

SIGNETS: Neural Network Architectures for m-QAM Soft Demodulation

ARAVIND VOGGU ¹, KANISH R ¹, Nishith Akula ¹, Lohitaksh Maruvada ¹, Takanori Shimizu ²,
and Madhav Rao ¹ (Senior Member, IEEE)

¹International Institute of Information Technology Bangalore, KA 560100, India

²Sony India Software Centre, Bengaluru, KA 560103, India

Corresponding author: Madhav Rao (e-mail: mr@iiitb.ac.in)

ABSTRACT This paper presents a novel approach to Quadrature Amplitude Modulation (QAM) demodulation using neural networks, addressing the limitations of traditional demodulation techniques in complex channel conditions. Through systematic Neural Architecture Search and Hyper-parameter optimization, we develop a family of Convolutional Neural Network architectures that demonstrate robust performance across challenging channel conditions, including multi-path fading, inter-symbol interference, and non-linear distortions, without requiring explicit channel estimation. We comparatively evaluate the members of the family of networks to find a Pareto-optimal Neural Network Demodulator with a balance of demodulation accuracy and computational cost, achieving an average accuracy of 99.658% across 4 dB to 24 dB SNR while requiring less than 16,000 Floating-Point Operations (FLOPs) for every demodulated QAM-16 symbol. A practical Field Programmable Gate Array (FPGA) implementation that achieves 2.52 Million bits per second throughput while maintaining 99.55% demodulation accuracy through structured pruning and quantization-aware training is presented. Experimental validation over acoustic channels demonstrates superior performance compared to traditional techniques, with performance further enhanced through fine-tuning.

INDEX TERMS CNN, Communication, FPGA, LLR, Machine Learning, Multipath Components, NAS, Neural Networks, QAM, Rayleigh Fading Channel, Soft Demodulation

I. INTRODUCTION

Modern digital communication systems rely critically on sophisticated signal demodulation techniques that can extract information accurately from complex transmission environments. Quadrature Amplitude Modulation (QAM) has emerged as a fundamental modulation scheme enabling high spectral efficiency in contemporary communication infrastructures. Over the years, numerous demodulation techniques based on stringent and complex Mathematical models have been developed for demodulating QAM signals. Among these, the log-MAP algorithm [1], introduced in 1994, has been shown to be statistically optimal, albeit impractical from an implementation standpoint because of the cost of computation needed. Subsequent research has focused on developing approximations to this algorithm to reduce its computational complexity, making it practical for deployment.

While the log-MAP algorithm has been foundational, and its sub-optimal but computationally efficient approximations remain a cornerstone of modern QAM demodulator designs, their performance degrades under dynamic channel condi-

tions, non-linear distortions, and in the presence of multipath interference. As the demand for faster and more reliable data transfer escalates—driven by the proliferation of high-speed networks, expectations of ubiquitous connectivity, and emerging modes of communication [2], [3], the limitations of conventional demodulation techniques in complex real-world scenarios have become increasingly apparent.

The contemporary technological ecosystem presents unique opportunities for re-imagining signal processing techniques. The advances in Machine Learning, particularly Neural Network (NN) architectures, training techniques, and the development of specialized hardware for accelerating compute for Neural Networks, have catalyzed a paradigm shift in solving historically intractable engineering challenges [4]–[8]. Notably, Neural Networks have also received recognition in the domain of Computer Networking and Digital communication, employed in applications such as Network Intrusion Detection [9], Modulation Recognition [10], [11], Error Coding [12], and more.

Neural Networks offer a compelling alternative by leveraging data-driven learning to implicitly model channel distortions and non-linearities. Notably, NN-based demodulators circumvent the need for explicit channel estimation and iterative decoding, thereby reducing algorithmic complexity. For instance, recent work [13] demonstrates that neural demodulators achieve comparable Bit Error Rate (BER) performance to log-MAP with lower power consumption. The inherent parallelism of Neural Networks also enables relatively straightforward demodulator throughput scaling.

Despite these advantages and promising initial research, existing Neural Network based demodulators have yet to achieve widespread deployment. Most existing research remains confined to idealized simulations, with evaluations often restricted to Additive White Gaussian Noise (AWGN) channels or simplistic fading models and limited real-world evaluation. Real-world deployments are further hindered by reliance on power-intensive GPUs, TPUs, and similar expensive application specialized hardware, making large-scale deployment impractical and cost-prohibitive. This gap between theoretical promise and practical viability motivates our work.

In this paper, we address these challenges by developing a family of Neural Network Architectures and training techniques that exhibit excellent performance even in complex channel conditions. We study the family of networks to evaluate the aspects of their architecture that affect their viability and accuracy for QAM demodulation. Additionally, we present the results of implementing one of the Neural Networks on a Field Programmable Gate Array (FPGA) device, including latency, resource utilization, and throughput.

The remainder of this paper is structured as follows: Section II reviews related work and identifies key limitations in existing NN-based approaches to demodulation. Section III formalizes the QAM demodulation problem and operational constraints. Section IV details the data generation methodology and simulation environment for NN Demodulator training and evaluation. Section V presents the Neural Network Architecture Search (NAS) process and Hyper-parameter optimization for demodulator NNs. Section VI evaluates the selected architecture across diverse channel conditions and input data configurations. Section VII discusses FPGA resource utilization and ASIC synthesis outcomes. Section IX concludes with future research directions, including the potential for online learning in adaptive demodulation.

II. PREVIOUS WORK

Traditional QAM demodulation techniques are primarily based on classical signal processing algorithms, such as phase-locked loops, matched filters, and coherent detection mechanisms. These methods are well-suited for high signal-to-noise ratio (SNR) environments and rely on precise synchronization and accurate channel estimation to achieve optimal performance. Maximum Likelihood Estimation (MLE) and the Log-Map algorithm have emerged as statistically optimal approaches for soft-demodulation in terms of bit error

rate. However, these techniques encounter significant challenges when applied to higher-order QAM under complex channel conditions, such as severe fading or interference, due to their computational complexity and dependency on idealized assumptions.

The advent of Neural Networks has prompted researchers to explore their potential as alternatives to traditional communication techniques. Early efforts utilized feed-forward and Convolutional Neural Networks (CNNs) to perform tasks such as modulation classification [11] and symbol detection. Prior research in neural network-based demodulation has explored various approaches, though with certain limitations in scope and practical implementation. This section provides an extensive review of previous Neural Network based approaches to demodulation, and FPGA implementations of NN based solutions. A summary of key works is presented in Table 1, which highlights the methodologies, modulation schemes, channel models, and limitations of prior research.

