AirNN: Over-the-Air Computation for Neural Networks via Reconfigurable Intelligent Surfaces

Sara Garcia Sanchez, Guillem Reus-Muns, Carlos Bocanegra, Yanyu Li, Ufuk Muncuk, Yousof Naderi, Yanzhi Wang, Stratis Ioannidis and Kaushik R. Chowdhury, *Senior Member, IEEE*

***Abstract*—Over-the-air analog computation allows offloading computation to the wireless environment through carefully constructed transmitted signals. In this paper, we design and implement the first-of-its-kind convolution that uses over-the-air computation and demonstrate it for inference tasks in a convolutional neural network (CNN). We engineer the ambient wireless propagation environment through reconfigurable intelligent surfaces (RIS) to design such an architecture, which we call 'AirNN'. AirNN leverages the physics of wave reflection to represent a digital convolution, an essential part of a CNN architecture, in the analog domain. In contrast to classical communication, where the receiver must *react* to the channel-induced transformation, generally represented as finite impulse response (FIR) filter, AirNN *proactively* creates the signal reflections to emulate specific FIR filters through RIS. AirNN involves two steps: first, the weights of the neurons in the CNN are drawn from a finite set of channel impulse responses (CIR) that correspond to realizable FIR filters. Second, each CIR is engineered through RIS, and reflected signals combine at the receiver to determine the output of the convolution. This paper presents a proof-of-concept of AirNN by experimentally demonstrating convolutions with over-the-air computation. We then validate the entire resulting CNN model accuracy via simulations for an example task of modulation classification.**

***Index Terms*—over-the-air computation, analog convolution, reconfigurable intelligent surface, convolutional neural network, programmable wireless environment**

I. Introduction

New and emerging Internet of Things (IoT) applications require collecting and processing large amounts of data, generally transmitted over the wireless channel [1]. In this context, *over-the-air* analog computation has been proposed as a alternative to all-digital approaches using acoustic [2], optical [3] and RF [4] signals. The core idea is to take advantage of additional degrees of freedom in the environment to partially offload computation into the wireless domain. Ideally, communications signals that carry information from the source are also controlled and modified by the environment such that the received signal emulates the end result of a mathematical operation. Recent results, albeit limited to pure simulation studies, have demonstrated remarkable promise for operations like data aggregation [5] and processing in recurrent neural networks [2].

S. Garcia Sanchez, G. Reus-Muns, C. Bocanegra, Y. Li, U. Muncuk, Y. Naderi, Y. Wang, S. Ioannidis and K. Chowdhury are with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, 02115 USA e-mail: ({sgarcia, greusmuns, bocanegrac}@coe.neu.edu, {li.yanyu, u.muncuk}@northeastern.edu, naderi@coe.neu.edu, yanz.wang@northeastern.edu, ioannidis@ece.neu.edu, krc@coe.neu.edu).

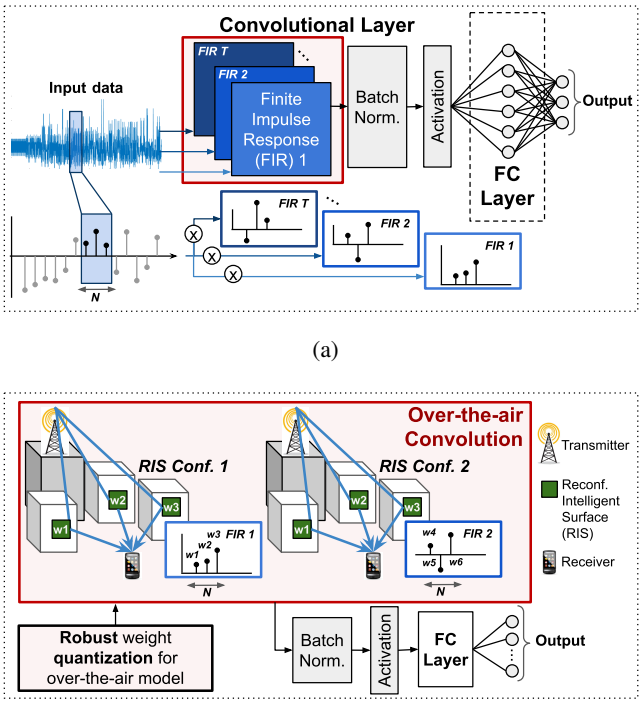


Fig. 1: (a) Generic CNN architecture, highlighting the convolution step (shown in red box), with input data in the form of raw IQ samples and where digital convolution operations in software are represented as a bank of FIR filters, (b) AirNN architecture shows the same convolution operation with over-the-air computation, using a RIS network. Different RIS configurations result in specific channel transformations, equivalent to the FIR filter responses of the digital convolution operations shown in (a).

The wireless research community has applied machine learning (ML) methods to physical layer related problems of protocol classification [6], adversarial activity detection, modulation classification [7] and RF fingerprinting [8], among others. In particular, the ML solutions proposed in these works are based on a special class of architectures named convolutional neural networks (CNNs). Fig. 1a shows a generic CNN processing chain, composed of a convolutional layer, followed by a fully connected (FC) layer that predicts the output and where raw in-phase/quadrature (IQ) samples are fed to the neural network as example of an input.

Given the interest in applying CNNs on RF signals and the promise of analog computation, this paper poses the following question: *what if we were able to realize analog convolu-*

tions using over-the-air computation accurately enough to substitute their digital equivalents in a CNN? We describe a methodology to achieve this objective and demonstrate it experimentally. We then show how this analog convolution impacts more complex mathematical operations, such as a CNN (that may have hundreds of such convolution operations).

• **Programming the Environment:** We propose a radically different approach by shifting the burden of executing the convolution operation from dedicated digital devices into the ambient environment. First, we note that the output of a convolution operation is a time series of samples. Each sample is calculated as the addition of the element-wise product between a finite impulse response (FIR) filter and a subset of sequential samples of the input data. In this work, we perform convolutions in the analog domain leveraging wireless signals and their physical interaction with the propagation environment. Specifically, we relate the addition operation to the interference phenomena that occurs when different multipath components of the transmitted signal are naturally combined at the receiver location. Moreover, we relate each of the element-wise products to the interaction between a sample of the input data (transferred to a wireless signal) with one of the wireless channel multipath components. Broadly, we relate the digital FIR filter in a convolution operation with the channel impulse response (CIR) of the wireless channel (see Fig.1). Our goal is then to *program* the CIR to implement different FIR filters to convolve an input signal with.

We propose to leverage a network of reconfigurable intelligent surfaces (RIS) that cover the principal multipath components in the wireless propagation environment. An RIS is capable of imparting changes on the phase and amplitude of the reflected signal [9]. In signal processing, such changes are characterized as complex-valued weights. Therefore, changing the configuration of the RIS is equivalent to implementing a range of complex product operations. Following this approach, specific samples of the input signal are transmitted towards individual RIS. As a result, signals interact with a carefully engineered propagation and reflection environment and combine at the receiver, emulating the mathematically equivalent outcome of passing the signal through a digital *convolutional filter* present in a CNN. As this step happens over-the-air, we refer to the resulting architecture as 'AirNN': our prototype testbed for over-the-air convolutions. We can extend this concept from a *single* convolution computation to a number of them performed in succession. Fig. 1b shows two configurations of RIS that give rise to two different desired FIR filters, which convolve with the transmitted signal.

• **Challenges in Designing AirNN:** While the domain of analog computation has existed for over a decade [10], combining wireless signals to emulate a digital convolution operation has not been attempted before. It is noted that AirNN relies on representing a convolutional filter of size N in a CNN as an N-tap FIR filter. This leverages the mathematical equivalence between the latter and the N tap discrete version of the CIR. In order to realize this equivalence in practice, we identify several challenges that need to be addressed.

First, the CIR depends on the transmitted signal and the multipath components of the environment, which the RIS can influence to a significant extent, but not perfectly. Moreover, an RIS can only implement a finite set of complex-valued weights that is dictated by its hardware constraints. This motivates the design of an efficient optimization loop: we must be able to train a CNN with quantized weights, drawn from a very limited candidate set, that corresponds to the feasible CIR set that can be attained trough the use of RIS in practice. This mapping between RIS configuration and CIR deviates over time as the wireless channel conditions change. Therefore, we need to engineer repeatable conditions during testing while accommodating ambient factors that cannot be controlled. Second, from a systems viewpoint, we need to create a network of programmable, low-cost RIS that is time-synchronized and responds to control directives to change each RIS reflection ability. Finally, we should demonstrate that the accuracy of a CNN with experimentally computed convolution in AirNN is comparable to its all-digital CNN running on a GPU.

• **Summary of Contributions in AirNN:** Our main contributions are as follows:

(1) We formulate and experimentally demonstrate the theory that maps digital (processing-based) and analog convolutions with over-the-air computation using programmable RIS.

(2) We propose a method to train CNNs with a quantized set of weights drawn from the RIS-engineered candidate set without appreciable loss of accuracy for a task of modulation classification, compared to unconstrained training. We include measures to increase resiliency when the wireless channel changes over time.

(3) As a systems contribution, we implement a software-framework to control the RIS network called AirNNOS that synchronizes and aligns start times of the transmitters and the receiver, as well as reconfigures the RIS on demand to change their reflection coefficients.

(4) Given the measured error of the convolution performed in AirNN, we show through simulations that the experimentally derived analog convolution is accurate enough to run inference on trained neural networks, with an average deviation in testing accuracy of 3.2% for a range of medium-to-high SNR of [6, 30] dB compared to classical, GPU-based inference.

