LB-SciFi: Online Learning-Based Channel Feedback for MU-MIMO in Wireless LANs

***Abstract*—Multi-user MIMO (MU-MIMO) is a key technology for current and next-generation wireless local area networks (WLANs). While it has widely been deployed in WLANs, its potential is not fully exploited in real-world systems. This can be attributed to the large airtime overhead induced by channel acquisition in existing MU-MIMO protocols, which significantly compromises the throughput gain of MU-MIMO. In this paper, we present LB-SciFi, a learning-based channel feedback framework for MU-MIMO in WLANs. LB-SciFi takes advantage of recent advances in deep neural network autoencoder (DNN-AE) to compress channel state information (CSI) in 802.11 protocols, thereby conserving airtime overhead and improving spectral efficiency. The key component of LB-SciFi is an online DNN-AE training scheme, which allows an AP to train DNN-AEs by leveraging the side information of existing 802.11 protocols. With this training scheme, DNN-AEs are capable of significantly lowering the airtime overhead for MU-MIMO while preserving its backward compatibility with incumbent Wi-Fi client devices. We have implemented LB-SciFi on a wireless testbed and evaluated its performance in indoor wireless environments. Experimental results show that LB-SciFi offers an average of 73% airtime overhead reduction and increases network throughput by 69% on average.**

***Index Terms*—Wi-Fi protocols, MU-MIMO, channel state information, neural network, autoencoder, overhead**

I. Introduction

The proliferation of wireless devices, combined with the growth of Internet-based wireless applications such as online streaming and video chatting, has led to continuously increasing demands for wireless services in indoor environments such as smart homes, university campuses, football stadiums, and airports. As one of the largest wireless networks in real world, wireless local area networks (WLANs) carry the most wireless data traffic (even more than cellular networks) and play a pivotal role in our society. To meet the increasing demands for data services in WLANs, multi-user multiple-input multiple-output (MU-MIMO) is a key technology. It allows an access point (AP) to serve multiple users simultaneously and therefore can significantly improve the spectral efficiency. Given its potential, MU-MIMO has been specified in the IEEE 802.11 standards [1], [2] and widely been deployed on commercial Wi-Fi devices, e.g., Wi-Fi routers, laptops, and phones.

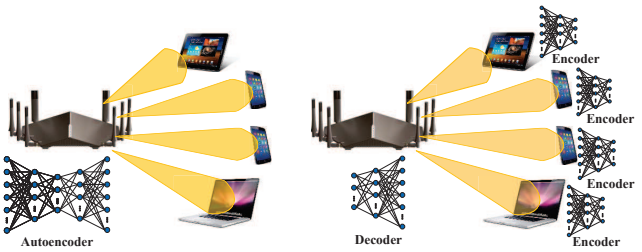
In real-world WLANs, the downlink typically has higher demands for data services compared to the uplink. To support downlink MU-MIMO communications in WLANs, an AP needs to access short-term channel state information (CSI) for the construction of beamforming filters. The filters will then be used to project the precoded signals onto the AP's multiple antenna elements so that each user can decode its data packets. Thus, CSI at the AP is essential to enabling downlink MU-MIMO transmissions. There are two channel acquisition methods for an AP to obtain CSI: i) *implicit channel acquisition*, and ii) *explicit channel acquisition*. The implicit method is based on channel reciprocity. The AP infers the downlink CSI through the estimation of uplink CSI and periodic channel calibrations [3]. This method, however, requires an extra radio frequency (RF) chain on hardware or a sophisticated algorithm for channel calibration, and may not be suited for implementation on low-cost Wi-Fi devices [4]–[6].

The explicit method is based on channel feedback over uplink over-the-air channel. Each user first estimates the downlink CSI and then reports the estimated CSI to the AP. Given its amenability to implementation, this method has been adopted by the IEEE 802.11 standards [1], [2] and been implemented on commercial Wi-Fi systems. However, due to its reliance on over-the-air CSI feedback, it suffers from large airtime overhead. The large overhead of this method can be attributed to the large number of subcarriers in WLANs' OFDM modulation, each of which has a channel matrix to be reported. Existing 802.11 protocols may group subcarriers for CSI feedback to reduce the overhead. Apparently, such a naive scheme will lead to an inferior beamforming performance and drastically compromises the throughput gain of MU-MIMO. While there are many results of MU-MIMO in the literature, the CSI compression for 802.11 MU-MIMO protocols is highly overlooked and its progress remains limited.

In this paper, we study explicit channel acquisition in 802.11 MU-MIMO protocols with the objective of minimizing CSI feedback airtime overhead while preserving CSI feedback accuracy. Toward this objective, we propose a learning-based channel feedback framework (called LB-SciFi[1]) for 802.11 protocols to reduce their airtime overhead by taking advantage of recent advances in deep neural network autoencoder (DNN-AE). Fig. 1 shows the basic idea of LB-SciFi, which is composed of two phases: *online training* and *real-time exploitation*. In the *training* phase, LB-SciFi trains DNN-AEs at the AP by leveraging side information from existing 802.11 MU-MIMO protocols, and thus require no extra effort from user devices. In the *exploitation* phase, LB-SciFi uses the trained DNN-AEs to compress CSI for efficient feedback. Given the redundancy of CSI and the effectiveness of DNN-AEs, LB-SciFi can reduce the airtime overhead significantly without sacrificing CSI feedback accuracy.

The main challenge in the design of LB-SciFi is the online training of DNN-AEs, which should be capable of capturing the kernel space of all possible channels in a given wireless environment through the learning of collected CSI at the AP. To address this challenge, we design an efficient training scheme for the DNN-AEs, which jointly optimize the structure of DNN-AEs, the collection of training data, and the prepro-

[1]LB-SciFi stands for Learning-Based compression for Ψ (Sci) and Φ (Fi), which are the CSI for feedback in the MU-MIMO protocols specified in the IEEE 802.11ac/ax [1], [2].



(a) Online training. (b) Real-time exploitation.

Fig. 1: An overview of DNN-AEs for channel feedback compression in 802.11 MU-MIMO protocols.

cessing of collected data by leveraging the existing feedback data in existing 802.11 MU-MIMO protocols. Specifically, the proposed training scheme meticulously chooses the ψ and ϕ angles from Givens Rotation (GR) as the DNN-AEs input based on a defined power spectral entropy (PSE). Moreover, several important engineering problems have been addressed to make DNN-AEs work in real-world wireless environments.

This paper advances the state-of-the-art in the following respects.

- We propose to employ DNN-AEs for CSI compression in 802.11 MU-MIMO protocols, and have designed an online training scheme for DNN-AEs while imposing no computation burden on user devices.
- Based on the DNN-AEs, we have designed a learning-based channel feedback framework (LB-SciFi) for downlink MU-MIMO. This framework can dramatically reduce the CSI feedback airtime overhead for 802.11 MU-MIMO protocols without sacrificing CSI feedback accuracy.
- We have built a prototype of LB-SciFi and evaluated its performance in real-world indoor environments. Our experimental results show that LB-SciFi reduces the CSI feedback airtime overhead by 73% and improves the throughput of MU-MIMO by 69% on average.

II. Related Work

We focus our literature review on research efforts studying low-overhead channel acquisition methods for MU-MIMO transmissions in WLANs and cellular networks.

Channel Acquisition in WLANs: As the core technology of existing WLANs, MU-MIMO markedly improves users experience with high throughout and low latency. However, airtime overhead from channel acquisition is a real barrier toward fully exploiting the potential of MU-MIMO. Given the severity of this issue, research efforts have been devoted to studying the effect of channel acquisition parameters on network throughput or completely altering the channel acquisition paradigm to enhance network throughput [7]–[18].

Pioneering work [7]–[10] studied the underlying relationship between network throughput and channel acquisition parameters. The outcome was not surprising; full exploitation of MU-MIMO requires a timely CSI through a frequent channel acquisition. The large airtime overhead, however, drastically compromises the throughput gain of MU-MIMO. [11]–[15] aimed at lowering the frequency of channel acquisitions to reduce channel feedback overhead for MU-MIMO protocols. However, the airtime overhead was still too large. [16]–[18] revisited existing channel acquisition paradigm and explored new methods for efficient channel acquisition.

Thus far, there is no efficient method for CSI compression to reduce feedback overhead. Our work fills this gap by leveraging recent advances in artificial neural networks to compress CSI. The resultant CSI feedback will entail much less overhead compared to existing 802.11 protocols.

Channel Acquisition in Cellular Networks: Compared to WLANs, the need for low-overhead channel acquisition methods in cellular networks is appreciated earlier as the emergence of massive-MIMO revealed the drawbacks of traditional techniques. Toward this objective, the underlying correlation of CSI reports has been used for compression by removing the redundant correlated information [19]–[24]. In particular, temporal correlation [19]–[21], spectral correlation [22]–[24], and spatial correlation [19], [25] have been explored to minimize the representation of CSI. Channel reciprocity [26]–[28] and outdated CSI [29] have also been studied to enhance the efficiency of channel acquisition.