Early attempts at Neural Network based demodulators were directed at demodulating simpler modulation schemes. An Amplitude Modulation (AM) demodulator was described in [14] and showcases its FPGA implementation. Their use of NN eliminated the intermediate frequency conversion stage and allowed for use at wider bandwidths. However, they use 64-bit Floating Point arithmetic, which restricts deployment to a limited set of accelerators while reducing the throughput of the demodulator. Deep Convolutional Neural Networks [26] were used to demodulate Frequency Shift Keying modulation over a Rayleigh Fading Channel in [15]. An Elman Neural Network [27] is used for Binary Frequency Shift Keying demodulation over AWGN channels in [16], and for Amplitude Shift Keying (ASK), Phase Shift Keying (PSK), and Frequency Shift Keying (FSK) in [17]. Variational Auto Encoders (VAEs) [28] are used for interference cancellation and signal detection in [18]. A Neural Network is pre-trained and then fine-tuned for demodulation of Binary Phase Shift Keying (BPSK) modulation over short-range multi-path channels in [19]. Nearly optimal demodulation of Golden Angle Modulation (GAM) over AWGN channels using Neural Networks was demonstrated in [20]. The network is shown to have achieved this at lower computational complexity than a traditional Maximum Likelihood (ML) demodulator. A Deep Neural Network that uses Transfer Learning [29]–[31] to learn channel conditions from Pilot Symbols for BPSK demodulation is presented in [21]. A 1-Dimensional Recurrent CNN designed to demodulate BPSK, and QPSK is presented in [22], and is evaluated over an AWGN channel simulation. Notably, this paper uses a Parametric Leaky ReLU [32], [33] activation function to mitigate vanishing gradients when training the network. A Mixed Neural Network composed of CNN and Recurrent Neural Network (RNN) is proposed for ASK, FSK, and QAM-16 demodulation over AWGN and Rayleigh Fading channels in [23]. Soft-Decisions from a QAM-4 Zero-forcing equalizer are converted to Hard-Decisions using a Neural Network in [24]. A standards compliant real-time Neural Network receiver for 5G-NR is de-

Table 1. Summary of existing Neural Network-Based Approaches to Demodulation

Reference	Modulation	Channel Model	NN Architecture	Hardware	Key Contributions / Notes
KV et al. [14]	AM	AWGN	Feed-forward	XC7VX690T Virtex-7 FPGA	Broader bandwidth operation through elimination of the Intermediate Frequency stage. Implementation utilizes 64-bit floating point arithmetic, which constrains both deployment capabilities and maximum throughput.
Mohammad et al. [15]	FSK	Rayleigh	Deep CNN	NVIDIA GT 840m GPU	The computational expense of deep network architectures limits practical application. Authors report a theoretical throughput of 123.33 kbps.
Li et al. [16]	BFSK	AWGN	Elman NN	–	Recurrent Architecture is used to demodulate a BFSK signal in an AWGN simulation with a baud rate of 400 bits per second. Authors claim strong anti jamming capabilities.
Amini et al. [17]	ASK, PSK, FSK	AWGN	Probabilistic NN	–	Authors evaluate NNs for multiple modulations and claim computational efficiency advantages over conventional feed-forward and Elman neural networks, though quantitative throughput comparisons are absent from the analysis.
Wai et al. [18]	BFSK, 4FSK	Experimental Setup	VAE	2.2 GHz Intel Xeon CPU	An experimental setup involving a signal generator and a signal transceiver was used to simulate real-world like interference. Deep Variational Auto Encoders were used in a novel way for interference cancellation. The authors report a 2.17 kbps throughput for 4FSK.
Fang et al. [19]	BPSK	Short Range Multi-path	Twice Trained Network	–	Authors propose a novel transfer learning methodology intended to reduce computational complexity. However, no throughput or computational complexity measurements were provided. Evaluation conducted via multi-path channel simulation models real-world propagation conditions better than other works.
He et al. [20]	GAM	AWGN	Neural Network	–	Demonstrates near-optimal bit-error performance for Golden Angle Modulation while achieving reduced computational complexity compared to Maximum Likelihood demodulation approaches.
Ahmad et al. [21]	BPSK	AWGN, Rayleigh, Experimental setup	Deep NN	USRP SDR	Neural network architecture applied to both modulation and demodulation processes. Parametric Leaky ReLU activation investigated as a solution to gradient vanishing phenomena. The architectural complexity suggests limited throughput capability.
Zhao et al. [22]	BPSK, QPSK	AWGN	1D CNN	NVIDIA GTX 2080TI GPU and eight Intel i7-7700 CPUs	NN employed for both modulation and demodulation. Parametric Leaky ReLU evaluated in addressing vanishing gradients. Size of the proposed network implies comparatively low throughput.
Wu et al. [23]	BFSK, QPSK, and 16QAM	AWGN, Rayleigh	CNN, RNN, Hybrid	NVIDIA GTX 1080 GPU	Comparative analysis of multiple neural network architectures, with the most efficient implementation achieving 160 kbps throughput, while the highest accuracy network operates at 67 kbps for BFSK. Performance metrics for alternative modulation schemes are not provided.
Polvani et al. [24]	4QAM	M-MIMO Rayleigh	Feed-forward	–	Implements demodulation through transformation of Zero Forcing equalizer soft-estimates to binary values. Demonstrates reduced computational requirements relative to a traditional method.
Wiesmayr et al. [25]	QPSK, 16QAM, 64QAM	3GPP TLD-B, 3GPP TDL-C	Var-MCS NRX with Var-IO layers	NVIDIA A100 GPU	Presents a real-time 5G-NR receiver achieving sub-millisecond inference latency on A100 GPU hardware with multi-modulation support. Implementation viability is limited by the substantial computational requirements of the architecture
Shental et al. [13]	Multiple QAM	CDL	Feed-forward	–	Demodulation by NN is modeled as an approximation to log-MAP and is shown to perform better than other approximations. Levenberg-Marquardt optimization algorithm was employed for faster convergence.

scribed in [25] and is shown to have an inference time shorter than a millisecond when tested on an NVIDIA A100 GPU. A feed-forward Neural Network is used for soft-demodulation of multiple QAM and is evaluated over a Clustered Delay Line (CDL) channel model in [13]. The paper suggests that the Levenberg-Marquardt back-propagation algorithm [34] shortens the training time for Neural Network demodulators with low parameter count.

While previous research has highlighted the potential of Neural Networks in addressing non-linearities and impairments in communication systems, significant limitations persist. First, most studies evaluate their models under simplistic channel assumptions, such as AWGN or Rayleigh fading, without considering more realistic multi-path fading or interference scenarios. Second, the focus is often restricted to simple modulation schemes like BPSK or FSK, with limited applicability to higher-order modulations such as QAM. A significant gap in the existing literature is the limited investigation into the relationship between Neural Network architectures and demodulation performance. Most studies present single architectural solutions without comparative analysis of different network configurations or systematic evaluation of architectural choices. Furthermore, while various works demonstrate promising results in simulation, there is a notable lack of hardware implementation studies, particularly for more complex modulation schemes and channels.

III. PROBLEM INTRODUCTION

We aim to address the problem of demodulating QAM (Quadrature Amplitude Modulation) symbols using Neural Networks in this work. QAM is a widely used modulation scheme in modern communication systems, particularly in applications that require high-speed data transmission, including WiFi, Digital Video Broadcasting (DVB), and 4G LTE [35]–[37]. Our focus is to replace traditional demodulators with a Neural Network based approach to directly map received signals to their original transmitted data bits. In this section, we formalize the communication model and the scope of this work while clarifying the assumptions and limitations.

Consider an M -QAM modulation scheme where $M = 2^k$ and $k \in \mathbb{N} - \{1\}$, denoting the number of bits per symbol. Each symbol $s \in \mathcal{C} \subset \mathbb{C}$ corresponds to a unique k -bit vector $\mathbf{c} \in \{0, 1\}^k$. The constellation $\mathcal{C}$ consists of M symbols that are spaced in a rectangular grid in the complex plane. For a rectangular M -QAM, the In-phase (I) and Quadrature (Q) components are independently spaced at equal intervals. Formally:

$$\mathcal{C} = \{(2p + 1 - \sqrt{M}) + j(2q + 1 - \sqrt{M}) \mid p, q \in \{0, 1, \dots, \sqrt{M} - 1\}\}.$$

A mapping function $\mu : \{0, 1\}^k \rightarrow \mathcal{C}$ uniquely maps the input k -bit vectors to symbols in the constellation $\mathcal{C}$. This mapping can be arbitrary. Let $\mathbf{b}_k = \{b_1, b_2, \dots, b_k\} \in \{0, 1\}^k$ represent a vector of k bits to be transmitted, where

these bits are assumed to be independent and identically distributed (i.i.d.) with equal probability of occurrence. As μ uniquely maps $\mathbf{b}_k$ to symbols in $\mathcal{C}$, the symbols too have an independent and identical distribution, and are equally likely to be transmitted. This assumption can be reasonably accommodated when generating training and validation data, for small values of M ($M < 10$) typical in contemporary communication systems.