II. Related Work

The area of analog computing is in a nascent stage [11][12]. Within the physics community, the work in [11] surveys the state-of-the-art metastructures for performing analog computation. The seminal work in [12] uses a chaotic cavity as a random medium and a simple phase- binary metasurface reflecting-array to shape the wave field and perform desired operations. We note that a variety of approaches spanning digital, analog, hybrid and FPGA-based solutions have been studied to accelerate training and inference in NNs [13]. The authors in [14] propose a method to train end-to-end analog NN using stochastic gradient descent by varying the conductance of programmable resistive devices and diodes. In-situ learning for a memristor-based multi-layer perceptron is demonstrated in [15]. In [16] the authors implement an optical neural chip to realise complex-valued NN. Despite

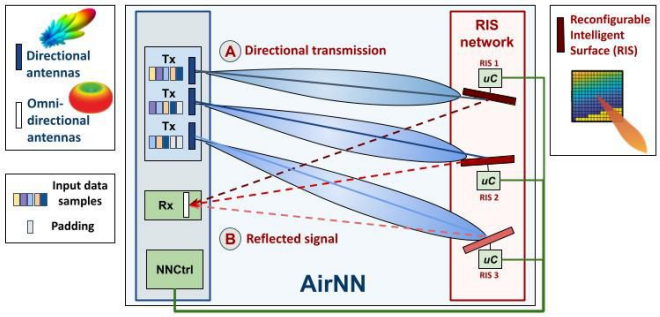


Fig. 2: AirNN system components and transmission sequence: AirNN uses different directive antennas to forward the signals of interest to the network of programmable RIS, controlled via the neural network controller (`NNCtrl`) module. The RIS reflect the signals with desired channel transformations that combine at the receiver.

their promising results, all these approaches propose on-chip implementations, an essential difference from AirNN that uses the wireless propagation environment as part of the computation entity. The authors in [2] leverage wave physics properties to experimentally demonstrate an analog recurrent NN using acoustic signals. Apart from the difference on the use of acoustic -instead of RF- signals, their approach lacks the programmability provided by use of RIS.

Specific to the RF domain, Over-the-Air Computation (AirComp) has been receiving special attention for solving problems such as data aggregation [22], efficient battery recharging trough beamforming [23][24] and local model uploading for federated learning [25] [26]. These works however, target different applications from what we achieve in AirNN, i.e., over-the-air computation for realizing convolutions that are part of a CNN. Our focus on the convolution operation is motivated by the remarkable performance that CNNs have shown within the deep learning community in fields such as computer vision, signal processing or RF signal classification [30]. This performance has attracted numerous research efforts towards realizing convolution implementations, as this processing step alone consumes over 80% of the total computation during the forward propagation step [31].

Regarding the use of RIS, multiple works in the RF domain have included them as part of their solution to perform product operations. Several works enhance the channel conditions by compensating for destructive interference [27], boosting the received power [28] or maximizing the achievable hybrid rate of all users in a network [29] by configuring a network of RIS. However, all above works ([22]- [29]) are validated in simulation, not providing insights on the implementation feasibility of their approaches. Different from these works that mostly rely on AirComp to perform data aggregation tasks, this paper is, to the best of our knowledge, the first experimental demonstrator of using RF signals and a network of RIS to perform over-the-air synchronised products and additions on a prototype testbed, as an alternative to digital convolutions. It is also the first work that analyses the cumulative effect of over-the-air computations on more complex processing that includes multiple convolution operations in a CNN.

III. AirNN Operational Overview

In AirNN, we perform convolutions making use of a network of programmable RIS, multiple transmitters (Tx) and a single receiver (Rx), as shown in Fig. 2. The network controller (`NNCtrl`) orchestrates all processes between transmitters, receiver and RIS in a centralized manner. First, it creates several copies of the input signal and introduces one sample delays among different copies through padding (see Fig. 2). Each of these copies are then fed to different transmitters. Following this step, each transmitter forwards its version of the input signal using a directional antenna (shown by link `A`) towards a specific RIS. The RIS effect on its incident signal is equivalent to one sample obtained from the element-wise product operation in the convolution. The network controller (`NNCtrl`) adjusts the reflection angles of the different RIS, which modifies the taps of the convolutional filter. The `NNCtrl` also adjusts transmission time with sample level accuracy for all transmitters to ensure that the reflected copies of the signal (shown by link `B`) combine in a deterministic manner at the receiving antenna. Once all these copies combine, the cumulative effect at the receiver resembles the processing of the same input signal as if it passed through a convolutional layer used in a CNN.

As discussed in Sec.I, assuming the availability of RIS configurations that perfectly replicate any targeted convolutional FIR filter is not realistic. Therefore, the `NNCtrl` is tasked to train the network with a quantized set of weights, dictated by the set of reflections that our network of RIS can generate. During inference, the `NNCtrl` notifies the RIS network with the updated realizable RIS configurations that result in the desired convolution. It uses a dedicated control plane that interacts with the microcontrollers at the RIS (see Fig.2).

IV. Convolutions with Over-the-Air Computation

In this section, we first explain the theory behind AirNN, namely, how to map the computation of digital convolutions to over-the-air signal transformations by the wireless channel. We then describe how RIS help us engineer such transformations as well as the system challenges we need to address to experimentally demonstrate such concept in AirNN.

A. Theory for Mapping the Process of Convolution

• **Convolution in a Digitally Constructed CNN:** In a given CNN, the convolutional filters are learned during the training process. These filters activate neurons when a specific feature of interest is detected during testing. For 1-D inputs to the CNN, typical for streaming IQ samples from a wireless signal, each of such filters can be represented as an FIR of length N i.e. N taps, filter order $L = N - 1$. This essentially is a vector of N complex weights, each weight defining a specific amplitude and phase of that particular filter tap. As an example, consider the output of a filter of length N in Eq. 1, where $\mathbf{w} = \{w_0, w_1, ..., w_{N-1}\} \in \mathbb{C}$ are the complex weights that are applied to the incoming stream of samples. The filter

order L also gives the number of input samples needed to generate a single sample at the output.

$$y[n] = w_0 * x[n+\frac{L}{2}] + \cdots + w_{\frac{L}{2}} * x[n] + \cdots + w_L * x[n-\frac{L}{2}], \quad (1)$$

• **Convolution in the Wireless Channel:** Our goal is simple: we wish to artificially construct signal transformations in the physical environment during testing that precisely maps to the above vector **w** that we obtained during training time. We leverage the fact that, when a signal is transmitted over the air, the reflections from the environment cause copies of the same signal to arrive at the receiver with different amplitude, phase, and time delays, collectively referred to as *multipath*. This phenomenon is characterized by the CIR, where each path is defined by the tuple of complex transformations in amplitude and phase and the instant of arrival at the receiver. This multipath results in an FIR filter of order $N-1$, where N is the total number of paths. Here, the first path is associated with the Line of Sight (LoS) component, whereas the $N-1$ later paths arise from Non-Line of Sight (NLoS).

B. Engineering Convolutions using RIS

In AirNN we use N *programmable paths* to implement an N tap FIR filter. Each programmable path is created by focusing the signal towards an RIS that is configured to implement a feasible FIR tap. In AirNN, an RIS is a planar array of passive reflective antennas, where each such antenna has a selectable range of impedance matching circuits. These circuits are programmable, and by activating one over the others, we change the impedance of the corresponding reflective antenna. This alters the antenna reflection coefficient, which then changes the phase of the reflected signal. The RIS-engineered reflections allow flexibility in imparting the desired complex-valued amplitude and phase changes to the signal travelling on a given multipath component. Thus, the signal reflection upon the RIS implements the product operation in the convolution. However, the set of candidate options is limited, i.e., the feasible code-book is constrained by the number of available RIS, the selectable circuit combinations within each RIS reflective antenna array, and the geometry of the propagation environment.

C. Systems Challenges in AirNN

While the concept of AirNN is intuitive, there are several systems challenges for practical realization, as we briefly covered in the introduction and further describe below.

(Ch1) Complex-Valued Convolutions: Complex numbers are used jointly to represent amplitude and phase information in the RF domain. Thus, mapping real-valued convolutional layer filters to the complex-valued CIR is not feasible. We can only use complex-valued neural networks, as we describe in Sec. V-A.

(Ch2) RIS Based Weight Constraints: The number of possible FIRs that we can engineer via RIS is limited. In the digital domain, this constrains the set of feasible FIR filters that can be used during the CNN training stage. Thus, AirNN must quantize the CNN weights that correspond to only realizable (i.e., RIS-engineered) FIR filters, as given in Sec. V-B.

(Ch3) Receiver Noise: Even if the channel remains time-invariant and the RIS configuration are static, there exists thermal noise. We need to account for this stochastic noise, especially as the reflected signals are low in amplitude and barely above noise floor. We explain how we achieve this for additive white Gaussian noise via a correction factor in Sec. V-C.

(Ch4) RIS-Path Separation: The FIR filter taps that we obtain through AirNN must be equally spaced in time, as is also assumed in the digital version. In the wireless domain, this is challenging as the arrival time of the signal depends on separation distances and the sampling rate. AirNN addresses this via a multi-transmitter (see Fig. 2), that ensures sufficient path separation. We explain this in Sec. VI-A.

(Ch5) Meaningful CIR Variations: The LoS path dominates over the NLoS paths resulting from RIS reflections in terms of received signal strength. To ensure that the artificially constructed NLoS paths shape the CIR precisely (despite the overbearing LoS path), we use directional antennas at the transmitters as explained in Sec. VI-B.

(Ch6) Channel Variations: When the wireless channel changes, prior configured RIS may generate older and outdated CIR values. To prevent re-training the neural network or repeating the mapping between RIS configurations and generated CIR, AirNN compensates for channel variations from a pre-determined baseline, as we show in Sec. VI-C.

(Ch7) Precise Synchronization: Long symbol times can disrupt the system as the CIR may change beyond the estimated value. Given the concise time window to perform a convolution, all transmitters must adjust their start time to achieve μs-level synchronization, for Mbps-level data rate. AirNN solves this problem by padding the sequence at each transmitter with zeroes, precisely achieving one sample delay between any two successive signals, as detailed in Sec. VII-C.

