

Toward Energy-Efficient Massive MIMO: Graph Neural Network Precoding for Mitigating Non-Linear PA Distortion

Thomas Feys, Liesbet Van der Perre, François Rottenberg

Abstract

Massive MIMO systems are typically designed assuming linear power amplifiers (PAs). However, PAs are most energy efficient close to saturation, where non-linear distortion arises. For conventional precoders, this distortion can coherently combine at user locations, limiting performance. We propose a graph neural network (GNN) to learn a mapping between channel and precoding matrices, which maximizes the sum rate affected by non-linear distortion, using a high-order polynomial PA model. In the distortion-limited regime, this GNN-based precoder outperforms zero forcing (ZF), ZF plus digital pre-distortion (DPD) and the distortion-aware beamforming (DAB) precoder from the state-of-the-art. At an input back-off of -3 dB the proposed precoder compared to ZF increases the sum rate by 8.60 and 8.84 bits/channel use for two and four users respectively. Radiation patterns show that these gains are achieved by transmitting the non-linear distortion in non-user directions. In the four user-case, for a fixed sum rate, the total consumed power (PA and processing) of the GNN-precoder is 3.24 and 1.44 times lower compared to ZF and ZF plus DPD respectively. A complexity analysis shows six orders of magnitude reduction compared to DAB precoding. This opens perspectives to operate PAs closer to saturation, which drastically increases their energy efficiency.

Index Terms

MIMO systems, neural networks, nonlinear distortion, power amplifiers.

I. INTRODUCTION

A. Problem Statement

The wireless communications sector has seen a continuous increase in its carbon footprint and electricity usage [1], [2], leading to concerns about meeting the targets set forth by Europe's Green Deal [3], the United Nations Sustainable Development Goal (SDG) [4] and the Paris Agreement [5]. This growth in emissions is in stark contrast with today's climate ambitions, where emissions must shrink rather than grow. Next to this, energy consumption has become an important factor in the operating expenses of wireless networks, which is further expected to worsen. Both the exponential growth in data [6] and the increased price of energy, contribute to this.

The authors are with ESAT-DRAMCO, Campus Ghent, KU Leuven, 9000 Ghent, Belgium (email: thomas.feys@kuleuven.be). We would like to thank NVIDIA for providing the GPU that greatly accelerated our simulations.

The base station (BS) is responsible for the majority of the energy consumption of mobile networks [7]. Within the BS, the power amplifiers (PAs) are responsible for a major component of the energy consumption, which is typically $\sim 50\text{--}80\%$ [8]–[10]. Operating the PAs in an energy-efficient manner has been a significant challenge. PAs are most efficient close to their saturation point, where non-linear distortion arises [11]. The non-linear distortion limits the signal quality, leading to lower signal-to-noise-and-distortion ratios (SNDRs). This leads to a trade-off between energy efficiency and linearity. In the past, this trade-off between energy efficiency and linearity has been skewed towards linearity to maintain system capacity. This was done by operating PAs at a considerable back-off power in order to stay in the linear regime, which is detrimental for the energy efficiency. As a result, the current technology’s energy efficiency of PAs is typically low, ranging from 5% to 30% [9].

In this study, we present an approach to improve the energy efficiency of PAs by operating them closer to saturation. To do so, we study how a neural network can learn a massive MIMO (mMIMO) precoder that maximizes the sum rate, in the presence of non-linear distortion. This approach achieves a high capacity while using less back-off, thereby improving the energy efficiency. This in turn can reduce the operating expenses of wireless networks. Moreover, the cost of the PAs is an important component in the Bill-of-Materials (BOM) of wireless access equipment [8]. This cost is highly related to the maximum output power. Hence, by operating the PAs closer to saturation a lower maximum output power can be used to achieve the same average output power. The proposed method can be of particular interest for millimeter-wave systems which are typically highly non-linear [12], [13].

B. State-of-the-Art

The energy efficiency of the PA is typically limited by the need for a large back-off to remain within its linear regime. This is becoming less preferable given the growing emphasis on reducing energy consumption. Efforts to linearize the PA such as digital pre-distortion (DPD) are used in practical systems [14]. However, in mMIMO systems, DPD has to be deployed at each antenna. Recent studies show that when 5G systems would use fully digital beamforming, a large number of antennas and a large bandwidth, these DPDs would introduce a power consumption that is even higher than the PA consumption itself [14]. Moreover, the performance of DPD is limited by clipping, i.e., the PA can only be linearized up to the saturation point, so that a relatively large back-off is still required, even for a perfect DPD. Peak-to-average power ratio (PAPR) reduction techniques have been studied, which can reduce the amount of back-off needed [15]. For instance, in [16] a constant-envelope precoder has been proposed. However, practical implementations encountered challenges due to the substantial complexity of digital signal processing.

Various other approaches to the distortion problem have been proposed. For instance, in [17] a neural network-based digital predistorter is proposed. This DPD is applied before the precoding stage which leads to a complexity that scales with the number of users, rather than the number of antennas. Similarly, in [18] an end-to-end system is proposed where neural-network based precoding and decoding are jointly learned. In this scheme, the transmit symbol is essentially pre-distorted using a neural network, after which a zero-forcing precoder is applied, finally a neural network-based decoder is deployed at the receiver side. Alternatively, in [19] joint precoding and power allocation is proposed to solve the sum rate maximization problem in the presence of non-linear PAs. The authors

solve the optimization problem using an iterative approach that alternates between optimizing the precoding and power allocation matrices. In a similar vein, in [20] a multiplier punitive method based power allocation algorithm is proposed which maximizes the spectral efficiency with awareness of the non-linear amplifiers. Additionally, in [21] the problem of non-linear PAs in distributed networks is considered. The authors propose a constrained unsupervised learning approach to solve the sum rate maximization problem in the presence of non-linear PAs with a particular focus on reducing the communication overhead between the distributed access points.

Another compelling direction aims at incorporating knowledge of the distortion into the precoder design [22]–[25]. This allows for the spatial suppression of the distortion in the user directions, producing considerable gains over classical precoders. Moreover, this offers an advantage over DPD based solutions as the distortion created by saturation based clipping can be radiated in non-user directions, which improves the signal quality. Unfortunately, these solutions are still limited in their practical implementation, either because of their computational complexity or in terms of simplifying assumptions that need to be addressed. More specifically, in [25], the solution to the precoding problem is obtained by solving a non-convex optimization problem with a projected gradient descent-based procedure. As the problem is non-convex, the procedure is executed multiple times to obtain a close-to-optimal solution, which requires a high computational complexity. As an alternative solution for the problem, the authors in [22] derived a globally optimal closed-form solution for the simplified single-user case, a third order PA model and a line-of-sight (LOS) channel, which was later extended to a general channel in [23]. There is thus a need for a solution that has low complexity and can address the challenging case of spatially multiplexing many users. Additionally, in order to work close to saturation, this solution should be designed for polynomial PA models that have a higher order than the third order model, as the third order model is only viable far away from saturation.

C. Contributions

In this work, we propose the use of a graph neural network (GNN) to find a mapping from the channel matrix to the precoding matrix. This mapping is learned under the presence of non-linear PAs operating close to their saturation point, which increases their energy efficiency but as a consequence introduces non-linear distortion. The need for machine learning arises from the non-linear and non-convex nature of the optimization problem, which limits classical linear signal processing solutions. By learning the mapping from channel matrix to precoding matrix, a lot of the complexity is offloaded to the training step, which reduces the online computational complexity. This allows for a practical low-complexity solution even for the multi-user precoding problem under the presence of high-order non-linear PA distortion. Previous works have not addressed this problem with such a practical low-complexity solution. As such, this opens perspectives to operate PAs closer to their saturation point, which drastically increases their energy efficiency.

More specifically, the structure of the paper and our contributions are as follows:

- Section II describes the system model along with the non-linear PA models which are used to train and evaluate the proposed GNN precoder. Next to this, the optimization problem considering high-order PAs is formulated. It is outlined how this formulation is used as a loss function to train a neural network precoder.

- We compare a number of neural network architectures for linear precoding in Section III-A. This is done under the assumption of a simplified third-order PA model. A GNN is shown to outperform others such as the circular convolutional neural network (CCNN) presented in our previous work [26], where a CCNN was studied in the context of a simplified third-order PA model. Based on this comparison the GNN is selected for further training and evaluation on a high-order PA model.
- In Section III-B, we characterize the GNN architecture and how it can be adopted for precoding. This architecture is further extended in Section III-C to include the signal-to-noise ratio (SNR) as an additional input, which allows the precoder to perform well over the full SNR range, rather than having to be retrained at each SNR point to reach the desired performance.
- The complexity of the proposed GNN precoder is analysed in Section IV. Additionally, the complexity of the benchmark distortion-aware beamforming (DAB) precoder from [25] has been computed and used to compare to the GNN precoder.
- Section V provides extensive simulation results to validate the performance of the GNN-based precoder under the presence of high-order non-linear PAs. This includes the computation of the power consumption reduction, where the total consumed power (PA and processing) of the GNN precoder is quantified.
- Section VI concludes the paper.

Notations

Vectors and matrices are denoted by bold lowercase and bold uppercase letters respectively. A vector function is denoted by a bold letter while a scalar function is denoted by a non-bold letter. Superscripts $(\cdot)^*$, $(\cdot)^T$ and $(\cdot)^H$ stand for the conjugate, transpose and Hermitian transpose operators respectively. Subscripts $(\cdot)_m$ and $(\cdot)_k$ denote the antenna and user index. The expectation is denoted by $\mathbb{E}(\cdot)$. The $M \times M$ identity matrix is given by \mathbf{I}_M . The main diagonal of a square matrix \mathbf{A} is given by $\text{diag}(\mathbf{A})$. The trace of a matrix is given by $\text{Tr}(\cdot)$. The element-wise or Hadamard product of two matrices is denoted by $\mathbf{A} \odot \mathbf{B}$. The element at location (i, j) in matrix \mathbf{A} is indicated as $[\mathbf{A}]_{i,j}$. $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denotes a graph where \mathcal{V} is the set of nodes and \mathcal{E} the set of edges. The edge going from node $a \in \mathcal{V}$ to node $b \in \mathcal{V}$ is denoted as $(a, b) \in \mathcal{E}$. The neighborhood of a node a is denoted as $\mathcal{N}(a) = \{b \in \mathcal{V} : (a, b) \in \mathcal{E}\}$.

II. PROBLEM FORMULATION

A. System Model

A downlink mMIMO system with non-linear PAs and neural network-based linear precoding is considered as depicted in Fig. 1. In this system, K single-antenna users are served by the BS using M transmit antennas. The complex symbol intended for user k is denoted as s_k , it is assumed to be zero mean circularly symmetric complex Gaussian with unit variance. This is a valid assumption for e.g., orthogonal frequency division multiplexing (OFDM)

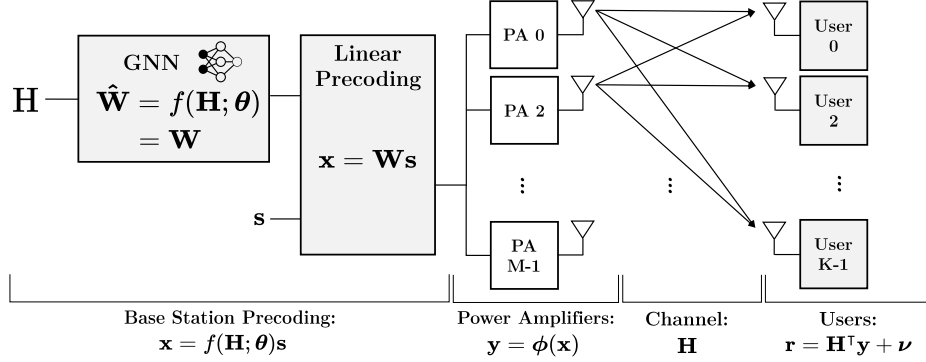


Fig. 1: Massive MIMO downlink with neural network-based linear precoding and non-linear amplifiers at the transmitter.

systems [27]. The symbols for different users are assumed to be uncorrelated. The linearly precoded symbol at antenna m is given by

$$x_m = \sum_{k=0}^{K-1} w_{m,k} s_k, \quad (1)$$

here $w_{m,k}$ is the precoding coefficient for user k at antenna m . In matrix form, the precoded symbol vector $\mathbf{x} \in \mathbb{C}^{M \times 1}$ is

$$\mathbf{x} = \mathbf{W}\mathbf{s} \quad (2)$$

where $\mathbf{W} \in \mathbb{C}^{M \times K}$ is the precoding matrix and $\mathbf{s} \in \mathbb{C}^{K \times 1}$ the symbol vector. The amplified transmit vector $\mathbf{y} \in \mathbb{C}^{M \times 1}$ is then given by

$$\mathbf{y} = \phi(\mathbf{x}) \quad (3)$$

where $\phi(\mathbf{x}) = [\phi(x_0), \dots, \phi(x_{M-1})]^T$ denotes the element-wise non-linear transformation caused by the PAs. Alternatively, in scalar notation the amplified signal at antenna m can be denoted as $y_m = \phi(x_m)$. The received signal vector $\mathbf{r} \in \mathbb{C}^{K \times 1}$ is

$$\mathbf{r} = \mathbf{H}^T \mathbf{y} + \mathbf{v} = \mathbf{H}^T \phi(\mathbf{W}\mathbf{s}) + \mathbf{v}$$

with $\mathbf{H} \in \mathbb{C}^{M \times K}$ being the channel matrix. The vector $\mathbf{v} \in \mathbb{C}^{K \times 1}$ contains independently and identically distributed (i.i.d.) zero mean complex Gaussian noise samples with variance σ_v^2 .

