Multi-Objective DNN-Based Precoder for MIMO Communications

Xinliang Zhang, *Graduate Student Member, IEEE*, and Mojtaba Vaezi, *Senior Member, IEEE*

***Abstract*— This paper introduces a unified deep neural network (DNN)-based precoder for two-user multiple-input multiple-output (MIMO) networks with five objectives: data transmission, energy harvesting, simultaneous wireless information and power transfer, physical layer (PHY) security, and multicasting. First, a rotation-based precoder is developed to solve the above problems independently. Rotation-based precoding is a new precoding and power allocation scheme that beats existing solutions for PHY security and multicasting and is reliable in different antenna settings. Next, a DNN-based precoder is designed to unify the solution for all objectives. The proposed DNN concurrently learns the solutions given by conventional methods, i.e., analytical or rotation-based solutions. A binary vector is designed as an input feature to distinguish the objectives. Numerical results demonstrate that, compared to the conventional solutions, the proposed DNN-based precoder reduces on-the-fly computational complexity more than an order of magnitude while reaching near-optimal performance (99.45% of the averaged optimal solutions). The new precoder is also more robust to the variations of the numbers of antennas at the receivers.**

***Index Terms*— Deep learning, precoding, MIMO, physical layer, SWIPT, wiretap channel, energy harvesting, beamforming.**

I. Introduction

WIRELESS communication faces unprecedented challenges in terms of diverse objectives (e.g., throughput, energy efficiency, security, and delay) and emerging applications (e.g., Internet of things (IoT), wearables, drones, etc.). As a recent example, with a daily average data rate over 16.6 Gigabytes, the communication traffic for in-home data usage during the COVID-19 outbreak in March 2020 has increased 18 percent compared to the same period in 2019 [2]. Such multifaceted challenges are conventionally addressed separately in the physical layer (PHY) because it is not possible to come up with one optimal solution satisfying all of those diverse and, at times, clashing requirements and objectives. However, in practice, many of those objectives should be met simultaneously in some applications, e.g., in IoT devices with limited computational resources that need to harvest energy for their transmission). Conventional solutions may even differ only if the number of antennas at the users.

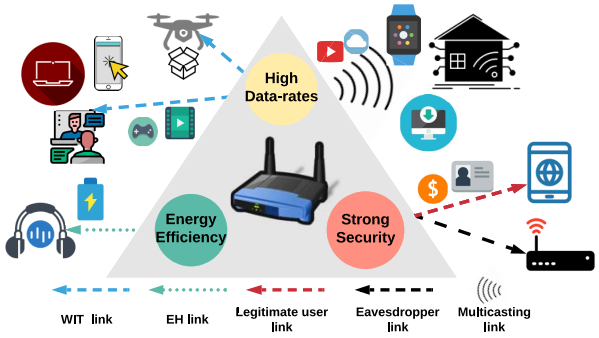


Fig. 1. A system with multiple communication services.

Motivated by the above, a streamlined system (illustrated in Fig. 1) is unified with prolific transmission functions: high data rates, strong security, and efficient energy exploitation. The integrated transmission system is required for three facets (categories of tasks) simultaneously: 1) data transmission such as wireless information transmission (WIT) and multicasting; 2) green communication, including energy harvesting (EH) and simultaneous wireless information and power transfer (SWIPT); 3) secure communication, e.g., physical layer (PHY) security. This full-featured system motivates us to consider the following question: How can we integrate all of these facets into one system with acceptable or even better performance?

To answer this question, it is enlightening to understand the current approaches to address those objectives. As an essential part of the multiple-input and multiple-output (MIMO) communication systems, *precoding* and *power allocation* schemes (or equivalently, transmit covariance matrix design) are typically used to address each of those facets independently of the others. More specifically, for each of the five objectives we mentioned (i.e., WIT, EH, SWIPT, PHY security, and multicasting), one or more independent solutions are developed in the literature (see Table I). For some objectives, such as WIT, an optimal closed-form solution is known, which is obtained via celebrated singular value decomposition (SVD) and water-filling [3]. Linear precoding and power allocation solutions for EH and SWIPT can be found in [4] and [5]. For others, such as PHY security in the MIMO wiretap channel, only sub-optimal or iterative solutions are known in general

Manuscript received July 6, 2020; revised December 19, 2020 and February 24, 2021; accepted April 2, 2021. Date of publication April 7, 2021; date of current version July 15, 2021. This article was presented in part at the IEEE Global Communications Conference, December 2020. The associate editor coordinating the review of this article and approving it for publication was A. Alkhateeb. *(Corresponding author: Mojtaba Vaezi.)*

The authors are with the Department of Electrical and Computer Engineering, Villanova University, Villanova, PA 19085 USA (e-mail: xzhang4@villanova.edu; mvaezi@villanova.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCOMM.2021.3071536.

Digital Object Identifier 10.1109/TCOMM.2021.3071536

0090-6778 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: VILLANOVA UNIVERSITY. Downloaded on July 24,2021 at 02:23:56 UTC from IEEE Xplore. Restrictions apply.

TABLE I
THE DESIRED OBJECTIVES AND EXISTING SOLUTIONS

Configuration	Objective Function	Reference
$\mathcal{O}_1$	WIT	[3]
$\mathcal{O}_2$	EH	[4]
$\mathcal{O}_3$	SWIPT	[4], [5]
$\mathcal{O}_4$	PHY Security	[6]–[9]
$\mathcal{O}_5$	Multicasting	[11], [12]

[6]–[8] as the problem is not convex. Among them, generalized singular value decomposition (GSVD)-based precoding [6] is a fast, sub-optimal solution, whereas alternating optimization with water-filling (AO-WF) [7] has better performance but requires much more time. Yet, those methods may not be close to the capacity in some antenna settings [8], [9]. The spatial correlation of the MIMO wiretap channel with EH constraints is considered in [10]. Lastly, multicasting is a min-max fair problem to enlarge the transmission rate for all users. In the multiple-input single-output (MISO) case, semidefinite relaxation (SDR) techniques yield a closed-form solution [11]. In the MIMO case, a cyclic alternating ascent (CAA) linear precoding is proposed in [12].

As listed in Table I, various different approaches are used to design precoder for the problems and, yet, some of them are not effective in all antenna settings. Among those problems, $\mathcal{O}_3$ to $\mathcal{O}_5$ are more challenging. In the first part of this paper, we apply *rotation-based precoding* (RP) to the latter three problems. This approach uses one method of solution for all those three problems.[1] More importantly, in general, it results in better performance for these objectives when compared with existing methods. RP can be applied to all of those problems to unify the optimization approach. However, the optimization problems corresponding to those objectives are still solved individually.

In the second part of this paper, we introduce a unified deep neural network (DNN)-based precoder to solve optimization problems corresponding to those five objectives ($\mathcal{O}_1$ to $\mathcal{O}_5$) at once. This will settle the question we raised earlier in this paper. The new question is how we can "teach" a DNN [13] to concurrently and effectively "learn" all objectives together? To this end, we utilize a supervised DNN to learn from the solutions given by the RP by using the backpropagation algorithm which updates internal parameters from presentation layers [13]. We introduce an input feature to distinguish the objectives, and we interpret "learn" as the act of choosing the best precoder. DNN is a good "learner" due to its sensitivity to the same types of input and output pairs, even if mathematical models/solutions for those pairs are totally different. DNN-based precoding can realize unification[2] by regulating all the input objectives with one structure.

[1]The RP can be applied to all of the five problems listed in Table I. However, RP has no advantage over the existing solutions for WIT and EH as analytical solutions are available for them.

[2] It is worth noting that this paper provides a unification of five objectives ($\mathcal{O}_1$- $\mathcal{O}_5$) by training one DNN, and will finally obtain the output for each of the objectives. The objectives are needed one at a time, not concurrently. It is different from multi-objective optimizations, such as [14], which 'jointly' optimize multiple objectives in one problem.

Before this work, DNN has separately been applied to many communications problems independently. To name a few, in [15], DNN is employed to model a Markov chain to obtain the rate-energy region of SWIPT in practical EH circuits. In [16], an autoencoder is proposed in which DNN learns the optimal mapping from the encoder to the decoder for PHY security. In [17], a DNN-based precoder for the wiretap channel is considered for specific antenna settings. We do not expect that supervised DNN to significantly surpass conventional mathematical methods in communications; nevertheless, DNN holds promise for many front-end technologies in complex scenarios [18], such as spectrum intelligence which deceptively manages the radio resource [19]–[21]; transmission intelligence which focuses on reaching channel estimation and characterization [22], [23]; network intelligence which enhances the communication quality at the system-level [24].

A. Motivation and Contribution

Apart from unifying precoder design for multi-objective systems discussed earlier, computational efficiency is another reason that motivates us to investigate a DNN-based precoder.

DNN is a universal function approximator [25] that has achieved a remarkable capacity for algorithm learning [26]. DNN is essentially a set of filters that is applied repeatedly to batches of the input. The network owns fixed times of convolution operation on weight matrices which can avoid the endless loop of iteration. In other words, DNN is able to achieve high resource utilization, e.g., matrix rather than vector operations on graph processing unit (GPU), which in turn can substantially reduce the computation costs without sacrificing accuracy. As a data-driven technique, DNN is usually arranged offline and only needs to be performed once.

In this paper, we design a unified precoder based on a DNN architecture that explores a different way of thinking for wireless communication systems. Our main contributions are:

- We introduce rotation-based precoding and power allocation for multiple objectives, including WIT, EH, SWIPT, PHY security, and multicasting. An immediate benefit of this method is that it converts the positive semidefinite (PSD) constraint of the covariance matrix into linear constraints to simplify the optimization problems. Further, this method can improve the performance of the state-of-the-art in the wiretap and multicasting.
- Rotation-based precoders are designed for SWIPT, PHY security, and multicasting, and have a more stable and even better performance than existing methods. Higher secrecy rates are achieved over MIMO wiretap channels with different numbers of antennas. Besides, it enlarges the data transmission rates for multicasting with lower computational complexity.
- Then, we propose a unified DNN-based precoder which "learns" from the conventional mathematical models, including analytical solutions or rotation-based methods. This precoding scheme can solve all objectives at the same time, since the network is trained by all the functions together. To distinguish the objectives, an input

Authorized licensed use limited to: VILLANOVA UNIVERSITY. Downloaded on July 24,2021 at 02:23:56 UTC from IEEE Xplore. Restrictions apply.

feature utilizing a binary code is designed. This feature provides a mode selection to realize the multiple functions in one DNN.

- The proposed DNN-based precoding is more efficient than the state-of-the-art iterative solutions and could be even better than the analytical solutions. Specifically, it is more than an order of magnitude faster than numerical solutions using a central processing unit (CPU). In the scenario where a GPU is affordable (like base stations), it takes an average execution time around 0.40ms and 0.043ms corresponding to 10 and 100 channels concurrently. The realization of DNN can be parallelized in GPU, so that the computational burden is comparable to that of the analytical solutions, when they exist.