Our work is orthogonal to these research efforts in the following two aspects: i) Our work focuses on indoor WLANs, which differ from cellular networks in terms of CSI format, network architecture, data collection, data processing, and system implementation. ii) While the above efforts focused on theoretical exploration, our work focuses on practical design based on real-world 802.11 protocols.

III. Problem Description

In this section, we first offer a primer of existing 802.11 MU-MIMO protocols and underscore their airtime overhead issue. Then, we will state our design objective and challenges.

A. Existing 802.11 MU-MIMO Protocols

Consider a WLAN as shown in Fig. 1(a), where a multi-antenna AP is serving a set of user devices (a.k.a. stations or STAs for brevity). The AP is equipped with N_{ap} antennas, and an STA is equipped with N_{sta} antennas. Due to the physical size and power limits, an STA typically has less antennas than an AP, i.e., $N_{sta} < N_{ap}$. In such a WLAN, MU-MIMO is widely used to exploit the spatial degrees of freedom (DoF) of asymmetric antenna configuration by enabling the AP to serve multiple STAs simultaneously. The application of MU-MIMO not only improves spectral efficiency and user scheduling flexibility but it also reduces packet delay at the MAC layer and enhances fairness in resource allocation.

To enable MU-MIMO transmissions in real-world WLANs, protocols with explicit channel acquisition have been specified in the IEEE 802.11 standards [1], [2]. Fig. 2 shows an existing 802.11 MU-MIMO protocol, which is composed of the following four phases:

- *MU-MIMO Announcement:* The AP selects a subset of STAs for the downlink MU-MIMO transmission based on some pre-defined criteria.[2] After user selection, the AP

[2]Note that user selection is not in the scope of this paper, and there are many prior results on user selection for MU-MIMO.

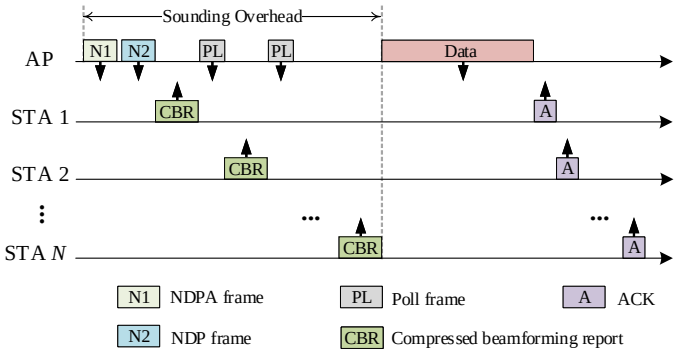


Fig. 2: A MU-MIMO protocol specified in IEEE 802.11ac [1].

broadcasts a Null Data Packet Announcement (NDPA) to inform the STAs of MU-MIMO transmission, followed by an NDP for those selected STAs to estimate downlink CSI.

- *Channel Feedback:* After estimating CSI, the selected STAs feed back their CSI to the AP sequentially following the poll frames from the AP, as shown in Fig. 2. The CSI feedback procedure will be detailed shortly.
- *Data Transmission:* Upon obtaining CSI from all STAs, the AP uses CSI to construct beamforming filters and performs downlink data transmission.
- *Acknowledgment:* After decoding the packets, all STAs sends an ACK/NACK to the AP to indicate the success/failure of their packet detection.

In the channel feedback phase, if an STA sends raw CSI to the AP, it entails a huge amount of airtime overhead and thus negates the throughput gain of MU-MIMO. To reduce the airtime overhead, 802.11 protocols have employed angle-based CSI feedback instead of raw CSI feedback in the spatial domain and specified subcarrier grouping in the spectral domain. We detail them below.

Angle Feedback in Spatial Domain: Referring to the protocol in Fig. 2, once an STA has received the NDP from the AP, it estimates the downlink CSI, i.e., $\mathbf{H}(k) \in \mathbb{C}^{N_{sta} \times N_{ap}}$, $1 \leq k \leq N_{sc}$, where N_{sc} is the number of valid subcarriers. Instead of reporting the complex entries of $\mathbf{H}(k)$, the STA reports two sets of angles ($\mathbf{\Psi}$ and $\mathbf{\Phi}$) to the AP to reduce the feedback overhead. A high-level description of computing $\mathbf{\Psi}$ and $\mathbf{\Phi}$ is given in Alg. 1. This conversion is also known as Givens Rotations. Details of computing the angles can be found in [30]. With these two sets of angles, the AP can reconstruct the essential spatial information of $\mathbf{H}(k)$, which suffices for beamforming operations at the AP.

In this method, the number of generated angles in $\mathbf{\Phi}$ is $N_\phi = (N_{ap}N_{sta} - N_{sta}^2/2 - N_{sta}/2)N_{sc}$, so is the number of angles in $\mathbf{\Psi}$. These angles need to be reported to the AP via the uplink over-the-air channels. In 802.11 standards [1], two types of quantization are specified for CSI feedback:

- *Type 0:* 5 bits for angles in $\mathbf{\Psi}$ and 7 bits for angles in $\mathbf{\Phi}$,
- *Type 1:* 7 bits for angles in $\mathbf{\Psi}$ and 9 bits for angles in $\mathbf{\Phi}$.

Subcarrier Grouping in Spectral Domain: In a typical environment of WLANs, adjacent subcarriers experience highly correlated channel responses from the medium. Therefore, instead of reporting CSI for every individual subcarrier, an STA may group multiple neighboring subcarriers together for CSI feedback. Per IEEE 802.11ac [1], the number of subcarriers in a group, denoted by N_g, can be 1, 2, or 4, depending on network configuration. In IEEE 802.11ax [2], N_g can also be 8 or 16 due to its small subcarrier spacing.

Algorithm 1 A high-level description of computing $\mathbf{\Psi}$ and $\mathbf{\Phi}$ at an STA specified in the IEEE 802.11ac/ax [1], [2].

Inputs: Estimated channel at an STA, i.e., $\mathbf{H}(k) \in \mathbb{C}^{N_{sta} \times N_{ap}}$, $1 \leq k \leq N_{sc}$
Outputs: Computed angles, i.e., $\mathbf{\Psi}$ and $\mathbf{\Phi}$
Set $\mathbf{\Psi} = \{\}$ and $\mathbf{\Phi} = \{\}$
for ($k = 1$; $k \leq N_{sc}$; k++) **do**
  $[\mathbf{U},\ \mathbf{\Sigma},\ \mathbf{V}] = svd\,(\mathbf{H}(k))$
  $\mathbf{V}' = \mathbf{V}\,(:,\, 1{:}N_{sta})$
  for ($l = 1$; $l \leq N_{sta}$; l++) **do**
    $\psi_k := phase\_extraction(\mathbf{V}'\,(:, l))$
    $\phi_k := givens\_rotations(\mathbf{V}'\,(:, l))$
    $\mathbf{\Psi} := \{\mathbf{\Psi}\ \ \psi_k\}$ and $\mathbf{\Phi} := \{\mathbf{\Phi}\ \ \phi_k\}$
  end for
end for
Quantizing every angle in $\mathbf{\Psi}$ using p bits, $p \in \{5, 7\}$
Quantizing every angle in $\mathbf{\Phi}$ using q bits, $q = p + 2$.

Large Airtime Overhead: Even with the spatial- and spectral-domain compression, 802.11 MU-MIMO protocols still come with a large amount of airtime overhead, which significantly compromises the throughput gain of MU-MIMO. For example, for an STA with 4 antennas and an AP with 8 antennas, the CSI feedback could be as large as 19.7 kbit for 20 MHz bandwidth and 170.4 kbit for 160 MHz bandwidth. The problem of CSI feedback airtime overhead becomes increasingly acute as the evolution of WLANs is accommodating more subcarriers in a certain frequency band. For example, IEEE 802.11ax employs 256 subcarriers over 20 MHz for packet transmissions, which is four times greater than that of IEEE 802.11ac. The increase of subcarriers makes it urgent to resolve the CSI feedback overhead problem.

B. Our Objective and Challenges

We aim to reduce the CSI feedback airtime overhead by taking advantage of recent advances in DNN-AE, which has been successfully used for data compression and feature extraction in other fields such as image and video compression. Toward this aim, we will compress the angles in $\mathbf{\Psi}$ and $\mathbf{\Phi}$ in the spectral domain by removing their information redundancy caused by channel correlation.

While the idea is straightforward, there are challenges in the design of practical DNN-AEs that are amenable to real-world applications. The challenges lie in the following respects: i) The configuration of DNN-AEs should be meticulously selected, including the number of layers in autoencoder, the number of neurons on each layer, and the preprocessing of input data. The configuration of DNN-AEs is of great importance as it dictates the compression rate, the information loss, and the required data amount and computational power

for training. ii) The training of DNN-AEs should be online and transparent to user devices. User devices are typically constrained by their computational capability and battery power. It is desirable that the training of DNN-AEs, which is computational demanding, will not put a burden on user devices. In what follows, we propose LB-SciFi to address these two challenges.

