* (2020).

[54] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[56] Jiwei Wang, Yiqiang Chen, Yang Gu, Yunlong Xiao, and Haonan Pan. 2018. SensoryGANs: An effective generative adversarial framework for sensor-based human activity recognition. In *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[57] 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.

[58] Jinqiang Wang, Tao Zhu, Liming Chen, Huansheng Ning, and Yaping Wan. 2023. Negative selection by clustering for contrastive learning in human activity recognition. *IEEE Internet of Things Journal* (2023).

[59] Jinqiang Wang, Tao Zhu, Jingyuan Gan, Liming Luke Chen, Huansheng Ning, and Yaping Wan. 2022. Sensor data augmentation by resampling in contrastive learning for human activity recognition. *IEEE Sensors Journal* 22, 23 (2022), 22994–23008.

[60] Xingchen Wang, Maria Kyrarini, Danijela Ristić-Durrant, Matthias Spranger, and Axel Gräser. 2016. Monitoring of gait performance using dynamic time warping on IMU-sensor data. In *2016 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. IEEE, 1–6.

[61] Crystal T Wei, Ming-En Hsieh, Chien-Liang Liu, and Vincent S Tseng. 2022. Contrastive heartbeats: Contrastive learning for self-supervised ECG representation and phenotyping. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1126–1130.

[62] Hao Wen, Yuanchun Li, Zunshuai Zhang, Shiqi Jiang, Xiaozhou Ye, Ye Ouyang, Yaqin Zhang, and Yunxin Liu. 2023. AdaptiveNet: Post-deployment Neural Architecture Adaptation for Diverse Edge Environments. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–17.

[63] Garrett Wilson, Janardhan Rao Doppa, and Diane J Cook. 2021. Calda: Improving multi-source time series domain adaptation with contrastive adversarial learning. *arXiv preprint arXiv:2109.14778* (2021).

[64] Huatao Xu, Pengfei Zhou, Rui Tan, and Mo Li. 2023. Practically Adopting Human Activity Recognition. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*. 1–15.

[65] Huatao Xu, Pengfei Zhou, Rui Tan, Mo Li, and Guobin Shen. 2021. LIMU-BERT: Unleashing the Potential of Unlabeled Data for IMU Sensing Applications. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*. 220–233.

[66] 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.

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

[68] Shibo Zhang, Yaxuan Li, Shen Zhang, Farzad Shahabi, Stephen Xia, Yu Deng, and Nabil Alshurafa. 2022. Deep learning in human activity recognition with wearable sensors: A review on advances. *Sensors* 22, 4 (2022), 1476.

[69] Xingxuan Zhang, Linjun Zhou, Renzhe Xu, Peng Cui, Zheyan Shen, and Haoxin Liu. 2022. Towards unsupervised domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 4910–4920.

[70] Ye Zhang, Longguang Wang, Huiling Chen, Aosheng Tian, Shilin Zhou, and Yulan Guo. 2022. IF-ConvTransformer: A framework for human activity recognition using IMU fusion and ConvTransformer. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 2 (2022), 1–26.

[71] Zengwei Zheng, Junjie Du, Lin Sun, Meimei Huo, and Yuanyi Chen. 2018. TASG: An augmented classification method for impersonal HAR. *Mobile Information Systems* 2018 (2018), 1–10.

[72] Zhijun Zhou, Yingtian Zhang, Xiaojing Yu, Panlong Yang, Xiang-Yang Li, Jing Zhao, and Hao Zhou. 2020. Xhar: Deep domain adaptation for human activity recognition with smart devices. In *2020 17th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON).* IEEE, 1–9.

A Appendix: The Degrees of Domain Shifts in Different Scenarios

Different cross-domain scenarios are considered in Section 6 to provide a comprehensive evaluation of ContrastSense and selected baselines. A quantitative analysis based on the Maximum Mean Discrepancy (MMD) is provided to show the degree of domain shifts between source and target domains across the considered datasets and scenarios. MMD is a widely used metric for measuring the difference in data distributions between domains [2]. A larger MMD value indicates a larger domain shift. The mean MMD between source and target domains over five random splits, along with the standard deviations, are measured and reported. The results for the IMU and EMG data are presented in Table 10 and Table 11.

Table 10. The mean MMD distances between source domains and target domains on the IMU datasets in different cross-domain scenarios. The standard deviations of the MMD value are included in the parentheses.

