CHALMERS
UNIVERSITY OF TECHNOLOGY

Benchmarking End-to-end Learning of MIMO Physical-Layer Communication

Downloaded from: https://research.chalmers.se, 2026-09-12 04:36 UTC

Citation for the original published paper (version of record):
Song, J., Häger, C., Schröder, J. et al (2020). Benchmarking End-to-end Learning of MIMO Physical-Layer Communication. 2020 IEEE Global Communications Conference, GLOBECOM 2020 - Proceedings. http://dx.doi.org/10.1109/GLOBECOM42002.2020.9322115

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

© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

(article starts on next page)

Benchmarking End-to-end Learning of MIMO Physical-Layer Communication

Jinxiang Song*, Christian Häger*, Jochen Schröder†, Tim O'Shea‡, Henk Wymeersch*
*Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden
†Department of Microtechnology and Nanoscience, Chalmers University of Technology, Gothenburg, Sweden
‡Virginia Tech and DeepSig, Inc., Arlington, VA, USA

GLOBECOM 2020 - 2020 IEEE Global Communications Conference | 978-1-7281-8298-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/GLOBECOM42002.2020.9322115

***Abstract*—End-to-end data-driven machine learning (ML) of multiple-input multiple-output (MIMO) systems has been shown to have the potential of exceeding the performance of engineered MIMO transceivers, without any 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. We study closed-loop MIMO, open-loop MIMO, and multi-user MIMO (MU-MIMO) and show that the gains of ML-based communication in the former two cases can be to a large extent ascribed to implicitly learned geometric shaping and bit and power allocation, not to learning new spatial encoders. For MU-MIMO, we demonstrate the feasibility of a novel method with centralized learning and decentralized executing, outperforming conventional zero-forcing. For each scenario, we provide explicit descriptions as well as open-source implementations of the selected neural-network architectures.**

I. Introduction

The ever-growing demand for higher data rates has led to the rapid development of wireless communication systems. One of the most important developments is multiple-input multiple-output (MIMO) transmission [1], where information across multiple antenna elements is encoded using spatial-multiplexing or spatial-diversity schemes such that the throughput or reliability of communication can be improved in various channel conditions. Conventional MIMO communication systems are generally divided into closed-loop and 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). Various algorithms have been proposed 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, where most works have focused on supervised ML for a *specific functional block* such as modulation recognition [3], blind MIMO detection [4], MIMO channel estimation [5], and channel decoding [6]. These ML-based methods have led to algorithms that either perform better or exhibit lower complexity than model-based algorithms. In contrast to focusing on a specific functional block, *end-to-end learning* has been proposed to optimize the transmitter and receiver jointly [7]. The workhorse of end-to-end learning is the autoencoder (AE), which employs two deep 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 [8], and nonlinear optical [9]. In cases where no channel model is available, a surrogate channel can first be learned [10] or the transmitter can be designed as a reinforcement learning (RL) agent [11], which can operate even with limited reward feedback [12].

For MIMO communication, there has been limited treatment of AEs. To the best of our knowledge, the only directly related works are [13], [14]. In [13], open-loop and closed-loop MIMO was studied, leading to better performance than the selected benchmark methods. In the extension [14], finite quantization of the CSI was considered, which was demonstrated to improve performance in some conditions. While [13], [14] have shown promising performance of ML-based MIMO communication, the proposed AEs were trained under some nonstandard assumptions, specifically in terms of CSI availability at the receiver and power normalization at the transmitter, as explained below.

In this paper, we build on the approach proposed in [13], [14], with the aim to better understand what performance can be achieved by ML-based MIMO solutions under realistic training assumptions, how and why they outperform standard benchmarks, and what limitations are imposed by NN architectures. Our main contributions in this work are as follows:

- We revisit the MIMO systems in [13], [14] and evaluate the AEs under more standard training assumptions. In particular, while CSI in [13], [14] was assumed to be estimated at the receiver, it was not actually used as a receiver input. Also, power normalization was applied after the channel-matrix multiplication (cf. [14, 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. Moreover, reproducible open-source

This work was supported by the Knut and Alice Wallenberg Foundation, grant No. 2018.0090, and the Swedish Research Council under grant No. 2018-0370. The work of C. Häger was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant No. 749798.

978-1-7281-8298-8/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on April 26,2021 at 12:31:17 UTC from IEEE Xplore. Restrictions apply.

implementations of our AEs are also provided.