B. Non-Linear PA Models

The input signal of the PAs is considered to be a bandpass signal, which is described by the following complex baseband representation

$$x(t) = A(t)e^{j\varphi(t)} \quad (4)$$

The output of a general non-linear PA with function $\phi(\cdot)$ is given by [11]

$$y(t) = \phi(x(t)) \quad (5)$$

$$= \phi_A(A(t)) e^{j\varphi(t) + \phi_\varphi(A(t))}, \quad (6)$$

where $\phi_A(\cdot)$ and $\phi_\varphi(\cdot)$ represent the amplitude modulation to amplitude modulation (AM/AM) and amplitude modulation to phase modulation (AM/PM) transfer function respectively. In this work, the non-linear PA is modeled as a complex valued polynomial of order $2N + 1$. Furthermore, only the odd order polynomial coefficients are considered given that they contribute to spectral components near the carrier frequency, while the even order coefficients cause spectral components at multiples of the carrier frequency [11]. Hence, the output of the PA at antenna m is given by

$$\phi(x_m) = \sum_{n=0}^N \beta_{2n+1} x_m |x_m|^{2n} \quad (7)$$

$$= \beta_1 x_m + \beta_3 x_m |x_m|^2 + \dots + \beta_{2N+1} x_m |x_m|^{2N} \quad (8)$$

where β_{2n+1} are complex coefficients that model both AM/AM and AM/PM distortion. Note that this formulation assumes that all M PAs have the same non-linear characteristic.

In order to obtain proper polynomial coefficients at the desired input back-off (IBO), a least squares regression of the polynomial model to the modified Rapp model [28] is performed. The AM/AM and AM/PM distortion of the modified Rapp model are

$$\phi_A(x_m) = \frac{|x_m|}{\left(1 + \left|\frac{x_m}{\sqrt{p_{\text{sat}}}}\right|^{2S}\right)^{\frac{1}{2S}}} \quad (9)$$

$$\phi_\varphi(x_m) = \frac{A|x_m|^q}{1 + \left|\frac{x_m}{B}\right|^q}. \quad (10)$$

The modified Rapp model coefficients are set as follows¹: $S = 2$, $q = 4$, $A = -0.315$, $B = 1.137$ and the saturation power of the PA, p_{sat} is scaled in order to produce the desired IBO according to $\text{IBO} = p_{\text{in}}/p_{\text{sat}}$, with p_{in} being the average input power at each PA.

Additionally, as a benchmark a perfect DPD is considered. This is modeled as a soft limiter, i.e., a linear AM/AM characteristic up to a certain saturation point where the output of the PA is clipped [11], which is equivalent to the AM/AM characteristic of the Rapp model for S going to infinity. The AM/AM characteristic is thus modeled as

$$\phi_A(x_m) = \begin{cases} |x_m| & \text{for } |x_m| \leq \sqrt{p_{\text{sat}}} \\ \sqrt{p_{\text{sat}}} & \text{for } |x_m| > \sqrt{p_{\text{sat}}} \end{cases}, \quad (11)$$

while the AM/PM conversion is zero. An overview of these PAs models is given in Fig. 2.

¹Adapted from [29] to a PA with unit gain.

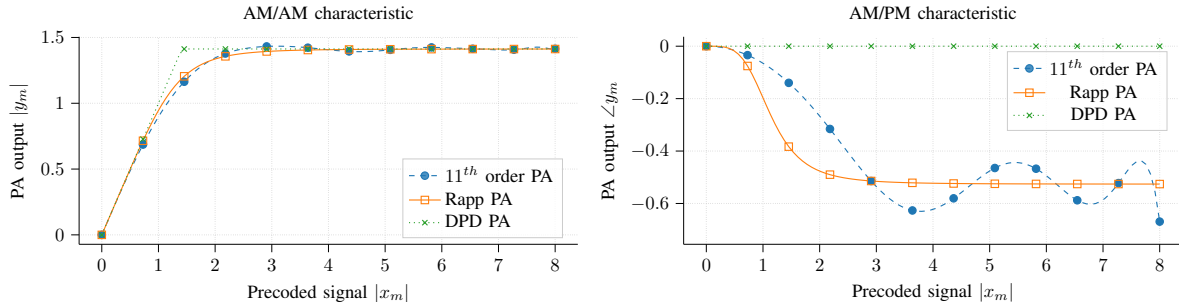


Fig. 2: AM/AM and AM/PM characteristics of the considered power amplifier models: modified Rapp model, 11th order polynomial model and a soft limiter/perfect DPD.

C. Radiation Patterns

In Section V, at some places, the radiation pattern of the proposed precoder is evaluated. For this, a pure LOS channel is considered. The channel from antenna m to user k can then be written as

$$h_{m,k} = \sqrt{\beta_k} e^{-jm \frac{2\pi}{\lambda_c} d \cos(\theta_k)}. \quad (12)$$

Here, β models the path loss, while $m \frac{2\pi}{\lambda_c} d \cos(\theta_k)$ represents the antenna dependent phase shift when considering a uniform linear array (ULA) and a narrowband system. Additionally, λ_c denotes the carrier wavelength, θ_k the user angle and d the antenna spacing. The radiation pattern in an arbitrary direction $\tilde{\theta}$ can be obtained by

$$P(\tilde{\theta}) = \mathbb{E} \left(\left| \sum_{m=0}^{M-1} y_m e^{-jm \frac{2\pi}{\lambda_c} d \cos(\tilde{\theta})} \right|^2 \right). \quad (13)$$

The radiation pattern of the linearly amplified signal is obtained by replacing y_m by x_m in (13) and is denoted as $P_{\text{lin}}(\tilde{\theta})$. The radiation pattern of the non-linear distortion is denoted as $P_{\text{dist}}(\tilde{\theta})$ and is obtained by replacing y_m by $\sum_{n=1}^N \beta_{2n+1} x_m |x_m|^{2n}$ in (13). Additionally, the signal-to-distortion ratio (SDR) radiation pattern is defined as $P_{\text{SDR}}(\tilde{\theta}) = \frac{P_{\text{lin}}(\tilde{\theta})}{P_{\text{dist}}(\tilde{\theta})}$.

D. Problem Formulation

An achievable sum rate R_{sum} , i.e., a lower bound on the capacity, is obtained by considering that the noise and distortion are jointly Gaussian distributed and independent from the data symbols, this can be seen as a worst case

$$R_{\text{sum}} = \sum_{k=0}^{K-1} \log_2(1 + \text{SNIDR}_k), \quad (14)$$

where SNIDR_k is the signal-to-noise-and-interference-and-distortion ratio (SNIDR) at user k . It can be computed based on the Busgang decomposition [30], which implies that the received signal for user k can be written as $r_k = B_k s_k + d_k + v_k$. Here, d_k captures both the non-linear distortion and inter-user interference, which is uncorrelated to the transmit signal s_k and the noise v_k . The linear gain is given by $B_k = \mathbb{E}(r_k s_k^*) / p_k$, with $p_k = \mathbb{E}(s_k s_k^*)$ [30]. The received signal variance for user k is given by $|B_k|^2 p_k$. The distortion and inter-user

interference can be computed as $\mathbb{E}(|d_k|^2) = \mathbb{E}(|r_k|^2) - |B_k|^2 p_k - \sigma_{v_k}^2$, given that d_k , s_k and v_k are uncorrelated. The SNIDR for user k is then given by

$$\text{SNIDR}_k = \frac{|B_k|^2 p_k}{\mathbb{E}(|d_k|^2) + \sigma_v^2}. \quad (15)$$

This general expression for the SNIDR can be evaluated numerically for a general PA model and is used for the evaluation in Section V.

For training the neural networks (NNs), the $(2N+1)^{\text{th}}$ order polynomial PA model in (7) is assumed, which leads to an analytical expression for the SNIDR [25], [31]. By applying Bussgang's theorem [30] to the amplification stage, we can write the amplified signal as

$$\phi(\mathbf{x}) = \mathbf{G}\mathbf{x} + \mathbf{e}, \quad (16)$$

with $\mathbf{e} \in \mathbb{C}^{M \times 1}$ the non-linear distortion term and $\mathbf{G} \in \mathbb{C}^{M \times M}$ a diagonal matrix containing the Bussgang gains with the diagonal entries being $[\mathbf{G}]_{m,m} = \mathbb{E}[\phi(x_m)x_m^*]/\mathbb{E}[|x_m|^2]$.

When assuming the $(2N+1)^{\text{th}}$ order polynomial model and linear precoding ($\mathbf{x} = \mathbf{W}\mathbf{s}$), the gain matrix \mathbf{G} can be written as a function of the precoding matrix \mathbf{W} [31]

$$\mathbf{G}(\mathbf{W}) = \sum_{n=0}^N (n+1)! \beta_{2n+1} \mathbf{I}_M \text{diag}(\mathbf{C}_x)^n \quad (17)$$

The input covariance matrix is given by $\mathbf{C}_x = \mathbb{E}[\mathbf{x}\mathbf{x}^H] = \mathbf{W}\mathbf{W}^H$. From [25], [31], the covariance matrix of the non-linear distortion \mathbf{e} can be derived as

$$\mathbf{C}_e(\mathbf{W}) = \sum_{n=1}^N \mathbf{L}_n \mathbf{C}_x \odot |\mathbf{C}_x|^{2n} \mathbf{L}_n^H, \quad (18)$$

with

$$\mathbf{L}_n = \frac{1}{\sqrt{n+1}} \sum_{l=n}^N \binom{l}{n} (l+1)! \beta_{2l+1} \mathbf{I}_M \text{diag}(\mathbf{C}_x)^{l-n}. \quad (19)$$

The received signal at user k can then be written as

$$r_k = \underbrace{\mathbf{h}_k^T \mathbf{G}(\mathbf{W}) \mathbf{w}_k s_k}_{\text{desired signal}} + \underbrace{\sum_{k' \neq k} \mathbf{h}_k^T \mathbf{G}(\mathbf{W}) \mathbf{w}_{k'} s_{k'}}_{\text{inter-user interference}} + \underbrace{\mathbf{h}_k^T \mathbf{e}}_{\text{received non-linear distortion}} + \underbrace{v_k}_{\text{noise}}. \quad (20)$$

This leads to the following SNIDR expression for user k

$$\text{SNIDR}_k(\mathbf{W}) = \frac{|\mathbf{h}_k^T \mathbf{G}(\mathbf{W}) \mathbf{w}_k|^2}{\sum_{k' \neq k} |\mathbf{h}_k^T \mathbf{G}(\mathbf{W}) \mathbf{w}_{k'}|^2 + \mathbf{h}_k^T \mathbf{C}_e(\mathbf{W}) \mathbf{h}_k^* + \sigma_v^2}. \quad (21)$$

Given this expression for the SNIDR, an achievable sum rate can be computed using (14). As such, the optimization problem we aim to solve can be formulated as

$$\begin{aligned} \max_{\mathbf{W}} \quad & R_{\text{sum}}(\mathbf{W}) \\ \text{s.t.} \quad & \mathbb{E} \left(\sum_{m=0}^{M-1} |x_m|^2 \right) = \text{Tr}(\mathbf{W}\mathbf{W}^H) \leq P_T, \end{aligned} \quad (22)$$

where P_T is the total transmit power. The aim is thus to find a precoding matrix which maximizes the sum rate, subject to a power constraint², while the system is affected by PA non-linearities.

E. Neural Network Training

The neural network-based precoder considered in this work takes as input the channel matrix \mathbf{H} and outputs the linear precoding matrix $\hat{\mathbf{W}}$. This neural network can be represented as $\hat{\mathbf{W}} = f(\mathbf{H}; \boldsymbol{\theta})$ i.e., a learned non-linear function mapping which is parameterized by $\boldsymbol{\theta}$. Training is done in a self-supervised manner by maximizing the sum rate in order to obtain the neural network parameters

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} -R_{\text{sum}}(f(\mathbf{H}; \boldsymbol{\theta})). \quad (23)$$

Here the sum rate R_{sum} is given by (14), where the SNIDR is computed according to (21). Since the SNIDR in (21) directly depends on the output of the neural network (i.e., the precoding matrix $\hat{\mathbf{W}}$), the gradients of this loss function, with respect to the neural network parameters $\boldsymbol{\theta}$, can directly be computed using backpropagation. Given these gradients, the parameters of the NN are updated using the Adam optimizer [32].

F. Benchmark Algorithms

Throughout this study, the proposed solution is compared against a number of benchmark algorithms. First, maximum ratio transmission (MRT) and zero forcing (ZF) precoding are considered [33]

$$\mathbf{W}^{\text{MRT}} = \alpha \mathbf{H}^* \quad (24)$$

$$\mathbf{W}^{\text{ZF}} = \alpha \mathbf{H}^* (\mathbf{H}^T \mathbf{H}^*)^{-1}, \quad (25)$$

where $\alpha = \sqrt{P_T / \text{Tr}(\mathbf{W}\mathbf{W}^H)}$ is a power normalization constant. The MRT and ZF precoding matrices are obtained under the assumption that the PA is linear. Hence, these two algorithms act as a benchmark in the linear regime where no distortion is present. Additionally, they represent a worst-case scenario, highlighting how the system's performance deteriorates when non-linear distortion is present and no mitigation techniques are adopted.

