Benchmarking and Interpreting End-to-end Learning of MIMO and Multi-User Communication

Citation for the original published paper (version of record):

Song, J., Häger, C., Schröder, J. et al (2022). Benchmarking and Interpreting End-to-end Learning of MIMO and Multi-User Communication. IEEE Transactions on Wireless Communications, 21(9): 7287-7298. http://dx.doi.org/10.1109/TWC.2022.3157467

N.B. When citing this work, cite the original published paper.

(article starts on next page)

Benchmarking and Interpreting End-to-End Learning of MIMO and Multi-User Communication

Jinxiang Song, *Graduate Student Member, IEEE*, Christian Häger, *Member, IEEE*,
Jochen Schröder, *Member, IEEE*, Timothy J. O'Shea, *Senior Member, IEEE*,
Erik Agrell, *Fellow, IEEE*, and Henk Wymeersch, *Senior Member, IEEE*

Abstract—**End-to-end autoencoder (AE) learning has the potential of exceeding the performance of human-engineered transceivers and encoding schemes, without a priori knowledge of communication-theoretic principles. In this work, we aim to understand to what extent and for which scenarios this claim holds true when comparing with fair benchmarks. Our particular focus is on memoryless multiple-input multiple-output (MIMO) and multi-user (MU) systems. Four case studies are considered: two point-to-point (closed-loop and open-loop MIMO) and two MU scenarios (MIMO broadcast and interference channels). For the point-to-point scenarios, we explain some of the performance gains observed in prior work through the selection of improved baseline schemes that include geometric shaping as well as bit and power allocation. For the MIMO broadcast channel, we demonstrate the feasibility of a novel AE method with centralized learning and decentralized execution. Interestingly, the learned scheme performs close to nonlinear vector-perturbation precoding and significantly outperforms conventional zero-forcing. Lastly, we highlight potential pitfalls when interpreting learned communication schemes. In particular, we show that the AE for the considered interference channel learns to avoid interference, albeit in a rotated reference frame. After de-rotating the learned signal constellation of each user, the resulting scheme corresponds to conventional time sharing with geometric shaping.**

Index Terms—**Autoencoders, deep learning, digital signal processing, end-to-end learning, interference channel, machine learning, MIMO broadcast, wireless communications.**

I. INTRODUCTION

DEMAND for higher data rates has led to the continued development of ever more performant wireless communication systems. One of the most important developments has been multiple-input multiple-output (MIMO) transmission [1], where information across multiple antenna elements is encoded using spatial-multiplexing or spatial-diversity schemes to enhance throughput and reliability of communication systems. Conventional MIMO communication systems are often classified as closed-loop or open-loop. In open-loop systems, channel state information (CSI) is only available at the receiver, while in closed-loop systems, the transmitter also has access to CSI (either through explicit feedback or via channel reciprocity). Several approaches have been used for both open-loop and closed-loop systems, including maximum-likelihood detection, zero-forcing (ZF) precoding, minimum mean-square-error (MMSE) equalization, space-time block coding, and singular value decomposition (SVD) with waterfilling [2, Chapter 11].

Recent years have witnessed a resurgence of interest in machine-learning (ML) techniques for communication systems. Most work has focused on supervised learning for *specific functional blocks* such as modulation recognition [3], MIMO detection [4]–[7], MIMO channel estimation [8], and channel decoding [9], [10]. These ML-based methods have led to algorithms that often perform better or exhibit lower complexity than model-based algorithms. In contrast to focusing on specific functional blocks, *end-to-end learning* has been proposed to optimize the transmitter and receiver jointly [11]. The workhorse of end-to-end learning is the autoencoder (AE), which employs two neural networks (NNs) to encode and decode messages into a learned latent representation which passes through a physical communication channel. This method has been successfully applied to a wide variety of channels, including, e.g., linear wireless [12], [13], and nonlinear optical [14], [15] ones. In cases where no differentiable channel model is available, a surrogate channel can first be learned [16], [17] or the transmitter can be designed as a reinforcement-learning agent [18], which can be trained even with limited reward feedback [19].

In this paper, we consider the application of end-to-end learning to MIMO systems assuming both point-to-point and multi-user (MU) transmission scenarios. For these applications, there has been limited treatment of AEs. In [20], open-loop and closed-loop MIMO were studied, leading to better performance than the selected benchmark methods. In the extension [21], finite quantization of the CSI was considered, which was demonstrated to further improve performance under some conditions. While [20], [21] have shown promising

Jinxiang Song, Christian Häger, Erik Agrell, and Henk Wymeersch are with the Department of Electrical Engineering, Chalmers University of Technology, 41296 Gothenburg, Sweden (e-mail: jinxiang@chalmers.se; christian.haeger@chalmers.se; agrell@chalmers.se; henkw@chalmers.se).

Jochen Schröder is with the Department of Microtechnology and Nanoscience, Chalmers University of Technology, 41296 Gothenburg, Sweden (e-mail: jochen.schroeder@chalmers.se).

Timothy J. O'Shea is with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061 USA, and also with DeepSig, Inc., Arlington, VA 22203 USA (e-mail: oshea@vt.edu).

TABLE I

Overview of the Considered Scenarios, Best-Performing Baseline Schemes, and High-Level Conclusions in This and Previous Works. (AE: Autoencoder, STBC: Space-Time Block Code, SVD: Singular Value Decomposition, GS: Geometric Shaping)

	(i) open-loop MIMO	(ii) closed-loop MIMO	(iii) MIMO broadcast	(iv) interference channel
reference(s)	[20], [21]	[20], [21]	[22]*	[11]
baseline	Alamouti STBC w/ QAM	SVD w/ QPSK and equal power	Tomlinson–Harashima	time sharing w/ QAM
conclusion	*AE outperforms baseline*	*AE outperforms baseline*	*AE outperforms baseline*	*AE outperforms baseline*
our baseline	Alamouti STBC w/ GS	SVD w/ GS, bit & power loading	vector-perturbation precoding	time sharing w/ GS
our conclusion	*AE matches baseline*	*baseline outperforms AE†*	*baseline outperforms AE*	*baseline outperforms AE*

*Independently proposed in this paper (preliminary results presented in [23]).
†AE outperforms baseline for certain channel singular values, see Sec. IV-C.

performance of AE-based MIMO communication, the proposed systems were trained under some nonstandard assumptions, specifically regarding CSI availability at the receiver and power normalization at the transmitter, as explained in more detail below. Besides [20], [21], MIMO AEs were also studied in [23], [24] for the noncoherent case, where neither the transmitter nor the receiver has access to CSI. Regarding MU communication, the authors in [11] have shown that the AE framework can be extended to include multiple transmitter–receiver pairs. They considered a conventional Gaussian interference channel and showed that the learned communication scheme achieves better performance than the selected time-sharing baseline.

In this paper, we build on the approaches proposed in [11], [20], [21] with the aim to better understand what performance gains can be achieved by AE-based MIMO and MU systems under more realistic training assumptions when compared to fair benchmarks. To that end, the channel models considered are assumed to be memoryless, as in [11], [20], [21]. Moreover, we also provide additional interpretations of the learned communication schemes. A particular emphasis in this work is placed on selecting baseline schemes with geometric shaping (GS), see, e.g., [25]. Shaped modulation formats for Gaussian channels are also readily available in open databases [26]. Our main contributions in this work are as follows:

- For the MIMO systems in [20], [21], we analyze and evaluate the corresponding AEs under more standard training assumptions. In particular, while CSI in [20], [21] was assumed to be estimated at the receiver, it was not actually used as a receiver input. Moreover, power normalization was applied after the channel-matrix multiplication (cf. [21, Eqs. (2), (3)]), which cannot be done in practical systems. By contrast, our AEs always use the CSI as an additional receiver input and power normalization is performed prior to the channel. Additionally, reproducible open-source implementations of our AEs and benchmark schemes are also provided.[1]

- We then explain some of the performance gains obtained by the trained AEs through the selection of improved baseline schemes compared to [20], [21]. In particular, for open-loop MIMO, we show that previously observed

performance gains can be partially attributed to an implicit GS of the underlying signal constellation. For closed-loop MIMO, we use an SVD-based benchmark similar to [20], [21], but augment it through GS as well as additional bit and power allocation. This closes the performance gap to the AE, indicating that the ML-based solution learns to implement similar functionalities in a data-driven fashion.

- We then propose and analyze a novel AE system for a MIMO broadcast scenario, where a single multi-antenna transmitter sends information to multiple single-antenna users.[2] For such a system, we extend the training methodology in [21] to account for the joint loss function of all users. The resulting AE is shown to provide performance between nonlinear vector-perturbation precoding [27] and conventional transmitter ZF, significantly outperforming the latter over a wide range of signal-to-noise ratios (SNRs). In parallel to our work, a similar scenario was also recently considered in [22]. This work is discussed in more detail in Sec. V-C.

- Lastly, we revisit the interference-channel scenario in [11] where significant performance gains were demonstrated by AE-based communication compared to the considered time-sharing baseline scheme. After augmenting the time-sharing scheme with a geometrically-shaped signal constellation, we find that the improved baseline performs similarly to (and in some cases even better than) the AE. The improved baseline also allows us to provide an additional theoretical interpretation of the learned AE scheme in terms of a "rotated" time-sharing scheme.

An overview of the considered scenarios including the best-performing baseline schemes and high-level conclusions can be found in Table I. We note that the underlying assumptions for each scenario (e.g., the fading model or the number of transmit/receive antennas) are consistent with prior work, which allows us to make direct comparisons to the corresponding results. We comment on some of the limitations of these assumptions in Sec. VII.

The remainder of the paper is structured as follows. In Sec. II, a brief introduction to AE-based communication is given. The four scenarios listed in Table I are then studied in Secs. III (open-loop MIMO), IV (closed-loop MIMO),

[1]The complete source code to reproduce all results in this paper is available at https://github.com/JSChalmers/DeepLearning_MIMO.git

[2]This scenario was suggested as a possible extension in [21, Sec. V].

