# Learning to Modulate for Non-coherent MIMO

Wang, Ye; Koike-Akino, Toshiaki

TR2020-071 June 09, 2020

**Abstract**

The deep learning trend has recently impacted a variety of fields, including communication systems, where various approaches have explored the application of neural networks in place of traditional designs. Neural networks flexibly allow for data-driven optimization, but are often employed as black boxes detached from direct application of domain knowledge. Our work considers learning-based approaches to end-to-end design of modulation and signal detection for the non-coherent multi-input multi-output (MIMO) channels. We demonstrate that simulation-driven optimization can outperform traditional Grassmann designs. Additionally, we show the feasibility of noncoherent MIMO communications over extremely short channel coherence time, with as few as two time slots, which have never been explored in existing literature due to design hardness.

*IEEE International Conference on Communications (ICC)*

# Learning to Modulate for Non-coherent MIMO

Ye Wang, Toshiaki Koike-Akino
Mitsubishi Electric Research Laboratories (MERL), 201 Broadway, Cambridge, MA 02139, USA.
{yewang, koike}@merl.com

***Abstract*—The deep learning trend has recently impacted a variety of fields, including communication systems, where various approaches have explored the application of neural networks in place of traditional designs. Neural networks flexibly allow for data-driven optimization, but are often employed as black boxes detached from direct application of domain knowledge. Our work considers learning-based approaches to end-to-end design of modulation and signal detection for the non-coherent multi-input multi-output (MIMO) channels. We demonstrate that simulation-driven optimization can outperform traditional Grassmann designs. Additionally, we show the feasibility of non-coherent MIMO communications over extremely short channel coherence time, with as few as two time slots, which have never been explored in existing literature due to design hardness.**

***Index Terms*—non-coherent MIMO, deep learning, neural networks, space-time coding**

## I. Introduction

The application of machine learning techniques to communication systems has recently received increased attention [1]–[12]. Common to these approaches is the data-driven optimization of artificial neural networks (NN) to serve as various communication system components, instead of traditional approaches that are systematically driven by models and theory. The promise of such approaches is that learning could potentially overcome situations where limited models are inaccurate and complex theory is intractable. This can be viewed as part of a "deep learning" trend, where the enthusiastic application of modern deep neural networks have widely impacted a variety of fields [13].

We consider an end-to-end, learning-based approach to optimize the modulation and signal detection for non-coherent, multiple-input multiple-output (MIMO) systems, i.e., communication with multiple transmit and receive antennas, where the channel coefficients are unknown. The *end-to-end* aspect refers to the joint optimization of both the signal constellation and decoder as they interact through simulated transmission over a MIMO channel. As noted in the literature [1], [2], this general concept is analogous to training an autoencoder, but with a noisy channel inserted between the encoder and decoder, which has led several works [1]–[10] to use deep neural networks to realize both the encoder and decoder mappings. Related work [3] and [4] also consider the MIMO channel, although with channel state information (CSI) available, and the latter also examines a multi-user interference channel. We focus on the non-coherent MIMO system as a countermeasure for pilot contamination issues [19]

One aim of our paper is to reconsider the benefits of employing NNs and demonstrate an effective learning-based approach that eschews them altogether. Although most related papers used deep layers to encode data, mapping from a finite message space to channel symbols does not require any NN encoder but lookup table (or single linear layer with one-hot encoder) since any arbitrary constellation can be represented with a single layer in principle. Non-coherent MIMO decoding theory [14] guides us to a simplified decoder architecture that avoids employing NNs, while still retaining the ability to perform simulation-driven optimization. We evaluate and compare this network-less approach versus employing an NN decoder, and find that they perform comparably.

With our learning-based approach, we also demonstrate that non-coherent MIMO communication is feasible even at extremely short coherence time, i.e., with the channel coefficients stable for as few as two time slots. Unlike various conventional approaches [14]–[19] to MIMO modulation design such as Grassmann space-time codes, which have limitations on time slots versus antennas, the learning-based approach is not limited by analytical design constraints. Relaxing these constraints is also supported by the recent extension by [20] of MIMO capacity theory [21], [22], which shows that the conventional unitary, isotropically distributed inputs are no longer capacity achieving when antennas exceed time slots.

The key contributions of the paper are summarized below:

- We apply machine learning to optimize space-time constellations in non-coherent MIMO systems.
- We optimize encoder lookup tables without relying on deep network architectures.
- We compare NNs and model-based detectors to demodulate the space-time constellations.
- Our learned modulation and detection schemes outperform traditional designs in some SNR regimes.
- We demonstrate that non-coherent MIMO is feasible even for extremely short coherence time.

*Notations:* We use uppercase/lowercase bold letters, e.g., $\mathbf{X}$ and $\mathbf{m}$, to denote matrices/vectors. A circularly-symmetric Gaussian distribution with zero mean and $\sigma^2$ variance is denoted by $\mathcal{CN}(0, \sigma^2)$. We write $\mathbf{X}^\dagger$ to denote the conjugate transpose of $\mathbf{X}$, and $\mathbf{I}_m$ to denote the $m \times m$ identity matrix. We use $\mathbb{E}[\cdot]$, $\|\cdot\|$, $\mathbb{R}$ and $\mathbb{C}$ to denote expectation, Frobenius norm, set of real numbers, and set of complex numbers, respectively.