Second, the zero third-order distortion (Z3RO) precoder from [22] is considered. This precoder was designed for the single-user case, assuming a third-order PA model. It maximizes the SNR while nulling the third-order distortion term at the user location. This is done by saturating a number of antennas with an opposite phase shift, as can be seen in (26). The closed-form expression for the precoding coefficient at antenna m is given by

$$w_m^{\text{Z3RO}, M_s} = \alpha h_m^* \begin{cases} -\gamma & \text{if } m = 0, \dots, M_s - 1 \\ 1 & \text{otherwise} \end{cases}, \quad (26)$$

here $\gamma = \left(\frac{\sum_{m'=M_s}^{M-1} |h_{m'}|^4}{\sum_{m''=0}^{M_s-1} |h_{m''}|^4} \right)^{1/3}$ is the gain of the saturated antennas and M_s is the number of saturated antennas, which is set to $M_s = 1$ in all simulations, as this gives the globally optimal solution [23]. In [23], this

²For simplicity, the power constraint is taken before the PA, which neglects the non-linearly amplified power, which is small compared to the full transmit power. Due to the saturation characteristic of the PA, this can be seen as an upper bound on the actual transmit power.

precoder was shown to be optimal in the single-user case and when a third-order PA model is assumed. Hence, in this study, it is used to evaluate how close the NN precoder is to the optimal solution, under these assumptions.

For the more general multi-user case and when high-order PAs are assumed, the problem becomes non-convex, consequently, no optimal solution is known. However, in [25] the non-convex optimization problem is approximately solved using an iterative projected gradient ascent method, which is denoted as the DAB precoder. For the multi-user and high-order PA case, this algorithm is used as a benchmark. The precoding matrix obtained by this algorithm is denoted as \mathbf{W}^{DAB} and can be obtained by performing the following algorithm for I iterations

$$\mathbf{W}^{(i+1)} = \left[\mathbf{W}^{(i)} + \mu^i \nabla_{\mathbf{W}} R_{\text{sum}} \left(\mathbf{W}^{(i)} \right) \right]_{\mathbb{E} \|\phi(\mathbf{W}_s)\|^2 = P_T}^+ \quad (27)$$

Here μ^i is the step size for the gradient update at iteration i and $[\cdot]_{\mathbb{E} \|\phi(\mathbf{W}_s)\|^2 = P_T}^+$ denotes the projection of the obtained solution, such that the power constraint is met³.

III. NEURAL NETWORKS FOR PRECODING

A. Suitable Neural Network Architectures

Neural networks can approximate any continuous non-linear function with arbitrary accuracy [34]. Subsequently, they can be used to learn a mapping from channel matrix \mathbf{H} to precoding matrix $\hat{\mathbf{W}}$. Unfortunately, the universal approximation theorem is not constructive, i.e., it does not provide a way to select the NN architecture that can achieve this arbitrary accuracy. A multilayer perceptron (MLP) is a general fully connected feedforward neural network, which is a universal function approximator [34]. As such, in theory, MLPs could be used to learn the mapping between channel and precoding matrix. In practice, because of the general structure of MLPs, this leads to a very large number of trainable parameters. Consequently, these MLPs are very hard to train and have a high computational complexity at inference time. As such, it is advantageous to select a NN architecture, with a certain inductive bias, that suits the learning task. The inductive bias limits the types of functions that can be learned, thus decreasing the size of the hypothesis space that the NN covers. If this inductive bias corresponds to the learning task, the desired function is still covered by the (reduced) hypothesis space of the NN. This produces more scalable NNs, with fewer learnable parameters, that are easier to train and have a lower computational complexity [35].

In order to reduce the hypothesis space for the precoding problem the permutation equivariance property of precoding can be utilized. From [36], it is clear that precoding is permutation equivariant with respect to the users and antennas. To clarify, if the order of the users or antennas in the channel matrix \mathbf{H} is permuted, the order of the precoding vectors in \mathbf{W} should be permuted accordingly, but the sum rate remains unchanged. More formally, a precoding function $\mathbf{W} = f(\mathbf{H})$ is permutation equivariant if the following holds $f(\mathbf{\Pi}_1 \mathbf{H} \mathbf{\Pi}_2^T) = \mathbf{\Pi}_1 f(\mathbf{H}) \mathbf{\Pi}_2^T$, where $\mathbf{\Pi}_1, \mathbf{\Pi}_2$ are permutation matrices [36]. Hence, when a NN architecture naturally exhibits this permutation equivariance property, it is beneficial for the precoding task. CCNNs are shift equivariant [36], more formally this implies that $f_{\text{CCNN}}(\mathbf{S}_1 \mathbf{H} \mathbf{S}_2^T) = \mathbf{S}_1 f_{\text{CCNN}}(\mathbf{H}) \mathbf{S}_2^T$, where \mathbf{S}_1 and \mathbf{S}_2 are shift matrices i.e., circulant permutation

³Note that in [25] the power constraint is considered after the PA in contrast to our approach where the power constraint is enforced before the PA. This implies that our proposed precoder abides by both power constraints, i.e., before and after the PA, while the DAB precoder only abides by the constraint after the PA.

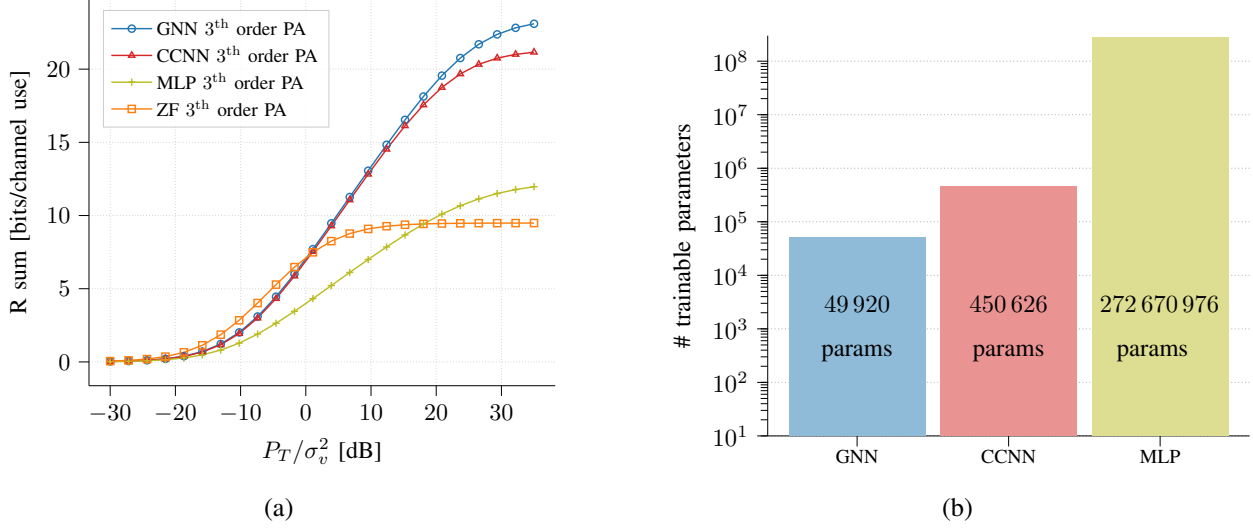


Fig. 3: (a) Comparison of an MLP, CCNN from [26] and GNN for $M = 64$, $K = 2$. Trained and evaluated on a third-order non-linear PA at -3 dB IBO, with $\beta_3 = -0.07781605 - 0.0401193j$. (200K training samples, 10K test samples) (b) Comparison of the number of trainable parameters of these three architectures.

matrices [36]. In [36] it is shown that CCNNs can learn permutation equivariant functions for certain values of the learned weight matrices. However, in general, the functions learned by the CCNN are not permutation equivariant. This implies that the inductive bias of the CCNN is more general than solely consisting of permutation equivariant functions. This means that the hypothesis space of a CCNN is larger than that of a NN that learns purely permutation equivariant functions. In our previous work [26], we showed that these CCNNs can be used to learn precoding functions that can cancel third-order non-linear distortion. In the current study, we show that GNNs can produce even better results, in the presence of both third and higher order non-linear distortion. GNNs are naturally permutation equivariant [37], which gives them the appropriate inductive bias for precoding.

To illustrate this, Fig. 3a denotes the performance of the different NN architectures for a scenario where $M = 64$, $K = 2$ and third-order non-linear amplifiers are considered. All 3 architectures are matched in terms of their representation capacity i.e., each of them has 6 layers and 64 features⁴ per layer. From this image it is clear that the GNN achieves the highest sum rate as compared to the CCNN and MLP. The CCNN achieves a slightly lower sum rate, because of the shift invariance it captures, which is still a relatively good inductive bias for precoding. The MLP has a very general structure and as such can barely produce a sum rate that is higher than that of the ZF precoder. Moreover, in Fig. 3b the number of learnable parameters of each NN architecture is illustrated. This again highlights the proficiency of the GNN as it only has 50 000 learnable parameters as compared to the 450 000 and 270 000 000 of the CCNN and MLP.

⁴64 features indicates that the real and imaginary part of each precoding coefficient is represented by a feature vector of dimension 64, resulting in a total of $64MK2$ features.

B. Graph Neural Network Precoding

GNNs generally come in two forms, namely GNNs that learn a hidden representation of the nodes in a graph and GNNs that learn a hidden representation of the edges in a graph. For the precoding problem, we focus on GNNs that learn representations of the edges of the graph. In general, the goal is thus to start from a number of input edge features and learn edge embeddings. Such a NN consists of a number of layers or message-passing iterations, where each layer/iteration produces a new representation/embedding of the edges. An embedding/representation of edge (a, b) in layer l of a GNN is expressed as $\mathbf{z}_{(a,b)}^{(l)}$. A typical GNN layer consists of two steps, namely, a message-passing/aggregation step and an update step. Hence, a single layer for edge (a, b) can be expressed as

$$\mathbf{z}_{(a,b)}^{(l+1)} = \text{UPDATE}^{(l)}(\mathbf{z}_{(a,b)}^{(l)}, \mathbf{m}_{\mathcal{N}(a)}^{(l)}, \mathbf{m}_{\mathcal{N}(b)}^{(l)}) \quad (28)$$

Here $\mathbf{m}_{\mathcal{N}(a)}, \mathbf{m}_{\mathcal{N}(b)}$ are 'messages' that are aggregated from the neighboring edges of the two nodes connected to edge (a, b) , these messages are defined as

$$\mathbf{m}_{\mathcal{N}(a)}^{(l)} = \text{AGGREGATE}^{(l)}(\{\mathbf{z}_{(a,x)}^{(l)}, \forall x \in \mathcal{N}(a)\}) \quad (29)$$

$$\mathbf{m}_{\mathcal{N}(b)}^{(l)} = \text{AGGREGATE}^{(l)}(\{\mathbf{z}_{(b,x)}^{(l)}, \forall x \in \mathcal{N}(b)\}). \quad (30)$$

Here 'UPDATE' and 'AGGREGATE' are differentiable functions. For a comprehensive overview of commonly used update and aggregate functions we refer to [37].

More specifically, the precoding problem can be represented as learning a mapping from a $M \times K$ MIMO channel matrix to a $M \times K$ precoding matrix over a graph, a visualization of this is given in Fig. 4. The graph can be represented as follows: each antenna m and user k is denoted by a node. The edges between the antennas and users represent the wireless channel and have as input features the channel coefficients. This graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the set of nodes and \mathcal{E} the set of edges. \mathcal{V} contains all user and antenna nodes, i.e., $m \in \mathcal{V} \forall m$ and $k \in \mathcal{V} \forall k$. \mathcal{E} contains all edges between the antennas and users i.e., $(m, k) \in \mathcal{E} \forall m, k \in \mathcal{V}$. This graph is undirected, meaning that $(m, k) \in \mathcal{E} \leftrightarrow (k, m) \in \mathcal{E}$, as is the physical channel between each user and BS antenna. For this graph, a single GNN layer that updates the representation of edge (m, k) can be defined as

$$\mathbf{z}_{(m,k)}^{(l+1)} = \text{UPDATE}_{\text{edge}}^{(l)}(\mathbf{z}_{(m,k)}^{(l)}, \mathbf{m}_{\mathcal{N}(m)}^{(l)}, \mathbf{m}_{\mathcal{N}(k)}^{(l)}) \quad (31)$$

$$= \sigma\left(\mathbf{W}_{\text{edge}}^{(l)} \mathbf{z}_{(m,k)}^{(l)} + \mathbf{W}_m^{(l)} \mathbf{m}_{\mathcal{N}(m)}^{(l)} + \mathbf{W}_k^{(l)} \mathbf{m}_{\mathcal{N}(k)}^{(l)}\right), \quad (32)$$

where $\mathbf{z}_{(m,k)}^{(l+1)}$ denotes the hidden representation of the edge (m, k) at layer/iteration $l + 1$ of the GNN, $\sigma(\cdot)$ is a non-linear activation function and the matrices $\mathbf{W}_{\text{edge}}^{(l)}, \mathbf{W}_m^{(l)}, \mathbf{W}_k^{(l)} \in \mathbb{R}^{d_{l+1} \times d_l}$ are learnt using a form of stochastic gradient descent. Note that the dimensions of these matrices, d_{l+1} and d_l , determine the number of features in layer $l + 1$ and l respectively. The aggregation operations are illustrated in Fig. 4a and 4b and are defined as follows

$$\mathbf{m}_{\mathcal{N}(m)}^{(l)} = \frac{1}{|\mathcal{N}(m)|} \sum_{k' \in \mathcal{N}(m)} \mathbf{z}_{(m,k')}^{(l)} \quad (33)$$

$$\mathbf{m}_{\mathcal{N}(k)}^{(l)} = \frac{1}{|\mathcal{N}(k)|} \sum_{m' \in \mathcal{N}(k)} \mathbf{z}_{(m',k)}^{(l)}. \quad (34)$$