- We then explain some of the performance gains obtained by the trained AEs through the selection of improved baseline schemes compared to [13], [14]. In particular, for open-loop MIMO, we show that performance gains can be partially attributed to an implicit geometric shaping of the underlying signal constellation. For closed-loop MIMO, we use an SVD-based benchmark similar to [13], [14], but augment it through additional bit and power allocation. This partially closes the performance gap to the AE, indicating that the ML-based solution learns to implement similar functionalities in a data-driven fashion.
- Lastly, we consider a multi-user MIMO (MU-MIMO) scenario, where a single multi-antenna transmitter sends information to multiple single-antenna users.[1] For such a system, we extend the training methodology in [14] to account for the joint loss function of all users and show that the resulting ML-based system achieves better performance compared to the considered baseline approach of applying transmitter ZF.

Notation

We will use the following notations: $[a,b]^M$ is the M–fold Cartesian product of the $[a,b]$–interval. $\mathcal{CN}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$ denotes the distribution of a proper complex Gaussian random vector with mean $\boldsymbol{\mu}$, covariance matrix $\boldsymbol{\Sigma}$, evaluated in $\boldsymbol{x}$ ($\boldsymbol{x}$ may be omitted to represent the entire distribution). A matrix $\boldsymbol{X}$ is converted to a vector by stacking the columns, denoted by $\mathrm{vec}(\boldsymbol{X})$.

II. Background and Baseline Schemes

In this section, we describe the open-loop MIMO, closed-loop MIMO, and MU-MIMO systems under consideration and provide the benchmark transmitter and receiver algorithms. The channel at discrete time k is denoted by $\boldsymbol{H}_k \in \mathbb{C}^{N_R \times N_T}$ for N_R receive and N_T transmit antennas. The channel is drawn from a stationary distribution $\boldsymbol{h}_k = \mathrm{vec}(\boldsymbol{H}_k) \sim p(\boldsymbol{h})$ and is assumed to be block fading with duration $N_B \geq N_T$. The transmitter can send sequences of messages belonging to a set $\mathcal{M} = \{1, 2, \ldots, M\}$. The transmission rate is assumed to be fixed and forward error correcting coding is not considered. An average transmit power of P_T is assumed.

A. Open-loop MIMO

In open-loop systems, CSI is available at the receiver but not at the transmitter. Conventional transmit approaches include space-time block codes (STBCs), which are described next.

The transmitter generates L messages, maps each to a complex data symbol $s_{k,l}$ 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 $\boldsymbol{X}_k = [\boldsymbol{x}_{k,1}, \ldots, \boldsymbol{x}_{k,N_B}]$, with the property that $\boldsymbol{X}_k^{\mathsf{H}}\boldsymbol{X}_k = P_T\boldsymbol{I}_P$. If each of the L complex data symbols corresponds to $\log_2(M)$ bits (i.e., one message), then the total bit rate is $r = L\log_2(M)/N_B$. In this paper, we restrict ourselves to the Alamouti code [15] (see also [14, Fig. 6]), where $N_B = 2$, $L = 2$, with $r = \log_2(M)$.

The receiver observes

$$\boldsymbol{Y}_k = \boldsymbol{H}_k\boldsymbol{X}_k + \boldsymbol{N}_k, \tag{1}$$

where $\mathrm{vec}(\boldsymbol{N}_k) \sim \mathcal{CN}(\boldsymbol{0}, N_0\boldsymbol{I}_{N_RN_B})$ is i.i.d. Gaussian noise. The receiver then applies maximum-likelihood detection to $\boldsymbol{Y}_k = [\boldsymbol{y}_{k,1}, \ldots, \boldsymbol{y}_{k,P}]$, which can be achieved through low-complexity linear processing [16]. Other (less complex) receiver approaches for open-loop MIMO include ZF and MMSE detection, which are not considered here.

B. Closed-loop MIMO

In closed-loop MIMO systems, the CSI is estimated at the receiver side, and then fed back to the transmitter. The most common approach is SVD-based MIMO, which we describe next.