## II. Modulation Optimization for MIMO Systems

### A. Non-Coherent MIMO Channel

We consider transmission over MIMO channels with $m$ transmitter antennas and $n$ receiver antennas. When transmit-

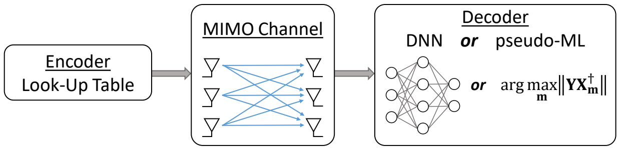
Encoder
Look-Up Table
MIMO Channel
Decoder
DNN
or
pseudo-ML
or
$\arg\max_{\mathbf{m}} \|\mathbf{Y}\mathbf{X}_{\mathbf{m}}^{\dagger}\|$

Fig. 1: End-to-end learning for modulation and detection, with encoder signal constellation specified by a lookup table and decoder realized as neural network or pseudo-ML decoding.

ting a message using $L$ channel symbols, the received signal $\mathbf{Y}$ is an $n \times L$ complex-valued matrix given by $\mathbf{Y} := \mathbf{H}\mathbf{X}+\mathbf{Z}$, where $\mathbf{X}$ is an $m \times L$ complex-valued matrix representing the transmitted signal, $\mathbf{H}$ is the $n \times m$ complex, random channel matrix, and $\mathbf{Z}$ is an $n \times L$ complex-valued matrix representing additive white Gaussian noise. The elements of the channel matrix $\mathbf{H}$ are i.i.d. $\mathcal{CN}(0, 1/m)$ and are independent of the noise $\mathbf{Z}$, which is i.i.d. $\mathcal{CN}(0, \sigma^2)$. We constrain the transmission to have average power $\mathbb{E}[\|\mathbf{X}\|^2/(mL)] = 1$, such that the average signal-to-noise ratio (SNR) is given by $1/\sigma^2$.

We focus on the *non-coherent* scenario where the random channel matrix $\mathbf{H}$ is unknown (i.e., no CSI), but fixed over the $L$ channel uses. Non-coherent MIMO systems are particularly advantageous over coherent counterparts, which must rely on pilots, for the case when the fading channel rapidly changes, e.g., in bullet train and vehicular communications. More recently, non-coherent techniques received much attention as a viable counter solution to prevent pilot contamination issues [23]–[25] in massive user communications.

### B. Encoder Parameterization

The encoder maps a $k$-bit message to an $L$ symbol transmission across $m$ antennas. Any encoder mapping, $f : \{0, 1\}^k \rightarrow \mathbb{C}^{m \times L}$, can be parameterized by a simple lookup table (LUT) specified by a codebook matrix $\mathbf{C} \in \mathbb{C}^{2^k \times mL}$. For power efficiency, the mean row of $\mathbf{C}$ is subtracted from each row of $\mathbf{C}$ to produce the centered codebook matrix $\overline{\mathbf{C}}$. Then, the average power constraint is enforced by scaling $\overline{\mathbf{C}}$ to produce centered and normalized code matrix $\widetilde{\mathbf{C}} := \overline{\mathbf{C}}\sqrt{2^k mL}/\|\overline{\mathbf{C}}\|$. To encode a message $\mathbf{m} \in \{0, 1\}^k$, the encoder mapping selects the row in $\widetilde{\mathbf{C}}$ indexed by the integer value of $\mathbf{m}$, and reshapes it to an $m \times L$ matrix to form the transmitted signal $\mathbf{X}_{\mathbf{m}} := f_{\mathbf{C}}(\mathbf{m}) \in \mathbb{C}^{m \times L}$. Note that this encoder procedure is equivalent to a single linear layer with batch normalization without relying on deep layers.

For non-coherent MIMO systems, space-time constellations based on Grassmannian manifold [14]–[19] have been widely investigated due to capability of simplified maximum-likelihood (ML) decoding. However, it was proven in [20] that Grassmann constellation is not optimal to achieve capacity and beta-variate modulation was proposed instead. Nevertheless, it is still an open problem to design such space-time constellations which are efficiently decodable without CSI. In this paper, we apply the end-to-end machine learning technique to optimize constellations and blind decoders. Note that most end-to-end learning approaches [1]–[10] use deep neural networks to realize encoder mapping functions, while the simple LUT approach described above is sufficient to represent arbitrary mapping functions. This is particularly suited to when the cardinality of $2^k$ is moderately small.

### C. Decoder Realizations

As the receiver has no CSI, we need to employ blind detection methods for non-coherent MIMO systems. The optimal ML detection [26] for non-coherent channels is often cumbersome to implement unless the space-time constellation is in Grassmannian manifold. We consider two parametric, soft-output decoders that approximate the unnormalized, log-likelihoods for each possible message, and thus output a real-valued vector of length $2^k$. For both decoders, the softmax operation is applied to the output vector (by exponentiating each element and then scaling to normalize the sum to one) to produce a stochastic vector, denoted by $P^{\theta}_{\mathbf{m}|\mathbf{Y}}$, that approximates the posterior distribution $P_{\mathbf{m}|\mathbf{Y}}$. Note that applying the softmax operation to the vector of unnormalized, log-likelihoods $\{\log \alpha P_{\mathbf{Y}|\mathbf{m}}(\mathbf{Y}|\mathbf{m})\}_{\mathbf{m} \in \{0,1\}^k}$, for some constant $\alpha > 0$, would yield the corresponding posterior distribution $\{P_{\mathbf{m}|\mathbf{Y}}(\mathbf{m}|\mathbf{Y})\}_{\mathbf{m} \in \{0,1\}^k}$.

