SHRINK: Reducing MIMO Feedback Overhead in Wi-Fi with Dynamic Data-Driven Channel Sounding

K M Rumman*, Francesca Meneghello*†, Khandaker Foysal Haque*,
Francesco Gringoli§, Francesco Restuccia*

*Institute for the Wireless Internet of Things, Northeastern University, United States, †Department of Information Engineering, University of Padova, Italy, §Department of Information Engineering, University of Brescia, Italy

Abstract

The performance of multiple-input, multiple-output (MIMO) systems highly depends on the precision of channel estimates provided by the mobile users. However, the current Wi-Fi standard requires an update interval of 10 ms, irrespective of the channel dynamics. This imposes a substantial overhead for the MIMO channel estimation. Recent work mainly targets different compression strategies, potentially compromising precoding accuracy and, in turn, the network performance. In stark opposition, we propose SHRINK, a framework to dynamically adapt the feedback transmission rate to the propagation environments and performance requirements. SHRINK determines whether the users should send back their channel estimates by predicting network performance through a data-driven analysis of prior and current channel estimates. We have experimentally evaluated SHRINK using off-the-shelf Wi-Fi devices in multiple environments, including an anechoic chamber, and benchmarked its performance against several state-of-the-art approaches. Experimental results show that SHRINK reduces airtime and data overhead by 81% on average compared to the IEEE 802.11 standard without impacting the precoding performance. Moreover, SHRINK outperforms state-of-the-art approaches by an average gain of 33.6% in airtime and data overhead reduction, corresponding to an increase in throughput of 24.5%.

CCS Concepts

• **Networks → Network protocol design**.

ACM Reference Format:

K M Rumman, F. Meneghello, K. F. Haque, F. Gringoli, F. Restuccia. 2025. SHRINK: Reducing MIMO Feedback Overhead in Wi-Fi with Dynamic Data-Driven Channel Sounding. In *International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc '25), October 27–30, 2025, Houston, TX, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3704413.3764421

1 Introduction

Multiple-input, multiple-output (MIMO) has been proposed to improve spectrum efficiency in Wi-Fi by allowing simultaneous transmission of information to multiple users in the same frequency band by using multiple antennas [1], thus significantly improving spectrum efficiency [2]. While the theoretical capacity of MIMO systems increases with the number of antennas, in practice it is limited by the overhead imposed by the *channel sounding* procedure, which is essential to enable MIMO transmissions and does not scale well with the number of antennas [3]. Indeed, by using the currently adopted procedure in the IEEE 802.11ax/be standards with M antennas and bandwidth B, the feedback size increases as $M^2 \cdot B$. For example, a 16×16 Wi-Fi MIMO network at 160 MHz of bandwidth would require 77 KB per feedback instance, which is 20 times the size of a 4×4 feedback. This leads to an overhead of 61.5 Mbps for a typical sounding rate of 10 ms [4].

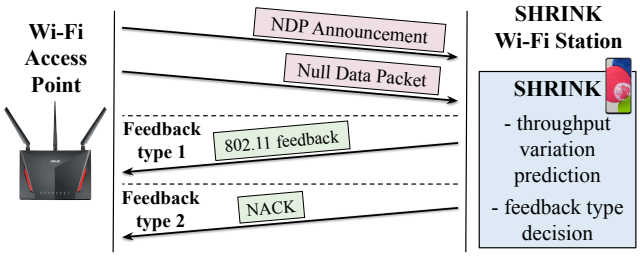


Figure 1: SHRINK sounding. Based on the data-driven throughput variation prediction, the STA feeds back to the AP the IEEE 802.11 compressed channel feedback or a shorter NACK.

To reduce overhead, researchers have proposed several feedback compression strategies [5–10] – discussed in detail in Section 6. However, compression is applied in the same way regardless of the actual changes in the wireless propagation environment. Conversely, overhead can be further reduced by adapting the sounding rate to the changes in the radio channel [11–13]. i.e., the less the channel changes over time, the less the channel has to be estimated. For example, Bejarano et al. proposed MUTE [11], which uses prior channel state information (CSI) estimates to predict the variation in the propagation channel. In [12, 13], Ma et al. and Su et al. proposed to adjust the sounding interval based on the station (STA) throughput estimated by the access point (AP). However, the decision algorithms in [11–13] are based on estimates of the channel variation or the throughput degradation based on prior data, which we show to ultimately lead to sub-optimal performance.

To address this issue, in this paper, we present SHRINK, which is executed at the STA and analyzes the *current channel* rather than using historical data. *As shown in Section 5.5, this key feature allows* SHRINK *to outperform existing approaches.* As depicted in Figure 1, SHRINK predicts the throughput variation that would occur when not feeding back the new channel estimate to the AP. Based on this, the STA decides whether to feed back the standard 802.11 feedback or a not acknowledgement (NACK) frame to inform the AP that the channel can be considered static and a channel estimate update is not needed, thus substantially reducing the overhead due to channel estimation and ultimately increasing network throughput.

MobiHoc '25, Houston, TX, USA

ACM ISBN 979-8-4007-1353-8/25/10
https://doi.org/10.1145/3704413.3764421

1.1 Summary of Novel Contributions

• We propose SHRINK, a new channel sounding procedure where the STAs decide whether to update the channel estimate available at the AP based on their *local prediction* of the throughput variation. While prior approaches select the STAs to be sounded at the AP, SHRINK is implemented and executed at the STAs. As such, the decision relies on the most recent channel and throughput measurements, thus drastically improving the performance. Note that SHRINK can be applied both in single-user MIMO (SU-MIMO) and multi-user MIMO (MU-MIMO) networks;
• We design a data-driven algorithm to predict the throughput variation that would occur by using the prior channel estimate at the AP for precoding. The algorithm is executed by every STA and is trained to automatically link the variations in the channel estimate with the changes in the throughput experienced at the STA. This information is used by each STA to decide whether to transmit the new estimate;
• We evaluate SHRINK through experimental evaluation performed on commercial-off-the-shelf (COTS) Wi-Fi devices. We perform an extensive data collection campaign in the three different environments, including an anechoic chamber. While prior work performs evaluations through software-defined radios (SDRs) and/or emulations, *in this work, we show for the first time that reducing the channel sounding is possible on COTS devices and is effective in increasing the network throughput.* We have compared SHRINK against existing baselines MUTE [11], Dynamic sounding [12] and Motion-aware MIMO [13]. Experimental results show that SHRINK reduces airtime and data overhead by 0.3 ms and 0.23 KB on average with respect to the state of the art approaches, which corresponds to an average throughput gain of 24.5%. For full reproducibility of the findings, we released our datasets and codes at https://github.com/RummaN38/SHRINK_MobiHoc.

2 Background and Motivation

We consider an IEEE 802.11 wireless network consisting of a set of STAs, each identified by an index $i \in \{1, \dots, S\}$ and equipped with N_i antennas. The STAs are served with $N_{ss,i}$ data streams each by an M-antenna AP. We use K to indicate the number of orthogonal frequency-division multiplexing (OFDM) subcarriers used. Note that K varies with the bandwidth and the sub-channel spacing Δf.

2.1 IEEE 802.11 Channel Sounding

In MIMO transmissions, data streams are linearly combined using a precoding matrix $\mathbf{W}$ derived from the channel frequency response (CFR) $\mathbf{H}_i$. The AP acquires each $\mathbf{H}_i$ through the *channel sounding* mechanism summarized in Figure 2 and detailed next.

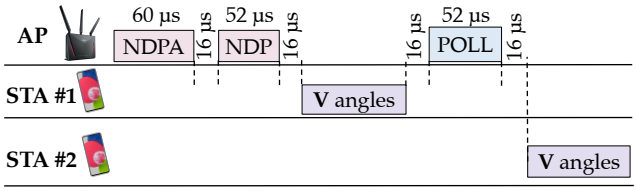


Figure 2: Sounding packets exchange in a Wi-Fi network following the standard 802.11 procedure.

