Enhancing Wi-Fi Sensing Capabilities with Self-Supervised Learning

Amirhossein Mohammadi [1]

Abstract

This project utilizes a self-supervised learning (SSL) framework to derive representations of Channel State Information (CSI), focusing on the DirectCLR approach within contrastive SSL methodologies. It encodes CSI data into robust embeddings without relying on labeled data, addressing practical scenarios where label acquisition is costly. Our approach encompasses various data augmentation techniques, a novel tailored backbone architecture for representation extraction, and the implementation of DirectCLR for learning representations. Empirical results demonstrate the effectiveness of our method in learning representations from an unlabeled dataset, showing its competitiveness with methods like SimCLR.

1. Introduction

Wi-Fi devices have become ubiquitous, commonly found in virtually every room. Over the past few decades, their widespread adoption has encouraged researchers to explore the use of information these devices capture for sensing purposes. Specifically, the technology underlying Wi-Fi communication, known as orthogonal frequency division multiplexing (OFDM), CSI for each transmit and receive antenna pair across all carrier frequencies. Researchers utilize CSI to develop various sensing applications that leverage existing wireless communication infrastructure. This approach is particularly attractive because it preserves privacy and involves no additional financial costs.

The potential of CSI data has led to its application in several innovative areas, including motion detection (Gong et al., 2015) (Liu et al., 2017), activity recognition (Fang et al., 2016) (Duan et al., 2018), gesture recognition(Li et al., 2016) (Qian et al., 2017), and human presence detection (Gong et al., 2016) (Palipana et al., 2016). These applications demonstrate how Wi-Fi can extend beyond mere communication to play a crucial role in environmental sensing.

However, a significant challenge in this field is the scarcity of labeled datasets. Deep learning for sensing applications typically requires a substantial amount of labeled data for supervised training. Given the high costs associated with obtaining labels, researchers are increasingly turning to self-supervised learning methods. These methods enable the extraction of meaningful representations from data without labeled examples. Models developed using SSL can later be utilized to extract useful features for various downstream tasks, offering a cost-effective alternative to traditional supervised learning approaches.

In this project, we aim to learn representations from unlabeled CSI data using an SSL approach. We employ various augmentation techniques to feed our self-supervised model and propose a new backbone architecture specifically tailored for CSI data used in this project. We have effectively trained our model and tested its accuracy using a linear head on our learned representations, demonstrating promising results in practical applications.

2. Preliminaries

- **CSI in Wireless Networks:** In all wireless systems, it is crucial to estimate the behavior of channels for the precise detection of transmitted symbols. The symbol $x(t)$ reaches the receiver through the model $y(t) = H(t) \cdot x(t) + n$, and without knowledge of the channel $H(t)$, the recovery of transmitted symbols at the receiver becomes quite challenging. Hence, wireless devices are in a constant process of estimating the channel state information. For WiFi systems that employ Orthogonal Division Multiplexing Modulation, the channel state information is represented by a three-dimensional tensor $H \in \mathbb{R}^3$, which depicts the channel's characteristics.

$$H_{t,y,z} = \alpha(t,y) \sum_n a_n(t) e^{-j2\pi d_{y,n} f_z + \phi(y,z)} \quad (1)$$

We are interested in understanding the physical meaning behind each entry of the CSI, as described by equa-

[1]Department of Electrical Engineering, York University, Toronto, Canada. Correspondence to: Amirhossein Mohammadi <amirmhd@yorku.ca>.

Project Report for EECS6327 (Fall 2020), Department of Electrical Engineering and Computer Science, York University.

tion (Eq. 1). At each time stamp t, there exists a 2D matrix; the first axis corresponds to receiver antennas, and the second axis corresponds to subcarrier frequencies. In the area of wireless communication, to optimize the frequency spectrum usage, different frequency components z are utilized, and multiple receiver antennas y are employed to reduce the error rate. Given the estimated channel, we can recover transmitted symbols by $\hat{x} = H^{-1}y$. Nevertheless we leverage information within CSI to solve a classiciation problem based on the representation that we found.