*1) Pseudo-ML (pML) Decoder:* If the codewords are orthonormal, that is, $\mathbf{X}_{\mathbf{m}}\mathbf{X}_{\mathbf{m}}^{\dagger} = L \cdot \mathbf{I}_m$ for all $\mathbf{m} \in \{0, 1\}^k$, then the ML decoding rule is simplified in [14] to be

$$\arg\max_{\mathbf{m} \in \{0,1\}^k} \left\|\mathbf{Y}\mathbf{X}_{\mathbf{m}}^{\dagger}\right\|^2, \tag{1}$$

since the terms $\|\mathbf{Y}\mathbf{X}_{\mathbf{m}}^{\dagger}\|^2$ are proportional to $\log \alpha P(\mathbf{Y}|\mathbf{m})$, for some $\alpha > 0$ that is constant with respect to $\mathbf{m}$. This decoder immediately inspires a soft-output decoder that simply scales the objective in (1) with a parameter $\theta \geq 0$ to output

$$\left\{\theta \|\mathbf{Y}\mathbf{X}_{\mathbf{m}}^{\dagger}\|^2\right\}_{\mathbf{m} \in \{0,1\}^k}. \tag{2}$$

The parameter $\theta$ both accounts for the fact that $\|\mathbf{Y}\mathbf{X}_{\mathbf{m}}^{\dagger}\|^2$ is only proportional to $\log \alpha P(\mathbf{Y}|\mathbf{m})$, and allows the confidence of the decoder to be tuned, which is particularly important since it will be employed while enforcing the orthonormal constraint (i.e., $\mathbf{X}_{\mathbf{m}}\mathbf{X}_{\mathbf{m}}^{\dagger} = L\mathbf{I}_m$) in only a soft manner. Hence, we call this the *pseudo*-ML (pML) decoder. Smaller/larger $\theta$ indicates lower/higher confidence, as the corresponding posterior estimate $P^{\theta}_{\mathbf{m}|\mathbf{Y}}$ (produced by applying the softmax operation) approaches uniform as $\theta \rightarrow 0$ and certainty as $\theta \rightarrow \infty$. This parameter $\theta$ will be optimized by machine learning techniques.

*2) Neural Network (NN) Decoder:* Alternatively, a soft-output decoder can be realized with an NN, which serves as a parametric approximation for the mapping

$$g_{\theta} : \mathbb{C}^{n \times L} \rightarrow \mathbb{R}^{2^k}, \tag{3}$$

where $\theta$ denotes the parameters specifying the weights of the NN layers. The network is applied to the received signal to yield an approximation of the log-likelihoods, to which the softmax operation is applied to produce the corresponding posterior estimate $P^{\theta}_{\mathbf{m}|\mathbf{Y}} := \mathrm{SoftMax}(g_{\theta}(\mathbf{Y}))$. Note that the above nonbinary output mapper can be readily modified to produce bit-wise soft outputs in $\mathbb{R}^k$ dimension, each of real values represents log-likelihood ratios for the case of bit-interleaved coded-modulation (BICM) systems.

The specific network architectures used in this paper are detailed alongside discussion of the results in Section III-A. In order to handle a complex-valued matrix as input, $\mathbf{Y}$ is simply decomposed into its real and imaginary components and vectorized, i.e., $\mathbf{Y}$ is represented as a real-valued vector of length $2nL$. Fig. 1 summarizes our approach.

### D. Optimization Objective

The main optimization objective is to minimize the cross-entropy loss, which is given below, with respect to the encoder parameter $\mathbf{C}$ and decoder parameter $\theta$,

$$\mathbb{E}\big[-\log P^{\theta}_{\mathbf{m}|\mathbf{Y}}(\mathbf{m}|\mathbf{Y})\big] = \mathcal{H}(\mathbf{m}|\mathbf{Y}) + \mathrm{KL}(P_{\mathbf{m}|\mathbf{Y}} \| P^{\theta}_{\mathbf{m}|\mathbf{Y}}), \tag{4}$$

where $P^{\theta}_{\mathbf{m}|\mathbf{Y}}$ is produced by applying the softmax operation to the log-likelihoods produced by either decoder given by (2) or (3), as described in Section II-C. Here, $\mathcal{H}(\cdot)$ and $\mathrm{KL}(\cdot\|\cdot)$ denote entropy and Kullback–Leibler divergence, respectively. From the above equation, the ideal optimization of the decoder should cause the estimated posterior $P^{\theta}_{\mathbf{m}|\mathbf{Y}}$ to converge toward the true posterior $P_{\mathbf{m}|\mathbf{Y}}$, and the overall optimization is equivalent to maximizing the mutual information $\mathcal{I}(\mathbf{m};\mathbf{Y}) = \mathcal{H}(\mathbf{m}) - \mathcal{H}(\mathbf{m}|\mathbf{Y})$, with respect to the signal constellation, since $\mathcal{H}(\mathbf{m}) = k$ is constant.