IV. LB-SciFi: A Learning-Based Feedback Framework

To reduce the CSI feedback airtime overhead, we propose LB-SciFi for CSI compression. The core components of LB-SciFi are two DNN-AEs, which compress CSI at each STA and decompress CSI at the AP. Fig. 1 shows the basic idea of LB-SciFi, which is composed of two phases: *online training* and *real-time exploitation*. As shown in Fig. 1(a), the online training is done at the AP by taking advantage of the side information ($\mathbf{\Psi}$ and $\mathbf{\Phi}$) from existing 802.11 protocols. Once the training of DNN-AEs is completed, the AP broadcasts the weights of the DNN-AEs to all STAs and enters into the exploitation phase as shown in Fig. 1(b). In the exploitation phase, each STA uses DNN-AEs to compress its CSI and reports the compressed CSI to the AP. The AP uses DNN-AEs to decompress the received CSI for the construction of beamforming filters. In what follows, we first describe the DNN-AEs settings and then present the online training scheme, followed by the explanation of the exploitation phase and the analysis of compression capability of LB-SciFi.

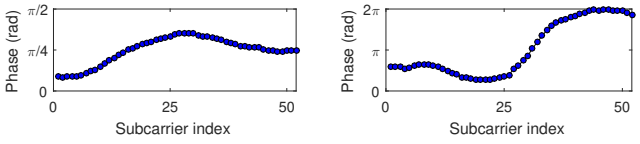
A. DNN-AEs

Autoencoder is a type of artificial neural network used to learn efficient data coding in a self-supervised manner. One of its applications is to learn a representation for a set of data for dimensionality reduction. Autoencoders are effectively used for solving many applied problems, ranging from face recognition to acquiring the semantic meaning of words. In this work, we take advantage of recent advances in DNN-AEs to compress CSI for 802.11 MU-MIMO protocols. We consider a DNN-AE as shown in Fig. 1, which is composed of two parts: encoder and decoder. The encoder will be used on each STA to compress its estimated CSI for feedback, and the decoder will be used at the AP to recover CSI for construction of beamforming filters.

Compressibility of $\mathbf{\Psi}$ and $\mathbf{\Phi}$: Before delving into the details of DNN-AEs, we introduce a metric to quantify compressibility of angles on an observation basis. The compressibility metric will lay the foundation for our design of DNN-AEs. Consider an angle sequence $\overrightarrow{\theta} = [\theta_1, \theta_2, \cdots, \theta_K]$. Denote its FFT output as $\overrightarrow{\vartheta} = [\vartheta_1, \vartheta_2, \cdots, \vartheta_K]$. Then, we define PSE of $\overrightarrow{\theta}$ as follows:

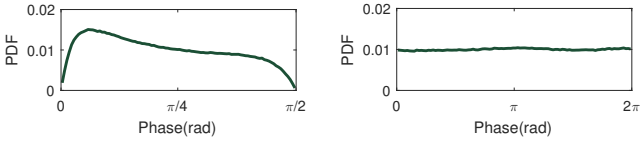
$$\text{PSE}(\overrightarrow{\theta}) = -\frac{1}{\log_2 K} \sum_{k=1}^{K} p(\vartheta_k) \log_2 p(\vartheta_k), \tag{1}$$

where $p(\vartheta_k) = \frac{|\vartheta_k|^2}{\sum_{i=1}^{K} |\vartheta_i|^2}$ [31]. Apparently, the PSE of an angle sequence is bounded in $[0\ 1]$. In real systems, since it is impossible for the AP to know the distribution and probability of θ in the course of measurement, we are therefore unable to analytically approach the entropy of angles. PSE provides a powerful tool to empirically approach the entropy of angles on observation basis. Its value reflects the uncertainty of a random angle or fluctuations of a measured angle over subcarriers. Intuitively, a low value of PSE indicates high compressibility, while a high value of PSE indicates low compressibility.



(a) An angle in $\mathbf{\Psi}$ on 52 subcarriers (measured PSE is 0.11). (b) An angle in $\mathbf{\Phi}$ on 52 subcarriers (measured PSE is 0.25).

Fig. 3: Angle instances in $\mathbf{\Psi}$ and $\mathbf{\Phi}$ as well as their PSE.



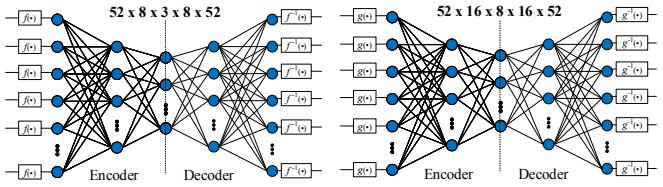
(a) PDF of the angles in $\mathbf{\Psi}$. (b) PDF of the angles in $\mathbf{\Phi}$.

Fig. 4: Distribution of the measured angles over all subcarriers and at many locations in a real-world office environment.

In WLANs, STAs are semi-stationary and work on a limited bandwidth. In such an environment, the channels between an AP and STAs are prone to be frequency-flat, and the channel responses on adjacent subcarriers are highly correlated. Fig. 3 exhibits an angle in $\mathbf{\Psi}$ and an angle in $\mathbf{\Phi}$ over 52 valid subcarriers in 20 MHz bandwidth at 2.484 GHz as well as their PSE values. It is evident that both PSE values are much less than 1, indicating the compressibility of the angles.

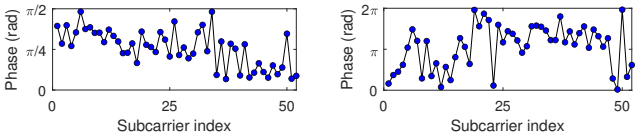
Separate DNN-AEs for $\mathbf{\Psi}$ and $\mathbf{\Phi}$: For an STA, it needs to first compress $\mathbf{\Psi}$ and $\mathbf{\Phi}$, and then report compressed $\mathbf{\Psi}$ and $\mathbf{\Phi}$ to the AP. A natural question to ask is whether an STA should use the same DNN-AE for both $\mathbf{\Psi}$ and $\mathbf{\Phi}$. To explore an answer to this question, we empirically study the compressibility of the angles in $\mathbf{\Psi}$ and $\mathbf{\Phi}$. Specifically, we collected the CSI angles for $\mathbf{\Psi}$ and $\mathbf{\Phi}$ at the STAs that were widely distributed in a real-world office environment, and plotted the PDF of the collected angles. Fig. 4 shows our measured results. We can see that the angles in $\mathbf{\Psi}$ is non-uniformly distributed, while the angles in $\mathbf{\Phi}$ are almost uniformly distributed. Based on collected CSI angles, the measured PSE of $\mathbf{\Psi}$ is 0.09, and the measured PSE of $\mathbf{\Phi}$ is 0.23. The measurement results indicate that the angles in $\mathbf{\Psi}$ and $\mathbf{\Phi}$ have different levels of compressibility. Given that the compression ratio is determined by the DNN-AE structure (the ratio of dimension of the input layer to that of the latent layer), we employ two different DNN-AEs for the compression of $\mathbf{\Psi}$ and $\mathbf{\Phi}$.

DNN-AEs Settings: Another question to ask is about the parameters of the two DNN-AEs for $\mathbf{\Psi}$ and $\mathbf{\Phi}$, including the number of layers and the number of neurons on each layer. Unfortunately, there is no systematic approach that we can utilize to determine the optimal settings for these parameters. Therefore, we resort to experimental tests. We tried different



(a) DNN-AE for $\mathbf{\Psi}$. (b) DNN-AE for $\mathbf{\Phi}$.

Fig. 5: Illustration of two different DNN-AEs for $\mathbf{\Psi}$ and $\mathbf{\Phi}$.



(a) A garbage angle in $\mathbf{\Psi}$. (b) A garbage angle in $\mathbf{\Psi}$.

Fig. 6: Examples of garbage sequences.

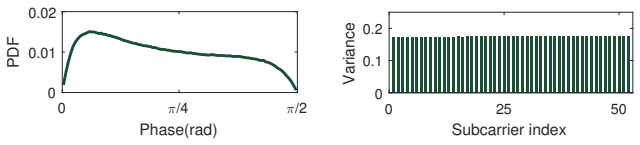
parameters for the two DNN-AEs and studied their performance in terms of computational complexity, compression rate, and compression information loss. Eventually, we choose the DNN-AEs shown in Fig. 5, where a $52 \times 8 \times 3 \times 8 \times 52$ DNN-AE is used for $\mathbf{\Psi}$ and a $52 \times 16 \times 8 \times 16 \times 52$ DNN-AE is used for $\mathbf{\Phi}$.

B. Online Training: Data Collection

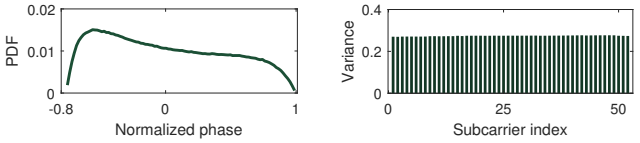
As illustrated in Fig. 1, the AP takes advantage of existing 802.11 protocols to train the DNN-AEs. That is, AP and STAs perform downlink MU-MIMO transmissions using the 802.11 protocol as shown in Fig. 2. In the meantime, the AP trains the DNN-AEs using reported CSI (uncompressed $\mathbf{\Psi}$ and $\mathbf{\Phi}$) from the STAs. By doing so, the AP can train the DNN-AEs by collecting side information from the existing MU-MIMO protocol, and the training remains transparent to the STAs. In the course of data collection, care should be taken for the following two tasks.