V (MIMO broadcast), and VI (interference channel), where each section contains a detailed description of the baseline scheme(s), AE implementation, as well as numerical results and a discussion. Finally, the paper is concluded in Sec. VII.

Notation: $\mathbb{Z}$, $\mathbb{R}$, and $\mathbb{C}$ denote the sets of integers, real numbers, and complex numbers, respectively. We use boldface letters to denote vectors and matrices (e.g., $\boldsymbol{x}$ and $\boldsymbol{A}$). $(\cdot)^{\mathsf{T}}$ and $(\cdot)^{\mathsf{H}}$ denote transpose and conjugate transpose, respectively. For a vector $\boldsymbol{x} = [x_1, \ldots, x_n]^{\mathsf{T}}$, $[\boldsymbol{x}]_i = x_i$ returns the i-th element of $\boldsymbol{x}$, $\|\boldsymbol{x}\|^2 = \sum_{i=1}^{n} |x_i|^2$ denotes the squared Euclidean norm, and $\mathrm{diag}(\boldsymbol{x})$ is the matrix whose diagonal entries are the elements of $\boldsymbol{x}$. A matrix $\boldsymbol{X}$ is converted to a vector by stacking the columns, which is denoted by $\mathrm{vec}(\boldsymbol{X})$. $\boldsymbol{I}_n$ is the $n \times n$ identity matrix. $[a, b]^M$ is the M-fold Cartesian product of the interval $[a, b]$. $\mathcal{CN}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the distribution of a proper complex Gaussian random vector with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, evaluated at $\boldsymbol{x}$ ($\boldsymbol{x}$ may be omitted to represent the entire distribution). Lastly, $\mathbb{E}\{\cdot\}$ denotes expected value.

II. AUTOENCODER-BASED COMMUNICATION SYSTEMS

In this section, we start by briefly reviewing AE-based communication assuming transmission over the memoryless (complex-valued) additive white Gaussian noise (AWGN) channel

$$\boldsymbol{y}_k = \boldsymbol{x}_k + \boldsymbol{n}_k, \tag{1}$$

where $\boldsymbol{x}_k, \boldsymbol{y}_k \in \mathbb{C}^{N_B}$ are the channel input and output vector in the k-th transmission block, N_B denotes the number of channel uses per block, and $\boldsymbol{n}_k \sim \mathcal{CN}(\boldsymbol{0}, N_0 \boldsymbol{I}_{N_B})$ is independent and identically distributed (i.i.d.) Gaussian noise. The specific AE implementations for the considered MIMO and MU scenarios are then described in detail in the following sections.

A. Transmitter and Receiver Design

AE-based end-to-end learning was originally proposed in [11]. The general idea is to reinterpret the design of a communication system as a reconstruction task that jointly optimizes parameterized transmitters and receivers. To that end, the transceiver for the AWGN channel (1) can be implemented by a pair of NNs $f_\tau : \mathcal{M} \to \mathbb{C}^{N_B}$ and $f_\rho : \mathbb{C}^{N_B} \to [0, 1]^M$, where $\mathcal{M} = \{1, 2, \ldots, M\}$ is the message set and τ and ρ are the transmitter and receiver NN parameters, respectively. More precisely, a message $m_k \in \mathcal{M}$ is first encoded as an M-dimensional "one-hot" vector, where the m_k-th element is 1 and all the others are 0. This vector is then used as the input to the transmitter NN.[3] The NN is assumed to have $2N_B$ output neurons, which form the real and imaginary part of the unnormalized transmit vector denoted by $\tilde{\boldsymbol{x}}_k = \tilde{f}_\tau(m_k) \in \mathbb{C}^{N_B}$. The average transmit power is defined as $P_T = \mathbb{E}\{\|\boldsymbol{x}_k\|^2\}/N_B$ and enforced by a normalization layer such as

$$\boldsymbol{x}_k = \frac{\tilde{\boldsymbol{x}}_k \sqrt{N_B P_T}}{\sqrt{\frac{1}{M} \sum_{i=1}^{M} \|\tilde{f}_\tau(i)\|^2}}, \tag{2}$$

[3]In principle, other message encodings can also be used, see [11] for details, which are particularly important for large message sets.

where $\boldsymbol{x}_k = f_\tau(m_k)$ denotes the entire transmitter mapping. The vector $\boldsymbol{x}_k$ is then sent over the channel (1) and the receiver NN processes the received vector $\boldsymbol{y}_k$ by generating an M-dimensional probability vector $\boldsymbol{q}_k = f_\rho(\boldsymbol{y}_k)$, where the components of $\boldsymbol{q}_k$ can be interpreted as the estimated posterior probabilities of the messages. Finally, the transmitted message is estimated according to $\hat{m}_k = \arg\max_m [\boldsymbol{q}_k]_m$.

B. End-to-End Training Procedure

To optimize the transmitter and receiver parameters, it is important to have a suitable optimization criterion. Due to the fact that optimization relies on the empirical computation of gradients, a criterion like block error rate (BLER) $\Pr\{\hat{m}_k \neq m_k\}$ cannot be used directly. Instead, a commonly used criterion is the categorical cross-entropy loss function [11] defined by

$$\mathcal{J}_{\mathrm{CE}}(\tau, \rho) = -\mathbb{E}\{\log[f_\rho(\boldsymbol{y}_k)]_{m_k}\}, \tag{3}$$

where the dependence of $\mathcal{J}_{\mathrm{CE}}(\tau, \rho)$ on τ is implicit through the distribution of the channel output $\boldsymbol{y}_k$, which is a function of the channel input $f_\tau(m_k)$. This loss function is also adopted for all scenarios in this paper, either directly or in the form of a weighted average (for cases involving multiple users), as explained in detail below. In practice, $\mathcal{J}_{\mathrm{CE}}$ is usually approximated via Monte Carlo estimation. More specifically, a batch (or minibatch) of B samples is randomly chosen in each gradient step and $\mathcal{J}_{\mathrm{CE}}$ is approximated according to

$$\hat{\mathcal{J}}_{\mathrm{CE}} = -\frac{1}{B} \sum_{k=1}^{B} \log[f_\rho(\boldsymbol{y}_k)]_{m_k}. \tag{4}$$

Optimization of the NNs can be performed by minimizing $\hat{\mathcal{J}}_{\mathrm{CE}}$ through the widely used Adam optimizer [28], or a variety of similar stochastic gradient descent optimizers.

III. OPEN-LOOP MIMO

In this section, we consider an open-loop MIMO system where a transmitter with N_T antennas sends sequences of messages to a transmitter with N_R antennas. We note that for all scenarios in this paper, the information rate is always assumed to be fixed and forward error correcting coding is not considered.

A. Background and Baseline Schemes

The channel matrix at discrete time k is denoted by $\boldsymbol{H}_k \in \mathbb{C}^{N_R \times N_T}$. The channel is drawn from a stationary distribution and is assumed to be block fading with duration $N_B \geq N_T$. In open-loop systems, CSI is available at the receiver but not at the transmitter. Conventional transmit approaches for open-loop MIMO systems include space-time block codes (STBCs) [29]–[31], which are described next.

The transmitter generates L messages, maps each message to a data symbol $s_{k,l} \in \Omega$ from a complex signal constellation $\Omega \subset \mathbb{C}$, and then encodes $\boldsymbol{s}_k = [s_{k,1}, \ldots, s_{k,L}]^{\mathsf{T}}$ using a STBC with rate $L/N_B \leq 1$. The resulting N_B coded vectors of length N_T are denoted by $\boldsymbol{X}_k = [\boldsymbol{x}_{k,1}, \ldots, \boldsymbol{x}_{k,N_B}]$, with the property that $\mathbb{E}\{\boldsymbol{X}_k^{\mathsf{H}} \boldsymbol{X}_k\} = P_T \boldsymbol{I}_{N_B}$, where P_T is
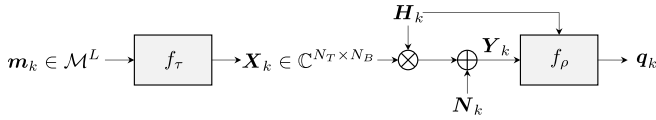
Fig. 1.   Open-loop MIMO channel AE, where the transmitter learns a rate L/N_B code without CSI, while the receiver learns a decoder in the presence of CSI. The channel is drawn i.i.d. from the underlying distribution.

the total average transmit power, summed over all transmit antennas. If each of the L complex data symbols corresponds to $\log_2(M)$ bits (i.e., one message), then the total bit rate is $r = L\log_2(M)/N_B$. The receiver observes

$$\boldsymbol{Y}_k = \boldsymbol{H}_k \boldsymbol{X}_k + \boldsymbol{N}_k, \tag{5}$$

where $\text{vec}(\boldsymbol{N}_k) \sim \mathcal{CN}(\boldsymbol{0}, N_0 \boldsymbol{I}_{N_R N_B})$ is i.i.d. Gaussian noise. The receiver then applies maximum-likelihood detection to $\boldsymbol{Y}_k = [\boldsymbol{y}_{k,1}, \ldots, \boldsymbol{y}_{k,N_B}]$ according to

$$\hat{\boldsymbol{s}}_k = \underset{\boldsymbol{s}_k \in \Omega^L}{\arg\min} \|\text{vec}(\boldsymbol{Y}_k - \boldsymbol{H}_k \boldsymbol{X}_k)\|^2, \tag{6}$$

which can be achieved through low-complexity linear processing [31]. Other (less complex) receiver approaches for open-loop MIMO include ZF and MMSE detection, which are not considered here as they are suboptimal.

In this paper, we restrict ourselves to the Alamouti STBC [29], where $N_T = 2$, $N_B = 2$, $L = 2$, with $r = \log_2(M)$. As an example, the Alamouti STBC for $N_R = 1$ is defined by the mapping

$$\boldsymbol{x}_{k,1} = [s_{k,1}, s_{k,2}]^\mathsf{T}, \tag{7}$$
$$\boldsymbol{x}_{k,2} = [-s_{k,2}^*, s_{k,1}^*]^\mathsf{T}. \tag{8}$$

At the receiver, one may first form a combination of the two received symbols $\boldsymbol{Y}_k = [y_{k,1}, y_{k,2}]$ according to

$$\tilde{s}_{k,1} = h_{k,1}^* y_{k,1} + h_{k,2} y_{k,2}^*, \tag{9}$$
$$\tilde{s}_{k,2} = h_{k,2}^* y_{k,1} - h_{k,1} y_{k,2}^*, \tag{10}$$

where $\boldsymbol{H}_k = [h_{k,1}, h_{k,2}]$ is the 1×2 channel matrix in this case. An optimal decision can then be made separately based on $\tilde{s}_{k,1}$ and $\tilde{s}_{k,2}$.