B. Organizations and Notations

The remainder of this paper is organized as follows. We introduce the system models for the five objectives in Section II, and formulate rotation-based precoding in Section III. In Section IV, we propose the DNN architecture for our unified precoder. We illustrate the training process and results in Section V. Finally, we conclude the paper in Section VI.

Notations: Bold lowercase letters denote column vectors and bold uppercase letters denote matrices. $a_{i,j}$ represents the entry (i, j) of matrix $\mathbf{A}$. $\mathrm{vec}(\cdot)$ vectorizes a matrix by cascading columns. $|\cdot|$, $(\cdot)^T$, $\ln(\cdot)$, $\mathrm{tr}(\cdot)$ are the absolute value, Euclidean norm, transpose, natural logarithm, respectively. $\mathrm{diag}(\cdot)$ designates the diagonal matrix of the set inside. $\mathrm{sign}(\cdot)$ extracts the sign of a real number. $E\{\cdot\}$ is the expectation of random variables. $[x]^+$ expresses the maximum value of 0 and x. $\mathbf{I}_a$ is an $a \times a$ identical matrix, and $\mathbf{0}_{a\times b}$ ($\mathbf{1}_{a\times b}$) is all-zeros (all-ones) matrix of dimension $a \times b$.

II. System Models and Mathematical Preliminaries

A. Channel Model

In this paper, we consider a MIMO wireless communication system with one transmitter and two receivers. The transmitter (Tx) is equipped with m transmit antennas and broadcasts information to the users. Inside Tx, a linear precoder is applied as shown in Fig. 2. In this figure, $\mathbf{s} \triangleq [s_1, \ldots, s_m]^T$ is an independent and unit power symbol vector, that is, $\mathbb{E}\{\mathbf{s}\mathbf{s}^T\} = \mathbf{I}_m$. $\mathbf{\Lambda} \triangleq \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$ represents the power allocation matrix, and $\mathbf{V} \in \mathbb{R}^{m\times m}$ is the precoding matrix. Then, the transmitted signal $\mathbf{x}$ is

$$\mathbf{x} = \mathbf{V}\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{s}, \tag{1}$$

whose covariance matrix is $\mathbf{Q} \triangleq \mathbb{E}\{\mathbf{x}\mathbf{x}^T\} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$. The channel input is subject to an average total power constraint

$$\mathrm{tr}(\mathbb{E}\{\mathbf{x}\mathbf{x}^T\}) \le P. \tag{2}$$

At the receivers' side, user equipment 1 (UE1) and user equipment 2 (UE2) are equipped with n_1 and n_2 antennas, respectively. It is assumed that the transmission is over a flat fading channel. The input-output relations are given as

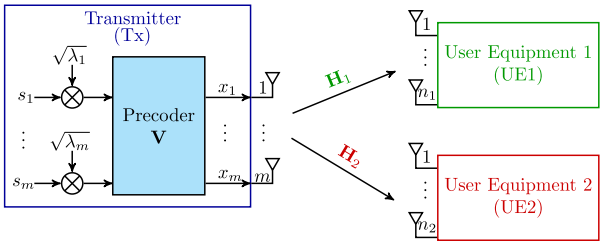


Fig. 2. The two-user MIMO network with m, n_1, and n_2 antennas at Tx, UE1, and UE2, respectively. Here, $\mathbf{x} \in \mathbb{R}^{m\times 1}$ is the transmitted signal, $\mathbf{V} \in \mathbb{R}^{m\times m}$ is the precoding matrix, and $\lambda_1, \ldots, \lambda_m$ are the powers allocated to symbols $s_1, \ldots, s_m$, respectively.

$$\mathbf{y}_1 = \mathbf{H}_1\mathbf{x} + \mathbf{w}_1, \tag{3a}$$

$$\mathbf{y}_2 = \mathbf{H}_2\mathbf{x} + \mathbf{w}_2, \tag{3b}$$

in which $\mathbf{y}_1 \in \mathbb{R}^{n_1\times 1}$ and $\mathbf{y}_2 \in \mathbb{R}^{n_2\times 1}$ are received signals at UE1 and UE2, $\mathbf{H}_1 \in \mathbb{R}^{n_1\times m}$ and $\mathbf{H}_2 \in \mathbb{R}^{n_2\times m}$ are the channels corresponding to UE1 and UE2, and $\mathbf{w}_1 \in \mathbb{R}^{n_1\times 1}$ and $\mathbf{w}_2 \in \mathbb{R}^{n_2\times 1}$ are independent and identically distributed (i.i.d) Gaussian noises with zero means and identity covariance matrices. The above-mentioned MIMO communication system can have multiple objectives as described in the following. Throughout this paper, $\mathcal{O}_i$ refers to objective i, $i \in \{1, \ldots, 5\}$, as described in Table I.

B. Objectives

1) WIT ($\mathcal{O}_1$): In this objective, UE1 acts as an information decoding user who seeks for highest transmission rate over the MIMO channel, while UE2 is ignored. The information transmission capacity $\mathcal{C}_1$ is obtained by solving the following problem [3]

$$\text{(P1)} \quad \mathcal{C}_1 \triangleq \max_{\mathbf{Q}} \frac{1}{2}\log|\mathbf{I}_{n_1} + \mathbf{H}_1\mathbf{Q}\mathbf{H}_1^T|, \tag{4a}$$

$$\text{s.t. } \mathbf{Q} \succeq \mathbf{0}, \mathbf{Q} = \mathbf{Q}^T, \mathrm{tr}(\mathbf{Q}) \le P. \tag{4b}$$

The optimal solution of (P1) is obtained using singular value decomposition (SVD) and water-filling algorithm [3]. The optimal covariance matrix in (P1) can be expressed as

$$\mathbf{Q}_1^* = \mathbf{A}\mathbf{B}\mathbf{A}^T, \tag{5}$$

in which $\mathbf{A}$ is obtained as the *right-singular vectors* of the channel $\mathbf{H}_1$, and $\mathbf{B}$ is obtained from water-filling algorithm [3]. Later, we will use this analytical solution to generate training sets for $\mathcal{O}_1$.

2) EH ($\mathcal{O}_2$): EH refers to transmitting electrical energy originated from a power source. The transmitter emits radio-frequency signals, and UE2, as an EH user, tries to maximize energy transmission efficiency. The objective function of this problem is shown in [4]

$$\text{(P2)} \quad \mathcal{C}_2 \triangleq \max_{\mathbf{Q}} \eta \cdot \mathrm{tr}(\mathbf{H}_2\mathbf{Q}\mathbf{H}_2^T), \tag{6a}$$

$$\text{s.t. } \mathbf{Q} \succeq \mathbf{0}, \mathbf{Q} = \mathbf{Q}^T, \mathrm{tr}(\mathbf{Q}) \le P \tag{6b}$$

Authorized licensed use limited to: VILLANOVA UNIVERSITY. Downloaded on July 24,2021 at 02:23:56 UTC from IEEE Xplore. Restrictions apply.

where η is the *converting rate* of the harvested energy and, without loss of generality, we assume $\eta = 1$ throughout the paper. The optimal analytical solution is given in [4]. It applies SVD to decompose channel $\mathbf{H}_2$ as $\mathbf{H}_2 = \mathbf{EFG}^T$, in which $\mathbf{E}$ and $\mathbf{G}$ are orthonormal matrices and $\mathbf{F}$ is a diagonal matrix that contains non-negative singular values. If the diagonal elements of $\mathbf{F}$ are in descending order, then the optimal solution of (P2) is given as [4]

$$\mathbf{Q}_2^* = P\mathbf{g}_1\mathbf{g}_1^T \tag{7}$$

where $\mathbf{g}_1$ is the first column of $\mathbf{G}$.

3) SWIPT ($\mathcal{O}_3$): SWIPT is to balance the WIT from Tx to UE1 and the EH at UE2 simultaneously. As defined in [4], SWIPT characterizes the optimal trade-off between the maximum energy and information transfer by the *rate-energy region* which is formed as [4],

$$\text{(P3)} \quad \mathcal{C}_3 \triangleq \max_{\mathbf{Q}} \frac{1}{2}\log|\mathbf{I}_{n_1} + \mathbf{H}_1\mathbf{Q}\mathbf{H}_1^T|, \tag{8a}$$

$$\text{s.t. } \eta \cdot \mathrm{tr}(\mathbf{H}_2\mathbf{Q}\mathbf{H}_2^T) \geq \bar{\mathcal{E}}, \tag{8b}$$

$$\mathbf{Q} \succeq \mathbf{0}, \mathbf{Q} = \mathbf{Q}^T, \mathrm{tr}(\mathbf{Q}) \leq P, \tag{8c}$$

in which $\bar{\mathcal{E}}$ is a dynamic threshold representing the required minimum energy harvested by UE2. The value of $\bar{\mathcal{E}}$ is in the range from minimum ($\mathcal{E}_{\min}$) to maximum ($\mathcal{E}_{\max}$),

$$\bar{\mathcal{E}} \triangleq \mathcal{E}_{\min} + q(\mathcal{E}_{\max} - \mathcal{E}_{\min}), \tag{9}$$

where q is called the *normalized EH level* and varies from 0% to 100%. Since we are looking for the maximum rate-energy boundary, $\mathcal{E}_{\min}$ is defined as the energy received by UE2 when UE1 achieves the maximum data rate, i.e., (P1) reaches its optimal. Then, we have $\mathcal{E}_{\min} = \eta \cdot \mathrm{tr}(\mathbf{H}_2\mathbf{Q}_1^*\mathbf{H}_2^T)$ where $\mathbf{Q}_1^*$ is given in (5). On the other hand, $\mathcal{E}_{\max}$ can be obtained when UE2 reaches the maximum EH level by solving (P2), i.e., $\mathcal{E}_{\max} = \mathcal{C}_2$. When $q = 0\%$ or $q = 100\%$, (P3) degenerates to (P1) and (P2), respectively.

4) PHY Security ($\mathcal{O}_4$): Under this objective, UE1 is a legitimate user and requires a message while keeping it secret from an eavesdropper, UE2. The precoder is expected to maximize the secrecy transmission rate [27]

$$\text{(P4)} \quad \mathcal{C}_4 \triangleq \max_{\mathbf{Q}} \frac{1}{2}\log\frac{|\mathbf{I}_{n_1} + \mathbf{H}_1\mathbf{Q}\mathbf{H}_1^T|}{|\mathbf{I}_{n_2} + \mathbf{H}_2\mathbf{Q}\mathbf{H}_2^T|}, \tag{10a}$$

$$\text{s.t. } \mathbf{Q} \succeq \mathbf{0}, \mathbf{Q} = \mathbf{Q}^T, \mathrm{tr}(\mathbf{Q}) \leq P. \tag{10b}$$

An optimal analytical solution for MIMO wiretap channel only exists in limited cases, including the case when $n_t = 2$ [9]. Other solutions are numerical like GSVD [6], AO-WF [7], etc.

