Dual-Stream Contrastive Learning for Channel State Information Based Human Activity Recognition

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

19-02-2022 / 01-03-2022

CITATION

Xu, Ke; Wang, Jiangtao; Zhang, Le; Zhu, Hongyuan; Zheng, Dingchang (2022): Dual-Stream Contrastive Learning for Channel State Information Based Human Activity Recognition. TechRxiv. Preprint.
<https://doi.org/10.36227/techrxiv.19200440.v1>

DOI

[10.36227/techrxiv.19200440.v1](https://doi.org/10.36227/techrxiv.19200440.v1)

Dual-Stream Contrastive Learning for Channel State Information Based Human Activity Recognition

Ke Xu, Jiangtao Wang, Le Zhang, Hongyuan Zhu, Dingchang Zheng

Abstract—WiFi-based human activity recognition (HAR) has been extensively studied due to its far-reaching applications in health domains, including elderly monitoring, exercise supervision and rehabilitation monitoring, etc. Although existing supervised deep learning techniques have achieved remarkable performances for these tasks, they are however data-hungry and hence are notoriously difficult due to the privacy and incomprehensibility of WiFi-based HAR data. Existing contrastive learning models, which are mainly designed for computer vision, cannot guarantee their performance on channel state information (CSI) data.

To this end, we propose a new dual-stream contrastive learning model, that can process and learn the raw WiFi CSI data in a self-supervised manner. More specifically, our proposed method, coined as DualConFi, takes raw WiFi CSI data as input and consists of a channel stream and a temporal stream to learn highly-discriminative spatiotemporal features under a mutual information constraint using unlabeled data. We demonstrate the efficacy of our model on three publicly available CSI data sets in various experiment settings, including linear evaluation, semi-supervised, and transfer learning. We show that DualConFi is able to perform favorably against challenging baselines in each setting. Moreover, by studying the effects of different transform functions on CSI data, we finally verify the effectiveness of highly-discriminative features.

Index Terms—human activity recognition, Self-supervised Learning, Channel state information

I. INTRODUCTION

Human activity recognition (HAR) aims to identify various human activities automatically in a diverse range of human-centric applications, especially in healthcare domains and tasks, such as elderly monitoring [1], fall detection [2], sign language recognition [3] and respiration monitoring [4]. For example, it can be used to detect and analyze the activities of the elderly to predict potential health issues. It can also be used for gesture recognition to provide service for people with hearing loss. Besides, it can be used for early diagnosis for people living alone who are at risk of chronic respiratory diseases. These applications are based on the belief that

This paragraph of the first footnote will contain the date on which you submitted your paper for review. It will also contain support information, including sponsor and financial support acknowledgment. For example, "This work was supported in part by the U.S. Department of Commerce under Grant BS123456."

The next few paragraphs should contain the authors' current affiliations, including current address and e-mail. For example, F. A. Author is with the National Institute of Standards and Technology, Boulder, CO 80305 USA (e-mail: author@boulder.nist.gov).

Fig. 1. A human's activity brings changes to the dynamic reflection propagation path which lead to unique variations on WiFi signals. The static reflection path is caused by static objects while the static direct path follows line-of-sight (LoS).

knowing one's real-time context can significantly improve people's safety, efficiency, and quality of life. Due to its importance, extensive research efforts have been devoted to HAR in recent years, including contact-based and contactless activity recognition [5], [6]. Contact-based approaches, such as using gyroscopes to detect falls [7], only work when users wear sensors. Compared to contact-based approaches, contactless activity recognition has the advantage of non-intrusiveness, which uses audio [8], infrared sensors [9] or cameras [10]. However, there are some existing shortcomings of these contactless approaches, including sensitivity to working conditions, requirements for dedicated devices and privacy issues. E.g., cameras have illumination requirements, need line-of-sight, and can be risky to personal privacy.

To fill this gap, the omnipresent WiFi signals have been investigated to provide activity information for HAR. Because WiFi-based methods use off-the-shelf devices and are more privacy-friendly, they significantly improve the usability and practicality of indoor HAR applications and have been increasingly popular. They employ existing WiFi devices to pinpoint the physical presence of a user directly by passively learning the changes in WiFi propagation, building upon the principle that human actions between WiFi transmitters and receivers(as shown in Fig. 1) incur subtle yet unique variations (a.k.a. multi-path and the fading effect) [11].

The most commonly used WiFi signal data is channel state information because it is fine-grained and containing abundant environmental information that would benefit HAR tasks.

However, it is hard to obtain satisfactory results solely relying on extracting simple yet effective patterns because the CSI signals are often incomprehensible and noisy. Based on this observation, recent researchers widely utilize deep learning methods to process CSI-based HAR tasks [3], [12]. Deep learning technology can automatically extract useful patterns underneath raw data and improve performance without expert knowledge.

However, recent success is owing to supervised learning models, which require an enormous amount of well-labeled data to solve the desired task. Because of the privacy issue of personal activity data, it is hard to collect sufficient user-activity instances. Moreover, compared with images (used in computer vision), the raw CSI data is not readable for humans, which makes collect then post-label unlabelled data far more challenging. In addition, most labeled CSI datasets are obtained in well-controlled laboratory settings and hence might not generalize well to real-living context which covers a long-tail distribution.