B. Autoencoder Design and Training

For an open-loop MIMO system with CSI available to the receiver, the AE implementation is visualized in Fig. 1. The transmitter $f_\tau : \mathcal{M}^L \to \mathbb{C}^{N_T \times N_B}$ maps L consecutive messages $\boldsymbol{m}_k = [m_1, \ldots, m_L]^\mathsf{T} \in \mathcal{M}^L$ to N_B coded vectors according to

$$\boldsymbol{X}_k = [\boldsymbol{x}_{k,1}, \ldots, \boldsymbol{x}_{k,N_B}] = f_\tau(\boldsymbol{m}_k), \tag{11}$$

where $\boldsymbol{x}_{k,p}$, $p = 1, \ldots, N_B$, is a column vector of length N_T. An average power constraint according to $\sum_{p=1}^{N_B} \mathbb{E}\{\|\boldsymbol{x}_{k,p}\|^2\} = N_B P_T$ is enforced through a normalization layer similar to (2). Inside $f_\tau(\cdot)$, an encoding of $\boldsymbol{m}_k$ to an M^L-dimensional one-hot vector is used. The receiver $f_\rho : \mathbb{C}^{N_R \times N_B} \times \mathbb{C}^{N_R \times N_T} \to [0,1]^{M^L}$ observes

TABLE II
NN PARAMETERS FOR (I) OPEN-LOOP MIMO, (II) CLOSED-LOOP MIMO, (III) MIMO BROADCAST, AND (IV) INTERFERENCE CHANNEL

			transmitter(s) f_τ/f_{τ_i}			receiver(s) f_ρ/f_{ρ_i}	
	layer	input	hidden	output	input	hidden	output
(i)	# of layers	-	3	-	-	3	-
	# of neurons	M^2	64	8	12	512	M^2
	act. function	-	ReLU	linear	-	ReLU	softmax
(ii)	# of layers	-	3	-	-	3	-
	# of neurons	$M+8$	1024	8	16	1024	M
	act. function	-	ReLU	linear	-	ReLU	softmax
(iii)	# of layers	-	3	-	-	3	-
	# of neurons	M^2+8	512	8	6	256	M
	act. function	-	ReLU	linear	-	ReLU	softmax
(iv)	# of layers	-	1	-	-	1	-
	# of neurons	M	256	8	8	256	M
	act. function	-	ReLU	linear	-	ReLU	softmax

TABLE III
TRAINING PARAMETERS FOR (I) OPEN-LOOP MIMO, (II) CLOSED-LOOP MIMO, (III) MIMO BROADCAST, AND (IV) INTERFERENCE CHANNEL

	(i)	(ii)	(iii)	(iv)
optimizer	Adam	Adam	Adam	Adam
learning rate	10^{-3}	10^{-3}	10^{-3}	10^{-3}
batch size B	65536	10240	10240	65536
grad. steps G	2×10^4	2×10^6	4×10^5	5×10^4
training SNR	15 dB ($M=4$) 18 dB ($M=16$)	5 dB, 10 dB 15 dB, 12 dB	12 dB, 15 dB 18 dB, 20 dB	11 dB (E_b/N_0)

$\boldsymbol{Y}_k = [\boldsymbol{y}_{k,1}, \cdots \boldsymbol{y}_{k,N_B}]$ as in (5) and generates a probability vector $\boldsymbol{q}_k \in [0,1]^{M^L}$ according to

$$\boldsymbol{q}_k = f_\rho(\boldsymbol{Y}_k, \boldsymbol{H}_k), \tag{12}$$

in which both the CSI $\boldsymbol{H}_k$ and the observation matrix $\boldsymbol{Y}_k$ are provided to the receiver. In our implementation, the CSI is first converted to a real-valued vector of length $2N_R N_T$ and then concatenated to the observation matrix, which is also converted to a real-valued vector. Finally, an estimate of the transmitted message vector $\hat{\boldsymbol{m}}_k$ can be obtained based on $\arg\max_m [\boldsymbol{q}_k]_m$ by inverting the one-hot encoding.

C. Numerical Results and Discussion

The channel is assumed to be Rayleigh fading, i.e., $\text{vec}(\boldsymbol{H}_k) \sim \mathcal{CN}(\boldsymbol{0}, \boldsymbol{I}_{N_R N_T})$. The system performance is measured in terms of the BLER $= \Pr\{\hat{\boldsymbol{m}}_k \neq \boldsymbol{m}_k\}$ as a function of the average SNR $= P_T/(N_T N_0)$. We use the parameters $N_T = 2$, $N_R = 1$, $N_B = 2$, $L = 2$, and $M \in \{4, 16\}$. In this paper, all AEs are implemented as multi-layer fully-connected NNs, where the rectified linear unit (ReLU) is chosen as the activation function. To optimize the number of hidden layers and the number of neurons per layer, several AEs with different sizes are trained for each of the considered scenarios, and we then choose the AEs with the best performance. The resulting NN parameters for all scenarios are summarized in Table II.[4] Moreover, all AEs

[4]We remark that the NN parameters used in this paper are not guaranteed to be fully optimal, though further optimization of the NN parameters are not expected to improve the AE performance significantly.
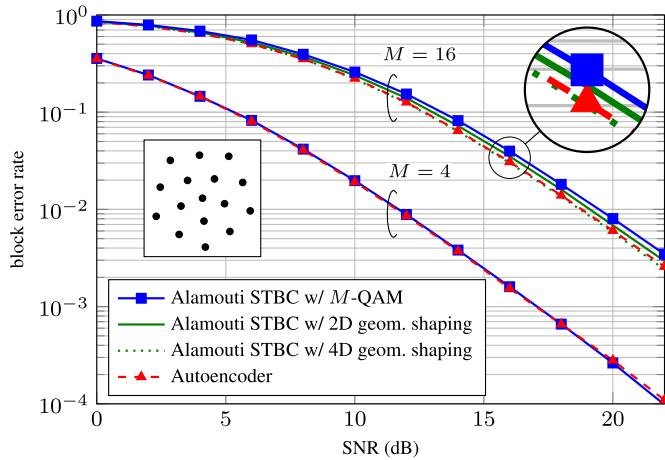
Fig. 2. BLER of the open-loop MIMO AE and the baseline scheme consisting of standard M-QAM signal constellations, an Alamouti STBC, and a maximum-likelihood receiver. The improved baselines for $M = 16$ use geometrically-shaped signal constellations in two and four dimensions, respectively.
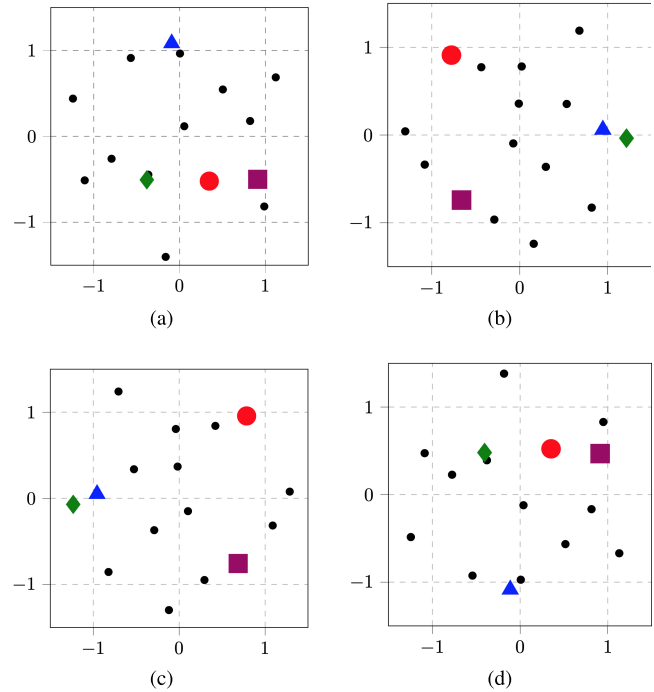

Fig. 3. Learned transmitted symbols of the open-loop MIMO AE for $M = 4$. (a) first antenna at time slot 1, (b) second antenna at time slot 1, (c) first antenna at time slot 2 and (d) second antenna at time slot 2. Constellation points for 4 out of 16 messages are highlighted with colored markers.

are trained by using the Adam optimizer [28], where the learning rate, batch size, and the number of gradient steps are summarized in Table III. In particular, i.i.d. training samples are randomly generated in each training iteration, and the total number of samples used for training each of the considered AEs is $B \times G$, where G denotes the number of gradient steps. For the performance evaluation, independent testing data are continuously generated until at least 5×10^5 errors are counted for each considered SNR.

Fig. 2 shows the achieved BLER over a range of SNRs (red triangles). As a reference, the performance of the baseline Alamouti scheme with M-QAM constellations is also shown (blue squares). For $M = 4$, the AE achieves very similar performance to the baseline scheme, indicating that the combination of a QPSK constellation and Alamouti STBC is near-optimal in this case. For $M = 16$, the AE outperforms the baseline scheme at medium-to-high SNRs by about $0.6\,\text{dB}$ when standard 16-QAM is used as the signal constellation. In order to improve the baseline for $M = 16$, we also used two geometrically-shaped (GS) signal constellations, which were obtained by training a standard AE over an AWGN channel.[5] The first constellation has 16 points in two dimensions and is shown in the inset figure in Fig. 2. Its performance sits approximately halfway between the AE and the STBC with 16-QAM. The second constellation has $M^2 = 256$ points in four dimensions. In this case, the constellation is first mapped to $\boldsymbol{s}_k = [s_{k,1}, s_{k,2}]^\mathsf{T}$, after which the standard Alamouti code can be applied. When this four-dimensional constellation is used instead, the baseline scheme has essentially the same performance as the AE-based approach.[6]

---

[5]To obtain each of the GS signal constellations, we trained several pairs of AEs over the AWGN channel at different SNRs and then chose the one with the best performance.