The 802.11 AP periodically broadcasts null data packet (NDP) frames previously announced by null data packet announcement (NDPA) transmissions. The NDP contains training fields used to estimate the MIMO channel between the STAs and the AP. Using this NDP, each STAs estimates the CFR between each pair of transmitter and receiver antennas. This creates a $K \times M \times N_i$ matrix $\mathbf{H}_i$ for each STA. Next, the STA selects the number of OFDM sub-channels for which to feed back the information to the AP. This is done to reduce the dimension of the feedback and, in turn, the latency introduced by control data transmission. We will use $\tilde{K}$ to indicate the number of sub-channels for which the feedback is reported. This strategy is referred to as *sub-channel grouping* and the sampling factor is included by the STA in the feedback using the grouping sub-field in the feedback frame. Hence, the $\tilde{K} \times M \times N_i$ CFR is compressed through singular value decomposition (SVD). Only the first $N_{ss,i}$ columns of the right singular matrix are retained as they suffice to obtain proper precoding. This $\tilde{K} \times M \times N_{ss,i}$ matrix is the beamforming feedback matrix and is indicated by $\mathbf{V}_i$ [14]. Next, Givens rotations are used to obtain the beamforming feedback angles (referred to as '**V**' angles in the following), from which the $\mathbf{V}_i$ is fully reconstructed. The angles are identified by symbols ϕ and ψ and are then quantized. Their number and the number of bits per angle are specified in the IEEE 802.11 standard [15, 16]. Finally, the quantized angles are transmitted to the AP, which reconstructs $\mathbf{V}_i$ to be used for precoding the multiple data streams [17].

2.2 IEEE 802.11 Airtime and Data Overhead

The 802.11 channel sounding procedure requires each STA i to transmit $\left[\sum_{\ell=1}^{\min(N_{ss,i}, M-1)} (M-\ell)\right] \cdot (b_\phi + b_\psi) \cdot \tilde{K}$ bits, where b_ϕ and b_ψ are the number of bits used to quantize the ϕ and ψ angles, respectively. However, it is known that quantization leads to higher bit error rate (BER) [18]. In addition, the amount of channel feedback data remains substantial. To clarify this issue, Figure 3 presents an analytical analysis of the amount of feedback data and airtime overhead in 802.11 networks when varying the MIMO configurations and the operational bandwidth. For the analysis, we used $b_\phi = 4$ and $b_\psi = 2$ as the number of quantization bits, which is the lowest amount of bits that can be used for SU-MIMO feedback and $N_g = 16$ as the sub-channel grouping factor.

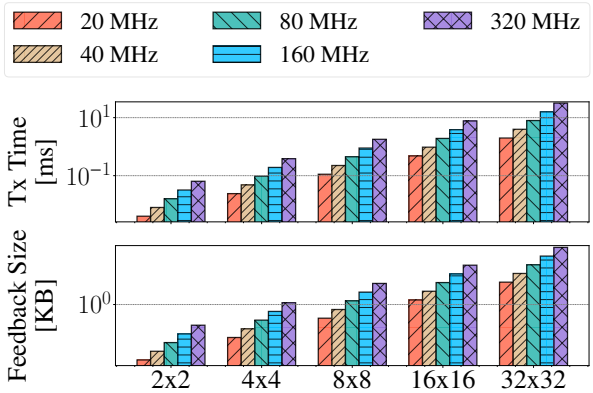


Figure 3: Feedback transmission time [ms] and size [KB] considering different MIMO configurations for the 802.11ax standard. The logarithmic scale is adopted.

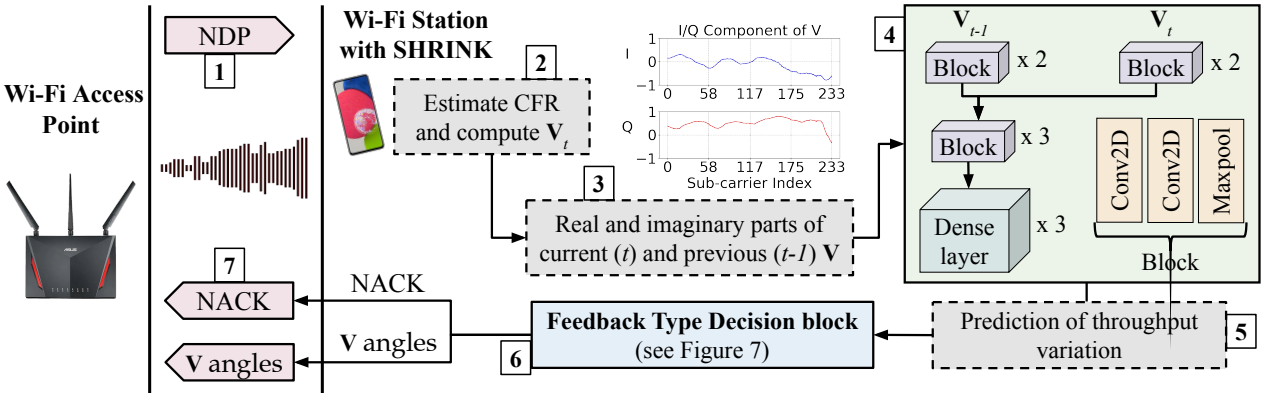


Figure 4: SHRINK sounding procedure. Through a learning-based approach, the STA decides whether to feed back to the AP the V angles or a NACK to inform the AP to use the previous channel estimate for precoding.

For example, with 32 transmitting antennas, 320 MHz bandwidth and 32 data streams, the feedback size is 95.23 KB which corresponds to an airtime overhead of about 31.7 ms. This amounts to 3 times the suggested sounding rate value of 10 ms [4] and, in turn, makes it unfeasible. In addition, excessive feedback size reduces the network throughput (control data overhead). *As such, minimizing the feedback transmission rate implicitly leads to improving the overall network performance.*

3 The SHRINK Channel Sounding

In SHRINK, each STA decides whether to update the channel information available at the AP for precoding or let the AP use the previous estimate. The decision is based on a prediction about the throughput variation that would occur if the previous feedback is used. Our objective is to provide a mechanism that builds upon the standard 802.11 sounding and allows improving the network performance. To this end, we design a message exchange procedure for the SHRINK sounding where the AP continues to transmit the NDP in broadcast to all the connected STAs. Next, each STA decides whether to transmit the compressed channel feedback as required by the 802.11 standards, or a NACK packet to inform the AP about the decision of not providing a new estimate. Figure 5 depicts an example of SHRINK, where STA1 transmits only a NACK while STA2 transmits the 802.11 compressed CSI feedback.

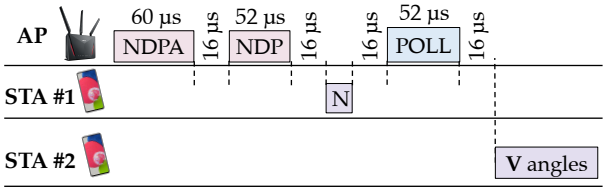
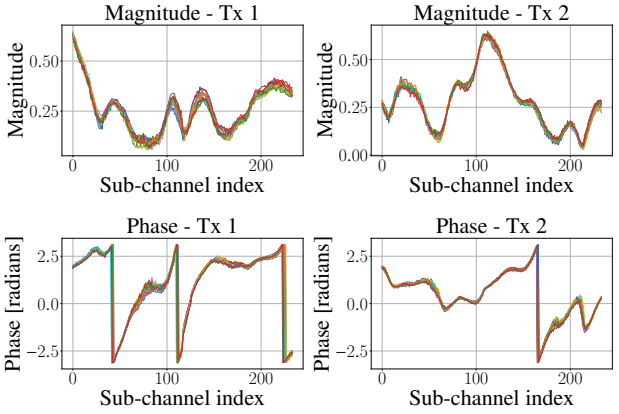


Figure 5: Example of SHRINK channel sounding. STA 1 decides not to update the estimate available at the AP and, in turn, only reports a NACK (indicated by "N").