- **Contrastive SSL for Representation Learning:** There is a multitude of unsupervised learning methods that can help to find representations of data. Contrastive SSL is a recent approach in SSL that has gained popularity due to its powerful performance in computer vision tasks like (Chen et al., 2020) , (Zbontar et al., 2021), and (Jing et al., 2022). It tries to find representations of data by minimizing the distance between the embedding vectors of augmented views of training instances. The main issue with SSL is that the model tends to output the same embedding for each training instance; this problem is also known in the literature as dimensional collapse. Contrastive methods attempt to solve this collapse not only by reducing the distance between embedding vectors but also by increasing the distance between negative pairs in the embedding space. Their approach for solving collapse problem can be easily observed in a typical contrastive loss like InfoNCE loss function. Equation 2 shows infoNCE loss function formula for one example i.

$$L = -\log \frac{\exp(-\|z_i - z_i'\|^2/2)}{\sum_{j \neq i} \exp(-\|z_i - z_j\|^2/2) + \exp(-\|z_i - z_i'\|^2/2)} \tag{2}$$

3. Dataset

Understanding the data is the most crucial step in designing machine learning models. In this section, we will explore the nature of the data to pave the way for creating our model in subsequent sections. Briefly, the dataset used in this project is the SignFi dataset (Ma et al., 2018) each sample with dimensions $x_i \in \mathbb{R}^{200\times30\times3}$. This dataset contains 276 classes, which represent different sign language gestures. The features are CSI captured using WiFi modules. We have briefly introduced the mathematical model in communication systems for finding CSI in Section 2; now we will delve deeper into each feature dimension one by one.

Channel Dimension: Analogous to the RGB channels in image data, the third dimension represents the number of receiver antennas (three). These antennas capture the same signal from transmitter, yet their distinct locations cause variations in the data phase, similar to how RGB channels capture different color data but are correlated.

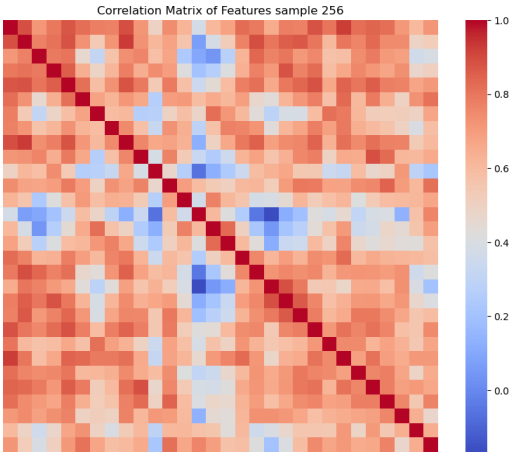

Figure 1. Correlation coefficients along the sub-carrier axis. The sample is chosen randomly, and for a given time and channel, we have computed the correlation coefficients of different sub-carriers.

Sub-carrier Dimension: The dataset encompasses 30 sub-carriers, indicating transmission over 30 distinct frequency components. As a wave propagates from a transmitter to a receiver, different frequency components of WiFi will behave almost the same in the environment, suggesting that along the sub-carrier dimension, we have high correlation. To test our hypothesis about our data, refer to the table in Figure 1, which shows correlation coefficients of sub-carriers for a random sample, with a fixed timestamp, and channel.

Time Stamp Dimension: We can use our domain knowledge to hypothesize what we should expect from the data and then test that hypothesis along this dimension. Certainly, our timestamps represent a random process, but if we take a closer look at Eq. (1), we can see that for fixed time indices y, z, the CSI is a summation of complex-valued random variables. From the central limit theorem, we expect that each element of H will have a complex normal distribution, and thus its absolute value should follow a Rayleigh distribution. Our observations also confirm this; Figure 3 shows the distribution of the real part of the timestamps for different subcarriers.

One important note to consider here is that our time series is a natural signal, after all. Therefore, like all natural signals, it exhibits local dependencies in the time domain and does not feature impulsive jumps. Even if we view it mathematically, we can model our time series using a random process R_t where $R_0 = 0$. We can observe that $R_{t1} - R_{t2} \to 0$ as $t_1 - t_2 \to 0$.