Let $\mathbf{x}_n$ be a vector containing n samples obtained after mapping an arbitrary length input bit vector $\mathbf{b}_n$ to symbols in $\mathcal{C}$. If these symbols are transmitted through a channel with one Line-of-Sight (LoS) path and N Non-Line-of-Sight (NLoS) paths between the transmitter and the receiver, then the vector of samples observed at the receiver, $\mathbf{y}_n$, is expressed as stated in (1), assuming the channel's spectral and temporal characteristics remain constant through the transmission.

$$y[n] = \sum_{i=0}^N g_i e^{j\phi_i} x[n - d_i] + w[n] \quad (1)$$

where:

ϕ_i : Phase delay of the i -th path

d_i : Delay of the i -th path

g_i : Gain of the i -th path

$i : 0, 1, \dots, N$ for the LoS and N NLoS paths

$w[n] : \mathcal{CN}(0, \sigma^2)$, Additive White Complex Gaussian Noise

The demodulators task is to map $\mathbf{y}_n$ to soft estimates $\hat{\mathbf{b}}_n \in \mathbb{R}^k$, where each entry corresponds to the LLR for the associated bit:

$$\hat{b}_i = \log \frac{\Pr(b_i = 0 \mid \mathbf{y}_n)}{\Pr(b_i = 1 \mid \mathbf{y}_n)}, \quad i = 1, \dots, n.$$

A complete contemporary communication system would include several intermediate steps between modulation and demodulation. For instance, an IEEE 802.11ac WiFi transmission would involve Orthogonal Frequency Division Multiplexing (OFDM) and up-conversion to carrier frequency. Similarly, a WiFi receiver performs carrier acquisition, synchronization, channel estimation and equalization from pilot symbols, before the samples are presented to a QAM demodulator. From the perspective of the modulator and the demodulator, these intermediate steps compose a composite discrete channel. Similar to traditional demodulators, the proposed Neural Network demodulator is designed without assumptions about this composite channel.

While conventional approaches rely on explicit equalization and AWGN-based Log-Likelihood Ratios (LLR) approximations (e.g., max-log-MAP), our NN demodulator learns to infer LLRs directly from distorted observations ($\mathbf{y}_n$) through data-driven training. Neural Networks have been shown to be suitable for demodulation of complex, non-rectangular constellations [20], however, we limit the scope of this work and evaluation to equally spaced rectangular constellations.

IV. DATA GENERATION

The data for training the demodulator is generated in accordance with the assumptions outlined in Section III. The process begins with the generation of bits from a uniform pseudo-random source. These bits are subsequently mapped into QAM symbols. To model real-world transmission characteristics and practices better, we implement symbol shaping through Root Raised Cosine (RRC) filtering. Each symbol s undergoes RRC filtering with a roll-off factor of $\beta = 0.5$ and is sampled at $N_s = 4$ samples per symbol prior to transmission.

The composite transmission channel is modeled as discrete multi-path channel with AWGN. Specifically, we use a Clustered Delay Line (CDL) channel model with one Line-of-Sight (LoS) path, three Non-Line-of-Sight (NLoS) clusters, and added AWGN. The average delay and gains of these paths are periodically perturbed during data generation to ensure the robustness and generalisability of our demodulator. The channel is assumed to be frequency flat, and both the transmitter and receiver are considered stationary.

At the receiver, matched RRC filtering is applied to the received signal and the NN demodulator is trained on these filtered samples. An overview of this process is provided in Fig. 1. To ensure the robustness of the demodulator, the training and evaluation datasets incorporate varying SNR levels ranging from 4 to 24 dB, along with randomized channel realizations.

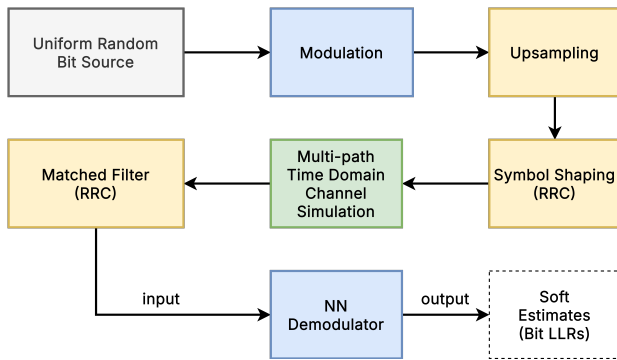


Figure 1. Overview of data generation for NN demodulator training.

V. SEARCH FOR A SUITABLE NEURAL NETWORK ARCHITECTURE

Designing an efficient neural network architecture is crucial for achieving both high accuracy and computational efficiency. Convolutional Neural Networks (CNNs) are particularly well-suited for signal demodulation due to their inherent ability to detect local patterns through convolution operations, which aligns with our requirement to identify symbol patterns and detect interference between adjacent symbols. Since symbol interference is predominantly local (distant symbols have minimal interference), we opted for CNNs over more complex architectures designed for long-range pattern recognition such as RNNs, LSTMs, or Transformers. While prior works

have explored various hand-designed model configurations, we employed hyper-parameter optimization to systematically determine an optimal CNN architecture suited for demodulation. Optuna [38], a state-of-the-art hyper-parameter optimization framework, was used to explore the search-space efficiently. We formulate the optimization as a minimization problem over the validation loss and the number of Floating Point Operations (FLOPs) needed to run the model. The optimization was performed on a cluster of six RTX 4090 GPUs.

The architecture consists of 1D convolutional layers followed by fully connected layers as shown in Fig. 2. The search included the following hyper-parameters:

- Number of Convolution layers, and the number of filters in each Convolution layer: (CL_1, CL_2, CL_3) ,
- Convolution Kernel Size,
- Number of Neurons in each Dense layer: (DL_1, DL_2) , and Dropout rate,
- Batch Size for training, and
- Learning Rate and its decay rate.

Optuna's Bayesian Optimization framework enables systematic exploration of neural architecture hyper-parameters through an adaptive, multi-objective search strategy. By probabilistically modeling the sensitivity of each hyper-parameter to the optimization objectives, the algorithm prioritizes resource allocation towards high-impact dimensions of the search space. We augment this approach with an early stopping mechanism that prunes sub-optimal trials early during the training phase, thereby further prioritizing resources towards more promising trials. The relative importance of hyper-parameters as estimated by Optuna when using the Adam Optimizer are presented in Fig. 3.