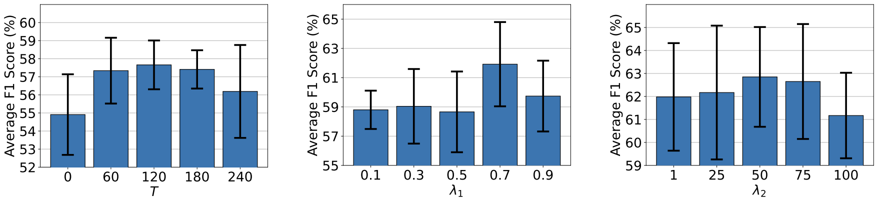
Scenario	Cross-user			
Dataset	HHAR	Motion	Shoaib	HASC
MMD	0.0445 (0.0178)	0.0105 (0.0081)	0.0069 (0.0033)	0.0351 (0.0447)
Scenario	Cross-device	Cross-position	Cross-user-device	Cross-user-position
Dataset	HASC	Shoaib	HASC	Shoaib
MMD	0.1044 (0.0771)	0.1773 (0.0881)	0.0258(0.0150)	0.0314 (0.0126)
Scenario	Cross-datasets			
Target Dataset	HHAR	Motion	Shoaib	HASC
MMD	1.1627 (0)	0.2940 (0)	0.2425 (0)	0.5940 (0)

In the cross-device, cross-position, cross-user-device, and cross-user-position scenarios, only one dataset is included for each, as explained in Section 6.5. In the cross-dataset scenarios, the standard deviations are consistently 0 because the source and target domains remain the same across different splits. For example, when the target domain is the HHAR dataset, the source domains are always the remaining three datasets.

As shown in Table 10, the MMD values for the HHAR and HASC datasets in the cross-user scenario are larger than those for the other two IMU datasets, indicating larger domain shifts in the HHAR and HASC datasets. This may explain the worse performance of methods on the HHAR and HASC datasets in Table 2 compared with the results on the other two datasets. The MMD values for the cross-device and cross-position scenarios are higher than those for the cross-user scenarios, suggesting that the discrepancies between data from different devices

Table 11. The mean MMD distances between source domains and target domains on the EMG datasets in different cross-domain scenarios.

Scenario	Cross-user		
Dataset	Myo	DB4	DB5
MMD	0.0577 (0.0254)	0.1579 (0.0916)	0.1427 (0.1325)
Scenario	Cross-datasets		
Target Dataset	Myo	DB4	DB5
MMD	0.4407 (0)	0.0782 (0)	1.3674 (0)



(a) The time window length T

(b) The CDL weight λ_1

(c) The penalty weight λ_2

Fig. 10. Sensitivity analysis with three key parameters in ContrastSense. The unit of T is the number of samples, which means the T samples around the query sample are excluded from the time-window selection. For example, when $T = 120$, the 60 samples ahead of the query sample x_k and the 60 samples after x_k would be excluded.

and on-body positions are larger than those between different users. The cross-dataset scenario encompasses multiple domain shifts simultaneously, resulting in a larger MMD distance compared with all other scenarios.

In Table 11, the MMD for the Myo dataset is smaller than that for the other two datasets in the cross-user scenario, which may contribute to better performance of methods on the Myo dataset compared with their performance on the other two datasets in Table 4.

B Appendix: Sensitivity Analysis

A sensitivity analysis is performed on HAR with IMU over several key parameter settings of ContrastSense, including the time window length T, the CDL weight λ_1, and the parameter-wise penalty weight λ_2. Similar results can be obtained for GR with EMG.

B.1 Impact of the Time Window Length

As shown in Fig. 10(a), a greater number of adjacent samples are excluded from the contrast process with a larger time window, which enhances the feature quality and improves the model performance. However, a larger time window also increases the risk of discarding samples from different classes and leads to a performance decrease, especially from classes that shift frequently in time-sequenced data. Specifically, while the optimal average F1 score across the four datasets is achieved with a time window of $t = 120$ (the value we adopt), the model shows the best performance on the HASC dataset at $t = 60$ due to its rapid motion transitions. Therefore, we recommend shorter time windows for datasets with frequent activity changes. The possibility of applying automatic time window length determination is further discussed in Section 7.

B.2 Impact of the CDL Weight

Fig. 10(b) shows the effect of the CDL weight λ_1. As λ_1 increases, the model acquires more domain-invariant knowledge, enabling it to handle domain shifts more effectively. However, when λ_1 exceeds 0.7, the model's performance deteriorates. A possible reason is that if the model prioritizes optimizing CDL, it may learn inadequate high-level features from the unlabeled data for the downstream tasks. ContrastSense requires an appropriate weight λ_1 for the learning with CDL and SInfo loss to ensure the production of sufficient high-level features for the downstream tasks, as well as domain-invariant features to address the domain shifts.