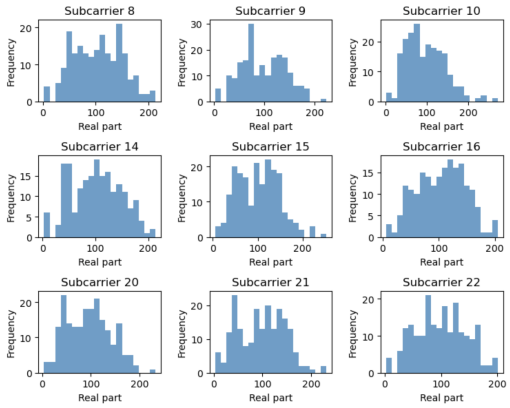

Figure 2. The histogram of the real part of timestamps for different subcarriers with a fixed channel. As we can see, they approximately follow a normal distribution.

4. Methodology

In this section, we will introduce our approach for solving the problem of finding representations for CSI data. We start with the data augmentation methods that are used to different views of each sample, then we continue with the proposed architecture of the backbone for finding representations of the data, and finally, we introduce the SSL approach used in this project.

4.1. Data Augmentation

To ensure robustness and generalizability in contrastive SSL methods, careful implementation of data augmentation is essential. Augmentation techniques aim to create diverse and informative views of each data sample, which are key to developing useful representations. However, the process requires precision: overly strong augmentations can lead to model collapse and failure, whereas overly weak augmentations may not provide enough information for the model to learn effectively. In the following sections, we will detail the specific augmentation methods and hyper parameters employed in this project

1. **Time Permutation**: This method divides the time series into three segments and randomly permutes these segments within the time domain. This augmentation challenges the model to learn representations that are invariant to changes in the sequence order of events, thus capturing the underlying patterns without relying on the absolute positioning of data points in time.

2. **Time Jittering**: We introduce random noise to the time series data, where the noise is normally distributed with a variance of 1. This method is intended to make the model robust to small fluctuations and variations in the data, simulating potential real-world disturbances that could affect the measurements.

3. **Time Warping**: By applying a time warp with an intensity of 0.5, we non-linearly stretch or compress the time axis of the data. This augmentation simulates variations in the speed at which time-varying events occur, aiding the model in learning to recognize patterns irrespective of their temporal scaling.

4. **Amplitude Scaling**: The amplitude of the data is scaled by a factor randomly chosen from a uniform distribution within the range of 0.9 to 1.1 (intensity 1 ± 0.1). This scaling mimics changes in signal strength, which can occur due to varying environmental conditions or sensor sensitivities.

4.2. Backbone

In what follows, we will explain the architecture used for extracting representations from the CSI data. We will apply the information from the previous section to extract patterns based on our understanding of the data.

Along the time axis, we have local dependencies as shown in Section 3, so we extract patterns and useful motifs using convolutional layers. For the first two layers, we suggested using 1D convolutional layers along the time domain axis, aiming to extract higher-level features from the data while treating the antenna axis as our channel axis. After these two layers, a 2D convolutional layer has been used where along the time domain we have filters of the same size, but along the subcarrier axis, we have a filter size equal to the number of subcarriers. The rationale behind this is that due to the fact that all subcarriers are correlated (see Section 3), we need a fully connected layer to capture the interplay between all indices. This 2D convolution is followed by three more 1D convolutional layers.

The sequence of convolutional layers starts with lighter filters and progressively moves to denser filters (up to 1024 channels), increasing the model's capacity to capture more complex temporal features. Therefore, the dimension of the representation is $z \in \mathbb{R}^{1024}$. Batch normalization follows each convolution to stabilize learning and improve convergence speed, while dropout layers are interspersed to prevent overfitting.

4.3. Self Supervised Loss

In what follows, we will explain how we utilize SSL to shape our representation space. The method we are using is based on directCLR (Jing et al., 2022), which directly sends the representations to the embedding space without any projectors.

Projectors in SSL are used as intermediate layers between the backbone output (representations) and the loss function.