The multi-objective optimization landscape is visualized in Fig. 4, which presents a scatter plot of BER versus FLOP count across 250 architectural trials. Through the empirical determination of the Pareto front, we identify non-dominated architectures that optimally balance computational complexity and error-correction capability. Notably, FLOP count exhibits a non-linear relationship with BER below 10^{-3} , necessitating a pragmatic trade-off between inference throughput and accuracy for deployment in latency-sensitive communication systems. Modern communication systems virtually always use a Forward Error Correction (FEC) mechanism, and we contextualize these results by analyzing the operational thresholds of modern FEC schemes. As demonstrated in Fig. 5, FEC codes (LDPC, Turbo, and convolutional) achieve an asymptotic BER=0 below a certain BER in the un-coded stream. Crucially, when the pre-FEC BER reaches 10^{-3} , all considered FEC implementations successfully decode to error-free outputs. This establishes the 10^{-3} BER as a *golden target*, which balances accuracy and throughput, beyond which only marginal gains of accuracy at the expense of significant loss of throughput are possible.

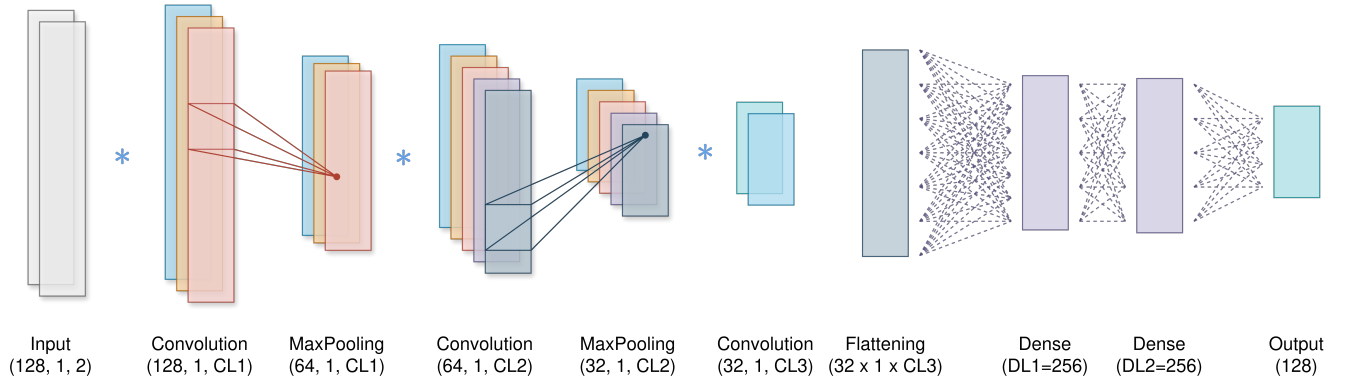


Figure 2. Architecture of CNN Demodulator with Tensor dimensions.

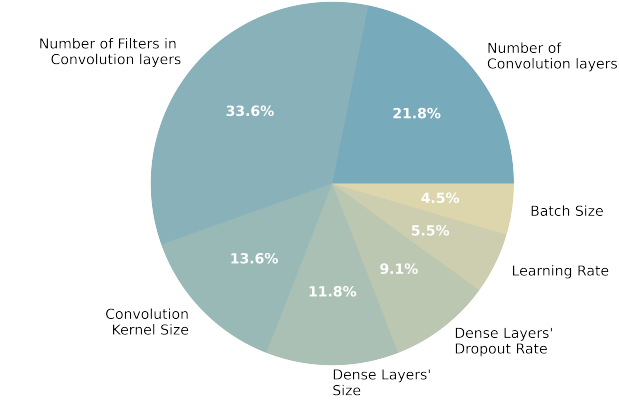


Figure 3. Relative sensitivity of optimization objectives to considered Hyper-parameters.

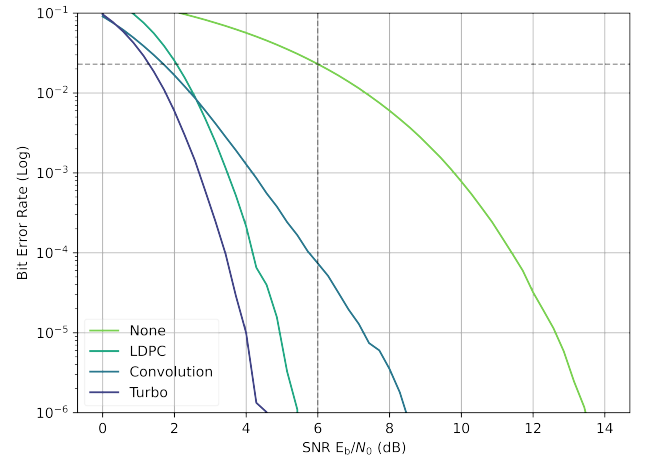


Figure 5. BER versus SNR graph for various Forward Error Correction codes. The simulation used BPSK modulation in an AWGN channel with a code-rate of 0.5 for all codes.

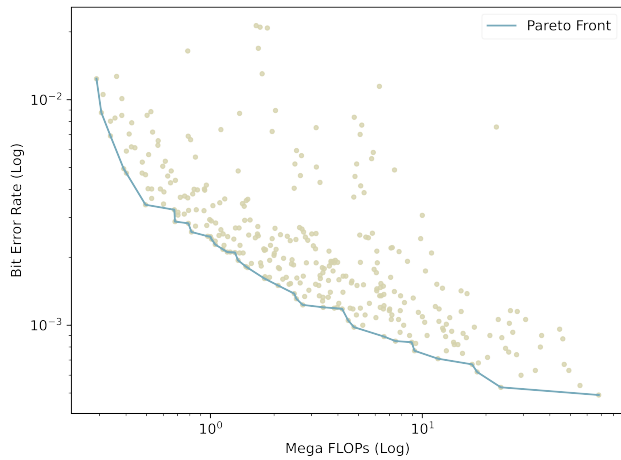


Figure 4. BER versus FLOPs for all models in the search-space. BER shown is the average across 4 dB to 24 dB SNR. Lower is better for both axes.

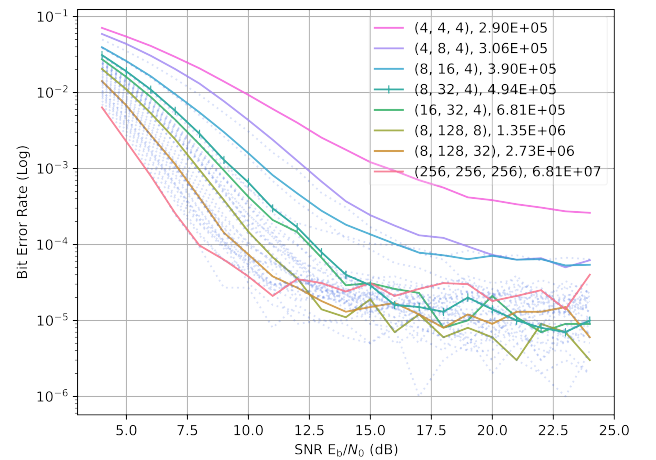


Figure 6. BER versus SNR plot for every model on the Pareto-front. A few distinct models are highlighted and labeled while the rest are shown as pale blue dots. The numbers inside parenthesis represent model configurations (CL_1 , CL_2 , CL_3) whereas the number next to it is the FLOP count of the model (Lower is better).

