# CMDNet: Learning a Probabilistic Relaxation of Discrete Variables for Soft Detection With Low Complexity

Edgar Beck, *Graduate Student Member, IEEE*, Carsten Bockelmann, *Member, IEEE*, and Armin Dekorsy, *Senior Member, IEEE*

***Abstract*— Following the great success of Machine Learning (ML), especially Deep Neural Networks (DNNs), in many research domains in 2010s, several ML-based approaches were proposed for detection in large inverse linear problems, e.g., massive MIMO systems. The main motivation behind is that the complexity of Maximum A-Posteriori (MAP) detection grows exponentially with system dimensions. Instead of using DNNs, essentially being a black-box, we take a slightly different approach and introduce a probabilistic Continuous relaxation of disCrete variables to MAP detection. Enabling close approximation and continuous optimization, we derive an iterative detection algorithm: Concrete MAP Detection (CMD). Furthermore, extending CMD by the idea of deep unfolding into CMDNet, we allow for (online) optimization of a small number of parameters to different working points while limiting complexity. In contrast to recent DNN-based approaches, we select the optimization criterion and output of CMDNet based on information theory and are thus able to learn approximate probabilities of the individual optimal detector. This is crucial for soft decoding in today's communication systems. Numerical simulation results in MIMO systems reveal CMDNet to feature a promising accuracy complexity trade-off compared to State of the Art. Notably, we demonstrate CMDNet's soft outputs to be reliable for decoders.**

***Index Terms*— Maximum a-posteriori (MAP), individual optimal, massive MIMO, concrete distribution, Gumbel-softmax, machine learning, neural networks.**

## I. Introduction

COMMUNICATIONS is a long standing engineering discipline whose theoretical foundation was laid by Claude Shannon with his landmark paper "A Mathematical Theory of Communication" in 1948 [1]. Since then, the theory has evolved into an own field known as information theory today and found its way into many other research areas where data or information is processed including artificial intelligence and especially its subdomain Machine Learning (ML). Information theory relies heavily on description with probabilistic models playing a significant role for design of new generations of cellular communication systems from 2-6G with respective increases in data rate. Probabilistic models have shown to be advantageous also in the ML research domain. Accordingly, both fields, communications and ML, have touched repeatedly in the past (see, e.g., [2]–[4]).

In the early 2010s, a special class of these models gave rise to several breakthroughs in data-driven ML research: Deep Neural Networks (DNNs). Inspired by the brain, several layers of artificial neurons are stacked on top of each other to create an expressive feed forward DNN able to approximate arbitrarily well [5] and thus to learn higher levels of abstraction, i.e., features, present in data [6]. This is of crucial importance for tasks where there are no well-established models but data to be collected. Previously considered intractable to optimize, dedicated hardware and software, i.e., Graphics Processing Units (GPU) and automatic differentiation frameworks [7], innovation to DNN models [8], [9] and advancements in training [8] have made it possible to build algorithms that equal or even surpass human performance in specific tasks such as pattern recognition [10] and playing games [11]. The impact included all ML subdomains, e.g., classification [9], [10] in supervised learning, generative modeling in unsupervised learning [12] and Q-learning in reinforcement learning [11].

### A. ML in Communications

The great success of DNNs in many domains has stimulated large amount of work in communications just in recent years [6]. Especially in problems with a model deficit, e.g., detection in molecular and fiber-optical channels [13], [14], or without any known analytical solution, e.g., finding codes for AWGN-channels with feedback [15], DNNs have already proven to allow for promising application. Notably, the authors of the early work [16] demonstrate a complete communication system design by interpreting transmitter, channel and receiver as an autoencoder which is trained end-to-end similar to one DNN. The resulting encodings are shown to reach the Bit Error Rate (BER) performance of handcrafted systems in a

Manuscript received February 23, 2021; revised June 16, 2021 and August 13, 2021; accepted September 11, 2021. Date of publication September 22, 2021; date of current version December 16, 2021. This work was partly funded by the German Ministry of Education and Research (BMBF) under grant 16KIS1028 (MOMENTUM). The associate editor coordinating the review of this article and approving it for publication was F. Gao. *(Corresponding author: Edgar Beck.)*

The authors are with the Department of Communications Engineering, University of Bremen, 28359 Bremen, Germany (e-mail: beck@ant.uni-bremen.de; bockelmann@ant.uni-bremen.de; dekorsy@ant.uni-bremen.de).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCOMM.2021.3114682.

Digital Object Identifier 10.1109/TCOMM.2021.3114682

simple AWGN scenario. A model-free approach based on reinforcement learning is proposed in [17]. Using advances in unsupervised learning, also blind channel equalization can be improved [18].

In contrast to typical ML research areas, a model deficit does not apply to wireless communications. The models, e.g., AWGN, describe reality well and enable development of optimized algorithms. However, those algorithms may be too complex to be implemented. This algorithm deficit applies to the core problem typical for communications: classification in large inverse problems. Therefore, it is crucial to find an approximate solution with an excellent trade-off between detection accuracy and complexity.

### *B. Related Work*

A prominent example for large inverse problems under current deep investigation and a key enabler for better spectral efficiency in 5G/6G are massive Multiple Input Multiple Output (MIMO) systems [19]. In an uplink scenario, a Base Station (BS) is equipped with a very large number of antennas (around 64-256) and simultaneously serves multiple single-antenna User Equipments (UEs) on the same time-frequency resource. As a first step in receiver design, different tasks such as channel equalization/estimation and decoding are typically split to lower complexity. But still, an algorithm deficit applies to both MIMO detection and decoding of large block-length codes, e.g., LDPC and Polar codes, since Maximum A-Posteriori (MAP) detection has high computational complexity growing exponentially with system or code dimensions. Even its efficient implementation, the Sphere Decoder (SD), remains too complex in such a scenario [20].

Hence, in communications history, many suboptimal solutions have been proposed to overcome the complexity bottleneck of the optimal detectors. One key approach is to relax the discrete Random Variables (RV) to be continuous: Remarkable examples include Matched Filter (MF), Zero Forcing and MMSE equalization. But linear equalization with subsequent detection leads to a strong performance degradation compared to SD in symmetric systems.

A heuristic based on the latter is the V-Blast algorithm which first equalizes and then detects one layer with largest Signal-to-Noise Ratio (SNR) successively to reduce interference iteratively. A more efficient and sophisticated implementation, MMSE Ordered Successive Interference Cancellation (MOSIC), is based on a sorted QR decomposition of a MMSE extended system matrix with post sorting and offers a good trade-off between complexity and accuracy [21].

Pursuing another philosophy of mathematical optimization, the SemiDefinite Relaxation (SDR) technique [22] treats MIMO detection as a non-convex homogeneous quadratically constrained quadratic problem and relaxes it to be convex by dropping the only non-convex requirement. Proving to be a close approximation, SDR is more complex than MOSIC and solved by interior point methods from convex optimization.

Furthermore, also probabilistic model-based ML techniques were introduced to improve the trade-off and to integrate detection seamlessly with decoding: Mean Field Variational Inference (MFVI) provides a theoretical derivation of soft Successive Interference Cancellation (SIC) and the Bethe approach lays the foundation for loopy belief propagation [23]. Simplifying the latter, Approximate Message Passing (AMP) is derived known to be optimal for large system dimensions in i.i.d. Gaussian channels and computational cheap [24]. As a further benefit, soft outputs are computed, today a strict requirement to account for subsequent soft decoding. But in practice, the performance of probabilistic approximations like MFVI and AMP suffers if the approximating conditions are not met, i.e., from the full-connected graph structure and finite dimensions in MIMO systems, respectively.

More recent work considers DNNs for application in MIMO systems and focus on the idea of deep unfolding [25], [26]. In deep unfolding, the number of iterations of a model-based iterative algorithm is fixed and its parameters untied. Further, it is enriched with additional weights and non-linearities to create a computational efficient DNN being optimized for performance improvements in MIMO detection [27], [28], belief propagation decoding [29]–[31] and MMSE channel estimation [32]. The former approach DetNet, a generic DNN model with a large number of trainable parameters based on an unfolded projected gradient descent, proves DNNs to allow for a promising trade-off between accuracy and complexity. In [33], unfolding of an extension of AMP to unitarily-invariant channels, the Orthogonal AMP (OAMP), into OAMPNet is proposed adding only 2 trainable parameters per layer. Offering promising performance, the complexity bottleneck of one matrix inversion per iteration makes this model-driven approach rather unattractive compared to DetNet. Another DNN-like network MMNet is inspired by iterative soft thresholding algorithms and AMP [34]: Striking the balance between expressiveness and complexity, and exploiting spectral and temporal locality, MMNet can be trained online for any realistic channel realization if coherence time is large enough. Since online training is in general wasteful, an efficient implementation non-trivial and requires particularly deep analysis, we focus in this work on offline learning. One major drawback of the latter approaches is that they focus on MIMO detection and do not provide soft outputs.

### *C. Main Contributions*

The main contributions of this article are manifold: Inspired by recent ML research, we first introduce a CONtinuous relaxation of the probability mass function (pmf) of the disCRETE RVs by a probability density function (pdf) from [35], [36] to the MAP detection problem. The proposed CONCRETE relaxation offers many favorable properties: On the one hand, the pdf of continuous RVs converges to the exact pmf in the parameter limit. On the other hand, we notice good algorithmic properties like avoiding marginalization and allowing for differentiation instead. By this means, we replace exhaustive search by computationally cheaper continuous optimization to approximately solve the MAP problem in any probabilistic non-linear model. We name our approach Concrete MAP Detection (CMD).

Second, following the idea of Deep Unfolding, we unfold the gradient descent algorithm into a DNN-like model CMDNet with a fixed number of iterations to allow for parameter optimization and to further improve detection accuracy while limiting complexity. By this means, we are able to combine the advantages of DNNs and model-based approaches. As the number of parameters is small, we are able to dynamically adapt them to easily adjust CMDNet to different working points. Further, the resulting structure potentially allows for fast online training of CMDNet.

Thirdly, we derive the optimization criterion from an information theoretic perspective and are hence able to provide probabilities of detection, i.e., reliable soft outputs. We show that optimization is then equivalent to learning an approximation of the Individual Optimal (IO) detector. This allows us to account for subsequent decoding, e.g., in MIMO systems, in contrast to literature [28], [34].

Finally, we provide numerical simulation results for use of CMD and CMDNet in MIMO systems including a variety of simulation setups, e.g., correlated channels, revealing CMDNet to be a generic and promising approach competitive to State of the Art (SotA). Notably, we show superiority to other recently proposed ML-based approaches and demonstrate with simulations in coded systems CMDNet's soft outputs to be reliable for decoders as opposed to [28]. Furthermore, by estimating the computational complexity, we prove CMD to feature a promising trade-off between detection accuracy and complexity. Notably, only the Matched Filter has lower complexity.

In the following, we first introduce the concrete relaxation to MAP detection in Section II using the example of an inverse linear problem. In Section III, we follow a different route and explain how to learn the posterior, i.e., replacing it by some tractable approximation. To yield a suitable model for this approximation, we propose to unfold CMD into CMDNet which we are then able to train by variants of Stochastic Gradient Descent (SGD). Finally in Section IV and V, we provide numerical results of the bit error performance in comparison to other SotA approaches using the example of uncoded and coded MIMO systems and summarize the main results, respectively.