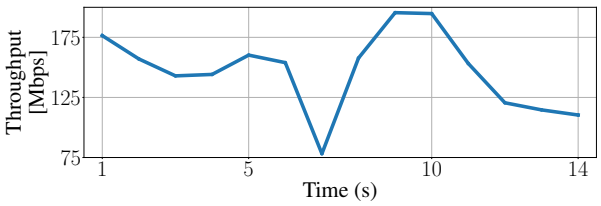
3.1 A Walkthrough of SHRINK

Figure 4 summarizes the proposed sounding procedure. The key intuition of SHRINK is to predict the expected throughput variation with a learning-based algorithm. Specifically, upon receiving the NDP (**step 1**), the STA estimates the CFR and obtains its compressed representation $\mathbf{V}_i$ following the procedure in Section 2.1 (**step 2**).

The main idea is to estimate the expected throughput variation rather than relying on the similarities between different CFRs. Indeed, we have experimentally verified that even CFR apparently very similar in shape (magnitude and phase) across consecutive sounding rounds can lead to significant throughput degradation at the STA if not promptly reported to the AP to update the precoding. To prove this point, Figure 6 shows the feedback matrices obtained at a STA together with the throughput variation over 14 different time slots when the AP keeps using the first channel estimate ($t = 1$) for precoding. The results are reported for two different transmitter antennas and show that while the feedback matrices appear similar, the throughput oscillates between 200 Mbps and 75 Mbps.



(a) Magnitude and phase of the feedback matrices $\mathbf{V}_i$ for two different antennas. The data for 14 sounding rounds (collected at 1 s rate) are plotted superimposed with different colors.



(b) Throughput variation at the STA over the 14 seconds associated with the 14 sounding rounds.

Figure 6: Correlation between the variations in time of the feedback matrices and the throughput.

To address this issue, the learning-based block – depicted in green in Figure 4 – takes as input the real and imaginary parts of the previous and current $\mathbf{V}_i$ matrices and predicts the throughput variation that would occur by using the previous channel estimate during the current sounding round (**steps 3-4** in Figure 4). Specifically, we designed a VGG-based convolutional neural network (CNN) architecture [19] to predict the variation in the throughput at the STA happening in the case the AP precodes with the previous channel estimate instead of using the current one (**step 5**). The description of the CNN-based throughput variation predictor is presented in Section 3.2. Hence, an algorithm decides whether to send the estimate (**step 6** in Figure 4), as detailed in Section 3.3. Based on this, the STA transmits a NACK or the $\mathbf{V}_i$ angles (**step 7**). For simplicity, we will omit the STA index i in the following.

3.2 SHRINK Learning Architecture

The learning block takes as input the compressed feedback matrix $\mathbf{V}_t$ estimated by the STA during the current time slot t along with the feedback matrix obtained during the previous sounding round $\mathbf{V}_{t-1}$. The feedback matrices are two-dimensional vectors, where the first dimension represents the number of OFDM sub-channels while the second indicates the number of spatial streams. Since these matrices are complex-valued, their real and imaginary parts are extracted and concatenated along the channel dimension. The output of the learning block is the predicted throughput variation at the STA, identified by $T_{\mathrm{P},t} = f(\boldsymbol{\theta}, \mathbf{V}_t, \mathbf{V}_{t-1})$, where f represents the function approximated by the learning algorithm and $\boldsymbol{\theta}$ is the vector containing its parameters. We use $T_{\mathrm{A},t}$ to indicate the actual throughput variation measured by the STA after data transmission.

The $\mathbf{V}_t$ and $\mathbf{V}_{t-1}$ matrices are first processed in a parallel fashion through two individual CNN branches, each consisting of two convolutional blocks which extract meaningful features for throughput variation prediction. Each convolutional block comprises two convolutional layers followed by a max-pooling layer, where the convolutional layers have an increasing number of filters (64, 64, 128, 256, and 512) with a kernel size of 3×3 and rectified linear units (ReLU). Padding is also applied in convolutional layers to preserve the spatial dimensions, which are only reduced through max-pooling with 2×1 kernels. In total, the number of trainable parameters is 64.63 million. The outputs of these branches are then concatenated along the channel dimension and forwarded through three convolutional blocks. The output of the final convolutional block is then flattened and processed by three fully connected layers with ReLU activation function. Finally, a dense layer with linear activation outputs the predicted throughput variation $T_{\mathrm{P},t}$.

This learning block is trained by forwarding the pairs of current and previous channel estimates in the training dataset. The predicted throughput variation is compared with the actual variation available in the training dataset for each pair of current and previous channel estimates. We use the mean squared error (MSE) as loss function, i.e.,

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N_{\mathrm{batch}}} \sum_{b=0}^{N_{\mathrm{batch}}-1} \left(T_{\mathrm{P},b} - T_{\mathrm{A},b}\right)^2, \tag{1}$$

where N_{batch} is the batch size. The network weights are updated using the Adam optimizer [20].

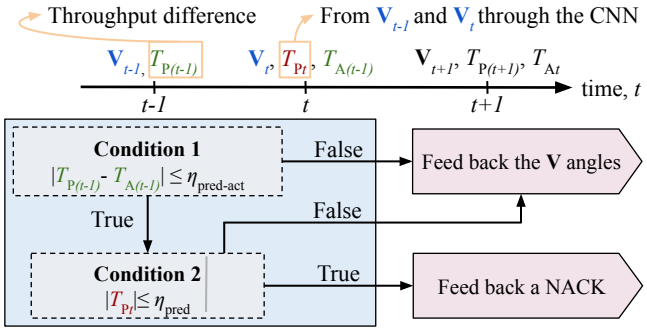


Figure 7: Feedback type decision block as detailed in Algorithm 1. The throughput variation predicted by the learning block in Figure 4 ($T_{\mathrm{P}t}$) and the difference between the predicted and the actual throughput variation at the previous step ($|T_{\mathrm{P}(t-1)} - T_{\mathrm{A}(t-1)}|$) are used to decide whether to feed back the V angles or a NACK.

3.3 Feedback Type Decision Block

The feedback type decision block processing is summarized in Figure 7 and detailed in Algorithm 1. This block decides whether to send back the compressed feedback matrix $\mathbf{V}_t$ or the NACK. The actual throughput variation is accounted for in the decision as it provides an indication of the predictive model accuracy. Specifically, at each sounding round t, the decision block evaluates the accuracy of the previous throughput variation prediction $T_{\mathrm{P}(t-1)}$ by comparing it with the actual variation $T_{\mathrm{A}(t-1)}$. A user-defined threshold $\eta_{\mathrm{pred-act}}$ decides whether the difference is acceptable for the specific application (Condition 1 in lines 1-2 in Algorithm 1).

Algorithm 1: Feedback type decision block

Input: Actual throughput variation $T_{\mathrm{A}(t-1)}$; predicted throughput variation $T_{\mathrm{P}(t-1)}$, $T_{\mathrm{P}t}$; thresholds $\eta_{\mathrm{pred-act}}$, η_{pred}
Output: Feedback

1 ***Check Condition 1***
2 **if** $|T_{\mathrm{P}(t-1)} - T_{\mathrm{A}(t-1)}| \leq \eta_{\mathrm{pred-act}}$ **then**
3 ***Check Condition 2***
4 **if** $|T_{\mathrm{P}t}| \leq \eta_{\mathrm{pred}}$ **then**
5 Feedback ← NACK;
6 **else**
7 Feedback ← $\mathbf{V}_t$ angles;
8 **else**
9 Feedback ← $\mathbf{V}_t$ angles;

Condition 1 is false. If the difference between the predicted and actual throughput variation is higher than $\eta_{\mathrm{pred-act}}$, the STA feeds back the $\mathbf{V}_t$ angles (line 9 in Algorithm 1). This indicates that the predictor was not accurate during the last sounding round. Hence, the AP is provided with the most updated channel estimate.