VI. RESULTS

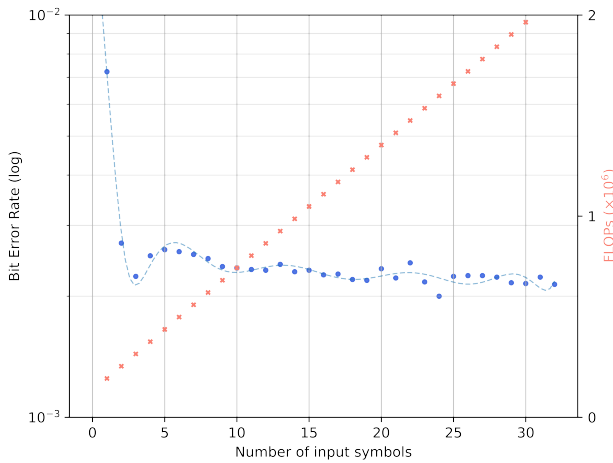


Figure 7. BER of a model with increasing number of input symbols.

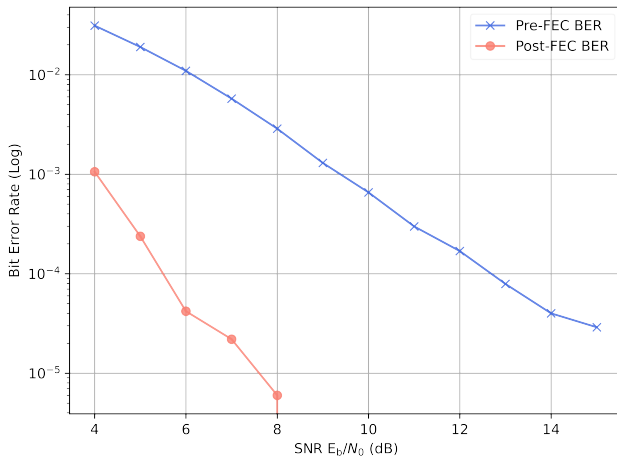


Figure 8. BER vs SNR graph of the selected model.

After 250 trials, we identified a Pareto-optimal inference-efficient model with a 99.658% average demodulation accuracy across 4 dB to 24 dB SNR. The BER versus SNR graphs for all the models on the Pareto-front are shown in the Fig. 6. The optimal architecture derived from the Network Architecture search was then tested in a channel simulation as described in Section III with data generated as shown in Section IV. We have evaluated the model with a varying number of inputs. The average demodulation accuracy of the model across 4 dB to 24 dB SNR as the number of input symbols changes is shown in the Fig. 7, along with the FLOP count of the model. A significant reduction in BER was observed when the model processed multiple input symbols simultaneously. This improvement is attributed to the model's enhanced capacity for inter-symbol interference (ISI) mitigation when provided with extended temporal context, enabling learned compensation of channel memory effects through joint symbol analysis. We also observe that the FLOP

count of the model increases linearly with the number of input symbols.

The observed performance-complexity tradeoffs suggest that multi-symbol processing architectures achieve substantial BER improvements without incurring prohibitive computational overhead. A BER versus SNR graph of the evaluated model is presented in the Fig. 8.

VII. HARDWARE IMPLEMENTATION

The practical adoption of neural network-based demodulation systems in real-world applications requires the development of optimized hardware implementations capable of meeting stringent real-time processing requirements. The fundamental obstacle lies in the substantial amount of Floating-Point Operations (FLOPs) required for Neural Network inference. Recent advancements in optimizing methodologies such as Pruning, Weight Sharing, and Quantization Aware Training (QAT) have demonstrated that substantial computation and power savings reductions are achieved without significant accuracy degradation [39]–[41]. Building upon these techniques, we employ layer-wise structured pruning for filters with a pruning schedule that starts with a 25% pruning ratio and decreases by 0.25% every 10 epochs. With this methodology, we increased the sparsity of the model during training and identified the filters in the layers that have completely become sparse. We constructed a new, smaller model, eliminating all the filters in the layers that have entirely become sparse, and created a smaller network with no reduction in accuracy.

Layer-Wise, Symmetric-Quantization with Brevitas framework was employed to realize the NN design in hardware. Optimal bit-width allocation for the layers was determined through statistical analysis of the distribution of weights and activation outputs across network layers, which was used to convert the Floating Point operations in the model to Integer operations. This approach yielded an optimal quantized model, with 4-bit Integer quantized weights and an 8-bit Integer quantized Activation function (W4A8) configuration, achieving 99.57% classification accuracy on our demodulation task, representing only a 0.09% accuracy reduction compared to the full-precision baseline. The optimized network was subsequently exported in qONNX format for hardware deployment, ensuring FPGA-agnostic compatibility. We employ the FINN framework for FPGA implementation. The FINN framework is specifically designed for deploying Quantized Neural Networks on FPGA devices and taking advantage of the heterogeneous resources on FPGAs such as DSP and BRAM via hardware-aware compilation. We evaluated multiple FINN configurations on the Xilinx ZCU102 FPGA evaluation platform with the clock period setting of 20 ns, analyzing throughput-resource utilization tradeoffs through parametric benchmarking and summarizing the results in Table 2. To obtain the maximum frequency of operation and throughput in hardware, we increased the frequency of the clock until a timing violation occurred, which resulted in a functional violation. We obtained an optimized

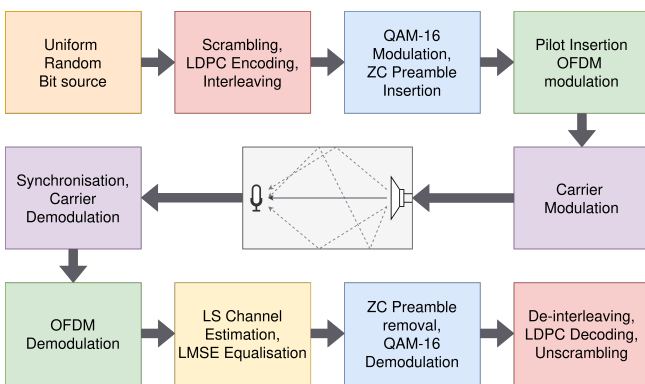
Table 2. Hardware utilization report from FINN framework. The maximum available quantity of each resource type shown in parentheses in the column headers.

FINN Target FPS	CLB LUTs (274080)	CLB Registers (548160)	CARRY8 (34260)	F7 Muxes (137040)	F8 Muxes (68520)	CLB (34260)	LUT Logic (274080)	LUT Memory (144000)	Block RAM (912)	DSPs (2520)	Frequency (MHz)	Throughput (Mbps)
850	28745	26580	2202	602	264	6360	26746	1999	151	5	166	0.5175
1500	28847	27797	2208	603	195	6464	26736	2111	148.5	6	175	0.646
2000	28808	26808	2218	549	142	6395	26713	2095	140.5	8	175	1.028
3500	29040	27134	2216	622	120	6534	26985	2145	141	15	150	1.83
6000	29124	27506	2210	762	296	6552	26741	2383	142.5	19	166	2.52
Resource Utilization Percentage (6000 FPS)	10.63%	5.02%	6.45%	0.56%	0.43%	19.12%	9.76%	1.65%	15.62%	0.75%		

implementation for the FPS setting of 6000, which achieves a sustained throughput of 2.52 Million Bits per second while maintaining 99.55% demodulation accuracy.