The block fading duration is irrelevant, but should be long enough to allow feedback and use of the CSI $\boldsymbol{H}_k$. The CSI is known to both transmitter and receiver, allowing both to compute the SVD $\boldsymbol{H}_k = \boldsymbol{U}_k\boldsymbol{\Sigma}_k\boldsymbol{V}_k^{\mathsf{H}}$, where $\boldsymbol{\Sigma}_k = \mathrm{diag}[\sigma_{k,1}, \ldots, \sigma_{k,R_H}]$, $\sigma_{k,1} \geq \sigma_{k,2} \geq \cdots \geq \sigma_{k,R_H} > 0$, in which 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 message error rate according to

$$\begin{aligned}
\underset{\Omega_i, P_{T,i}}{\text{maximize}} \quad & \textstyle\prod_{i=1}^{R_H}(1 - P_e(\Omega_i, \gamma_i)) && \text{(2a)}\\
\text{s.t.} \quad & \textstyle\prod_{i=1}^{R_H}|\Omega_{k,i}| = M, && \text{(2b)}\\
& \textstyle\sum_{i=1}^{R_H} P_{T,i} \leq P_T, && \text{(2c)}\\
& \gamma_i = \frac{\sigma_{k,i}^2 P_{T,i}}{N_0}, && \text{(2d)}
\end{aligned}$$

where $P_e(\Omega, \gamma)$ is the symbol error probability of constellation Ω under the specific receive 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 $\boldsymbol{x}_k = \boldsymbol{V}_k\boldsymbol{s}_k$, where $\mathbb{E}\{\|\boldsymbol{x}_k\|^2\} \leq P_T$, is sent over the channel.

The receiver observes $\boldsymbol{y}_k = \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{n}_k$ and applies a combiner $\boldsymbol{U}_k^{\mathsf{H}}$, leading to the observation

$$\hat{\boldsymbol{y}}_k = \boldsymbol{U}_k^{\mathsf{H}}\boldsymbol{H}_k\boldsymbol{V}_k\boldsymbol{s}_k + \boldsymbol{U}_k^{\mathsf{H}}\boldsymbol{n}_k = \boldsymbol{\Sigma}_k\boldsymbol{s}_k + \boldsymbol{U}^{\mathsf{H}}\boldsymbol{n}_k \tag{3}$$

where $\boldsymbol{U}^{\mathsf{H}}\boldsymbol{n}_k$ has the same distribution as $\boldsymbol{n}_k$. Maximum likelihood recovery of the transmitted messages is straightforward, since $\boldsymbol{\Sigma}_k$ is a diagonal matrix.

C. MU-MIMO

We consider a downlink MU-MIMO system where there are a transmitter with N_T antennas and N_R receivers each with one antenna, where $N_T \geq N_R$. To eliminate the interference among different users, various algorithms including linear and non-linear precoding have been proposed. In this paper, we consider a linear precoding scheme referred to as transmitter ZF, which we describe next.

[1] This scenario was suggested as a possible extension in [14, Sec. V].

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on April 26,2021 at 12:31:17 UTC from IEEE Xplore. Restrictions apply.

Similar to the closed-loop MIMO case, the block fading duration is irrelevant, but should be long enough to allow feedback and use of the CSI. The local CSI $\boldsymbol{h}_{k,i}^{\mathsf{T}} \in \mathbb{C}^{1\times N_T}, i = 1,\dots,N_R$ is estimated at each user and fed back to the transmitter. Thus, the transmitter has knowledge of the full CSI $\boldsymbol{H}_k = [\boldsymbol{h}_{k,1},\dots,\boldsymbol{h}_{k,N_R}]^{\mathsf{T}}$, while each receiver only has access to the local CSI.

The transmitter encodes N_R messages $\boldsymbol{s}_k = [s_{k,1},\cdots,s_{k,N_R}]^{\mathsf{T}}$ with the pseudo-inverse of the channel matrix $\boldsymbol{H}_k^{\dagger}$, so that $\boldsymbol{x}_k = \alpha \boldsymbol{H}_k^{\dagger}\boldsymbol{s}_k$ is sent over the channel, where $\boldsymbol{H}_k^{\dagger} = \boldsymbol{H}_k^{\mathsf{H}}(\boldsymbol{H}_k\boldsymbol{H}_k^{\mathsf{H}})^{-1}$ and α is set to ensure that $\mathbb{E}\{\|\boldsymbol{x}_k\|^2\} \leq P_T$. If each of the messages corresponds to $\log_2(M)$ bits, the sum-rate of the system is $r = N_R\log_2(M)$.

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.

III. Autoencoders for MIMO Systems

In this section, we first describe the idea behind AE-based communication for a single-input single-output system. We then describe the AE implementation for the open-loop, closed-loop, and MU-MIMO systems. For all scenarios, the transmitters are denoted by $f_\tau(\cdot)$ and the receivers by $f_\rho(\cdot)$.

A. AE-based Communication Systems

AE-based end-to-end learning was proposed in [7]. For a single-input single-output system, the transceiver is implemented by a pair of multi-layer NNs $f_\tau : \mathcal{M} \to \mathbb{C}$ and $f_\rho : \mathbb{C} \to [0,1]^M$, where τ and ρ are the transmitter and receiver parameters.