Condition 1 is true. If the previous throughput variation prediction is good enough, i.e., Condition 1 in line 2 of Algorithm 1 is verified, the STA checks the current throughput variation prediction $T_{\mathrm{P}t}$. If the predicted throughput variation is smaller than or equal to threshold η_{pred} (Condition 2 in line 3-4 in Algorithm 1) the STA assumes that the channel is almost static and the previous

channel estimate can be used for precoding. The STA informs the AP about this by feeding back a NACK frame (line 5 in Algorithm 1). Otherwise, the new channel estimate is fed back to the AP (line 7).

3.4 SHRINK: Putting Everything Together

As shown in Section 5.2, throughput prediction accuracy decreases over time, requiring periodic fine-tuning of predictor parameters. As such, we have designed a continual learning (CL) strategy which is executed when the prediction performance drops below a predefined threshold, as detailed in Algorithm 2.

Algorithm 2: Complete SHRINK sounding procedure

Input: $T_{A(t-1)}$, $T_{P(t-1)}$, $\eta_{\text{pred-act}}$, η_{pred}, N_S

1 **while** STATIC == **True do**
2 Estimate T_{Pt} using SHRINK CNN in Section 3.2;
3 Execute Algorithm 1 (see Section 3.3);
4 $\varepsilon_{\text{cum}} \leftarrow \sum_{\tau=t-N_S}^{t-1} (T_{P\tau} - T_{A\tau})^2$;
5 **if** $\varepsilon_{\text{cum}}/N_S > \eta_{\text{pred-act}}$ **then**
6 STATIC ← **False**;
7 Go to line 8;
8 **while** STATIC == **False do**
9 Compute correlation among the last N_S estimated $\mathbf{V}$ matrices;
10 **if** correlation > 0.9 `// channel static but inaccurate estimates`
11 **then**
12 STATIC ← **True**;
13 Apply CL to update the CNN parameters $\boldsymbol{\theta}$ using the last N_S estimated $\mathbf{V}_t$ matrices (see Section 3.5);
14 $t \leftarrow t+1$; `// next sounding round`
15 Go to line 1;
16 **else**
`// channel not static`
17 **for** $i \leftarrow 1$ **to** N_S **do**
18 $t \leftarrow t+1$; `// next sounding round`
19 Feedback ← $\mathbf{V}_t$ angles;
20 $\varepsilon_{\text{cum}} \leftarrow \varepsilon_{\text{cum}} + (T_{P(t-1)} - T_{A(t-1)})^2$;
21 Estimate T_{Pt} using SHRINK CNN in Section 3.2;
22 **if** $\varepsilon_{\text{cum}}/N_S < \eta_{\text{pred-act}}$ **then**
23 STATIC ← **True**;
24 $t \leftarrow t+1$; `// next sounding round`
25 Go to line 1 `// the channel stabilized and the previous training is still good`
26 **else**
27 $\varepsilon_{\text{cum}} \leftarrow 0$;
28 Go to line 17 `// channel not static`

After estimating the throughput variation T_{Pt} using the CNN-based prediction block (line 2, Algorithm 2, see Section 3.2) and executing the feedback type decision block (line 3, see Section 3.3), the cumulative error over the most recent N_S samples ε_{cum} (line 4) determines whether the SHRINK CNN parameters should be updated through CL. Specifically, when the average cumulative error $\varepsilon_{\text{cum}}/N_S$ exceeds the threshold $\eta_{\text{pred-act}}$, the STA assumes that the CNN prediction module is not working properly. This can happen due to one of the following reasons: (i) the environment became highly dynamic, or (ii) the environment is still almost static but the propagation statistics changed and, in turn, the CNN-based model is no longer able to predict well the throughput variation. In the first case, the STA should keep transmitting the $\mathbf{V}$ angles instead of the NACK until the wireless channel stabilizes again. In the second case, the STA should use the last N_S estimated $\mathbf{V}_t$ matrices, representing the new channel conditions, to update the CNN model's parameters through CL. Note that in the first case, refining the model would be ineffective as the wireless propagation environment is not static and, the transmission of the most updated channel estimates cannot be avoided. To determine whether the situation is (i) or (ii), the STA analyzes the correlation among the N_S most recent channel estimates (line 9 in Algorithm 2). If the correlation index exceeds 0.9, the STA assumes the channel is almost static but the previously trained model is no longer able to predict the throughput variation with adequate accuracy (lines 10-12). Hence, it uses the last N_S estimated $\mathbf{V}_t$ matrices to update the CNN model's parameters using the CL strategy detailed in Section 3.5 (line 13). Afterwards, the STA goes back to executing the SHRINK sounding procedure, i.e., Algorithm 2 execution iterates using the updated CNN (lines 14-15). On the contrary, if the correlation among the collected N_S $\mathbf{V}_t$ matrices is below 0.9, the channel is considered dynamic (line 16, Algorithm 2). Hence, the STA executes the standard 802.11 procedure, i.e., it feeds back the $\mathbf{V}_t$ angles (line 19), keeping track of the cumulative error (ε_{cum}) in estimating the throughput (lines 20-21). After N_S sounding rounds (lines 17-18), the STA checks if the average cumulative error $\varepsilon_{\text{cum}}/N_S$ is below the threshold $\eta_{\text{pred-act}}$ (line 22), which means that the channel stabilized and the previously trained CNN model remains valid (line 23). In this case, the STA resumes to use the SHRINK approach by iterating Algorithm 2 using the previously trained CNN (lines 24-25). Otherwise, the STA resets the ϵ_{cum} counter (line 27) and continues feeding back $\mathbf{V}_t$ angles (line 28), waiting for the channel to stabilize.

3.5 Continual Learning Mechanism

We employed the elastic weight consolidation (EWC) technique [21, 22] as the CL strategy to refine the model in real time. Refinement is triggered by Algorithm 2 when the channel is nearly static, but the CNN model fails to predict its behavior (line 11, Algorithm 2). The EWC procedure is summarized in Algorithm 3.

Algorithm 3: EWC continual learning procedure

Input: Model parameters $\boldsymbol{\theta}$, datasets $\boldsymbol{D}_{\text{train}}$, $\boldsymbol{D}_{\text{refine}}$
Output: Updated model parameters $\tilde{\boldsymbol{\theta}}$

1 **Step 1: Compute Fisher information matrix on the original dataset**
2 Use Equation 2 to compute the FIM for each CNN parameter θ_ℓ considering the model's gradients of the loss with respect to the samples in $\boldsymbol{D}_{\text{train}}$;
3 **Step 2: Train the CNN with the new dataset**
4 Initialize the refined CNN: $\tilde{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}$;
5 Train the CNN $\tilde{\boldsymbol{\theta}}$ using dataset $\boldsymbol{D}_{\text{refine}}$ and the loss function in Equation 3.

To avoid incurring catastrophic forgetting, a key challenge in CL, we first estimate the importance of each parameter of the CNN-based architecture in performing the prediction of the throughput variation. Specifically, we use the Fisher information matrix (FIM)

F as a measure for this. Indicating with $n \in \{1, \ldots, N\}$ the elements in the $\boldsymbol{D}_{\text{train}}$ dataset originally used to train the model, the ℓ-th element of the FIM, i.e., the one associated with the ℓ-th parameter of the CNN architecture θ_ℓ, is obtained as (**Step 1** in Algorithm 3)

$$\mathbf{F}_\ell = \frac{1}{N} \sum_{n=1}^{N} \left(\frac{\partial \mathcal{L}_n}{\partial \theta_\ell} \right)^2, \tag{2}$$

where $\mathcal{L}_n$ is the loss obtained for the n-th data sample using Equation 1, and $\frac{\partial \mathcal{L}_n}{\partial \theta_\ell}$ is the gradient of the loss with respect to the parameter θ_ℓ. At this point, the CNN-based architecture is trained using the new dataset $\mathbf{D}_{\text{refine}}$ (**Step 2** in Algorithm 3). The updated CNN parameter vector, indicated by $\tilde{\boldsymbol{\theta}}$, is initialized with the parameters of the originally trained model $\boldsymbol{\theta}$. Hence, the parameters are updated by backpropagating the gradient of the loss function defined as

$$\tilde{\mathcal{L}}(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) = \mathcal{L}(\boldsymbol{\theta}) + \lambda \cdot \mathcal{L}_{\text{EWC}}(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}), \tag{3}$$

where $\mathcal{L}_{\text{EWC}}(\boldsymbol{\theta}) = \mathbf{F} \cdot (\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta})^2$ and λ is the regularization parameter that controls the importance of the EWC penalty during the retraining phase. The SHRINK CNN-based model requires on average 95 seconds for training and 15 seconds for CL retraining. During the retraining phase, the system follows the standard 802.11 channel sounding procedure to maintain the MIMO communication active.