The unsupervised pre-training of models is highly successful in many machine learning applications in the situations of insufficient labels. One of unsupervised methods is contrastive learning, which aims to learn useful representation without human annotations. A common paradigm of contrastive learning methods is that they aim to learn representations that are invariant under different data augmentations. This is typically achieved by maximizing similarity of representations obtained from different data augmentations of a sample using the same network. Those data augmentations can be easily achieved through predefined rules without human annotation. Contrastive learning has emerged as a popular and promising approach for overcoming the limitations associated with the lack of labels in computer vision [13]. However, applying contrastive learning to CSI data is still difficult since we need to address two challenges.

Challenge 1: existing contrastive learning model ignores the useful time-over-channel (*representing the time dependence of activity*) and channel-over-time features (*representing the reflecting path of WiFi signal of different frequency*) contains exact temporal and spatial information for downstream HAR tasks, according to Fresnel model [14].

Challenge 2: existing works mainly focus on the computer vision while the effect of different transform functions is still unclear for CSI data's contrastive learning.

To address these issues, we propose a novel mutual information enhanced dual-stream contrastive learning model to exert the advances of contrastive learning techniques in HAR task. Our model contains two streams of contrastive learning (i.e., contrastive learning for temporal information and spatial information) to extract highly-discriminative representations for activity classification. In addition to using contrastive learning to enforce the alignment and uniformity of features [15], a mutual information (MI) constraint is also employed in our model to ensure that the learned features retain sufficient semantic information from the input. Moreover, we also propose a novel transform function that is specific to channel stream of CSI data for extracting spatial information. Finally, We also conduct extensive evaluations to demonstrate

the effect of different transform function combinations on both temporal and channel stream of CSI data.

In summary, our main contributions are as follows:

- We propose a novel MI enhanced dual-stream contrastive learning model for CSI-based HAR task. To the best of our knowledge, we are the first work which utilize both temporal and channel information with contrastive learning on sensor data.
- We analyse the performance of different transform function combinations on CSI-based HAR task and designed a novel transform function to effectively extracts spatial information.
- We demonstrate through extensive evaluation based on three open datasets that our method is superior to state-of-the-art (SOTA) baselines on multivariate time series in supervised, semi-supervised and transfer learning scenarios.

II. RELATED WORK

Contrastive channel state information (CSI) human activity recognition (HAR) involves two aspects: contrastive learning and CSI-based HAR. Next, we discuss related studies from these two aspects.

A. CSI-based HAR

Researchers have proposed a variety of methods on this task. These methods can be divided into traditional and deep learning-based methods. Traditional methods rely on hand-craft feature extraction, while deep learning-based methods are data-driven.

Traditional CSI HAR methods can be divided into two types: pattern-based methods and model-based methods. Model-based methods utilize the mathematical or physical model that depicts the unique relationship between CSI signal variation and human behaviors. For example, Hao et al. propose an indoor localization approach based on the Fresnel model [14]. Pattern-based methods usually extract discriminative and representative features manually and feed them into the model for specific downstream tasks. Some of the popular techniques are support vector machines (SVM) [16] and hidden Markov model (HMM) [11].

In another recent line of work, deep learning-based methods are standing at the forefront due to their automatic feature extraction with full annotation. Yousefi et al. utilized LSTM to extract temporal information in CSI-based HAR data [17]. On this basis, Chen et al. argued that traditional LSTM networks ignored the backward direction's information during prediction and proposed ABLSTM for HAR with WiFi CSI [12]. Li et al. [18] showed that existing methods treated each frame as a single temporal point which ignores the informative context and only considered channel-over-time features while disregarding the time-over-channel features. Hence, they proposed THAT and gained SOTA performance.

The aforementioned works are supervised-learning-based, which still suffer from insufficient labelled data. Although our intuition comes from THAT, the major difference is that we extend it with unsupervised learning, which can utilize

Fig. 2. Overview of our approach. We use two encoders $f_t(\cdot)$ and $f_c(\cdot)$ and projection head $g(\cdot)$ to optimize the contrastive loss $NT-Xent$; while two encoders and the MI estimator $m(\cdot)$ are used to maximize the MI between two streams. Without loss of generality, we use two X s to denote the input for both views because they could come from different instances. We also use t and t' to indicate different transform functions caused by randomization augmentations as described in Section III.

massive unlabeled data to fill this gap. A classical illustration of unsupervised feature learning is the autoencoder, which learns to map an input onto a lower-dimensional embedding so that reconstructing the original input from such a space incurs a lower error. Gao et al. developed a sparse auto-encoder network for CSI-based localization and activity recognition [19]. Although there are some works based on unsupervised learning, how to effectively implement contrastive learning in CSI-based HAR is still under-explored. To address this concern, we propose a novel dual-stream contrastive learning model for CSI data.

B. Contrastive Learning

Lately, contrastive-based self-supervised methods achieve strong performances close to the supervised baseline in computer vision [13], [20]. The fundamental idea is that they use instance-wise supervision without human annotation to learn meaningful embeddings.

The first common form of contrastive learning is parallelly producing two noise versions of one anchor instance (image in computer vision), aiming to maximize the similarity of representations of these two augmented versions while minimizing those augmented versions from different instances [21], [13], [22]. These methods train a Siamese networks using augmented views generated by applying transform functions on each instances of a batch with contrastive loss.

Another popular way is building dictionary explicitly or implicitly. [23] maintain a memory bank whose tokens are sampled from data and are represented by an encoder network. The encoder network is trained to perform dictionary lookup: an encoder “query” should be similar to its matching key and dissimilar to others. In another recent line of work, [20] adopt a momentum encoder as a way of building large and consistent dictionaries. They use asymmetric learning updates wherein momentum encoders are updated separately from the main network.