1) Transmitter: Given a message $m_k \in \mathcal{M}$, the transmitter generates $x_k = f_\tau(m_k)$, where an average power constraint $\mathbb{E}\{|x_k|^2\} \leq P_T$ is enforced by a normalization layer. The message m_k is assumed to be encoded to an M–dimensional "one-hot" vector $\boldsymbol{l}_k \in \{0,1\}^M$, where the m–th element is 1 and all the others are 0.

2) Receiver: The complex symbol x_k is sent over the channel, and the receiver processes the received symbol y_k by first generating an M-dimensional probability vector $\boldsymbol{q}_k = f_\rho(y_k)$, 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 $\hat{m}_k = \arg\max_m [\boldsymbol{q}_k]_m$, where $[\boldsymbol{x}]_m$ returns the m-th element of $\boldsymbol{x}$.

3) End-to-end learning: 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 symbol error rate (SER) $\Pr\{\hat{m}_k \neq m_k\}$ cannot be used directly. Instead, a commonly used criterion is the categorical cross-entropy loss function ($\mathcal{J}_{\text{CE}}$) defined by

$$\mathcal{J}_{\text{CE}}(\tau,\rho) = -\mathbb{E}_{m_k,y_k}\{\log[f_\rho(y_k)]_{m_k}\}, \tag{4}$$

where the dependence of $J_{\text{CE}}(\tau,\rho)$ on τ is implicit through the distribution of the channel output y_k, which is a function of the channel input $f_\tau(m_k)$. In practice, $\mathcal{J}_{\text{CE}}$ is usually approximated via Monte-Carlo estimation according to $\hat{\mathcal{J}}_{\text{CE}} = \frac{1}{B_S}\sum_{i=1}^{B_S}\{\log[f_\rho(y_k)]_{m_k}\}$, where B_s is the batch size. And optimization of the NNs can be performed by minimizing $\hat{\mathcal{J}}_{\text{CE}}$ through the widely used Adam optimizer [17].

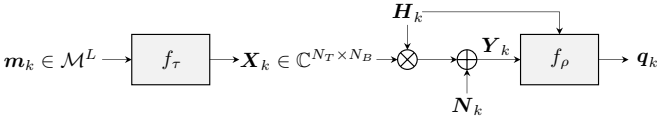

Fig. 1: Open-loop MIMO channel AE, where the transmitter learns a rate L/N_B code without CSI, while the receiver learns a decoder in the presence of CSI. The channel is drawn i.i.d. from the underlying distribution.

B. Open-loop MIMO AE

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 $\boldsymbol{m}_k = (m_1,\dots,m_L) \in \mathcal{M}^L$ into N_B coded vectors according to

$$\boldsymbol{X}_k = [\boldsymbol{x}_{k,1},\dots,\boldsymbol{x}_{k,N_B}] = f_\tau(\boldsymbol{m}_k), \tag{5}$$

where $\boldsymbol{x}_{k,p}$, $p = 1,\dots,N_B$, is a column vector of length N_T. An average power constraint according to $\sum_{p=1}^{N_B}\mathbb{E}\{\|\boldsymbol{x}_{k,p}\|^2\} \leq N_B P_T$ is enforced. 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 $\boldsymbol{Y}_k = [\boldsymbol{y}_{k,1},\cdots\boldsymbol{y}_{k,N_B}]$ as in (1) and generates a probability vector $\boldsymbol{q}_k \in [0,1]^{M^L}$ according to

$$\boldsymbol{q}_k = f_\rho(\boldsymbol{Y}_k, \boldsymbol{H}_k), \tag{6}$$

in which the CSI $\boldsymbol{H}_k$ is concatenated to the observations $\boldsymbol{Y}_k$ and then provided to the receiver.[2] Finally, the transmitted message is estimated as $\hat{m}_k = \arg\max_m [\boldsymbol{q}_k]_m$.

Note that while the transmitter does not have access to *instantaneous* CSI in the learning process, it can obtain knowledge of the CSI distribution $p(\boldsymbol{h})$, i.e., *statistical* CSI.