## II. Concrete Relaxation of MAP Problem

### A. System Model and Problem Statement

To motivate the concrete relaxation, we consider a probabilistic and (possibly) non-linear observation model described by a continuous and differentiable pdf $p(\mathbf{y}|\mathbf{x})$. Based on this model, the task is to classify/detect the discrete multivariate RV $\mathbf{x}$, i.e., $\mathbf{x} = \{x_n\}_{n=1}^{N_\mathrm{T}}$ whose i.i.d. elements are from a set $\mathcal{M}$, given the observation $\mathbf{y} \in \mathbb{C}^{N_\mathrm{R}\times 1}$.

To illustrate our findings with an example typically encountered in communications, we focus on a linear complex-valued observation model, e.g., MIMO system, although the following derivations hold without loss of generality for general $p(\mathbf{y}|\mathbf{x})$. We first exclude coding from our model:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{1a}$$

$$\text{with} \quad p(\mathbf{y}|\mathbf{x},\mathbf{H},\sigma_\mathrm{n}^2) = \frac{1}{\pi^{N_\mathrm{R}}\sigma_\mathrm{n}^{2N_\mathrm{R}}} e^{-\frac{1}{\sigma_\mathrm{n}^2}(\mathbf{y}-\mathbf{H}\mathbf{x})^H(\mathbf{y}-\mathbf{H}\mathbf{x})}. \tag{1b}$$

There, a linear channel $\mathbf{H} \in \mathbb{C}^{N_\mathrm{R}\times N_\mathrm{T}}$ with statistic $p(\mathbf{H})$, e.g., such that taps $h_{mn} \sim \mathcal{CN}(0, 1/N_\mathrm{R})$ are i.i.d. Gaussian distributed, introduces correlation between the elements $x_n$ with $\mathrm{E}[|x_n|^2] = 1$ from typical modulation sets $\mathcal{M}$, e.g., BPSK, 8-PSK or 16-QAM. Then, Gaussian noise $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma_\mathrm{n}^2\mathbf{I}_{N_\mathrm{R}})$ with variance $\sigma_\mathrm{n}^2$ distributed according to $p(\sigma_\mathrm{n}^2)$ interferes. The matrix $\mathbf{I}_{N_\mathrm{R}}$ denotes the identity matrix of dimension $N_\mathrm{R} \times N_\mathrm{R}$. For the following derivations, note that we are able to replace $\mathbf{y}$ by one total observation $\tilde{\mathbf{y}}$ including RVs $\mathbf{H}$ and $\sigma_\mathrm{n}^2$ without loss of generality since $\mathbf{x}$, $\mathbf{H}$ and $\sigma_\mathrm{n}^2$ are statistically independent:

$$p(\tilde{\mathbf{y}}|\mathbf{x}) = p(\mathbf{y},\mathbf{H},\sigma_\mathrm{n}^2|\mathbf{x}) = p(\mathbf{y}|\mathbf{x},\mathbf{H},\sigma_\mathrm{n}^2)\cdot p(\mathbf{H})\cdot p(\sigma_\mathrm{n}^2). \tag{2}$$

In this detection problem, there exist two optimal detectors from a probabilistic Bayesian viewpoint: First, we have the likelihood function $p(\mathbf{y}|\mathbf{x})$ but would like to infer the most likely transmit signal $\mathbf{x}$ based on an a-posteriori pdf $p(\mathbf{x}|\mathbf{y})$. Using Bayes rule, we are able to reform the MAP problem w.r.t. the known likelihood into

$$\hat{\mathbf{x}} = \underset{\mathbf{x}\in\mathcal{M}^{N_\mathrm{T}\times 1}}{\arg\max}\ p(\mathbf{x}|\mathbf{y}) \tag{3a}$$

$$= \underset{\mathbf{x}\in\mathcal{M}^{N_\mathrm{T}\times 1}}{\arg\max}\ p(\mathbf{y}|\mathbf{x})\cdot p(\mathbf{x}) \tag{3b}$$

$$= \underset{\mathbf{x}\in\mathcal{M}^{N_\mathrm{T}\times 1}}{\arg\min}\ -\ln p(\mathbf{y}|\mathbf{x}) - \ln p(\mathbf{x}) \tag{3c}$$

where $p(\mathbf{x})$ is the known a-priori pdf. Since the RV is discrete, i.e., $x_n \in \mathcal{M}$, an exhaustive search over all element combinations is required to solve the MAP problem becoming computational intractable for large system dimensions. Note that the Sphere Detector (SD) provides an efficient implementation [20]. Second, we notice that the MAP detector only delivers the most likely received vector $\mathbf{x}$. Hence, it minimizes frame error rate and provides hard decisions.

In coded systems with soft decoders usually employed today, delivering soft information is a strict requirement. The Individual Optimal (IO) detector delivers such soft output as probabilities and is optimal in terms of minimizing the Symbol Error Rate (SER) per individual symbol without coding. It is obtained by evaluating the marginal posterior distribution w.r.t. every single $x_n$:

$$\hat{x}_n = \underset{x_n\in\mathcal{M}}{\arg\max}\ p(x_n|\mathbf{y}) = \underset{x_n\in\mathcal{M}}{\arg\max}\ \frac{\sum\limits_{\mathbf{x}\setminus x_n} p(\mathbf{y}|\mathbf{x})\cdot p(\mathbf{x})}{\sum\limits_{x_n}\sum\limits_{\mathbf{x}\setminus x_n} p(\mathbf{y}|\mathbf{x})\cdot p(\mathbf{x})}. \tag{4}$$

However, it has higher complexity due to required marginalization w.r.t. $\mathbf{x}$. Since the MAP detector performance coincides with the IO detector in the high SNR regime and is of lower complexity, we restrict to the MAP detector as a benchmark in simulations without coding.

### B. Concrete Distribution

We now focus on the following question to improve the performance complexity trade-off: How to model the prior information $p(\mathbf{x})$ accurately by some approximation $p(\tilde{\mathbf{x}})$? In [37], we proposed to use ML tricks from [35], [36] to achieve this and to make inference computationally tractable. The idea was recently discovered in the ML community in the context of unsupervised learning of generative models [35], [36]. There, marginalization to compute the objective function, the evidence, becomes intractable. Therefore, the Evidence is replaced by its Lower BOund (ELBO) by means of an auxiliary posterior function. But optimizing w.r.t. the ELBO results in high variance of the gradient estimators. For variance reduction, the so called reparametrization trick is used and leads to an optimization structure similar to an autoencoder known as the variational autoencoder [23]. There, the stochastic node is reparametrized by a continuous RV, e.g., a Gaussian, and its parameters, e.g., mean and variance. In contrast to continuous RVs, reparametrization of discrete RVs is not possible. Hence, a CONtinuous relaxation of disCRETE RVs, the CONCRETE distribution, was proposed in [35], [36] independently.

To explain the introduction of this relaxation to MAP detection, let us assume that we have the discrete binary RV $x \in \mathcal{M}$ with $\mathcal{M} = \{-1, +1\}$. Further, we define the discrete RV $\mathbf{z}$ as a one-hot vector where all elements are zero except for one element, i.e., $\mathbf{z} \in \{0,1\}^{2\times 1}$ with two possible realizations $\mathbf{z}_1 = [1,0]^T$, $\mathbf{z}_2 = [0,1]^T$. In addition, we describe the values of $\mathcal{M}$ by the representer vector $\mathbf{m} = [-1,1]^T$. That way, we can write $x = \mathbf{z}^T\mathbf{m}$, e.g., $x = [1,0]\cdot[-1,1]^T = -1$. Now, the one-hot vector $\mathbf{z} \in \{0,1\}^{M\times 1}$ represents a categorical RV with $M = |\mathcal{M}|$ classes. Connecting Monte Carlo methods to optimization [35], the Gumbel-Max trick states that we are able to generate samples, i.e., classes, of such a categorical RV or pmf $p(x)$ by sampling an index $i^*$ from $M$ continuous i.i.d. Gumbel RVs $g_i$ known from extreme value theory:

$$i^* = \underset{i=1,\dots,M}{\arg\max}\ \ln p(x = m_i) + g_i. \tag{5}$$

Defining the function $\text{one-hot}(i^*)$ which sets the $i^*$-th element in the one-hot vector $z_{i^*} = 1$ and $z_{l\neq i^*} = 0$, the Gumbel-Max trick hence allows to sample one-hot vectors $\mathbf{z}$. Thus, we are able to reparametrize $\mathbf{z}$ through a continuous multivariate Gumbel RV $\mathbf{g} \in \mathbb{R}^{M\times 1}$ and a vector $\boldsymbol{\alpha} \in [0,1]^{M\times 1}$ of class probabilities $p(x = m_k)$ with $\sum_{k=1}^{M}\alpha_k = 1$:

$$\mathbf{z} = \text{one-hot}\left(\underset{i=1,\dots,M}{\arg\max}\ [\ln(\boldsymbol{\alpha}) + \mathbf{g}]\right). \tag{6}$$

Note that (6) and equally $x$ are still discrete RVs, i.e., $p(\mathbf{z}) \hat{=} p(x)$, but represented in probabilistic sense by continuous RVs $\mathbf{g}$. To arrive at a continuous RV, we now replace the one-hot and $\arg\max$ computation in (6) by the softmax function [35], [36]:

$$\tilde{\mathbf{z}} = \sigma_\tau(\mathbf{g}) = \frac{e^{(\ln(\boldsymbol{\alpha})+\mathbf{g})/\tau}}{\sum_{i=1}^{M} e^{(\ln\alpha_i + g_i)/\tau}}. \tag{7}$$

The resulting RV $\tilde{\mathbf{z}} \in [0,1]^{M\times 1}$ is the so called concrete or Gumbel-softmax RV and now continuous, e.g., $\tilde{\mathbf{z}} = [0.2, 0.8]^T$. It is controlled by a parameter, the softmax temperature $\tau$. The distribution of $\tilde{\mathbf{z}}$ in (7) was found to have a closed form density in [35], [36] which gives the definition of the concrete distribution:

$$p(\tilde{\mathbf{z}}|\boldsymbol{\alpha},\tau) = (M-1)!\ \tau^{M-1}\prod_{k=1}^{M}\left(\frac{\alpha_k \tilde{z}_k^{-\tau-1}}{\sum_{i=1}^{M}\alpha_i \tilde{z}_i^{-\tau}}\right). \tag{8}$$

With $\tilde{\mathbf{z}}$, we are finally able to relax the discrete RV $x$ into a continuous RV $\tilde{x}$ by defining $\tilde{x} = \tilde{\mathbf{z}}^T\mathbf{m}$. Now, our derivation of the relaxation is complete. In Fig. 1, we illustrate the distribution $p(\tilde{x})$ for the special case $M = 2$ of binary RVs in comparison to the original categorical pmf $p(x)$, i.e., a Bernoulli pmf. It has the following properties [35]: First, we are able to reparametrize the concrete RV $\tilde{\mathbf{z}}$ and hence the RV $\tilde{x}$ by Gumbel variables $\mathbf{g}$, a direct result from the initial idea (7). Moreover, the smaller $\tau$, the more $\tilde{\mathbf{z}}$ approaches a categorical RV and the approximation becomes more accurate. Thus, the statistics of $x$ and $\tilde{x}$ remain the same for $\tau \to 0$.