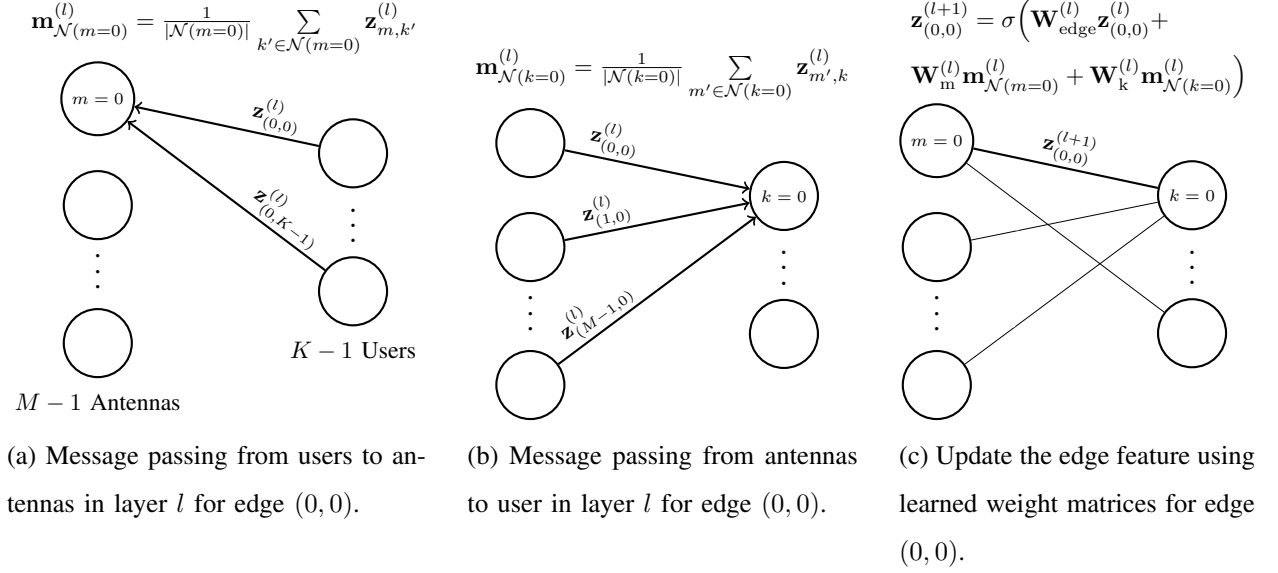


Fig. 4: GNN visualized for a single edge $(0, 0)$ in the graph. In (a) and (b) message passing is performed. Note that in the first layer $l = 0$ the channel coefficients act as input edge features i.e., $\mathbf{z}_{(m,k)}^{(0)} = [\Re\{h_{m,k}\}, \Im\{h_{m,k}\}]^\top$. In (c) the messages and edge features are used to update the edge feature which acts as the output of this layer. This is repeated for each layer to generate the outputs of the network. The outputs of the network are the edge features in the final layer which represent the precoding coefficients $\mathbf{z}_{(m,k)}^{(L-1)} = [\Re\{w_{m,k}\}, \Im\{w_{m,k}\}]^\top$. (Note that the power normalization layer is omitted for simplicity.)

The input and output for edge (m, k) of the network are defined in the following way

$$\mathbf{z}_{(m,k)}^{(0)} = [\Re\{[\mathbf{H}]_{m,k}\}, \Im\{[\mathbf{H}]_{m,k}\}]^\top \quad (35)$$

$$\mathbf{z}_{(m,k)}^{(L-1)} = [\Re\{[\hat{\mathbf{W}}]_{m,k}\}, \Im\{[\hat{\mathbf{W}}]_{m,k}\}]^\top. \quad (36)$$

In order to respect the power constraint, a power normalization layer is added, which consists of a scalar normalization given by

$$\hat{\mathbf{W}}^{norm} = \alpha \hat{\mathbf{W}}, \quad (37)$$

with $\alpha = \sqrt{P_T / \text{Tr}(\hat{\mathbf{W}} \hat{\mathbf{W}}^H)}$.

C. Leveraging SNR Regime Information

For the neural network architecture as described in the Section III-B, the network is typically trained at a fixed SNR ($= P_T / \sigma_v^2$) point. This results in the fact that the performance of the neural network is best close to the SNR value it is trained at. In this section, it is outlined how the SNR can be included as an input edge feature. This allows the neural network to be trained at a number of SNR points, depending on the input the neural network can

differentiate in which SNR regime it is operating and deliver the best performance. To do so the input to the GNN is defined in the following way

$$\mathbf{z}_{(m,k)}^{(0)} = \left[\Re\{\mathbf{H}\}_{m,k}, \Im\{\mathbf{H}\}_{m,k}, \text{SNR}_{\text{norm}} \right]^T. \quad (38)$$

To ensure stability during training we include the normalized SNR as input rather than the absolute SNR. The normalized SNR is defined as follows

$$\text{SNR}_{\text{norm}} = \frac{1}{\text{SNR}_{\text{max}}} \frac{P_t}{\sigma_v^2}. \quad (39)$$

Training is done for an SNR ($= P_T/\sigma_v^2$) range of -30 to 30 dB, with a stepsize of 5 dB, hence $\text{SNR}_{\text{max}} = 30$ dB. During training, the SNR for each training example is randomly selected from this range. During testing, the sum rate is evaluated for an SNR range of -30 to 30 dB, however this time with a spacing of 2.5 dB. This ensures that SNR points for which the model is not trained are also tested.

IV. COMPLEXITY AND REAL-TIME ANALYSIS

In this section, the complexity of the proposed solution is quantified and compared against the benchmark precoders. Next to this, it is outlined which operations of the GNN can be performed in parallel in order to produce faster execution times for real-time implementations. In order to quantify the computational complexity of the proposed solution, the number of real floating point additions and multiplications are computed. Note that an addition and multiplication carry a different computational cost with the multiplication being the most costly one. However this highly depends on the hardware platform [38]. As such, both the number of real floating point additions and multiplications are computed. Additionally the number of floating point operations (FLOPs) is computed. A FLOP is defined as either a real floating point multiplication or addition. When considering complex multiplications or additions a conversion to real multiplications and additions is made. A complex multiplication requires 4 real multiplications and 2 real additions, while a complex addition requires 2 real additions. Note that the number of computations needed to perform the power normalization of the precoders is neglected, as this is equal across the precoders⁵. An overview of the complexity analysis can be found in Table I.

A. Complexity of the GNN

In order to perform one forward pass of the GNN (32) has to be computed L times, namely the number of layers in the GNN, for each edge $(m, k) \in \mathcal{E}$ i.e., MK times. Note that our GNN architecture has $L = 8$ layers, meaning it consists out of an input layer, 6 hidden layers and 1 output layer. The number of in- and output features is fixed to $d_0 = d_{L-1} = 2$ and the number of features in each hidden layer is constant at $d_l = 128 \forall l \in \{2, \dots, L-2\}$. The GNN first performs message passing as illustrated in Fig. 4a and 4b i.e., computing $\mathbf{m}_{\mathcal{N}(m)}^{(l)}$ and $\mathbf{m}_{\mathcal{N}(k)}^{(l)}$, which can be done by computing (33) and (34) this has the following complexity:

⁵Note that the DAB precoder employs a more complex power normalization which is performed after the amplification stage. However, a more simple power normalization before the PA could be used for this precoder, which would render the complexity of the normalization across all precodes equal. As such we decided not to include this term into the complexity analysis.

- Input layer: $2(M - 1) + 2(K - 1)$ real floating point additions
- Output and Hidden layers: $(L - 1)(d_l(M - 1) + d_l(K - 1))$ real floating point additions

Subsequently, the update step is performed as illustrated in Fig. 4c. This involves a number of matrix multiplications with the learned weight matrices, followed by the addition of these products, this has the following complexity:

- Input and output layer: $12d_l$ real multiplications and $11d_l - 2$ real additions
- Hidden layers: $(L - 2)3d_l^2$ real multiplications and $(L - 2)(3d_l^2 - d_l)$ real additions

Note that the complexity of the non-linear activation function is neglected as it is often approximated based on lookup tables, which requires few computations [39].

In total the GNN needs the following number of real floating point multiplications

$$\underbrace{MK}_{\text{edges}} \left(\underbrace{6d_l}_{\text{input layer}}^{(a)} + \underbrace{(L - 2) 3d_l^2}_{\text{hidden layers}}^{(a)} + \underbrace{6d_l}_{\text{output layer}}^{(a)} \right) \quad (40)$$

and the following number of real floating point additions

$$\underbrace{MK}_{\text{edges}} \left(\underbrace{5d_l + 2(M - 1 + K - 1)}_{\text{input layer}}^{(a)} + \underbrace{(L - 2) \left(3d_l^2 - d_l + d_l(M - 1 + K - 1) \right)}_{\text{hidden layers}}^{(a)} + \underbrace{6d_l - 2}_{\text{output layer}}^{(a)} \right). \quad (41)$$

where the term (a) denotes the computations needed to perform the matrix multiplication with the learned weight matrices followed by the addition of the resulting products and (b) denotes the computations needed for the message passing. This scales as $\mathcal{O}(MKd_l^2L)$ floating point multiplications and $\mathcal{O}(MKd_l^2L + M^2Kd_lL + MK^2d_lL)$ floating point additions. In terms of FLOPs this corresponds to $\mathcal{O}(MKd_l^2L + M^2Kd_lL + MK^2d_lL)$. Ultimately, it is important to highlight that the utilization of NNs offers a distinct advantage. Namely, a substantial portion of the computational tasks can be effectively parallelized, thereby resulting in a faster execution time. In our specific scenario, the execution of (32) can be parallelized across all edges, yielding a reduced number of FLOPs that need to be executed serially, namely $\mathcal{O}(d_l^2L + Md_lL + Kd_lL)$. This can be leveraged to achieve real-time operation.

B. Complexity of Zero Forcing Precoder

In order to compute the ZF precoder as given in (25), first a matrix multiplication of a $K \times M$ with an $M \times K$ matrix is computed, which requires MK^2 complex multiplications and $K^2(M - 1)$ complex additions, this corresponds to $2MK^2 - K^2$ FLOPs. Second, the inverse of a $K \times K$ matrix needs to be computed, which can efficiently be done by first computing a Cholesky factorization, followed by taking the inverse of the resulting lower triangular matrix [40]. This requires $\frac{1}{2}K^3 + \frac{3}{2}K^2$ complex multiplications, $\frac{1}{2}K^3 - \frac{1}{2}K^2$ complex additions and the computation of K square-root operations, which is equivalent to $K^3 + K^2 + K$ complex FLOPs. Finally, a matrix multiplication of an $M \times K$ with an $K \times K$ matrix needs to be computed, which requires MK^2 complex multiplications and $MK(K - 1)$ complex additions which results in $2MK^2 - MK$ complex FLOPs. This results in a total number of $2MK^2 + \frac{1}{2}K^3 + \frac{3}{2}K^2$ complex multiplications and $2MK^2 + \frac{1}{2}K^3 - \frac{3}{2}K^2$ complex additions. This is equivalent to $8MK^2 + 2K^3 + 6K^2$ real multiplications, $8MK^2 + 2K^3$ real additions or $16MK^2 + 4K^3 + 6K^2$ real FLOPs. This scales as $\mathcal{O}(MK^2 + K^3)$ FLOPs.

C. Complexity of Distortion-aware Beamforming Precoder

The DAB precoder executes an iterative optimization procedure for $P = 50$ different initializations. For each initialization, (27) is computed for $I = 1000$ iterations. This requires $2MK$ real additions and $2MK$ real multiplications, not taking into account the computation of the gradient $\nabla_{\mathbf{W}} R_{\text{sum}}(\mathbf{W}^{(i)})$. In order to compute the gradient, [25] employs a finite differences approximation which is computed for each element as

$$[\nabla_{\mathbf{W}} R_{\text{sum}}(\mathbf{W})]_{m,k} \approx \frac{\left(R_{\text{sum}}(\mathbf{W} + \Delta_{\Re}^{(m,k)}) - R_{\text{sum}}(\mathbf{W}) \right) + j \left(R_{\text{sum}}(\mathbf{W} + \Delta_{\Im}^{(m,k)}) - R_{\text{sum}}(\mathbf{W}) \right)}{\delta},$$

where $\Delta_{\Re}^{(m,k)} \in \mathbb{R}^{M \times K}$ is a matrix containing only zeros except for δ at row m and column k , where δ is a small positive constant. Similarly $\Delta_{\Im}^{(m,k)} \in \mathbb{R}^{M \times K}$ contains all zeros except for $j\delta$ at row m and column k . The computation of the additions, subtractions and quotient in this equation requires 4 real additions and 2 real multiplications. Additionally, in this equation the sum rate needs to be evaluated 3 times according equations (14), (21), (17), (18) and (19). Equation (14) requires $K(K-1)$ complex additions, (21) requires $(K+1)(M^2-1)$ complex additions, 2 real additions, $(K+1)(M^2-1)+1$ complex multiplications and 1 real multiplication. Equation (17) requires $M^2(K-1)+5M$ complex additions and M^2K+9M complex multiplications, (18) requires $5M^3-5M^2+4M$ complex additions and $5M^3+15M$ multiplications and (19) requires $10M$ additions and $10M$ multiplications. This leads to the following number of real floating point additions and multiplications respectively

$$PI [60M^4K^2 + 24M^3K^3 + 12M^3K^2 + 12M^2K^3 + 420M^2K^2 + 15MK^2 + 4MK]$$

$$PI [60M^4K^2 + 24M^3K^3 - 24M^3K^2 + 6M^2K^3 + 324M^2K^2 - 3MK^3 + 3MK^2 + 6MK]$$

This scales as $\mathcal{O}(PIM^4K^2 + PIM^3K^3)$ real floating point additions, $\mathcal{O}(PIM^4K^2 + PIM^3K^3)$ real floating point multiplications or $\mathcal{O}(PIM^4K^2 + PIM^3K^3)$ FLOPs.