C. Closed-loop MIMO AE

In the closed-loop MIMO system, CSI is known to both transmitter and receiver. The AE is implemented as shown in Fig. 2. To provide the transmitter with CSI, the transmitter 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 vectors $\boldsymbol{x}_k = f_\tau(m_k, \boldsymbol{H}_k)$ in which the transmitter is provided with the message m_k as well as the CSI $\boldsymbol{H}_k$. A one-hot encoding is used with vectors of length M and a normalization layer ensures $\mathbb{E}\{\|\boldsymbol{x}_k\|^2\} \leq P_T$

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 similar to the open-loop MIMO case, the transmitted message is estimated as $\hat{m}_k = \arg\max_m [\boldsymbol{q}_k]_m$, where $\boldsymbol{q}_k = f_\rho(\boldsymbol{y}_k, \boldsymbol{H}_k)$ is a probability vector obtained in the same way as in (6).

[2]The CSI is first converted into a real-valued vector of length $2N_RN_T$, and then concatenated to the observations, which are also converted into a real-valued vector.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on April 26,2021 at 12:31:17 UTC from IEEE Xplore. Restrictions apply.

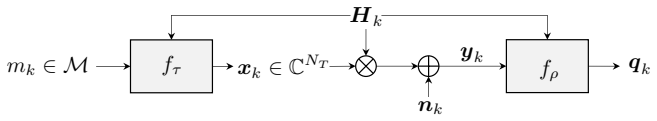

Fig. 2: Closed-loop MIMO AE, in which both transmitter and receiver have access to CSI.

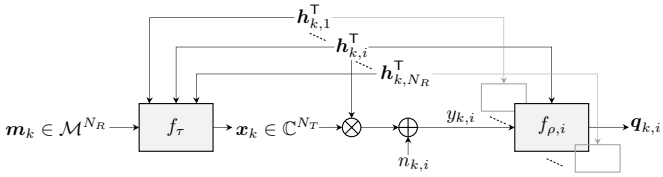

Fig. 3: MU-MIMO AE, in which the transmitter encodes messages for the individual users, based on full CSI, while each user observes only a local measurement and local CSI.

D. MU-MIMO AE

For a MU-MIMO system, each receiver only has access to the local CSI, while the transmitter has knowledge of the full CSI. The AE implementation is visualized in Fig. 3.

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. With $\boldsymbol{m}_k = [m_{k,1}, \cdots, m_{k,N_R}]^{\mathsf{T}}$,

$$\boldsymbol{x}_k = f_\tau(\boldsymbol{m}_k, \boldsymbol{H}_k) \tag{7}$$

where an average power constraint $\mathbb{E}\{\|\boldsymbol{x}_k\|^2\} \leq P_T$ is enforced. One-hot encoding of $\boldsymbol{m}_k$ to a vector of length M^{N_R} is applied.

The N_R receivers are implemented by N_R NNs, each with different parameters, of the form $f_{\tau,i} : \mathbb{C} \times \mathbb{C}^{N_T} \to [0,1]^M$. Each user i observes $y_{k,i} = \boldsymbol{h}_{k,i}^{\mathsf{T}} \boldsymbol{s}_k + n_{k,i}$ and generates a probability vector $\boldsymbol{q}_{k,i} \in [0,1]^M$ according to

$$\boldsymbol{q}_{k,i} = f_{\rho,i}(y_{k,i}, \boldsymbol{h}_{k,i}), \tag{8}$$

in which the receiver 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 $m_{k,i} = \arg\max_m [\boldsymbol{q}_{k,i}]_m$.

In order to train the MU-MIMO AE, the cross-entropy loss function defined in (4) cannot be used directly, as we now have several receivers that need to be optimized. Instead, we apply a joint loss function

$$\begin{aligned} &\mathcal{J}_{\text{CE}}(\tau, \rho_1, \cdots, \rho_{N_R}) = \\ &-\sum_{i=1}^{N_R} \mathbb{E}_{m_{k,i}, \boldsymbol{y}_{k,i}, \boldsymbol{h}_{k,i}} \left\{ \log[f_{\rho_i}(\boldsymbol{y}_{k,i})]_{m_{k,i}} \right\}, \end{aligned} \tag{9}$$

which can again be optimized using the Adam optimizer.

IV. Numerical Results

In this section, we provide simulation results for the open-loop, closed-loop, and MU-MIMO systems.[3] For all 3 cases, the channel model is assumed to be Rayleigh fading, i.e., $p(\boldsymbol{h}) = \mathcal{CN}(\boldsymbol{h}; \boldsymbol{0}, \boldsymbol{I}_{N_R N_T})$, and the system performance is measured in terms of SER as a function of the average signal-to-noise ratio (SNR), defined as $\text{SNR} = P_T/N_0$.