Avoiding Garbage-In/Garbage-Out: To collect a meaningful dataset for training DNN-AEs, the AP needs to block out garbage CSI reports from STAs. In real WLANs, an STA may fail in estimating accurate CSI due to various sources of errors such as time and frequency synchronization errors. Fig. 6 shows examples of garbage reports from an STA which failed in time synchronization. As a garbage report has intrinsically a noise-like behavior, several dominant components exist in its spectral representation. Therefore, the PSE of such a report is high likely to be overly high. The AP leverages PSE metric in (1), and blocks out the sequences with abnormal PSE. The abnormality is detected by adjusting appropriate thresholds. In our experiment, we assumed that an abnormal angle in $\mathbf{\Psi}$ has PSE≥ 0.25 and that an abnormal angle in $\mathbf{\Phi}$ has PSE≥ 0.5.

Avoiding Overrepresentation: Another important task of the AP is to prepare a balanced data set. In a typical WLAN, a *static* STA like smart TV remains at a fixed location without quitting the WLAN, while a *mobile* STA wanders through coverage range and may quit the WLAN for a while. A static STA may temporally experience correlated large-scale fading, making its historical CSI reports highly correlated. In light of this, the CSI reports from static STAs might be over-represented, making the DNN-AEs biased in favor of



(a) Probability density function and variance before preprocessing



(b) Probability density function and variance after preprocessing

Fig. 7: The probability and variance of the angles in $\mathbf{\Psi}$ before and after rectification.

themselves. To avoid overrepresentation, the AP prepares PSE distribution of angle sequences, partitions the PSE range into multiple regions, and choose a certain number of sequences within each region.

C. Online Training: Data Preprocessing

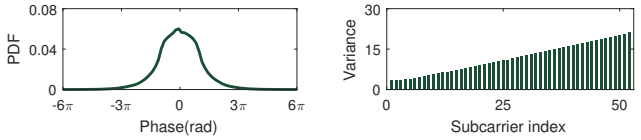
After clearing and balancing the collected datasets, the AP preprocesses the datasets before feeding them into DNN-AEs for training. In what follows, we first describe the purpose of data preprocessing and then present the preprocessing procedure for the two sets of angles.

Purpose of Data Preprocessing: To avoid biased training and boost the convergence for DNN-AEs, we wish to obtain the training datasets with a normalized zero-mean probability density function and uniform subcarrier-wise variance in the feasible space [32]. Such datasets are more likely to render an unbiased training for the DNN-AEs and yield a high compression ratio. Unfortunately, the collected angles in $\mathbf{\Psi}$ and $\mathbf{\Phi}$ do not meet these two conditions (normalized zero-mean distribution and flat subcarrier-wise variance). Therefore, we preprocess the collected datasets with the aim of rectifying their distributions to accelerate the training.

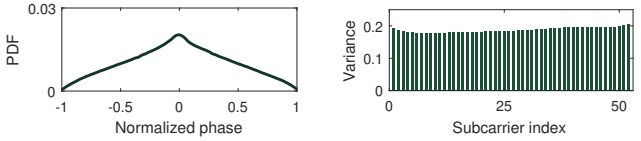
Preprocessing of Angles in $\mathbf{\Psi}$: Fig. 7(a) shows the probability and variance of the angles in $\mathbf{\Psi}$ before the preprocessing. As it can be seen, the angles in $\mathbf{\Psi}$ are non-uniformly distributed within their range. To alleviate this issue, we apply a rectification function $f(\cdot)$ at the encoder and de-rectification function $f^{-1}(\cdot)$ at the decoder, as shown in Fig. 5(a). Here, we employ $f(\psi_k) = \alpha\left(\psi_k - \bar{\psi}\right)$ as the rectification function, where $\bar{\psi}$ is the average of the angles in $\mathbf{\Psi}$ and α is a normalization constant. In our experiments, we use $\bar{\psi} = 0.68$ rad and $\alpha = 1.12$. After the rectification, the angles will have zero mean and uniform variance over different subcarriers, thereby improving the convergence of the DNN-AEs [33] and avoiding zigzag behavior in gradient descent algorithms [34].

Fig. 7(b) shows the probability and variance of the angles in $\mathbf{\Psi}$ after the preprocessing. As it can be seen, the probability density function has a zero mean after the preprocessing, which leads to a disciplined training for the corresponding DNN-AE.

Preprocessing of Angles in $\mathbf{\Phi}$: Compared to $\mathbf{\Psi}$, the preprocessing of $\mathbf{\Phi}$ is a bit more tricky. Fig. 8 shows the probability



(a) Probability density function and variance before preprocessing.



(b) Probability density function and variance after preprocessing.

Fig. 8: The probability and variance of the angles in $\mathbf{\Phi}$ before and after rectification.

density function and subcarrier-wise variance of the angles in $\mathbf{\Phi}$ measured in real WLANs. The non-uniform probability distribution, non-uniform variance, high variance on each subcarrier, and the large range (even beyond $[-4\pi,\ 4\pi]$) make the angles in $\mathbf{\Phi}$ unsuited for training. Preprocessing is needed to rectify the dataset to improve the convergence of the DNN-AE and avoid biased training.

One approach that one may think of to rectify the angles is to wrap the angles into $[0,\ 2\pi)$ using a simple function $g(\phi) = \mathrm{mod}(\phi, 2\pi)$. This approach, however, is not an effective one. Fig. 9 shows an example of this rectification function. It can be seen that the rectified angle curve appears to be discontinuous. However, the discontinuity of the rectified data cannot be captured by the DNN-AE, as illustrated in the figure. Therefore, a continuous rectification function is needed for the preprocessing of $\mathbf{\Phi}$.

In light of this requirement, we propose a piece-wise function to rectify the angles in $\mathbf{\Phi}$ before feeding them into the DNN-AE:

$$g(\phi_k) = \begin{cases} \frac{1}{2\pi}(\phi_k - 0.07) & \text{if } \min_k(\phi_k) < 0, \\ \frac{1}{2\pi}(\phi_k - 6.16) & \text{if } \max_k(\phi_k) > 2\pi, \\ \frac{1}{\pi}(\phi_k - 3.13) & \text{otherwise,} \end{cases} \tag{2}$$

for $k = 1, 2, \cdots, 52$. In this equation, the values of 0.07, 6.16, and 3.13 are the mean of the angles in their respective category and obtained from our experimental measurements. It is noteworthy that coefficient $1/\pi$ in the third equation in (2) differs from the other two. This is because the angles in this category have a small range and thus a small normalization coefficient is used for scaling.

Fig. 8(b) shows the probability density function and subcarrier-wise variance of all the angles in $\mathbf{\Phi}$ after preprocessing. Compared to the distribution and variance before preprocessing as shown in Fig. 8(a), it is evident that this preprocessing can flatten both probability and variance distributions, making the DNN-AE easy to converge.

Given that $g(\phi_k)$ is used for data preprocessing on the encoder side, an inverse function is needed on the decoder side to recover the original angles. However, $g(\phi_k)$ is a piece-wise function and it is not invertible. To address this challenge, we use two bits to indicate the sub-function used for rectification, i.e., "00" means $g(\phi_k) = \frac{1}{2\pi}(\phi_k - 0.07)$, "01" means $g(\phi_k) = \frac{1}{2\pi}(\phi_k - 6.16)$, and "10" means $g(\phi_k) = \frac{1}{\pi}(\phi_k - 3.13)$. With these two bits, the decoder is capable of constructing $g^{-1}(\phi_k)$ and inversing the preprocessing at the encoder. In the exploitation phase, each STA should send these indication bits to the AP via the over-the-air uplink channel. It is worth noting that these indication bits are of very small size compared to conventional CSI feedback.

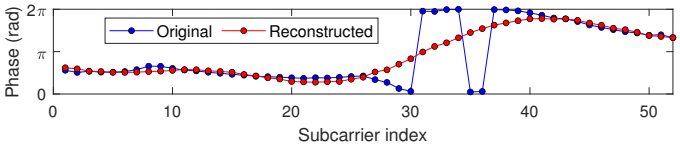


Fig. 9: Illustrating the underlying problem of the rectification function $g(\phi) = \mathrm{mod}(\phi, 2\pi)$ for the angles in $\mathbf{\Phi}$.