D. Numerical Comparison

An overview of the complexity of the GNN and the different benchmarks is given in Table I. Additionally, a numerical example of the number of FLOPs is given for the following case: $P = 50$, $I = 1000$, $M = 64$, $K = 4$, $d_l = 128$ and $L = 8$. This illustrates that the GNN precoder has a complexity that is six orders of magnitude smaller as compared to the DAB precoder. Moreover, when the ability of the GNN to parallelize the computations across the edges is utilized, the number of FLOPs that need to be executed in serial is eight orders of magnitude smaller as compared to the DAB precoder. Next to this, we see that when the GNN is parallelized, the number of serial FLOPs is approximately 200 times higher than the ZF precoder.

To keep a concise and fair comparison between the GNN and the benchmarks, only the number of FLOPs is taken into account. However, specialized NN accelerators can improve the inference times of NNs and reduce the energy required to run them [38], [41]. Next to this a myriad of acceleration techniques can be applied to neural networks such as tensor decomposition, pruning, general matrix multiplication, Winograd transformation, etc. [41]. Additionally, NNs are known to be able to work on a lower bit resolution, leading to the use of fixed point arithmetic

TABLE I: Complexity of the GNN, ZF and DAB precoder. Example for the values $P = 50$, $I = 1000$, $M = 64$, $K = 4$, $d_l = 128$ and $L = 8$.

	GNN	GNN (Serial Operations)	DAB	ZF
Additions	$\mathcal{O}(MKd_l^2L + M^2Kd_lL + MK^2d_lL)$	$\mathcal{O}(d_l^2L + Md_lL + Kd_lL)$	$\mathcal{O}(PIM^4K^2 + PIM^3K^3)$	$\mathcal{O}(K^3 + K^2M)$
Multiplications	$\mathcal{O}(MKd_l^2L)$	$\mathcal{O}(d_l^2L)$	$\mathcal{O}(PIM^4K^2 + PIM^3K^3)$	$\mathcal{O}(K^3 + K^2M)$
FLOPs	$\mathcal{O}(MKd_l^2L + M^2Kd_lL + MK^2d_lL)$	$\mathcal{O}(d_l^2L + Md_lL + Kd_lL)$	$\mathcal{O}(PIM^4K^2 + PIM^3K^3)$	$\mathcal{O}(K^3 + K^2M)$
Example (\sim FLOPs)	$51 \cdot 10^6$	$20 \cdot 10^4$	$14 \cdot 10^{12}$	1000

rather than floating point arithmetic, which leads to even faster inference times [41]. Exploration of these avenues can lead to faster and more efficient execution of the GNN.

V. SIMULATION RESULTS

For the following simulations, the polynomial coefficients are obtained by a least squares regression of an 11th order model to the modified Rapp model as described in Section II-B. This regression is performed at the desired IBO, the polynomial coefficients corresponding to the IBO value can be found in Appendix A. For some specific cases a 3th order PA model at an IBO = -3 dB is considered, the PA parameters for this model are⁶ $\beta_1 = 1$, $\beta_3 = -77.82 - 40.12j \cdot 10^{-3}$. For all simulations, the total transmit power is $P_T = M$. Hence, the average power at the input of each PA is $p_{\text{in}} = P_T/M = 1$. The linear PA gain is set to one. Training and testing are done at an IBO of -3 dB, unless specified otherwise, this saturates the PAs significantly more than current cellular systems that require 9-12 dB back-off. After training, the NNs are evaluated based on the sum rate given in (14) where the SNIDR is computed numerically according to (15). The training set consists of 500 000 generated channels. These are generally sampled from a complex normal distribution with zero mean and variance one $[\mathbf{H}]_{i,j} \sim \mathcal{CN}(0, 1)$ if a Rayleigh fading channel is assumed. When considering radiation patterns, a deterministic LOS channel is generated according to (12), where the user angle (in degrees) is randomly sampled from a discrete uniform distribution $\theta_k \sim \mathcal{U}\{0, 180\}$. The hyperparameters of the neural networks are selected by using a validation set of size 2000, while for the simulations performed in this section, an independent test set of size 10 000 is used. For training, a batch size of 64 is used, with an initial learning rate of 5×10^{-3} , which is reduced if the validation loss reaches a plateau. The network is trained for 50 epochs with early stopping if the validation loss does not further decrease.

A. Single-User Case

In this section the performance of the GNN as described in Section III-B is evaluated, for the single-user case $K = 1$. This GNN takes as input the channel matrix \mathbf{H} and outputs the precoding matrix $\hat{\mathbf{W}}$. For the evaluations performed in this section, the GNN consists of 8 layers (i.e., $L = 8$), with each hidden layer having $d_l = 128$ features. Each GNN-layer is described by (32). Where the non-linear activation function in each layer is a leaky rectified linear unit (LReLU), except in the last GNN-layer where a linear activation is used. After this linear activation, the power normalization layer as described in (37) is performed. During training, P_T/σ_v^2 is set to 20 dB.

⁶Obtained using a least squares regression of the 3th order PA model to the modified Rapp model at an IBO = -3 dB.

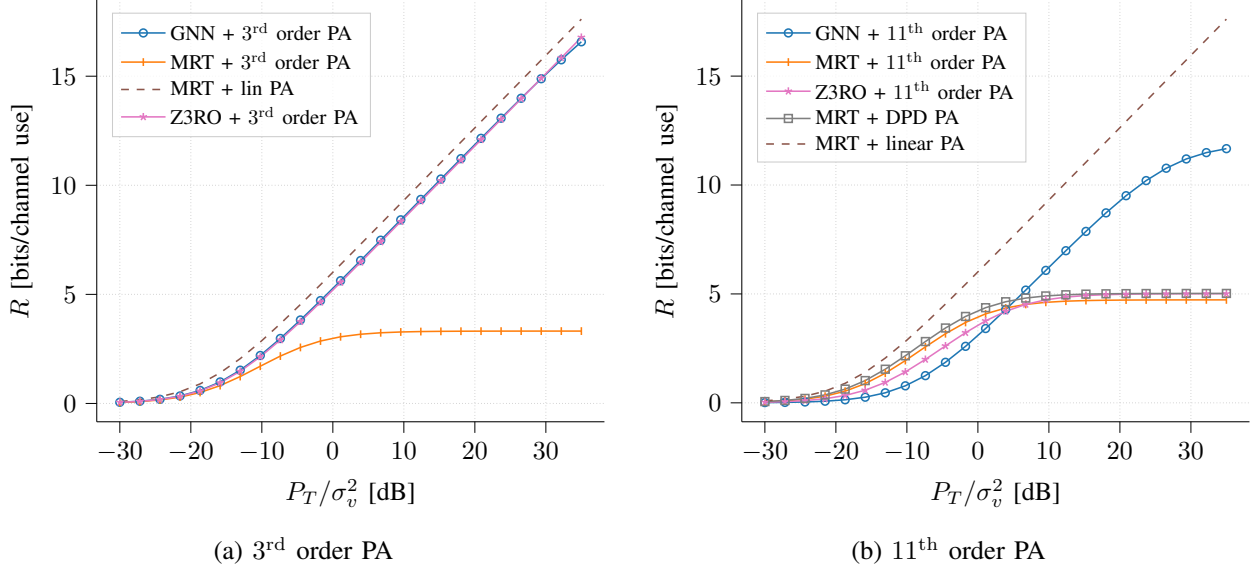


Fig. 5: Achievable rates averaged over the channel realizations taken from the test set evaluated on a 3rd order PA (a) and an 11th order PA (b). Comparing the GNN precoder against MRT, MRT plus a perfect DPD and the Z3RO precoder from [22] for $M = 64$ and $K = 1$. Observations: (a) GNN closely matches the optimal Z3RO precoder. (b) GNN copes better with high-order effects in the PA.

For the single-user case when one assumes the simplified third-order polynomial PA model (i.e., $N = 1$ in (7)) a closed-form solution exists, namely the Z3RO precoder. Therefore, first it is investigated whether the GNN precoder can achieve the same performance as the optimal Z3RO precoder under these assumptions. In Fig. 5a the achievable rates of the GNN, Z3RO and MRT precoder are depicted as a function of P_T/σ_v^2 . The MRT precoder is designed under the assumption of a linear PA. As a consequence, when a non-linear PA is present, the user performance degrades as compared to the linear case, as can be seen in Fig. 5a. Additionally, it can be seen in Fig. 5a that the rate of the Z3RO precoder is not limited by distortion but grows logarithmically with P_T/σ_v^2 . Indeed, the Z3RO precoder is able to mitigate the third-order distortion in the user direction by saturating one (or a few) antennas with an opposite phase shift, which comes at the cost of a small reduction in array gain. When comparing the GNN precoder, trained on a third-order PA model, against the Z3RO precoder we see that the GNN achieves similar performance as the Z3RO precoder. Indeed, when comparing the amplitude and phase of both precoders in Fig. 6a, it is clear that the GNN also saturates one or a few of the antennas with an opposite phase shift. In conclusion, for $K = 1$, the GNN has learned a similar precoding structure as the optimal Z3RO precoder.

Second, the case of a high-order PA model is studied, namely an eleventh order PA ($N = 5$). In this case the Z3RO precoder is no longer optimal as can be seen in Fig. 5b. Under this PA model the rate of the Z3RO precoder is limited by distortion as P_T/σ_v^2 grows, this is also the case for the MRT precoder even when it is combined with an ideal DPD. When comparing against the GNN precoder, trained on an eleventh order PA, Fig. 5b illustrates how the GNN drastically outperforms MRT, Z3RO precoding and even MRT coupled with a perfect DPD. The GNN

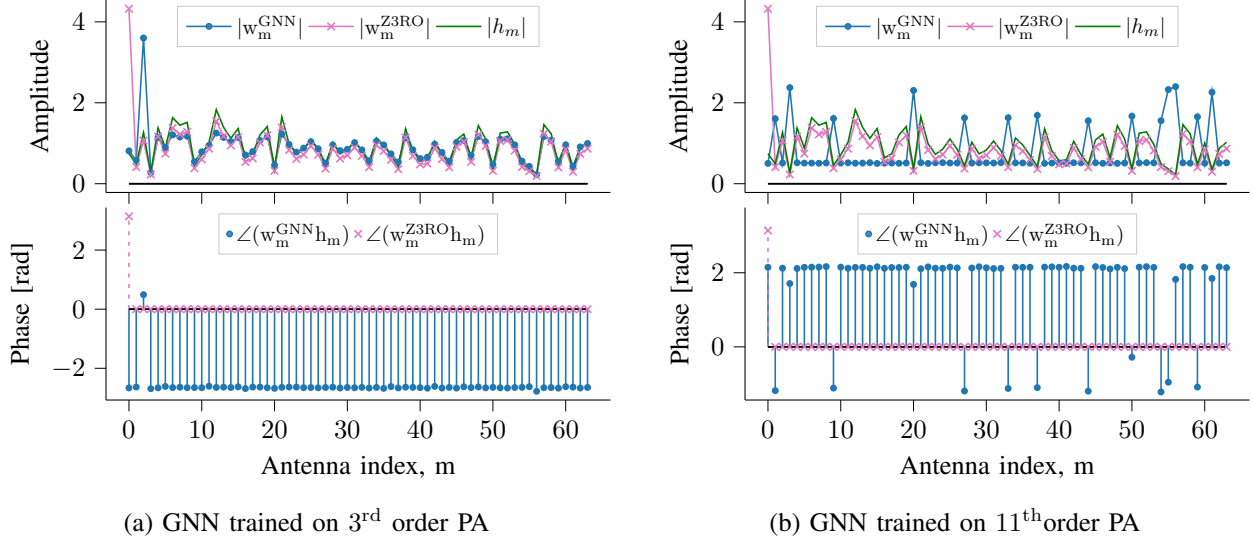


Fig. 6: Amplitude and phase plot per antenna, GNN compared to the Z3RO precoder from [22]. When trained on the third-order model (a), the GNN replicates the behavior of the optimal Z3RO precoder i.e., saturate one antenna with an opposite phase shift to cancel the third order distortion. When trained on an eleventh order PA (b), the GNN saturates more antennas in order to spatially suppress the high-order distortion.

provides a rate which nearly grows logarithmically with P_T/σ_v^2 and is only limited by distortion for very large P_T/σ_v^2 . Note that the GNN precoder only outperforms the benchmarks for large P_T/σ_v^2 i.e., when the system is mostly corrupted by distortion and not by noise.