As mentioned earlier, the pML decoder given by (2) is formulated assuming orthonormal codewords that satisfy $\mathbf{X}_{\mathbf{m}}\mathbf{X}_{\mathbf{m}}^{\dagger} = L\mathbf{I}_m$ for all $\mathbf{m} \in \{0,1\}^k$. We enforce orthonormality as a soft constraint by introducing an additional *orthonormal-loss* term given by

$$\ell(\mathbf{C}) := \frac{1}{2^k m^2} \sum_{\mathbf{m}\in\{0,1\}^k} \left\|\mathbf{X}_{\mathbf{m}}\mathbf{X}_{\mathbf{m}}^{\dagger}/L - \mathbf{I}_m\right\|^2.$$

The optimization objective that we use for the pML decoder is formed by combining this orthonormal loss with the primary cross-entropy loss as follows

$$\min_{\mathbf{C},\theta} \mathbb{E}\big[-\log P^{\theta}_{\mathbf{m}|\mathbf{Y}}(\mathbf{m}|\mathbf{Y})\big]\big(1 + \lambda\ell(\mathbf{C})\big), \tag{5}$$

where $\lambda > 0$ is a weighting parameter to control the impact of the orthonormal loss term. Note that rather simply adding on the orthonormal loss term, i.e., using an objective of the form $\mathbb{E}[-\log P^{\theta}_{\mathbf{m}|\mathbf{Y}}(\mathbf{m}|\mathbf{Y})] + \lambda\ell(\mathbf{C})$, the loss terms have been multiplicatively combined in (5). We found from analyses that this improved the reliability of convergence, possibly since these loss terms might decay at very different rates making it difficult to tune $\lambda$ in an additive combination.

## III. Performance Analysis

We evaluate communicating $k \in \{2,4,6,8\}$ bits over $L \in \{2,4\}$ channel uses. For $L = 2$ time slots, we vary the number of receiver antennas $n \in \{2,3,4\}$, while keeping the number of transmit antennas fixed at $m = 2$, since theory [21], [22] teaches that unilaterally increasing transmit antennas $m > L$ does not increase capacity. We also tested increasing $m > L$ and found that it resulted in performance nearly identical to $m = L$. For $L = 4$ time slots, we vary both the number of transmit and receive antennas $(m,n) \in \{(2,2),(3,3),(4,4)\}$. For each operating point (combination of parameters $k, L, m, n$), we evaluated both the pML and NN decoders, by optimizing each across a variety of hyperparameters, and selecting the best performing codes. Further details about the network architectures and training procedures are given in Sections III-A and III-B.

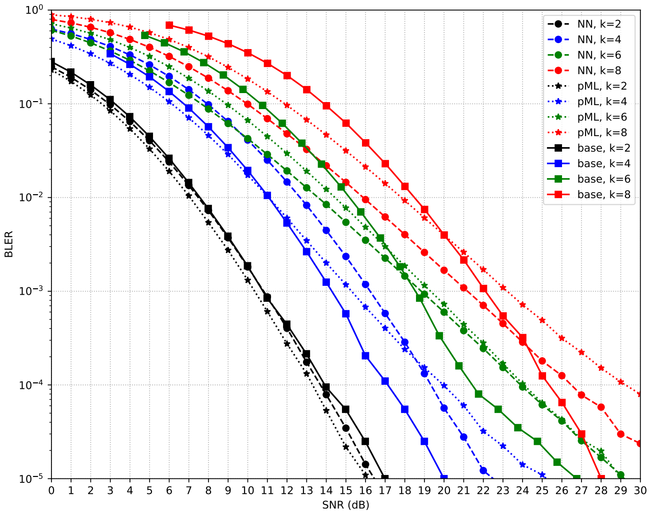

Fig. 2: BLER performance comparison for $L = 4$ and $(m,n) = (2,2)$, between our learned codes (with NN and pML decoders) and the baseline scheme of [16].

In Fig. 2, we compare the block-error rate (BLER) performance of our learned schemes against the analytical code constructions of [16], which are limited to $L \geq 4$ and $m = 2$. Note that our learned schemes can outperform (by several dB) the baseline, particularly at lower SNR regimes, at which the encoder was optimized. Our BLER results across more parameters are shown in Figs. 3 and 4 for $L = 2$ and Figs. 5 and 6 for $L = 4$. Since existing Grassmann code designs require $L > m$, our demonstration of feasibility for learning-based code design at $L = m = 2$ is novel to the best of author's knowledge. Note that for several operating points (six for $L = 2$ and two for $L = 4$), the pML results exhibit large error floors, while the NN results generally do not. At other operating points, the results between NN and pML are similar (although sometimes slightly better or worse). Figs. 7 through 10 depict the achievable throughput performance estimated from the cross-entropy loss given by (4), via

$$\frac{k - \mathbb{E}\big[-\log P^{\theta}_{\mathbf{m}|\mathbf{Y}}(\mathbf{m}|\mathbf{Y})\big]}{L} \lessapprox \frac{I(\mathbf{m};\mathbf{Y})}{L}.$$

The theoretical capacity lower-bounds derived in [20], which are tight only in the high SNR regimes, are also shown in Figs. 7 through 10 for comparison.