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

TABLE I: Neural-network parameters for (i) closed-loop, (ii) open-loop, and (iii) MU-MIMO systems

		transmitter f_τ			receiver f_ρ		
	layer	input	hidden	output	input	hidden	output
(i)	# of layers	-	3	-	-	3	-
	# of neurons	M	64	8	12	512	M
	act. function	-	ReLU	linear	-	ReLU	softmax
(ii)	# of layers	-	2	-	-	3	-
	# of neurons	$M+8$	256	8	16	2048	M
	act. function	-	ReLU	linear	-	ReLU	softmax
(iii)	# of layers	-	3	-	-	3	-
	# of neurons	$M+8$	512	8	6	256	M
	act. function	-	ReLU	linear	-	ReLU	softmax

A. NN Architectures and Training Procedure

All AEs are implemented as multi-layer fully-connected NNs, where the ReLU function is chosen as the activation function. Detailed NN parameters are summarized in Table I, where the shown parameters for the open-loop case are only used for $M = 4$. For $M = 16$, the number of hidden layers for both the transmitter and receiver NNs is increased to 5. All AEs are trained by using the Adam optimizer [17] with a learning rate 0.001 and batch size 2048.

B. Open-loop MIMO

We use the parameters $N_T = 2, N_R = 1, N_B = 2, L = 2$, and $M \in \{4, 16\}$. The training is performed at $\text{SNR} = 15$ dB for $M = 4$ and $\text{SNR} = 18$ dB for $M = 16$ over $3\,200\,000$ random channel realizations.

Fig. 4 shows the achieved SER results over a range of SNRs. As a reference, the performance of the baseline scheme described in Sec. II-A is also shown. 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 significantly at high SNR when standard 16-QAM is used as the signal constellation. In order to improve the baseline, we also used a geometrically-shaped signal constellation for $M = 16$, which was obtained by training a standard single-input single-output AE (see Sec. III-A) over an AWGN channel at $\text{SNR} = 12$ dB. When this geometrically-shaped constellation is used instead, the baseline scheme has essentially the same performance as the AE-based approach, indicating that the AE learns to perform a joint optimization over the signal constellation and STBC.

Fig. 5 visualizes the learned transmitted symbols for $M = 4$. 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 subplot 1 are symmetric with respect

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on April 26,2021 at 12:31:17 UTC from IEEE Xplore. Restrictions apply.

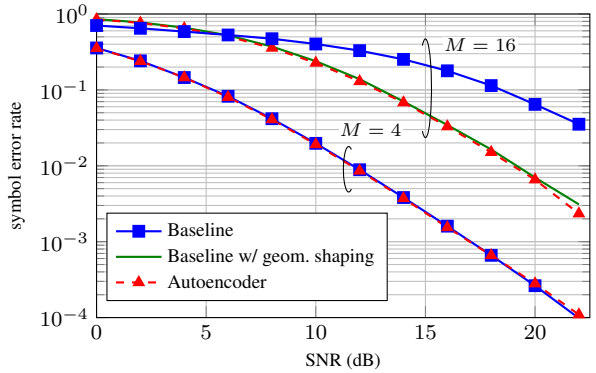

Fig. 4: SER of the open-loop MIMO AE and the baseline scheme consisting of standard M-QAM signal constellations, an Alamouti STBC, and a maximum-likelihood receiver. The improved baseline for $M = 16$ uses a geometrically-shaped signal constellation.

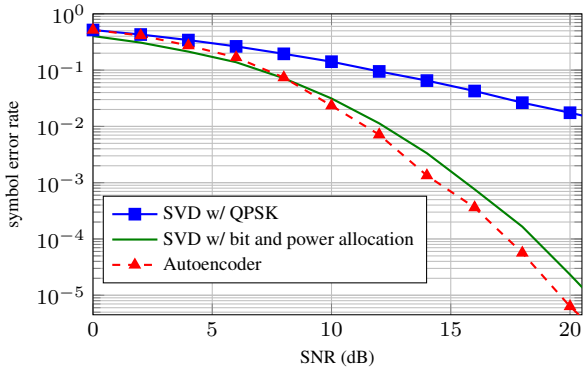

Fig. 6: SER of the closed-loop MIMO AE for $M = 16$ and the baseline scheme consisting of a QPSK constellation, SVD-based signal processing, and a maximum-likelihood receiver. The improved baseline uses bit and power allocation assuming BPSK, QPSK, 8-QAM, and 16-QAM constellations.