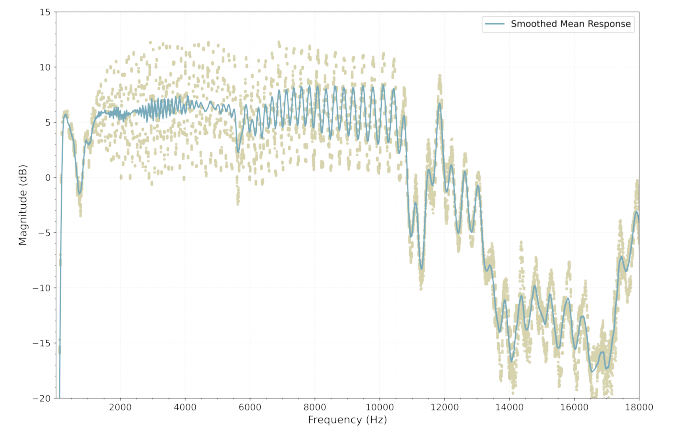
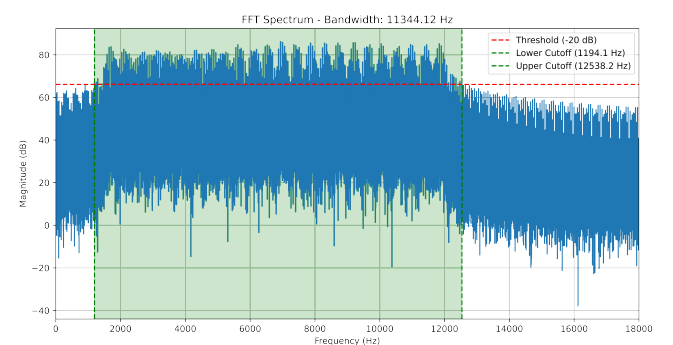
VIII. EXPERIMENTAL VALIDATION

To validate the real-world applicability of our proposed neural demodulator, we designed a controlled experiment emulating a multipath propagation environment. Given the constraints of a laboratory setup, we employed an audio-based transmission system, leveraging the slower propagation speed of sound compared to electromagnetic waves to achieve meaningful multipath effects within a limited physical space. Our experimental testbed implemented a complete OFDM communication chain using acoustic transmission between a speaker and microphone pair, as illustrated in Figure 9. The system operates with a 10 kHz bandwidth centered at 7 kHz carrier frequency, with all signals sampled at 44.1 kHz. The frequency response of the speaker-microphone setup is illustrated in Figure 10, while the spectral characteristics of the generated OFDM signal are shown in Figure 11.

**Figure 9.** System architecture for audio-based OFDM transmission and reception.

The transmitter chain incorporates several standard digital communication processing blocks. Input data undergoes scrambling to eliminate long sequences of identical bits, followed by Low-Density Parity-Check (LDPC) Forward Error Correction (FEC) encoding [42]. The encoded bits are then interleaved to mitigate the impact of burst errors on decoding

performance. This processed data stream is mapped using 16-QAM modulation, with a Zadoff-Chu (ZC) [43] sequence inserted as a preamble for frame synchronization and Kronecker structure pilot symbols embedded for channel estimation. The resulting symbol stream undergoes OFDM modulation before being modulated onto the final carrier and emitted via a speaker. The transmitted signal is captured by a microphone sampling at 44.1 kHz.

**Figure 10.** Measured frequency response characteristics of the speaker-microphone acoustic channel.**Figure 11.** Spectral characteristics of the transmitted OFDM signal.

We implemented two distinct receiver architectures to enable performance comparison. For the baseline demodula-

tion approach, the received signal is processed through a conventional pipeline. Synchronization is achieved through correlation with the known ZC sequence. The signal is then downconverted to baseband and OFDM-demodulated to extract the subcarriers. Channel estimation is performed using the Least Squares (LS) method with linear interpolation between pilot symbols, followed by Linear Minimum Mean Squared Error (LMMSE) equalization [44]. The equalized symbols are demodulated using conventional 16-QAM soft-demodulation. Finally, the data undergoes de-interleaving, LDPC decoding, and descrambling to recover the transmitted bits. In contrast, our neural network demodulator replaces the entire channel estimation, interpolation, equalization, preamble removal, and QAM demodulation with a single Neural Network that directly produces demodulated soft symbols.

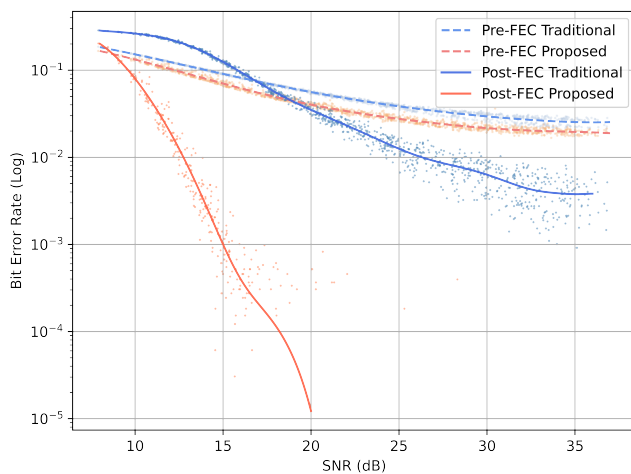


Figure 12. SNR versus BER performance comparison between baseline traditional equalization technique and the proposed ML demodulator.

Figure 12 presents the BER versus SNR performance comparison between the aforementioned baseline traditional pilot-aided equalization-based demodulation and our proposed ML demodulator. The neural network architecture employed in these experiments corresponds to the optimized design detailed in previous sections of this paper. Although pre-FEC bit error rates show similar performance between the baseline and proposed approaches (with a slight consistent advantage for the proposed method), post-FEC performance demonstrates substantial improvements for the proposed model-based demodulation method. This enhancement is likely stems from the superior quality of the LLRs generated by the proposed NN demodulator, which provide more reliable soft information for the LDPC decoder.

It should be noted that the SNR calculations presented in Figure 12 may contain inherent limitations in accuracy. The SNR was computed based on the ratio of transmitted signal power to ambient noise power before and after transmission, which does not account for all potential noise and distortion sources. For instance, nonlinear effects such as hysteresis in the speaker and microphone membranes introduce additional signal distortion. Furthermore, quantization noise from

the analog-to-digital converter (ADC) and digital-to-analog converter (DAC), along with other unmodeled noise contributions, may lead to an overestimation of the SNR. Consequently, the true SNR may be lower than the values indicated in the figure. These considerations should be taken into account when interpreting the absolute SNR values, though the relative performance comparisons between methods remain valid.

We would also like to note that the reported performance metrics for the proposed neural network-based demodulator are achieved under "blind decoding" conditions, wherein the demodulating model operates without access to estimated SNR or knowledge of the original pilot symbol values, both of which are readily available to the baseline demodulator. While pilot symbol samples are incorporated into the input fed to the machine learning model, the corresponding reference pilot values are randomly generated and remain unknown to the network during training and inference. These pilot symbols serve exclusively as additional contextual information to enable the model to mitigate multi-path interference effects through learned feature extraction.

The proposed neural demodulator's performance can be enhanced through fine-tuning with minimal channel-specific data before inference. Fine-tuning involves adapting the pre-trained model parameters using a small dataset collected from the current channel, typically requiring only a few hundred training samples. This process updates the network weights through standard backpropagation, allowing the model to better characterize current channel conditions such as changing multi-path effects, ambient noise levels, and frequency response variations. Figure 13 shows four example images transmitted through our acoustic system, comparing demodulation results from: the baseline traditional demodulator, the proposed demodulator without adaptation, and the proposed demodulator with channel-specific fine-tuning. The proposed method with fine-tuning exhibits the best performance overall with lower BER and higher Structural Similarity Index Measure (SSIM) [45].