B.3 Impact of the Parameter-wise Penalty Weight

The results displayed in Fig. 10(c) show the performance of ContrastSense with varied weight λ_2 for the parameter-wise penalty. The parameter-wise penalty acts as a constraint during fine-tuning to prevent the loss of domain-invariant knowledge acquired with CDL. A larger λ_2 imposes a greater penalty on adjusting the parameters in the encoders that are crucial for domain-invariant knowledge. However, if λ_2 continues to rise, the penalties for all parameters become overly high. Consequently, the model may become inadequately tuned for the tasks.

C Appendix: Statistical Tests on the Main Results

We conducted comprehensive statistical tests to rigorously evaluate the performance of ContrastSense. Friedman's test [7] was first performed to compare the overall differences among all methods. This was followed by Wilcoxon signed-rank tests [7] to assess the pairwise differences between ContrastSense and each baseline method, as recommended by prior works [7, 44]. These tests were applied to the F1 scores reported in the key results of this paper, covering Table 2, Table 4, Table 6, Table 7, Figure 5, Figure 6, Figure 8, and Figure 9. The test statistics and corresponding p-values are summarized in Table 12 and Table 13. Each row in the Wilcoxon signed-rank test results represents the comparison between ContrastSense and a specific baseline method.

Table 12. The statistical tests compare the F1 scores of ContrastSense with the baselines on the IMU datasets. The values outside the brackets represent the test statistics, while the values inside the brackets indicate the corresponding p-values.

Statistical Test	Method	Table 2	Table 6	Fig. 5	Fig. 8	Fig. 9
Friedman's test	–	118 (3.0e-16)	19.0 (2.7e-4)	1.6e3 (3.6e-197)	505.9 (2.7e-67)	500.7 (2.7e-66)
Wilcoxon test	ColloSSL	210 (9.5e-7)	–	–	–	–
	LIMU	208 (2.9e-6)	10 (6.3e-2)	2.8e4 (9.5e-35)	2.5e3 (2.1e-13)	2.5e3 (1.7e-13)
	Mixup	199 (5.2e-5)	10 (6.3e-2)	2.6e4 (2.3e-28)	2.5e3 (3.1e-13)	2.6e3 (1.4e-13)
	GILE	210 (9.5e-7)	10 (6.3e-2)	2.9e4 (2.3e-41)	2.5e3 (9.0e-12)	2.5e3 (1.0e-12)
	FMUDA	210 (9.5e-7)	10 (6.3e-2)	2.8e4 (2.2e-37)	2.5e3 (1.2e-12)	2.5e-3 (3.5e-12)
	CMUDA	210 (9.5e-7)	10 (6.3e-2)	2.8e4 (3.5e-38)	2.5e3 (1.7e-12)	2.5e3 (2.2e-12)
	CPCHAR	208 (2.9e-6)	10 (6.3e-2)	2.6e4 (4.9e-27)	2.4e3 (1.3e-10)	2.4e3 (1.4e-10)
	ClusterHAR	195 (1.3e-4)	10 (6.3e-2)	2.8e4 (5.1e-37)	2.6e3 (1.2e-13)	2.6e3 (1.4e-13)

As shown in Table 12 and Table 13, the test statistic is 118, with a p-value of 3.0e-16 for the F1 score results in Table 2, indicating more than 99.9% confidence that the methods perform differently. The Wilcoxon signed-rank test further reveals that there is more than 99.9% confidence that ContrastSense outperforms the baseline methods

Table 13. The statistical tests compare the F1 score of ContrastSense and the baselines on the EMG datasets. The values outside the brackets represent the test statistics, while the values inside the brackets indicate the corresponding p-values.