5) Multicasting ($\mathcal{O}_5$): In this configuration, Tx offers a multicast message to multiple users, such as advertisements and emergency alerts. To ensure that the multicast message can be decoded by everyone, the multicast rate is limited to the minimum rate of the receivers. The transmission rate of this problem is formulated as [12]

$$\text{(P5)} \quad \mathcal{C}_5 \triangleq \max_{\mathbf{Q}} \min_{u=1,\ldots,U} \frac{1}{2}\log|\mathbf{I}_{n_u} + \mathbf{H}_u\mathbf{Q}\mathbf{H}_u^T|, \tag{11a}$$

$$\text{s.t. } \mathbf{Q} \succeq \mathbf{0}, \mathbf{Q} = \mathbf{Q}^T, \mathrm{tr}(\mathbf{Q}) \leq P. \tag{11b}$$

In the MISO case, SDR techniques yield a closed-form solution [11]. In the MIMO case, CAA is proposed in [12], and [28] mentions that the problem can be solved by semidefinite programming (SDP) directly. Moreover, [29] introduced a nonlinear random search with rotation parameters. However, the existing methods are limited to the computational complexity or are only available for a specific number of antennas.

As we saw, (P1)-(P5) have different expressions and solutions in general. As the objective changes, the corresponding solution changes completely. In addition, best solutions for (P3)-(P5) are iterative, which are slow and incur high complexity. To tackle this, we first propose a unified solution for (P3)-(P5), which is robust and reliable in a variety of antenna settings. Then, we propose DNN-based precoding that can solve (P1)-(P5) simultaneously and efficiently.

III. Rotation-Based Precoding

In this section, we introduce RP and apply it to (P3)-(P5). We should highlight that RP can also be applied to (P1)-(P2), but these two problems have competitive analytical solutions, and there is no need for a new solution. Besides, an arbitrary covariance matrix $\mathbf{Q}$ can be formed by non-negative eigenvalues and rotation angles [8, Lemma 1]. Hence, the optimization problem can be transformed to an equivalent parameterized optimization using RP.

A. Rotation-Based Precoding (RP)

The covariance matrix $\mathbf{Q}$ can be formed using the eigenvalue decomposition as

$$\mathbf{Q} \triangleq \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T, \tag{12}$$

in which $\mathbf{\Lambda} \in \mathbb{R}^{m\times m}$ is a diagonal matrix, whose diagonal elements $[\lambda_1, \ldots, \lambda_m]$ are non-negative due to the PSD constraint. Then, the average power constraints in (P1)-(P5) are equivalent to $\sum_{i=1}^m \lambda_i \leq P$. Thus, the PSD and power constraints can be represented as a set of linear constraints

$$\{\lambda_i | \lambda_i \geq 0, \quad \sum_{i=1}^m \lambda_i \leq P\}. \tag{13}$$

Besides, $\mathbf{V} \in \mathbb{R}^{m\times m}$ is an orthonormal matrix due to the symmetric property of $\mathbf{Q}$. It can be modeled as a Given's matrix [8], [30] also named as a rotation matrix

$$\mathbf{V} = \prod_{i=1}^{m-1}\prod_{j=i+1}^{m} \mathbf{V}_{i,j}, \tag{14}$$

where $\mathbf{V}_{i,j}$ is an identity matrix except for four elements

$$\begin{bmatrix} v_{i,i} & v_{i,j} \\ v_{j,i} & v_{j,j} \end{bmatrix} = \begin{bmatrix} \cos\theta_{i,j} & -\sin\theta_{i,j} \\ \sin\theta_{i,j} & \cos\theta_{i,j} \end{bmatrix}. \tag{15}$$

Intuitively, for any vector $\mathbf{v}$ in $\mathbb{R}^{m\times 1}$ vector space, $\mathbf{V}_{i,j} \cdot \mathbf{v}$ represents a rotating from the ith standard basis to the jth standard basis with a certain rotation angle $\theta_{i,j}$. In total, we need[3]

$$n_a = \frac{1}{2}m(m-1), \tag{16}$$

[3]Specially, for $m = 1$, $\mathbf{Q}$ becomes a scalar. In RP, we only have one eigenvalue and no rotation angles. In such a case, (16) becomes 0.

Authorized licensed use limited to: VILLANOVA UNIVERSITY. Downloaded on July 24,2021 at 02:23:56 UTC from IEEE Xplore. Restrictions apply.

rotation angles to represent $\mathbf{V}$ in (14). There is no constraint on rotation angles, i.e., $\theta_{i,j} \in \mathbb{R}$. In [8, Lemma 1], we have proved that an arbitrary covariance matrix $\mathbf{Q}$ can be represented by m non-negative eigenvalues and n_a rotation angles. Therefore, the optimization on $\mathbf{Q}$ can be equivalently transformed to optimization parameters using RP with the constraint (13).

It is worth mentioning that the order of multiplication in (14) is not unique and different order will lead to different rotation angles $\theta_{i,j}$. In this paper, without loss of generality, we use the order defined in (14). Then, the rotation parameter vector can be defined as

$$\mathbf{r} \triangleq [\boldsymbol{\lambda}, \boldsymbol{\theta}]^T \tag{17}$$

where

$$\boldsymbol{\lambda} \triangleq [\lambda_1, \ldots, \lambda_m] \text{ and } \boldsymbol{\theta} \triangleq [\theta_{1,2}, \ldots, \theta_{m-1,m}]. \tag{18}$$

To this end, $\mathbf{Q}$ can be specified by the parameter vector $\mathbf{r}$ with the new constraint

$$\mathbf{L}\mathbf{r} \leq \mathbf{b}, \tag{19}$$

where

$$\mathbf{L} \triangleq \begin{bmatrix} -\mathbf{I}_m & \mathbf{0}_{m\times n_a} \\ \mathbf{1}_{1\times m} & \mathbf{0}_{1\times n_a} \end{bmatrix} \text{ and } \mathbf{b} \triangleq \begin{bmatrix} \mathbf{0}_{1\times m} \\ P \end{bmatrix}. \tag{20}$$

B. Rotation-Based Precoder for $\mathcal{O}_3$ to $\mathcal{O}_5$

The problems (P3)-(P5) are challenging and optimal analytical precoding matrices are not known. In the following, we apply RP to the problems, which can parameterize all of the problems with rotation angles and power allocation parameters. One benefit of the new formulation is that it transforms the matrix constraints $\mathbf{Q} \succeq \mathbf{0}$ into a set of linear constraints and hence simplifies the optimization problems.

1) RP for SWIPT: Applying the RP on (P3), the objective function of SWIPT becomes

$$\text{(P3a)} \quad \mathcal{C}_3 = \max_{\mathbf{r}} \frac{1}{2}\log|\mathbf{I}_{n_1} + \mathbf{H}_1\mathbf{Q}\mathbf{H}_1^T|, \tag{21a}$$

$$\text{s.t. } \eta \cdot \mathrm{tr}(\mathbf{H}_2\mathbf{Q}\mathbf{H}_2^{\mathrm{T}}) \geq \bar{\mathcal{E}}, \tag{21b}$$

$$\mathbf{L}\mathbf{r} \leq \mathbf{b}, \tag{21c}$$

in which the three constraints (symmetric, PSD, and total power constraint) in (8c) are converted to one linear constraint as (21c). This problem can be solved by a general optimization tool such as `fmincon` in MATLAB. Here, (21c) is a linear inequality constraint and (21b) can be added as a non-linear constraint in `fmincon`. For initialization of $\mathbf{Q}$ we use the solution of (P2), i.e., $\mathbf{Q}_2^*$ in (7). Then, we obtain the initial value $\mathbf{r}$ using (12)-(15) or Algorithm 1 in [8]. Finally, we can obtain $\mathbf{Q}_3^*$ which is defined to be the optimal solution for $\mathcal{O}_3$.

2) RP for PHY Security: Applying RP to PHY Security problem results in

$$\text{(P4a)} \quad \mathcal{C}_4 = \max_{\mathbf{r}} \frac{1}{2}\log\frac{|\mathbf{I}_{n_1} + \mathbf{H}_1\mathbf{Q}\mathbf{H}_1^T|}{|\mathbf{I}_{n_2} + \mathbf{H}_2\mathbf{Q}\mathbf{H}_2^T|}, \tag{22a}$$

$$\text{s.t. } \mathbf{L}\mathbf{r} \leq \mathbf{b}. \tag{22b}$$

The rotation-based modeling transforms the constraints (10b) of (P4) into (22b). Then, this new optimization problem can be solved by a convex toolbox such as `fmincon` in MATLAB. Then, we can obtain $\mathbf{Q}_4^*$ for $\mathcal{O}_4$. Although the PHY security is known as a non-convex problem, the performance of RP is more reliable compared with existing solutions, such as GSVD [6] and AO-WF [7].

3) RP for Multicasting: In this paper $U = 2$ and (P5) can be reformed as

$$\text{(P5a)} \quad \mathcal{C}_5 = \max_{\mathbf{r}} \min\{R_u\}, \quad u = 1, 2, \tag{23a}$$

$$\text{s.t. } \mathbf{L}\mathbf{r} \leq \mathbf{b} \tag{23b}$$

where R_u represents the WIT rates of UE1 and UE2, i.e.,

$$R_u(\mathbf{Q}) \triangleq \frac{1}{2}\log|\mathbf{I}_{n_u} + \mathbf{H}_u\mathbf{Q}\mathbf{H}_u^T|, \quad u = 1, 2. \tag{24a}$$

(P5a) is the minimum of two WIT problems represented in (P1) which is concave [3]. Thus, (P5a) is concave. Define the optimal solutions $\mathbf{Q}_1^{*(1)}$ and $\mathbf{Q}_1^{*(2)}$ for R_1 and R_2 in (24a). Then, (P5a) in (23) can be solved by three sub-cases:

- *Case 1:* $R_1(\mathbf{Q}_1^{*(1)}) \leq R_2(\mathbf{Q}_1^{*(1)})$, then the optimal multicast covariance matrix of (23) is $\mathbf{Q}_5^* = \mathbf{Q}_1^{*(1)}$.
- *Case 2:* $R_1(\mathbf{Q}_1^{*(2)}) \geq R_2(\mathbf{Q}_1^{*(2)})$, the optimal multicast covariance matrix of (23) is $\mathbf{Q}_5^* = \mathbf{Q}_1^{*(2)}$.
- *Case 3:* Otherwise, we solve the rotation parameters in (23) using `fmincon`.

Since the first two sub-cases are actually WIT problems with analytical solutions, the efficiency of the solution improves compared to iterative solutions such as CAA [12] and SDP. Till now, the solutions $\mathbf{Q}_3^*$ to $\mathbf{Q}_5^*$ for $\mathcal{O}_3$ to $\mathcal{O}_5$ are obtained using the RP, respectively. In the next section, we propose a supervised DNN that learns from the above solutions and find the covariance matrices corresponding to (P1)-(P5) at once.