Additionally, methods based on feature clustering use one augmented version of a sample to compute codes for the loss, and another augmented version of the same sample to predict these codes [24].

Compared to building a dictionary and clustering-based methods, the implementation of the first way is more straightforward. However, it has a drawback: the first way requires a huge batch size to attain a competitive result, which demands enormous GPU memory. Our model adopts the first way because of its simplicity.

III. METHOD

In this section, we firstly analyze the critical factors in data augmentation for CSI data and provide the details of our proposed transformation functions. Finally, we elaborate on the dual-stream contrastive learning framework as well as the network architectures.

We first briefly present the notations and background knowledge before elaborating the proposed method. We assume that we have access to N CSI records $\{X^i\}_{i=1}^N$. Each record $X^i = [x^{i1}, x^{i2}, \dots, x^{iC}]$ is presented in a $T \times C$ dimensional space, where T and C stands for the number of time length and channels, respectively. We also have two sets of randomized transform functions (i.e. $t_t, t'_t \sim \mathcal{T}_t$ and $t_c, t'_c \sim \mathcal{T}_c$) over the temporal and channel stream, respectively. The input X is processed by those transform functions and generate $4N$ views (4 for each record), which are denoted by $v_{l_t}, v_{l_c}, v_{m_t}, v_{m_c}$ ($l \in [1, N], m \in [N+1, 2N]$) respectively.

On top of that, we have the temporal based encoders $f_t(\cdot)$ to further encode v_{l_t}, v_{m_t} and yield the final temporal representation of $h_{l_t}, h_{m_t} \in \mathbb{R}^{d_t}$. In the same way, we use the channel based encoders $f_c(\cdot)$ to yield the final channel representation of $h_{l_c}, h_{m_c} \in \mathbb{R}^{d_c}$. After concatenating temporal and channel representations, a projection head of $g(\cdot)$ is utilized to get the final embedding $r \in \mathbb{R}^{d_t+d_c}$. The system illustration is present in Fig. 2.

Fig. 3. Multi-scale Convolution Augmented Transformer.

A. Data augmentation

Data augmentation or transformation is an important component in contrastive learning. Practical transform functions should force the network to learn task-related semantic features. By way of illustration, spatial information and temporal information are essential for HAR tasks.

It is straightforward to apply temporal flip or segmentation to extract temporal information for a sequential signal like CSI. However, the spatial information of CSI is not presented in a direct way like the continuity between pixels of video-based HAR data. Fresnel Penetration Model [14] reveals that the spatial information in the CSI data is reflected in the form of static propagation paths and dynamic propagation paths of the WiFi signal, which is related to patterns between channels. Similar to video-based methods, which extract spatial information using transform functions by focusing on continuity between pixels (like jigsaw [25]), our transform function acts as a cross-channel encoder to learn the intrinsic channel pattern related to the signal propagation paths (i.e., spatial information).

More specifically, we consider the following transform functions:

- **Channel Sobel** (ours): We randomly select channels from the record X^i and process them with 1D Sobel operator to make the spatial augmentations consistent along the temporal dimension with unprocessed channels. Mathematically, we have $X = \{x^{i1}, x^{i2}, \dots, \hat{x}^{is}, \dots, x^{iC}\}$, where $\hat{x}^{is} = [1, 0, -1] * x^{is}$.
- **Noise**: Random noise signals with a mean of zero and a standard deviation of -0.05 are added to the data sample.
- **Negate**: The values of the sample are multiplied by a factor of -1 .
- **Time flip**: The entire window of the sample is flipped in the time direction.
- **Time permutation (Time perm.)**: The signal is segmented into four different sections, and a random permutation is performed on the segments and then recombined.

As shown in [13], a composite transform function could usually yield more complex augmentation effects, which are beneficial for the final performance. Motivated by this phenomenon, we consider our transformation as a composite augmentation function which consists of a cascade of two transform functions shown above. We first compare the average performance of each transform function in one stream when the other stream undergoes different transform functions. Our rationale is that, a good transform function should generalize well when the other stream has different augmented views. Through this process, we pick the top two-ranked transform functions for each stream. Finally, for each stream,

TABLE I
DATASET STATISTICS.

Dataset	# Rec.	# Classes.	# Channels.
Office Room	557	7	30×3
Falldefi	581	2	30×3
Signfi	2760	276	30×3

we further study the effect on the order for applying those two functions. We empirically found that this strategy works well in our framework and more advanced designs on the composite augmentations, for instance, consisted of more transform functions, are absolutely compatible with our framework but beyond the scope of the current study.

Through this process, we identify two composite augmentations finally: noise followed by time flip for temporal stream and channel Sobel followed by noise for channel stream, respectively. More ablative results on this part are summarized in Section IV-F.2.

B. DualConfi Contrastive Learning

Before being processed by our model, aforementioned functions transform given N instances into $2N$ temporal-channel view tuples, that is, (v_t, v_c) and (v_{m_t}, v_{m_c}) . We consider view tuples of the same instance as positive pairs, while view tuples of different instances are negative pairs. For example, if $(v_t = t_t(X^i), v_c = t_c(X^i))$ and $(v_{m_t} = t_t(X^i), v_{m_c} = t_c(X^i))$, we consider $\{(v_t, v_c), (v_{m_t}, v_{m_c})\}$ as a positive pair.