Statistical Test	Method	Table 4	Table 7	Fig. 6
Friedman's test	–	49.3 (8.0e-6)	1.5 (0.472)	610.6 (1.6e-48)
Wilcoxon test	Mixup	118 (9.2e-5)	6.0 (0.125)	1.4e4 (1.5e-16)
	ConSSL	120 (3.0e-5)	6.0 (0.125)	1.6e4 (1.4e-31)
	CALDA	118 (9.2e-5)	6.0 (0.125)	1.4e4 (1.5e-16)

in Table 2, with an average F1-score improvement of 8.9% over the best baseline (Mixup). Similar conclusions are drawn from the other results. The smallest p-value in Table 12 and Table 13 is 0.125, derived from the results in Table 7. Although ContrastSense shows a 4.3% average F1-score improvement over Mixup in Table 7, its improvement on the DB5 dataset is less pronounced. This may be due to Mixup's effective augmentation of samples that better approximate the data distribution of the DB5 dataset. Future work will explore enhanced data augmentation techniques to further integrate them into the ContrastSense pipeline. Additionally, the p-value for Friedman's test in Table 7 is 0.472, likely due to the limited number of baselines and test cases in the cross-dataset scenarios. Since Friedman's test compares the ranks of different methods, a smaller number of methods (in this case, only four) results in less pronounced rank differences. Overall, these statistical tests demonstrate that ContrastSense consistently outperforms state-of-the-art baselines.

D Appendix: Results Comparison on Each Dataset

In the main paper, the average F1 score is reported on the IMU datasets and EMG datasets in Fig. 5 and Fig. 6, respectively. To provide a detailed performance comparison, the results of ContrastSense and baselines on each dataset are presented in Fig. 11, Fig. 12, Fig. 13, Fig. 14, Fig. 15, and Fig. 16[1].

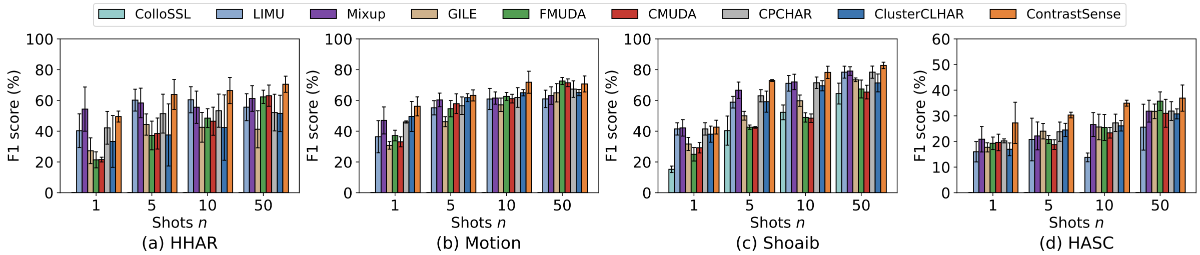
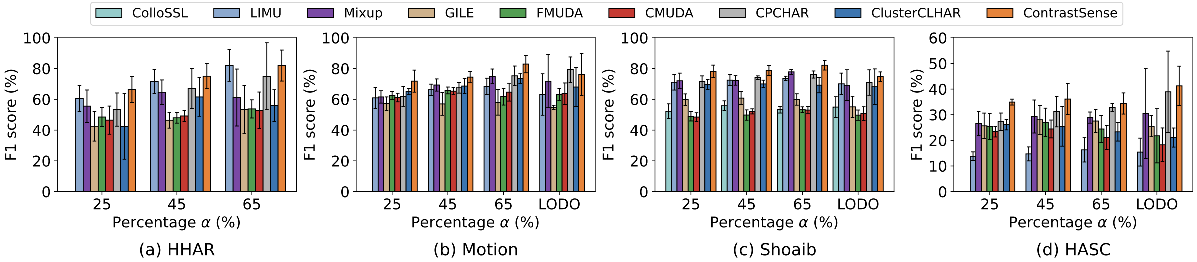


Fig. 11. Performance comparison with varied shots n on the cross-user experiment for HAR with IMU.

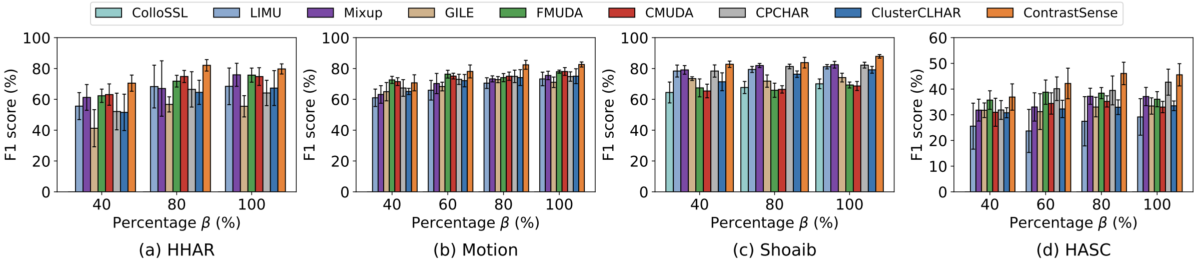
As shown in Fig. 11, Fig. 12, Fig. 13, Fig. 14, Fig. 15, and Fig. 16, ContrastSense outperforms the baselines in most cases and its average performance on the IMU and EMG datasets is consistently better (shown in Fig. 5). On the HHAR dataset, when β is 40%, 80%, and 100%, ContrastSense achieves F1 scores that are 7.4%, 7.2%, and 3.8% higher than the best baseline, respectively. We notice that some baselines may perform better than ContrastSense in few cases, e.g., Mixup outperforms ContrastSense when $n = 1$ on the HHAR dataset. The reason