The RP-based solution can be applied to a more general case with multiple users, i.e., $U > 2$. With more users, the capacity evaluation can be more complex, and the rotation parameters may increase or remain the same depending on the specific problem. For example, for multicasting ($\mathcal{O}_5$) even if there are multiple users there is one covariance matrix to be optimized [12], and thus, the DNN training and accuracy will not be affected. On the other hand, if the number of covariance matrices increases by introducing more users in the network, the DNN-based precoder should be retrained. For example, $\mathcal{O}_1$, MIMO transmission with multiple users, also known as Gaussian MIMO broadcast channel [31], and $\mathcal{O}_4$, Gaussian MIMO multi-receiver wiretap channel [32], the covariance matrix $\mathbf{Q}$ is constructed by U independent covariance matrices, i.e., $\mathbf{Q} = \sum_{u=1}^{U} \mathbf{Q}_u$. When applying RP for user u, we have $\mathbf{Q}_u = \mathbf{V}_u\boldsymbol{\Lambda}_u\mathbf{V}_u^T$. Thus, we will have independent rotation parameters for each user to optimize and train the DNN.

IV. A UNIFIED DNN-BASED PRECODER

In this section, we introduce a unified DNN-based precoding and power allocation scheme, including the DNN structure, the input features, and network outputs. DNN can increase efficiency by unifying the solution for all of the problems together in contrast to the conventional methods which perform optimization one by one.

Authorized licensed use limited to: VILLANOVA UNIVERSITY. Downloaded on July 24,2021 at 02:23:56 UTC from IEEE Xplore. Restrictions apply.

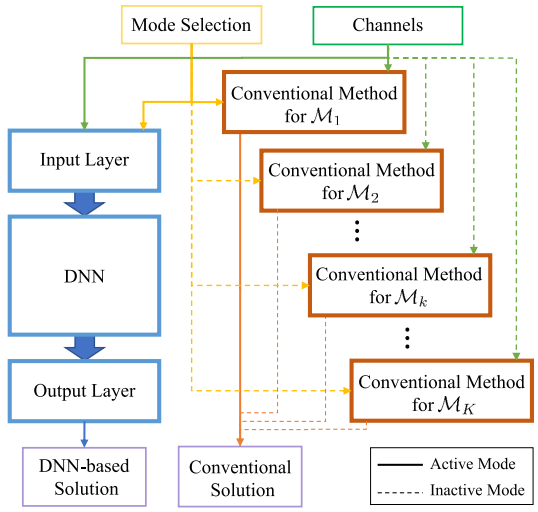


Fig. 3. The structure of the proposed DNN-based precoding. The currently selected mode is $\mathcal{M}_k$.

Before talking about the details of the DNN, we indicate that $\mathcal{O}_3$ (the SWIPT problem) will be differentiated for nine normalized EH level in (9) for

$$q \in \{90\%, 80\%, \ldots, 10\%\}. \tag{25}$$

These sub-problems are named $\mathcal{O}_3(90\%)$ to $\mathcal{O}_3(10\%)$ as shown in Table II. Then, for the proposed DNN, a unique index $\mathcal{M}_k$, $k \in \{1, \ldots, K\}$, represents different modes. Therefore, we have $K = 13$ modes in total. It is worth clarifying that $\mathcal{O}_i$ refers to the configurations of precoding objectives, i.e., WIT, EH, etc., while we use $\mathcal{M}_k$ for integer index and simplifying the expressions. These are listed in Table II for clarity.

A. The DNN Structure

The structure of the proposed DNN-based precoding is demonstrated in Fig. 3. At the top, the parameters of the two users, including mode and channel selection. The DNN-based precoder can provide the precoding solution directly, while it is necessary to activate one of the conventional methods according to the user requirement. The DNN-based precoder can be divided into three parts shown by different colors in Fig. 3. These are 1) the input layer, which pre-processes the input information and generates a feature vector for DNN; 2) DNN is applied to achieve the non-linear mapping between the features and demanded outputs; 3) the output layer that maps the output to a covariance matrix for precoding.

As shown in Fig. 4, the architecture of the DNN precoder has ten fully-connected hidden layers equipped with *parametric rectified linear units* (PReLU) [33] as activation functions. PReLU is defined as

$$f(y) = \begin{cases} y, & y \geq 0, \\ \alpha y, & y < 0 \end{cases} \tag{26}$$

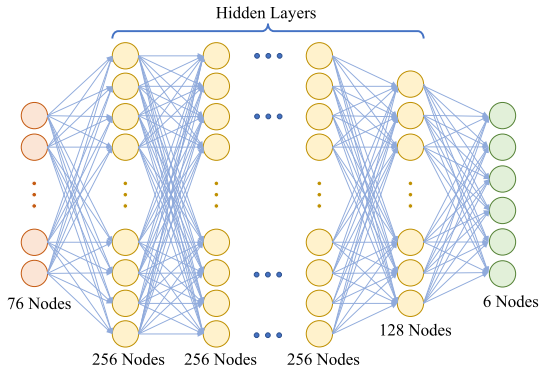


Fig. 4. The DNN architecture.

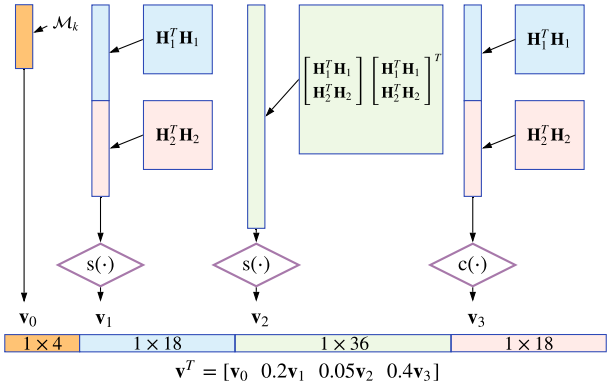


Fig. 5. Feature design for the input layer.

where $\alpha = 0.25$ is a trainable initialized parameter.[4] PReLU extends the freedom of the DNN to mimic a mapping and to prevent over-fitting at the same time.

B. Input Features

The schematic diagram of the input layer is shown in Fig. 5. The required inputs are $\mathbf{H}_1$, $\mathbf{H}_2$, and mode index, which is given by $k \in \{1, \ldots, K\}$. The feature vector contains four sub-features, namely, $\mathbf{v}_0$, $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$ which are

$$\mathbf{v}_0 \triangleq \mathbf{c}_k, \tag{27a}$$
$$\mathbf{v}_1 \triangleq \mathrm{s}(\mathrm{vec}(\mathbf{F}))^T, \tag{27b}$$
$$\mathbf{v}_2 \triangleq \mathrm{s}(\mathrm{vec}(\mathbf{F}^T\mathbf{F}))^T, \tag{27c}$$
$$\mathbf{v}_3 \triangleq \mathrm{c}(\mathrm{vec}(\mathbf{F}))^T \tag{27d}$$

where $\mathbf{c}_k \in \mathbb{R}^{1\times 4}$ is a *binary-code vector* identifying the k-th objective. The objectives and the corresponding code are listed in Table II. $\mathbf{F}$ contains channel information defined as

$$\mathbf{F} \triangleq [\mathbf{H}_1^T\mathbf{H}_1 \;\; \mathbf{H}_2^T\mathbf{H}_2]. \tag{28}$$

$\mathrm{s}(x)$ is the element-wise square root function keeping the sign of input x, i.e.,

$$\mathrm{s}(x) \triangleq \mathrm{sign}(x) \cdot |x|^{\frac{1}{2}}. \tag{29}$$

[4]We have examined the performance of PReLU initialized with a fixed value 0.25 given in [33] and random uniformly distributed values. The fixed initialization achieves better performance, especially for $\mathcal{O}_4$.

Authorized licensed use limited to: VILLANOVA UNIVERSITY. Downloaded on July 24,2021 at 02:23:56 UTC from IEEE Xplore. Restrictions apply.

TABLE II
CODE VECTOR DESIGN FOR EACH MODE

Objective	Mode	Code ($\mathbf{c}_k$)	Objective	Mode	Code ($\mathbf{c}_k$)
$\mathcal{O}_1$	$\mathcal{M}_1$	0001	$\mathcal{O}_3(40\%)$	$\mathcal{M}_8$	1000
$\mathcal{O}_2$	$\mathcal{M}_2$	0010	$\mathcal{O}_3(30\%)$	$\mathcal{M}_9$	1001
$\mathcal{O}_3(90\%)$	$\mathcal{M}_3$	0011	$\mathcal{O}_3(20\%)$	$\mathcal{M}_{10}$	1010
$\mathcal{O}_3(80\%)$	$\mathcal{M}_4$	0100	$\mathcal{O}_3(10\%)$	$\mathcal{M}_{11}$	1011
$\mathcal{O}_3(70\%)$	$\mathcal{M}_5$	0101	$\mathcal{O}_4$	$\mathcal{M}_{12}$	1100
$\mathcal{O}_3(60\%)$	$\mathcal{M}_6$	0110	$\mathcal{O}_5$	$\mathcal{M}_{13}$	1101
$\mathcal{O}_3(50\%)$	$\mathcal{M}_7$	0111			

Similarly, $\mathrm{c}(x)$ is the element-wise cubic root function, which defined as

$$\mathrm{c}(x) \triangleq \mathrm{sign}(x) \cdot |x|^{\frac{1}{3}}. \tag{30}$$

Among the sub-feature vectors, $\mathbf{v}_0$ represents the feature with respect to the mode index. With such a definition, the DNN has better performance in recognizing the input objectives. Besides, $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$ are formed based on channel matrices. We use $\mathbf{H}_1^T\mathbf{H}_1$ and $\mathbf{H}_2^T\mathbf{H}_2$ rather than using $\mathbf{H}_1$ and $\mathbf{H}_2$ directly, since

$$|\mathbf{I} + \mathbf{HQH}| = |\mathbf{I} + \mathbf{H}^T\mathbf{HQ}|, \tag{31a}$$
$$\mathrm{tr}(\mathbf{HQH}^T) = \mathrm{tr}(\mathbf{H}^T\mathbf{HQ}), \tag{31b}$$

can be applied to optimization problems introduced in Section II. With such a definition, the dimension of input features is related only to m rather than m, n_1, and n_2. Furthermore, non-linear combinations of channels are also considered to deliver more information to the DNN in order to achieve a better non-linear capability.

Considering that the distribution of the channel elements is Gaussian, the elements of $\mathbf{F}$ have a high density concentrating around 0. This decreases the fairness and distinguishability of the input features. To overcome this problem, we use square root and cubic root in (27b)-(27d) to flatten the distribution of the features to some degree. Finally, the input feature vector $\mathbf{v}$ is a cascade of these sub-feature vectors

$$\mathbf{v} \triangleq [\mathbf{v}_0, 0.2\mathbf{v}_1, 0.05\mathbf{v}_2, 0.4\mathbf{v}_3]^T \tag{32}$$

where the coefficients are chosen experimentally to normalize the sub-feature vectors and improve the accuracy of the DNN and the speed of learning [34], [35]. In this paper, we consider $P = 20$ and $m = 3$ while n_1 and n_2 can be any number based on specific cases. In this scenario, the size of the input of the DNN is 76.[5]

C. Network Outputs

Since the covariance matrix is symmetric, the output (vector $\mathbf{q}$) contains only upper triangular elements of $\mathbf{Q}$. That is,

$$\mathbf{q} \triangleq [q_{1,1}, q_{2,2}, q_{3,3}, q_{1,2}, q_{2,3}, q_{1,3}]^T. \tag{33}$$

Then, $\mathbf{Q}$ can be assembled as

$$\mathbf{Q} = \begin{bmatrix} q_{1,1} & q_{1,2} & q_{1,3} \\ q_{2,1} & q_{2,2} & q_{2,3} \\ q_{3,1} & q_{3,2} & q_{3,3} \end{bmatrix} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T. \tag{34}$$

[5]The input vector $\mathbf{v}$ contains 27 pairs of the same features since $\mathbf{H}_1^T\mathbf{H}_1$ and $\mathbf{H}_2^T\mathbf{H}_2$ are symmetric. Such redundancy will be automatically reduced by the first hidden layer of the DNN [35].