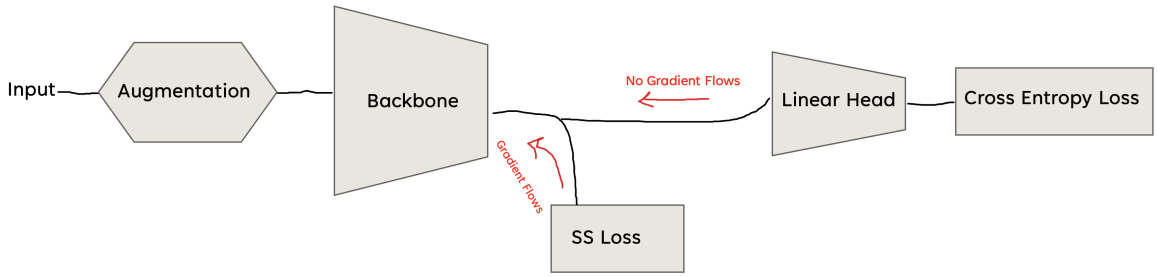

Figure 3. The flow chart of the overall model. Pay attention that there are no gradients coming back from cross-entropy loss functions, yet the backbone will be updated using gradients coming from the InfoNCE loss function.

The primary role of these projectors is to transform the raw representations into a space where similar instances are closer together, while dissimilar ones are farther apart, effectively enhancing the discriminative properties of the embeddings. This transformation is crucial in helping the learning algorithm focus on the most relevant features for discrimination, thus potentially improving the performance of downstream tasks.

Projectors were previously considered an indispensable part of SSL. However, (Jing et al., 2022) demonstrated that due to implicit regularization, we can eliminate the projectors and directly optimize the representations. Specifically, they showed that in cases of overparameterized linear projectors, the projectors tend to become low-rank and find a flat local optimum. Therefore, they suggested that instead of using a weight matrix for the projector, which over time tends to become low-rank, we can directly send a subset of the representations to the InfoNCE loss function. Therefore, we only send `r = z[:, 0:d0]` to the infoNCE loss function as our embedding vectors. Note `d0` is a hyper parameter and need to be optimized in order to find the best one.

4.4. Putting Everything Together

We now conclude our methodology and bring all the pieces together. At each mini-batch, we generate two views of each training sample using the augmentation techniques introduced in Section 4.1, then we obtain representations using the proposed backbone architecture, and finally, we feed a subset of these representations to the InfoNCE loss function to train the backbone. To evaluate the proposed algorithm, we have also designed an 'online head,' as is customary in the SSL literature. The online head consists of a linear layer followed by a cross-entropy loss function that is trained simultaneously with the backbone. However, the gradients do not flow backward from the online head, as it is detached from the computational graph. The sole purpose of the online head is to assess the quality of the representations by determining the accuracy on the validation dataset.

5. Results

Here we share the performance of the algorithm and compare it with other methods. We have trained the algorithm for 300 epochs and measured its accuracy with a linear online head followed by a cross-entropy loss function. Moreover, we have compared the algorithm with SimCLR (Chen et al., 2020), using a linear projector. In figure 4 the accuracy of these two algorithm has been mentioned.

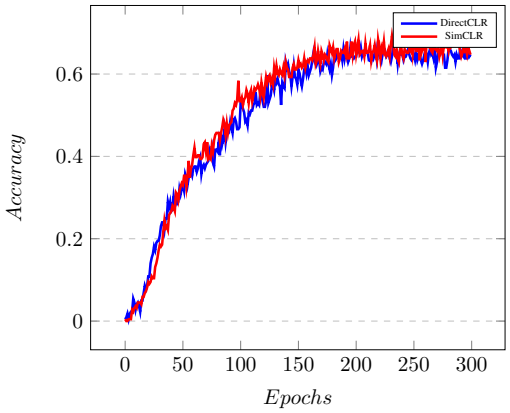

Figure 4. Comparison of DirectCLR (Jing et al., 2022) appraoch with the SimCLR (Chen et al., 2020) algorithm for SignFi CSI data.

Algorithm	Accuracy (%)	Prjector
SimCLR	66.8	2-layered linear projector
DirectCLR	65.2	no project

Table 1. Comparison of Algorithm Accuracies

6. Discussion and Conclusion

We presented a self-supervised learning method for finding representations of CSI data. The representations we discovered do not depend on labels, making them highly useful in practical scenarios where we have free data in WiFi routers every where. In such situations, acquiring labels can be

costly given the vast amounts of data involved. The representations performed fairly well relative to the number of classes and samples.