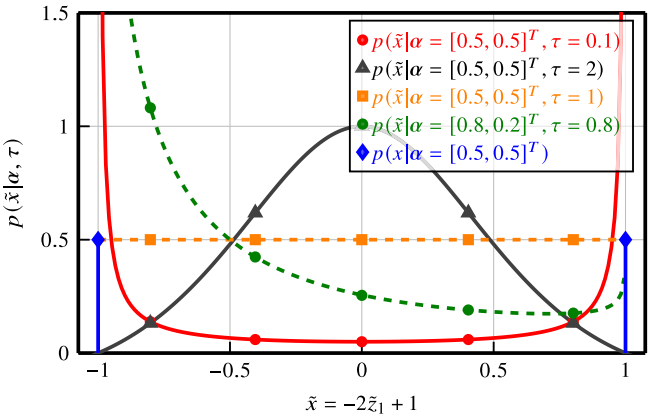


Fig. 1. The concrete pdf $p(\tilde{x}|\boldsymbol{\alpha},\tau)$ shown for different parameter sets and $M = 2$. It relaxes the Bernoulli pmf $p(x|\boldsymbol{\alpha})$ into the interior. Notably, for $\tau \leq (M-1)^{-1}$, it is log-convex and log-concave otherwise. Symmetry results if $\alpha_1 = \ldots = \alpha_M$.

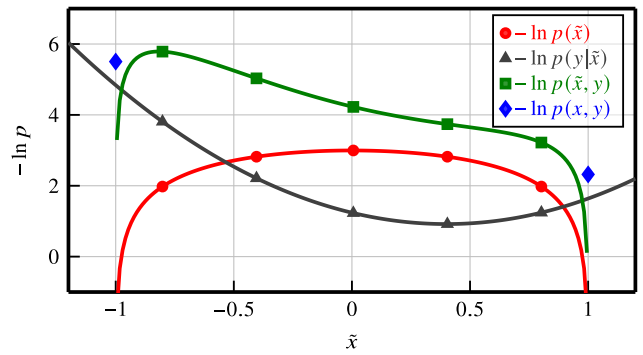


Fig. 2. Exemplary plot of the concrete binary MAP cost function (green) for model (1) (with $N_T = 1$, $\mathbf{H} = 1$, $y = 0.4$, $\sigma_n^2 = 4$, $\alpha_1 = 0.5$, $\alpha_2 = 0.5$ and $\tau = 0.1$) and the contribution of conditional (black) and prior pdf (red) to it. The original binary MAP cost function (blue) is shown for comparison.

### C. Reparametrization

In [37], our idea is to use the concrete distribution in order to relax the MAP problem (3c) to

$$\hat{\mathbf{x}} = \underset{\tilde{\mathbf{x}} \in [\min(\mathcal{M}), \max(\mathcal{M})]^{N_\mathrm{T} \times 1}}{\arg\min} \; - \ln p(\mathbf{y}|\tilde{\mathbf{x}}) - \ln p(\tilde{\mathbf{x}}). \quad (9)$$

Note that the original MAP problem is included or recovered in the zero temperature limit $\tau \to 0$. Moreover, the objective function in (9) may be non-convex as illustrated in Fig. 2 for $M = 2$. The conditional pdf $p(\mathbf{y}|\tilde{\mathbf{x}})$ is log-concave and the prior concrete pdf $p(\tilde{\mathbf{x}})$ log-convex for $\tau \leq (M-1)^{-1}$ [35], so the negative log joint distribution $p(\mathbf{y}, \tilde{\mathbf{x}})$ forms a non-convex objective function (9). The reparametrization of $\tilde{\mathbf{z}}$ by $\mathbf{g}$ helps to rewrite (9) by expressing each $\tilde{x}_n$ in $\tilde{\mathbf{x}}$ with (7) by the RV $\mathbf{g}_n$, $n = 1, \ldots, N_\mathrm{T}$, of i.i.d. Gumbel RVs $g_{kn}$:

$$\tilde{\mathbf{x}}(\mathbf{G}) = \begin{bmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_{N_\mathrm{T}} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{z}}_1^T \\ \vdots \\ \tilde{\mathbf{z}}_{N_\mathrm{T}}^T \end{bmatrix} \mathbf{m} = \begin{bmatrix} \sigma_\tau(\mathbf{g}_1)^T \\ \vdots \\ \sigma_\tau(\mathbf{g}_{N_\mathrm{T}})^T \end{bmatrix} \mathbf{m} \quad (10)$$

$$\text{with} \quad \mathbf{G} = \left[\mathbf{g}_1 \cdots \mathbf{g}_{N_\mathrm{T}}\right] \in \mathbb{R}^{M \times N_\mathrm{T}}. \quad (11)$$

By doing so, we will obtain an unconstrained optimization problem w.r.t. matrix $\mathbf{G}$. Now, we reformulate the relaxed MAP problem (9): This means, we replace the likelihood $p(\mathbf{y}|\tilde{\mathbf{x}})$ by $p(\mathbf{y}|\mathbf{G})$ and introduce the Gumbel distribution $p(g_{kn}) = \exp(-g_{kn} - \exp(-g_{kn}))$ as the new prior distribution:

$$\hat{\mathbf{G}} = \underset{\mathbf{G} \in \mathbb{R}^{M \times N_\mathrm{T}}}{\arg\min} \; - \ln p(\mathbf{y}|\mathbf{G}) - \ln p(\mathbf{G}) \quad (12a)$$

$$= \underset{\mathbf{G} \in \mathbb{R}^{M \times N_\mathrm{T}}}{\arg\min} \; - \ln p(\mathbf{y}|\mathbf{G}) - \sum_{n=1}^{N_\mathrm{T}} \sum_{k=1}^{M} \ln p(g_{kn}) \quad (12b)$$

$$= \underset{\mathbf{G} \in \mathbb{R}^{M \times N_\mathrm{T}}}{\arg\min} \; \underbrace{- \ln p(\mathbf{y}|\mathbf{G}) + \mathbf{1}^T \mathbf{G} \mathbf{1} + \mathbf{1}^T e^{-\mathbf{G}} \mathbf{1}}_{L(\mathbf{G}, \tau)}. \quad (12c)$$

However, due to the softmax and exponential terms in $L(\mathbf{G}, \tau)$, (12c) has no analytical solution. Furthermore, $L(\mathbf{G}, \tau)$ may be non-convex: For real-valued model (1) with $N_\mathrm{T} = 1$, $\mathbf{H} = 1$ and $M = 2$, the first term is a vertically shifted, squared and scaled two-dimensional non-convex sigmoid function w.r.t. $g_1$ and $g_2$. The operations applied to the sigmoid do not change non-convexity. Also the sum of this non-convex term and convex functions, i.e., linear and exponential functions, remains non-convex.

### D. Gradient Descent Optimization

One common strategy for solving the non-linear and non-analytical problem (12c) is to use a variant of gradient descent based approaches. Since we aim to reduce complexity, we choose the most basic form steepest descent. The minimum is approached iteratively by taking gradient descent steps until the necessary condition

$$\frac{\partial L(\mathbf{G}, \tau)}{\partial \mathbf{G}} = \mathbf{0} \quad (13)$$

is fulfilled. We point out that convergence to the global solution depends heavily on the starting point initialization since the objective function may be non-convex. A reasonable choice of starting point value is $\tilde{\mathbf{x}}^{(0)} = \mathrm{E}[\mathbf{x}] = \boldsymbol{\alpha}^T \cdot \mathbf{m}$, i.e., the expected value of the true discrete RV $\mathbf{x}$. We achieve this by setting $\mathbf{G}^{(0)} = \mathbf{0}$ and $\tau = 1$. After some tensor/matrix calculus and by noting that every $\tilde{x}_n$ only depends on one $\mathbf{g}_n$, the gradient descent step for (12c) in iteration $j$ is:

$$\mathbf{G}^{(j+1)} = \mathbf{G}^{(j)} - \delta^{(j)} \cdot \left.\frac{\partial L(\mathbf{G}, \tau)}{\partial \mathbf{G}}\right|_{\mathbf{G} = \mathbf{G}^{(j)}} \quad (14a)$$

$$\frac{\partial L(\mathbf{G}, \tau)}{\partial \mathbf{G}} = -\left[\frac{\partial \tilde{x}_1(\mathbf{g}_1)}{\partial \mathbf{g}_1} \cdots \frac{\partial \tilde{x}_{N_\mathrm{T}}(\mathbf{g}_{N_\mathrm{T}})}{\partial \mathbf{g}_{N_\mathrm{T}}}\right] \cdot \mathrm{diag}\left\{\frac{\partial \ln p(\mathbf{y}|\mathbf{G})}{\partial \tilde{\mathbf{x}}}\right\} + 1 - e^{-\mathbf{G}} \quad (14b)$$

$$\frac{\partial \tilde{x}_n(\mathbf{g}_n)}{\partial \mathbf{g}_n} = \frac{1}{\tau^{(j)}} \cdot \left[\mathrm{diag}\left\{\sigma_\tau(\mathbf{g}_n)\right\} \cdot \mathbf{m} - \sigma_\tau(\mathbf{g}_n) \cdot \tilde{x}_n(\mathbf{g}_n)\right]. \quad (14c)$$

The operator $\mathrm{diag}\{\mathbf{a}\}$ creates a diagonal matrix with the vector $\mathbf{a}$ on its main diagonal. The step-size $\delta^{(j)}$ can be chosen adaptively in every iteration $j$ just as the parameter $\tau^{(j)}$. For example, we can follow a heuristic schedule like in simulated annealing: We start with a large $\tau^{(j)}$ and decrease until we approach the true prior pdf for $\tau^{(j)} \to 0$. Finally, after the last iteration $N_\mathrm{it}$, we get as a result the continuous estimate $\mathbf{G}^{(N_\mathrm{it})}$. For approximate detection of $\mathbf{x}$ in (3c), the estimate has to be transformed back to the discrete domain by quantizing $\tilde{\mathbf{x}}$ onto the discrete set $\mathcal{M}$:

$$\hat{\mathbf{x}} = \underset{\mathbf{x} \in \mathcal{M}^{N_\mathrm{T} \times 1}}{\arg\min} \; \left\|\mathbf{x} - \tilde{\mathbf{x}}\left(\mathbf{G}^{(N_\mathrm{it})}\right)\right\|_2. \quad (15)$$

In the following, we name this detection approach Concrete MAP Detection (CMD). It is generic and applicable in any differentiable probabilistic non-linear model. For our guiding example of a linear Gaussian model (1), we are able to give the explicit expression of

$$\frac{\partial \ln p(\mathbf{y}|\mathbf{G})}{\partial \tilde{\mathbf{x}}} = -\frac{2}{\sigma_\mathrm{n}^2} \cdot \left[\mathbf{H}^H \mathbf{H} \tilde{\mathbf{x}}(\mathbf{G}) - \mathbf{H}^H \mathbf{y}\right] \quad (16)$$

in (14b). This means that further only elementwise nonlinearities and matrix vector multiplications are present in this example. As a final remark, we note that our implementation of Section IV relies on scaling of the objective function by the noise variance parameter, i.e., $\sigma_\mathrm{n}^2 \cdot L(\mathbf{G}, \tau)$. Although scaling does not change the optimization problem, we observed that this slightly modified version of (14) is numerically more stable.