We searched over fewer hyperparameters (optimization instances) for the pML decoder cases, which may have played

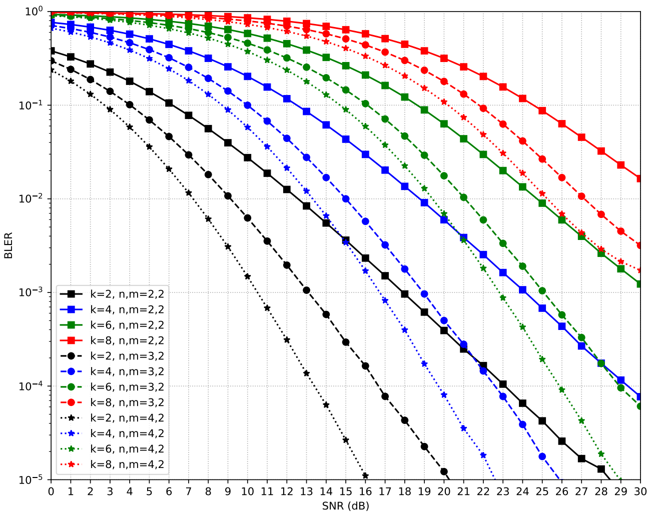

Fig. 3: BLER performance for NN decoder at $L = 2$.

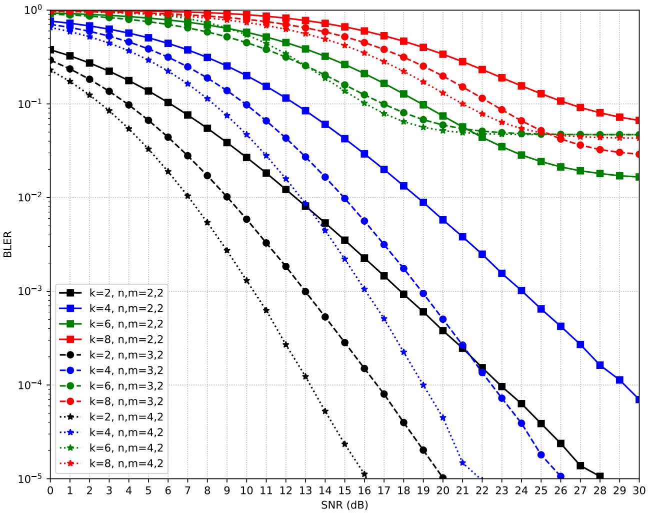

Fig. 4: BLER performance for pML decoder at $L = 2$.

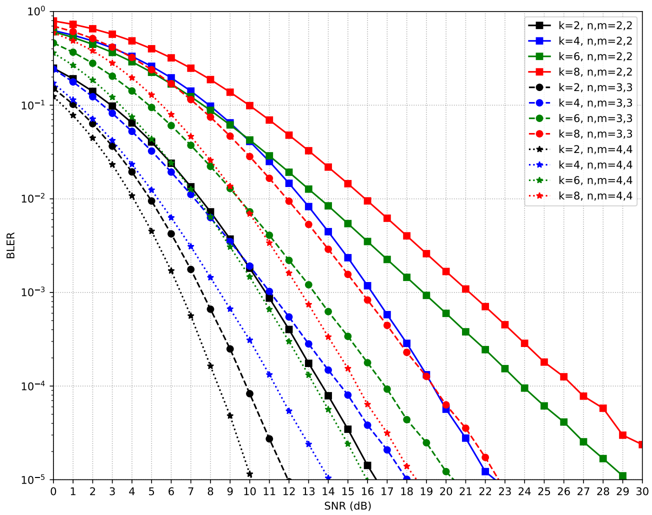

Fig. 5: BLER performance for NN decoder at $L = 4$.

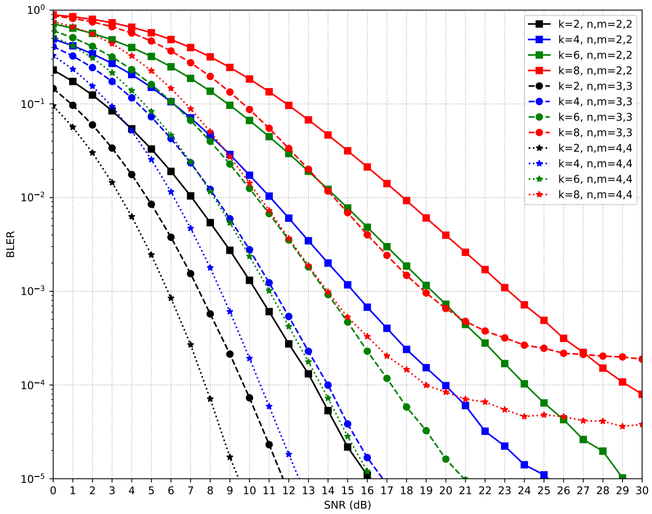

Fig. 6: BLER performance for pML decoder at $L = 4$.

a role in the optimization failing in some cases. Interestingly, despite the orthonormal loss-term, only one operating point ($k = 2$, $L = 4$, $m = n = 2$) resulted in the codebook for the pML decoder converging to orthonormal codewords. However, we did find that the presence of the orthonormal loss-term improved the optimization success rate. From throughput performance, we can confirm that NN decoder outperforms pML decoder, approaching close to the theoretical bounds. Two examples of learned signal constellations are shown in Fig. 11.