V. AirNN Neural Network Design

In this section, we explain how we design AirNN by addressing the challenges Ch1, Ch2 and Ch3. We then address the remaining challenges in Sec. VI.

A. Design Complex-Valued CNN (Ch1)

To facilitate the mapping between the neural network weights and the RIS-engineered CIR, we design a neural network model based on complex-valued data and weights [32]. Given that the convolution operator ($*$) is distributive, we express the output of a complex convolutional layer ϕ as:

$$y = \phi_{w_R}(x_R) - \phi_{w_I}(x_I) + j(\phi_{w_I}(x_R) + \phi_{w_R}(x_I)) \quad (2)$$

where y is the output of the complex convolution, x and w represent the input and weights of the convolutional layer and $x_{R/I}$, $w_{R/I}$ are the real/imaginary parts of x and w, respectively. The distributive property also applies to the product-sum operation of fully connected (FC) layers. Thus, we design complex-valued layers (ϕ_w) using two real-valued layers, where each one of them independently represents the real (ϕ_{w_R}) and imaginary parts (ϕ_{w_I}). The seminal work in [32] provides a detailed explanation of complex neural network theory and implementation.

B. Constrained Weight Quantization (Ch2)

We use a quantization-enabled approach to train the neural network with the set of feasible weights provided by the RIS-engineered environment. Let the weights of a complex convolution layer be:

$$\begin{aligned} W =&\{w_1, ..., w_f, ..., w_F\}, \quad w_f \in \mathbb{C}^N \\ w_f =&[w_f^1, ..., w_f^n, ..., w_f^N], \quad w_f^n \in \mathbb{C}, \end{aligned} \tag{3}$$

with $w_f \in \mathbb{C}^N$, $\mathbb{C}^N$ being a complex-valued N dimensional space, $w_f^n \in \mathbb{C}$ and where W is the set of F FIR filters (w_f) with length N that represent the layer weights. As we described in Sec. IV-A, there is limited freedom in implementing an FIR filter using RIS. Therefore, we constrain the weights w_f^n for each filter tap with index n to a candidate set S_n of implementable values, defined as:

$$S_n = \{c_1^n, ..., c_s^n, ..., c_{|S_n|}^n\}, \quad c_s^n \in \mathbb{C}, \quad 1 < n < N. \tag{4}$$

Here $|S_n|$ is the size of the constrained set and c_s^n represents each of its complex-valued elements. During training, we compute the Euclidean distance (D) from every individual weight $w_f^n \in W$ to all weight candidates $c_s^n \in S_n$. Then, we define the nearest neighbor of w_f^n as:

$$w_f^{n\prime} = \arg\min_{c \in S_n} D(c, w_f^n). \tag{5}$$

While training the model, the weight values w_f^n are rounded to their nearest neighbors $w_f^{n\prime}$ to perform forward propagation, following Eq. 5. However, the derivative of the rounding function is zero throughout and cannot be trained via classic backpropagation. We solve this by employing the Straight Through Estimator (STE) approach [33], [34], which assumes the derivative of the discrete rounding function to be **1**. While other approaches based on ADMM [35] have also been proposed, we select STE due to its faster training and convergence. Then, the forward and backward propagation steps can be expressed as:

$$\textbf{Forward: } \mathcal{L} = \phi_{w'}(input); \quad \textbf{Backward: } \frac{\partial \mathcal{L}}{\partial w} = \frac{\partial \mathcal{L}}{\partial w'}, \tag{6}$$

where $\mathcal{L}$ can be any form of loss function. Here, the gradient of w is approximated to the gradient of w', which is the fundamental working principle of STE.

C. Handling Errors in Weights (Ch3)

As we mentioned in Sec. IV-C, the receiver introduces thermal noise that causes random variations, denoted henceforth as $\epsilon \in \mathbb{C}$, into the RIS-engineered CIR. Such CIR variations follow a Gaussian distribution with standard deviation σ, i.e., $\epsilon \sim \mathcal{CN}(0, \sigma^2)$ [36].

Due to noise and changing wireless environment, the current CIR may have a mismatch with the filters identified by the RIS, and yet we desire the CNN to be robust without appreciable fall in accuracy. In order to solve this problem, we modify Eq. 5 by adding the term ϵ, as given below:

$$w_f^{n\prime} = \arg\min_{c \in S_n} D(c, w_f^n) + \epsilon. \tag{7}$$

As opposed to previous data augmentation approaches, the variable ϵ is applied during training directly to the weights to increase the robustness of the model as well as during testing. In each forward propagation step, weights are first quantized to the target constraint and noise is added. After the forward loss has been computed, we use backpropagation and obtain gradients for w'. As previously mentioned, STE is employed to approximate gradients for w, such that w is updated via Stochastic Gradient Descent (SGD).

VI. AirNN Transmitter Design

In this section, we address the design challenges Ch4, Ch5 and Ch6 introduced in Sec. IV-C.

A. Multi-Transmitter (Ch4)

The straightforward implementation of FIR filter taps in the CNN requires (i) constant inter-path time arrivals from consecutive RIS paths, i.e., $t_{RIS_{i+1}} - t_{RIS_i} = \Delta_t, \forall i \in \{0, ...N-1\}$, and (ii) exact match between these inter-path time arrivals and the communication symbol time, i.e., $\Delta_t = Ts$. Here, the first condition imposes a hard constraint on the physical deployment of RIS in the environment, forcing all RIS paths lengths to be *exact* multiples of one another. To achieve this high (sample-level) precision, AirNN accommodates a software-based temporal adjustment over the transmitted frames, as we discuss in Sec.VII-C. The second condition requires sampling rates ($Fs = 1/Ts$) that may not be compliant with the expected rate at the receiver. For example, for a total separation of 2m between two signal paths, the arrival time difference is 66.7 ns, which needs a sampling rate of up to 150 MS/s. AirNN solves this via a multi-transmitter system, where each transmitter sends the signal with a time delay of precisely one sample with respect to the next, maintaining equal spacing between arriving signals. For instance, with $Fs = 1$ MS/s, we create a convolution output sample per microsecond if all signal paths are equal in traversed distance.

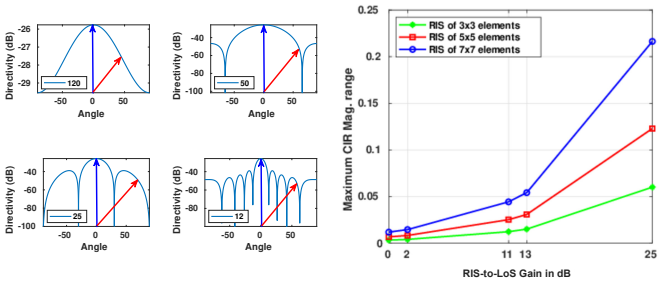
B. Directional Antennas (Ch5)

While a multi-transmitter system ensures fine-grained temporal separation of the signal paths, the use of omnidirectional antennas at the transmitters brings additional challenges to implement the desired FIR filter taps using RIS. Specifically, for omnidirectional transmissions, the received signal from a RIS roughly drops at least 10 dB. Moreover, in such transmissions there may exist a strong LoS component as well as reflections from multiple *uncontrolled* scatterers (other than our RIS), present in the environment. The combination of these two factors drastically limit the power contribution of the signal reflected from the RIS at the receiver, which in turn reduce the amplitudes of the FIR filter taps.

In order to study this problem, we formally express the delay profile in a setup with a single transmitter, single receiving antenna and N RIS as:

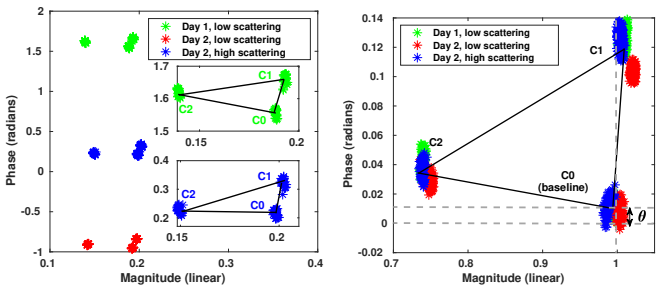
$$S(t) = (P_t - L_{LoS})\delta(t_{LoS}) + \sum_{i=1}^{N} (P_t - L_{RIS_i})\delta(t_{RIS_i}), \tag{8}$$

with P_t (dBm) the transmitted power. The terms L_{LoS} and L_{RIS_i} (dB) represent the losses for the LoS path and the i^{th} RIS path, $i = \{1, 2, ..., N\}$, respectively. Following the



(a) Directivities (b) CIR range

Fig. 3: (a) Higher antenna directivities tend to lead to higher RIS-to-LoS power ratios (difference between black and red arrows), which translates into (b) higher achievable CIR magnitude range, as the LoS component does not neglect the RIS contribution.