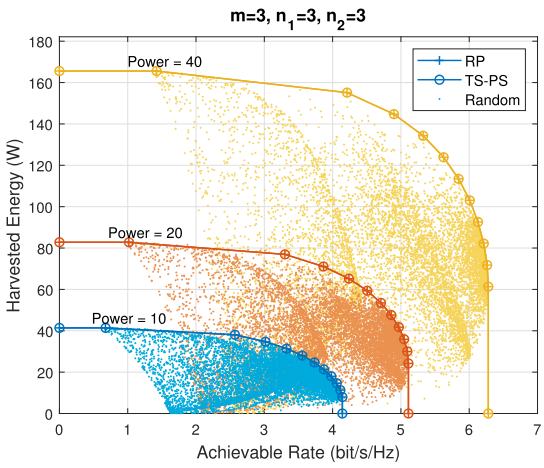


Fig. 6. Comparisons of SWIPT region for the RP and TS-PS for 10,000 random trials. The colors distinguish different transmit powers.

The precoding and power allocation matrices $\mathbf{V}$ and $\mathbf{\Lambda}$, respectively, are obtained by eigenvalue decomposition. To ensure the PSD and total power constraints, negative diagonal elements in $\mathbf{\Lambda}$ are normalized to zero and the trace is scaled to P. In the training procedure, $\mathbf{q}$ is known through RP corresponding to the problem discussed in Section III. Whereas, during testing, $\mathbf{Q}$ will be obtained from the output vector and the precoding solution is obtained by eigenvalue decomposition [17].

V. TRAINING PROCEDURE AND NUMERICAL RESULTS

In this section, we initially verify the performance of RP which is used to train the network. Then, we explain the details of our data set and the training procedure. Finally, we examine the performance of the proposed DNN-based precoding.

A. Performance of RP

1) SWIPT ($\mathcal{O}_3$): In Fig. 6, the performance of RP is compared with the *time-switching and power-splitting* (TS-PS) [4] and random trials of $\mathbf{Q}$. For RP and TS-PS, eleven thresholds $\bar{\mathcal{E}}$ equally dividing the interval $[\mathcal{E}_{\min}, \mathcal{E}_{\max}]$ are considered in the cases of $P = 10$, 20, and 40 (W). The channels are assumed to be quasi-static, flat Rayleigh fading channels, in which the channel coefficients are randomly and independently generated using the standard Gaussian distribution:

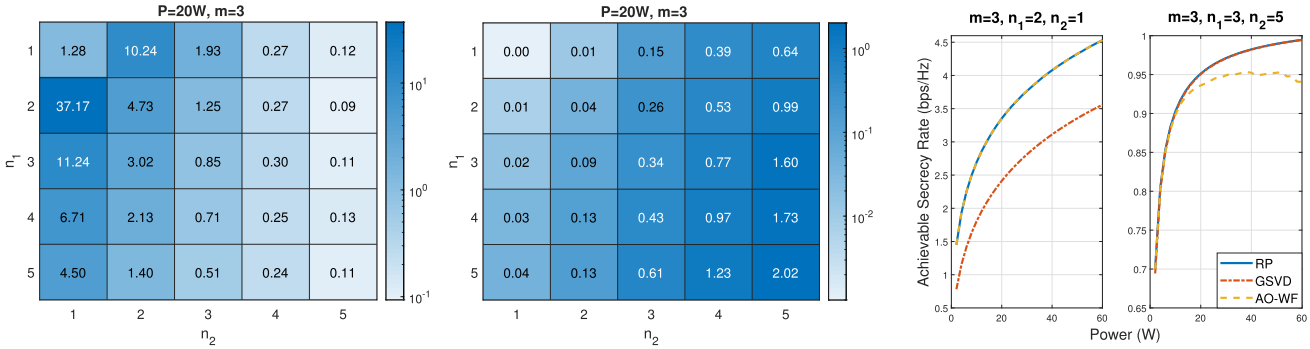
$$\mathbf{H}_1 = \begin{bmatrix} -2.2975 & 0.4896 & -1.8310 \\ 1.4576 & -0.6100 & 0.3800 \\ 0.8998 & 0.0916 & -0.3128 \end{bmatrix}, \tag{35a}$$

$$\mathbf{H}_2 = \begin{bmatrix} -0.3276 & 3.3159 & -0.9956 \\ 1.5765 & 0.2604 & 0.2578 \\ -0.3337 & 1.1478 & -0.3364 \end{bmatrix}. \tag{35b}$$

For both channels, RP can reach the same rate-energy region as TS-PS. The random trials are based on 10,000 realizations of $\mathbf{Q}$.

2) PHY Security ($\mathcal{O}_4$): In this subsection, we consider the MIMO wiretap channel. The performance of RP is compared with GSVD [6] and AO-WF [7] in Fig. 7. The achievable

Authorized licensed use limited to: VILLANOVA UNIVERSITY. Downloaded on July 24,2021 at 02:23:56 UTC from IEEE Xplore. Restrictions apply.



(a) Relative improvement (%) of RP to GSVD. (b) Relative improvement (%) of RP to AO-WF. (c) Typical cases that RP exceeds GSVD or AO-WF.

Fig. 7. Comparisons between the secrecy transmission rate of the RP with GSVD and AO-WF.

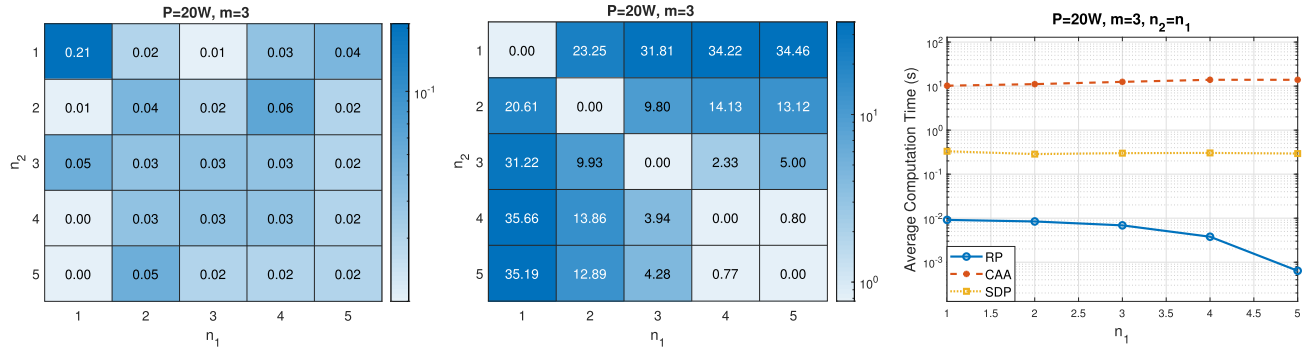


Fig. 8. Comparisons between the multicast transmission rate and computational cost of the RP with CAA and SDP.

secrecy rates are averaged over 200 random channel realizations, where the channels are generated as independent standard Gaussian random variables. Fig. 7(a) and Fig. 7(b) represent relative improvements defined as

$$\eta_g = \frac{(R_r - R_g)}{R_g} \times 100\%, \tag{36a}$$

$$\eta_a = \frac{(R_r - R_g)}{R_g} \times 100\%, \tag{36b}$$

in which η_g and η_a represent the percentages that RP exceeds GSVD and AO-WF, i.e., the bluer, the better. R_r, R_g, and R_a are the average secrecy rate achieved by RP, GSVD, and AO-WF, respectively. $P = 20$W and $m = 3$ are set in this figure. Each cell denotes a pair of n_1 and n_2. The RP can highly outperform GSVD for a fewer number of antennas; it can also outperform AO-WF with multiple antennas. There is a noticeable gap between RP and GSVD when the eavesdropper has a smaller number of antennas. For a larger n_2, RP is capable of reaching a higher secrecy rate compared to AO-WF. Further illustrations are shown in Fig. 7(c) considering two cases over 200 channel realizations. The plots show the average secrecy rates versus transmit power in the case of $m = 3$, $n_1 = 2$, $n_2 = 1$ and $m = 3$, $n_1 = 3$, $n_2 = 5$. We see that RP can perform stably and reliably in those cases. Moreover, the average time costs over all cases in Fig. 7(b) of RP is 27.65ms which is less than 38.26ms achieved by AO-WF.

3) Multicasting ($\mathcal{O}_5$): For multicasting, we compare RP with the CAA [12] and standard SDP techniques. Here, we apply CVX [36] to realize SDP solutions for multicasting.

We investigate a variety of combinations of $\{n_1, n_2\}$, and the results are listed in Fig. 8. Similar to (36a)-(36b), we define the relative improvement factors η_c and η_s representing the percentages that RP exceeds CAA and SDP, respectively. It can be seen from Fig. 8(a) that RP slightly beats CAA in all settings. Besides, RP outperforms SDP when $n_1 \neq n_2$. This advantage is especially remarkable when $n_1 > n_2 = 1$ or $n_2 > n_1 = 1$ in Fig. 8(b). It is worth noting that, in the case of $n_1 = n_2$, RP has a very close performance to SDP. This can be found in the diagonal of Fig. 8(b).

Moreover, the benefits of RP in reducing complexity cannot be ignored, which is analyzed in Fig. 8(c). Fig. 8(c) compares the time cost when $n_1 = n_2$ where SDP works well. It can be seen that RP has the best efficiency, while CAA is computationally expensive due to the successive optimization SDP problem. The improvement of time efficiency is partially due to the sub-cases we divided in Section III-B3. In *Case-1* and *Case-2*, the solution can be obtained analytically, which is much efficient than *Case-3*. The probability of *Case-3* is estimated by the Monte Carlo method. The results are obtained over 20,000 random channels for each specific n_1 and n_2 with fixed $P = 20$W and $m = 3$. As shown in Fig. 9, the probability of *Case-3* is reduced with the increase of n_1 and n_2, which explains the drop in the time cost of RP in Fig. 8(c).

In summary, RP provides a unified solution for the studied precoding problems. It is feasible for SWIPT. Also, it is reliable for PHY security and multicasting problems in the variety of the number of transmit and receive antennas.

Authorized licensed use limited to: VILLANOVA UNIVERSITY. Downloaded on July 24,2021 at 02:23:56 UTC from IEEE Xplore. Restrictions apply.