4 Experimental Setup

We evaluated the performance of SHRINK using a MU-MIMO testbed consisting of one AP and two STAs, deployed in a study room, a laboratory space, and an anechoic chamber, as depicted in Figure 8.

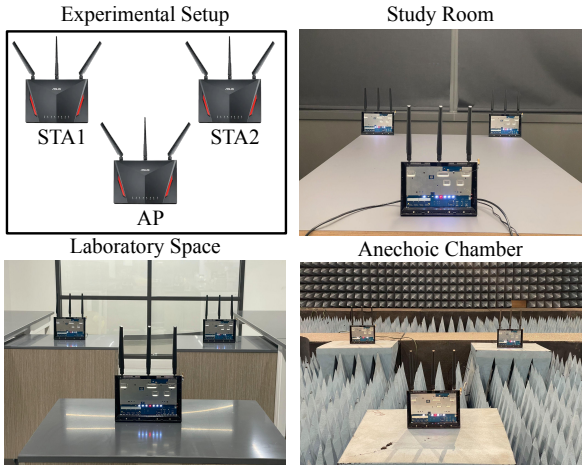


Figure 8: Experimental setup for SHRINK performance assessment in the three different environments.

4.1 Hardware Setup

We used COTS Asus RT-AC86U 802.11ac Wi-Fi routers as both AP and STAs. Four antennas were enabled at the AP while the STAs have one antenna each. Working with commercial devices allows us to effectively demonstrate the impact of our SHRINK sounding method. However, this is challenging as vendors do not provide access to inner operations such as channel estimation. Hence, to implement SHRINK, we had to reverse-engineer the channel sounding procedure on the Asus RT-AC86U devices and modify its functionality. The Wi-Fi chipset in such devices is the Broadcom BCM4365, featuring a D11 microcontroller that can be modified using the Nexmon framework [23]. We designed and implemented the necessary firmware modifications at the STA using Nexmon [24], while the AP was kept unmodified. The modifications have been implemented in one or both STAs, depending on the specific evaluation. This way, one STA can be used as the standard benchmark for performance comparison. Indeed, being served in MU-MIMO mode, the two STAs normally experience the same average throughput. We used the Wi-BFI tool [25] to reconstruct the **V** matrices from the **V** angles extracted from the Broadcom chipset at the modified STAs. The **V** matrices serve as input for our feedback type decision block to decide the type of feedback to be used (see Section 3.3).

4.2 Network Setup

The Wi-Fi network managed by the AP was operating following IEEE 802.11ac standards on channel 157 with 80 MHz bandwidth. The sub-channel grouping factor N_g was 1, while the number of bits for quantization was set to $b_\phi = 9$ bits and $b_\psi = 7$ bits for ϕ and ψ respectively. These parameters were automatically set by the unmodified AP and we do not have control over them. `iperf` sessions between the AP and each of the connected STAs (modified and unmodified) were established to evaluate the SHRINK effectiveness. UDP packets (1500 bytes-long) were transmitted to saturate channel capacity. We collected **V** matrices thanks to Nexmon-based firmware modifications and throughput directly from the `iperf` sessions. Considering the selected operational parameters, the size of the feedback matrix **V** is 234×4×1, where 234 identifies the number of OFDM data sub-carriers, and 4 and 1 are the numbers of transmit antennas at the AP and receiver antennas at the STA respectively. As we only have control over the STAs, we emulated the behavior of the AP at the reception of NACK as the feedback by feeding it back the previous channel estimate when no update is required. In turn, the unmodified AP uses such a previous channel estimate for precoding. We chose this approach to avoid modifying the AP and letting it operate following the standard 802.11 procedures.

4.3 Learning Setup

To train the CNN-based throughput variation predictor we collected real channel data for 2,500 sounding rounds on each of the different environments in almost static conditions, i.e., no people were performing highly dynamic activities in the environment, and in dynamic conditions, i.e., people moving in the scene. Note that while the measurements collected in the anechoic chamber do not include external interference sources, the laboratory space and the study room are uncontrolled wireless environments, i.e., other Wi-Fi APs and STAs were operating simultaneously with our system during data collection.

5 Performance Evaluation

In this section, we first evaluate the performance of the CNN model in predicting the throughput variation for different environments, together with the effectiveness of the CL method using different numbers of samples. Hence, we compare SHRINK with the existing 802.11 sounding and other state-of-the-art approaches. In our evaluations, we set $\eta_{\text{pred-act}}$ and η_{pred} in Algorithms 1-2 to 20 Mbps.

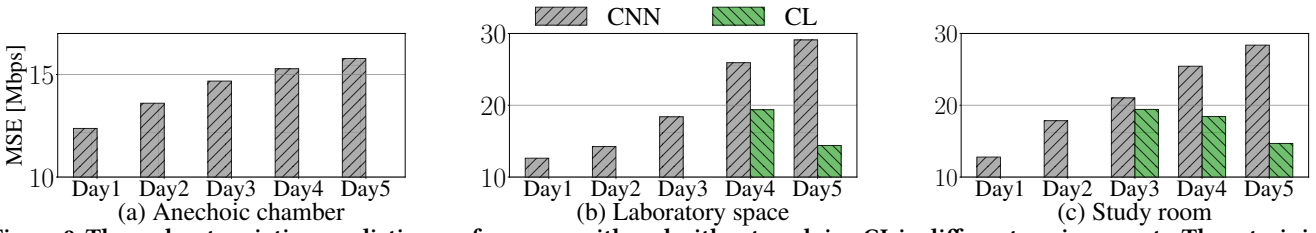


Figure 9: Throughput variation prediction performance with and without applying CL in different environments. The retraining through CL has been applied when the MSE is higher than $\eta_{\text{pred-act}} = 20$.

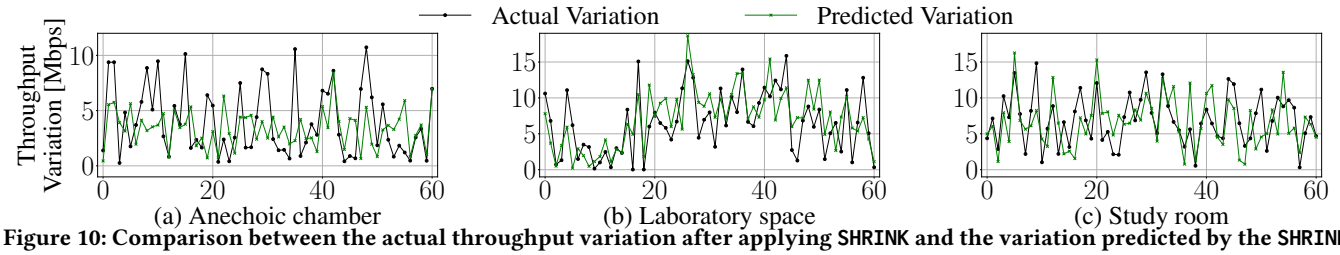


Figure 10: Comparison between the actual throughput variation after applying SHRINK and the variation predicted by the SHRINK CNN-based approach in the three evaluation environments for 60 sounding and transmission rounds.

5.1 Selection of the N_S hyperparameter

Figure 11 shows the performance of the proposed CL approach when changing the number of samples used to retrain the network at runtime. We considered the laboratory space and the study room as experimental locations. The test sample sizes for the retraining phase range from 64 to 192. Figure 11 illustrates that the MSE in predicting throughput variation decreases with more samples. Based on these results, we set N_S, i.e., the number of samples to be used for SHRINK CL, to 160, balancing retraining performance and time. Indeed, increasing the number of samples from 160 to 192 yields negligible MSE reduction (0.19 Mbps and 0.17 Mbps in the laboratory space and study room, respectively) while extending sample acquisition and retraining time. Instead, the retraining phase should remain short to reduce the overhead of continuously transmitting **V** angles, which introduces an airtime overhead for control data.