1) **Architecture**: Next, we provide the detail the specific implementation of our neural network architecture. There are two encoders in our model. Both of them are built up with Multi-scale Convolution Augmented Transformer (MCAT) layers proposed in [18] and 1D convolution layers. Fig. 3 demonstrate the detail of MCAT layer. There are two sequentially stacked sub-layers in an MCAT layer: (1) a multi-head self-attention, and (2) a multi-scale CNN. Each of the two sub-layers is encompassed by a residual connection [26] and a layer normalization [27].

The temporal encoder $f_t(\cdot)$ extracts feature along temporal axis. Starting with a position encoding module [18], $f_t(\cdot)$ consists of five MCAT layers whose multi-head attention sub-layer has 5 heads and dimensionality equals 90. The Channel encoder $f_c(\cdot)$ extracts feature along channel axis. $f_c(\cdot)$ consists of one MCAT layer whose multi-head attention sub-layer has 200 heads and dimensionality equals 2000. We emphasize that the model design is not our aim. The temporal and channel encoder can be replaced by any other backbone for CSI data self-supervised learning.

On top of base encoders, the projection head is a two-layer multilayer perceptron with 288 and 64 neurons, respectively. The first layer is followed by a ReLU activation function. Likewise, the classifier for HAR task is a single-layer multilayer perceptron with k neurons, where k denotes activities' number. Finally, our mutual information estimator $m(\cdot)$ is a neural network containing three fully connected layers with 16, 16, and 1 neurons, respectively.

TABLE II
LINEAR SEPARABILITY ON OFFICE ROOM, FALLDEFI, AND SIGNFI.

Models	Supervision	Office Room		Falldefi		Signfi	
		Accuracy	F1	Accuracy	F1	Accuracy	F1
Supervised	Supervised	94.11	94.13	73.83	73.53	84.28	83.61
Random init.	Unsupervised	15.26	6.32	49.49	40.74	0.32	0.00
AutoEncoder	Unsupervised	56.97	57.22	57.91	57.09	26.16	27.23
Rocket	Unsupervised	84.80	84.80	60.85	59.03	43.84	41.00
Causal CNN	Unsupervised	76.78	76.77	67.86	67.52	44.02	42.49
DualConFi (ours)	Unsupervised	90.14	90.11	74.87	73.86	55.25	53.15
DualConFi-FT (ours)	Unsupervised	94.64	94.64	80.17	80.00	87.44	86.55

2) *Optimization*: The final loss function for each pair (i, j) is defined as:

$$\mathcal{L}_{l,m} = NT\text{-}Xent_{l,m} + \beta(I_l + I_m) \quad (1)$$

where β is the MI coefficient and $\beta \leq 0$. When $\beta < 0$, the model is trained to maximize MI; when $\beta = 0$, no MI constrain is applied. The second and first items on the right side of the equation are mutual information and contrastive loss, which will be introduced following.

a) Mutual information: MI can be expressed as

$$I(X; Z) := H(X) - H(X|Z), \quad (2)$$

where H is the Shannon entropy, and $H(X|Z)$ is the conditional entropy of Z given X . The InfoMax principle [28], as prescribed for neural networks, advocates maximizing MI between the input and representation to preserves maximum information:

$$\max_{f_1 \in \mathcal{F}_1, f_2 \in \mathcal{F}_2} I(X; (f_1(X), f_2(X))) \quad (3)$$

where X is the given input while $f_1(\cdot) = f_t(t_t(\cdot))$ and $f_2(\cdot) = f_c(\text{transpose}(t_c(\cdot)))$ generate representations in our case. Former work [29] gives the theoretical proof of that the MI between $f_1(X)$ and $f_2(X)$ is the lower bound of $\max_{f_1 \in \mathcal{F}_1, f_2 \in \mathcal{F}_2} I(X; (f_1(X), f_2(X)))$:

$$I(f_1(X); f_2(X)) \leq I(X; (f_1(X), f_2(X))). \quad (4)$$

Hence we can maximize the mutual information between two stream's representations to maximize the MI between the input and output. We apply a neural networks as mutual information estimator $m(\cdot)$, which accepts h_{l_t} and h_{l_c} as input to calculate the mutual information between h_{l_t} and h_{l_c} :

$$I(h_{l_t}; h_{l_c}) = m(h_{l_t}, h_{l_c}). \quad (5)$$

b) Contrastive loss Finally, after concatenating h_{l_t} with h_{l_c} , we feed the output of concatenating into the projection head and get the final embedding:

$$r_l = g(\text{concat}(h_{l_t}, h_{l_c})),$$

The projection head $g(\cdot)$ is used to project the representations in the latent space into another embedding space for calculating contrastive loss between each view pair (l, m) . The contrastive loss function, which defines the contrastive prediction task, accepts the final embedding r_l and r_m as

input. In our case, the *NT-Xent* (the normalized temperature-scaled cross entropy loss) [13] was adopted. Let $\text{sim}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\top \mathbf{v} / \|\mathbf{u}\| \|\mathbf{v}\|$ denote the dot product between l_2 normalized $\mathbf{u}$ and $\mathbf{v}$ (i.e. cosine similarity). For a pair of views (l, m) , the contrastive loss function is:

$$NT\text{-}Xent_{l,m} = -\log \frac{\exp(\text{sim}(r_l, r_m))}{\sum_{k=1}^{2N} \mathbb{I}_{[k \neq l]} \exp(\text{sim}(r_l, r_k) / \tau)} \quad (6)$$

where $\text{sim}(r_l, r_m)$ denotes the cosine similarity between the embedding vectors r_l and r_m , $\mathbb{I}_{[k \neq l]} \in [0, 1]$ is an indicator function evaluating to 1 iff $k \neq l$.