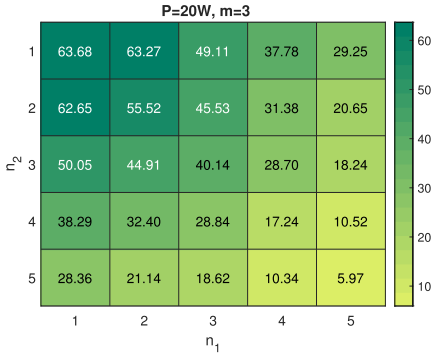


Fig. 9. The probability (%) that *Case-3* occurs when the RP is used for multicasting problem.

TABLE III
DATASETS GENERATION

Stage	m	n_1	n_2	Number of Channels	Number of Samples
Training	3	$1,\ldots,5$	$1,\ldots,5$	2,000,000	26,000,000
Test	3	$1,\ldots,5$	$1,\ldots,5$	10,000	130,000

TABLE IV
HYPER-PARAMETERS IN TRAINING PROCEDURE

Hyper-parameter	Value	Hyper-parameter	Value
Initial learning rate	0.001	Mini batch size	5000
Learn rate dropping factor	0.8	Max epochs	50
Learn rate dropping period	1		

B. Complexity Analysis of RP

Matrix multiplication and inversion yield the complexity of $\mathcal{O}(\Gamma^3)$ where $\Gamma = \max(m, n_1, n_2)$ [37]. The `fmincon` algorithm applies a quasi-Newton algorithm, namely, Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm together with interior-point method, which has the complexity of $\mathcal{O}(n^2)$ [38], where n is the size of input variables which is the total number of optimized rotation parameters, i.e., $\frac{1}{2}m(m+1)$ [8]. Thus, the overall complexity of RP together with `fmincon` is $\mathcal{O}(m^4 + \Gamma^3)$. For $\mathcal{O}_3$, SWIPT [4, Algorithm] uses the ellipsoid method, which has the computation complexity of $\mathcal{O}(n^2)$ per step. n is the number of variables, which is two in [4, Algorithm], and the step is $\frac{1}{n+1}$. Overall, SWIPT [4, Algorithm] achieves the $\mathcal{O}(\frac{\Gamma^3}{\epsilon})$, in which ϵ is the convergence tolerance of algorithm. For $\mathcal{O}_4$, AO-WF [7, Algorithm 1] yields $\mathcal{O}(\frac{\Gamma^3}{\epsilon}\log(1/\epsilon))$, and GSVD-based precoding [6] has the complexity of $\mathcal{O}(\Gamma^3 + \Gamma\log(1/\epsilon))$ [37]. More details can be found in [8] and the reference therein. For $\mathcal{O}_5$, CAA applies a second-order cone program (SOCP), and use the interior point method in SeDuMi, which yields the complexity of $\mathcal{O}(a^2b^{2.5}+b^{3.5})$ [39], where a is the number of input variables and b is the number of rows of the linear matrix inequality ($a = m^2$, and $b = 2$ is the number of users in [12, Formula (18)]). Hence, CAA [12, Algorithm 1] has the complexity of $\mathcal{O}(\frac{1}{\epsilon}(m^4 + \Gamma^3))$.

C. Data Set Generation and Training Procedure

In order to evaluate the performance of the proposed DNN-based precoding, we generate over two million random realizations of $\mathbf{H}_1$ and $\mathbf{H}_2$. Each element of the channels follows $\mathcal{N}(0, 1)$. From these channel realizations, 2,000,000 channels contribute to the training set and 10,000 of them are used for testing. Since the number of transmit antennas m and power P are fixed at the Tx side, we set $m = 3$ and $P = 20$W in all training and test sets. The number of antennas of UE1 and UE2, i.e., n_1 and n_2, are randomly chosen from 1 to 5, covering most cases of user devices.

For each channel realization in the training and test sets, we generate 13 samples (defined as input-output pairs) corresponding to the K objectives in Table II. In such K samples, each one contains an input feature vector according to (32) and a corresponding output vector $\mathbf{q}$ formulated as (33) given by the solution of the RP method. That is, for $\mathcal{O}_1$ and $\mathcal{O}_2$, we have analytical solutions given in Section II-B1 and II-B2; while for $\mathcal{M}_3$ to $\mathcal{M}_5$, the RP method for each is given in Section III-B1 to III-B3. Therefore, the training set has 26,000,000 samples and the test set has 130,000 samples. The details of training and test sets are summarized in Table III. The large training set can reduce the chance of over-fitting, as noted in [40].

The training procedure is executed on a single graphical card (NVIDIA GeForce GTX 1660Ti) using Adam [41] as an optimization method. All training procedures share the same group of hyper-parameters as listed in Table IV. The learning rate controls how quickly the DNN can change the weights, and it drops by 20% after one epoch in this paper. Mini batch size indicates how many samples are considered together for one update of the DNN weights. We choose a small mini-batch size to escape or reduce the over-fitting problem [42]. A relatively small mini-batch size is also helpful to improve the performance of generalization. The weights in DNN will fall into a flat local minimum rather than a sharp local minimum [43]. Max epochs denotes the times that all data set has been taken into the training procedure. After the training process, the DNN-based precoding is ready for testing.

D. Numerical Results

In this part, we evaluate the performance of the proposed unified DNN-based precoding in different antenna settings. The DNN precoder is used in the following way. Two users are equipped with any number of antennas covering from 1 to 5, respectively. For a given channel pairs $\mathbf{H}_1$ and $\mathbf{H}_2$ with a required mode from Table II, the input layer converts the channels and the mode to sub-feature vectors as (27b)-(27d) and (27a), respectively. Then, the output $\mathbf{q}$ in (33) can be found in the output layer.

The evaluation of the DNN-precoder contains three metrics:

1) The mean square error (MSE) of elements in $\mathbf{Q}$ provided by the DNN-based precoding;
2) The performance compared to the conventional methods.
3) Time consumption for each objective.

1) MSE of Elements of $\mathbf{Q}$*:* The MSE is evaluated to ensure feasibility. The MSE is defined as the variance between the

Authorized licensed use limited to: VILLANOVA UNIVERSITY. Downloaded on July 24,2021 at 02:23:56 UTC from IEEE Xplore. Restrictions apply.

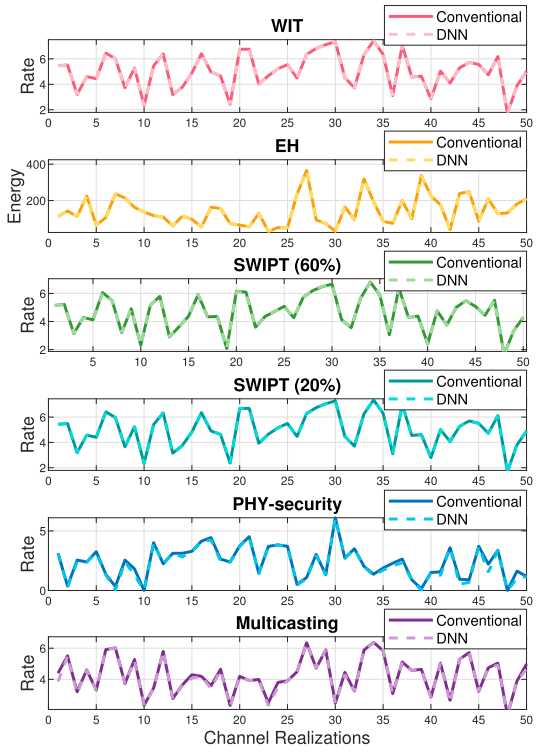


Fig. 10. The performance of the proposed DNN-based precoder compared with conventional methods on 5 objectives. In this figure, the achievable rate is in bit per second per hertz (bit/s/Hz) and harvested energy is in Watt (W).

TABLE V
AVERAGE ACHIEVABLE RATE (Bit/s/Hz) OR HARVESTED ENERGY (W)

Objective	Conventional	DNN	Accuracy (%)
$\mathcal{O}_1$	4.9807	4.9794	99.97
$\mathcal{O}_2$	130.99	130.78	99.84
$\mathcal{O}_3(60\%)$	4.5703	4.5688	99.97
$\mathcal{O}_3(20\%)$	4.9344	4.9326	99.96
$\mathcal{O}_4$	2.3353	2.2041	94.38
$\mathcal{O}_5$	4.0055	3.9707	99.13

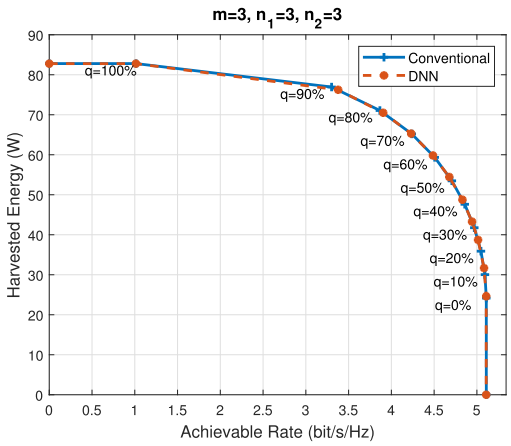


Fig. 11. The rate-energy region achieved by DNN-precoder for SWIPT.

$q_{i,j}$ given by the DNN-based precoder and the one obtained by the corresponding conventional method. The MSE is between 0.0652 and 0.0918 averaged over 130,000 samples in the test set, which indicates the capability of DNN-based precoding.

2) Performance Evaluation: The performance of the proposed DNN-based precoder for each objective is shown in Fig. 10, where the achievable rate is in bit/s/Hz and harvested energy in Watt is normalized by the baseband symbol period [4]. Here we plot the first fifty channel realizations from the test set for each objective. We choose $\mathcal{O}_3(60\%)$ and $\mathcal{O}_3(20\%)$ as representatives for SWIPT. In each subfigure, the solid line represents the achievable rate or harvested energy obtained by conventional methods (analytical solutions for $\mathcal{O}_1$ and $\mathcal{O}_2$, and RP for $\mathcal{O}_3$ to $\mathcal{O}_5$), whereas the dashed light-colored line is the result given by the DNN-precoder. It can be seen that the results are almost fitting the corresponding conventional solutions.

The average achievable rates or harvested energy are reported in Table V. The *accuracy* is defined as the percentage of DNN-precoder to the conventional methods, i.e.,

$$\zeta_d^{(k)} \triangleq \frac{R_d^{(k)}}{R_c^{(k)}} \times 100\%, \qquad k \in \{1, \ldots, 13\} \tag{37}$$

where k is the mode index, $R_d^{(k)}$ and $R_c^{(k)}$ are the results (achievable rate or harvested energy) of DNN-precoder and the conventional methods. Average accuracy is listed in Table V. On average, the accuracy is 99.45% among all tasks. The performance of the DNN-precoder could be seen the same as the RP method except for $\mathcal{O}_4$.