(a) (b)

Fig. 4: (a) Measured CIR for three RIS configurations (C0, C1, and C2) under varying scattering profiles. We observe a similar relative distance of the CIR magnitude and phase between different RIS configurations as the scattering profile changes. (b) AirNN adapts to varying scattering conditions using LS equalization with respect to *a prori* chosen baseline C0.

interpretation of a given RIS as an array of diffuse reflective antennas [37], and considering that each RIS is formed by M such antennas, we estimate L_{RIS_i} from:

$$L_{RIS_i} = 10log \mid \sum_{m=1}^{M} l_{RIS_i^m} e^{j\phi_i^m} \mid, \tag{9}$$

where $l_{RIS_i^m}$ represents the path loss associated with a particular reflective antenna m. This loss value depends on the carrier frequency, the distance between transmitter to RIS and RIS to receiver, RIS dimensions, transmitter and receiver antenna gains in the direction of each reflective antenna and the angle of incidence of the signal wavefront to the RIS plane. The term $e^{j\phi_i^m}$ in Eq.9 represents the phase of the incoming signal from element m of the RIS to the receiver, and thus, the received power is determined by the interference of the incoming signal from all $m = \{1, 2, ..., M\}$ reflective antennas of the RIS. The phase ϕ_i^m is given by $\phi_i^m = k(d_{(Tx,_i^m)} + d_{(_i^m,Rx)}) + \phi_{S_i^m}$ with $k = \frac{2\pi}{\lambda}$ as the wave number and λ as the wavelength. Lastly, $d_{(Tx,_i^m)}$, $d_{(_i^m,Rx)}$ represent the distances from transmitter to reflective antenna m and from that same antenna to receiver, respectively. The term $\phi_{S_i^m}$ gives the configurable phase shift introduced by reflective antenna m. Importantly, the estimation of $l_{RIS_i^m}$ follows a *product-distance* path loss model [37], where the power decays with the squared product between $d_{(Tx,_i^m)}$ and $d_{(_i^m,Rx)}$, a much sharper decay compared to the squared of $d_{(Tx,Rx)}$ of the LoS component. We estimate the term L_{LoS} in Eq.8 from the Friis equation and the delay terms $\delta(t_{LoS})$ and $\delta(t_{RIS_i})$ by dividing the known distances with c, the speed of light in vacuum. Thus, we estimate the CIR component associated to the RIS i path as:

$$h_i = \sqrt{l_{RIS_i}} \sum_{m=1}^{M} e^{j\phi_i^m}. \tag{10}$$

In Fig. 3b, we show the simulated maximum range for the magnitude of the received signal given in Eq.10 as a function of the RIS-to-LoS power ratio, for different transmitter antenna radiation patterns (Fig. 3a), as defined by their respective 3-dB beamwidth $BW_{3dB} = \{360°, 120°, 50°, 25°, 12°\}$ and different number of antenna elements with $M = \{49, 36, 25, 9, 4\}$. In Fig. 3a, the black arrow points to the RIS, while the red arrow points directly to the receiver, located at $45°$ from the RIS direction. We observe that for low antenna directivity, the high power of the LoS component compared to that of the signal reflected from the RIS renders any manipulation of the RIS ineffective. Hence, AirNN uses directional antenna elements at the transmitter that (i) boost the power of the reflected signals from the RIS paths and (ii) mitigate the degrading impact of the LoS signal along with the effect of additional ambient scatterers not controlled within AirNN.

C. Compensating for Channel (Ch6)

Due to the non-stationarity of wireless channels, the CIR engineered by certain RIS configurations may change over time, not resulting in the exact weights corresponding to the digital convolution, unless the CNN architecture is re-trained for every new scattering profile. Instead, AirNN uses a channel tracking and correction C0 to ensure that the weights of the CNN, as decided by the RIS configuration, remain valid even under new channel conditions caused by slow fading. This ensures that the received signal at the receiver always experiences a fixed and constant phase of zero degrees and unit magnitude when using the baseline configuration at every RIS.

We explain this process in Fig. 4, where we consider three different RIS configurations for illustration purpose, denoted by $C0$, $C1$, and $C2$. The process is as follows: using the AirNN setup, we send a known preamble sequence from a transmitter pointing to an RIS and collect samples of the received signal on two different days and scattering profiles. These profiles include cases of low impact (few meters away) and high impact (few cm away) scatterers, respectively. We then extract the preamble sequence at the receiver by cross-correlating the received samples with the preamble that is known at the receiver. From the received and known preambles, we estimate the channel for each RIS configuration, day and scattering profile using Least Squares (LS) channel estimation. Although we use a single RIS in Fig. 4, the same channel estimation approach is applicable to multiple transmitters pointing to different RIS by using unique preamble sequences at each transmitter. Since directional transmissions

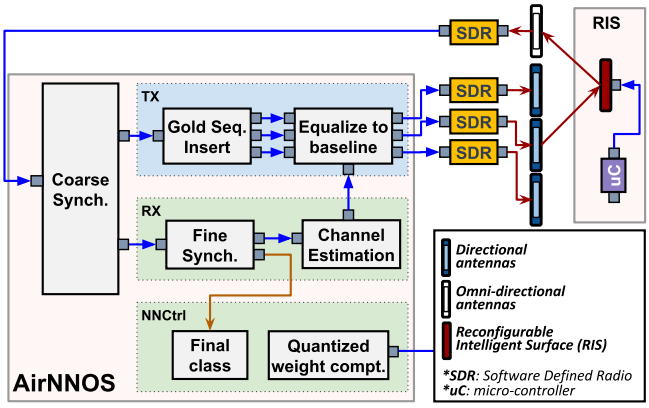


Fig. 5: AirNNOS software blocks within the modules of transmitter, receiver and `NNCtrl`.

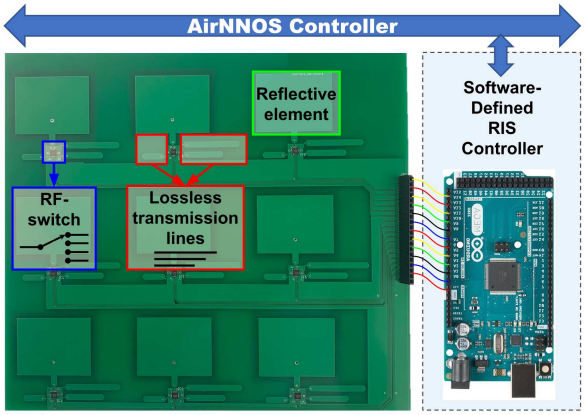


Fig. 6: Hardware prototype of a RIS with 9 patch reflective antennas whose signal reflections can be changed through software running in the controller by selecting transmission lines via the RF-switch.

mitigate the multipath effect, we denote $\hat{\mathbf{h}}_{C0} \in \mathbb{C}$ as the estimated narrowband channel at the receiver.

From our experimental results, we make two observations. First, we note that a change in the channel environment introduces variations within the over-the-air generated filter coefficients (Fig. 4a). Second, the relative distance between any pair of clusters of channel coefficients resulting from specific RIS configurations remains constant between deployments (Fig. 4a). In this example, AirNN takes configuration $C0$ as the baseline to generate the desired unit magnitude and no phase rotation, as shown in Fig. 4b. To accomplish this, the transmitter inverts the estimated channel vector as $\mathbf{p} = (\hat{\mathbf{h}}_{C0})^{-1} \in \mathbb{C}$ that is used to equalize the channel transformation for a other given RIS configuration. We capture such transformation by θ $(0 \leq \theta \leq \pi/2)$ computed as $\cos(\theta) = Re\{\mathbf{h}^H\mathbf{p}\}/(||\mathbf{h}||||\mathbf{p}||)$. We then apply the phase rotation to the CIR estimation of $C0$, $C1$, and $C2$. We note that this approach is applicable to any n_C arbitrary number of RIS configurations. The condition is that the data for the configurations of interest is collected along with the data for the baseline configuration $C0$ within the channel coherence time (T_c). For a large number of RIS configurations, this process is performed over multiple T_c in the initial offline C0, where in each T_c the CIR for the baseline configuration $C0$ and a different subset of the configurations of interest are estimated. During testing, only data for $C0$ and the configurations that generate the CIR that map the desired FIR taps need to be collected within T_c. Although we illustrate this process for a single RIS in Fig. 4, AirNN takes the same approach for every RIS.

VII. System Implementation

We highlight the main components of AirNN from a hardware viewpoint in Fig. 5. We describe our implementation using COTS Software Defined Radios (SDRs) in Sec. VII-A and the RIS hardware design and implementation in Sec. VII-B. We describe AirNNOS software that drives operations and controls the RIS units in Sec. VII-C.

A. Hardware Components

• **SDRs:** Our implementation is composed of four Ettus USRP X310 SDRs, each attached to a UBX 160MHz daughterboard, which can flexibly digitize up to 200 MSamples per second. Three SDRs serve in the transmitter interface, and a single SDR serves in the receiver interface. The SDRs are connected to the host machine via a 1 Gbps Ethernet link. We synchronize all SDRs in frequency and time through an Ettus OctoClock CDA-299. The octoclock helps correct the CFO of the multiple transmitters and the receiver.

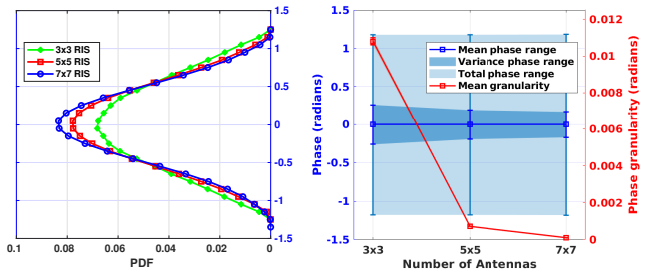
• **Antennas:** The receiver is attached to a VERT 2450 dual-band omnidirectional vertical antenna with 3 dBi gain. The transmitters have directional patch antennas with 18° of 3-dB beamwidth in azimuth and elevation and a grating lobe at 90° from their broadside direction. They operate in the 2.4GHz band.

B. RIS Hardware Design and Fabrication

The concept of loss-based transmission line for phase shifting has been implemented before [9], which we modify to realize AirNN. Our fabricated RIS is shown in Fig. 6.