As expected, directCLR performed comparably to SimCLR, although SimCLR achieved slightly better results. I believe that if we had used a ResNet architecture for backbone, directCLR might outperform SimCLR because the gradient flow in backward passes can optimize parts of the representation that are not passed to the InfoNCE loss, as shown in the original paper.

Moreover, we experimented different backbone structures, training them in a supervised fashion with labels. It appears that changing the backbone structure to something more conventional, such as starting with 2D convolutions and increasing the channel dimension as we progress (similar to VGG16 (Simonyan & Zisserman, 2015)), does not significantly alter the outcomes. In my experiments, an architecture similar to VGG16 achieved an accuracy of around 93%, while my proposed architecture reached 96%.

Still, although the backbone architecture is not critically important, the augmentation technique plays a crucial role in SSL. Both the techniques and their parameters are significant. For instance, using channel shuffling destroys the algorithm's learning curve, or using values smaller or larger than those specified in the project results in the model not training properly.

In total, the project results show that we are able to find representations from CSI data which are much better than a linear transformation like PCA. However, it seems that we need to significantly improve the algorithm in order to have trustworthy and practical applications. I believe that using more augmentation techniques specifically tailored for CSI data can significantly enhance the model's performance. For future work, we suggest increasing the variety of augmentation techniques to improve the representations learned by the SSL algorithm.

References

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations, 2020.

Duan, S., Yu, T., and He, J. Widriver: Driver activity recognition system based on wifi csi. *International Journal of Wireless Information Networks*, 25:1–11, 06 2018. doi: 10.1007/s10776-018-0389-0.

Fang, B., Lane, N. D., Zhang, M., Boran, A., and Kawsar, F. Bodyscan: Enabling radio-based sensing on wearable devices for contactless activity and vital sign monitoring. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '16, pp. 97–110, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342698. doi: 10.1145/2906388.2906411. URL https://doi.org/10.1145/2906388.2906411.

Gong, L., Yang, W., Man, D., Dong, G., Yu, M., and Lv, J. Wifi-based real-time calibration-free passive human motion detection. *Sensors*, 15(12):32213–32229, 2015. ISSN 1424-8220. doi: 10.3390/s151229896. URL https://www.mdpi.com/1424-8220/15/12/29896.

Gong, L., Yang, W., Zhou, Z., Man, D., Cai, H., Zhou, X., and Yang, Z. An adaptive wireless passive human detection via fine-grained physical layer information. *Ad Hoc Networks*, 38:38–50, 2016. ISSN 1570-8705. doi: https://doi.org/10.1016/j.adhoc.2015.09.005. URL https://www.sciencedirect.com/science/article/pii/S1570870515002127.

Jing, L., Vincent, P., LeCun, Y., and Tian, Y. Understanding dimensional collapse in contrastive self-supervised learning, 2022.

Li, H., Yang, W., Wang, J., Xu, Y., and Huang, L. Wifinger: talk to your smart devices with finger-grained gesture. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, pp. 250–261, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450344616. doi: 10.1145/2971648.2971738. URL https://doi.org/10.1145/2971648.2971738.

Liu, J., Wang, L., Guo, L., Fang, J., Lu, B., and Zhou, W. A research on csi-based human motion detection in complex scenarios. In *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pp. 1–6, 2017. doi: 10.1109/HealthCom.2017.8210800.

Ma, Y., Zhou, G., Wang, S., Zhao, H., and Jung, W. Signfi: Sign language recognition using wifi. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2 (1), mar 2018. doi: 10.1145/3191755. URL https://doi.org/10.1145/3191755.

Palipana, S., Agrawal, P., and Pesch, D. Channel state information based human presence detection using non-linear techniques. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*, BuildSys '16, pp. 177–186, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342643. doi: 10.1145/2993422.2993579. URL https://doi.org/10.1145/2993422.2993579.

Qian, K., Wu, C., Zhou, Z., Zheng, Y., Yang, Z., and Liu, Y. Inferring motion direction using commodity wi-fi for interactive exergames. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pp. 1961–1972, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346559. doi: 10.1145/3025453.3025678. URL https://doi.org/10.1145/3025453.3025678.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition, 2015.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction, 2021.