IV. EXPERIMENTS

In this section, we perform a comprehensive evaluation of our framework on three publicly available CSI-based HAR datasets to determine the quality of learned representations, data efficiency, and transferability of the features. Furthermore, we also conduct experiments for comparing different transform functions' performance and evaluating modules' contributions.

A. Training and Baselines

We compare our framework with following baselines:

- **Random init.**: It randomly initializes parameters and freeze the model except for the last layer classifier.
- **Supervised**: It trains **THAT** [18] model in a fully supervised learning manner with default setting.
- **AutoEncoder**: It is trained the using reconstruction loss (mean squared error). We follow the structure in [30]. We use Adam as our optimizer with a learning rate of 0.001. The learning rate is exponentially decayed by 0.5 $\times$, if validation loss does not decrease for 10 epochs. We use a mini-batch size of 32 in the GPU.
- **Rocket [31]**: It is a non-deep learning baseline, which uses a massive amount (10000 in our experiment) of random generated convolutional kernels to process multivariate time series data. Its convolutional kernels are fixed, so we only use an Adam optimizer with learning rate of for the linear classifier.
- **Causal CNN [32]**: It combines an encoder based on causal dilated convolutions with a triplet loss employing time-based negative sampling. We use an Adam optimizer with a learning rate of 0.001. A mini-batch size of 32

TABLE III
CATEGORICAL PERFORMANCE COMPARISON ON OFFICE ROOM, FALLDEFI, AND SIGNFI UNDER TRANSFER SETUP.

Datasets	Supervised		AutoEncoder		Rocket		Causal CNN		DualConFi (ours)	
	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
OR→F	64.18	63.82	59.83	58.14	60.85	59.03	68.89	68.84	76.58	76.56
F→OR	69.16	68.90	41.07	40.51	84.80	84.80	84.11	83.94	80.71	80.92
S→OR	72.32	72.17	47.32	47.13	84.80	84.80	79.82	79.71	85.36	85.76
S→F	74.02	74.02	59.15	57.86	60.85	59.03	64.10	64.03	78.97	78.86
Avg. Rank	3.25		5		2.75		2.5		1.5	

is used in the GPU. The learning rate is exponentially decayed by 0.5×, if validation loss does not decrease for 10 epochs.

Both **Rocket** and **Causal CNN** achieve SOTA performance on other multivariate time series datasets such as UEA Time Series Classification Archive [33]. However, as far as we know, their performance on dataset with WiFi data is not reported. Besides, our **Supervised** model, **THAT**, is the de-facto SOTA supervised model on WiFi CSI-based HAR task.

Our model is trained using SGD optimizer with learning rate of 0.001. During unsupervised pre-training, the learning rate is modified by a cosine annealing schedule. The maximum pre-training epoch is set to 1500 with an early stop strategy for all models (except **Rocket**, which does not need pre-training). All experiments are conducted on a Nvidia Quadro P6000 GPU.

B. Datasets

We evaluate our method on three publicly available datasets, i.e., office room, Falldefi, and Signfi. The brief description is provided below, with Table I summarizing dataset characteristics:

- **Office Room** [17] contains 557 records of 7 classes of activities, i.e., lie down, fall, pickup, run, sit down, stand up, and walk.
- **Falldefi** [34] contains 1070 records with the label of fall or not. We select 581 records from it for the experiment.
- **Signfi** [3] contains 8280 records of 276 classes of sign gestures performed by five users. We choose the subset that was collected in the home environment for the experiment.

For all datasets, we adopt preprocessing method mentioned in [18].

C. Linear Evaluation

In order to evaluate the linear separability of different methods' representations, we implement the linear classifier to conduct a series of experiments [13]. After randomly splitting the datasets into train, validation and test sets with-holding 60% for training, 20% for validation and the rest 20% for testing, we record models' average accuracy for 5-independent experiments on three datasets described above. Table II summarizes the average results of the experiments on three datasets. Because of the model structure difference, not all models need unsupervised pre-training. To be more

specific, for **Random init.** and **Rocket** whose encoders are fixed, we directly train the linear classifier on top of the encoder. For other unsupervised methods, we train the linear classifier on top of the frozen encoder, which is pre-trained with our self-supervision tasks, using fully annotated data of the downstream task. For the **Supervised** baseline, we train the full model on the training set from scratch.

According to Table II, **DualConFi** not only get close results to the **Supervised** baseline, but can also achieve new SOTA results after fine-tuning. For example, **DualConFi** achieves the accuracy of 90.14%, 74.87% and 55.25% compared to the fully-supervising accuracy of 94.11%, 73.83% and 84.28% on three datasets, respectively. In terms of F1-score, **DualConFi** also achieve the similar results. Notably, the performances of **DualConFi-FT** get surprising promotions with fine-tuning the last two layers of encoder, 4%-32% accuracy and F1-score gains on all datasets, which outperform the **Supervised** baseline. Compared with other unsupervised methods, **DualConFi** achieves higher performances. Table II shows that on three datasets, without finetuning the encoder, our model has around 6%-10% accuracy promotions compared to the second-best result. It is worth noting that, all unsupervised models except **DualConFi-FT** perform poorly on Signfi dataset. We suppose the enormous class number lead to the performance degenerate of the linear classifier with only a single layer neural network when the encoder is frozen. In a word, the results show that our model can learn high-quality representation, which reveals the linear separability in data, thus achieving the best performances on three datasets compared to all baseline methods.