[1] In Fig. 12, we omit results in the LODO setting on the HHAR dataset since there is only one test domain when $\alpha = 65\%$, which corresponds to the LODO setting. Similarly, in Fig. 13, results are excluded for HHAR and Shoaib datasets when $\beta = 60\%$, which is based on the fact that the size of the labeled source domains remains consistent (one domain) at $\beta = 60\%$ and $\beta = 40\%$.



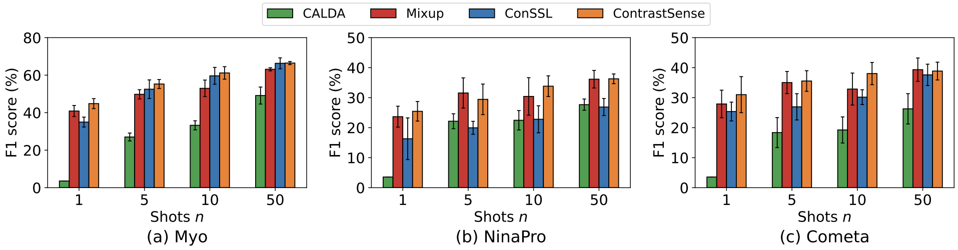
(a) HHAR (b) Motion (c) Shoaib (d) HASC

Fig. 12. Performance comparison with varied percentages α on the cross-user experiment for HAR with IMU.



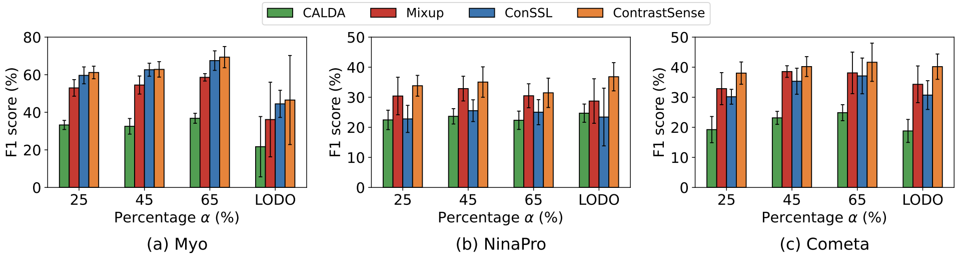
(a) HHAR (b) Motion (c) Shoaib (d) HASC

Fig. 13. Performance comparison with varied percentages β on the cross-user experiment for HAR with IMU.



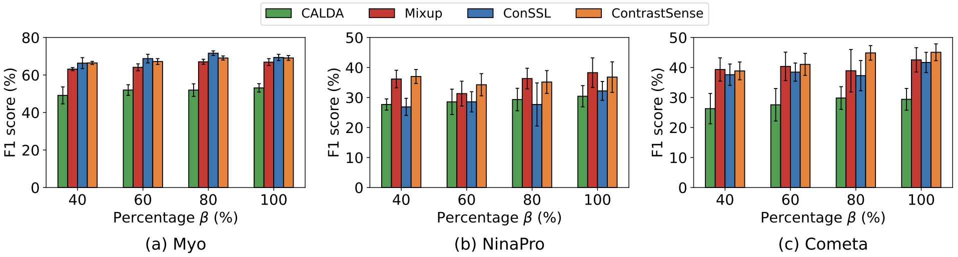
(a) Myo (b) NinaPro (c) Cometa

Fig. 14. Performance comparison with varied shots n on the cross-user experiment for GR with EMG.



(a) Myo (b) NinaPro (c) Cometa

Fig. 15. Performance comparison with varied percentages α on the cross-user experiment for GR with EMG.