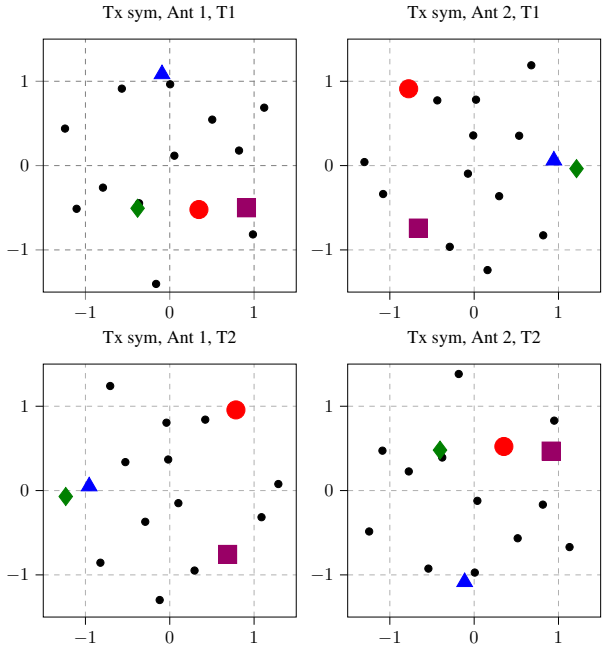

Fig. 5: Learned transmitted symbols of the open-loop MIMO AE for $M = 4$. (a) first antenna at time $p = 1$, (b) second antenna at $p = 1$, (c) first antenna at $p = 2$ and (d) second antenna at $p = 2$. Constellation points for 4 out of 16 messages are highlighted with colored markers.

to subplot 4 along the x-axis, while subplot 2 is symmetric with respect to subplot 3 along the y-axis.

C. Closed-loop MIMO

We use the parameters $N_T = 2, N_R = 2$, and $M = 16$, corresponding to rate $r = 4$. Training is performed at a fixed SNR $= 15$ dB over $40\,960\,000$ random channel realizations. Different from the open-loop case, we notice that a lot more data samples are required for converging to a good solution.

As a baseline, we simulate the performance of the SVD-based approach, in which the 2×2 MIMO channel is parallelized into two sub-channels. We first consider the same baseline as in [14], where equal power is used at each antenna, and both streams use QPSK modulation. However, depending on the channel realization, the two individual sub-channels will have different link quality, and bit and power allocation is usually used to improve the overall system performance. To that end, an improved baseline scheme was simulated by solving (2) using exhaustive search assuming that the set of available signal constellations is BPSK, QPSK, 8–QAM, and 16–QAM. Fig. 6 shows the achieved SER for both the AE implementation as well as the two SVD-based approaches. While the AE achieves better performance than the SVD-based approach including bit and power allocation, we believe that the baseline could be further improved by changing the available set of signal constellations. These results indicate that for a fixed targeted transmission rate, the closed-loop MIMO AE learns to implicitly perform a combination of constellation shaping, bit allocation, and power allocation jointly.

D. MU-MIMO

We use the parameters $N_T = 2, N_R = 2$, and $M = 16$, corresponding to a sum rate $r = 4$. Training is performed at a fixed SNR $= 15$ dB over $6\,400\,000$ channel realization. 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 network architecture is used for both users.

Fig. 7 shows the achieved SER results for the MU-MIMO AE as well the performance of baseline approach with transmitter ZF described in Sec. II-C. It can be seen that the AE-based MU-MIMO scheme achieves significantly better performance than the ZF-based approach, indicating that for independently operating receivers, the AE can learn novel encodings. Further improving the baseline scheme for the MU-MIMO case is part of ongoing research.

E. Training Complexity

Fig. 8 visualizes the evolution of training loss as a function of the number of gradient updates. Surprisingly, for the MU-MIMO case, even though the training process does not appear to have fully converged yet, the AE still outperforms the standard transmitter ZF approach quite significantly. We

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on April 26,2021 at 12:31:17 UTC from IEEE Xplore. Restrictions apply.

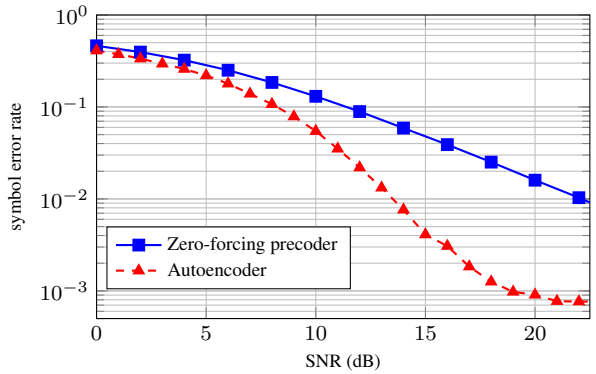

Fig. 7: SER of the MU-MIMO AE for $M = 16$, 2 transmit antennas, and 2 users and the baseline scheme consisting of a QPSK constellation, ZF transmitter, and a maximum-likelihood receiver.