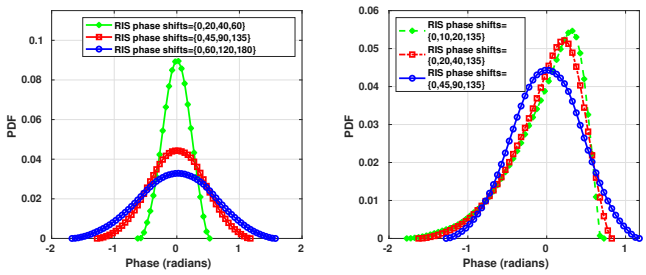
To access the feasibility of generating over-the-air FIR weights, we next study several RIS parameters, including (i) the type and number of patch reflective antennas within a RIS, the intra-RIS separation of the reflective antennas, and (ii) the phase shifts that these antennas may generate. We leverage the signal propagation model presented in Sec. VI-B to assess the impact of several RIS parameters on the feasible over-the-air generated FIR weights. We include simulations using a topology of a single RIS placed equidistantly from the transmitter and receiver antennas at 2.5 meters, while transmitting and receiving antennas are separated by 5 meters.

• **Reflective Elements: (type, size, distribution and number):** Each reflective element within our RIS is a switchable patch-type antenna of dimension $\lambda/2$ inserted between a two-layer PCB dielectric substrate and a full metal sheet at the bottom layer. We select a RIS of $M = 9$ reflective antennas, with a 3x3 layout in a 2D-plane. We space the antennas a distance of $\lambda/2$ to reduce the effect of mutual coupling between



(a) (b)

Fig. 7: CIR phase range and distribution w.r.t. RIS dimensions (i.e., number of reflective antennas). (a) A larger dimension does not result in a larger CIR phase range, but (b) results in a better granularity (less CIR weight quantization).



(a) Shift span (b) Shift distribution

Fig. 8: Study of the CIR phase range and distribution w.r.t. the RIS. (a) A larger shift span achieves a broader range of CIR phase values. (b) A uniform inter-shift spacing results in a uniform CIR phase distribution.

neighboring elements, as well as grating lobes in the RIS radiation pattern [38].

To assess how AirNN can benefit from having a larger number of reflective antennas, we configure our simulations with three different RIS sizes, i.e., 3x3, 5x5 and 7x7, and evaluate the obtained phase span, as well as the resulting phase granularity. Here, the term *span* refers to the difference between the maximum and the minimum induced phase shifts possible at the receiver, whereas the term *granularity* refers to the minimum phase difference between any two realizable phases at the receiver. Interestingly, having a larger RIS dimension does not lead to a larger span, as we show in Fig. 7a. Although a larger number of reflective elements achieve higher granularity, as seen in Fig. 7b, this improvement increases the size and cost of the RIS. For example, the improvement in granularity obtained by using a 7x7 instead of a 3x3 antenna RIS increases the manufacturing price by $200. These findings motivate us to select a small antenna set of $M = 9$ reflective antennas, with a 3x3 layout in a 2D-plane. These antennas are finally printed on a RIS PCB board with a FR-4 epoxy glass substrate of dimension 20cm $\times$ 20cm $\times$ 0.16cm.

• **Phase Shifts (angular range and inter-shift distances)**: We connect each of the nine reflective antennas to three lossless transmission lines of different lengths through a single RF switch. The resulting four phases per antenna (including no phase shift) enables $(4)^9$ configurations per RIS, generating a rich diversity of distinct signal reflections at the receiver. This design is finally printed on a RIS PCB board with a FR-4 epoxy glass substrate of dimension 20cm $\times$ 20cm $\times$ 0.16cm. A general purpose HMC7992 RF switch [39] connected to an Arduino Mega2560 μcontroller activates the selected line per antenna in real-time.

By selecting the length of our transmission lines, we can alter each reflective element impedance, which in turn changes their reflective coefficient and, consequently, introduces a phase shift to the signal reflected by that particular element. Our implementation allows four possible shifts per element, although the overall phase at the receiver is a combination of the individual shifts introduced by each of these nine reflective antennas.

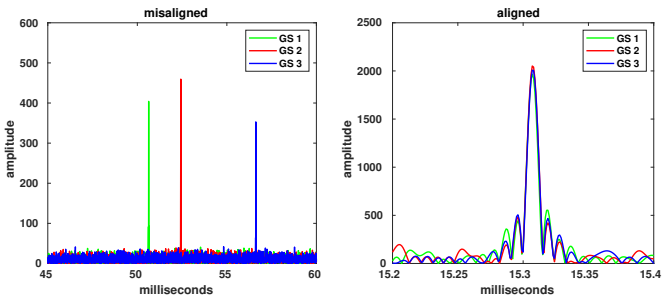
We determine next the *span* of possible phases, considering all possible combinations of the transmission line selections per reflective antenna. To do so, we simulate a single RIS unit of 3x3 dimensions with four transmission lines but with different upper bounds of the maximum transmission line induced shift. We show this analysis in Fig. 8a for such maximum shifts of $60°$, $135°$, and $180°$. We observe that the span at the receiver for the transmission line shift of $180°$ is over double that of a lower maximum value of $60°$. Next, we study how the inter-shift angular distance shapes the range of realizable phases at the receiver. We consider three combinations- with uniformly distributed phase shifts between transmission lines, narrowly spaced, and widely spaced shifts. From Fig. 8b we see that the PDF for uniform spacing follows a Gaussian phase distribution. Conversely, non-uniform phase spacing results in a Rician distribution. Thus, we design transmission lines to generate uniformly separated phase shifts to enable a maximum span at the receiver, i.e., $\{45°, 90°, 135°\}$.

C. AirNNOS Controller (Ch7)

We create AirNNOS, an orchestrating software framework that controls the following processes (see Fig. 5):

• **Transmission/reception sequence:** The receiver collects IQ samples and forwards them to the *Coarse synch* block that performs basic energy detection. At this stage, the *Coarse synch* module redirects the incoming IQ samples to the associated transmitters thread, which in turn processes the samples for AirNN operation and then re-transmits over the air. At this time, the *Coarse synch* switches its active output port to forward the incoming samples, i.e., resulting from the convolution, to the processing blocks within the receiver (see *AirNN* output). After fine-grained synchronization at the receiver, samples are fed to the `NNCtrl` to complete the CNN processing.

•**Pre-processing at the transmitter:** At the transmitter, we generate a set of orthogonal Gold sequences (GS) [40] used as preamble signals as they have desirable properties of good auto- and cross-correlation. We uniquely assigns one sequence to the set of IQ samples being sent over each transmit antenna. Thus, a transmission is composed of a GS appended to the received samples from the Tx. The benefit of using GS is two-fold: on the one hand, GS guarantees precise time synchronization for generating paths-delays that match the



(a) (b)

Fig. 9: Precise synchronization at the transmitter. (a) Misaligned inter-RIS path arrivals, and (b) precise synchronization in AirNN (see Sec. IV-C).

temporal distribution of desired FIR filters (see Sec. VI-A). On the other hand, GS offers a way to estimate and compensate for the channel variations over time (Sec. VI-C), as we explain next.

• **Synchronization and channel estimation at the receiver:** Although the SDRs used as transmitters start their transmissions at the same PPS instant, AirNN requires more fine-grained precision to realize the desired filter taps (Fig. 9a) when transmitting with high bitrate. To achieve this, the receiver computes the symbol misalignment between all the transmit streams via the GS by setting the last received stream as a reference. It then sends back this information along with the channel state information or CSI (*Channel estimation* computed at the receiver) to the transmitter. The SDRs used as transmitters delay their signals with additional zero-padding to sync with other peer-transmitters. We note that this padding is different from the padding used to generate one sample delayed versions of the same signals at each transmitter, shown in Fig. 2. Only with accurate time alignment (Fig. 9b) can AirNN generate the desired temporal displacement by deferring transmissions precisely by one sample with respect to other transmitters (see Sec.VI-A).

In Algorithm 1, we provide a pseudo-code that summarizes the tasks AirNN performs during the offline mapping stage as well as during the online stage that is run during testing, as described in Secs.VI and VII. .

VIII. Performance Evaluation

In this section, we first validate the equivalence between the analog convolution with over-the-air computation using our proposed AirNN system and its digital counterpart. Then, using as example the problem of digital modulation classification, we compare in simulation the classification accuracy achieved via different deep learning approaches. This comparison includes classical CNN trained using standard methods on a GPU (Classical-CNN), quantized CNN (QM-CNN) from Sec. V-B, and robust-quantized CNN (RQM-CNN from Sec. V-C).

A. Experimental Testbed

We use the hardware components as described in Sec.VII-A and fabricate three RIS units using the approach in Sec.VII-B.

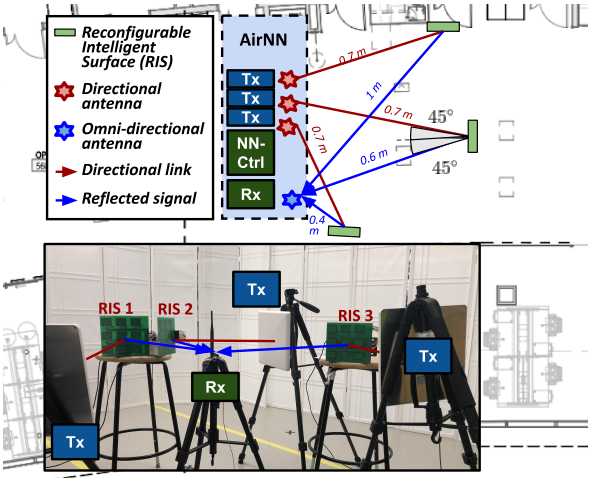


Fig. 10: Floor layout plan and experimental setup for AirNN testbed using our custom-built RIS.