(a) Myo

(b) NinaPro

(c) Cometa

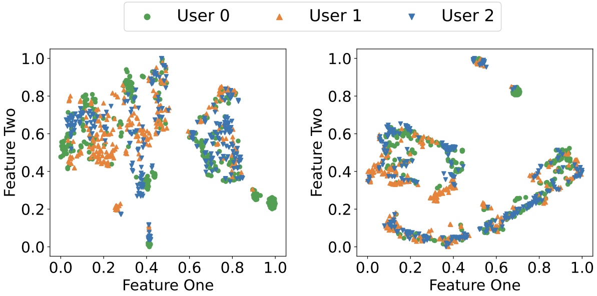
Fig. 16. Performance comparison with varied percentages β on the cross-user experiment for GR with EMG.

might be that Mixup augments the limited labeled data to more labeled data with larger diversity, which improves the model performance when presented with larger label scarcity, whereas ContrastSense does not leverage data augmentation techniques in the fine-tuning stage to improve data diversity. In the future, we will further investigate the usage of data augmentation techniques or learning-based data synthesis [35, 56] to enhance the real-life deployability of our method. Besides, we notice that in Fig. 15 on the Myo dataset, the performance of ContrastSense and baselines in the LODO setting are worse than with a smaller α. The reason might be users with significantly different data distributions are randomly selected for test in some splits in the LODO setting. Besides, we have conducted the statistical tests on those results in Appendix C, which suggest that ContrastSense consistently outperforms those baselines in those settings. Please refer to Appendix C for more details about the statistical tests.

E Appendix: Additional Ablation Study

More detailed ablation studies are conducted on the IMU datasets to present the effectiveness of the CDL and the negative selection. Similar results can be obtained on the EMG dataset.

E.1 Effectiveness of the CDL



(a) w/o CDL

(b) w/ CDL

Fig. 17. Visualization of features from three users. (a) The features learned without adopting CDL are separate, and (b) the features learned with adopting CDL are well aligned.

To further understand the effect of CDL, the representations extracted without CDL and with CDL are visualized in Fig. 17. It can be seen that the features of the three users are separate from each other due to the domain discrepancy when trained only with InfoNCE loss. After CDL is incorporated, the features are aligned, which further validates the effectiveness of CDL in extracting generalizable features.

E.2 Effectiveness of the Negative Selection

Table 14. Effectiveness of the negative selection. All methods utilize the samples in the domain queues for a fair comparison.

Design	HHAR		Motion		Shoaib		HASC		Average		Weighted	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
W/ Both	**64.66**	**60.79**	**71.02**	**69.34**	**76.25**	**76.02**	**43.77**	**33.85**	**63.92**	**60.00**	**64.71**	**61.01**
W/ time window selection	58.76	55.65	69.68	67.49	75.96	75.53	41.39	31.98	61.45	57.66	62.27	58.74
W/ similarity-based selection	64.24	60.48	69.45	67.58	71.95	71.65	40.86	32.41	61.63	58.03	62.20	58.81
domain-wise InfoNCE	61.35	59.27	69.24	66.84	72.08	71.39	40.88	32.64	60.89	57.53	61.44	58.34
InfoNCE	54.94	51.17	69.98	68.13	71.19	70.20	37.83	30.13	58.48	54.91	58.80	55.36

An ablation study is further conducted to highlight the impact of the two steps involved in the negative selection process for SInfo loss in Table 14. Compared with InfoNCE loss, domain-wise InfoNCE selects samples within the same domain as the negatives, which prevents the model from capturing the domain-related knowledge. It improves the average F1 score by 2.6% compared with using InfoNCE loss. In contrast, the similarity-based selection in ContrastSense retains some negatives from different domains that are difficult to distinguish from positives. Consequently, it yields an average F1 score of 58.03% and an average accuracy of 61.63%, surpassing domain-wise InfoNCE. Domain-wise InfoNCE overlooks valuable information embedded in samples from various domains, whereas the similarity-based selection effectively harnesses this information. When only the time window selection is adopted, the average performance increases by 2.8% in terms of F1 score and 3.0% in terms of accuracy, underscoring the positive impact of excluding adjacent samples to improve feature quality during pretraining. When both steps are synergistically combined, the average F1 score and accuracy exhibit improvements of 5.1% and 5.4% respectively, compared with the results obtained using InfoNCE loss. This combination solidifies the effectiveness of both steps in enhancing the overall performance of the model.

Received 1 May 2024; revised 1 August 2024; accepted 20 Sept 2024