### E. Special Case: Binary Random Variables

Noting that the softmax function (7) is normalized, we are able to eliminate one degree of freedom in matrix $\mathbf{G} \in \mathbb{R}^{M \times N_\mathrm{T}}$ along dimension $M$. For the special case of binary RVs or $M = 2$ classes, this means that the matrix $\mathbf{G}$ can be reduced to a vector $\mathbf{s} \in \mathbb{R}^{N_\mathrm{T} \times 1}$ of logistic RVs to derive a different algorithm of low complexity. Here, we only briefly summarize the result of binary CMD in a real-valued system

model and refer the reader to [37] for the complete derivation:

$$\mathbf{s}^{(j+1)} = \mathbf{s}^{(j)} - \delta^{(j)} \cdot \frac{\partial L(\mathbf{s},\tau)}{\partial \mathbf{s}}\bigg|_{\mathbf{s}=\mathbf{s}^{(j)}} \tag{17a}$$

$$\frac{\partial L(\mathbf{s},\tau)}{\partial \mathbf{s}} = -\frac{\partial \tilde{\mathbf{x}}(\mathbf{s})}{\partial \mathbf{s}} \cdot \frac{\partial \ln p(\mathbf{y}|\mathbf{s})}{\partial \tilde{\mathbf{x}}} + \tanh\left(\frac{\mathbf{s}}{2}\right) \tag{17b}$$

$$\overset{(1)}{=} \frac{1}{\sigma_{\mathrm{n}}^2} \cdot \frac{\partial \tilde{\mathbf{x}}(\mathbf{s})}{\partial \mathbf{s}} \cdot \left[\mathbf{H}^T\mathbf{H}\tilde{\mathbf{x}}(\mathbf{s}) - \mathbf{H}^T\mathbf{y}\right] + \tanh\left(\frac{\mathbf{s}}{2}\right) \tag{17c}$$

$$\frac{\partial \tilde{\mathbf{x}}(\mathbf{s})}{\partial \mathbf{s}} = \frac{1}{2\tau^{(j)}} \cdot \operatorname{diag}\left\{1 - \tilde{\mathbf{x}}^2(\mathbf{s})\right\} \tag{17d}$$

$$\tilde{\mathbf{x}}(\mathbf{s}) = \tanh\left(\frac{\ln\left(1/\alpha - 1\right) + \mathbf{s}}{2\tau^{(j)}}\right). \tag{17e}$$

The final step consists again of quantization - in this case it simplifies to the sign function: $\hat{\mathbf{x}} = \operatorname{sign}(\tilde{\mathbf{x}}(\mathbf{s}^{(N_{\mathrm{it}})}))$.

## III. Learning to Relax

Although being simple and computational efficient, using a gradient descent approach like (14) and (17) leads to several inconveniences. Regarding theoretical properties, a major drawback becomes apparent: Convergence of the gradient descent steps to an optimum is slow since consecutive gradients are perpendicular. Also practical questions arise: How to choose the parameters $\tau^{(j)}$ and $\delta^{(j)}$ and the number of iterations $N_{\mathrm{it}}$ for a good complexity performance trade off? And how are we able to deliver soft information, e.g., probabilities, to a soft decoder which is standard in today's communication systems?

Our idea is to improve CMD by learning and in particular the idea of deep unfolding to address these questions. This means we have to deal with

A. how learning is defined
B. the application of deep unfolding to CMD.

### A. Basic Problem of Learning

To introduce our notation of learning, we revisit our basic task of MAP detection. Ideally, we would like to infer the most likely transmit signal $\mathbf{x}$ based on an a-posteriori pdf $p(\mathbf{x}|\mathbf{y})$. But as pointed out earlier, evaluation of $p(\mathbf{x}|\mathbf{y})$ has intractable complexity. For this reason, we propose to relax the MAP problem and CMD, respectively.

Another idea to tackle this problem is to approximate this pdf $p(\mathbf{x}|\mathbf{y})$ by another computationally tractable pdf $q(\mathbf{x}|\mathbf{y})$, e.g., by calculation of $q(\mathbf{x}|\mathbf{y})$ using few samples/observations $\mathbf{x}$, and to use this pdf for inference. Note that this approach includes cases where we do not know the pdf $p(\mathbf{x}|\mathbf{y})$ completely. The quality of the approximation can be quantified by the information theoretic measure of Kullback-Leibler (KL) divergence:

$$D_{\mathrm{KL}}\left(p \parallel q\right) = \sum_{\mathbf{x}\in\mathcal{M}^{N_{\mathrm{T}}\times 1}} p(\mathbf{x}|\mathbf{y}) \ln \frac{p(\mathbf{x}|\mathbf{y})}{q(\mathbf{x}|\mathbf{y})} \tag{18}$$

$$= \mathrm{E}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{y})}\left[\ln \frac{p(\mathbf{x}|\mathbf{y})}{q(\mathbf{x}|\mathbf{y})}\right]. \tag{19}$$

Just as the Mean Square Error (MSE), the measure of KL divergence can be used to define an optimization problem targeting at a tight $q(\mathbf{x}|\mathbf{y})$ as a solution. This brings me to a crucial viewpoint of this article: **Learning is defined to be the optimization process aiming to derive a good approximation $q(\mathbf{x}|\mathbf{y})$ of $p(\mathbf{x}|\mathbf{y})$, i.e.,**

$$q^*(\mathbf{x}|\mathbf{y}) = \arg\min_{q} D_{\mathrm{KL}}\left(p \parallel q\right). \tag{20}$$

This kind of problem is also referred to as Variational Inference (VI). We can rewrite the KL divergence into a sum of cross entropy $\mathcal{H}(p,q)$ and entropy $\mathcal{H}(p)$:

$$D_{\mathrm{KL}}\left(p \parallel q\right) = \mathrm{E}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{y})}[-\ln q(\mathbf{x}|\mathbf{y})] - \mathrm{E}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{y})}[-\ln p(\mathbf{x}|\mathbf{y})] \tag{21}$$

$$= \mathcal{H}(p,q) - \mathcal{H}(p). \tag{22}$$

Since we defined the basic learning problem (20) w.r.t. approximation $q$, we can neglect the entropy term $\mathcal{H}(p)$ independent of $q$ and use cross entropy as the learning criterion. If we further restrict $q$ to a model $q(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})$ with parameters $\boldsymbol{\theta}$, the optimization problem now reads:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{H}(p,q). \tag{23}$$

We note that problem (23) is solved separately for each $\mathbf{y}$ and thus parameters $\boldsymbol{\theta}$ need to be continuously updated in an online learning procedure. Since this procedure is not computationally efficient, we follow an offline learning strategy known as Amortized Inference [23] and define one inference distribution $q(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})$ for any value $\mathbf{y}$:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}[\mathcal{H}(p(\mathbf{x}|\mathbf{y}), q(\mathbf{x}|\mathbf{y},\boldsymbol{\theta}))] \tag{24}$$

$$= \arg\min_{\boldsymbol{\theta}} \mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}\left[\mathrm{E}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{y})}[-\ln q(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})]\right] \tag{25}$$

$$\approx \arg\min_{\boldsymbol{\theta}} -\frac{1}{N}\sum_{i=1}^{N} \ln q(\mathbf{x}_i|\mathbf{y}_i,\boldsymbol{\theta}), \quad N\to\infty. \tag{26}$$

Rewriting the optimization criterion of (24) into (27), as shown at the bottom of the page,
for our guiding example (1), we note that we are able to amortize across all observations $\tilde{\mathbf{y}}$ from (2) and hence to obviate the need for online training also for each channel $\mathbf{H}$ and noise variance $\sigma_{\mathrm{n}}^2$ at the potential cost of accuracy.

The final result (26) equals the maximum likelihood problem in supervised learning. We make use of it in the following since it allows for numerical optimization based on $N$ data points $\{\mathbf{x}_i, \mathbf{y}_i\}$. Furthermore, it proves to be a Monte Carlo

$$\mathrm{E}_{\tilde{\mathbf{y}}\sim p(\tilde{\mathbf{y}})}\left[\mathrm{E}_{\mathbf{x}\sim p(\mathbf{x}|\tilde{\mathbf{y}})}[-\ln q(\mathbf{x}|\tilde{\mathbf{y}},\boldsymbol{\theta})]\right] = \mathrm{E}_{\sigma_{\mathrm{n}}^2\sim p(\sigma_{\mathrm{n}}^2)}\left[\mathrm{E}_{\mathbf{H}\sim p(\mathbf{H})}\left[\mathrm{E}_{\mathbf{y}\sim p(\mathbf{y}|\mathbf{H},\sigma_{\mathrm{n}}^2)}\left[\mathrm{E}_{\mathbf{x}\sim p(\mathbf{x}|\tilde{\mathbf{y}})}[-\ln q(\mathbf{x}|\tilde{\mathbf{y}},\boldsymbol{\theta})]\right]\right]\right] \tag{27}$$

approximation of (24) and is hence well motivated from information theory [23].

### B. Idea of Unfolding and Application to CMD

Learning gives us the ability to obtain a tractable approximation $q(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})$. But it remains one question: How to choose a suitable functional form of $q(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})$ of low complexity and for good performance? We follow the idea of deep unfolding from [25], [26] and apply it to our model-based approach CMD with parameters $\boldsymbol{\theta} = \left\{\tau^{(0)},\ldots,\tau^{(N_{\text{it}})},\delta^{(0)},\ldots,\delta^{(N_{\text{it}}-1)}\right\} \in \mathbb{R}^{(2N_{\text{it}}+1)\times 1}$ able to relax tightly. Thereby, we combine strengths of DNNs and the latter: DNNs are known to be universal approximators [5] and their fixed structure of parallel computations layer per layer allows to define a good performance complexity trade off at run time. But if the model is dynamic and changes, e.g., the channel or noise over time, reiterated optimization of (23), i.e., possibly wasteful online training, is required and the benefit disappears. Fortunately, we know our model (1), a MIMO channel, well and are able to use generative model-based approaches which mostly rely on a suitable approximation of (20) for computational tractability. For example, MFVI and AMP belong to this algorithm family. By model-based DNN design, we introduce varying model parameters like channel or noise explicitly and in a more sophisticated way into the DNN design and thus make efficient offline learning from (26) at only a small cost of accuracy possible. Indeed, training of a DNN for our guiding example (1) simply fed with inputs $\mathbf{y}$ and $\mathbf{H}$, reshaped as a vector, does not converge/lead to satisfactory results if trained offline [28].

This means we unfold the iterations (14) of CMD into a DNN by untying the parameters $\tau^{(j)}$ and $\delta^{(j)}$. Furthermore, we fix the complexity by setting the number of iterations $N_{\text{it}}$. The resulting graph illustrated in Fig. 3 for binary CMD and (1) has a DNN-like structure which should be able to generalize and approximate well at the same time. Owing to the skip connection from $\mathbf{s}^{(j)}$ to $\mathbf{s}^{(j+1)}$ on the right hand side, the structure resembles a Residual Network (ResNet) layer which is SotA in image processing [9]. It is a result of the gradient descent approach which allows to interpret optimization of ResNets as learning gradient descent steps. The reason for the success of ResNet lies in the skip connection: The training error is able to backpropagate through it to early layers which allows for fast adaptation of early weights and hence fast training of DNNs. This makes CMD especially suitable for online training proposed in [34] and allows for refinement in application.