D. Online Training: Settings and Procedure

Training Procedure and Hyper-Parameter Tuning: We train the DNN-AEs shown in Fig. 5 using the preprocessed datasets. For the two DNN-AEs, each hidden layer is composed of a fully-connected layer followed by a batch-normalization layer to speed up the training convergence [35]. Also, rectified linear unit (ReLU) activation function is used. The DNN-AEs are trained to minimize loss function, which is defined as the relative error:

$$L(\mathbf{x}, \hat{\mathbf{x}}) = \frac{\|\hat{\mathbf{x}} - \mathbf{x}\|}{\|\mathbf{x}\|}, \tag{3}$$

where $\mathbf{x}$ and $\hat{\mathbf{x}}$ represent the input sample and the corresponding reconstructed sample, respectively. The networks are trained using Adam optimizer [36]. We started the training with an initial learning rate of 0.001 and reduced it with a decay rate of 0.98 following a step-wise approach. The batch size was set to 256. All parameters were initialized using Xavier initialization [37]. Dropout [38] is applied to all hidden layers to prevent over-fitting and improve the generalization of the model. The final architectures are the result of random search over hyper-parameters. All DNN-AEs are trained end-to-end using Pytorch v1.4 library [39].

Readiness of DNN-AEs for Exploitation: While the AP trains the DNN-AEs whenever it receives a batch of CSI reports from the STAs, a natural question to ask is about the criteria for the completion of its training phase. In our experiments, we check the loss function of validation data to determine the readiness of the DNN-AEs. If the loss function of validation data is consistently less than 1.5%, we consider the completion of the training phase and the DNN-AEs are ready to use. The AP then broadcast the weights and bias values of the encoder parts of the two DNN-AEs as well as the preprocessing parameters to the STAs, so that the STAs can reconstruct the encoder part to compress the angles in $\mathbf{\Psi}$ and $\mathbf{\Phi}$, as shown in Fig. 1(b). It is worth pointing out that the airtime overhead of DNN-AEs broadcast is not an issue for two reasons. First, the broadcast takes place once for a very long period of time. Second, the broadcast is not time-sensitive and the AP can broadcast whenever it gets the resource.

Keep Training DNN-AEs: While the AP has broadcast the DNN-AEs to the STAs, there might be some STAs incapable

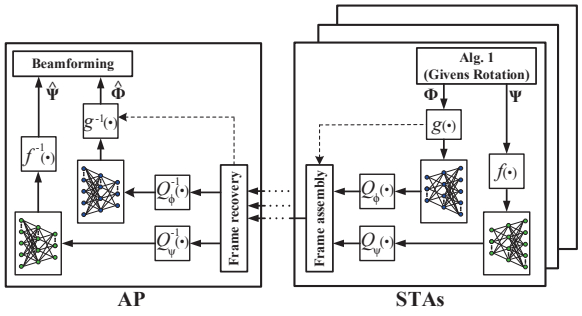


Fig. 10: Illustrating CSI compression at STA and CSI decompression at AP.

of utilizing the DNN-AEs for CSI compression. For example, some incumbent STAs may support MU-MIMO but do not support autoencoder-based CSI compression. In such a case, the AP can instruct these STAs to report CSI without compression and use the uncompressed CSI reports for the construction of beamforming filters as that in exiting 802.11 protocols. In the meantime, the AP can use the uncompressed CSI reports from those STAs to keep training the DNN-AEs.

Updating DNN-AEs: During the training in exploitation phase, the AP will periodically use validation data to check the loss function. It rebroadcasts the DNN-AEs to STAs whenever it detects a stable improvement in trained DNN-AEs. Furthermore, the AP rebroadcasts the updated DNN-AEs to the STAs whenever it observes an increase (e.g., 5%) in downlink packet error rate. Such an event simply means that the DNN-AEs in use are outdated. It is noteworthy that we did not observe a failure of the DNN-AEs in our experiments even though we moved the testbed significantly. We enforce this mechanism just to improve the robustness of our design.

E. Real-Time DNN-AEs Exploitation: CSI Compression

After the AP completes the training phase, the WLAN enters into the exploitation phase. In this phase, the AP and STAs still use the existing MU-MIMO protocols shown in Fig. 2 for downlink MU-MIMO transmissions, except that DNN-AEs are used for CSI compression of the channel feedback. In what follows, we describe CSI compression at an STA and CSI decompression at the AP, respectively.

STA-Side Operations: Fig. 10 shows the CSI compression operations at a STA. The STA first estimates the CSI and then converts the estimated CSI to two sets of angles ($\mathbf{\Psi}$ and $\mathbf{\Phi}$). Then, the two sets of angles are preprocessed and fed into the encoders of DNN-AEs for compression. After that, quantization is performed on the output, followed by frame assembly for uplink CSI report. One question to ask is how many bits should be used for the quantization of the output of DNN-AEs' encoders. While there is no analytical guidance to answer this question, we resort to experimental tests. We found that the angles in $\mathbf{\Phi}$ are more sensitive to quantization errors than the angles in $\mathbf{\Psi}$. We also observed that the setting of 5 bits for each output of $\mathbf{\Psi}$'s DNN-AE and 8 bits for each output of $\mathbf{\Phi}$'s DNN-AE is a good trade-off between performance and airtime overhead. In our experiments, we will stick to this quantization setting.

AP-Side Operations: Fig. 10 shows the CSI decompression operations at the AP, which try to recover the original CSI based on the compressed angles from an STA. The decompressed CSI will be used to construct the beamforming filters (e.g., using SVD-based precoding methods) for downlink MU-MIMO transmissions.

F. Compression Ratio and Airtime Overhead

As presented in Section III-A, the existing MU-MIMO protocols employ two types of CSI feedback quantization options and can group different numbers of subcarriers for CSI feedback. Then, the number of bits required for CSI feedback can be expressed as $N_{sc}N_a(p+q)/N_g$, where N_{sc} is the number of valid subcarriers, N_a is the number of angle-sequences in $\mathbf{\Psi}$ or $\mathbf{\Phi}$, p and q are the number of quantization bits as shown in Alg. 1, and N_g is the number of subcarriers in a group. Per the IEEE 802.11ac, we have $(p,q) \in \{(5,7),(7,9)\}$, $N_g \in \{1,2,4\}$.

LB-SciFi uses two DNN-AEs to compress the angle sequences in $\mathbf{\Psi}$ or $\mathbf{\Phi}$. Based on the DNN-AEs settings and quantization bits as shown in Fig. 10, the number of feedback bits can be expressed as $N_a(5\times 3+8\times 8+2) = 81N_a$. Therefore, the compression ratio of LB-SciFi can be written as:

$$compression\_ratio = 1 - \frac{81N_g}{52(p+q)}, \tag{4}$$

where $(p,q) \in \{(5,7),(7,9)\}$ and $N_g \in \{1,2,4\}$ as specified in the IEEE 802.11ac [1].

Based on (4), it is easy to check that LB-SciFi can achieve significant compression compared to the existing protocols. The compression ratio ranges from 48.1% to 90.3%, depending on the setting of the existing channel feedback protocol. While LB-SciFi can significantly reduce the quantity of CSI feedback, a question to ask is about the quality of its compressed feedback, including the feedback error and the impact on downlink MU-MIMO. We will provide experimental results to answer this question in the next section.

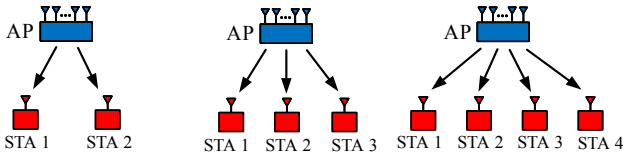
V. Experimental Evaluation

In this section, we evaluate the performance LB-SciFi in comparison with existing 802.11 protocols in an indoor wireless environment. We first describe our implementation and experimental settings and then explore the experimental data to examine the performance of LB-SciFi as well as its impacts on the overall network throughput.

Notation: For ease of exposition, we use 802.11-TiGj to denote the IEEE 802.11 MU-MIMO protocol with Type i feedback and j subcarriers in a group, where $i \in \{0,1\}$ and $j \in \{1,2,4\}$ (see Section III-A and [1]). Particularly, 802.11-T1G1 represents the finest feedback (minimum feedback error and maximum overhead), and 802.11-T0G4 represents the coarsest feedback (maximum feedback error and minimum overhead). We therefore will use these two protocols as our comparison baseline to evaluate LB-SciFi.

A. Experimental Setup and Implementation

Downlink MU-MIMO: We consider a WLAN as shown in Fig. 11, where the AP can serve two, three, or four STAs



(a) Two-user case. (b) Three-user case. (c) Four-user case.

Fig. 11: Experimental setup for downlink MU-MIMO.