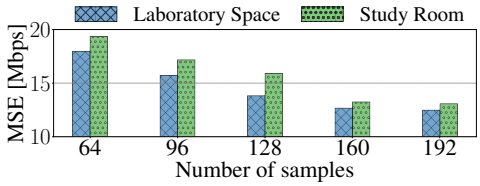


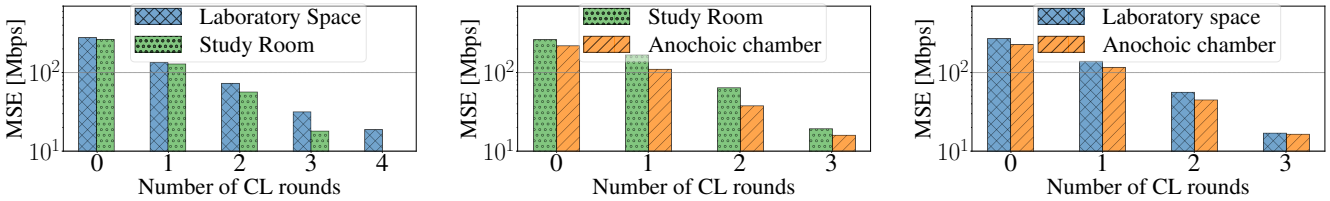
Figure 11: Throughput variation prediction performance when applying CL with 64, 96, 128, 160, and 192 new data samples in different environments.

5.2 Throughput Predictor Performance

We collected data from the three different environments and trained a separate CNN model for each environment. We evaluated the generalization performance of the trained CNN model on five different days. While no macro-movements are being performed, small variations in the propagation environment always happen, also linked with temperature and humidity variations, causing slight degradation in the performance of the CNN-based throughput variation predictor over time as depicted in Figure 9. The performance decrease is more significant in the laboratory space and in the study room, where the MSE increases by more than 15 Mbps over five different days. To counteract this, the SHRINK CL mechanism described in Algorithm 3 is executed to restore adequate throughput variation prediction performance. The second set of bars in Figure 9 illustrates the MSE decrease obtained after applying CL. The results show that when the MSE exceeds the 20 Mbps threshold, the CL block retrains the CNN to adapt it to the new conditions. Notably, in the anechoic chamber, the MSE remains consistently below the defined threshold over five different days (see Figure9a). In turn, SHRINK does not start a CL phase to retrain the model for this environment. Instead, in the laboratory environment (see Figure 9b), the MSE stays within the threshold until the third day but reaches 25.93 Mbps and 29.10 Mbps on the fourth and fifth day. Hence, SHRINK triggers the CL phase. This decreases the MSE to 19.38 Mbps and 14.40 Mbps, respectively. Similarly, in the study room (see Figure 9c), the MSE exceeds the threshold starting from the third day, (the MSE is 21.02 Mbps, 25.44 Mbps, and 28.38 Mbps respectively for the third, fourth and fifth day). Through retraining, the MSE is kept below the threshold (the values are 19.42 Mbps, 18.44 Mbps, and 14.68 Mbps, respectively). Figure 10 provides some visual examples of actual versus SHRINK CNN-based predicted throughput variations in all environments. The results show that the throughput variation predicted by SHRINK effectively follows the same trend as the actual variation, confirming the results in Figure 9.

5.3 Predictor Generalization Performance

We assessed the generalization performance of the CNN-based throughput variation predictor by training it in one environment and testing it in the others. Figure 12 presents the results for different combinations of training-testing environments. The leftmost groups of bars labeled as '0' in the x-axis, represent the MSE when using the pre-trained model in the new environments without applying retraining through CL, while the subsequent bars show the MSE reduction after an increasing number of rounds of CL refinements (see Section 3.5). Each CL round considers $N_S = 160$ new data samples, i.e., 1.6 seconds of new channel estimate recordings.



(a) Trained in the anechoic chamber, tested in the laboratory space and study room. **(b) Trained in the laboratory space, tested in the study room and anechoic chamber.** **(c) Trained in the study room, tested in the laboratory space and anechoic chamber.**

Figure 12: Environment generalization performance of SHRINK CNN-based throughput variation predictor.

The CL refinement stops as soon as the MSE threshold $\eta_{\text{pred-act}} = 20$ Mbps is met. The results show that less than five CL rounds are needed to adapt the model to new environments, and three rounds are sufficient in most cases. For example, Figure 12a shows that the MSE of a model pre-trained in the anechoic chamber reaches 19.18 Mbps and 18.37 Mbps after four and three rounds in the laboratory space and the study room, respectively. Similarly, when trained in the laboratory space, the predictor requires three CL rounds to adapt to the study room and the anechoic chamber, reaching an MSE of 19.57 Mbps and 16.23 Mbps respectively (see Figure 12b). Three CL rounds are enough also when pre-training the predictor in the study room: the MSE reaches 17.25 Mbps in the laboratory space and 16.73 Mbps in the anechoic chamber (see Figure 12c). Overall, the results show that SHRINK can quickly adapt to new environments with less than 5 seconds of new data samples. This is key for the effective deployment of sounding rate adaptation strategies in commercial Wi-Fi devices.

5.4 SHRINK Channel Sounding Performance

We evaluated the performance of SHRINK channel sounding in terms of feedback type, data and airtime overhead, and compared it with the existing IEEE 802.11ac procedure. The metrics are obtained by averaging the results over 160 sounding rounds. Figure 13 shows the feedback type selected by the two STAs in the network when they both use the SHRINK mechanism. In the laboratory space, STA1 and STA2 transmit NACK, thus avoiding **V** angle transmission, for respectively 87.5% and 85.6% of the total sounding rounds. This reduces the airtime overhead associated with control data transmission, unlocking spectrum resources that can be used for data transmissions. Figures 14 and Figure 15 compare SHRINK and 802.11 channel sounding in terms of data and airtime overhead. The average data overhead per sounding round decreases from 1.602 KB (IEEE 802.11ac) to 0.227 KB (SHRINK). Regarding the airtime overhead, while the 802.11 sounding procedure occupies the channel for 2.135 ms, SHRINK requires only 0.303 ms on average. In turn, as shown in Figure 16, SHRINK increases throughput by more than 22 Mbps compared to the IEEE 802.11 approach, with STA1 and STA2 achieving average throughput of more than 120 Mbps. A similar analysis was carried out in the study room, and the anechoic chamber (see Figure 13, Figure 14, Figure 15 and Figure 16). In the study room, the STAs transmit NACK for more than 80% of sounding rounds (see Figure 13), reducing data and airtime overhead by more than 1.34 KB and 1.79 ms on average (see Figures 14-15), with throughput increases by more than 23 Mbps and average throughput of more than 114 Mbps (see Figure 16). With the more stable wireless channel in the anechoic chamber, the two STAs select

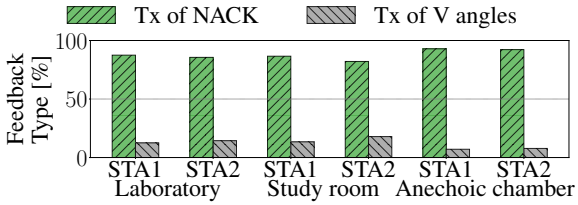


Figure 13: Feedback type distribution using the SHRINK sounding procedures in different environments.

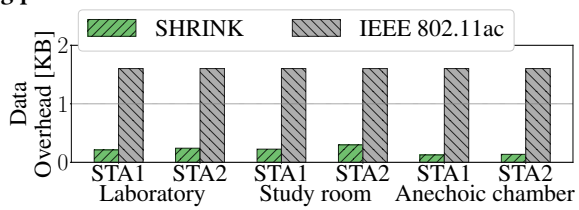


Figure 14: Data overhead using the SHRINK and the 802.11 sounding procedures in different environments.