As before, we have to define a final layer which is now also used for optimization. Usually, its output is chosen to be a continuous estimate of $\mathbf{x}$ and optimized w.r.t. the MSE criterion, see [28], [34]. This viewpoint relaxes the estimate $\hat{\mathbf{x}}$ into $\mathbb{R}^{N_{\text{T}}\times 1}$ and assumes a Gaussian distribution for errors at the output. In our case, the output would correspond to $\tilde{\mathbf{x}}(\mathbf{G}^{(N_{\text{it}})})$ from (15). But this is in contrast to our information theoretic viewpoint on learning which states that we want to approximate an output of the true pmf $p(\mathbf{x}|\mathbf{y})$. Like in MFVI, we assume a factorization of the approximating posterior to make it computationally tractable and derive our learning criterion:

$$\begin{aligned}
&\mathcal{H}\left(p,q\right)\\
&= -\sum_{\mathbf{x}\in\mathcal{M}^{N_{\text{T}}}} p(\mathbf{x}|\mathbf{y})\cdot\ln q(\mathbf{x}|\mathbf{y},\boldsymbol{\theta}) && (28)\\
&\overset{\text{MFVI}}{=} -\sum_{\mathbf{x}\in\mathcal{M}^{N_{\text{T}}}} p(\mathbf{x}|\mathbf{y})\cdot\ln\prod_{n=1}^{N_{\text{T}}} q_n(x_n|\mathbf{y},\boldsymbol{\theta}) && (29)\\
&= -\sum_{n}\sum_{x_n\in\mathcal{M}}\sum_{\mathbf{x}\setminus x_n\in\mathcal{M}^{N_{\text{T}}-1}} p(\mathbf{x}|\mathbf{y})\cdot\ln q_n(x_n|\mathbf{y},\boldsymbol{\theta})\\
&= -\sum_{n}\sum_{x_n\in\mathcal{M}} p(x_n|\mathbf{y})\cdot\ln q_n(x_n|\mathbf{y},\boldsymbol{\theta}) && (30)\\
&= \sum_{n=1}^{N_{\text{T}}}\mathcal{H}\left(p(x_n|\mathbf{y}),q_n(x_n|\mathbf{y},\boldsymbol{\theta})\right). && (31)
\end{aligned}$$

This interesting result shows that assuming MFVI factorization leads to an optimization criterion w.r.t. the soft output $p(x_n|\mathbf{y})$ of the IO detector (4). This soft output is required for subsequent decoding and thus exactly what we need.

The last step of our idea consists of inserting our unfolded CMD structure into $q_n(x_n|\mathbf{y},\boldsymbol{\theta})$. Hence, we propose to use a softmax function for the last layer being a typical choice for classification in discriminative probabilistic models. Fortunately, CMD already includes this softmax function as part of its structure so we rewrite

$$q_n(x_n|\mathbf{y},\boldsymbol{\theta}) = \prod_{k=1}^{M} q_{n,k}(x_n|\mathbf{y},\boldsymbol{\theta})^{(x_n=m_k)} = \prod_{k=1}^{M}\tilde{z}_{n,k}^{(x_n=m_k)} \tag{32}$$

with $\tilde{\mathbf{z}}_n = \sigma_{\tau^{(N_{\text{it}})}}(\mathbf{g}_n^{(N_{\text{it}})})$ from the last iteration $N_{\text{it}}$ of (14). To summarize, we optimize the parameter set $\boldsymbol{\theta}$ of our approximating pdf $q(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})$ based on CMD:

$$\begin{aligned}
\boldsymbol{\theta}^* &= \arg\min_{\boldsymbol{\theta}}\ \mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}[\mathcal{H}\left(p(\mathbf{x}|\mathbf{y}),q(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})\right)] && (33)\\
&\approx \arg\min_{\boldsymbol{\theta}}\ -\frac{1}{N}\sum_{i=1}^{N}\sum_{n=1}^{N_{\text{T}}}\begin{bmatrix} x_{i,n}=m_1\\ \vdots\\ x_{i,n}=m_M\end{bmatrix}^T\\
&\quad\cdot\ln\left(\sigma_{\tau^{(N_{\text{it}})}}(\mathbf{g}_n^{(N_{\text{it}})})\right). && (34)
\end{aligned}$$

As a side effect, we also learn to relax with CMD by $\tau^{(j)}$. We call this approach based on unfolding of CMD CMDNet. The optimization problem (34) can be efficiently solved by variants of SGD. Thanks to having a model, we are able to create infinite training and test data for reasonable approximation of (33) by (34) in every iteration of SGD. We notice that this is in contrast to classic data sets from the machine learning community.

## IV. Numerical Results

### A. Implementation Details/Settings

In order to evaluate the performance of the proposed approaches CMD and CMDNet, we present numerical simulation results of application in our guiding example for

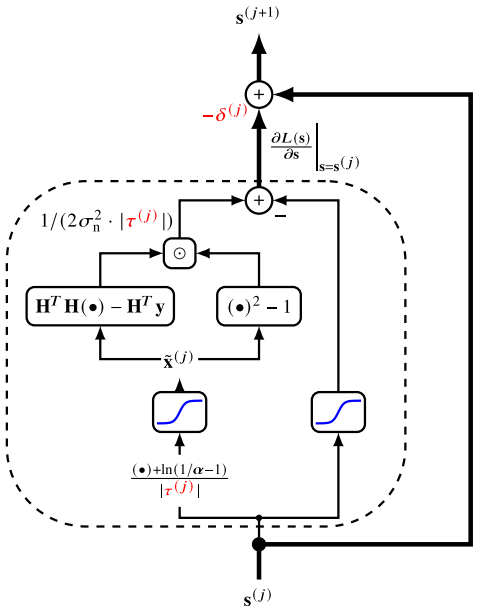


Fig. 3. One layer of the unfolded binary CMD algorithm CMDNet when applied to MIMO systems. In red: trainable parameters.

TABLE I
SIMULATION SCENARIOS

| Scenario | Sys. Dim. | Mod. | Corr. | Coding |
|---|---|---|---|---|
| Large MIMO | $32 \times 32$ | QPSK | no | no |
| MIMO | $8 \times 8$ | QPSK | no | no |
| Multi-class | $32 \times 32$ | 16-QAM | no | no |
| massive MIMO One-Ring | $64 \times 32$ | QPSK | 20° | no |
| Soft Output | $32 \times 32$ | QPSK | no | LDPC |

different MIMO systems with $N_\mathrm{T}$ transmit and $N_\mathrm{R}$ receive antennas given in Tab. I. We assume an uplink scenario with multiple UEs, each transmitting one symbol $x_n$ with equal a-priori probabilities $\alpha_1 = \ldots = \alpha_M$ to one BS. As an example, we assume the number of iterations or layers to be $N_\mathrm{it} = N_\mathrm{L} = 2N_\mathrm{T}$. For numerical optimization of the parameters $\delta^{(j)}$ and $\tau^{(j)}$ of CMDNet according to (34), we employ the Tensorflow framework in Python [7]. Here, we use Adam (Adaptive Moment Estimation) as a popular variant of SGD with a default batch size of $N_\mathrm{b} = 500$ and $N_\mathrm{epoch} = 10^5$ training iterations. Although providing fast convergence and requiring little hyperparameter tuning, it is known to generalize poorly [38]. Since we are able to generate a sufficient amount of training data, i.e., $N = N_\mathrm{b} \cdot N_\mathrm{epoch} = 5 \cdot 10^7$ to fulfill (33) by (34) approximately, we make sure that generalization to unseen data points is possible. As Tensorflow does not natively support computation with complex numbers, we transform the complex-valued system model (1) into its real-valued equivalent to allow for training and comparison to DNN-based approaches. This means, we restrict to QAM constellations with Gray encoding so that we have $\mathbf{x} \in \mathcal{M}^{2N_\mathrm{T}\times 1}$. As a training default, we choose the noise variance statistics $p(\sigma_\mathrm{n}^2)$ such that $E_\mathrm{b}/N_0 = 10\log_{10}(1/\sigma_\mathrm{n}^2) - 10\log_{10}(\log_2(M))$ is

TABLE II
SELECTED DETECTION ALGORITHMS

| Abbrev. | Complexity | Literature |
|---|---|---|
| MAP / SD | $\mathcal{O}(M^{\gamma N_\mathrm{T}})$, $\gamma \in (0,1]$ | [20] |
| SDR | $\mathcal{O}(\max(N_\mathrm{R}, N_\mathrm{T})^3 N_\mathrm{T}^{1/2} \log(1/\epsilon))$ | [22] |
| OAMPNet | $\mathcal{O}(N_\mathrm{L} N_\mathrm{T}^3)$ | [33] |
| MMSE / MOSIC | $\mathcal{O}(N_\mathrm{T}^3)$ | [34], [21] |
| DetNet | $\mathcal{O}(N_\mathrm{L}(N_\mathrm{T}N_\mathrm{R} + N_\mathrm{T}^2 M))$ | [27], [28] |
| MMNet (iid) | $\mathcal{O}(N_\mathrm{L} N_\mathrm{T}(N_\mathrm{T} + N_\mathrm{R} + M))$ | [34] |
| AMP | $\mathcal{O}(N_\mathrm{it} N_\mathrm{T}(N_\mathrm{R} + M))$ | [24] |
| CMD/ CMDNet | $\mathcal{O}(N_\mathrm{L} N_\mathrm{T}(N_\mathrm{R} + M))$ | [37] |
| MF | $\mathcal{O}(N_\mathrm{T} N_\mathrm{R})$ | |

uniformly distributed between $[4, 27]$ dB. We set the default parameter starting point to $\boldsymbol{\theta}_0$ with constant $\delta_0^{(j)} = 1$ and heuristically motivated and linear decreasing

$$\tau_0^{(j)} = \tau_\mathrm{max} - (\tau_\mathrm{max} - 0.1)/N_\mathrm{it} \cdot j \tag{35}$$

with $\tau_\mathrm{max} = 1/(M-1)$, $j \in [0, N_\mathrm{it}]$. With this choice, $p(\tilde{x})$ is always log-convex and hence reasonably approximating $p(x)$ (see Fig. 1). For training of DNN-based approaches DetNet and MMNet, we used the original implementations uploaded to GitHub (see [28], [34]) with only minor modifications to parametrization if beneficial. Consequently, we trained MMNet with CMDNet training SNR and layer number. Since we focus on offline derived or trained algorithms which are used for inference at run time, we used its i.i.d. variant. We always used the soft output version of DetNet with output normalization to 1 since we noted that performance is close to or better than the hard decision version. Furthermore, we compare CMD and CMDNet to several SotA approaches for MIMO detection (see Tab. II) choosing the number of Monte Carlo runs with data batches of size 10000 so that always 1000 errors are detected (100 for SD and SDR).

### B. Symmetric MIMO System

First, we test application of CMDNet in a large symmetric $32 \times 32$ / $64 \times 64$ MIMO system with i.i.d. Gaussian channel statistics $p(\mathbf{H})$ and QPSK/BPSK modulation. Fig. 4 shows the results in terms of BER as a function of $E_\mathrm{b}/N_0$. Owing to near-optimal performance, the SD is always provided as a benchmark in the following. In addition, we give the AWGN curve as a reference since it shows the maximum accuracy if $N_\mathrm{T} = N_\mathrm{R} \to \infty$ [24].