D. Semi-supervised Learning

A good embedding space should push towards together similar objects and repelled against different objects, whose performance on end tasks will strongly benefit from a small number of labels [35]. Hence, we next investigate the performance of our approach in a semi-supervised setting. We follow [36] and sample 2,5,10,20,50 labeled instances for each class on Office Room, while sample extra 100 and 150 labeled instances for Falldefi because of its bigger size for each category. Signfi is not included due to insufficient labeled instances per class. We randomly split the dataset into train set D_{trn} and test set D_{tst} according to the ratio of 80:20. Then, we form an unlabeled training set D_{utrn} through discarding labels of D_{trn} and sample some labeled instances from D_{trn}

Fig. 4. Office Room accuracy of models trained with few labels. Performances change when we increase the number of labeled instances for each class from 2 to all labels. The blue curve represent **Supervised** trained on few-labeled training set, the grey curve represent **DualConFi** and the orange one represents **DualConFi-FT**.

Fig. 5. Faldefi accuracy of models trained with few labels. Performances change when we increase the number of labeled instances for each class from 2 to all labels.

to form the few-labeled training set D_{fltrn} . On the one hand, as for **DualConFi**, we use D_{fltrn} to fit the linear classifier on top of the frozen encoder, which pre-trained on D_{utrn} in a self-supervised way. **DualConFi** is trained in the same way, except its last two layers of encoder is not frozen. On the other hand, **Supervised** baseline is learned from scratch using D_{fltrn} .

Fig. 4 and Fig. 5 compare our results with **Supervised** baseline. We can observe that on both datasets our **DualConFi** is commensurately good compared to **Supervised** baseline, especially in the low-data regime. Moreover, the version with fine-tuning of the last two layers **DualConFi** tremendously improves the performance and outperforms **Supervised** baseline on each sampling setup.

E. Transfer Learning

Good representations will generalize to a number of downstream tasks, which is the transferability of the representations [36]. To examine the transferability property of the features learned with our framework, we evaluate their performance across three CSI datasets. Selecting two datasets, we use one dataset as the source set and the other as target set. Because Signfi contains around 5 times instances than other two datasets, we evaluate all methods on dataset settings of Office Room $\rightarrow$ Faldefi (OR $\rightarrow$ F), Faldefi $\rightarrow$ Office Room

Fig. 6. Impact of MI coefficient. We evaluate MI coefficient β by Office Room accuracy and F1-score.

TABLE IV

ABLATION STUDY RESULTS COMPARED WITH THE FULL MODEL.

Models	Accuracy		
	Office Room	Faldefi	Signfi
DualConFi	90.14	74.87	55.25
-Temporal Encoder	64.64	57.95	11.52
-Channel Encoder	74.82	72.48	43.04

(F $\rightarrow$ OR), Signfi $\rightarrow$ Office Room (S $\rightarrow$ OR) and Signfi $\rightarrow$ Faldefi (S $\rightarrow$ F). All self-supervised models are pre-trained on the source set, freeze the feature extractor, then initialize and train a new classifier on the labeled target set. Of course, we do not utilize activity labels in the source set for self-supervised representation learning. For the **Supervised** baseline, we train the model on the fully annotated source set from scratch, freeze the feature extractor, and get the classifier in the same way. Table III demonstrate the results of transfer learning experiments.

It can be seen that our model performed best on 3 out of 4 datasets, achieving an average rank of 1.5, followed by **Causal CNN**. Since the action category of these three datasets is not the same, this leads to disparity in data distribution and causes the performance degeneration of the **Supervised** baseline. The convolutional kernels generated by **Rocket** are only related to time length and channel number of input data. Since datasets we used (Office Room, Faldefi, and Signfi) share the same time length and channel number, we can generate **Rocket** kernels once and use them for all datasets. Hence the transfer learning result for **Rocket** is the same as its linear evaluation. These observations reveal that our framework can learn the features with high transferability properties and can be easily transferred to different downstream tasks.

F. Ablation Study

1) *Effectiveness of MI*: In order to study the effectiveness of MI maximization, in Fig. 6 we show the impact of MI coefficient β to **DualConFi**'s accuracy on Office Room, where negative β represents maximize $I(h_t; h_c)$ and $\beta = 0$ is equivalent to eliminating the mutual estimator module from **DualConFi**. Note that the setting in this Section is not best

Temporal Transform							1st Transform						Temporal Transform					
Channel Transform	noise	negate	time flip	time perm	chn. Sobel	Average	2nd Transform	noise	negate	time flip	time perm	chn. Sobel	Channel Transform	noise, time flip, time flip, noise				
	noise	77.23%	65.54%	78.29%	77.92%	77.01%		75.20%	noise	79.37%	65.72%	77.75%		77.90%	75.93%	noise, chn. Sobel	87.52%	87.50%
	negate	78.11%	64.82%	78.10%	76.67%	75.59%		74.66%	negate	65.54%	65.19%	65.37%		63.93%	76.48%	chn. Sobel		
	time flip	79.19%	65.72%	78.46%	76.48%	75.05%		74.98%	time flip	77.93%	65.37%	78.47%		76.12%	75.41%			
	time perm	77.22%	65.36%	75.23%	78.11%	78.46%		74.88%	time perm	79.72%	63.93%	75.05%		75.41%	74.87%			
	chn. sobel	82.95%	65.20%	80.60%	79.72%	74.24%		76.54%	chn. sobel	78.45%	77.39%	75.23%		74.32%	77.03%	chn. Sobel, noise	90.14%	88.75%
	Average	78.94%	65.33%	78.14%	77.78%	76.07%												
(a)							(b)						(c)					

Fig. 7. (a) Compare the accuracy of different tasks. The red values in the average row or column indicate the two transform functions with the best accuracy. (b) The accuracy of applying the same composited transform function to both streams. For all entries, diagonal entries correspond to two same transform functions, and off-diagonals correspond to the composition of two transform functions (applied sequentially). (c) Compare different composite augmentation combinations by accuracy.