### A. *Neural Network (NN) Architectures*

We use two well-known NN architectures, the multilayer perceptron (MLP) and the Residual MLP (ResMLP) [27], [28], to realize the NN-based decoders discussed in Section II-C.

In the MLP architecture, the input vector $\mathbf{x}_0$ is mapped to the output vector $\mathbf{x}_{l+1}$ by applying a series of affine transformations and element-wise, nonlinear operations. The $l$ hidden (intermediate) layers and output layer (vector) of the network are given by $\mathbf{x}_{i+1} := \phi_i\big(\mathbf{W}_i\mathbf{x}_i + \mathbf{b}_i\big)$, for $i \in \{0, \ldots, l\}$, where $\{\mathbf{W}_i, \mathbf{b}_i\}_{i=0}^{l}$ are the affine transformation parameters that define the network, and $\phi_i(\cdot)$ denotes the element-wise application of the activation function $\phi_i$. For all of our MLP networks, we used the rectified linear unit (ReLU) for the hidden layers (i.e., $\phi_i(x) := \max(x, 0)$, for $i \in \{0, \ldots, l-1\}$) and the identity function for the output layer (i.e., $\phi_l(x) = x$). Note that the dimensions of the weight matrices $\mathbf{W}_i$ and bias vectors $\mathbf{b}_i$ are constrained by the desired input, output, and hidden layer dimensions.

In the ResMLP architecture, the input vector $\mathbf{x}$ is first mapped to an initial hidden vector $\mathbf{h}_0$ via an affine transformation, i.e., $\mathbf{h}_0 := \mathbf{W}_0\mathbf{x} + \mathbf{b}_0$. Then, over $l$ blocks, the hidden vector is updated according to $\mathbf{h}_i := F_{2i}\big(F_{2i-1}(\mathbf{h}_{i-1})\big) + \mathbf{h}_{i-1}$, for $i \in \{1, \ldots, l\}$, where $F_i(\cdot)$ denotes the sequential

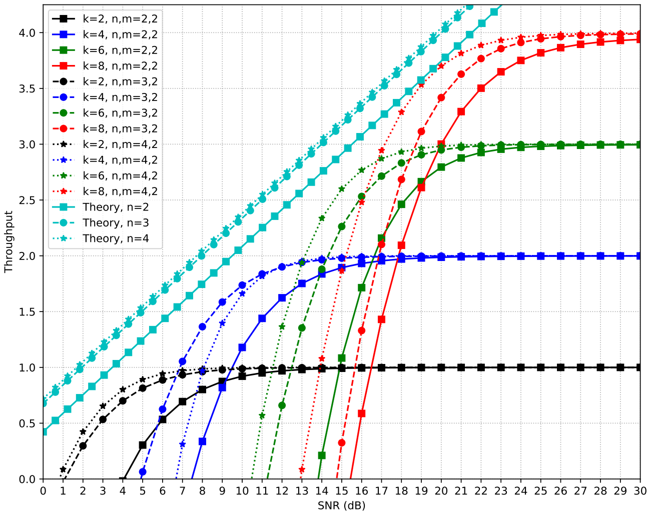

Fig. 7: Throughput comparison for NN decoder at $L = 2$.

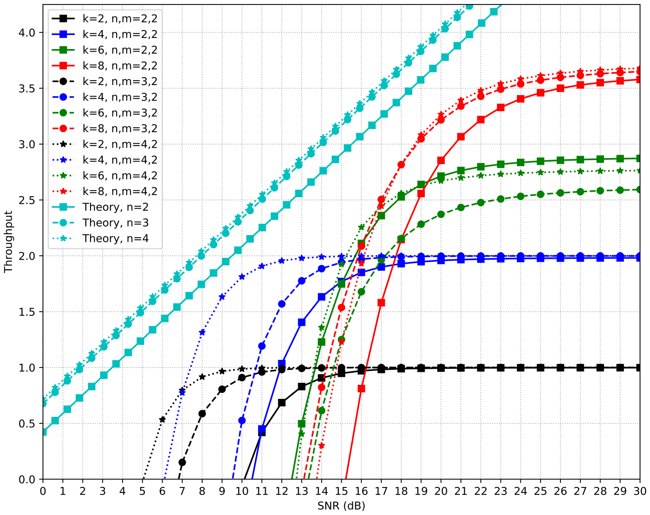

Fig. 8: Throughput comparison for pML decoder at $L = 2$.

application of batch-normalization [29], an activation function, and affine transform, i.e., $F_i(\mathbf{h}) := \mathbf{W}_i\phi_i(\text{BatchNorm}(\mathbf{h})) + \mathbf{b}_i$. The output is computed as $\mathbf{y} := \mathbf{W}_{2i+1}\phi_{2i+1}(\mathbf{h}_l) + \mathbf{b}_{2i+1}$.