Algorithm 1: Computation of Convolutions in AirNN

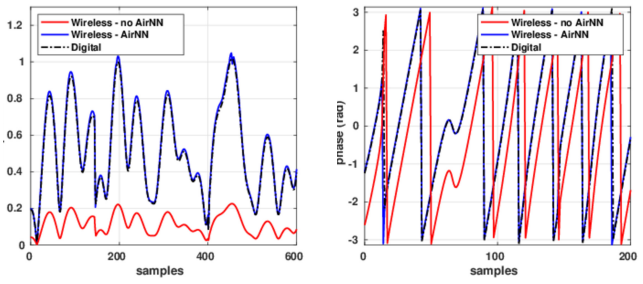
Input: RIS feasible configurations per Tx: $\{C_f\}_n = \{C0, C1, ..., Cn_C\}_n, \quad n = \{1, \cdots, N\}$.
Input: RIS desired configurations per Tx (FIR taps): $\{C_d\} = \{C_1, C_2, ..., C_N\}$.
Input: Input signal: x.
Output: Convolution output: y

1. **Stage** 1. Offline mapping (RIS configurations-FIR taps):
2. **for** $i = 0, \cdots, n_C$
3. Send GS_n from Tx_n with $\{C_i\}_n$ set at the RIS.
4. Collect y at the Rx and cross-correlate it with GS_n.
5. Estimate $h_{n,i}$ from known and extracted GS_n using Least-Squares.
6. **end**
7. **Return** mapping between $h_{n,i}$ and RIS configurations.
8. **Stage** 2. Online testing:
9. Get CIR for $\{C0\}_n, \{C_d\}$ within T_s, as above.
10. Estimate correction factor p_n per Tx from Sec.VI-C.
11. Generate zero-padded input signals: $\{x_1, x_2, ..., x_N\}$
12. Apply p_n to x_n, $\forall n, n = \{1, \cdots, N\}$
13. Trigger synchronized transmission of x_n from all Tx.
14. **Return:** Output of the convolution y collected at Rx.

The distance between the RIS and both transmitters and receiver is 0.7m. To achieve interference nulls at unintended RIS, given the transmitter antenna grating lobes, we orient each transmitter antenna towards a dedicated RIS with a 45° angle with respect to the plane of any other neighboring RIS (see Fig. 10). We use the frequency of 2.49 GHz and a sampling rate of 1 MS/s.

B. Validation of Convolutions in AirNN

We first demonstrate the capability of AirNN to generate a signal that matches the expected output of an all-digital FIR filter in a CNN. To this extent, AirNNOS first sends unique GS from each transmitter and estimates the CIR trough cross-correlation and LS estimation at the receiver for all RIS configurations, as a one-time initial step. We then train our QM-CNN model, constrained to the convolutional filter weights that our RIS can provide. At this stage, AirNNOS sends one sample



(a) Magnitude (b) Phase

Fig. 11: Comparison between over-the-air convolution with and without the use of AirNN RIS network. The former accurately realizes the desired convolutional filter with negligible error.

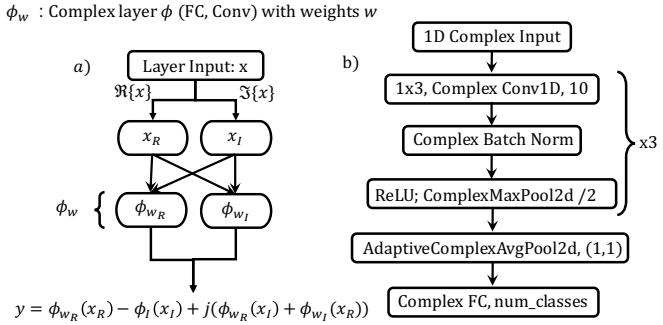
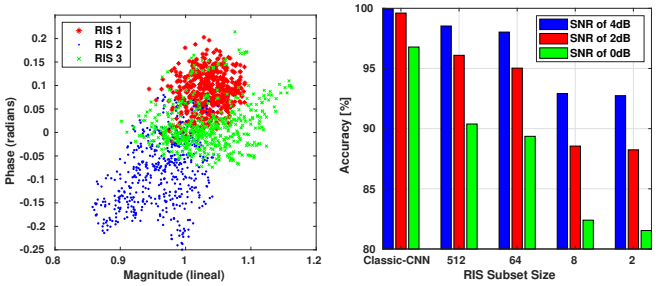


Fig. 12: Complex-valued layer (ϕ_w) diagram (a) and neural network architecture with complex weights used for modulation classification (b).



(a) (b)

Fig. 13: (a) Realizable FIR filter taps (magnitude and phase) using RIS. Each RIS provides a total of 512 phase-amplitude selections, any of which can be used as a filter tap; (b) Effect of quantization on the accuracy in modulation classification for different SNR values.

delayed versions of a BPSK digitally modulated signal from each transmitter, using directive antennas pointing to different RIS. Finally, after the signal has traversed the three RIS, we store the received signal at the receiver, whose magnitude and phase are shown in blue colour in Figs. 11a and 11b, respectively. We compare the similarity of the received signal to that of an all-digital convolution, shown in black colour, observing a Root Mean Square Error (RMSE) of 0.11 and 0.6 in magnitude and phase, respectively. When we store the received signal without controlling the RIS network (shown in red colour). The lack of temporal alignment and the RIS misconfiguration leads to a phase and magnitude mismatch with the all-digital convolution. This increases RMSE values to 0.46 and 2.58 in phase and magnitude, respectively.

C. AirNN for Modulation Classification

Next, we demonstrate that the convolution performed in AirNN is accurate enough to replace its digital equivalent for the real-world problem of modulation classification.

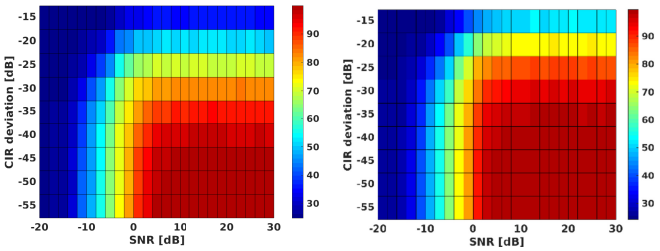
•**Dataset description:** We use the RADIOML 2018.01A dataset released in [41]. This includes signals collected from over-the-air transmissions modulated with 24 different schemes, i.e., from BPSK to 256QAM, under variable link qualities or SNR levels that range from -10 to 30dB. The data is organized in IQ sequences of 1024 I/Q samples, with 4096 sequences per modulation/SNR pair. Since this paper focuses on experimentally realizing convolutions with over-the-air computations (and not on improving on best-performing architecture for the problem of modulation classification), we consider a smaller subset of the problem with four of the most common classes of BPSK QPSK, 16QAM, and 32QAM. We split this reduced dataset into non-overlapping portions for training (60%), validation (20%), and testing (20%).

•**Architecture Description:** Our deep CNN model is composed of three sequential convolutional layers, i.e., a bundle of convolutional FIR filters followed by the pertinent batch normalization, activation (ReLu) and max pooling, an adaptive average pool layer and a single fully connected layer (Fig. 12b). We use PyTorch for implementation, with the number of filters as ten and learning rate $1e^{-4}$.

D. Impact of Quantization

Recall that we extract the set of feasible weights that our experimentally deployed RIS can realize, and use them to train our proposed QM-CNN (see Fig. 12). For a tractable analysis, we consider the lower-end of the shifting range, i.e., 0° and 45°, for each reflective element in all RIS. This gives us a total of $2^9 = 512$ different phase shifts for the reflected signal. We show the measured received CIR magnitude and phase at the receiver in Fig. 13a, where each point is an average of over ten transmissions. Multiple works have explored different bit-level quantization-aware training, such as 4-bit [42], [43], [44], [45], 2-bit (ternary) [46], [47], [48] or 1-bit (binary) [49], [50], [51] while preserving non-quantized network performance. This is also the approach we use as the starting point of this work.

To assess how quantization impacts accuracy, we generate smaller sets of candidate weights by randomly selecting subsets of size $\{2, 8, 64\}$ from the global set of 512. Here, 2 represents the most restrictive (or quantized) case, implying that the entire QM-CNN is constructed with two possible weights for each convolutional filter tap. Fig. 13b shows the average accuracy of QM-CNN when provided with various subset sizes for CIR weights and SNR values. Note that the SNR captures the wireless link quality of the input data and is provided by the dataset. Reducing the set of realizable weights impacts the QM-CNN accuracy, which falls more than an 8% for a quantization level below 64 and the lowest SNR evaluated of 0 dB. As the SNR increases up to 4 dB, the accuracy drops only a 2% for quantization levels above 8, becoming only



(a) (b)

Fig. 14: Accuracy in modulation classification using (a) quantized model (QM-CNN) and (b) our quantized robust approach (RQM-CNN).

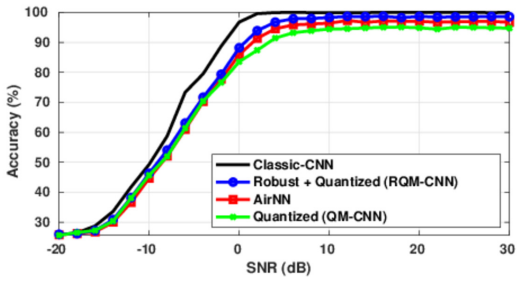


Fig. 15: Accuracy in modulation classification using all-digital convolution (Classic-CNN), quantized approaches QM-CNN and RQM-CNN, and our end-to-end AirNN system.

critical (7%) for a quantization as low as eight levels and below.

E. Robustness to Noise

We validate the robustness of the proposed RQM-CNN approach, which aims to mitigate deviations in RIS-engineered CIR weights due to noise. Fig. 4b gives a visualization of such a deviation illustrating the worst-case deviation measured empirically on the complete secondary path: transmitter to RIS to receiver, giving a CIR variance of -35 dB (σ^2 in Sec. V-C). Once we profile a range of possible CIR deviations, we train our RQM-CNN under an AWGN distribution within this CIR variance bound, following the steps described in Sec. V-C. We test the RQM-CNN performance for SNR levels between -20 and 30 dB on the primary link given as: transmitter to receiver, and CIR deviations between -55 and -15 dB over the secondary link transmitter to RIS to receiver. As opposed to this, simpler QM-CNN approach does not account for such over-the-air impairments during training. Results shown in Fig. 14 reveal that QM-CNN provides good performance for higher SNR and CIR deviations, but does not provide accuracy above 88% for SNR levels below 4dB and CIR levels above -35 dB (see Fig. 14a). The RQM-CNN approach achieves an accuracy of up to 96% in the same regimes (see Fig. 14b).