Linear detectors perform bad in this setup: Since the curve of the MF remains almost constant at BER $\approx 20\%$ and the Zero Forcer performs even worse, both are not shown in the following. At least, MMSE equalization leads to an acceptable BER but the curve is still separated by a 7 dB gap at $E_\mathrm{b}/N_0 = 13$ dB from SD's. In contrast, nonlinear SotA detectors like MOSIC, AMP and SDR technique (see Sec. I for algorithm details) have a strikingly better accuracy. Whereas AMP runs into an error floor for high SNR since then the message statistics are not Gaussian anymore in finite small-dimensional MIMO systems [24], SDR proves to be a

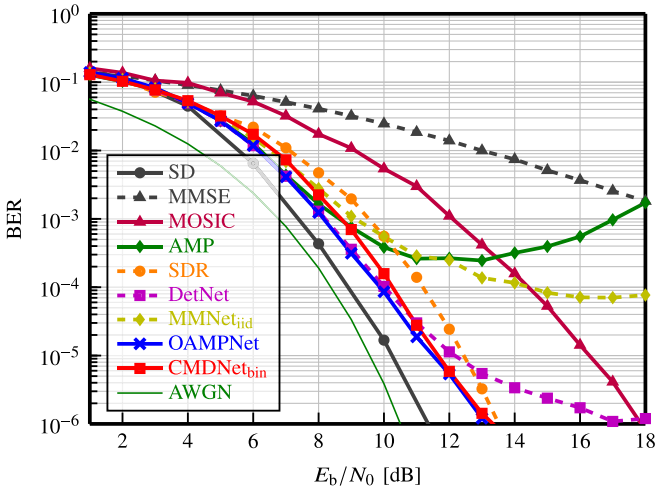


Fig. 4. BER curves of several detection methods in a $32 \times 32$ MIMO system with QPSK modulation. Effective system dimension is $64 \times 64$ and for iterative algorithms $N_{it} = N_L = 64$.

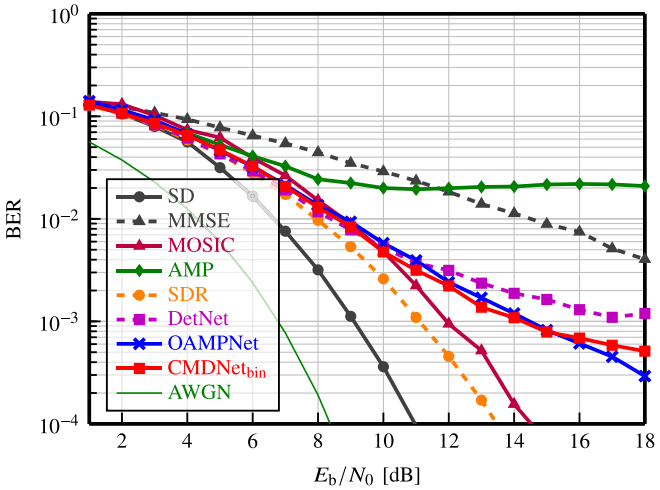


Fig. 5. BER curves of several detection methods in a $8 \times 8$ MIMO system with QPSK modulation. Effective system dimension is $16 \times 16$ and for iterative algorithms $N_{it} = N_L = 16$.

close relaxation by only dropping the non-convex requirement of $\text{rank}(\mathbf{x}\mathbf{x}^T) = 1$ [22].

Notably, our approach CMDNet in its binary version CMDNet$_{bin}$ from (17) performs even better than the latter, comparable to the best suboptimal approaches in this setup DetNet and OAMPNet. Further, CMDNet$_{bin}$ does not run into an error floor in the simulated SNR range like AMP and DetNet. Setting the accuracy in context to complexity (see Tab. II), this is impressive: Note that our approach is similar in asymptotic complexity to the light-weight algorithm AMP with $\mathcal{O}(N_L N_T(N_R + M))$ at inference run time after offline training whereas DetNet and OAMPNet are very complex DNN architectures. In particular, OAMPNet requires one costly matrix inversion per iteration resulting in high $\mathcal{O}(N_L N_T^3)$. In Sec. IV-G and Fig. 12, we give a more detailed complexity analysis and comparison illustrating CMD's promising accuracy complexity trade-off more clearly. In contrast, the other DNN-based approach MMNet$_{iid}$ with comparable low complexity fails to beat CMDNet$_{bin}$ and runs into an early error floor. Since we observed this behavior similar to AMP in all settings and MMNet is actually designed to perform well with fast online training, we omit further results. We conjecture that the denoising layers are insufficient expressive in the interference limited high SNR region with offline training.

Results in a smaller $8 \times 8$ MIMO system plotted in Fig. 5, show that all soft non-linear approaches except for SDR and MOSIC run into an error floor at lower SNR. Thus, we conjecture that they share the same suboptimality at finite system dimensions. They may rely on the statistics of the interference terms to be Gaussian like AMP which is only approximately true for large system dimensions. Apart from SDR and MOSIC, CMDNet$_{bin}$ manages to beat the more expressive and complex DNN models, i.e., DetNet and OAMPNet, and is close in accuracy to SDR for $E_b/N_0 < 10$ dB.

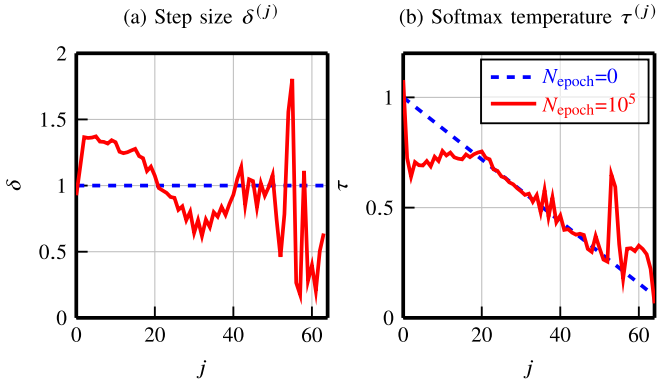


Fig. 6. Parameters $\boldsymbol{\theta}$ of CMDNet$_{bin}$ in a $32 \times 32$ MIMO system with QPSK modulation. Effective system dimension is $64 \times 64$.

### C. Algorithm and Parametrization

To investigate the influence of learning on CMDNet$_{bin}$ and the values of its parameters $\boldsymbol{\theta}$, we visualize them per layer $j$ in Fig. 6 for the $32 \times 32$ MIMO system considered before. Basically, we cannot observe any pattern after parameter optimization and interpretation seems very difficult.

Furthermore, we notice from Fig. 7 that starting point initialization $\boldsymbol{\theta}_0$ has a large impact on the optimum $\boldsymbol{\theta}_{10^5}$ found by SGD (after $N_{epoch} = 10^5$ iterations). If we use a starting point $\boldsymbol{\theta}_{0,splin}$ with linear decreasing

$$\tau_{0,splin}^{(j)} = \delta_{0,splin}^{(j)} = 1 - (1 - 0.01)/N_{it} \cdot j \tag{36}$$

for $j \in [0, N_{it}]$, a solution $\boldsymbol{\theta}_{10^5,splin}$ is learned allowing CMDNet to perform better in the low $E_b/N_0$ region from 6 to 10 dB. Notably, CMDNet even reaches the performance of the best suboptimal algorithm considered in this setup OAMPNet. To explain the error floor in the interference limited higher $E_b/N_0$ region in contrast to CMDNet with default training, we conjecture that a higher starting and correlating end step size (see Fig. 6) allows CMDNet to leave a local optimum with higher probability and to find a better one. On the contrary, a small step size enforces convergence to a local solution. In the noise limited $E_b/N_0$ region, noise removal is crucial and

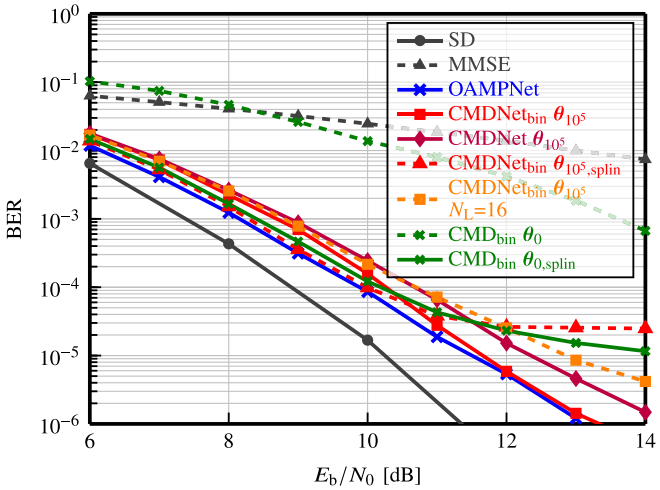


Fig. 7. BER curves of CMD and CMDNet with different parametrization or algorithmics in a $32 \times 32$ MIMO system with QPSK modulation. Effective system dimension is $64 \times 64$. Default number of iterations or layers is $N_{it} = N_L = 64$.

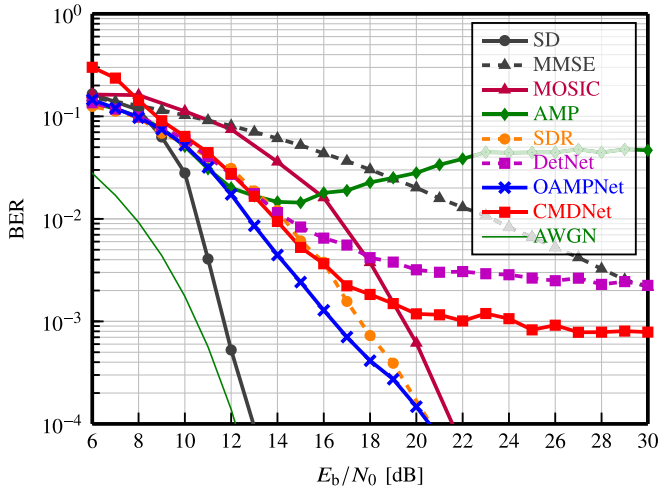


Fig. 8. BER curves of several detection methods in a $32 \times 32$ MIMO system with 16-QAM modulation. Effective system has dimension $64 \times 64$ and 4-ASK modulation and for iterative algorithms $N_{it} = N_L = 64$.

hence convergence. This means CMDNet can be optimized to different working points and is sensitive to starting point initialization. The result supports our view of a promising accuracy complexity trade-off: Since CMDNet only has a small parameter set, we are able to load the $\boldsymbol{\theta}$ dynamically for each $E_b/N_0$ to achieve the performance of the best suboptimal algorithm in all $E_b/N_0$ regions.

In particular, we are able to further decrease the number of parameters with negligible performance loss: CMDNet with only $N_L = 16$ layers performs equally well compared to default CMDNet with $N_L = 64$ at low $E_b/N_0$ and slightly worse at $E_b/N_0 = 12$ dB by 1 dB.

Without unfolding, heuristics for parameter selection are required similar to starting point initialization. The detection accuracy of CMD with such heuristic parameters $\boldsymbol{\theta}_{0,splin}$ is quite impressive since the BER curve is close to that of learned CMDNet with $\boldsymbol{\theta}_{10^5,splin}$. Therefore, we are able to use the plain algorithm CMD for detection. We note that this is not true with default parameters $\boldsymbol{\theta}_0$ and that performance can be quite different after optimization ($\boldsymbol{\theta}_{10^5}$).