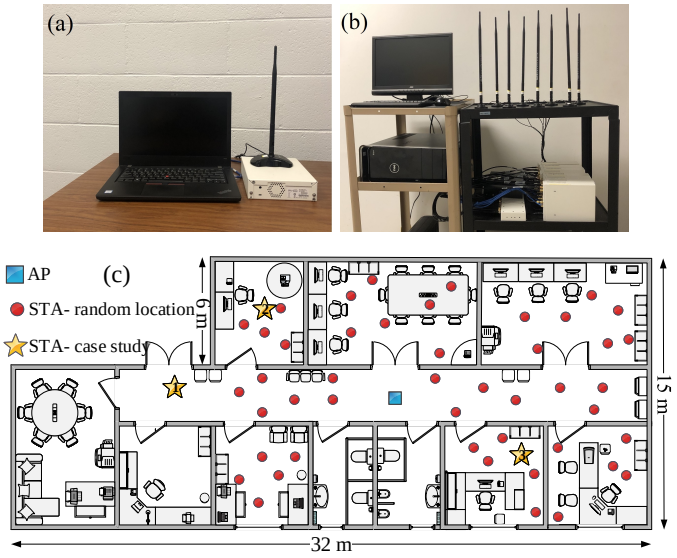
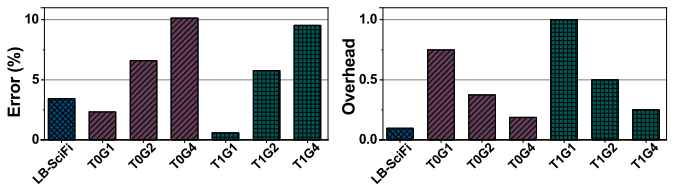


Fig. 12: Illustrating our wireless testbed and test environment. (a) Prototyped STA. (b) Prototyped AP. (c) Floor plan of tests.

simultaneously. When the AP obtains the compressed CSI reports from the STAs, it first recovers the required spatial information and then constructs the beamforming filters. While there are many different precoding methods in the literature, we used zero-forcing precoding method in our experiments owing to its popularity and ease of implementation.

Implementation of AP and STAs: Fig. 12(a–b) shows our wireless testbed. The AP and STAs are built using USRP N210 devices and general-purpose computers. Each USRP N210 device is equipped with VERT2450 Antenna for radio signal transmission/reception at 2.484 GHz. The computers are used for baseband signal processing and MAC protocol implementation. More specifically, the AP is implemented using a Dell Inspiron 3671 Desktop, which serves eight USRP N210 devices through a 10Gb fiber optic cable and a DGS-1210-20/ME Ethernet switch. Each STA is prototyped with a Lenovo ThinkPad T480 and one USRP N210 device.

Implementation of 802.11 Protocols: IEEE 802.11 protocols are implemented with the legacy PHY and MAC layers specifications. We use IEEE 802.11 frame format with 64 subcarriers for OFDM modulation. Out of these 64 subcarriers, 48 subcarriers carry payload and 4 subcarriers contain pilots. The sampling rate and carrier frequency are set to 20 MSps and 2.484 GHz, respectively. Also, the maximum transmission power is set to 15 dBm. All the necessary 802.11 baseband signal processing modules are realized with C++ in GNU Radio. For ease of implementation, our 802.11 protocols do not include user scheduling.



(a) Normalized feedback error (b) Normalized feedback overhead

Fig. 13: Feedback comparison between LB-SciFi and 802.11 protocols (T0G1, T0G2, T0G4, T1G1, T1G2, and T1G4).

Implementation of LB-SciFi: LB-SciFi is implemented on top of 802.11 protocols. It mainly deals with collecting datasets and training DNN-AEs. On our testbed, the training datasets are automatically generated in the 802.11 protocols. With the collected datasets, DNN-AEs are trained end-to-end using Pytorch v1.4 library [39] and Adam optimizer [36].

Experimental setting: Fig. 12(c) shows an office scenario where we conducted the experiments. The AP is placed at the spot marked as a blue square in the figure, while each STA is placed at a random location marked as a red circle.

B. DNN-AEs Training and Performance

Data Collection: We ran the MU-MIMO communications shown in Fig. 11(c) to collect data for DNN-AEs training at the AP. Specifically, the AP performs downlink MU-MIMO communications using the 802.11 protocols and, at the same time, it takes advantage of the CSI reports from the STAs for DNN-AEs training. To avoid overrepresentation of the collected data, we move the STAs around all locations. This can be achieved in real systems thanks to the mobility of some Wi-Fi devices such as phones and laptops. In our experiments, the AP eventually collected $60,000$ angles from the STAs for DNN-AEs training.

Feedback Error: With the completion of the first training, we examine the performance of the DNN-AEs. LB-SciFi introduces CSI error during the feedback. The feedback error can be attributed to two sources: *compression* and *quantization*. The compression error comes from the imperfection of the DNN-AEs, and the quantization error comes from the limited quantization bits. The normalized feedback error can be quantified by the loss function in (3). As a comparison baseline, we also measure the normalized feedback error in 802.11 protocols, where the error comes from the quantization of $\mathbf{\Psi}$ and $\mathbf{\Phi}$ as well as the subcarrier grouping.

Fig. 13(a) shows our measured normalized feedback errors. It can be seen that LB-SciFi has a larger feedback error than 802.11-T0G1/T1G1 protocols, and it has a smaller feedback error compared to 802.11-T0G2/T0G4/T1G2/T1G4 protocols. This is because 802.11-T0G1/T1G1 protocols do not compress the CSI in the spectral domain while other protocols naively compress CSI in the spectral domain.

Feedback Overhead: While LB-SciFi introduces larger error than 802.11-T0G1 and 802.11-T1G1, it uses much smaller uplink airtime resource for CSI feedback and therefore entails much smaller overhead. Fig. 13(b) compares the normalized feedback overhead of LB-SciFi with the existing 802.11 protocols. It can be seen that LB-SciFi entails much less

TABLE I: EVM specification in IEEE 802.11ac standard [1].

EVM (dB)	(inf -5)	[-5 -10)	[-10 -13)	[-13 -16)	[-16 -19)	[-19 -22)	[-22 -25)	[-25 -27)	[-27 -30)	[-30 -32)	[-32 -inf)
Modulation	N/A	BPSK	QPSK	QPSK	16QAM	16QAM	64QAM	64QAM	64QAM	256QAM	256QAM
Coding rate	N/A	1/2	1/2	3/4	1/2	3/4	2/3	3/4	5/6	3/4	5/6
γ(EVM)	0	0.5	1	1.5	2	3	4	4.5	5	6	20/3

overhead compared to 802.11 protocols. LB-SciFi's overhead is 0.1 while the lowest normalized overhead among IEEE 802.11 protocols is 0.2. Also, LB-SciFi's compression ratio ranges from 48.1% to 90.3%, thereby conserving much airtime resource for data transmissions.

C. LB-SciFi: Performance Metrics

We now focus on the overall performance of downlink MU-MIMO. To compare LB-SciFi and existing 802.11 protocols, we will consider the following performance metrics.

Error Vector Magnitude (EVM): EVM is widely used to assess the quality of received signals at a receiver device. It is defined as follows: $\text{EVM} = 10\log_{10}\left(\frac{\mathbb{E}[|\hat{X}-X|^2]}{\mathbb{E}[|X|^2]}\right)$, where X and $\hat{X}$ are the original and estimated signals on a subcarrier of an OFDM symbol, respectively.

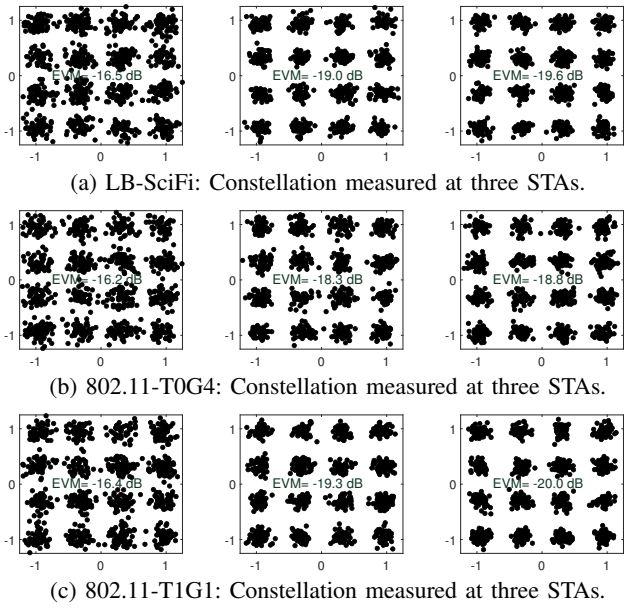
Gross Throughput: Gross throughput refers to the data rate achieved by a device (AP or STA) without taking into account the CSI overhead. For STA i, based on the EVM of its decoded signal, its gross throughput can be extrapolated as follows: $r_i = \frac{N_{\text{sp}}}{N_{\text{fft}}+N_{\text{cp}}} \cdot b \cdot \gamma\left(\text{EVM}_i\right)$, where $N_{\text{sp}} = 48$ is the number of subcarriers carrying payload, $N_{\text{fft}} = 64$ is FFT points, $N_{\text{cp}} = 16$ is the length of cyclic prefix, $b = 20$ is the sampling rate, and EVM_i is EVM of the STA i's decoded signal, and $\gamma(\text{EVM}_i)$ is the average number of bits carried by one subcarrier. This parameter is given in Table I. As such, the gross throughput at the AP can be computed by $r = \sum_i r_i$.