[6]Note that the four-dimensional format does not necessarily admit a low-complexity detection separately based on $\tilde{s}_{k,1}$ and $\tilde{s}_{k,2}$. In our implementation, the decoding is instead performed using (6), where the optimization is over all 256 constellation points.

The results presented here do not confirm the preliminary results presented in [21], where it was found that the AE outperforms the Alamouti scheme at high SNR. One potential reason for this discrepancy could be the different power normalization that is used in [21] after applying the channel matrix (cf. [21, Eq. (2)]). Instead, our results indicate that the AE learns to perform a joint optimization over the signal constellation and STBC, where the AE recovers the well-known Alamouti code for the considered scenario. To further support this observation, Fig. 3 visualizes the learned transmitted symbols for $M = 4$ after applying a 2-dimensional rotation to the symbols. Particularly, the constellation points for 4 out of $M^L = 16$ individual messages are highlighted by different markers. From these plots, one can observe that the learned constellation follows a very similar pattern as the Alamouti scheme, in the sense that the symbols in the upper left subplot (a) are symmetric with respect to the ones in the lower right (d) subplot along the x-axis, while the symbols in the upper right subplot (b) are symmetric with respect to the ones in the lower left subplot (c) along the y-axis.

IV. CLOSED-LOOP MIMO

In closed-loop MIMO systems, the CSI is available at both the transmitter and receiver. The most common approach in this case is SVD-based transmission, which we describe in the next subsection.

A. Background and Baseline Schemes

Both the transmitter and receiver compute the SVD

$$\boldsymbol{H}_k = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^\mathsf{H}, \tag{13}$$
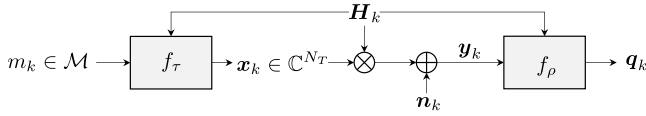
Fig. 4. Closed-loop MIMO AE, in which both the transmitter and receiver have access to CSI.

where $\mathbf{\Sigma}_k = \mathrm{diag}[\sigma_{k,1}, \ldots, \sigma_{k,R_H}]$, $\sigma_{k,1} \geq \sigma_{k,2} \geq \cdots \geq \sigma_{k,R_H} > 0$ and R_H is the rank of $\boldsymbol{H}_k$. Correspondingly, $\boldsymbol{U}_k \in \mathbb{C}^{N_R \times R_H}$ and $\boldsymbol{V}_k \in \mathbb{C}^{N_T \times R_H}$ are truncated unitary matrices. For each singular value $\sigma_{k,i}$, the transmitter chooses a constellation Ω_i from a set of available constellations, as well as a transmit power $P_{T,i} \geq 0$. This selection can be based on the total BLER according to

$$\underset{\Omega_i, P_{T,i}}{\text{minimize}} \ 1 - \prod_{i=1}^{R_H}(1 - P_e(\Omega_i, \gamma_i)) \tag{14a}$$

$$\text{s.t.} \ \prod_{i=1}^{R_H}|\Omega_{k,i}| = M, \tag{14b}$$

$$\sum_{i=1}^{R_H} P_{T,i} \leq P_T, \tag{14c}$$

$$\gamma_i = \frac{\sigma_{k,i}^2 P_{T,i}}{N_0}, \tag{14d}$$

where $P_e(\Omega, \gamma)$ is the symbol error probability of constellation Ω under the specific SNR γ. Hence, the rate is fixed to $r = \log_2(M)$. The corresponding symbol vector $\boldsymbol{s}_k = [s_{k,0}, s_{k,1}, \cdots, s_{k,R_H}]^\mathsf{T}$ is precoded by $\boldsymbol{V}_k$, so that $\boldsymbol{x}_k = \boldsymbol{V}_k \boldsymbol{s}_k$ is sent over the channel, where $\mathbb{E}\{\|\boldsymbol{x}_k\|^2|\boldsymbol{H}_k\} = P_T$. The receiver observes $\boldsymbol{y}_k = \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{n}_k$ and applies the combiner $\boldsymbol{U}_k^\mathsf{H}$, leading to the observation

$$\hat{\boldsymbol{y}}_k = \boldsymbol{U}_k^\mathsf{H}\boldsymbol{H}_k\boldsymbol{V}_k\boldsymbol{s}_k + \boldsymbol{U}_k^\mathsf{H}\boldsymbol{n}_k = \mathbf{\Sigma}_k\boldsymbol{s}_k + \boldsymbol{U}^\mathsf{H}\boldsymbol{n}_k. \tag{15}$$

Maximum-likelihood recovery of the transmitted messages is straightforward since $\mathbf{\Sigma}_k$ is a diagonal matrix.

B. Autoencoder Design and Training

The AE for a closed-loop MIMO system is implemented as shown in Fig. 4. To provide the transmitter with CSI, the corresponding NN is of the form $f_\tau : \mathcal{M} \times \mathbb{C}^{N_R \times N_T} \to \mathbb{C}^{N_T \times 1}$, yielding complex transmit vectors $\boldsymbol{x}_k = f_\tau(m_k, \boldsymbol{H}_k)$. As before, a one-hot encoding is used to map the message m_k to a vector of length M, which is then concatenated with the vectorized real and imaginary parts of the channel matrix. To enforce the power constraint $\mathbb{E}\{\|\boldsymbol{x}_k\|^2|\boldsymbol{H}_k\} = P_T$, the normalization layer is defined by

$$\boldsymbol{x}_k = \frac{\tilde{\boldsymbol{x}}_k\sqrt{P_T}}{\sqrt{\frac{1}{M}\sum_{i=1}^M \|\tilde{f}_\tau(i, \boldsymbol{H}_k)\|^2}}, \tag{16}$$

where $\tilde{\boldsymbol{x}}_k = \tilde{f}_\tau(m_k, \boldsymbol{H}_k)$ is the unnormalized NN output. Thus, even though $\boldsymbol{x}_k$ is a function of the (random) channel realization $\boldsymbol{H}_k$, the expectation $\mathbb{E}\{\|\boldsymbol{x}_k\|^2|\boldsymbol{H}_k\}$ is performed only over the messages. This ensures that the AE output is always normalized, even if the actual channel distribution deviates from the distribution used for training.

Finally, the receiver $f_\rho : \mathbb{C}^{N_R \times 1} \times \mathbb{C}^{N_R \times N_T} \to [0,1]^M$ observes $\boldsymbol{y}_k = \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{n}_k$ and, similarly to the open-loop MIMO case, the transmitted message is estimated as
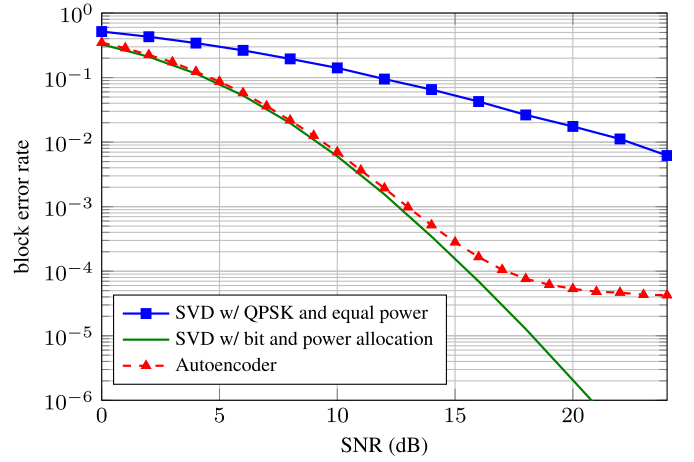

Fig. 5. BLER of the closed-loop MIMO AE for $M = 16$ and the baseline scheme consisting of a QPSK constellation, SVD-based signal processing, and a maximum-likelihood receiver. The improved baseline uses bit and power allocation.

$\hat{m}_k = \arg\max_m[\boldsymbol{q}_k]_m$, where $\boldsymbol{q}_k = f_\rho(\boldsymbol{y}_k, \boldsymbol{H}_k)$ is a probability vector obtained in the same way as in (12).

To generate a minibatch (of size B) for the Monte Carlo approximation of the cross-entropy loss (3), we first randomly generate B/M i.i.d. channel realizations. Then, for each channel realization all distinct M messages are assumed to be transmitted. Compared to the approach of generating random messages and channel realizations for each data sample, this has the advantage that the normalization factor in the denominator of (16) can be applied to M messages at once and does not need to be computed for every data sample in the batch. The same approach is used to generate the testing data.

C. Numerical Results and Discussion

We consider Rayleigh fading and use the parameters $N_T = 2, N_R = 2$, and $M = 16$, corresponding to rate $r = 4$. In this case, BLER $= \mathrm{Pr}\{\hat{m}_k \neq m_k\}$ and SNR $= P_T/(N_T N_0)$. The NN and training parameters are shown in Tabs. II and III, respectively. Compared to the open-loop case, we noticed that more data samples are required for converging to a good solution. Moreover, varying the SNR throughout the training was found to improve performance, which was not observed for the open-loop case. In particular, we train the AE consecutively at $5\,\mathrm{dB} \to 10\,\mathrm{dB} \to 15\,\mathrm{dB} \to 12\,\mathrm{dB}$, where each SNR is kept fixed for 5×10^5 iterations, giving 2×10^6 training iterations in total.