Algorithm 1: Implement simple data augmentation in a two-stream manner.

Input: CSI data X .

Parameter: transform function family $\mathcal{T}_t$ and $\mathcal{T}_c$, network f_t, f_c and g .

Output: $NT\text{-}Xent$ loss.

- 1: initialize transform function $t_t, t'_t \sim \mathcal{T}_t, t_c, t'_c \sim \mathcal{T}_c$
- 2: $v_t = t_t(X), v_c = t_c(X)$
- 3: $v_{m_t} = t'_t(X), v_{m_c} = t'_c(X)$
- 4: $r_l = g(\text{concat}(f_t(v_t), f_c(v_{m_c}^T)))$,
- 5: $r_m = g(\text{concat}(f_t(v_{m_t}), f_c(v_m^T)))$
- 6: **return** $NT\text{-}Xent(r_l, r_m)$

Algorithm 2: Implement data augmentation in a composite manner.

Input: CSI data X .

Parameter: transform function family $\mathcal{T}_t$ and $\mathcal{T}_c$, network f_t, f_c and g .

Output: $NT\text{-}Xent$ loss.

- 1: initialize transform function $t_t, t'_t \sim \mathcal{T}_t, t_c, t'_c \sim \mathcal{T}_c$
- 2: $v_l = t_t(t_c(X))$
- 3: $v_m = t'_t(t'_c(X))$
- 4: $r_l = g(\text{concat}(f_t(v_l), f_c(v_l^T)))$,
- 5: $r_m = g(\text{concat}(f_t(v_m), f_c(v_m^T)))$
- 6: **return** $NT\text{-}Xent(r_l, r_m)$

since we did not tune them elaborately according to the validation set. Fig. 6 demonstrates that β cannot be too large or small. Otherwise, the performances drop heavily, which confirms the effectiveness of our mutual information maximization module.

2) Transform functions: In order to shed light upon the effectiveness of transform functions, we design compared the effects of our transform function (channel Sobel) with others described in section III-A. Also, we compare data augmentation in a two-stream manner with a composite manner. Algorithm 1 and Algorithm 2 provide the detail of their implementation differences. In order to simplify the analysis, we disable the MI module in the system, but we make the comparisons are fair by conducting all experiments under

the same setting for 5 trials on Office Room dataset. Fig. 7 (a) shows the results of different transform functions in our implementation way, the red cell indicates high accuracy. Fig. 7 (b) presents the accuracy of applying data augmentation in a composite way, where green denotes the accuracy is lower than ours. Fig. 7 (c) demonstrates the accuracy of four combinations (i.e., noise followed by time flip vs. time flip followed by noise for temporal stream; noise followed by channel Sobel vs. channel Sobel followed by noise for temporal stream). First of all, compare all transform functions' average performance on both streams, we can find out the top two-ranked transform functions on channel stream are channel Sobel and noise, while on temporal stream are noise and time flip. This approves the effectiveness of our channel Sobel on extracting intrinsic channel patterns. Besides, Fig. 7 (c) shows the best sequence setting of the top two-ranked augmentations for both streams. Secondly, compared with our framework, fig (b) shows that the accuracy of the composite method suffers decline to a different extent in most cases. Moreover, most diagonal entries in (b) are red. We suppose that when both streams use the same type of transform function, our framework loses the advantage of extracting high discriminative features. These observations prove the effectiveness of our framework.

3) Framework modules: We evaluate the contribution of different modules (i.e., temporal stream and channel stream). In each experiment, we ablated a specific component from the full model. From Table 4, we can find out that compared with single-stream contrastive learning, the dual-stream structure gains a higher accuracy score on all datasets. This indicates that learning from both streams enhances the semantic features' quality. Table 4. shows that among the two streams, the ablation on the temporal module incurs a bigger decline (from -16.92% accuracy to -43.73% accuracy) on three datasets. This is in line with our intuition that features of the temporal stream are more important for HAR tasks. From the ablation test, we can conclude that both streams contribute significantly to the model performance.

V. CONCLUSION

This paper proposes a novel dual-stream contrastive learning framework for HAR tasks, which uses a dual-stream layout to learn highly-discriminative features and maximize the mutual

information to enhance the quality of final embedding. In addition to designing a novel and effective transform function, we also evaluate the performance of different transform function combinations. The experimental results show that our framework outperforms the SOTA baselines.

However, datasets used in this paper only consider CSI amplitude information since some datasets do not contain phase data. Designing an end-to-end self-supervised framework to process noisy CSI phase information is still a problem. Moreover, although we evaluate our framework on CSI data, our framework can be transferred to other sensor signals with spatial-temporal characteristics such as electroencephalography. These limitations will be considered in future work.