F. AirNN Performance

In this section, we compare the accuracy between the three all-digital CNN versions discussed so far: Classical-CNN and the quantized versions QM-CNN and RQM-CNN, as a function of the SNR level of input data. We also compare it with AirNN, using the experimentally derived convolution error on top of the RQM-CNN model. Here, QM-CNN and RQM-CNN are trained and tested as described in Sec. VIII-E. AirNN uses the same trained weights than RQM-CNN, but must operate within dynamic conditions that arise during testing. These modify the RIS-engineered FIR taps from the initial values acquired at the mapping stage (see Fig. 4b), which we earlier characterized as AWGN (see Sec. V-C).

We present the experimental results with AirNN in Fig. 15, where the CIR inaccuracies are selected from a Gaussian PDF (see Sec. V), ranging from -15dB to -50dB. In the figure, the Classical-CNN bounds the performance for any given SNR value. We observe a similar accuracy reported from all four models for very low SNR, i.e., between -20dB and -5dB, which is extremely challenging for the classification task. For higher SNR values, QM-CNN reports a lower maximum accuracy of 95%, while the robust training in RQM-CNN raises the accuracy up to 98%. AirNN closely follows the bound of the software-based RQM-CNN, with a drop in accuracy of only 2%, and an overall drop w.r.t. Classic-CNN of 3.2% for the SNR range of [6, 30] dB.

IX. Limitations and Future Opportunities

In this section, we identify limitations and open challenges of our approach and provide candidate solutions to speed up the practical deployment of AirNN.

- AirNN uses N transmitters to compute an over-the-air convolution. This specific implementation requires tight synchronization between transmitters and needs to scale in terms of cost and complexity with the FIR filter size *(N)*. This can be potentially addressed using antenna phased arrays with multiple RF chains that can construct many simultaneous beams as well as ubiquitous deployment of RIS in the ambient environment. Although our validation is limited in scale, there is potential for generalization in a more resource-rich network [52]
- The limited reflected power from our custom designed RIS constrain the separation distance between transmitters, receiver and RIS. Thus, they have to be carefully deployed to receive sufficient power. This situation can be mitigated with the use of larger RIS, improved RIS design with minimal losses and scenarios where higher power transmission become possible, e.g., in outdoor environments.
- Although AirNNOS is designed to tackle channel changes over time, it assumes that transmitters, receiver and RIS locations are fixed for both training and testing. Handling mobility requires interdisciplinary research in rapid wireless CIR estimation and lifelong-learning methods for RF pioneered by the ML community.
- AirNN only emulates the convolutional operation. It remains an open challenge to extend these ideas towards a complete CNN, all performed within the RF domain, which includes multi-layer convolutions and nonlinear activation functions.
- Performing a convolution with over-the-air computation involves higher power consumption and latency compared to its digital equivalent running on a GPU or FPGA. Our approach is best suited for scenarios where signals must be

transmitted over-the-air and AirNN merely provides an *add-on* functionality. Thus, supporting network infrastructure like multiple synchronized transmitters and RIS should not be solely present to realize AirNN.

Despite the above limitations, AirNN points towards an exciting computational domain involving RF signals. We will open-source design files for the RIS, code for AirNNOS and RIS simulation to equip the community with essential tools for future systems-focused work that can lead to full-fledged over-the-air CNNs.

X. Conclusions

We have demonstrated the feasibility of engineering convolution operations with over-the-air computation through a programmable RIS network that is precisely equivalent to its digital counterpart. We report an RMSE of 0.11 and 0.6 in magnitude and phase, respectively, in the output of the convolution between the analog and digital versions. Furthermore, we have shown how this operation, when included within the processing steps of a trained CNN is accurate enough to run inference on signal analysis tasks such as modulation classification. AirNN average testing accuracy is within 3.2% of the classical digital version under medium-to-high SNR conditions.

XI. Acknowledgement

The authors gratefully acknowledge the support from the NSF AI Institute for Future Edge Networks and Distributed Intelligence (AI-EDGE) (grant CNS-2112471), and the awards NSF CNS 1923789 and NSF CCF 1937500.

References

[1] A. Gadre, F. Yi, A. Rowe, B. Iannucci S. Kumar, "Quick (and Dirty) Aggregate Queries on Low-Power WANs", *19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 277-288, 2020.
T. W. Hughes, I. A. Williamson, M. Minkov, and S. Fan, "Wave Physics as an Analog Recurrent Neural Network," *Science advances*, vol. 5, no. 12, p. eaay6946, 2019.

[2] T. W. Hughes, I. A. Williamson, M. Minkov, and S. Fan, "Wave Physics as an Analog Recurrent Neural Network," *Science advances*, vol. 5, no. 12, p. eaay6946, 2019.

[3] X. Sui, Q. Wu, J. Liu, Q. Chen, and G. Gu, "A Review of Optical Neural Networks," *IEEE Access*, vol. 8, pp. 70773–70783, 2020.

[4] A. M. Elbir and K. V. Mishra, "A Survey of Deep Learning Architectures for Intelligent Reflecting Surfaces," *arXiv preprint arXiv:2009.02540*, 2020.

[5] D. Yu, S.-H. Park, O. Simeone, and S. S. Shitz, "Optimizing Over-the-Air Computation in IRS-Aided C-RAN Systems," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, IEEE, 2020.

[6] B. Zhao, S. Xiao, H. Lu, and J. Liu, "Waveforms classification based on convolutional neural networks," in *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 162–165, IEEE, 2017.

[7] J. Cai, C. Li, and H. Zhang, "Modulation Recognition of Radar Signal Based on an Improved CNN Model," in *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, pp. 293–297, IEEE, 2019.

[8] F. Restuccia, S. D'Oro, A. Al-Shawabka, M. Belgiovine, L. Angioloni, S. Ioannidis, K. Chowdhury, and T. Melodia, "DeepRadioID: Real-Time Channel-Resilient Optimization of Deep Learning-based Radio Fingerprinting Algorithms," in *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 51–60, 2019.

[9] M. Dunna, C. Zhang, D. Sievenpiper, and D. Bharadia, "ScatterMIMO: Enabling Virtual MIMO with Smart Surfaces," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pp. 1–14, 2020.

[10] B. E. Boser, E. Sackinger, J. Bromley, Y. Le Cun, and L. D. Jackel, "An analog neural network processor with programmable topology," *IEEE Journal of Solid-State Circuits*, vol. 26, no. 12, pp. 2017–2025, 1991.

[11] F. Zangeneh-Nejad, D. L. Sounas, A. Alù R. Fleury "Analogue computing with metamaterials", *Nature Reviews Materials*, vol. 6, no 3, p. 207-225, 2021.

[12] P. del Hougne G. Lerosey, "Leveraging Chaos for Wave-Based Analog Computation: Demonstration with Indoor Wireless Communication Signals", *Physical Review X*, vol. 8, no 4, p. 041037, 2018.

[13] J. Misra and I. Saha, "Artificial neural networks in hardware: A survey of two decades of progress," *Neurocomputing*, vol. 74, no. 1-3, pp. 239–255, 2010.

[14] J. Kendall, R. Pantone, K. Manickavasagam, Y. Bengio, and B. Scellier, "Training End-to-End Analog Neural Networks with Equilibrium Propagation," *arXiv preprint arXiv:2006.01981*, 2020.

[15] C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang, *et al.*, "Efficient and self-adaptive in-situ learning in multilayer memristor neural networks," *Nature communications*, vol. 9, no. 1, pp. 1–8, 2018.

[16] Zhang, H and Gu, M and Jiang, XD and Thompson, J and Cai, H and Paesani, S and Santagati, R and Laing, A and Zhang, Y and Yung, MH and others, "An optical neural chip for implementing complex-valued neural network," in *Nature Communications*, vol 12, pp. 1–11, 2021.

[17] A. P. James and L. O. Chua, "Analog Neural Computing with Super-resolution Memristor Crossbars," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2021.

[18] Y. Du, L. Du, X. Gu, J. Du, X. S. Wang, B. Hu, M. Jiang, X. Chen, S. S. Iyer, and M.-C. F. Chang, "An Analog Neural Network Computing Engine using CMOS-Compatible Charge-Trap-Transistor (CTT)," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 10, pp. 1811–1819, 2018.

[19] G. Cauwenberghs, "An Analog VLSI Recurrent Neural Network Learning a Continuous-Time Trajectory," *IEEE Transactions on Neural Networks*, vol. 7, no. 2, pp. 346–361, 1996.

[20] Y. Umuroglu, Y. Akhauri, N. J. Fraser, and M. Blott, "LogicNets: Co-Designed Neural Networks and Circuits for Extreme-Throughput Applications," in *2020 30th International Conference on Field-Programmable Logic and Applications (FPL)*, pp. 291–297, IEEE, 2020.

[21] D. Strukov, G. Indiveri, J. Grollier, and S. Fusi, "Building brain-inspired computing," *Nature Communications*, no. 10, pp. 4838–2019, 2019.

[22] D. Wen, G. Zhu, and K. Huang, "Reduced-Dimension Design of MIMO Over-the-Air Computing for Data Aggregation in Clustered IoT Networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2019.

[23] Z. Wang, Y. Shi, and Y. Zhou, "Wirelessly Powered Data Aggregation via Intelligent Reflecting Surface Assisted Over-the-Air Computation," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pp. 1–5, IEEE, 2020.