### *B. Training Procedures*

We perform the optimization of the objectives given in Section II-D with stochastic gradient descent (SGD), specifically the popular Adam [30] variant, which adaptively adjusts learning rates based on moment estimates. For each iteration, the expectations are approximated by the empirical mean over a batch of 10,000 uniformly sampled messages, randomly drawn along with random channel matrices and noise for the transmission of each message. Training was performed for up to 50,000 iterations, with early stopping applied to halt training when the objective fails to improve, while saving the best snapshot in terms of BLER. We implemented these simulations using the Chainer deep learning framework [31].

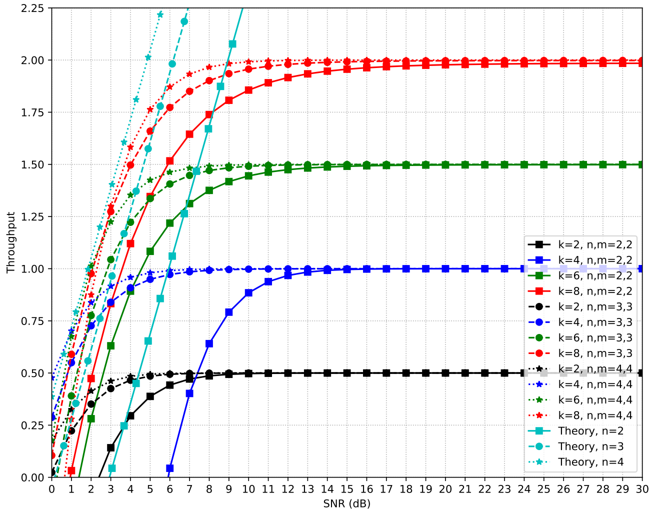

Fig. 9: Throughput comparison for NN decoder at $L = 4$.

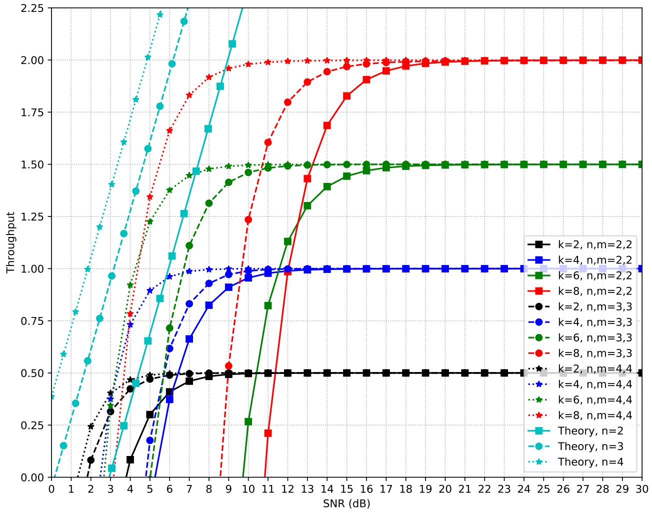

Fig. 10: Throughput comparison for pML decoder at $L = 4$.

For the NN decoder, we tried both the MLP and ResMLP architectures across the combination of $l \in \{1, 2, 3\}$ layers/blocks and $\{256, 500, 1000\}$ hidden layer dimensions. For the pML decoder, the main hyperparameter is just the weight $\lambda$ in the objective function given by (5), which we varied across $\lambda \in \{0.1, 0.3, 1.0, 3.0, 10.0\}$. For both decoders, an additional hyperparameter is the SNR used during training simulations, which we non-exhaustively varied from 10 dB to 30 dB in 5 dB increments, by trying a few for each operating point.

## IV. Discussion and Ongoing Work

We reevaluated the role of NNs in learning-based approaches to communications. We demonstrated that NNs can be avoided altogether while still employing the fundamentals of simulation-driven design optimization. Our learned modulation and detection schemes outperformed traditional designs at some SNR regimes. We also used this approach

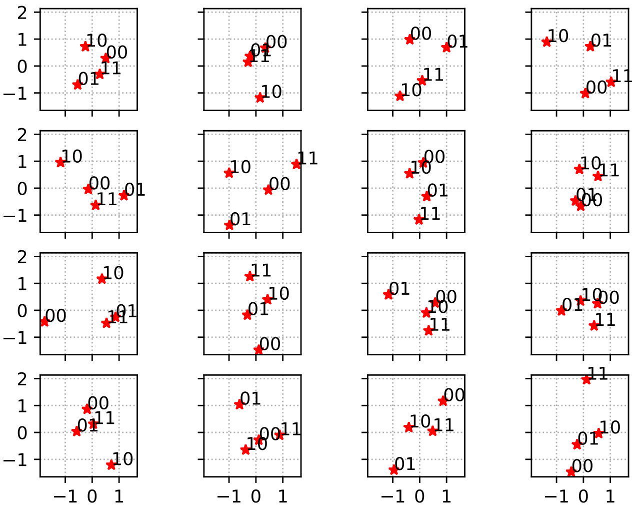

Fig. 11: Example signal constellation learned with pML decoder for $k = 2$, $L = 4$, $(m, n) = (4, 4)$.

to show the feasibility of non-coherent MIMO for coherence windows as short as two time slots. Our ongoing work includes further investigation into improving optimization stability and performance. The generalized log-likelihood ratio test (GLRT) decoder given by [18] does not require the codewords to be orthonormal, which would obviate the need for an orthonormal loss term. Due to the increased implementation and computational complexity, investigating this GLRT decoder remains ongoing work.