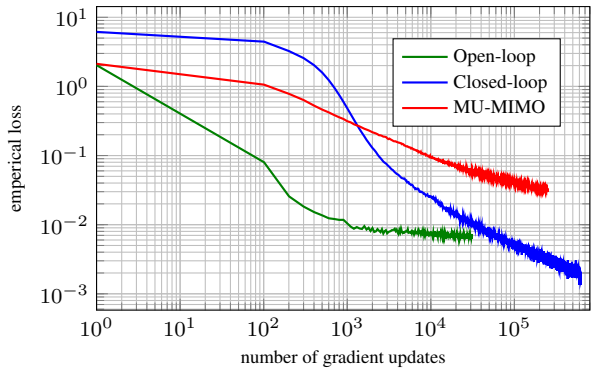

Fig. 8: Evolution of $\mathcal{J}_{\text{CE}}(\cdot)$ versus number of gradient updates.

believe that more training data or gradient updates can lead to even better performance.

V. Conclusion

In this work, we have evaluated several AE-based MIMO communication systems, in order to quantify gains with respect to fair benchmarks. The systems under consideration were open-loop MIMO, closed-loop MIMO and MU-MIMO, for which 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 the gains of the AE compared to the baselines can be partially attributed to geometric constellation shaping and optimized bit and power allocation. For MU-MIMO, we proposed a novel decentralized AE structure that is demonstrated to outperform ZF precoding. For each of these systems, we have provided open-source implementations.

Nevertheless, there are several important limitations of MIMO AEs, which deserve further study:

- *Training complexity:* The AE requires a very large amount of training data, with large batch sizes, in order to converge to a good solution. Smart selection of specific channel realizations can improve convergence speed.
- *Scalability:* With more transmit and receive antennas or more users, the complexity scaling of the NN (e.g., in terms of layers) is currently unknown and the employed one-hot encoding scheme causes input and output sizes to grow exponentially with number of antennas and rate. Alternative embeddings [18] 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.

References

[1] A. J. Paulraj, D. A. Gore, R. U. Nabar, and H. Bolcskei, "An overview of MIMO communications-a key to gigabit wireless," *Proceedings of the IEEE*, vol. 92, no. 2, pp. 198–218, 2004.

[2] S. K. Wilson, S. Wilson, and E. Biglieri, Eds., *Transmission techniques for Digital Communications*. Elsevier, 2016.

[3] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *International Conference on Engineering Applications of Neural Networks*. Springer, 2016, pp. 213–226.

[4] L. V. Nguyen, D. T. Ngo, N. H. Tran, and D. H. Nguyen, "Learning methods for MIMO blind detection with low-resolution adcs," in *IEEE International Conference on Communications (ICC)*, 2018.

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

[6] T. Gruber, S. Cammerer, J. Hoydis, and S. ten Brink, "On deep learning-based channel decoding," in *51st Annual Conference on Information Sciences and Systems (CISS)*, 2017.

[7] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[8] 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 Communications*, vol. 26, no. 5, pp. 77–83, 2019.

[9] S. Li, C. Häger, N. Garcia, and H. Wymeersch, "Achievable information rates for nonlinear fiber communication via end-to-end autoencoder learning," in *2018 European Conf. on Optical Communication (ECOC)*.

[10] 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 *IEEE Globecom Workshops*, Abu Dhabi, United Arab Emirates, 2018.

[11] F. A. Aoudia and J. Hoydis, "End-to-end learning of communications systems without a channel model," in *52nd Asilomar Conference on Signals, Systems, and Computers*, 2018, pp. 298–303.

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

[13] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Physical layer deep learning of encodings for the MIMO fading channel," in *55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2017, pp. 76–80.

[14] ——, "Deep learning based mimo communications," *arXiv preprint arXiv:1707.07980*, 2017.

[15] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, 1998.

[16] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block coding for wireless communications: performance results," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 3, pp. 451–460, 1999.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[18] P. Rodríguez, M. A. Bautista, J. Gonzalez, and S. Escalera, "Beyond one-hot encoding: lower dimensional target embedding," *Image and Vision Computing*, vol. 75, pp. 21–31, 2018.

Authorized licensed use limited to: Chalmers University of Technology Sweden. Downloaded on April 26,2021 at 12:31:17 UTC from IEEE Xplore. Restrictions apply.