The BLER achieved by the trained AE is shown in Fig. 5 (red triangles). As a baseline, we simulate the performance of the SVD-based approach, in which the 2×2 MIMO channel is parallelized into two subchannels. We first consider the same baseline as in [21], where equal power is used at each antenna and both streams use QPSK modulation (blue squares in Fig. 5). Similarly to what was observed in [21], the AE achieves significantly better performance than the SVD-based approach with QPSK and equal power allocation. However, depending on the channel realization, the two individual subchannels will have different link quality, and bit and power
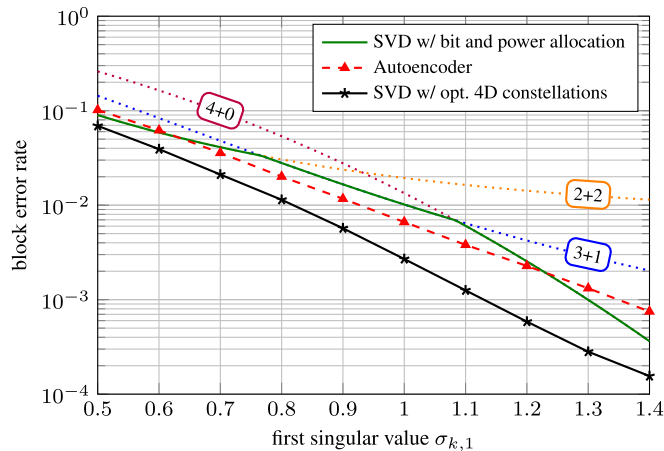
Fig. 6. BLER for the closed-loop MIMO system at SNR = 12 dB over channel matrices with fixed singular values (without retraining). The value of the second singular value is $\sigma_{k,2} = 0.5$. The dotted lines correspond to the SVD-based baseline with fixed bit allocation for the two parallel channels.

allocation are usually used to improve the overall system performance. To that end, an improved baseline scheme was simulated by solving (14) using exhaustive search assuming that the set of available signal constellations is BPSK, QPSK, c2_8 [26], and c2_16 [26], where the latter two are geometrically-shaped 2-dimensional constellations with 8 and 16 points, respectively.[7] As can be seen in Fig. 5, this improved baseline provides slightly better BLER than the AE at low SNR. At high SNR, the baseline significantly outperforms the AE, which exhibits an error floor between BLERs of 10^{-4} and 10^{-5}. This error floor is caused by the fact that the transmitter NN takes the channel as an input. Indeed, depending on the particular channel realization, we noticed that the transmitter NN sometimes produces a signal constellation that has very poor performance. While such outliers are rare, they dominate the average performance at very high SNR. We also note that the error floor can be lowered by retraining the AE at a higher SNR, but this may come at the expense of some performance loss in the low SNR regime.

The above results indicate that the closed-loop MIMO AE learns to implicitly perform a combination of GS, bit allocation, and power allocation. In fact, it is insightful to further examine the performance of the trained AE assuming that the singular values of the channel matrix remain constant. To evaluate the AE, channel matrices can be generated by using random unitary matrices for $\boldsymbol{U}_k$ and $\boldsymbol{V}_k$ in (13). Fig. 6 shows the resulting AE performance (without any retraining) as a function of the first singular value $\sigma_{k,1}$, where the second singular value is $\sigma_{k,2} = 0.5$. It can be seen that the AE actually outperforms the SVD-based baseline in the range $0.7 \leq \sigma_{k,1} \leq 1.2$, even when bit and power allocation are used.

To further improve the SVD-based baseline, we also trained a standard AE directly for the observation model (15) assuming fixed singular values, according to the methodology in

Sec. II. This essentially provides optimized 4-dimensional signal constellations over two parallel AWGN channels with different (but fixed) SNRs. The resulting performance is shown in Fig. 6 by the black markers, where the optimization is performed separately for each $(\sigma_{k,1}, \sigma_{k,2})$ with $\sigma_{k,1} \in \{0.5, 0.6, \ldots, 1.4\}$ and $\sigma_{k,2} = 0.5$. This approach provides the best performance among all considered schemes. However, it has the downside that a separate optimization is required for each pair of singular values. Nonetheless, this approach does provide additional insight into why the AE can outperform the SVD-based baseline with bit and power allocation for some channel configurations. In particular, the suboptimality of the latter scheme stems from the fact that the two parallel subchannels are treated independently, whereas the AE treats all available signal dimensions in a joint manner.

V. MIMO BROADCAST CHANNEL

In this section, we consider a downlink MIMO system where one transmitter with N_T antennas broadcasts messages to N_R receivers each with one antenna, where $N_T \geq N_R$. This scenario is sometimes also referred to as the multiple-input single-output broadcast channel [33].

A. Background and Baseline Schemes

It is assumed that local CSI $\boldsymbol{h}_{k,i}^{\mathsf{T}} \in \mathbb{C}^{1 \times N_T}$ is available at each receiver $i = 1, \ldots, N_R$, whereas the transmitter has knowledge of the full CSI $\boldsymbol{H}_k = [\boldsymbol{h}_{k,1}, \ldots, \boldsymbol{h}_{k,N_R}]^{\mathsf{T}}$. To manage the interference among different users, various algorithms have been proposed [27], [33]–[35]. In this paper, we consider both a linear precoding scheme referred to as transmitter ZF and a nonlinear vector-perturbation scheme. Both schemes are described next.

For linear precoding, the transmitter first maps N_R messages $m_{k,1}, \ldots, m_{k,N_R}$ to symbols $s_{k,1}, \ldots, s_{k,N_R}$. A precoding matrix $\boldsymbol{W}_k \in \mathbb{C}^{N_T \times N_R}$ is then used to encode $\boldsymbol{s}_k \triangleq [s_{k,1}, \cdots, s_{k,N_R}]^{\mathsf{T}}$ according to $\tilde{\boldsymbol{x}}_k = \boldsymbol{W}_k \boldsymbol{s}_k$. Afterwards, a normalized version $\boldsymbol{x}_k = \alpha \tilde{\boldsymbol{x}}_k$ is sent over the channel to ensure that $\mathbb{E}\{\|\boldsymbol{x}_k\|^2 | \boldsymbol{H}_k\} = P_T$, where $\alpha \triangleq \sqrt{P_T / \mathbb{E}\{\|\tilde{\boldsymbol{x}}_k\|^2 | \boldsymbol{H}_k\}}$ and the expectation is with respect to the messages of all users. The scaling factor α is assumed to be known to all receivers. Each of the symbols is assumed to carry $\log_2(M)$ bits and, consequently, the sum-rate of the system is $r = N_R \log_2(M)$. The precoding matrix is of the form

$$\boldsymbol{W}_k = \boldsymbol{H}_k^{\mathsf{H}} (\boldsymbol{H}_k \boldsymbol{H}_k^{\mathsf{H}} + \beta \boldsymbol{I}_{N_R})^{-1}, \qquad (17)$$

where $\beta \in \mathbb{R}$ is a regularization parameter. For ZF, we have $\beta = 0$ and $\boldsymbol{W}_k$ then corresponds to the pseudoinverse of the channel matrix. In this case, each user i observes $y_{k,i} = \boldsymbol{h}_{k,i}^{\mathsf{T}} \boldsymbol{x}_k + n_{k,i} = \alpha s_{k,i} + n_{k,i}$, from which $s_{k,i}$ can be recovered with low-complexity maximum-likelihood detection.

We also consider the nonlinear precoder proposed in [27]. Compared to ZF, the transmitter computes the unnormalized transmit vector according to $\tilde{\boldsymbol{x}}_k = \boldsymbol{W}_k(\boldsymbol{s}_k + \boldsymbol{p}_k^*)$, where $\boldsymbol{W}_k$ is again defined by (17) (potentially with $\beta > 0$),

$$\boldsymbol{p}_k^* = \underset{\boldsymbol{p}' \in A\mathbb{C}\mathbb{Z}^{N_R}}{\arg\min} \|\boldsymbol{W}_k(\boldsymbol{s}_k + \boldsymbol{p}')\|^2 \qquad (18)$$

---

[7]Rectangular 8-QAM and 16-QAM were used in [32] which give slightly worse performance.
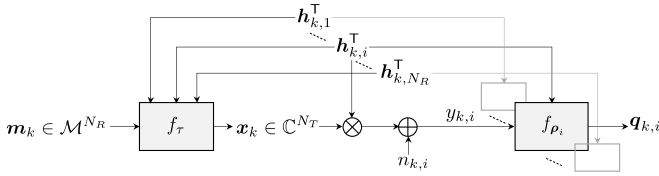
Fig. 7. MIMO broadcast AE, in which the transmitter encodes messages for the individual users, based on full CSI, while each user observes only a local measurement and local CSI.

is a perturbation vector from the scaled complex integer lattice $\mathbb{CZ}^{N_R} \triangleq \{\boldsymbol{x} + \jmath\boldsymbol{y} : \boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}^{N_R}\}$, and the scaling factor A depends on the modulation format. Each receiver first applies a modulo operation $z_{k,i} = \text{cmod}_A(y_{k,i}/\alpha) \in \mathbb{C}$, where $\text{cmod}_A(x_r + \jmath x_i) = \text{mod}_A(x_r) + \jmath\text{mod}_A(x_i)$ and

$$\text{mod}_A(x) \triangleq x - A\lfloor(x + A/2)/A\rfloor \tag{19}$$

for $x_r, x_i, x \in \mathbb{R}$. Afterwards, one can again apply low-complexity maximum-likelihood detection based on $z_{k,i}$.

B. Autoencoder Design and Training

The proposed AE implementation for the MIMO broadcast channel is visualized in Fig. 7. The transmitter $f_\tau : \mathcal{M}^{N_R} \times \mathbb{C}^{N_R \times N_T} \to \mathbb{C}^{N_T \times 1}$ maps individual messages $m_{k,i} \in \mathcal{M}$ for each user $i = 1, \cdots, N_R$ to N_T complex symbols according to $\boldsymbol{x}_k = f_\tau(\boldsymbol{m}_k, \boldsymbol{H}_k)$, where $\boldsymbol{m}_k = [m_{k,1}, \cdots, m_{k,N_R}]^\mathsf{T}$. One-hot encoding of $\boldsymbol{m}_k$ to a vector of length M^{N_R} is applied. The power constraint $\mathbb{E}\{\|\boldsymbol{x}_k\|^2|\boldsymbol{H}_k\} = P_T$ is enforced through a normalization layer similar to (16), where the sum in the denominator runs over the messages of *all* users.