It is worth mentioning that the work in [17] is specifically aimed at $\mathcal{O}_4$ and is only feasible for $\{n_1 = 4, n_2 = 3\}$ and $\{n_1 = 2, n_2 = 1\}$. The accuracy in [17] is 97.71% and 93.72%. In this paper, the DNN-precoder is able to achieve 96.94% and 93.15% on those two settings of n_1 and n_2. The slight loss is attributed to much wider antenna settings in this problem, as shown in Table III.

Next, the rate-energy region of SWIPT is demonstrated in more details. We have arranged nine modes for SWIPT inside the DNN-based precoding, which can be generated as an achievable rate-energy region for any arbitrary channel. In Fig.11, the DNN-based solution is compared with conventional methods using the channels in (35). Both of the methods have been executed for $q = [100\%, 90\%, \ldots, 0\%]$. For the DNN-based precoding, $q = 100\%$ is actually $\mathcal{O}_2$, EH; $q = 0\%$ is obtained by $\mathcal{O}_1$, the WIT. Those two methods provide almost the same rate-energy region.

3) Time Consumption: The average time consumption (averaged over 10,000 channels) of the conventional and DNN-based solutions is compared with conventional methods in Fig. 12. All objectives are implemented on the same CPU, channel by channel. For $\mathcal{O}_1$ and $\mathcal{O}_2$, conventional methods are more efficient since they are analytical solutions and are faster than the DNN-based precoder. The execution time for the analytical solution is calculated based on the average over 10,000 channel realization. The advantage of the DNN-based precoder will appear on other objectives where only numerical solutions exist. The conventional methods require around 16ms to 27ms to achieve solutions while the proposed DNN-based precoding needs less than 1.5ms. On average, it saves 91.91%

Authorized licensed use limited to: VILLANOVA UNIVERSITY. Downloaded on July 24,2021 at 02:23:56 UTC from IEEE Xplore. Restrictions apply.

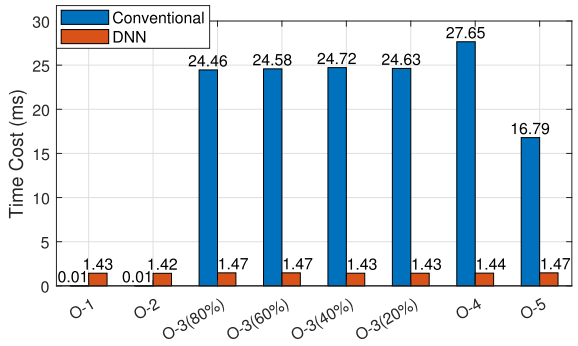


Fig. 12. The time cost of each objective.

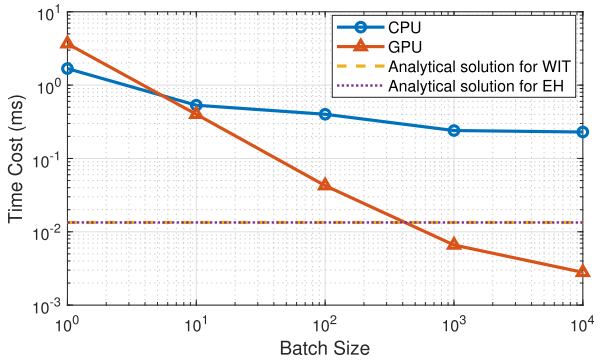


Fig. 13. Average execution time over different batch sizes.

of run-time if we assume that the four objectives occur with the same probability.

DNN-based precoding is beneficial for complicated equipment, such as base stations that serve a large number of users at the same time, where GPUs are affordable for more than one channel. The average time will reduce dramatically attributed to the parallel computing ability of GPU. We use the *batch size* denoting the number of channels or objectives processed simultaneously. Fig. 13 reveals the relation between batch size and time computation using CPU and GPU. For example, when batch size is 100, the time cost is 0.4007ms on CPU and 0.0426ms on GPU. Then, the computational load can be largely reduced. The time efficiency of DNN using GPU exceeds analytical solutions when processing more than 1000 realizations in one batch. It worth mentioning that the type of our GPU is Intel(R) Core(TM) i5-9400 @ 2.90GHz, and CPU is NVIDIA GTX1660Ti.

We understand that generating data sets and training the network is time-consuming. However, these procedures are performed only once and in an offline manner. Once this is done, the DNN-based precoder becomes a matrix multiplication that can be used as long as the assumptions on the channels are valid. On the other hand, for the conventional solution the optimization problem corresponding to each objective needs to be solved for each input channel independently. So, the DNN approach shows its advantages in long-term usages.

E. The Scale of DNN-Based Precoding

In this part, we evaluate the performance of the DNN-based precoding associate with the depth (number of hidden layers)

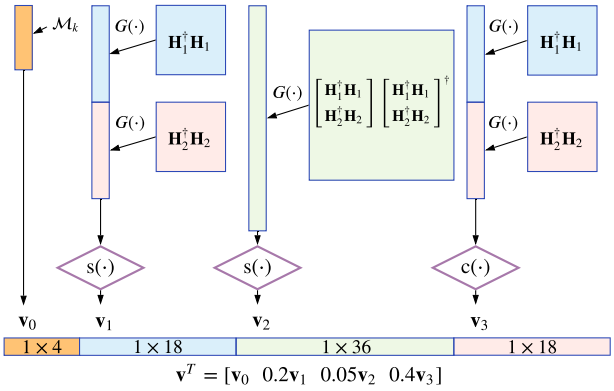


Fig. 14. Feature design for the complex channels.

and the width (number of hidden nodes) of the network. The proposed DNN has ten layers in which nine of them have 256 hidden nodes and the last one has 128 hidden nodes. So we define this network as 10 in depth and 256 in width. Other networks are trained and tested in the same way as we have done in the previous experiments. The performance is listed in Table VI, including the accuracy in (37) and average time cost. For $\mathcal{O}_1$ and $\mathcal{O}_3$, all of the networks work well and close to the RP. The performance and the time cost of $\mathcal{O}_2$, $\mathcal{O}_4$, and $\mathcal{O}_5$ are positively related to depth and width. As a balance of solution quality and time consumption, we choose the depth as 10 and width as 256 in this paper, even though better performance can be achieved using a deeper and wider network.

The performance of the DNN-based precoder is evaluated for different numbers of hidden layers (*depth*) and hidden nodes (*width*). Increasing the depth and width increases the performance but increases the computational complexity.

F. DNN-Based Precoder for Complex Channels

With a slight modification of its input and output, the proposed DNN-based precoder can be applied to complex channels. The complex communication system with two users can be described by (3a)-(3b) where $\mathbf{x} \in \mathbb{C}^{m\times 1}$, $\mathbf{H}_u \in \mathbb{C}^{n_u\times m}$, and $\mathbf{y}_u, \mathbf{w}_u, \in \mathbb{C}^{n_u\times 1}$ $u \in \{1, 2\}$ for user index. The covariance matrix $\mathbf{Q} \triangleq \mathbb{E}\{\mathbf{x}\mathbf{x}^\dagger\}$ is also complex, where $(\cdot)^\dagger$ represents Hermitian transpose. The objective functions will remain the same, except that coefficient $\frac{1}{2}$ will be removed from all capacity equations. So, the basic feature can still be $\mathbf{H}_u^\dagger\mathbf{H}_u$. Different from the real system case, $\mathbf{H}_u^\dagger\mathbf{H}_u$ is a complex and Hermitian. So, we modify the vectorization when building the input vector in Fig. 14. Compared with the input feature design for a real system, the only two differences are $\mathbf{H}_u^\dagger\mathbf{H}_u$ and the vectorization $G(\cdot)$, which is defined as

$$\mathbf{a} = G(\mathbf{A}) \triangleq \text{vec}\{\text{Re}[U(\mathbf{A})] - \text{Im}[L(\mathbf{A})]\}, \tag{38}$$

where vec$(\cdot)$ vectorizes a real matrix, $U(\mathbf{A})$ remains the upper triangular matrix of $\mathbf{A}$, and $L(\mathbf{A})$ keeps the lower triangular matrix. The output of the DNN precoder is design to be $\mathbf{q} = G(\mathbf{Q})$. The complex system needs extra hidden nodes or additional layers in the network. The training data

Authorized licensed use limited to: VILLANOVA UNIVERSITY. Downloaded on July 24,2021 at 02:23:56 UTC from IEEE Xplore. Restrictions apply.

TABLE VI
THE ACCURACY (%) AND TIME CONSUMPTION (MS) OF THE DNN-BASED PRECODING WITH DIFFERENT DEPTHS AND WIDTHS

Width	Depth	Objective								Time Cost
		$\mathcal{O}_1$	$\mathcal{O}_2$	$\mathcal{O}_3(80\%)$	$\mathcal{O}_3(60\%)$	$\mathcal{O}_3(40\%)$	$\mathcal{O}_3(20\%)$	$\mathcal{O}_4$	$\mathcal{O}_5$	(ms)
128	10	99.91	99.52	99.87	99.94	99.90	99.91	89.65	98.62	0.94
256	6	99.92	99.58	99.79	99.93	99.94	99.90	90.69	98.69	1.39
256	**10**	**99.97**	**99.83**	**99.95**	**99.97**	**99.97**	**99.96**	**94.38**	**99.13**	**1.44**
256	14	99.98	99.91	99.95	99.98	99.99	99.97	96.75	99.15	2.03
512	10	99.98	99.90	99.97	99.98	99.98	99.98	96.41	99.46	2.39

TABLE VII
THE ACCURACY (%) OF THE DNN-BASED PRECODING FOR COMPLEX CHANNELS

Test Set Distribution	Width	Depth	Objective							
			$\mathcal{O}_1$	$\mathcal{O}_2$	$\mathcal{O}_3(80\%)$	$\mathcal{O}_3(60\%)$	$\mathcal{O}_3(40\%)$	$\mathcal{O}_3(20\%)$	$\mathcal{O}_4$	$\mathcal{O}_5$
$\mathcal{CN}(0,1)$	256	10	99.95	99.63	99.90	99.95	99.95	99.95	85.95	99.27
	256	14	99.97	99.77	99.90	99.97	99.97	99.96	90.68	99.26
	512	10	99.98	99.83	99.98	99.98	99.98	99.98	90.99	99.50
$\mathcal{CN}(0,0.25)$	256	10	99.95	99.61	99.93	99.98	99.94	99.95	89.72	99.03
	256	14	99.97	99.77	99.83	100.00	99.95	99.95	92.87	99.01
	512	10	99.98	99.82	99.96	99.98	99.97	99.98	93.57	99.35

set needs to be regenerated based on complex channels and the corresponding solutions for each objective function. The learning rate dropping factor is 0.85; all the other training hyperparameters are the same as the DNN for real channels, given in Table VI. The experiment result is shown in Table VII. It can be seen that the performance of the DNN precoder is still very close to that of the conventional solutions, except for $\mathcal{O}_4$. $\mathcal{O}_4$ is unique since it is a non-convex problem and the conventional solutions may be terminated at a local minimum. This will lead to a mapping that is hard to learn and part of the training set may mislead the DNN.