## References

[1] T. J. O'Shea, K. Karra, and T. C. Clancy, "Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention," in *Signal Processing and Information Technology (ISSPIT), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 223–228.

[2] 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.

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

[4] T. Erpek, T. J. O'Shea, and T. C. Clancy, "Learning a physical layer scheme for the MIMO interference channel," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–5.

[5] H. Kim, Y. Jiang, S. Kannan, S. Oh, and P. Viswanath, "Deepcode: Feedback codes via deep learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 9458–9468.

[6] B. Karanov, M. Chagnon, F. Thouin, T. A. Eriksson, H. Bülow, D. Lavery, P. Bayvel, and L. Schmalen, "End-to-end deep learning of optical fiber communications," *arXiv preprint arXiv:1804.04097*, 2018.

[7] S. Dörner, S. Cammerer, J. Hoydis, and S. ten Brink, "Deep learning based communication over the air," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, 2018.

[8] F. A. Aoudia and J. Hoydis, "End-to-end learning of communications systems without a channel model," *arXiv preprint arXiv:1804.02276*, 2018.

[9] V. Raj and S. Kalyani, "Backpropagating through the air: Deep learning at physical layer without channel models," *IEEE Communications Letters*, vol. 22, no. 11, pp. 2278–2281, 2018.

[10] T. J. O'Shea, T. Roy, N. West, and B. C. Hilburn, "Physical layer communications system design over-the-air using adversarial networks," *arXiv preprint arXiv:1803.03145*, 2018.

[11] H. Kim, Y. Jiang, R. Rana, S. Kannan, S. Oh, and P. Viswanath, "Communication algorithms via deep learning," *arXiv preprint arXiv:1805.09317*, 2018.

[12] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2018.

[13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[14] B. M. Hochwald and T. L. Marzetta, "Unitary space-time modulation for multiple-antenna communications in Rayleigh flat fading," *IEEE transactions on Information Theory*, vol. 46, no. 2, pp. 543–564, 2000.

[15] B. M. Hochwald, T. L. Marzetta, T. J. Richardson, W. Sweldens, and R. Urbanke, "Systematic design of unitary space-time constellations," *IEEE transactions on Information Theory*, vol. 46, no. 6, pp. 1962–1973, 2000.

[16] X.-B. Liang and X.-G. Xia, "Unitary signal constellations for differential space-time modulation with two transmit antennas: parametric codes, optimal designs, and bounds," *IEEE Transactions on Information Theory*, vol. 48, no. 8, pp. 2291–2322, 2002.

[17] B. Hassibi and B. M. Hochwald, "Cayley differential unitary space-time codes," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1485–1503, 2002.

[18] T. Koike-Akino and P. Orlik, "High-order super-block GLRT for non-coherent Grassmann codes in MIMO-OFDM systems," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*. IEEE, 2010, pp. 1–6.

[19] T. Koike-Akino, P. V. Orlik, and K. J. Kim, "Pilot-less high-rate block transmission with two-dimensional basis expansion model for doubly-selective fading mimo systems," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–7.

[20] W. Yang, G. Durisi, and E. Riegler, "On the capacity of large-MIMO block-fading channels," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 2, pp. 117–132, 2013.

[21] E. Telatar, "Capacity of multi-antenna Gaussian channels," *European transactions on telecommunications*, vol. 10, no. 6, pp. 585–595, 1999.

[22] T. L. Marzetta and B. M. Hochwald, "Capacity of a mobile multiple-antenna communication link in Rayleigh flat fading," *IEEE transactions on Information Theory*, vol. 45, no. 1, pp. 139–157, 1999.

[23] T. L. Marzetta *et al.*, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, p. 3590, 2010.

[24] J. Hoydis, S. Ten Brink, and M. Debbah, "Massive mimo in the ul/dl of cellular networks: How many antennas do we need?" *IEEE Journal on selected Areas in Communications*, vol. 31, no. 2, pp. 160–171, 2013.

[25] R. R. Müller, L. Cottatellucci, and M. Vehkaperä, "Blind pilot decontamination," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 773–786, 2014.

[26] J.-C. Belfiore and A. M. Cipriano, "Space-time coding for non-coherent channels," in *Space-Time Wireless Systems: From Array Processing to MIMO Communications*. Cambridge University Press, 2006, ch. 10, p. 198.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[28] F. Huang, J. Ash, J. Langford, and R. Schapire, "Learning deep ResNet blocks sequentially using boosting theory," *arXiv preprint arXiv:1706.04964*, 2017.

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[30] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. [Online]. Available: https://arxiv.org/abs/1412.6980

[31] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: a next-generation open source framework for deep learning," in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.