When looking at the amplitude and phase of the GNN precoder trained on the 11th order PA in Fig. 6b it is clear that it no longer behaves as the Z3RO precoder (i.e., perform MRT precoding with one or a few saturated antennas with opposite phase shift to cancel distortion). Rather, it keeps the amplitude of nearly all transmit antennas constant except for a number of antennas which are saturated with a different phase shift. This avoids sending all distortion in the user direction as can be seen in the radiation patterns in Fig. 7. In Fig. 7a, the radiation pattern of the intended and distortion signal are depicted for the MRT precoder, here it is clear that the distortion is beamformed in the same direction as the intended signal, i.e., the user direction. Figure 7b, depicts the radiation patterns of the intended and distorted signals for the proposed GNN precoder. This figure illustrates that the distortion is more uniformly distributed for the GNN precoder and even close to zero in the user direction. The cost of this dispersed distortion is a reduction in array gain in the user direction. However, as Fig. 7c illustrates, the SDR radiation pattern for the GNN precoder has a clear peak in the user direction, while it is uniform for the MRT precoder.

B. Multi-User Case

In this section, the same GNN model as in the previous section is used, however, this time the multi-user case is considered. When multiple users are present, both the inter-user interference and distortion to multiple users have to be mitigated. In this more complex scenario, there is no closed-form solution available for the optimal precoder. As

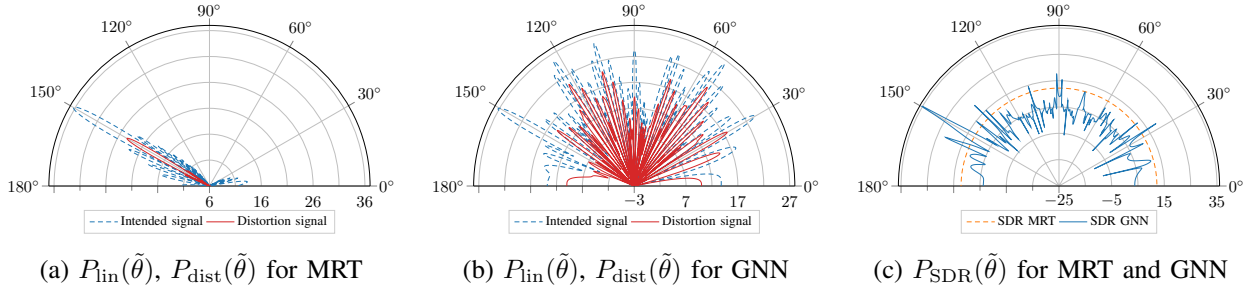


Fig. 7: Radiation pattern of the intended signal $P_{\text{lin}}(\tilde{\theta})$ and distortion signal $P_{\text{dist}}(\tilde{\theta})$ [dB] for an 11th order PA, $K = 1$ of the GNN precoder (a) and ZF precoder (b) for a pure LOS channel and a half-wavelength ULA. User angle is $\theta = 150^\circ$. In (c) the signal-to-distortion ratio (SDR) radiation pattern $P_{\text{SDR}}(\tilde{\theta})$ is depicted [dB].

such, the GNN is trained to learn how to perform this task. Additionally, the DAB precoder from [25] is considered, as it provides a way to solve the non-convex optimization problem, without any guarantees on optimality. Note that in [25] the optimization procedure is repeated for $P = 50$ different initializations and executed for $I = 100$ iterations per initialization. In [25] this procedure is reported to converge after $I = 100$ iterations. However, given the added complexity of our scenario (i.e., 64 vs. 16 Tx antennas and an 11th-order vs. a 5th-order PA) our simulations did not converge after $I = 100$ iterations. As such, the simulation results for the DAB precoder in this study are obtained by running the optimization procedure for $I = 1000$ iterations, repeated for $P = 50$ different initializations. In order to compensate for this added simulation time the DAB precoder is only evaluated on 10 channel realizations taken from the test set, while the GNN is evaluated on the full test set⁷.

In Fig. 8a, a comparison between the GNN, DAB, ZF and ZF precoder coupled with a perfect DPD is made. In this figure, the sum rate is depicted in function of P_T/σ_v^2 for $K \in \{2, 4\}$, when using the eleventh order PA model. Comparing the GNN precoder against the ZF precoder, for $K = 2$ the sum rate is increased by 8.60 bits/channel use at $P_T/\sigma_v^2 = 30$ dB, when using the proposed method, and 8.84 bits/channel use for $K = 4$. The GNN precoder shows superior performance over the DAB precoder. Next to this, it is shown that the proposed solution outperforms a perfect DPD combined with ZF, when the system is highly distortion limited (i.e., for high P_T/σ_v^2 levels). This shows the ability of the GNN to cancel high-order non-linear distortion in the multi-user scenario, which results in significant increases in capacity. Additionally, this illustrates that the higher the number of users becomes, the less gain is to be obtained, in terms of added rate per user, by using the GNN precoder. This is expected as non-linear distortion is more spatially spread out when more users are present [42]. In other words, less distortion is beamformed in the user directions, which leads to less potential gains for mitigating this distortion. Moreover, when more users are present, canceling all distortion to all users becomes more complex. Nevertheless, the GNN-based precoder achieves a significant increase in achievable sum rate as compared to the benchmark approaches.

⁷The results when evaluating the GNN on the same 10 channel realizations as the ones used to evaluate the DAB precoder match the results obtained when evaluating the GNN on the full test set.

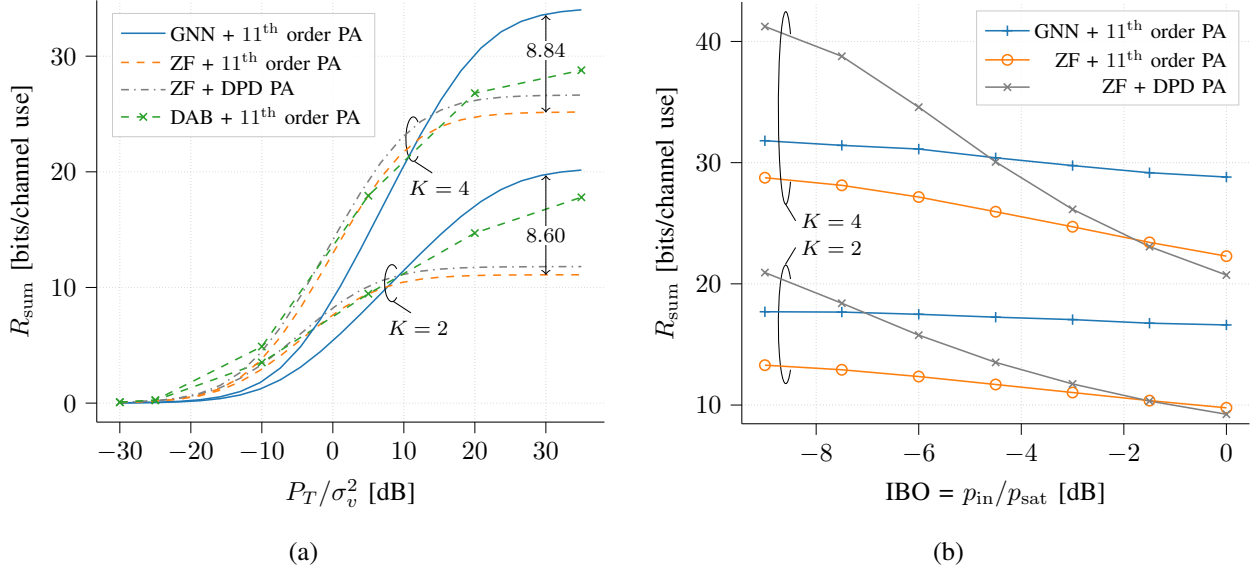


Fig. 8: Achievable rates averaged over the channel realizations taken from the test set evaluated on an 11th order PA. Comparing the GNN precoder against ZF and the DAB precoder from [25] for $M = 64$ and $K \in \{2, 4\}$ at $\text{IBO} = -3$ dB in (a). Note that the DAB precoder was only evaluated on 10 channel realizations to reduce simulation time. In (b) the achievable sum rates are plotted for $M = 64$, $K \in \{2, 4\}$, $P_T/\sigma_v^2 = 20$ dB with varied IBO. Comparing the GNN, ZF and ZF plus DPD. The GNN is retrained at each IBO point.

Figure 8b depicts the sum rate when p_{in} is fixed but p_{sat} is varied, resulting in a varied IBO. This shows that, for $K \in \{2, 4\}$, the GNN always outperforms the classical ZF precoder, when evaluated using the eleventh order PA model. Moreover, when $K = 2$, the GNN is able to achieve a nearly constant sum rate over a wide range of IBO. This illustrates the ability of the GNN to suppress nearly all high-order distortion in the user directions. When comparing the GNN to ZF plus a perfect DPD in Fig. 8b, it is evident that the GNN is most beneficial when a lot of distortion is present, i.e., at low back-off. However, we stress the fact that the GNN does not have to be used as a replacement, but could be used in combination with DPD. When (perfect) DPD is available, the PA characteristic after applying DPD can be modeled as a polynomial on which the GNN can be retrained. As such, the combination of both approaches could produce even better sum rates. However, this would add a significant complexity burden.

Finally, in Figs. 9 and 10 the radiation patterns for $K = 2, 4$ are depicted. In Figs. 9a and 9b, the radiation patterns of the intended and distorted signals are plotted for the ZF and GNN precoder respectively, for $K = 2$. Figure 9a clearly illustrates how for ZF precoding the distortion is beamformed in both user directions. Figure 9b reveals that for GNN precoding the distortion is mainly beamformed in the non-user directions, as the distortion in the user directions is very small, this however comes at the cost of a reduction in array gain for the linearly amplified signal. Nonetheless, in Fig. 9c the SDR radiation pattern is presented, which shows two clear peaks in the user directions, for the GNN precoder, while the SDR for the ZF precoder is nearly uniformly distributed. In Figs. 10a and 10b, the radiation patterns of the intended and distorted signals are plotted for the ZF and GNN precoder respectively,

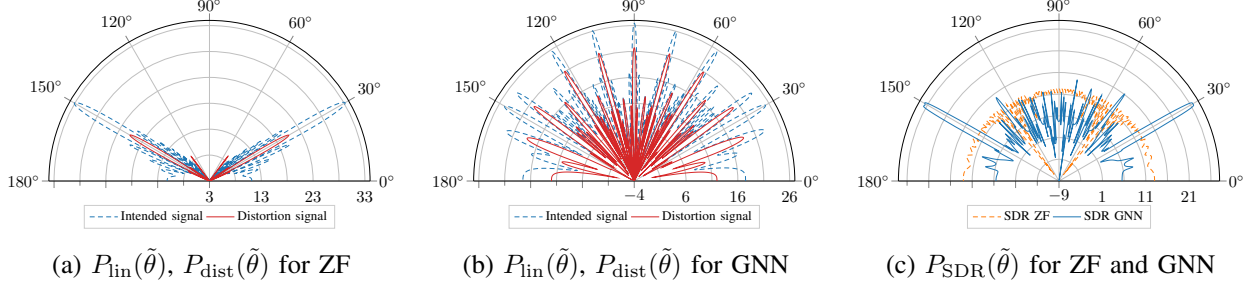


Fig. 9: Radiation pattern of the intended signal $P_{\text{lin}}(\tilde{\theta})$ and distortion signal $P_{\text{dist}}(\tilde{\theta})$ [dB] for an 11th order PA, $K = 2$ of the GNN precoder (a) and ZF precoder (b) for a pure LOS channel and a half-wavelength ULA. User angles are $\theta \in \{30^\circ, 150^\circ\}$. In (c) the signal-to-distortion ratio (SDR) radiation pattern $P_{\text{SDR}}(\tilde{\theta})$ is depicted [dB].

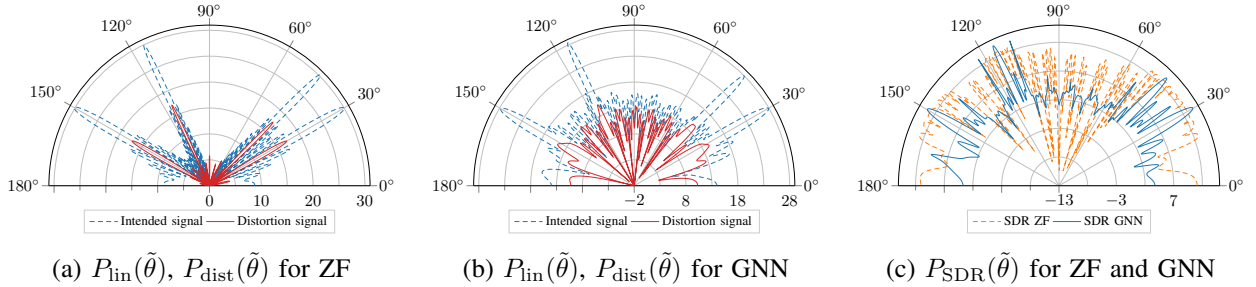


Fig. 10: Radiation pattern of the intended and distortion signal $P_{\text{lin}}(\tilde{\theta}), P_{\text{dist}}(\tilde{\theta})$ [dB] for an 11th order PA, $K = 4$ of the GNN (a) and ZF precoder (b) for a pure LOS channel and a half-wavelength ULA. User angles are $\theta \in \{30^\circ, 45^\circ, 115^\circ, 150^\circ\}$. In (c) the signal-to-distortion ratio (SDR) radiation pattern $P_{\text{SDR}}(\tilde{\theta})$ is depicted [dB].

for $K = 4$ users. For the ZF precoder, the distortion is again beamformed in the user directions while it is more spatially spread out for the GNN precoder. When looking at the SDR radiation pattern in Fig. 10c, we see a slight improvement of the SDR in the user directions for the GNN precoder as compared to the ZF precoder.