[24] V. Arun and H. Balakrishnan, "RFocus: Beamforming Using Thousands of Passive Antennas," in *17th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 20)*, pp. 1047–1061, 2020.

[25] H. Liu, X. Yuan, and Y.-J. A. Zhang, "Joint Communication-Learning Design for RIS-Assisted Federated Learning," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, IEEE, 2021.

[26] W. Ni, Y. Liu, and H. Tian, "Intelligent Reflecting Surfaces Enhanced Federated Learning," in *2020 IEEE Globecom Workshops (GC Wkshps*, pp. 1–6, IEEE, 2020.

[27] Z. Wang, J. Qiu, Y. Zhou, Y. Shi, L. Fu, W. Chen, and K. B. Letaief, "Federated Learning via Intelligent Reflecting Surface," *IEEE Transactions on Wireless Communications*, 2021.

[28] T. Jiang and Y. Shi, "Over-the-Air Computation via Intelligent Reflecting Surfaces," in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2019.

[29] W. Ni, Y. Liu, Z. Yang, and H. Tian, "Over-the-Air Federated Learning and Non-Orthogonal Multiple Access Unified by Reconfigurable Intelligent Surface," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, IEEE, 2021.

[30] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

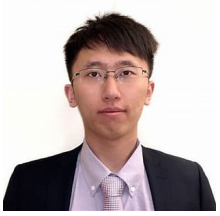
[31] X. Li, G. Zhang, H. H. Huang, Z. Wang, and W. Zheng, "Performance Analysis of GPU-Based Convolutional Neural Networks," in *2016 45th*

International conference on parallel processing (ICPP), pp. 67–76, IEEE, 2016.
[32] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep Complex Networks," in *International Conference on Learning Representations*, 2018.
[33] Y. Bengio, N. Léonard, and A. Courville, "Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation," *arXiv preprint arXiv:1308.3432*, 2013.
[34] P. Yin, J. Lyu, S. Zhang, S. Osher, Y. Qi, and J. Xin, "Understanding Straight-Through Estimator in Training Activation Quantized Neural Nets," in *International Conference on Learning Representations (ICLR)*, 2018.
[35] S.-E. Chang, Y. Li, M. Sun, R. Shi, H. K.-H. So, X. Qian, Y. Wang, and X. Lin, "Mix and Match: A Novel FPGA-Centric Deep Neural Network Quantization Framework," *arXiv preprint arXiv:2012.04240*, 2020.
[36] X. Zhang, *Gaussian Distribution*, pp. 425–428. Boston, MA: Springer US, 2010.
[37] Ö. Özdogan, E. Björnson, and E. G. Larsson, "Intelligent Reflecting Surfaces: Physics, Propagation, and Pathloss Modeling," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 581–585, 2019.
[38] C. A. Balanis, *Antenna Theory: Analysis and Design*. John wiley & sons, 2016.
[39] A. Devices, "HMC7992: Non-Reflective, Silicon SP4T Switch, 0.1 GHz to 6.0 GHz."
[40] Z. Xinyu, "Analysis of M-sequence and Gold-sequence in CDMA system," *IEEE International Conference on Communication Software and Networks (ICCSN)*, no. 1, pp. 466–468, 2011.
[41] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over the Air Deep Learning Based Radio Signal Classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.
[42] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "DoReFa-Net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients," *arXiv preprint arXiv:1606.06160*, 2016.
[43] J. Choi, Z. Wang, S. Venkataramani, P. I.-J. Chuang, V. Srinivasan, and K. Gopalakrishnan, "PACT: Parameterized Clipping Activation for Quantized Neural Networks," *arXiv preprint arXiv:1805.06085*, 2018.
[44] R. Gong, X. Liu, S. Jiang, T. Li, P. Hu, J. Lin, F. Yu, and J. Yan, "Differentiable Soft Quantization: Bridging Full-Precision and Low-Bit Neural Networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 4852–4861, 2019.
[45] G. Cheng, L. Ye, L. Tao, Z. Xiaofan, H. Cong, C. Deming, and C. Yao, "μL2Q: An Ultra-Low Loss Quantization Method for DNN Compression," *The 2019 International Joint Conference on Neural Networks (IJCNN)*, 2019.
[46] F. Li, B. Zhang, and B. Liu, "Ternary Weight Networks," *arXiv preprint arXiv:1605.04711*, 2016.
[47] C. Zhu, S. Han, H. Mao, and W. J. Dally, "Trained Ternary Quantization," in *International Conference on Learning Representations (ICLR)*, 2017.
[48] Z. He and D. Fan, "Simultaneously Optimizing Weight and Quantizer of Ternary Neural Network using Truncated Gaussian Approximation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11438–11446, 2019.
[49] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1," *arXiv preprint arXiv:1602.02830*, 2016.
[50] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks," in *European conference on computer vision (ECCV)*, pp. 525–542, Springer, 2016.
[51] X. Lin, C. Zhao, and W. Pan, "Towards Accurate Binary Convolutional Neural Network," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 345–353, 2017.
[52] Di Renzo, Marco and Debbah, Merouane and Phan-Huy, Dinh-Thuy and Zappone, Alessio and Alouini, Mohamed-Slim and Yuen, Chau and Sciancalepore, Vincenzo and Alexandropoulos, George C and Hoydis, Jakob and Gacanin, Haris and others, "Smart radio environments empowered by reconfigurable AI meta-surfaces: An idea whose time has come," in *EURASIP Journal on Wireless Communications and Networking*, pp. 1–20, 2019.

Sara Garcia Sanchez received the B.S. and M.S. degrees in Electrical Engineering from Universidad Politecnica de Madrid in 2016 and 2018, respectively, and the Ph.D. in Computer Engineering from Northeastern University, Boston, MA, in 2022. She currently holds a position as Research Scientist at the IBM Thomas J. Watson Research Center, NY. Her research interests include mmWave communications, reconfigurable intelligent surfaces and 5G standards.

Guillem Reus Muns received the B.Sc. degree in telecommunications engineering from the Polytechnic University of Catalonia (UPC-BarcelonaTech). He joined Northeastern University, USA, in 2018, where he got his M.Sc. in electrical and computer engineering and is currently working towards his Ph.D. His current interests include mobile communications, networked robotics, machine learning for wireless communications and spectrum access.

Carlos Bocanegra received the B.S. and M.S. degrees in Electrical Engineering from Polytechnic University of Catalonia in 2015 and Northeastern University in 2017, respectively, and Ph.D. in Computer Engineering from Northeastern University, in 2021. Currently, he holds a position of Senior Wireless System Engineer under Delart Tech working at Facebook Reality Labs where he devises simulation frameworks to study coexistence, latency, and power consumption for 5G NR and IoT. He has also conducted research and product development in companies such as NEC Laboratories America, Princeton, NJ; and MathWorks, Natick, MA. His research interests include the design and prototype of multi-antenna systems, coexistence in HetNets, machine learning for wireless applications, and virtualization at the RAN using Software Defined Radios.



Yanyu Li is a Ph.D. candidate at the Department of Electrical and Computer Engineering in Northeastern University, advised by Professor Yanzhi Wang. His research interests include deep learning, neural network architecture search, pruning and quantization.

Ufuk Muncuk received the Ph.D. degree in electrical and computer engineering from Northeastern University, Boston, MA, USA, in 2019. He is a Research Assistant Professor with the Electrical and Computer Engineering Department, Northeastern University. His research interests include design, optimization, and implementation for RF energy harvesting circuits and system design for RF energy and magnetic coupling-based energy transfer, intra-body transceivers, and cognitive radio systems. Dr. Muncuk was recipient of the Best Paper Runners-Up at ACM SenSyS in 2018, and Best Paper Awards at IEEE ICC in 2013 and IEEE GLOBECOM in 2019.

Yousof Naderi is a Research Assistant Professor in the Electrical and Computer Engineering Department at Northeastern University, Boston, MA. He received the Ph.D. degree in Electrical and Computer Engineering from Northeastern University, Boston in 2015. He was the recipient of NEU Ph.D. dissertation award in 2015, a finalist in the Bell Labs Prize competition in 2017, Best Paper Awards at the IEEE INFOCOM in 2018 and IEEE GLOBECOM in 2019. His research expertise lies in the design and development of AI-powered cyber-physical systems, intelligent surfaces for 6G and beyond, and self-powered networked robotics.

Yanzhi Wang is currently an Assistant Professor at the Department of ECE at Northeastern University, Boston, MA. His research focuses on model compression and platform-specific acceleration of deep learning architectures, maintaining the highest model compression rates on representative DNNs since 09/2018. His work on AQFP superconducting based DNN acceleration is by far the highest energy efficiency among all hardware devices. His recent research achievement, CoCoPIE, can achieve real-time performance on almost all deep learning applications using off-the-shelf mobile devices, outperforming competing frameworks by up to 180X acceleration. He received the U.S. Army Young Investigator Program Award (YIP), Massachusetts Acorn Innovation Award, Ming Hsieh Scholar Award, and other research awards from Google, MathWorks. etc.

Stratis Ioannidis is an Associate Professor in the Electrical and Computer Engineering Department of Northeastern University, in Boston, MA, where he also holds a courtesy appointment with the Khoury College of Computer Sciences. Prior to joining Northeastern, he was a research scientist at the Technicolor research centers in Paris, France, and Palo Alto, CA, as well as at Yahoo Labs in Sunnyvale, CA. His research interests span machine learning, distributed systems, networking, optimization, and privacy.

Kaushik Roy Chowdhury is a Professor at Northeastern University, Boston, MA. He is presently a co-director of the Platforms for Advanced Wireless Research (PAWR) project office. His current research interests involve systems aspects of networked robotics, machine learning for agile spectrum sensing/access, wireless energy transfer, and large-scale experimental deployment of emerging wireless technologies.