REFERENCES

- [1] C. Jobanputra, J. Bavishi, and N. Doshi, "Human activity recognition: A survey," *Procedia Computer Science*, vol. 155, pp. 698–703, 2019.
- [2] D. Zhang, H. Wang, Y. Wang, and J. Ma, "Anti-fall: A non-intrusive and real-time fall detector leveraging csi from commodity wifi devices," in *International Conference on Smart Homes and Health Telematics*. Springer, 2015, pp. 181–193.
- [3] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, "Signfi: Sign language recognition using wifi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, pp. 1–21, 2018.
- [4] Y. Zeng, D. Wu, R. Gao, T. Gu, and D. Zhang, "Fullbreathe: Full human respiration detection exploiting complementarity of csi phase and amplitude of wifi signals," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, pp. 1–19, 2018.
- [5] J. Huang, S. Lin, N. Wang, G. Dai, Y. Xie, and J. Zhou, "Tse-cnn: A two-stage end-to-end cnn for human activity recognition," *IEEE journal of biomedical and health informatics*, vol. 24, no. 1, pp. 292–299, 2019.
- [6] W. Huang, L. Zhang, Q. Teng, C. Song, and J. He, "The convolutional neural networks training with channel-selectivity for human activity recognition based on sensors," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 10, pp. 3834–3843, 2021.
- [7] Q. T. Huynh, U. D. Nguyen, L. B. Irazabal, N. Ghassemian, and B. Q. Tran, "Optimization of an accelerometer and gyroscope-based fall detection algorithm," *Journal of Sensors*, vol. 2015, 2015.
- [8] Y. Ren, C. Wang, J. Yang, and Y. Chen, "Fine-grained sleep monitoring: Hearing your breathing with smartphones," in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 1194–1202.
- [9] S. Tao, M. Kudo, and H. Nonaka, "Privacy-preserved behavior analysis and fall detection by an infrared ceiling sensor network," *Sensors*, vol. 12, no. 12, pp. 16920–16936, 2012.
- [10] J. K. Aggarwal and M. S. Ryo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, pp. 1–43, 2011.
- [11] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of wifi signal based human activity recognition," in *Proceedings of the 21st annual international conference on mobile computing and networking*, 2015, pp. 65–76.
- [12] Z. Chen, L. Zhang, C. Jiang, Z. Cao, and W. Cui, "Wifi csi based passive human activity recognition using attention based blstm," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2714–2724, 2018.
- [13] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.
- [14] W. Hao, Z. Daqing, N. Kai, L. Qin, L. Yuanhuai, W. Dan, G. Ruiyang, and X. Bing, "Mfdl: A multicarrier fresnel penetration model based device-free localization system leveraging commodity wi-fi cards," *arXiv preprint arXiv*, vol. 1707, 2017.
- [15] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *International Conference on Machine Learning*. PMLR, 2020, pp. 9929–9939.
- [16] H. Wang, D. Zhang, Y. Wang, J. Ma, Y. Wang, and S. Li, "Rt-fall: A real-time and contactless fall detection system with commodity wifi devices," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 511–526, 2016.
- [17] S. Yousefi, H. Narui, S. Dayal, S. Ermon, and S. Valaee, "A survey on behavior recognition using wifi channel state information," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 98–104, 2017.
- [18] B. Li, W. Cui, W. Wang, L. Zhang, Z. Chen, and M. Wu, "Two-stream convolution augmented transformer for human activity recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 286–293.
- [19] Q. Gao, J. Wang, X. Ma, X. Feng, and H. Wang, "Csi-based device-free wireless localization and activity recognition using radio image features," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10346–10356, 2017.
- [20] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.
- [21] M. Ye, X. Zhang, P. C. Yuen, and S.-F. Chang, "Unsupervised embedding learning via invariant and spreading instance feature," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6210–6219.
- [22] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," *arXiv preprint arXiv:2103.03230*, 2021.
- [23] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3733–3742.
- [24] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *arXiv preprint arXiv:2006.09882*, 2020.
- [25] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European Conference on Computer Vision*. Springer, 2016, pp. 69–84.
- [26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [27] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.
- [28] R. Linsker, "Self-organization in a perceptual network," *Computer*, vol. 21, no. 3, pp. 105–117, 1988.
- [29] M. Tschannen, J. Djolonga, P. K. Rubenstein, S. Gelly, and M. Lucic, "On mutual information maximization for representation learning," *arXiv preprint arXiv:1907.13625*, 2019.
- [30] D. Liu and G. Liu, "A transformer-based variational autoencoder for sentence generation," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–7.
- [31] A. Dempster, F. Petitjean, and G. I. Webb, "Rocket: exceptionally fast and accurate time series classification using random convolutional kernels," *Data Mining and Knowledge Discovery*, vol. 34, no. 5, pp. 1454–1495, 2020.
- [32] J.-Y. Franceschi, A. Dieuleveut, and M. Jaggi, "Unsupervised scalable representation learning for multivariate time series," *Advances in neural information processing systems*, vol. 32, 2019.
- [33] A. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh, "The uea multivariate time series classification archive, 2018," *arXiv preprint arXiv:1811.00075*, 2018.
- [34] S. Palipana, D. Rojas, P. Agrawal, and D. Pesch, "Falldet: Ubiquitous fall detection using commodity wi-fi devices," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, pp. 1–25, 2018.
- [35] Z. Fu, Y. Li, Z. Mao, Q. Wang, and Y. Zhang, "Deep metric learning with self-supervised ranking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 1370–1378.
- [36] A. Saeed, V. Ungureanu, and B. Gfeller, "Sense and learn: Self-supervision for omnipresent sensors," *arXiv preprint arXiv:2009.13233*, 2020.