C. Improvements in Power Consumption

1) *PA Power Consumption*: By spatially suppressing the non-linear distortion, a lower IBO can be used to achieve a desired sum rate. By allowing the PAs to work closer to saturation, their energy efficiency is improved. In this section the improvements in PA power consumption are quantified. For a class B amplifier the following model of the energy efficiency can be used [43]. The efficiency of PA m is defined as

$$\eta_m = \eta_{\text{max}} \sqrt{\frac{p_m}{p_{\text{sat}}}}. \quad (42)$$

Here p_{sat} is the saturation power of the PA, while η_{max} is the maximum achievable efficiency, for a class B amplifier $\eta_{\text{max}} = 0.785$ [43]. Next to this, p_m is defined as the power at the output of amplifier m

$$p_m = \mathbb{E}(|y_m|^2) = \mathbb{E}(|\phi(x_m)|^2) = \mathbb{E}\left(\left|\phi\left(\sum_{k=0}^{K-1} w_{m,k}s_k\right)\right|^2\right). \quad (43)$$

The total consumed PA power is then given by

$$p_{\text{cons,PA}} = \sum_{m=0}^{M-1} \frac{p_m}{\eta_m} = \frac{\sqrt{p_{\text{sat}}}}{\eta_{\text{max}}} \sum_{m=0}^{M-1} \sqrt{\mathbb{E}(|y_m|^2)}. \quad (44)$$

From this expression, it can be seen that the consumed power is not proportional to the average input back-off but rather to the sum of the square root of the output power of the PAs. In Fig. 11a the consumed power of the PAs in function of the input back-off is depicted for $K = 2$ and $K = 4$. This figure illustrates that the GNN has a lower PA consumed power as compared to the ZF and ZF + DPD precoding, for a same average input back-off (and thus a same transmit power). Intuitively, this lower consumption is due to the fact that the GNN saturates a number of antennas. These saturated antennas have a very high efficiency, while they also receive more power as compared to the less efficient antennas.

Moreover, for a fair comparison one needs to select a quality of service constraint, e.g., a sum rate constraint, and compare the consumed power for the different precoders meeting this rate constraint. As an example for $M = 64$, $K = 4$ let us consider a desired sum rate of 30 bits/channel use. From Fig. 11b the consumed power to achieve this rate can be seen, for GNN precoding this is $p_{\text{cons,PA}} = 85.57$ W, for ZF precoding $p_{\text{cons,PA}} = 284.29$ W and for ZF precoding + DPD this is $p_{\text{cons,PA}} = 125.98$ W.⁸ This reveals that the GNN precoder has a PA power consumption that is 3.32 times smaller as compared to ZF precoding and 1.47 times smaller as compared to ZF precoding plus DPD.

2) *Digital Signal Processing Power Consumption:* In this section the consumed power of the hardware accelerator needed to run the GNN is quantified. Considering the scenario were $M = 64$, $K = 4$, based on Section IV it is clear that 164 MFLOPs are required to compute one forward pass of the GNN. In order to establish the desired speed (FLOPs/s) of the hardware accelerator, the coherence time of the channel is computed to determine the time available to compute the precoding coefficients. The coherence time is proportional to the inverse of the maximum Doppler frequency $f_m = \frac{v}{c}f_c$ and is often defined as $T_c = \frac{1}{2f_m}$ [33]. Assuming a carrier frequency of $f_c = 5$ GHz and a receiver velocity of $v = 10$ m/s, this gives a coherence time of 3 ms. As a rule of thumb it is assumed that 10% of the coherence time can be used to compute the precoding coefficients. This establishes a required hardware accelerator speed of 549 giga floating point operations per second (GFLOPs/s). Based on [38] the state-of-the-art Manticore accelerator [44] is selected, which can achieve the desired speed and precision.⁹ The full Manticore design consists out of 4096 cores, however in [44] a prototype with 24 cores was developed and measured. This prototype is able to achieve 50 GFLOPs/s with a power consumption of 200 mW. Given the parallel architecture of the GNN it can be deployed across multiple cores. Hence in order to reach the desired speed of 549 GFLOPs/s

⁸Obtained using linear interpolation between the points of Fig. 11b.

⁹Note that the GNN considered in this work requires single precision (32 bit) floating point operations.

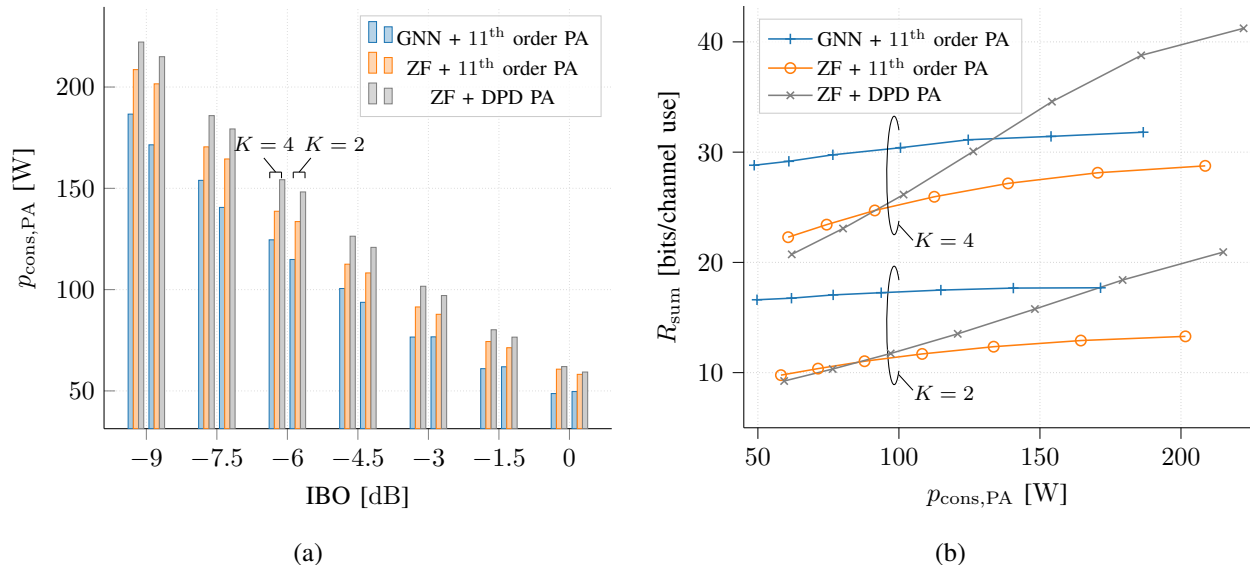


Fig. 11: (a) Consumed PA power in function of IBO (b) achievable sum rate in function of consumed PA power. Both (a) and (b) compare the GNN, ZF and ZF + DPD precoders for $M = 64$, $K \in \{2, 4\}$, $\frac{P_T}{\sigma_z^2} = 20$ dB.

264 cores can be utilized rather than 24, this leads to a total digital signal processing (DSP) power consumption of 2.2 W.

3) *Total Power Consumption:* An overview of the total consumed power (PA and processing power) for the GNN can be seen in Table II. From this table it can be seen that the total consumed power for GNN precoding is 87.77 W, for ZF precoding 284.29 W and for ZF + DPD 125.98 W. Note that as a worst-case scenario the processing power of ZF and DPD is assumed to be zero, while in reality this processing power will be substantial when DPD is implemented. Even in this scenario, the GNN precoder consumes 3.24 times less power as compared to ZF and 1.44 times less power as compared to ZF+DPD.

D. Validation on Rapp PA Model

In this section, the GNN precoder which is trained on the eleventh order PA model is evaluated on the modified Rapp model. The modified Rapp model is originally used to obtain the coefficients of the polynomial model by

TABLE II: Consumed power of the GNN, ZF and ZF + DPD precoders for $M = 64$, $K = 4$, for a fixed data rate of 30 bits/channel use. As a worst-case scenario the processing power for ZF and DPD are assumed to be zero.

Precoder	Precision	Accelerator power [W]	$p_{\text{PA,cons}}$ [W]	total power [W]
GNN + Manticore [44]	single precision (SP)	2.2	85.57	87.77
ZF + DPD	NA	NA	125.98	125.98
ZF	NA	NA	284.29	284.29

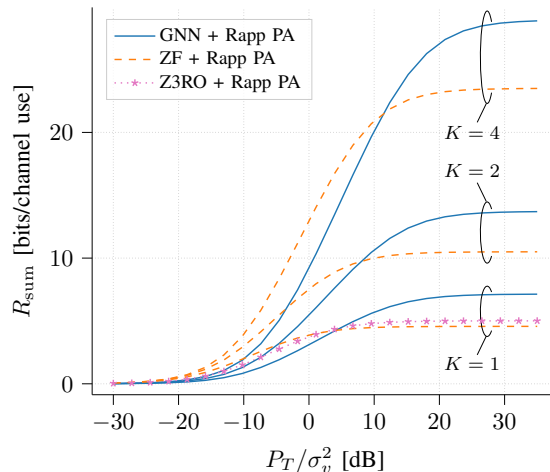


Fig. 12: Achievable sum rates averaged over 1000 Rayleigh channel realizations taken from the test set evaluated using the Rapp PA model (9, 10). $M = 64$, $K \in \{1, 2, 4\}$, with a varied P_T/σ_v^2 .

performing a least squares regression, as discussed in Section II-B. By evaluating the GNN on the modified Rapp model, the effects of using a polynomial model, rather than the actual PA characteristic can be studied. In Fig. 12 the rate is depicted in function of P_T/σ_v^2 when the modified Rapp model is used for the amplification stage. For the single-user case $K = 1$, this figure illustrates how the GNN precoder trained on a polynomial PA still outperforms the MRT and Z3RO precoders when the underlying modified Rapp model is assumed. For the multi-user case $K \in \{2, 4\}$ in Fig. 12, it is again clear that the GNN precoder outperforms the ZF precoder when the modified Rapp model is deployed as the amplification model. This highlights the fact that the polynomial model can successfully be used to train the GNN even if the underlying PA characteristic is slightly different.

E. Leveraging SNR Regime Information

In this section the performance of the GNN as described in Section III-C is evaluated. This GNN takes as input edge features the channel coefficients and the normalized SNR, and is denoted as GNN_{SNR} . GNN_{SNR} is compared against GNN_{base} which only takes the channel matrix as input. The GNN architecture is kept the same as in Section III-B except for the case where $K = 4$ where $d_l = 256$ features per layer are used rather than 128.

In Fig. 13a and 13b the sum rate is depicted in function of P_T/σ_v^2 for $K \in \{1, 2, 4\}$, for the eleventh order PA model. This figure illustrates the ability of GNN_{SNR} to leverage the extra SNR input to deliver performance over the full range of P_T/σ_v^2 . This alleviates the need for retraining the GNN at each value of P_T/σ_v^2 . In this way, GNN_{SNR} not only performs well in the distortion limited regime (i.e., for high P_T/σ_v^2) but is also capable of delivering the same performance as ZF in the noise limited regime (i.e., for low P_T/σ_v^2), which GNN_{base} was not able to do.

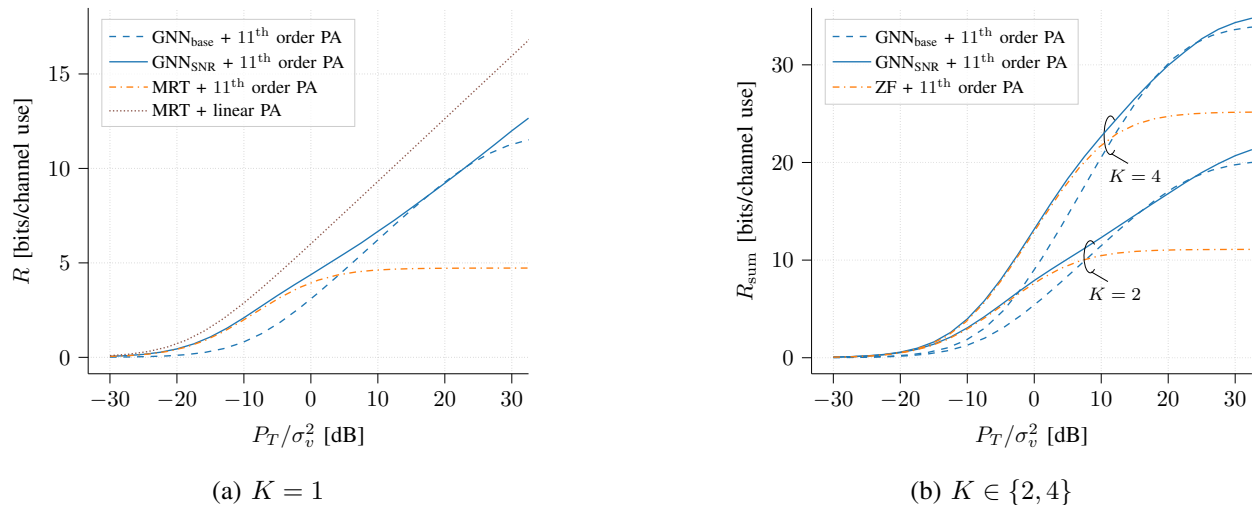


Fig. 13: Achievable rates for $M = 64$, $K \in \{1, 2, 4\}$ and an 11th order PA. Comparing the base GNN from Section III-B against the GNN that leverages SNR information from Section III-C.