The N_R receivers are implemented as N_R individual NNs of the form $f_{\rho_i} : \mathbb{C} \times \mathbb{C}^{N_T} \to [0, 1]^M$. In particular, each user i observes $y_{k,i} = \boldsymbol{h}_{k,i}^\mathsf{T}\boldsymbol{x}_k + n_{k,i}$ and generates a probability vector $\boldsymbol{q}_{k,i} \in [0, 1]^M$ according to

$$\boldsymbol{q}_{k,i} = f_{\rho_i}(y_{k,i}, \boldsymbol{h}_{k,i}), \tag{20}$$

where the receiver network is provided with its observation $y_{k,i}$ as well as the local CSI $\boldsymbol{h}_{k,i}$. Then, the transmitted message for the i-th user is estimated as $\hat{m}_{k,i} = \arg\max_m [\boldsymbol{q}_{k,i}]_m$.

In order to train the MIMO broadcast AE, the cross-entropy loss function defined in (3) cannot be used directly, as we now have several receivers that need to be optimized. Instead, we apply a joint loss function

$$\mathcal{J}_{\text{CE}}(\tau, \rho_1, \cdots, \rho_{N_R}) = -\frac{1}{N_R}\sum_{i=1}^{N_R} \mathbb{E}\left\{\log[f_{\rho_i}(\boldsymbol{y}_{k,i})]_{m_{k,i}}\right\} \tag{21}$$

which can again be optimized using the Adam optimizer.

C. Numerical Results and Discussion

As before, we consider Rayleigh fading and use the parameters $N_T = 2$, $N_R = 2$, and $M = 4$, corresponding to a sum-rate $r = 4$. Compared to the previous two cases, there are now three different NNs: one corresponding to the transmitter and two to the individual users, where the same NN architecture is used for both users, see Table II. For simplicity, it is assumed that both receiver NNs share the same
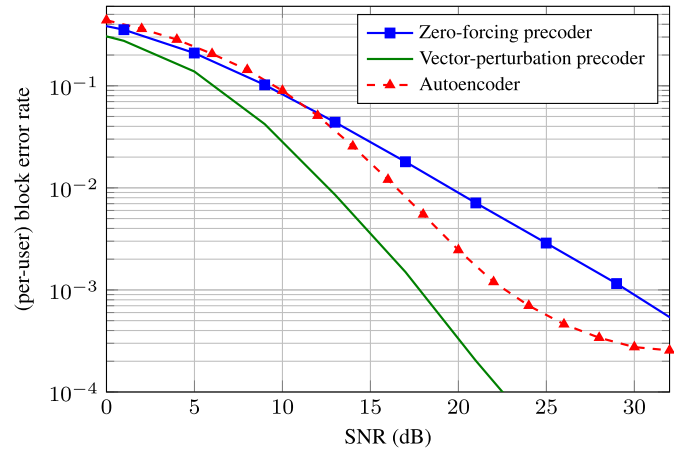

Fig. 8. BLER of the MIMO broadcast AE for $M = 4$, $N_T = 2$ transmit antennas, and $N_R = 2$ users. Both baseline schemes use QPSK modulation.

parameters, i.e., $\rho_1 = \rho_2$. Training is performed according to the parameters shown in Table III. Similarly to the closed-loop MIMO case, we found that it is beneficial to vary the SNR throughout the training.

Fig. 8 shows the achieved BLER $\Pr\{\hat{m}_{k,1} \neq m_{k,1}\}$ for the first user of the MIMO broadcast AE (red triangles) as a function of SNR $= P_T/(N_T N_0)$, where the BLER for the second user is nearly identical and omitted. The performance of the ZF baseline approach with QPSK modulation, i.e., $s_{k,i} \in \{\pm 1 \pm \jmath\}/2$, is also shown (blue squares). It can be seen that the AE-based broadcast scheme achieves significantly better performance than the ZF approach for SNRs above $11\,\text{dB}$. For example, a gain of around $6\,\text{dB}$ is achieved at a BLER of 10^{-3}. Similarly to the closed-loop MIMO case, the AE exhibits an error floor which is affected by the training SNR and stems from the fact that the channel realization is taken is an input to the transmitter NN.

As a second baseline, we simulate the performance of the nonlinear vector-perturbation precoder. For QPSK modulation $s_{k,i} \in \{\pm 1 \pm \jmath\}/2$, the scaling factor is $A = 2$ [27] and (18) is solved approximately through exhaustive search, where the search space is restricted by replacing the entire integer lattice $\mathbb{Z}^{N_R}$ with a finite set $\{-5, -4, \ldots, 4, 5\}^{N_R}$. The regularization parameter in (17) is set to $\beta = \xi/\text{SNR}$, where $\xi = 0.6$ was numerically optimized using a grid search. The resulting performance is shown by the solid green line in Fig. 8. It can be seen that this nonlinear precoder outperforms the other two approaches for all SNRs. Thus, our results show that the AE does not outperform a state-of-the-art baseline scheme for the considered MIMO broadcast scenario. However, we note that that the complexity associated with solving (18) is significant. Thus, the AE could potentially serve as a lower-complexity alternative, at the expense of some performance loss.

In parallel to our work, a related AE-based approach for the MIMO broadcast channel was recently proposed in [22]. In this work, it is shown that the considered AE achieves significant performance advantages over Tomlinson–Harashima precoding [36], [37] which is used as a benchmark. However, vector-perturbation precoding is known to outperform
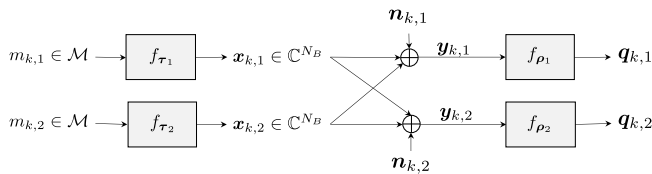
Fig. 9. Interference channel AEs, where two users communicate over the same physical channel.

Tomlinson–Harashima precoding, see, e.g., [38] for a comparison. Moreover, different block lengths are used in [22] for the AE implementation and the benchmark precoder. We also note that the AE design in [22] is different from ours in the sense that CSI is not provided as an input to the transmitter NN. Instead, the AE is trained and evaluated for the same fixed channel realization. As stated in [22], this has the downside that the AE needs to be retrained if the channel changes.

VI. INTERFERENCE CHANNEL

The last scenario we consider is the Gaussian interference channel, where N transmitter–receiver pairs, each having a single antenna, communicate over the same physical channel.

A. Background and Baseline Schemes

The interference channel is modeled by

$$\boldsymbol{Y}_k = \boldsymbol{H}_k \boldsymbol{X}_k + \boldsymbol{N}_k, \tag{22}$$

where $\boldsymbol{H}_k \in \mathbb{C}^{N \times N}$ is the channel matrix, $\boldsymbol{X}_k = [\boldsymbol{x}_{k,1}, \ldots, \boldsymbol{x}_{k,N}]^\mathsf{T}$, $\boldsymbol{Y}_k = [\boldsymbol{y}_{k,1}, \ldots, \boldsymbol{y}_{k,N}]^\mathsf{T}$, and $\boldsymbol{x}_{k,i}, \boldsymbol{y}_{k,i} \in \mathbb{C}^{N_B}$ are, respectively, the transmitted and received symbol vectors of the i-th user. As before, $\mathrm{vec}(\boldsymbol{N}_k) \sim \mathcal{CN}(\boldsymbol{0}, N_0 \boldsymbol{I}_{N N_B})$ is i.i.d. Gaussian noise.

As noted in [11], the optimal signaling scheme for the interference channel is a long-standing research problem. Existing approaches include, for example, superposition coding with private and common codebooks [39] or interference alignment [40]. In this paper, we restrict ourselves to the same scenario as considered in [11], where $[\boldsymbol{H}_k]_{i,j} = 1$ for all $i, j \in \{1, \cdots, N\}$. Moreover, it is assumed that all users have the same average power constraint $\mathbb{E}\{\|\boldsymbol{x}_{k,i}\|^2\} = N_B P_T$ for $i = 1, \ldots, N$. A simple baseline scheme in this case is to use a time-sharing approach, where the transmitters send their messages in a round-robin fashion while all other transmitters remain silent. This effectively orthogonalizes the interference channel into N parallel and independent Gaussian channels. This baseline scheme was also considered in [11] to benchmark the AE.

B. Autoencoder Design and Training

In the following, all users have the same message set $\mathcal{M}$. The generalization to different message sets for each user is straightforward. Each user maps their message $m_{k,i} \in \mathcal{M}$ to transmitted symbols via a transmitter NN $f_{\tau_i} : \mathcal{M} \to \mathbb{C}^{N_B}$ according to

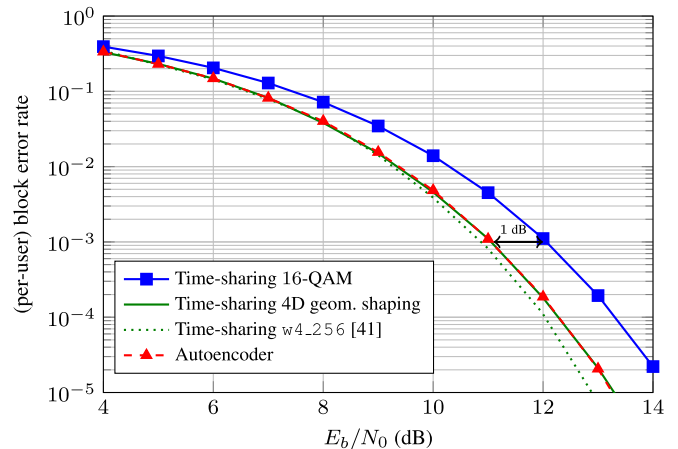$$\boldsymbol{x}_{k,i} = f_{\tau_i}(m_{k,i}), \tag{23}$$

Fig. 10. BLER for the interference-channel AEs with $N = 2$, $N_B = 4$, $M = 256$. The baseline corresponds to time sharing with three different modulation formats.

where we enforce $\mathbb{E}\{\|\boldsymbol{x}_{k,i}\|^2\} = N_B P_T$ through a standard normalization layer, similar to (2). After all users have transmitted their symbols over the channel (22), the receivers process the received symbol vectors $\boldsymbol{y}_{k,i}$ via an NN by generating M-dimensional probability vectors $\boldsymbol{q}_{k,i} = f_{\rho_i}(\boldsymbol{y}_{k,i})$ for $i = 1, \ldots, N$. The loss function for user i is the expected cross-entropy

$$\mathcal{J}_i(\boldsymbol{\theta}) = -\mathbb{E}\left\{\log[f_{\rho_i}(\boldsymbol{y}_{k,i})]_{m_{k,i}}\right\}, \tag{24}$$

where we use $\boldsymbol{\theta} = \{\tau_1, \ldots, \tau_N, \rho_1, \ldots, \rho_N\}$ to denote all transmitter and receiver NN parameters. Note that the expectation in (24) is over the channel noise and the transmitted messages of *all* users.