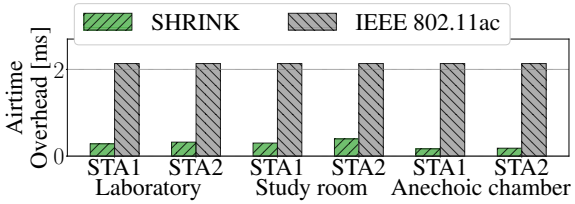


Figure 15: Airtime overhead using the SHRINK and the 802.11 sounding procedures in different environments.

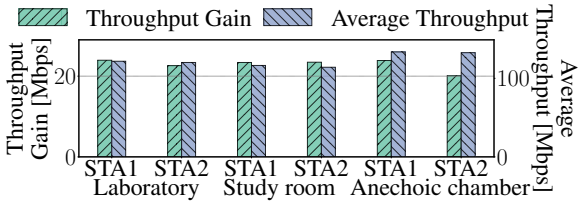


Figure 16: Average throughput and throughput gain for the SHRINK sounding procedures in different environments.

NACK feedback for 92.9% and 92.2% of the total sounding rounds (see Figure 13), reducing data and airtime overhead by an average of 1.47 KB and 1.96 ms (Figure 14 and Figure 15). Finally, throughput increases by more than 20 Mbps at both STAs with average values reaching more than 132 Mbps (see Figure 16).

We further evaluated SHRINK in a laboratory space, with and without human movement to assess the effectiveness of the feedback type decision block in Section 3.3. Results in Figure 18 show that in stable channels (no movement), SHRINK transmits NACK, reducing control data overhead. In contrast, the presence of human movement triggers more frequent transmissions of the **V** angles

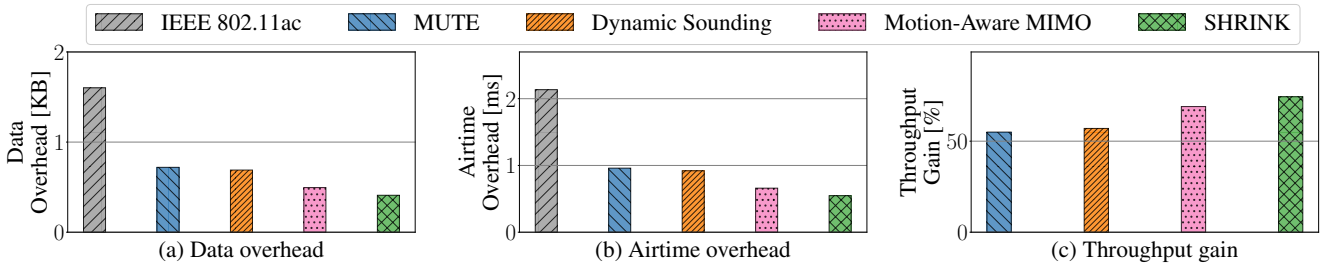


Figure 17: Comparative analysis of SHRINK, the 802.11 sounding procedure, and other three approaches in the literature for feedback rate reduction (MUTE [11], Dynamic sounding [12], and Motion-aware MIMO [13]).

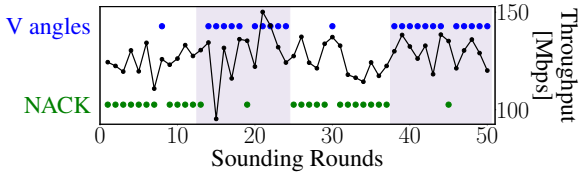


Figure 18: Feedback type distribution using SHRINK with (colored area) and without (no color) movement in the laboratory environment. The black line indicates the throughput.

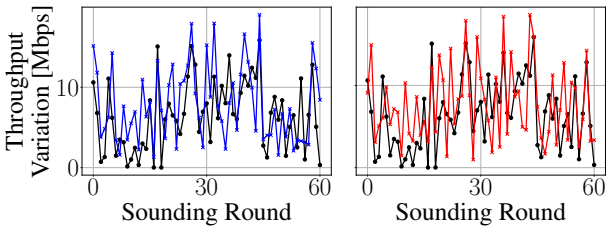


Figure 19: Actual throughput variation (black line) and variation estimated by Dynamic sounding [12] (blue line, left plot) and Motion-aware MIMO [13] (red line, right plot) in the laboratory space.

as the channel changes faster. The black line indicates the actual throughput measured by the STA and shows that although the feedback type changes, the performance does not degrade considerably.

5.5 Comparison with Existing Baselines

We compared SHRINK with three state-of-the-art approaches for sounding rate reduction: MUTE [11], Dynamic Sounding [12], and Motion-aware MIMO [13] (see the details in Section 6). Figure 17 shows that SHRINK outperforms these approaches in terms of data and airtime overhead, and overall throughput gain at the STA. Specifically, SHRINK reduces data and airtime overhead by 0.09 KB and 0.11 ms on average compared to Motion-aware MIMO, and by 0.31 KB, 0.28 KB, 0.41 ms, and 0.37 ms on average compared to MUTE and Dynamic Sounding. Finally, the throughput gain with respect to the IEEE 802.11 approach is 55%, 57%, and 69% for MUTE, Dynamic Sounding, and Motion-aware MIMO, respectively, while SHRINK achieves 74.4% (see Figure 17c). Overall, results show that performing sounding rate management at STAs, as in SHRINK, adapts more effectively to channel variations than AP-based schemes. As discussed in Section 6, in fact, SHRINK leverages recent channel estimates and throughput to select the appropriate feedback. Instead, MUTE, Dynamic Sounding, and Motion-aware MIMO rely on AP-based estimations, which are prone to estimation errors. In turn, using these estimates instead of actual values represents a sub-optimal approach for sounding rate adaptation.

5.6 Comparison with Existing Estimators

As a final evaluation, we also compared the performance of our throughput variation predictor with the throughput estimator used in Dynamic sounding [12] and in Motion-aware MIMO [13] sounding rate adaptation approaches (see the details in Section 6). In Figure 19, we show an example of the throughput variation estimated by Dynamic Sounding and Motion-aware MIMO in the laboratory space, while the prediction obtained through SHRINK is depicted in Figure 10b. Overall, the MSE in the throughput estimation (averaged over 480 samples) is 16.63 Mbps by Dynamic sounding and is 13.81 Mbps by Motion-aware MIMO, while it decreases to 13.08 Mbps when using our proposed approach (see Section 3). These results indicate that SHRINK's predictor outperforms the other approaches, further proving the effectiveness of performing the throughput prediction at the STAs where the actual channel and throughput measurements are available, rather than relying for the decision on the throughput estimated at the AP based on transmission metrics [12, 13].

6 Related Work

Channel sounding is essential to establish MU-MIMO connectivity in wireless local area networks (WLANs), enabling accurate channel assessment for precoding transmitted data or decoding received signals. It can be performed at the transmitter leveraging channel reciprocity (*implicit sounding*) or at the receiver (*explicit sounding*). The latter strategy is adopted by IEEE 802.11 [17]. However, explicit feedback leads to airtime overhead, which increases with the number of antennas and signal bandwidth [3], reducing the throughput gain that theory predicts [12]. To address this, sounding feedback compression and sounding rate adaptation have been proposed. Our proposed SHRINK approach falls within the latter, noting that both strategies are complementary to each other.