IX. CONCLUSION

This work demonstrates the viability of neural network-based approaches for QAM demodulation in practical communication systems. Through systematic architecture exploration and optimization, we have shown that carefully designed convolutional neural networks can achieve performance comparable to traditional demodulation techniques while offering several key advantages. The ability to process multiple symbols simultaneously enables effective handling of inter-symbol interference, while the learned compensation for channel memory effects eliminates the need for explicit channel estimation. Our hardware implementation results are particularly promising, showing that through structured pruning and quantization-aware training, neural network demodulators can achieve high throughput on FPGA platforms without significant accuracy degradation. The achieved throughput of 2.52 Million bits per second while maintaining 99.55% accu-

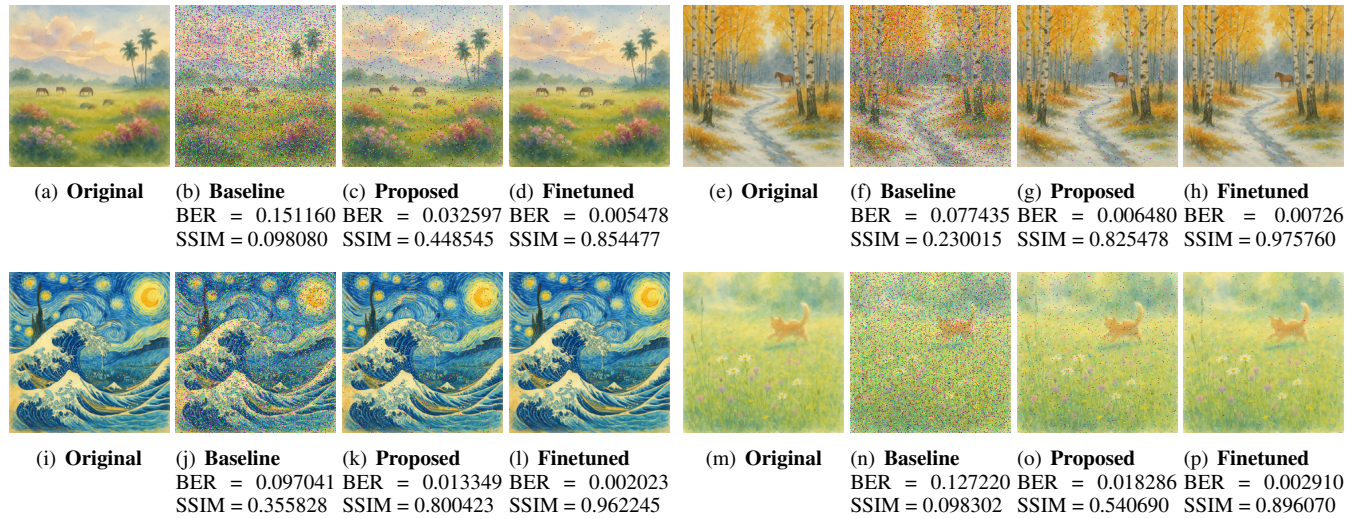


Figure 13. Comparison of image transmission quality through acoustic channel demodulation. From left to right: original images, results from baseline traditional demodulator, proposed Neural Network demodulator, and proposed Neural Network demodulator with fine-tuning. The fine-tuned model demonstrates superior Bit Error Rate and SSIM. The SNR was continuously varied throughout the experiment.

racy demonstrates the practical feasibility of deploying these systems in real-world applications. The experimental validation conducted under realistic channel conditions characterized by time-varying, frequency-selective fading with multipath interference demonstrates the robustness and adaptability of the proposed Neural Network demodulation technique.

Several directions for future research emerge from this work. First, the potential for online learning and adaptation of the neural network weights could enable dynamic adjustment to changing channel conditions, potentially improving performance in mobile scenarios. Second, the exploration of more sophisticated quantization schemes and hardware-aware neural architecture search could further optimize the implementation efficiency. Finally, extending this approach to higher-order QAM constellations and more complex channel models would broaden its applicability in next-generation communication systems. In conclusion, our results suggest that neural network-based demodulation represents a promising direction for future communication systems, particularly as hardware accelerators become more prevalent and the demand for robust performance in challenging channel conditions continues to grow. The demonstrated balance between computational efficiency and demodulation accuracy, coupled with successful hardware implementation, positions this approach as a viable alternative to traditional demodulation techniques in practical applications.

All the source code, datasets, and documentation related to this paper is made available for research use at: <https://github.com/zeroby0/signets>.

ACKNOWLEDGMENT

The images labeled "original" in Fig. 13 were generated via the image generation feature of OpenAI's ChatGPT.

References

- [1] J. Erfanian, S. Pasupathy, and G. Gulak. Reduced complexity symbol detectors with parallel structure for ISI channels. *IEEE Transactions*

- on Communications, 42(2/3/4):1661–1671, February 1994. Conference Name: IEEE Transactions on Communications.
- [2] Ruth G. Gebremedhin and Thomas L. Marzetta. Thermal Conduction as a Wireless Communication Channel. In *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pages 1085–1090, 2022.
- [3] Junsu Jang and Fadel Adib. Underwater backscatter networking. In *Proceedings of the ACM Special Interest Group on Data Communication, SIGCOMM '19*, pages 187–199, New York, NY, USA, 2019. Association for Computing Machinery. event-place: Beijing, China.
- [4] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Židek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021. Publisher: Nature Publishing Group.
- [5] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016.
- [6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20*, pages 1877–1901, Red Hook, NY, USA, December 2020. Curran Associates Inc.
- [7] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation, February 2021. arXiv:2102.12092 [cs].
- [8] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.
- [9] Fatima Laghrissi, Samira Douzi, Khadija Douzi, and Badr Hssina. Intrusion detection systems using long short-term memory (LSTM). *Journal of Big Data*, 8(1):65, May 2021.