To improve the performance of the DNN precoder on $\mathcal{O}_4$, one option is to increase the number of hidden nodes or layers, which is reasonable since the complex relation is more difficult to learn and we need three more output values compared to the real system DNN precoder. As we can see in rows 1-3 in Table VII, the accuracy of DNN improves with additional width or depth. Another way is to train DNN with higher variance inputs and test with a lower variance. Compare to rows 1-3, the performance of DNN is improved for rows 4-6, for $\mathcal{O}_4$. Rows 4-6 are tested with channels in a narrower region than the training set. The reason is, DNN precoder performs a set of regression tasks. Testing in a shrunk range will not be affected by the marginal effects.

VI. CONCLUSION

In this paper, a unified DNN-based precoder has been proposed for green, secure wireless transmission in two-user MIMO systems. Specifically, WIT, EH, SWIPT, PHY security, and multicasting problems have been considered. We first use rotation-based precoding to derive the transmit covariance matrix for the above problems from which the DNN learns. The overall performance of the rotation-based precoder is better than the existing methods for SWIPT, PHY security, and multicasting. These conventional methods based on the mathematical models are used for data set generation and training procedure of DNN. Next, a DNN-based precoder is designed to unify the solutions for different objectives. This DNN-based precoding can effectively optimize all objectives at the same time. In terms of achievable rates and harvested energy, The performance of the unified DNN-based precoder is similar to the method it learns from, whereas its time cost is substantially lower than the conventional iterative solutions. Due to its lower computational complexity and its high flexibility, the proposed precoding is suitable for emerging existing applications, where the low-latency and low-complexity devices are necessary.

REFERENCES

[1] X. Zhang and M. Vaezi, "A DNN-based multi-objective precoding for Gaussian MIMO networks," in *Proc. GLOBECOM - IEEE Global Commun. Conf.*, Dec. 2020.

[2] Statista. (2020). *COVID-19 Impact on Digital Communications in the U.S. 2020*. [Online]. Available: https://www.statista.com/topics/6241/coronavirus-impact-on-online-usage-in-the-us

[3] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. 2nd ed. Hoboken, NJ, USA: Wiley, 2006.

[4] R. Zhang and C. K. Ho, "MIMO broadcasting for simultaneous wireless information and power transfer," *IEEE Trans. Wireless Commun.*, vol. 12, no. 5, pp. 1989–2001, May 2013.

[5] J. Rostampoor, S. M. Razavizadeh, and I. Lee, "Energy efficient precoding design for SWIPT in MIMO two-way relay networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 9, pp. 7888–7896, Sep. 2017.

[6] S. A. A. Fakoorian and A. L. Swindlehurst, "Optimal power allocation for GSVD-based beamforming in the MIMO Gaussian wiretap channel," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2012, pp. 2321–2325.

[7] Q. Li, M. Hong, H.-T. Wai, Y.-F. Liu, W.-K. Ma, and Z.-Q. Luo, "Transmit solutions for MIMO wiretap channels using alternating optimization," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 1714–1727, Sep. 2013.

[8] X. Zhang, Y. Qi, and M. Vaezi, "A rotation-based method for precoding in Gaussian MIMOME channels," *IEEE Trans. Commun.*, vol. 69, no. 2, pp. 1189–1200, Feb. 2021.

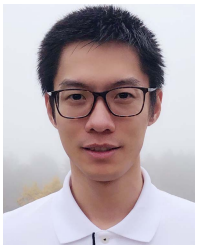
[9] M. Vaezi, W. Shin, and H. V. Poor, "Optimal beamforming for Gaussian MIMO wiretap channels with two transmit antennas," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6726–6735, Oct. 2017.

[10] J. Zhang, C. Yuen, C.-K. Wen, S. Jin, K.-K. Wong, and H. Zhu, "Large system secrecy rate analysis for SWIPT MIMO wiretap channels," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 1, pp. 74–85, Jan. 2016.

[11] N. D. Sidiropoulos, T. N. Davidson, and Z.-Q. Luo, "Transmit beamforming for physical-layer multicasting," *IEEE Trans. Signal Process.*, vol. 54, no. 6, pp. 2239–2251, Jun. 2006.

Authorized licensed use limited to: VILLANOVA UNIVERSITY. Downloaded on July 24,2021 at 02:23:56 UTC from IEEE Xplore. Restrictions apply.

[12] H. Zhu, N. Prasad, and S. Rangarajan, "Precoder design for physical layer multicasting," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5932–5947, Nov. 2012.
[13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
[14] Z. Li, S. Gong, C. Xing, Z. Fei, and X. Yan, "Multi-objective optimization for distributed MIMO networks," *IEEE Trans. Commun.*, vol. 65, no. 10, pp. 4247–4259, Oct. 2017.
[15] N. Shanin, L. Cottatellucci, and R. Schober, "Rate-power region of SWIPT systems employing nonlinear energy harvester circuits with memory," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–7.
[16] R. Fritschek, R. F. Schaefer, and G. Wunder, "Deep learning for the Gaussian wiretap channel," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.
[17] X. Zhang and M. Vaezi, "Deep learning based precoding for the MIMO Gaussian wiretap channel," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2019.
[18] F.-L. Luo, *Machine Learning for Future Wireless Communications*. Hoboken, NJ, USA: Wiley, 2020.
[19] W. Lee, "Resource allocation for multi-channel underlay cognitive radio network based on deep neural network," *IEEE Commun. Lett.*, vol. 22, no. 9, pp. 1942–1945, Sep. 2018.
[20] K. N. Doan, M. Vaezi, W. Shin, H. V. Poor, H. Shin, and T. Q. S. Quek, "Power allocation in cache-aided NOMA systems: Optimization and deep reinforcement learning approaches," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 630–644, Jan. 2020.
[21] S. D'Oro, A. Zappone, S. Palazzo, and M. Lops, "A learning approach for low-complexity optimization of energy efficiency in multicarrier wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3226–3241, May 2018.
[22] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognit. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.
[23] X. Li and A. Alkhateeb, "Deep learning for direct hybrid precoding in millimeter wave massive MIMO systems," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, Nov. 2019, pp. 800–805.
[24] S. Fan, H. Tian, and C. Sengul, "Self-optimization of coverage and capacity based on a fuzzy neural network with cooperative reinforcement learning," *EURASIP J. Wireless Commun. Netw.*, vol. 2014, no. 1, p. 57, Dec. 2014.
[25] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, Jan. 1989.
[26] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," 2016, *arXiv:1611.01578*. [Online]. Available: http://arxiv.org/abs/1611.01578
[27] T. Liu and S. Shamai, "A note on the secrecy capacity of the multiple-antenna wiretap channel," *IEEE Trans. Inf. Theory*, vol. 55, no. 6, pp. 2547–2553, Jun. 2009.
[28] P. H. Tan, J. Joung, and S. Sun, "Opportunistic multicast scheduling for unicast transmission in MIMO-OFDM system," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 3522–3527.
[29] M. Vaezi, Y. Qi, and X. Zhang, "A rotation-based precoding for MIMO broadcast channels with integrated services," *IEEE Signal Process. Lett.*, vol. 26, no. 11, pp. 1708–1712, Nov. 2019.
[30] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD, USA: Johns Hopkins Univ. Press, 2012.
[31] H. Weingarten, Y. Steinberg, and S. Shamai, "The capacity region of the Gaussian MIMO broadcast channel," in *Proc. Int. Symp. Inf. Theory (ISIT)*, 2004, p. 174.
[32] E. Ekrem and S. Ulukus, "The secrecy capacity region of the Gaussian MIMO multi-receiver wiretap channel," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 2083–2114, Apr. 2011.
[33] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.
[34] J. Sola and J. Sevilla, "Importance of input data normalization for the application of neural networks to complex industrial problems," *IEEE Trans. Nucl. Sci.*, vol. 44, no. 3, pp. 1464–1468, Jun. 1997.
[35] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks Trade* (Lecture Notes in Computer Science), vol. 7700. Berlin, Germany: Springer, 2012, pp. 9–48.
[36] M. Grant and S. Boyd. (Mar. 2014). *CVX: MATLAB Software for Disciplined Convex Programming, Version 2.1*. [Online]. Available: http://cvxr.com/cvx
[37] D. Park, "Weighted sum rate maximization of MIMO broadcast and interference channels with confidential messages," *IEEE Trans. Wireless Commun.*, vol. 15, no. 3, pp. 1742–1753, Mar. 2016.
[38] J. Nocedal and S. Wright, *Numerical optimization*. New York, NY, USA: Springer, 2006.
[39] Y. Labit, D. Peaucelle, and D. Henrion, "SEDUMI INTERFACE 1.02: A tool for solving LMI problems with SEDUMI," in *Proc. IEEE Int. Symp. Comput. Aided Control Syst. Design*, Sep. 2002, pp. 272–277.
[40] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2016.
[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[42] P. Goyal *et al.*, "Accurate, large minibatch SGD: Training ImageNet in 1 hour," 2017, *arXiv:1706.02677*. [Online]. Available: http://arxiv.org/abs/1706.02677
[43] N. Shirish Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," 2016, *arXiv:1609.04836*. [Online]. Available: http://arxiv.org/abs/1609.04836



Xinliang Zhang (Graduate Student Member, IEEE) received the B.Eng. and M.Eng. degrees in electrical engineering from Xidian University, Xi'an, China, in 2015 and 2018, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Villanova University, Villanova, PA, USA. His research interests include physical layer security, machine learning for wireless communications, MIMO networks, and signal processing.

Mojtaba Vaezi (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from the Amirkabir University of Technology (Tehran Polytechnic) and the Ph.D. degree from McGill University, all in electrical engineering.

From 2015 to 2018, he was with Princeton University as a Post-Doctoral Research Fellow and an Associate Research Scholar. Before joining Princeton, he was a Researcher with the Ericsson Research, Montreal, QC, Canada. He is currently an Assistant Professor of Electrical and Computer Engineering (ECE) with Villanova University. His research interests include the broad areas of signal processing and machine learning for wireless communications with an emphasis on physical layer security and fifth-generation (5G) and beyond radio access technologies. Among his publications in these areas is the book *Multiple Access Techniques for 5G Wireless Networks and Beyond* (Springer, 2019). He has co-organized six NOMA workshops at IEEE VTC 2017-Spring, Globecom'17, 18, and ICC'18, 19, 20. He was a recipient of several academic, leadership, and research awards, including the McGill Engineering Doctoral Award, the IEEE Larry K. Wilson Regional Student Activities Award in 2013, the Natural Sciences and Engineering Research Council of Canada (NSERC) Postdoctoral Fellowship in 2014, the Ministry of Science and ICT of Korea's Best Paper Award in 2017, the IEEE Communications Letters Exemplary Editor Award in 2018, and the 2020 IEEE Communications Society Fred W. Ellersick Prize. He is also an Editor of IEEE TRANSACTIONS ON COMMUNICATIONS and IEEE COMMUNICATIONS LETTERS.

Authorized licensed use limited to: VILLANOVA UNIVERSITY. Downloaded on July 24,2021 at 02:23:56 UTC from IEEE Xplore. Restrictions apply.