Finally, we compare the accuracy of algorithm CMDNet$_{bin}$ for the special case of binary RVs from (17) with that of the generic multi-class algorithm CMDNet from (14) since both are different. From Fig. 7, we observe that the performance is very similar and conjecture that CMDNet is capable of achieving the same accuracy if training is parameterized correctly.

### D. Multi-Class Detection

So far, only BPSK modulation and hence two classes have been considered. To test multi-class detection with $M = 4$ classes, we show numerical results in a $32 \times 32$ MIMO system with 16-QAM modulation being equivalent to a $64 \times 64$ 4-ASK MIMO system after transformation into the equivalent real-valued problem. Owing to now 3 degrees of freedom in the softmax function and denser symbol packing, we changed our batch size to $N_b = 1500$ and training SNR to higher $E_b/N_0 \in [10, 33]$, respectively. Setting the default starting point with $\tau_{max} = 2/(M-1) = 2/3$ so that the MAP criterion $\ln p(\tilde{\mathbf{x}}, \mathbf{y})$ becomes convex for a couple of iterations proves to be crucial for successful training of CMDNet with multiple classes. Without training parameter tuning, CMDNet performs even worse than the MMSE detector.

Fig. 8 shows BER curves in this system. Clearly, we can now observe a large gap between the BER curve of SD and that of all other suboptimal approaches. Comparing the latter, OAMPNet is superior over the whole SNR region. Observing a maximum 2 dB curve shift, we note that CMDNet is competitive to OAMPNet and SDR at $E_b/N_0 \in [10, 17]$ and when BER $= [10^{-2}, 10^{-3}]$ which is a typical working point of decoders whereas being much less complex. At higher SNR, an error floor follows. Although using a more expressive DNN model, DetNet now trained for $E_b/N_0 \in [9, 16]$ fails to beat CMDNet especially in this region.

### E. Massive MIMO System

Investigation in large symmetric MIMO systems reveals the potential and shortcomings of the algorithms. Rather in 5G, massive MIMO systems with $N_R > N_T$ are employed [19]. Assuming i.i.d. Gaussian channels, we shortly report the results of a $64 \times 32$ MIMO system with QPSK modulation: The BER curves of learning based approaches and SDR almost follow that of SD and thus suggest that they fit perfectly for application in massive MIMO.

However in practice, channels are spatially correlated at the receiver side due to good spatial resolution of BS' large arrays compared to the number of scattering clusters [19]. Hence, the results for i.i.d. Gaussian channel statistics $p(\mathbf{H})$ are less meaningful as noted in [34]. As a first and quick attempt towards a realistic channel model which captures its key characteristics, we test performance in the so-called One-ring model $p(\mathbf{H})$ assuming a BS equipped with a uniform linear antenna array [19], [28]. We parameterize the correlation matrices of every column in $\mathbf{H}$ with reasonable values: Assuming an urban cellular network, we set the angular spread to $20°$ and sample the nominal angle uniformly from $[-60°, 60°]$,

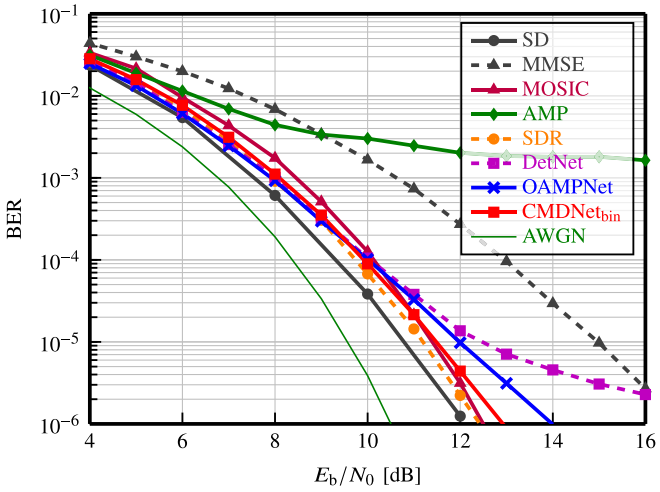


Fig. 9. BER curves of several detection methods in a correlated $64 \times 32$ MIMO system with QPSK modulation. The correlation matrices were generated according to a One-Ring model with 20° angular spread and 120° cell sector. Effective system dimension is $128 \times 64$ and for iterative algorithms $N_{\text{it}} = N_{\text{L}} = 64$.

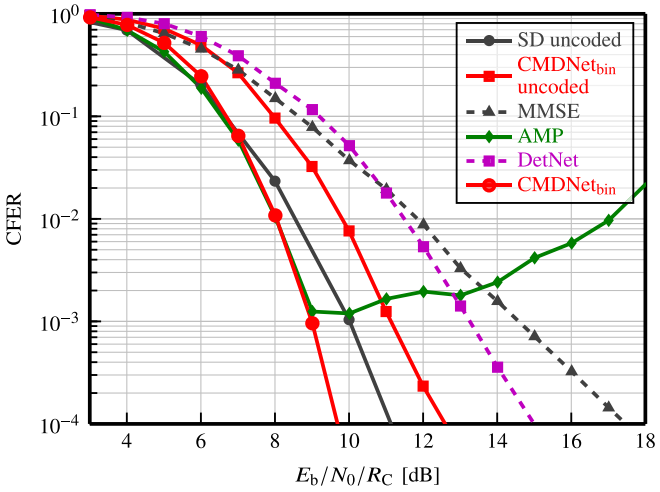


Fig. 10. CFER curves of a horizontally coded $32 \times 32$ MIMO system with QPSK modulation. A $128 \times 64$ LDPC code with belief propagation decoder was used. Effective system dimension is $64 \times 64$ and for iterative soft detectors $N_{\text{it}} = N_{\text{L}} = 64$.

i.e., 120° cell sector. Further, we place the antennas at half a wavelength distance.

From Fig. 9, it becomes evident that the performance loss of learning based approaches compared to SD in such a One-Ring model of dimension $64 \times 32$ is similar to the symmetric setting $32 \times 32$ in Fig. 4. Surprisingly, MOSIC and SDR now prove to be comparable whereas the BER of AMP degrades since the i.i.d. Gaussian channel assumption is not fulfilled anymore. Again, CMDNet outperforms other learning-based approaches DetNet and OAMPNet and performs very close to the best suboptimal algorithm SDR whereas being much less complex (see Tab. II and Fig. 12).

Considering the low complexity, we finally conclude that CMDNet performs surprisingly well in all previous settings. Hence, it proves to be a generic and hence promising detection approach.

### *F. Soft Output (Coded MIMO System)*

After investigation of detection performance in uncoded systems, we turn to an interleaved and horizontally coded $32 \times 32$ MIMO system with Rayleigh block fading reflecting our uplink model. We aim to verify whether not only hard decisions but also soft outputs generated by CMDNet and the soft output version of DetNet have high quality. This is especially important in practice since coding is an essential component besides equalization in today's communication systems. Therefore, we use a $128 \times 64$ LDPC code with rate $R_{\text{C}} = 1/2$ from [39] and at receiver side a belief propagation decoder with 10 iterations. The results in terms of Coded Frame Error Rate (CFER) as a function of $E_{\text{b}}/N_0/R_{\text{C}}$ are shown in Fig. 10. Owing to overwhelming computational complexity, we refrained from using the MAP solution with coding as a benchmark and instead show uncoded CMDNet and SD curves for reference. Strikingly, CMDNet with coding beats the latter and allows for a coding gain. In contrast, AMP with coding runs into an error floor after 9 dB: The output statistics become unreliable for high SNR in finite dimensional systems [24]. Surprisingly, although being one of the best detection methods in the uncoded setting, DetNet with coding performs close to MMSE equalization with soft outputs and thus worse than expected. Actually, the soft output version of DetNet should deliver accurate probabilities or Log Likelihood Ratios (LLRs) according to [28] after optimization.

Indeed, we visualize with an exemplary histogram of LLRs that this is not the case. In Fig. 11, we show the relative frequencies of LLRs of one symbol $x_n$ in one random channel realization $\mathbf{H}$ for $E_{\text{b}}/N_0 = 10$ dB. First, we note the histograms for $x_n = -1$ and $x_n = 1$ to be symmetric meaning that both algorithms fulfill a basic quality criterion. Furthermore, it can be clearly seen that DetNet mostly provides hard decisions with $\approx 97\%$ LLRs being $-\infty$ and $\infty$, respectively. Only a few values are close to 0. In contrast, CMDNet provides meaningful soft information resembling a mixture of Gaussians as expected from literature [40] ranging from $-30$ to 30. These results strongly indicate that the difference of soft output quality originates from different underlying optimization strategies: As pointed out in Section III-B, CMDNet relies on minimization of KL divergence between IO a-posteriori and approximating softmax pmf whereas the one-hot representation in DetNet is optimized w.r.t. MSE. We conclude that our approach yields a better optimization strategy.

### *G. Complexity Analysis*

Since complexity is the main driver for development of suboptimal algorithms like CMD instead of relying on MAP detection, we complete our numerical study by relating detection accuracy to results on the computational complexity given in Tab. II. With regard to CMD and CMDNet applied in our guiding example (1), the iterative asymptotic complexity of $\mathcal{O}(N_{\text{T}}(2N_{\text{R}} + 4M))$ or $\mathcal{O}(2N_{\text{T}}N_{\text{R}})$ for binary RVs is dominated by the matrix vector multiplications in $\mathbf{H}^T\mathbf{H}\tilde{\mathbf{x}}$,

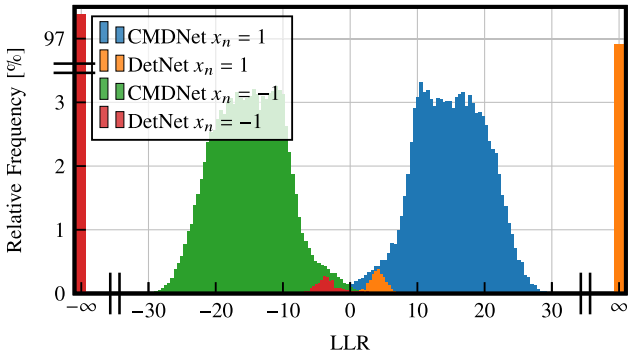


Fig. 11. Exemplary histogram showing the relative frequencies of LLRs of one symbol $x_n$ in one random channel realization $\mathbf{H}$ at $E_b/N_0 = 10$ dB.

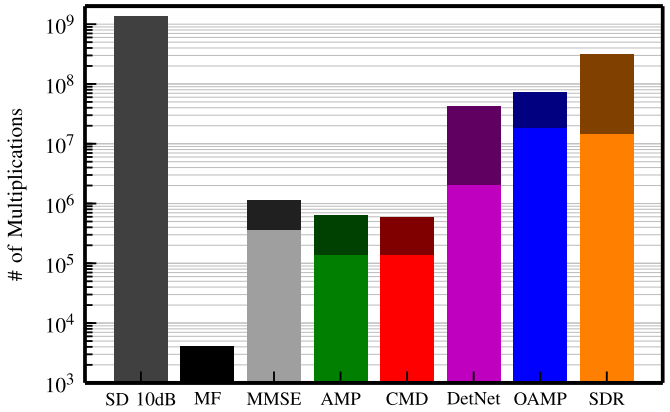


Fig. 12. Complexity of detection algorithms in terms of number of multiplicative operations in a $32 \times 32$ / $64 \times 64$ MIMO system: Light colored bars indicate a realistic low-complexity implementation with BPSK and dark colored bars the worst-case complexity with 16-QAM modulation.

i.e., CMD scales linearly with the input and output dimension as well as the number of classes. Clearly, CMD and CMDNet have very low complexity comparable to AMP and MMNet but with remarkable higher detection rate (see, e.g., Fig. 4). In most analyzed scenarios, the accuracy is even higher than DetNet's as well as OAMPNet's and on par with SDR's.