Net Throughput: The net throughput refers to the data rate achieved by a device after subtracting the overhead mainly caused by CSI feedback in the MU-MIMO protocols. Denote $\bar{r}$ as the net throughput achieved by the AP. Then, it can be expressed by: $\bar{r} = \frac{\sum_i t_i r_i}{\max_i\{t_i\}+t_{\text{overhead}}}$, where t_{overhead} is the time duration of overhead (NDPA, NDP, Poll, CBR, and ACK) and t_i is the time duration required by STA i for its downlink data transmission (see Fig. 2). While the value of t_{overhead} is fixed, the value of t_i is not. t_i is determined by the downlink data packet size and selected modulation and coding scheme. In real WLANs, a data packet should not exceed 2304 bytes [40]. In our experiments, we consider the maximum packet size to measure the lowest throughput gain that can be achieved by LB-SciFi.

D. Micro Performance of LB-SciFi: A Case Study

We use a case study to examine the micro performance of LB-SciFi. We consider the network shown in Fig 11(b) and place the three STAs at the spots marked with golden stars in Fig. 12(c). We compare the performance of LB-SciFi with 802.11-T1G1/T0G4 protocols.

EVM: We conduct downlink MU-MIMO transmissions using LB-SciFi, 802.11-T0G4, and 802.11-T1G1. Fig. 14 exhibits the constellation of the decoded data packet at each STA with the three protocols. As shown in Fig. 14(a), LB-SciFi achieved -16.5 dB EVM at STA 1, -19.0 dB EVM at STA 2, and -19.6 dB EVM at STA 3. In contrast, Fig. 14(b) shows the achieved EVM at the three STAs (-16.2 dB, -18.3 dB, and -18.8 dB) when 802.11-T0G4 is used; and Fig. 14(c) shows the achieved EVM at the three STAs (-16.4 dB, -19.3 dB, and -20.0 dB) when 802.11-T1G1 is used. It can be seen that LB-SciFi achieves an EVM performance similar to 802.11-T1G1 and outperforms 802.11-T0G4. We note that the constellations in Fig. 14 can be successfully decoded thanks to the powerful LDPC channel code. It is also worth pointing out that LB-SciFi is independent of modulation and coding schemes. Depending on channel quality, it can support any modulation and coding schemes.



(a) LB-SciFi: Constellation measured at three STAs.

(b) 802.11-T0G4: Constellation measured at three STAs.

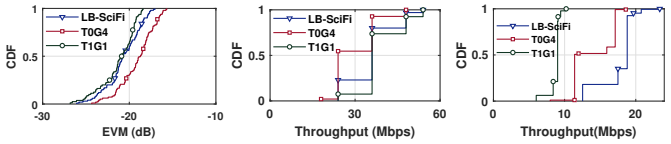
(c) 802.11-T1G1: Constellation measured at three STAs.

Fig. 14: Constellations of decoded signals at STAs when using LB-SciFi, 802.11-T0G4, and 802.11-T1G1.

TABLE II: Experimental results of the case study for comparing LB-SciFi with 802.11 protocols.

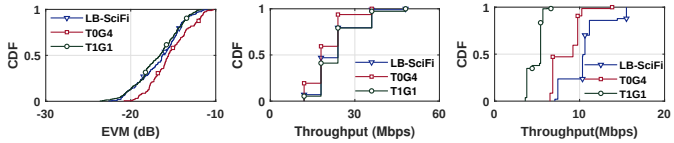
		STA 1	STA 2	STA 3	AP
LB-SciFi	EVM (dB)	-16.5	-19.0	-19.6	–
	Feedback overhead (kbit)	0.6	0.6	0.6	–
	Gross throughput (Mbps)	24.0	36.0	36.0	96.0
	Net throughput (Mbps)	15.9	23.9	23.9	63.7
T0G4	EVM (dB)	-16.2	-18.3	-18.8	–
	Feedback overhead (kbit)	1.1	1.1	1.1	–
	Gross throughput (Mbps)	24.0	24.0	24.0	72.0
	Net throughput (Mbps)	15.0	15.0	15.0	45.0
T1G1	EVM (dB)	-16.4	-19.3	-20.0	–
	Feedback overhead (kbit)	5.8	5.8	5.8	–
	Gross throughput (Mbps)	24.0	36.0	36.0	96.0
	Net throughput (Mbps)	9.4	14.1	14.1	37.6

Feedback Overhead: In the MU-MIMO transmissions, the CSI reports are transmitted from STAs to the AP using BPSK rate to ensure the feedback reliability [18]. Table II lists the feedback overhead using different protocols. As we can see from the table, LB-SciFi entails 0.6 kbit feedback overhead per STA. In contrast, 802.11-T0G4 entails 1.1 kbit feedback overhead per STA, and 802.11-T1G1 entails 5.8 kbit feedback overhead per STA.



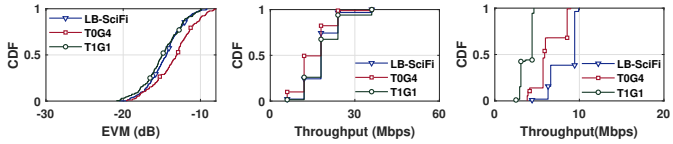
(a) EVM (b) Gross throughput (c) Net throughput

Fig. 15: Comparison of LB-SciFi and 802.11 protocols in the two-user MU-MIMO network.



(a) EVM (b) Gross throughput (c) Net throughput

Fig. 16: Comparison of LB-SciFi and 802.11 protocols in the three-user MU-MIMO network.



(a) EVM (b) Gross throughput (c) Net throughput

Fig. 17: Comparison of LB-SciFi and 802.11 protocols in the four-user MU-MIMO network.

Gross and Net Throughput: Table II lists each STA's and the AP's gross/net throughput. We can see that LB-SciFi's gross throughput is larger than 802.11-T0G4 but less than 802.11-T1G1. However, LB-SciFi's net throughput is larger than both of them. The overall net throughput gain of LB-SciFi is 41.7% over 802.11-T0G4 and 68.8% over 802.11-T1G1.

E. Macro Performance of LB-SciFi: Extensive Results

We now extend our case study to a more generic scenario. We consider the three networks in Fig. 11 and measure their performance at many different locations as shown in Fig. 12. Our evaluation methodology follows the previous case study. In what follows, we first present the measured results from the three networks and then summarize our observations.

Two-User MIMO: Fig. 15 presents the CDF of our measured EVM, gross throughput, and net throughput over all locations when the AP serves two STAs. Per Fig. 15(a), the average EVM of decoded signals at the two STAs is -20.7 dB for LB-SciFi, -19.1 dB for 802.11-T0G4, and -21.2 dB for 802.11-T1G1. Compared to T0G4, LB-SciF has 1.6 dB EVM improvement. Compared to T1G1, LB-SciF has 0.5 dB EVM degradation. Per Fig 15(b), LB-SciFi achieves an average of 35.8 Mbps per-STA gross throughput, while 802.11-T0G4 and 802.11-T1G1 achieve 30.2 Mbps and 38.7 Mbps, respectively. Per Fig 15(c), LB-SciFi achieves an average of 17.6 Mbps per-STA net throughput, while 802.11-T0G4 and 802.11-T1G1 achieve 14.1 Mbps and 8.8 Mbps, respectively. The results indicate that LB-SciFi offers 25.0% net throughput gain over 802.11-T0G0 and 99.8% gain over 802.11-T1G1.

Three-User MIMO: Fig. 16 presents the CDF of our measured EVM, gross throughput, and net throughput over all locations when the AP serves three STAs. Per Fig 16(a), the average EVM of decoded signals at the three STAs is -16.5 dB for LB-SciFi, -15.3 dB for 802.11-T0G4, and -16.8 dB for 802.11-T1G1. Per Fig 16(b), LB-SciFi achieves an average of 23.3 Mbps per-STA gross throughput, while 802.11-T0G4 and 802.11-T1G1 achieve 20.0 Mbps and 24.0 Mbps, respectively. Per Fig 16(c), LB-SciFi achieves an average of 10.5 Mbps per-STA net throughput, while 802.11-T0G4 and 802.11-T1G1 achieve 8.4 Mbps and 4.9 Mbps, respectively. Therefore, LB-SciFi offers 25.7% net throughput gain over 802.11-T0G4 and 116.8% net throughput gain over 802.11-T1G1.

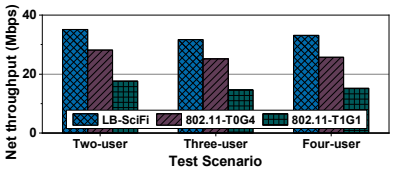


Fig. 18: Net throughput of LB-SciFi and 802.11 protocols.

Four-User MIMO: Fig. 17 presents the CDF of our measured EVM, gross throughput, and net throughput over all the locations when the AP serves two STAs. Per Fig 17(a), the average EVM of decoded signals at the four STAs is -14.5 dB for LB-SciFi, -13.4 dB for 802.11-T0G4, and -14.9 dB for 802.11-T1G1. Per Fig 17(b), LB-SciFi achieves an average of 18.3 Mbps per-STA gross throughput, while 802.11-T0G4 and 802.11-T1G1 achieve 15.6 Mbps and 19.0 Mbps, respectively. Per Fig 17(c), LB-SciFi achieves an average of 8.3 Mbps per-STA net throughput, while 802.11-T0G4 and 802.11-T1G1 achieve 6.4 Mbps and 3.8 Mbps, respectively. Therefore, LB-SciFi offers 28.9% net throughput gain over 802.11-T0G4 and 117.3% net throughput gain over 802.11-T1G1.