To optimize the parameters $\boldsymbol{\theta}$, a weighted average of the individual losses in (24) for $i = 1, \ldots, N$ can be used. To encourage equal system performance among users, the weights can further be chosen dynamically in each gradient-descent iteration, where the weight for user i is set proportionally to the corresponding per-user loss in the previous iteration [11]. For example, the common loss function for $N = 2$ users in iteration t is $\mathcal{J}_{\mathrm{CE}} = \alpha_t \mathcal{J}_1 + (1 - \alpha_t)\mathcal{J}_2$, where

$$\alpha_{t+1} = \frac{\hat{\mathcal{J}}_1(\boldsymbol{\theta}_t)}{\hat{\mathcal{J}}_1(\boldsymbol{\theta}_t) + \hat{\mathcal{J}}_2(\boldsymbol{\theta}_t)}, \quad t > 0, \tag{25}$$

with $\alpha_0 = 0.5$ and we recall that $\hat{\mathcal{J}}_i$ refers to the Monte Carlo approximation of the expectation in (24). A block diagram of the AE setup for the interference channel is shown in Fig. 9.

C. Numerical Results and Discussion

We consider the case where $N = 2$ users transmit over $N_B = 4$ complex channel uses and each user has a message set of cardinality $M = 256$. This corresponds to an uncoded transmission rate of $r = \log_2(M)/N_B = 8/4 = 2$ bits per channel use (bpcu) and user. The NN parameters are identical to the ones in [11, Table IV] and also shown in Table II. Note that for this setup, no parameters are shared between any of the four NNs. Training is performed using the Adam optimizer
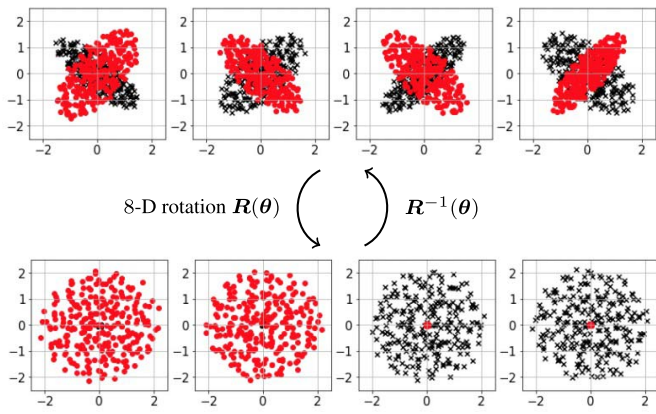
Fig. 11. Top: learned constellations for the interference channel (256 points per user), where different colors correspond to different users (cf. [11, Fig. 7 (d)]). Bottom: the same constellations after applying an optimized rotation matrix (see the appendix for details).

with learning rate 0.001 at $E_b/N_0 = P_T/(rN_0) = 11\,\mathrm{dB}$ (cf. Table III). We use the normalized SNR E_b/N_0 for this scenario to make it easier to compare to prior work in [11].

Fig. 10 shows the performance of the trained AE (red triangles) in terms of the per-user BLER $\Pr\{\hat{m}_{k,i} \neq m_{k,i}\}$ for the first user $i = 1$, where the performance of the second user is essentially the same and omitted from the plot. As a comparison, the time-sharing baseline is shown, where the two users alternate 16-QAM transmission (blue squares) which again gives a rate of $r = 2$ bpcu and user.[8] It can be seen that the AE outperforms this baseline by around $1\,\mathrm{dB}$ at a BLER of 10^{-3}.

The above results are consistent with the ones reported in [11, Fig. 6]. However, no explanation for the performance gain is provided in [11], where it is noted that the obtained results are "difficult to interpret". In the following, we aim to provide an explanation for the observed gains. First, we note that the baseline scheme can be improved by performing GS. In particular, since the two AEs jointly transmit messages over $N_B = 4$ channel uses, a time-sharing scheme with 2 users may utilize $N_B/2 = 2$ complex channel uses, i.e., 4 real dimensions. In other words, rather than time-sharing 16-QAM, a better baseline scheme is obtained by time-sharing an optimized 4-dimensional modulation format. To that end, we trained a conventional AE for a standard AWGN channel with $M = 256$ and $N_B = 2$, as explained in Sec. II. The performance when using the resulting AE in a time-sharing fashion is shown in Fig. 10 by the solid green line. Interestingly, this baseline gives the same BLER as the AE for the interference channel.

Indeed, we argue that this is not a coincidence and that the scheme learned by the two interference-channel AEs corresponds, in fact, to time sharing, albeit in a rotated reference frame. To see this, we plot the learned signal constellations for the two users in the top of Fig. 11 (which is similar to [11, Fig. 7 (d)]). As noted in [11], the learned constellation clouds resemble ellipses with orthogonal major axes

[8]These two cases are referred to as AE(4,8) and TS(4,8) in [11, Fig. 6].

and varying focal distances. We noticed that these elliptic shapes can be reproduced by applying a random 8-dimensional rotation matrix to the time-sharing AE scheme. Moreover, it is possible to find a rotation matrix that de-rotates the learned constellations in the top of Fig. 11 such that essentially all signal energy for the two users is confined to orthogonal time slots. The resulting constellations are shown in the bottom of Fig. 11. The details about how to obtain the underlying rotation matrix are given in the appendix.

Lastly, we note that optimized modulation formats in 4 dimensions have been studied before (see, e.g., [41]) and the baseline scheme can be further improved. The format with $M = 256$ points in [41] corresponds to the intersection of a 4-dimensional lattice and a spherical bounding region. This constellation is also available in [26] denoted by `w4_256`. Its performance in a time-sharing scheme is shown by the dotted green line in Fig. 10. It can be seen that the lattice-based format outperforms all other schemes discussed so far, where the gain is quite significant at high SNR. At this point, it is important to stress that the cross-entropy minimization used for training the AE does not necessarily minimize the BLER. Instead, an AE trained with cross-entropy loss maximizes a lower bound on the mutual information (MI), see, e.g., [14]. Indeed, it can be shown that the learned AE constellation for the time-sharing scheme achieves a higher MI than the lattice-based format `w4_256` over the standard AWGN channel (1) at high SNRs.

VII. CONCLUSION AND FUTURE WORK

In this work, we have evaluated several AE-based MIMO and MU communication systems in order to quantify and explain potential performance gains over fair benchmarks. The systems under consideration were open-loop MIMO, closed-loop MIMO, MIMO broadcast, and the interference channel. For all cases, the AE provides optimized mappings from messages to transmit vectors, as well as optimized detectors. For open-loop and closed-loop MIMO, we have shown that previously observed performance gains of the AE compared to the baselines can be partially attributed to geometric constellation shaping and optimized bit and power allocation. For MIMO broadcast, we have proposed a novel decentralized AE structure that performs close to nonlinear vector-perturbation precoding and significantly outperforms conventional ZF. Lastly, for the considered Gaussian interference channel, we have provided an interpretation of the learned AE-based communication scheme, thereby explaining the performance gains observed in prior work. In particular, we have shown that the AE learns a "rotated" time-sharing scheme.

In general, our work has shown that, for a wide variety of different scenarios, AE-based communication systems have the potential of learning very good solutions without a priori knowledge about complex mathematical tools or communication-theoretic principles. On the other hand, our work has also highlighted the fact that such systems do not necessarily perform better than state-of-the-art benchmarks, provided that the benchmarks are properly chosen. A particular emphasis in this work was placed on selecting benchmarks

that include known geometrically-shaped signal constellations, many of which are available in open databases such as [26]. Compared to previous work, the improved baseline schemes have allowed us to provide additional insights into AE-based systems and, in some cases, full interpretations of the learned communication schemes.

For future work, we believe that there are several important aspects concerning the use of AEs which deserve further study:

- *Channel Models:* Similar to related prior work, we have adopted memoryless channel models based on i.i.d. Rayleigh fading and AWGN. However, real wireless systems may follow a different fading model and suffer from additional impairments such as memory effects or nonlinearities caused by imperfect hardware. For such systems, existing design approaches potentially operate far from optimality and AE-based methods may provide significant performance gains. However, the AE architecture and training method would need to be appropriately modified, e.g., using orthogonal frequency-division multiplexing (OFDM) in the case of memory effects.

- *Training Complexity:* The considered AEs require a relatively large amount of training data, with large batch sizes, in order to converge to a good solution. Improving the convergence speed would allow for the exploration of a larger parameter space, for example in terms of the NN architecture, potentially leading to performance improvements.

- *Implementation Complexity:* Another important aspect is the computational complexity at runtime in practical implementations. While a thorough evaluation of the implementation complexity (including the associated performance–complexity trade-off) is beyond the scope of this paper, we note that model-compression techniques such as NN pruning can be used to significantly reduce the number of computations (often without much loss in performance).

- *Scalability:* With more transmit and receive antennas and/or more users, the complexity scaling of the corresponding NNs (e.g., in terms of layers) is currently unknown. Moreover, the employed one-hot encoding scheme causes input and output sizes to grow exponentially with the number of antennas and rate. This scalability issue may become even more severe when one considers dispersive channels in combination with OFDM, leading to hundreds or thousands of parallel channels. Alternative embeddings [42] or multi-hot sparse categorical cross entropy could help alleviate the latter issue. Both these issues affect training convergence (due to more trainable parameters) and runtime computational complexity.