Besides qualitative $\mathcal{O}(\cdot)$ analysis, we capture complexity quantitatively by counting the number of Multiplicative OPerations (MOPs) for one iteration and channel realization being the most common and costly floating point operations. In Fig. 12, we show the respective bar chart assuming a realistic low-complexity implementation in a $32 \times 32$ with QPSK ($M = 2$) and $N_L = 16$ and worst-case complexity implementation with 16-QAM modulation ($M = 4$) and $N_L = 64$, respectively. For BPSK and the lower bar of MMSE equalization, we assumed Gaussian elimination to solve the linear equation system and, for higher order QAM and the higher bar, LU decomposition. We estimate the upper bound on SDR MOP count by unadapted $\mathcal{O}(\max(N_R, N_T)^4 N_T^{1/2} \log(1/\epsilon))$ and the lower bound on MOPs to account for half of the FLOPS from [28] with inaccurate $\epsilon = 0.1$. The expected number of visiting nodes $\mathcal{O}(M^{\gamma N_T})$ of the SD is SNR dependent with $\gamma \in (0, 1]$ and was extracted from [20] for $E_b/N_0 = 10$ dB.

Apparently, only the very basic MF beats CMD and CMDNet in complexity at considerably worse detection accuracy. Approaches with comparable accuracy like DetNet, OAMPNet and SDR are 10-100 times more complex w.r.t. MOPs. We conclude that CMDNet offers an excellent accuracy complexity trade-off and note that AMP, MMNet, DetNet and CMDNet further come with the benefit of already delivering soft outputs.

As a final remark, note that complexity analysis depends on the assumptions made: If we, e.g., assume long channel coherence time intervals, MMSE and MOSIC are able to reuse its computations with only one matrix vector multiplication remaining for any further detection inside the interval effectively decreasing complexity. For the same reason, online learning approaches do not require further training inside the interval and could be feasible. Comparing training cost of all unfolding algorithms in Tab. III, we note that $N_b$ and $N_{epoch}$ lie in the same range. Hence, the forward pass of backpropagation

TABLE III
TRAINING COMPLEXITY

| Algo. | $\approx N_b$ | $\approx N_{epoch}$ | $\lvert\boldsymbol{\theta}\rvert$ |
|---|---|---|---|
| DetNet | 2000 | $10^5$ | $N_L[(\{2,4\}M + \{6,20\})N_T^2$ |
| {QPSK, 16-QAM} | -5000 | | $+(M + \{3,6\})N_T + 2]$ |
| OAMPNet | 1000 | $10^4$-$10^5$ | $2N_L$ |
| MMNet {iid, full} | 500 | $10^4$-$10^5$ | $\{2N_L, N_L N_T(N_R + 1)\}$ |
| CMDNet | 500 | $10^4$-$10^5$ | $2N_L + 1$ |

in SGD and respectively run time complexity from Fig. 12 as well as the number of parameters $|\boldsymbol{\theta}|$ to be optimized dominate training complexity. OAMPNet fails in the former and DetNet in the latter category with $|\boldsymbol{\theta}| \in [10^5, 10^7]$ assuming $N_L = \{16, 64\}$ and {QPSK, 16-QAM}. In contrast, CMDNet with low runtime complexity and only $|\boldsymbol{\theta}| = \{33, 129\}$ may be a promising online training approach similar to MMNet [34].

## V. CONCLUSION

In this article, we introduced the so called continuous relaxation of discrete RVs to the MAP detection problem. Allowing to replace exhaustive search by continuous optimization, we defined our classification approach Concrete MAP Detection (CMD), e.g., based on gradient descent. By unfolding CMD into a DNN CMDNet, we further were able to optimize its low number of parameters and hence to improve detection accuracy while limiting it to low complexity. As a side effect, the resulting structure has the potential to allow for fast online training. Using the example of MIMO detection, simulations reveal CMDNet to be a generic detection method competitive to SotA outperforming it in terms of complexity and other recently proposed ML-based approaches DetNet and MMNet in every considered scenario. Notably, we selected an optimization criterion grounded in information theory, i.e., cross entropy, and showed that it aims at learning an approximation of the individual optimal detector. By simulations in coded systems, we demonstrated its ability to provide reliable soft outputs as opposed to [28], being a

requirement for soft decoding, a crucial component in today's communication systems.

All these findings prove CMDNet to be a promising detection approach for application in future massive MIMO systems. Further research is required to evaluate its potential for fast online learning and to demonstrate its applicability to non-linear scenarios of other research domains.

## REFERENCES


[1] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.

[2] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.

[3] D. D. Lin and T. J. Lim, "A variational inference framework for soft-in soft-out detection in multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2345–2364, May 2009.

[4] E. Riegler, G. E. Kirkelund, C. N. Manchón, M.-A. Badiu, and B. H. Fleury, "Merging belief propagation and the mean field approximation: A free energy approach," *IEEE Trans. Inf. Theory*, vol. 59, no. 1, pp. 588–602, Jan. 2013.

[5] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, Jan. 1989.

[6] O. Simeone, "A very brief introduction to machine learning with applications to communication systems," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 4, pp. 648–664, Dec. 2018.

[7] M. Abadi *et al.* (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: https://www.tensorflow.org/

[8] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist. (AISTATS)*, Ft. Lauderdale, FL, USA, Jun. 2011, pp. 315–323.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[10] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 3642–3649.

[11] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[12] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. 27th Conf. Adv. Neural Inf. Process. Syst. (NIPS)*, Montreal, QC, Canada, 2014, pp. 2672–2680.

[13] N. Farsad and A. Goldsmith, "Neural network detection of data sequences in communication systems," *IEEE Trans. Signal Process.*, vol. 66, no. 21, pp. 5663–5678, Nov. 2018.

[14] 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.

[15] H. Kim, Y. Jiang, S. Kannan, S. Oh, and P. Viswanath, "Deepcode: Feedback codes via deep learning," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 194–206, May 2020.

[16] 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.

[17] 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.

[18] A. Caciularu and D. Burshtein, "Unsupervised linear and nonlinear channel equalization and decoding using variational autoencoders," *IEEE Trans. Cognit. Commun. Netw.*, vol. 6, no. 3, pp. 1003–1018, Sep. 2020.

[19] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Found. Trends Signals Process.*, vol. 11, nos. 3–4, pp. 154–655, 2017.

[20] J. Jaldén and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1474–1484, Apr. 2005.

[21] D. Wübben, R. Böhnke, V. Kühn, and K.-D. Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition," in *Proc. IEEE 58th Veh. Technol. Conf. (VTC-Fall)*, vol. 1, Orlando, FL, USA, Oct. 2003, pp. 508–512.

[22] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, May 2010.

[23] O. Simeone, "A brief introduction to machine learning for engineers," *Found. Trends Signal Process.*, vol. 12, nos. 3–4, pp. 200–431, Aug. 2018.

[24] C. Jeon, R. Ghods, A. Maleki, and C. Studer, "Optimality of large MIMO detection via approximate message passing," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Hong Kong, Jun. 2015, pp. 1227–1231.

[25] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, Mar. 2021.

[26] A. Balatsoukas-Stimming and C. Studer, "Deep unfolding for communications systems: A survey and some new directions," in *Proc. IEEE Int. Workshop Signal Process. Syst. (SiPS)*, Nanjing, China, Oct. 2019, pp. 266–271.

[27] 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.

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

[29] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," in *Proc. 54th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Monticello, IL, USA, Sep. 2016, pp. 341–346.

[30] 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.

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

[32] D. Neumann, T. Wiese, and W. Utschick, "Learning the MMSE channel estimator," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2905–2917, Jun. 2018.

[33] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "A model-driven deep learning network for MIMO detection," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Anaheim, CA, USA, Nov. 2018, pp. 584–588.

[34] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive neural signal detection for massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5635–5648, Aug. 2020.

[35] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Toulan, France, Apr. 2017, pp. 1–20.

[36] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-softmax," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Toulan, France, Apr. 2017, pp. 1–13.

[37] E. Beck, C. Bockelmann, and A. Dekorsy, "Concrete MAP detection: A machine learning inspired relaxation," in *Proc. 24th Int. ITG Workshop Smart Antennas (WSA)*, Hamburg, Germany, Feb. 2020, pp. 1–5.

[38] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The marginal value of adaptive gradient methods in machine learning," in *Proc. 30th Conf. Adv. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 4148–4158.

[39] M. Helmling *et al.* (2019). *Database of Channel Codes and ML Simulation Results*. [Online]. Available: www.uni-kl.de/channel-codes

[40] M. Cirkic, D. Persson, J. Larsson, and E. G. Larsson, "Approximating the LLR distribution for a class of soft-output MIMO detectors," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6421–6434, Dec. 2012.


**Edgar Beck** (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees (with honors) in electrical engineering from the University of Bremen, Germany, in 2014 and 2017, respectively, where he is currently pursuing the Ph.D. degree in electrical engineering with the Department of Communications Engineering (ANT). His research interests include several aspects of future 5G/6G systems: cognitive radio, compressive sensing, massive MIMO systems and in particular the fertile application of machine learning to wireless communications.

**Carsten Bockelmann** (Member, IEEE) received the Dipl.-Ing. and Ph.D. degrees in electrical engineering from the University of Bremen, Germany, in 2006 and 2012, respectively. Since 2012, he has been a Senior Research Group Leader with the University of Bremen coordinating research activities regarding the application of compressive sensing/sampling and machine learning to communication problems. His research interests include massive machine communication, ultra reliable low latency communications (5G) and industry 4.0, compressive sensing, channel coding, and transceiver design.

**Armin Dekorsy** (Senior Member, IEEE) is currently the Head of the Department of Communications Engineering, University of Bremen. He is also a Directory Board Member of the Gauss-Olbers Space Technology Transfer Center (GOC), University of Bremen. He is distinguished by more than ten years of industrial experience in leading research positions, namely DMTS at Bell Labs Europe and the Head of Research Europe Qualcomm Nuremberg, and by conducting (inter)national research projects (more than 25 BMBF/BMWI/EU projects) in affiliation with his scientific expertise shown by more than 200 journal and conference publications and more than 19 patents. He investigates new lines of research in wireless communication and signal processing for the baseband of transceivers of future communication systems, the results of which are transferred to the pre-development of industry through political and strategic activities. His current research focuses on distributed signal processing, compressed sampling, information bottleneck method, and machine learning leading to the further development of communication technologies for 5G/6G, industrial wireless communications, and NewSpace satellite communications. He is a Senior Member of the IEEE Communications and Signal Processing Society and the Head of the VDE/ITG Expert Committee “Information and System Theory”.