VI. CONCLUSION

In this study, a mMIMO system is considered, where a GNN is trained to learn a mapping between the channel and precoding matrix which maximizes the sum rate in the presence of high-order non-linear PA distortion. The complexity of the proposed solution is analysed and shown to be six orders of magnitude lower as compared to the benchmark DAB precoder. Simulation results indicate that the proposed solution offers an increased capacity as compared to ZF in the distortion-limited regime, i.e., an energy-efficient regime. For instance, at an IBO of -3 dB the proposed precoder produces an increase in sum rate of 8.60 and 8.84 bits/channel use for the two and four user cases respectively. Additionally, it is shown that the proposed solution achieves a higher sum rate as compared to ZF precoding plus DPD, as DPD is limited by the distortion caused by saturation based clipping. Radiation patterns demonstrate that the higher rate is achieved by spatially suppressing the distortion in the user-directions. By utilizing the proposed precoder, PAs can operate closer to saturation, which increases their energy efficiency. For instance, it is shown that, in the four user-case, for a fixed sum rate the PA consumed power of the proposed precoder is 3.32 times and 1.47 times lower as compared to ZF precoding and ZF plus DPD respectively. In addition, it is shown that the PA power consumption reduction is achieved at the price of a small increase in DSP consumption so that the total power consumption is still greatly reduced. Furthermore, the GNN is extended to incorporate additional knowledge, i.e., the SNR regime, which alleviates the need to retrain at each SNR point. These results are especially promising in the multi-user case where no closed-form/low-complexity solution to the distortion-aware precoding problem is available.

Future perspectives include the incorporation of additional knowledge into the precoder design, such as the PA parameters, which would allow the precoder to anticipate long-term changes in the PA characteristics. Additionally, more complex PA models could be investigated that capture memory effects. Finally, the impact of channel estimation

and PA parameter estimation errors on the performance of the proposed method should be evaluated in future studies.

REFERENCES

- [1] L. Belkhir and A. Elmeligi, "Assessing ICT global emissions footprint: Trends to 2040 & recommendations," *Journal of Cleaner Production*, vol. 177, pp. 448–463, Mar. 2018.
- [2] C. Freitag, M. Berners-Lee, K. Widdicks, B. Knowles, G. S. Blair, and A. Friday, "The real climate and transformative impact of ICT: A critique of estimates, trends, and regulations," *Patterns*, vol. 2, no. 9, p. 100340, 2021.
- [3] European Commission, "The European Green Deal," *COM (2019)*, November 2019.
- [4] United Nations, "The 2030 Agenda and the Sustainable Development Goals: An opportunity for Latin America and the Caribbean."
- [5] United Nations Environment Programme, "Paris Agreement," 12/12/2015. [Online]. Available: <https://wedocs.unep.org/20.500.11822/20830>
- [6] "Ericsson Mobility report November 2022," <https://www.ericsson.com/4ae28d/assets/local/reports-papers/mobility-report/documents/2022/ericsson-mobility-report-november-2022.pdf>.
- [7] L. M. P. Larsen, H. L. Christiansen, S. Ruepp, and M. S. Berger, "Toward Greener 5G and Beyond Radio Access Networks—A Survey," *IEEE Open Journal of the Communications Society*, vol. 4, pp. 768–797, 2023.
- [8] C. Andersson, J. Bengtsson, G. Byström, P. Frenger, Y. Jading, and M. Nordenström, "Improving energy performance in 5G networks and beyond," *Ericsson Technology Review*, vol. 2022, no. 8, pp. 2–11, 2022.
- [9] G. Auer, V. Giannini, C. Desset, I. Godor, P. Skillermark, M. Olsson, M. A. Imran, D. Sabella, M. J. Gonzalez, O. Blume, and A. Fehske, "How much energy is needed to run a wireless network?" *IEEE wireless communications*, vol. 18, no. 5, pp. 40–49, 2011.
- [10] C. Gabriel, A. Hansang, and A. Chern, "Green 5G: Building A Sustainable World," *Huawei Technologies Co.*, 2020. [Online]. Available: <https://www.huawei.com/en/public-policy/green-5g-building-a-sustainable-world>
- [11] T. Schenk, *RF Imperfections in High-rate Wireless Systems: Impact and Digital Compensation*. Dordrecht: Springer Netherlands, 2008.
- [12] N. N. Moghadam, G. Fodor, M. Bengtsson, and D. J. Love, "On the Energy Efficiency of MIMO Hybrid Beamforming for Millimeter-Wave Systems With Nonlinear Power Amplifiers," *IEEE Transactions on Wireless Communications*, vol. 17, no. 11, pp. 7208–7221, 2018.
- [13] T. Eriksson, W. Cao, and C. Fager, *Nonlinear Effects of Wireless Transceivers*. John Wiley & Sons, Ltd, 2019, pp. 1–30. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119471509.w5GRef009>
- [14] J. Zanen, E. Klumperink, and B. Nauta, "Power Efficiency Model for MIMO Transmitters Including Memory Polynomial Digital Predistortion," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 4, pp. 1183–1187, 2021.
- [15] P. P. Ann and R. Jose, "Comparison of PAPR reduction techniques in OFDM systems," in *2016 International Conference on Communication and Electronics Systems (ICCES)*, 2016, pp. 1–5.
- [16] S. K. Mohammed and E. G. Larsson, "Constant-Envelope Multi-User Precoding for Frequency-Selective Massive MIMO Systems," *IEEE Wireless Communications Letters*, vol. 2, no. 5, pp. 547–550, 2013.
- [17] C. Tarver, A. Balasoukas-Slimining, C. Studer, and J. R. Cavallaro, "Virtual DPD Neural Network Predistortion for OFDM-based MU-Massive MIMO," in *2021 55th Asilomar Conference on Signals, Systems, and Computers*, 2021, pp. 376–380.
- [18] X. Cheng, R. Zayani, M. Ferecatu, and N. Audebert, "Efficient Autoprecoder-based deep learning for massive MU-MIMO Downlink under PA Non-Linearities," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, 2022, pp. 1039–1044.
- [19] J. Jee, G. Kwon, and H. Park, "Joint Precoding and Power Allocation for Multiuser MIMO System With Nonlinear Power Amplifiers," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 8883–8897, 2021.
- [20] B. Liu, F. Rottenberg, and S. Pollin, "Power Allocation for Distributed Massive LoS MIMO with Nonlinear Power Amplifiers," in *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*, 2022, pp. 1–5.
- [21] J. Jee, G. Kwon, and H. Park, "Cooperative Beamforming With Nonlinear Power Amplifiers: A Deep Learning Approach for Distributed Networks," *IEEE Transactions on Vehicular Technology*, pp. 1–16, 2022.
- [22] F. Rottenberg, G. Callebaut, and L. Van der Perre, "Z3RO Precoder Canceling Nonlinear Power Amplifier Distortion in Large Array Systems," in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 432–437.
- [23] —, "The Z3RO Family of Precoders Cancelling Nonlinear Power Amplification Distortion in Large Array Systems," *IEEE Transactions on Wireless Communications*, vol. 22, no. 3, pp. 2036–2047, 2023.
- [24] T. Feys, G. Callebaut, L. Van der Perre, and F. Rottenberg, "Measurement-Based Validation of Z3RO Precoder to Prevent Nonlinear Amplifier Distortion in Massive MIMO Systems," in *2022 IEEE 95th Vehicular Technology Conference (VTC2022-Spring)*, 2022, pp. 1–5.

- [25] S. R. Aghdam, S. Jacobsson, and T. Eriksson, "Distortion-Aware Linear Precoding for Millimeter-Wave Multiuser MISO Downlink," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2019, pp. 1–6.
- [26] T. Feys, X. Mestre, and F. Rottenberg, "Self-Supervised Learning of Linear Precoders under Non-Linear PA Distortion for Energy-Efficient Massive MIMO Systems," in *ICC 2023 - IEEE International Conference on Communications*, 2023.
- [27] T. Hwang, C. Yang, G. Wu, S. Li, and G. Ye Li, "OFDM and Its Wireless Applications: A Survey," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 4, pp. 1673–1694, 2009.
- [28] C.-S. Choi *et al.*, "RF impairment models for 60GHz-band SYS/PHY simulation," *Project: IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)*, p. 17, 2006.
- [29] Nokia, "Realistic power amplifier model for the New Radio evaluation," *3GPP TSG-RAN WG4 Meeting 79, R4-163314*, May 2016.
- [30] O. T. Demir and E. Bjornson, "The Busgang Decomposition of Nonlinear Systems: Basic Theory and MIMO Extensions [Lecture Notes]," *IEEE Signal Processing Magazine*, vol. 38, no. 1, pp. 131–136, 2021.
- [31] N. N. Moghadam, G. Fodor, M. Bengtsson, and D. J. Love, "On the Energy Efficiency of MIMO Hybrid Beamforming for Millimeter-Wave Systems With Nonlinear Power Amplifiers," *IEEE Transactions on Wireless Communications*, vol. 17, no. 11, pp. 7208–7221, 2018.
- [32] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2014. [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [33] T. L. Marzetta, E. G. Larsson, H. Yang, and H. Q. Ngo, *Fundamentals of Massive MIMO*. Cambridge University Press, 2016.
- [34] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [35] P. W. Battaglia *et al.*, "Relational inductive biases, deep learning, and graph networks," *CoRR*, vol. abs/1806.01261, 2018. [Online]. Available: <http://arxiv.org/abs/1806.01261>
- [36] B. Zhao, J. Guo, and C. Yang, "Learning Precoding Policy: CNN or GNN?" in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, 2022, pp. 1027–1032.
- [37] W. L. Hamilton, "Graph Representation Learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 14, no. 3, pp. 1–159.
- [38] C. Silvano, D. Ielmini, F. Ferrandi, L. Fiorin, S. Curzel, L. Benini, F. Conti, A. Garofalo, C. Zambelli, E. Calore, S. F. Schifano, M. Palesi, G. Ascia, D. Patti, S. Perri, N. Petra, D. D. Caro, L. Lavagno, T. Urso, V. Cardellini, G. C. Cardarilli, and R. Birke, "A Survey on Deep Learning Hardware Accelerators for Heterogeneous HPC Platforms," 2023. [Online]. Available: <https://arxiv.org/abs/2306.15552>
- [39] F. Piazza, A. Uncini, and M. Zenobi, "Neural networks with digital LUT activation functions," in *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)*, vol. 2, 1993, pp. 1401–1404 vol.2.
- [40] R. Hunger, *Floating Point Operations in Matrix-vector Calculus*. Munich University of Technology, Inst. for Circuit Theory and Signal Processing, 2005. [Online]. Available: <https://books.google.be/books?id=EccIcgAACAAJ>
- [41] L. Baischer, M. Wess, and N. TaheriNejad, "Learning on Hardware: A Tutorial on Neural Network Accelerators and Co-Processors," 2021.
- [42] C. Mollen, U. Gustavsson, T. Eriksson, and E. G. Larsson, "Spatial Characteristics of Distortion Radiated From Antenna Arrays With Transceiver Nonlinearities," *IEEE transactions on wireless communications*, vol. 17, no. 10, pp. 6663–6679, 2018.
- [43] A. He, S. Srikanteswara, K. K. Bae, T. R. Newman, J. H. Reed, W. H. Tranter, M. Sajadieh, and M. Verhelst, "Power Consumption Minimization for MIMO Systems — A Cognitive Radio Approach," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 2, pp. 469–479, 2011.
- [44] F. Zaruba, F. Schuiki, and L. Benini, "Manticore: A 4096-Core RISC-V Chiptlet Architecture for Ultraefficient Floating-Point Computing," *IEEE Micro*, vol. 41, no. 2, pp. 36–42, Mar. 2021.

APPENDIX A
POLYNOMIAL PA PARAMETERS

TABLE III: PA parameters of the 11th order polynomial PA model at different back-off values ($\beta_1 = 1$).

IBO [dB]	$\beta_3 (\cdot 10^{-2})$	$\beta_5 (\cdot 10^{-3})$	$\beta_7 (\cdot 10^{-5})$	$\beta_9 (\cdot 10^{-7})$	$\beta_{11} (\cdot 10^{-9})$
-9	-4.38184836	1.50490437	-3.13452827	3.49967293	-1.59432984
	-10.1466832j	8.422084885j	-28.1868627j	42.06333106j	-23.1868139j
-7.5	-5.79334438	2.39315994	-5.57663136	6.65066314	-3.14808144
	-9.36769411j	7.94859107j	-26.92641291j	40.4837957j	-22.4280442j
-6	-7.50994886	3.66782506	-9.54049052	12.2703316	-6.13183499
	-8.42352484j	7.26453523j	-24.8371067j	37.5613932j	-20.8924283j
-4.5	-9.35828409	5.16172165	-14.4481282	19.4963213	-10.0752209
	-7.41305601j	6.46522185j	-22.2483069j	33.7874265j	-18.8479147j
-3	-11.1143930	6.60156653	-19.1451680	26.2822435	-13.6811147
	-6.30816977j	5.47141526j	-18.6610370j	28.0380833j	-15.4579691j
-1.5	-12.903319	8.21176444	-24.8588087	35.2215545	-18.8139985
	-5.49758824j	4.85204392j	-16.8144990j	25.6527492j	-14.3562319j
0	-14.4473655	9.58442261	-29.6362436	42.5309097	-22.9128062
	-4.67375592j	4.13617338j	-14.3570171j	21.9271142j	-12.2805850j