- *Rate adaptation:* The considered AEs have a fixed data rate, which limits possibilities for rate adaptation. New NN architectures are needed to provide rate-adaptive transmission.

APPENDIX

To de-rotate the learned signal constellations of each user for the interference channel in Sec. VI-C, we start by constructing an overall $n \times n$ rotation matrix

$$R(\boldsymbol{\theta}) = \prod_{\substack{i,j \in [n] \\ i < j}} \boldsymbol{G}^{ij}(\theta_{ij}), \tag{26}$$

where $n = 2N_B$, $[n] \triangleq \{1, 2, \ldots, n\}$, $\boldsymbol{G}^{ij}(\theta_{ij})$ is a Givens rotation matrix, and $\boldsymbol{\theta}$ is a vector of length $n(n-1)/2$ that contains all parameters, i.e., all individual rotation angles θ_{ij}. Then, let $\boldsymbol{R}_{\mathrm{u}}(\boldsymbol{\theta}), \boldsymbol{R}_{\mathrm{l}}(\boldsymbol{\theta}) \in \mathbb{R}^{n/2 \times n}$ denote the upper and lower half of $\boldsymbol{R}(\boldsymbol{\theta})$ and define

$$\tilde{\boldsymbol{X}}_1(\boldsymbol{\theta}) = \boldsymbol{R}_{\mathrm{l}}(\boldsymbol{\theta})\boldsymbol{X}_1, \tag{27}$$
$$\tilde{\boldsymbol{X}}_2(\boldsymbol{\theta}) = \boldsymbol{R}_{\mathrm{u}}(\boldsymbol{\theta})\boldsymbol{X}_2, \tag{28}$$

where $\boldsymbol{X}_1, \boldsymbol{X}_2 \in \mathbb{R}^{n \times M}$ are the learned AE signal constellations, i.e., each column in $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ corresponds to one constellation point for the first and second user, respectively. Note that for the example in Sec. VI-C, we have $n = 8$ and $M = 256$. Finally, $\boldsymbol{\theta}$ is optimized using conventional stochastic gradient descent with loss function

$$\mathcal{J}(\boldsymbol{\theta}) = \|\tilde{\boldsymbol{X}}_1(\boldsymbol{\theta})\|^2 + \|\tilde{\boldsymbol{X}}_2(\boldsymbol{\theta})\|^2 \tag{29}$$

and learning rate 0.001. The individual angles of the Givens rotation matrices are randomly initialized assuming a uniform distribution over the interval $[0, 2\pi]$. Note that the optimization outcome and the resulting rotation matrix are not unique because the constellation of each user can be arbitrarily rotated in 4 dimensions without affecting the loss (29).

REFERENCES

[1] A. J. Paulraj, D. A. Gore, R. U. Nabar, and H. Bölcskei, "An overview of MIMO communications—A key to gigabit wireless," *Proc. IEEE*, vol. 92, no. 2, pp. 198–217, Feb. 2004.

[2] S. K. Wilson, S. Wilson, and E. Biglieri, Eds., *Transmission Techniques for Digital Communications*. Amsterdam, The Netherlands: Elsevier, 2016.

[3] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Proc. Int. Conf. Eng. Appl. Neural Netw. (EANN)*, Aberdeen, U.K., 2016, pp. 213–226.

[4] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," in *Proc. IEEE 18th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Sapporo, Japan, Jul. 2017, pp. 1–5.

[5] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2554–2564, May 2019.

[6] Y.-S. Jeon, S.-N. Hong, and N. Lee, "Blind detection for MIMO systems with low-resolution ADCs using supervised learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–6.

[7] L. V. Nguyen, D. T. Ngo, N. H. Tran, and D. H. N. Nguyen, "Learning methods for MIMO blind detection with low-resolution ADCs," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6.

[8] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based channel estimation for beamspace mmWave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 852–855, Oct. 2018.

[9] T. Gruber, S. Cammerer, J. Hoydis, and S. T. Brink, "On deep learning-based channel decoding," in *Proc. 51st Annu. Conf. Inf. Sci. Syst. (CISS)*, Baltimore, MD, USA, Mar. 2017, pp. 1–6.

[10] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be'ery, "Deep learning methods for improved decoding of linear codes," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 119–131, Feb. 2018.

[11] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[12] S. Dörner, S. Cammerer, J. Hoydis, and S. T. Brink, "Deep learning based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.

[13] H. He, S. Jin, C.-K. Wen, F. Gao, G. Y. Li, and Z. Xu, "Model-driven deep learning for physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 77–83, Oct. 2019.

[14] S. Li, C. Häger, N. Garcia, and H. Wymeersch, "Achievable information rates for nonlinear fiber communication via end-to-end autoencoder learning," in *Proc. Eur. Conf. Opt. Commun. (ECOC)*, Rome, Italy, Sep. 2018, pp. 1–3.

[15] B. Karanov *et al.*, "End-to-end deep learning of optical fiber communications," *J. Lightw. Technol.*, vol. 36, no. 20, pp. 4843–4855, Oct. 15, 2018.

[16] H. Ye, G. Y. Li, B.-H.-F. Juang, and K. Sivanesan, "Channel agnostic end-to-end learning based communication systems with conditional GAN," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, United Arab Emireates, Dec. 2018, pp. 1–5.

[17] T. J. O'Shea, T. Roy, and N. West, "Approximating the void: Learning stochastic channel models from observation with variational generative adversarial networks," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Honolulu, HI, USA, Feb. 2019, pp. 681–686.

[18] F. A. Aoudia and J. Hoydis, "Model-free training of end-to-end communication systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2503–2516, Nov. 2019.

[19] J. Song, B. Peng, C. Häger, H. Wymeersch, and A. Sahai, "Learning physical-layer communication with quantized feedback," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 645–653, Jan. 2020.

[20] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Physical layer deep learning of encodings for the MIMO fading channel," in *Proc. 55th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Monticello, IL, USA, Oct. 2017, pp. 76–80.

[21] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Deep learning based MIMO communications," 2017, *arXiv:1707.07980*.

[22] A. G. Pathapati, N. Chakradhar, P. N. V. S. S. K. Havish, S. A. Somayajula, and S. Amuru, "Supervised deep learning for MIMO precoding," in *Proc. IEEE 3rd 5G World Forum (5GWF)*, Bangalore, India, Sep. 2020, pp. 418–423.

[23] Y. Wang and T. Koike-Akino, "Learning to modulate for non-coherent MIMO," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Dublin, Ireland, Jun. 2020, pp. 1–6.

[24] M. A. ElMossallamy, Z. Han, M. Pan, R. Jantti, K. G. Seddik, and G. Y. Li, "Noncoherent MIMO codes construction using autoencoders," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.

[25] G. J. Foschini, R. D. Gitlin, and S. B. Weinstein, "On the selection of a two-dimensional signal constellation in the presence of phase jitter and Gaussian noise," *Bell Syst. Tech. J.*, vol. 52, no. 6, pp. 927–965, Jul. 1973.

[26] E. Agrell. *Database of Sphere Packings*. Accessed: Oct. 13, 2020. [Online]. Available: https://codes.se/packings/

[27] B. M. Hochwald, C. B. Peel, and A. L. Swindlehurst, "A vector-perturbation technique for near-capacity multiantenna multiuser communication—Part II: Perturbation," *IEEE Trans. Commun.*, vol. 53, no. 3, pp. 537–544, Mar. 2005.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015.

[29] S. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 8, pp. 1451–1458, Oct. 1998.

[30] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: Performance criterion and code construction," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 744–765, Mar. 1998.

[31] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block coding for wireless communications: Performance results," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 3, pp. 451–460, Mar. 1999.

[32] J. Song, C. Häger, J. Schröder, T. O'Shea, and H. Wymeersch, "Benchmarking end-to-end learning of MIMO physical-layer communication," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Taipei, Taiwan, Dec. 2020, pp. 1–6.

[33] A. Wiesel, Y. C. Eldar, and S. Shamai (Shitz), "Zero-forcing precoding and generalized inverses," *IEEE Trans. Signal Process.*, vol. 56, no. 9, pp. 4409–4418, Sep. 2008.

[34] M. Joham, W. Utschick, and J. A. Nossek, "Linear transmit processing in MIMO communications systems," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2700–2712, Aug. 2005.

[35] D. Gesbert, M. Kountouris, R. W. Heath, Jr., C.-B. Chae, and T. Sälzer, "Shifting the MIMO paradigm," *IEEE Signal Process. Mag.*, vol. 24, no. 5, pp. 36–46, Sep. 2007.

[36] M. Tomlinson, "New automatic equaliser employing modulo arithmetic," *Electron. Lett.*, vol. 7, nos. 5–6, pp. 138–139, Mar. 1971.

[37] H. Harashima and H. Miyakawa, "Matched-transmission technique for channels with intersymbol interference," *IEEE Trans. Commun.*, vol. COM-20, no. 4, pp. 774–780, Aug. 1972.

[38] C. Windpassinger, R. F. H. Fischer, and J. B. Huber, "Lattice-reduction-aided broadcast precoding," *IEEE Trans. Commun.*, vol. 52, no. 12, pp. 2057–2060, Dec. 2004.

[39] T. Han and K. Kobayashi, "A new achievable rate region for the interference channel," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 1, pp. 49–60, Jan. 1981.

[40] V. R. Cadambe and S. A. Jafar, "Interference alignment and degrees of freedom of the K-user interference channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008.

[41] G. Welti and J. Lee, "Digital transmission with coherent four-dimensional modulation," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 4, pp. 497–502, Jul. 1974.

[42] P. Rodríguez, M. A. Bautista, J. Gonzàlez, and S. Escalera, "Beyond one-hot encoding: Lower dimensional target embedding," *Image Vis. Comput.*, vol. 75, pp. 21–31, Jul. 2018.