- [10] Nathan E. West and Timothy J. O'Shea. Deep Architectures for Modulation Recognition, March 2017. arXiv:1703.09197 [cs].
- [11] Peng He, Yang Zhang, Xinyue Yang, Xiao Xiao, Haolin Wang, and Rongsheng Zhang. Deep Learning-Based Modulation Recognition for Low Signal-to-Noise Ratio Environments. *Electronics*, 11(23), 2022.
- [12] Hyeji Kim, Yihan Jiang, Ranvir Rana, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. Communication Algorithms via Deep Learning, May 2018. arXiv:1805.09317 [stat].
- [13] Ori Shental and Jakob Hoydis. "Machine LLRning": Learning to Softly Demodulate. In *2019 IEEE Globecom Workshops (GC Wkshps)*, pages 1–7, December 2019.
- [14] Vineetha KV, Chinthala Ramesh, and Dhanesh G Kurup. Implementation of direct demodulator based on ANN using FPGA. *Alexandria Engineering Journal*, 108:730–753, December 2024.
- [15] Ahmad Saeed Mohammad, Narsi Reddy, Fathima James, and Cory Beard. Demodulation of faded wireless signals using deep convolutional neural networks. In *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 969–975, January 2018.
- [16] Min Li, HongSheng Zhong, and Min Li. Neural Network Demodulator for Frequency Shift Keying. In *2008 International Conference on Computer Science and Software Engineering*, volume 4, pages 843–846, December 2008.
- [17] M Amini and E Balarastaghi. Universal Neural Network Demodulator for Software Defined Radio. *International Journal of Machine Learning and Computing*, 1(3):305–310, 2011.
- [18] Ian Wong Chee Wai, Mohamed Hisham Jaward, Vishnu Monn Baskaran, Shuan-Ni Liang, Colin Chee Chong Hin, and Moh Lim Sim. Joint Interference Cancellation and Signal Detection Using Latent Space Representations in VAE. *IEEE Transactions on Consumer Electronics*, 70(1):197–208, February 2024. Conference Name: IEEE Transactions on Consumer Electronics.
- [19] Lanting Fang and Lenan Wu. Deep learning detection method for signal demodulation in short range multipath channel. In *2017 IEEE 2nd International Conference on Opto-Electronic Information Processing (ICOIP)*, pages 16–20, July 2017.
- [20] Boxiang He, Zitao Wu, and Fanggang Wang. Rethinking: Deep-learning-based Demodulation and Decoding, June 2022. arXiv:2206.06025 [cs].
- [21] Arhum Ahmad, Satyam Agarwal, Sam Darshi, and Sumit Chakravarty. DeepDeMod: BPSK Demodulation Using Deep Learning Over Software-Defined Radio. *IEEE Access*, 10:115833–115848, 2022. Conference Name: IEEE Access.
- [22] Ruobing Zhao, Jiao Wang, and Jianqing Li. An End-to-End Demodulation System Based on Convolutional Neural Networks. *Journal of Physics: Conference Series*, 2026(1):012006, September 2021.
- [23] Tian Wu. CNN and RNN-based Deep Learning Methods for Digital Signal Demodulation. In *Proceedings of the 2019 International Conference on Image, Video and Signal Processing, IVSP '19*, pages 122–127, New York, NY, USA, February 2019. Association for Computing Machinery.
- [24] Gabriel Polvani, Victor Croisfelt, and Taufik Abrão. Massive MIMO Demodulation Aided by NN. In *2021 IEEE URUCON*, pages 166–171, November 2021.
- [25] Reinhard Wiesmayr, Sebastian Cammerer, Fayçal Aït Aoudia, Jakob Hoydis, Jakub Zakrzewski, and Alexander Keller. Design of a Standard-Compliant Real-Time Neural Receiver for 5G NR, September 2024. arXiv:2409.02912 [cs].
- [26] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. Conference Name: Proceedings of the IEEE.
- [27] Jeffrey L. Elman. Finding Structure in Time. *Cognitive Science*, 14(2):179–211, 1990. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog1402_1.
- [28] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes, May 2014. arXiv:1312.6114 [stat] version: 10.
- [29] Mingsheng Long, Jianmin Wang, Jianguang Sun, and Philip S. Yu. Domain Invariant Transfer Kernel Learning. *IEEE Transactions on Knowledge and Data Engineering*, 27(6):1519–1532, June 2015. Conference Name: IEEE Transactions on Knowledge and Data Engineering.
- [30] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):9, May 2016.
- [31] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, 109(1):43–76, January 2021. Conference Name: Proceedings of the IEEE.
- [32] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML '10*, pages 807–814, Madison, WI, USA, June 2010. Omnipress.
- [33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, December 2015. ISSN: 2380-7504.
- [34] M.T. Hagan and M.B. Menhaj. Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993, November 1994. Conference Name: IEEE Transactions on Neural Networks.
- [35] Michelle X. Gong, Brian Hart, and Shiwen Mao. Advanced Wireless LAN Technologies: IEEE 802.11AC and Beyond. *GetMobile: Mobile Comp. and Comm.*, 18(4):48–52, January 2015.
- [36] Digital Video Broadcasting (DVB) Project. Digital video broadcasting (DVB); specification for the broadcast of digital television for reception by fixed installations (DVB-t). Technical report, European Telecommunications Standards Institute (ETSI), 1999.
- [37] Stefania Sesia, Issam Toufik, and Matthew Baker. LTE - the UMTS long term evolution: From theory to practice. Wiley, 2011. ISBN: 978-0470660340 Place: Chichester, UK Publisher: John Wiley & Sons.
- [38] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19*, pages 2623–2631, New York, NY, USA, July 2019. Association for Computing Machinery.
- [39] YingBo Fan, Wei Pang, and ShengLi Lu. HFPQ: deep neural network compression by hardware-friendly pruning-quantization. *Applied Intelligence*, 51(10):7016–7028, October 2021.
- [40] Etienne Dupuis, David Novo, Ian O'Connor, and Alberto Bosio. On the Automatic Exploration of Weight Sharing for Deep Neural Network Compression. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1319–1322, March 2020. ISSN: 1558-1101.
- [41] Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461(C):370–403, October 2021.
- [42] R. Gallager. Low-density parity-check codes. *IRE Transactions on Information Theory*, 8(1):21–28, January 1962.
- [43] D. Chu. Polyphase codes with good periodic correlation properties (Corresp.). *IEEE Transactions on Information Theory*, 18(4):531–532, July 1972.
- [44] Mehmet Kemal Ozdemir and Huseyin Arslan. Channel estimation for wireless ofdm systems. *IEEE Communications Surveys & Tutorials*, 9(2):18–48, 2007.
- [45] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.



ARAVIND R. VOGGU received his Bachelor's and Master's degrees in Electronics and Computer Engineering from the International Institute of Information Technology Bangalore, where he currently serves as a researcher. His work spans multiple disciplines with a focus on developing technologies to combat climate change, particularly through advanced seafloor mapping and analysis. His research portfolio includes the development of prehensile robotic arms, efficient image

compression and transmission techniques for low-power devices operating in challenging wireless environments, and innovative neural network approaches to demodulation in complex communication channels. Through this interdisciplinary approach, he contributes to the creation of technological solutions that enhance our understanding of climate systems and support environmental conservation efforts.

Email: aravind.reddy@iiitb.org



KANISHH R received a Bachelor of Engineering degree in Electronics and Communication from Rajalakshmi Engineering College, Chennai, India, in 2023. He is currently pursuing an M.S. by Research degree in VLSI at the International Institute of Information Technology Bangalore, Bengaluru, India. His research interests include parallel computing systems and hardware-software co-optimization methodologies for neural network deployment on edge devices and FPGAs. He is also

one of the recipients of the Micron University Research Alliance (URAM) scholarship for the year 2023.

Email: kanishh.r@iiitb.ac.in



NISHITH AKULA is currently pursuing the Integrated M.Tech degree in Electronics and Communication Engineering at the International Institute of Information Technology Bangalore, Karnataka, India. His research interests include FPGA-based deep learning accelerators, efficient neural network deployment on hardware, high-performance computing, and hardware-efficient algorithm design. He is currently working on an ASIC framework for neural network deployment, focusing on

optimizing computational efficiency and resource utilization while ensuring minimal latency and power consumption. His interests extend to hardware-software co-design methodologies, exploring innovative techniques to enhance neural network performance on FPGAs and ASICs.

Email: a.nishith@iiitb.ac.in



LOHITAKSH MARUVADA is currently pursuing his Integrated Master's degree in electronics and communication engineering at the International Institute of Information Technology, Bangalore, Karnataka, India

His research interest includes neural architectures, parallel computing, and hardware-aware architecture design. He is currently working on hardware aware architecture optimization and neural architecture search.

Email: lohitaksh.maruvada@iiitb.ac.in



MADHAV RAO is a Senior IEEE Member and is working as a faculty at IIIT-Bangalore. His group is working on VLSI Architecture Design. He teaches VLSI Architecture Design and Electronics courses at IIIT-Bangalore.

Email: mr@iiitb.ac.in

TAKANORI SHIMIZU is a department head at Sony India Software Centre, driving innovation in new business and technology.

Email: takanori.shimizu@sony.com

...