Sounding feedback compression attempts to reduce the feedback frame size. Several deep neural network (DNN)-based methods have been proposed [5–10]. CsiNet [5] employs a CNN-based autoencoder to compress estimated CSI at the receiver and reconstruct the original CSI at the transmitter for precoding. An online learning framework [6] leverages DNN autoencoders to compress CSI and side information from 802.11 protocols to train DNN-AEs, thus minimizing CSI feedback overhead while maintaining compatibility. SplitBeam [9] employs a split DNN approach with a bottleneck layer to reduce feedback and computing burden of STA. The quantization model in [10] dynamically selects quantization bits to balance

feedback accuracy and overhead. Although these approaches reduce control bits compared to 802.11, they are context-agnostic and offer a marginal improvement over channel sounding rate adaptation. Few approaches exist for sounding rate adaptation in Wi-Fi. MUTE [11] enables AP to decide which STAs update their channel estimate based on wireless channel statistics. AP triggers sounding when the measured channel variance exceeds a threshold. Opportunistic sounding rounds occur during idle downlink periods to sound as many required users as possible. The algorithm was implemented on WARP SDRs and evaluated with real-world data and emulation. Ma et al. [12] proposed a dynamic sounding approach, where the AP estimates STA throughput from successful transmissions, triggering sounding when the estimated throughput degrades. Validation was performed on a custom IEEE 802.11ac emulator with measured channel data. Su et al.[13] developed a K-nearest neighbors model in the AP to estimate throughput by correlating sequential channel measurements and finding the best match in a dataset of channel correlation and throughput. It jointly optimizes the sounding rate, spatial stream allocation, and client grouping, was evaluated via emulations with experimental data.

In contrast to these approaches that rely on coarse estimates, SHRINK operates at STAs, leveraging the actual channel estimate from the NDP and real throughput from the most recent receptions. This allows SHRINK to outperform existing sounding rate adaptation strategies as presented in Section 5.5 in Figure 17. To our knowledge, this is the first evaluation on COTS IEEE 802.11 devices, demonstrating promising results for next-generation Wi-Fi, whereas earlier studies relied on simulations, emulations, or SDRs.

7 Concluding Remarks

We have presented SHRINK, a novel explicit channel sounding mechanism to adapt the sounding feedback rate to the current wireless conditions. This is accomplished by predicting the network performance in terms of throughput variation at STAs through a learning-based approach. We have implemented SHRINK on commercial Wi-Fi devices by modifying the firmware only at the STAs while keeping the AP unmodified. Our experimental evaluation in three real scenarios (a study room, a laboratory space, and an anechoic chamber) has shown that SHRINK reduces airtime and data overhead by more than 81% compared to IEEE 802.11, thus leading to a 74% throughput increase. Currently, we have modified the Wi-Fi firmware only at the STA. Future research will include modifying the firmware at AP to capture the SHRINK NACK feedback frames and keep using the previous channel estimates for precoding data streams. Another research direction will involve integrating our SHRINK sounding rate adaptation technique with feedback compression strategies to further reduce the overhead.

Acknowledgments

This work has been supported in part by the National Science Foundation under grants CNS-2134973 and ECCS-2229472; by the Air Force Office of Scientific Research under grant FA9550-23-1-0261; by the Office of Naval Research under grant N00014-23-1-2221; and by the European commission under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU through project CAMELIA (CUP C93C24004880002) at the University of Padova and projects RESTART/EMBRACE (CUP E63C22002070006), SERICS/ISP5G+ (CUP D33C22001300002) and SERICS/SCARPHASE (CUP C89J24000580008) at the University of Brescia.

References

[1] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.
[2] Thomas L Marzetta, Erik G Larsson, and Hong Yang. *Fundamentals of Massive MIMO*. Cambridge University Press, 2016.
[3] Ehud Reshef and Carlos Cordeiro. Future Directions for Wi-Fi 8 and Beyond. *IEEE Communications Magazine*, 60(10):50–55, 2022.
[4] Matthew S Gast. *802.11ac: A Survival Guide: Wi-Fi at Gigabit and Beyond.* " O'Reilly Media, Inc.", 2013.
[5] ChaoKai Wen, WanTing Shih, and Shi Jin. Deep Learning for Massive MIMO CSI Feedback. *IEEE Wireless Communications Letters*, 7(5):748–751, 2018.
[6] Pedram Kheirkhah Sangdeh, Hossein Pirayesh, Aryan Mobiny, and Huacheng Zeng. LB-SciFi: Online Learning-Based Channel Feedback for MU-MIMO in Wireless LANs. In *Proc. of IEEE ICNP*, pages 1–11. IEEE, 2020.
[7] Ziqing Yin, Wei Xu, Renjie Xie, Shaoqing Zhang, Derrick Wing Kwan Ng, and Xiaohu You. Deep CSI compression for massive MIMO: A self-information model-driven neural network. *IEEE Transactions on Wireless Communications*, 21(10):8872–8886, 2022.
[8] Binggui Zhou, Xi Yang, Jintao Wang, Shaodan Ma, Feifei Gao, and Guanghua Yang. A Low-Overhead Incorporation-Extrapolation based Few-Shot CSI Feedback Framework for Massive MIMO Systems. *IEEE Transactions on Wireless Communications*, 2024.
[9] Niloofar Bahadori, Yoshitomo Matsubara, Marco Levorato, and Francesco Restuccia. SplitBeam: Effective and Efficient Beamforming in Wi-Fi Networks Through Split Computing. In *Proc. of IEEE ICDCS*, pages 864–874. IEEE, 2023.
[10] Ziyi Zuo, Yi Zhong, Yapu Li, Jun Zhang, and Xiaohu Ge. AIML-Enhanced Adaptive Quantization for Channel Sounding Feedback in Wireless Communications. In *Prof. of IEEE WCNC*, pages 1–6. IEEE, 2024.
[11] Oscar Bejarano, Eugenio Magistretti, Omer Gurewitz, and Edward W Knightly. MUTE: Sounding Inhibition for MU-MIMO WLANs. In *Proceedings of IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 135–143. IEEE, 2014.
[12] Xiaofu Ma, Qinghai Gao, Ji Wang, Vuk Marojevic, and Jeffrey H Reed. Dynamic Sounding for Multi-user MIMO in Wireless LANs. *IEEE Transactions on Consumer Electronics*, 63(2):135–144, 2017.
[13] Shi Su, Wai-Tian Tan, Xiaoqing Zhu, Rob Liston, Herb Wildfeuer, and Behnaam Aazhang. Motion-Aware Optimizations for Downlink MU-MIMO in 802.11 ax Networks. *IEEE Transactions on Network and Service Management*, 18(4):4088–4102, 2021.
[14] Eldad Perahia and Robert Stacey. *Next Generation Wireless LANs: 802.11n and 802.11ac*. Cambridge University Press, Cambridge, UK, Jun. 2013.
[15] IEEE Standard for Information technology–Telecommunications and information exchange between systemsLocal and metropolitan area networks–Specific requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications–Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz. *IEEE Std 802.11ac-2013 (Amendment to IEEE Std 802.11-2012)*, 2013.
[16] IEEE 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 1: Enhancements for High-Efficiency WLAN. *IEEE Std 802.11ax-2021 Amendment to IEEE Std 802.11-2020*, 2021.
[17] Eldad Perahia and Robert Stacey. *Next Generation Wireless LANs: Throughput, Robustness, and Reliability in 802.11n*. Cambridge Univ. Press, 2008.
[18] Francesca Meneghello, Khandaker Foysal Haque, and Francesco Restuccia. Evaluating the Impact of Channel Feedback Quantization and Grouping in IEEE 802.11 MIMO Wi-Fi Networks. *IEEE Wireless Communications Letters*, pages 1–1, 2024.
[19] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2014.
[20] Diederik P Kingma. Adam: A method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.
[21] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming Catastrophic Forgetting in Neural Networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.
[22] Zhizhong Li and Derek Hoiem. Learning Without Forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.
[23] Matthias Schulz, Daniel Wegemer, and Matthias Hollick. Nexmon: The C-based Firmware Patching Framework, 2017.
[24] Matthias Schulz, Daniel Wegemer, and Matthias Hollick. The Nexmon firmware analysis and modification framework: Empowering researchers to enhance Wi-Fi devices. *Computer Communications*, 129:269–285, 2018.
[25] Khandaker Foysal Haque, Francesca Meneghello, and Francesco Restuccia. Wi-BFI: Extracting the IEEE 802.11 Beamforming Feedback Information from Commercial Wi-Fi Devices. In *Proc. of ACM WiNTECH*, 2023.