Summary of Observations Ultimately, we focus on the net throughput achieved by the AP. Fig. 18 depicts the total net throughput achieved by the AP when it employs these three protocols. As it can be seen, the three protocols yield similar net throughput in two-user, three-user, and four-user MIMO cases. On average, LB-SciFi achieves 26.5% net throughput gain compared to 802.11-T0G4 and 111.3% throughput gain over 802.11-T1G1, i.e, an average of 68.9% net throughput gain over 802.11 protocols.

VI. Conclusion

In this paper, we presented LB-SciFi, an online learning-based channel feedback framework for existing IEEE 802.11 MU-MIMO protocols. LB-SciFi reduces the CSI feedback overhead for 802.11 protocols by leveraging recent advances in deep neural networks to compress CSI in the spectral domain without compromising the CSI feedback accuracy. The key component of LB-SciFi is an online training scheme, which requires no dedicated training datasets but takes advantage of available side information from existing 802.11 protocols to train the autoencoders. As such, LB-SciFi can be easily plugged into existing 802.11 protocols and thus amenable to practical implementation. We have built a prototype of LB-SciFi on a wireless testbed and evaluated its performance in indoor wireless environments. Experimental results show that LB-SciFi can reduce the feedback overhead by 73% and increases the network throughput by 69% on average.

REFERENCES

[1] IEEE 802.11ac, "IEEE standard for information technology local and metropolitan area networks part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 5: Enhancements for higher throughput," *IEEE Std. 802.11ac*, 2014.

[2] IEEE P802.11ax, "IEEE draft standard for information technology – telecommunications and information exchange between systems local and metropolitan area networks – specific requirements part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment enhancements for high efficiency WLAN," *IEEE P802.11ax/D4.0*, pp. 1–746, March 2019.

[3] K. Nishimori, T. Hiraguri, T. Mitsui, and H. Yamada, "Effectiveness of implicit beamforming with large number of antennas using calibration technique in multi-user MIMO system," *Electron.*, vol. 6, no. 4, p. 91, 2017.

[4] X. Jiang, A. Decurninge, K. Gopala, F. Kaltenberger, M. Guillaud, D. Slock, and L. Deneire, "A framework for over-the-air reciprocity calibration for TDD massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 9, pp. 5975–5990, 2018.

[5] X. Jiang and F. Kaltenberger, "Channel reciprocity calibration in TDD hybrid beamforming massive MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 3, pp. 422–431, 2018.

[6] H. Zhao, G. Pottie, and B. Daneshrad, "Reciprocity calibration of TDD MIMO channel for interference alignment," *IEEE Trans. Wireless Commun.*, 2020.

[7] G. Redieteab, L. Cariou, P. Christin, and J.-F. Helard, "PHY+ MAC channel sounding interval analysis for IEEE 802.11 ac MU-MIMO," in *Proc. Int. Symp. Wireless Commun. Syst. (ISWCS)*, pp. 1054–1058, 2012.

[8] E. Perahia and M. X. Gong, "Gigabit wireless LANs: An overview of IEEE 802.11 ac and 802.11 ad," *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 15, no. 3, pp. 23–33, 2011.

[9] M. Yazid and A. Ksentini, "Modeling and performance analysis of the main MAC and PHY features of the 802.11 ac standard: A-MPDU aggregation vs spatial multiplexing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10243–10257, 2018.

[10] J. Oh, H.-J. Hong, and H.-D. Choi, "Performance analysis for channel sounding in IEEE 802.11 ac network," in *Proc. Int. Conf. Inf. Commun. Technol. Convergence*, pp. 1240–1242, 2015.

[11] T. Nabetani, N. Madhavan, H. Mori, and T. Aoki, "A novel low-overhead channel sounding protocol for downlink multi-user MIMO in IEEE 802.11 ax WLAN," *IEICE Trans. Commun.*, vol. 101, no. 3, pp. 924–932, 2018.

[12] O. Bejarano, E. Magistretti, O. Gurewitz, and E. W. Knightly, "MUTE: Sounding inhibition for MU-MIMO WLANs," in *Proc. 11th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, pp. 135–143, 2014.

[13] X. Ma, Q. Gao, J. Wang, V. Marojevic, and J. H. Reed, "Dynamic sounding for multi-user MIMO in wireless LANs," *IEEE Trans. Consum. Electron.*, vol. 63, no. 2, pp. 135–144, 2017.

[14] T. Kuber, D. Saha, and I. Seskar, "Predicting channel transition for MU-MIMO beamforming," in *Proc. IEEE 5G World Forum (5GWF)*, pp. 83–88, 2018.

[15] J. Choi, S. Choi, and K. B. Lee, "Sounding node set and sounding interval determination for IEEE 802.11 ac MU-MIMO," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 10069–10074, 2016.

[16] R. E. Guerra, N. Anand, C. Shepard, and E. W. Knightly, "Opportunistic channel estimation for implicit 802.11 af MU-MIMO," in *Proc. 28th Int. Teletraffic Congr. (ITC)*, vol. 1, pp. 60–68, 2016.

[17] H. Zeng, "INFB: A low-overhead downlink MU-MIMO scheme for wireless LANs," in *Proc. IEEE Int. Conf. Netw. Protocols (ICNP)*, pp. 389–399, 2018.

[18] H. Lou, M. Ghosh, P. Xia, and R. Olesen, "A comparison of implicit and explicit channel feedback methods for MU-MIMO WLAN systems," in *Proc. IEEE 24th Annu. Int. Symp. Personal Indoor Mobile Radio Commun. (PIMRC)*, pp. 419–424, 2013.

[19] X. Li and H. Wu, "Spatio-temporal representation with deep neural recurrent network in MIMO CSI feedback," *IEEE Wireless Commun. Lett.*, 2020.

[20] T. Wang, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based CSI feedback approach for time-varying massive MIMO channels," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 416–419, 2018.

[21] C. Lu, W. Xu, H. Shen, J. Zhu, and K. Wang, "MIMO channel information feedback using deep recurrent network," *IEEE Commun. Lett.*, vol. 23, no. 1, pp. 188–191, 2019.

[22] C.-K. Wen, W.-T. Shih, and S. Jin, "Deep learning for massive MIMO CSI feedback," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 748–751, 2018.

[23] J. Guo, C.-K. Wen, S. Jin, and G. Y. Li, "Convolutional neural network based multiple-rate compressive sensing for massive MIMO CSI feedback: Design, simulation, and analysis," *IEEE Trans. Wireless Commun.*, 2020.

[24] Q. Yang, M. B. Mashhadi, and D. Gündüz, "Deep convolutional compression for massive MIMO CSI feedback," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process. (MLSP)*, pp. 1–6, 2019.

[25] Y. Liao, H. Yao, Y. Hua, and C. Li, "CSI feedback based on deep learning for massive MIMO systems," *IEEE Access*, vol. 7, pp. 86810–86820, 2019.

[26] Y. Yang, F. Gao, G. Y. Li, and M. Jian, "Deep learning-based downlink channel prediction for FDD massive MIMO system," *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 1994–1998, 2019.

[27] Z. Liu, L. Zhang, and Z. Ding, "Exploiting bi-directional channel reciprocity in deep learning for low rate massive MIMO CSI feedback," *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 889–892, 2019.

[28] M. S. Safari, V. Pourahmadi, and S. Sodagari, "Deep UL2DL: Data-driven channel knowledge transfer from uplink to downlink," *IEEE Open J. Veh. Technol.*, vol. 1, pp. 29–44, 2019.

[29] Y. Jang, G. Kong, M. Jung, S. Choi, and I.-M. Kim, "Deep autoencoder based CSI feedback with feedback errors and feedback delay in FDD massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 833–836, 2019.

[30] J. H. Suh, J. Zhu, and O. Aboul-Magd, "System and method for quantization of angles for beamforming feedback," Feb. 6 2018. US Patent 9,887,749.

[31] A. Humeau-Heurtier, C.-W. Wu, S.-D. Wu, G. Mahé, and P. Abraham, "Refined multiscale Hilbert–Huang spectral entropy and its application to central and peripheral cardiovascular data," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 11, pp. 2405–2415, 2016.

[32] D. Kim, "Normalization methods for input and output vectors in backpropagation neural networks," *Int. J. Comput. Math.*, vol. 71, no. 2, pp. 161–171, 1999.

[33] Y. Le Cun, I. Kanter, and S. A. Solla, "Eigenvalues of covariance matrices: Application to neural-network learning," *Phys. Rev. Lett.*, vol. 66, no. 18, p. 2396, 1991.

[34] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*, pp. 9–48, Springer, 2012.

[35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[36] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *Int. Conf. Learn. Represent.*, 12 2015.

[37] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Stat.*, pp. 249–256, 2010.

[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[39] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

[40] A. Perez, *Wi-Fi Integration to the 4G Mobile Network*. John Wiley & Sons Inc., 2018.