Neural Network Approaches for Data Estimation in Unique Word OFDM Systems

Stefan Baumgartner *Graduate Student Member, IEEE*, Gergő Bognár, Oliver Lang *Member, IEEE*, and Mario Huemer *Senior Member, IEEE*

***Abstract*—Data estimation is conducted with model-based estimation methods since the beginning of digital communications. However, motivated by the growing success of machine learning, current research focuses on replacing model-based data estimation methods by data-driven approaches, mainly neural networks (NNs). In this work, we particularly investigate the incorporation of existing model knowledge into data-driven approaches, which is expected to lead to complexity reduction and / or performance enhancement. We describe three different options, namely "model-inspired" pre-processing, choosing an NN architecture motivated by the properties of the underlying communication system, and inferring the layer structure of an NN with the help of model knowledge. Most of the current publications on NN-based data estimation deal with general multiple-input multiple-output (MIMO) communication systems. In this work, we investigate NN-based data estimation for so-called unique word orthogonal frequency division multiplexing (UW-OFDM) systems. We highlight differences between UW-OFDM systems and general MIMO systems one has to be aware of when using NNs for data estimation, and we introduce measures for a successful utilization of NN-based data estimators in UW-OFDM systems. Further, we investigate the use of NNs for data estimation when channel coded data transmission is conducted, and we present adaptions to be made, such that NN-based data estimators provide satisfying performance for this case. We compare the presented NNs concerning achieved bit error ratio performance and computational complexity, we show the peculiar distributions of their data estimates, and we also point out their downsides compared to model-based equalizers.**

***Index Terms*—Data estimation, neural networks, unique word OFDM**

I. Introduction

On the receiver side of wireless digital communication systems, data estimation, also referred to as equalization, is conducted to reconstruct the transmitted data that have been disturbed during transmission. Traditionally, this task is accomplished with model-based estimation methods. That is, the data transmission is described by physical and mathematical models, such that on basis of these models statistical estimation methods can be developed to estimate the transmitted data. This established approach has many advantages, e.g., the derived estimation methods are well-interpretable, and often performance bounds can be derived. However, there are also some downsides. Model-based estimation methods yielding optimal performance are generally computationally infeasible, which requires resorting to less complex, suboptimal methods in practice. Furthermore, modeling inaccuracies may lead to severe performance degradation, and the empirical statistical behavior of available data cannot be utilized for improving the estimation results. With data-driven machine learning methods, some of the aforementioned issues of model-based approaches can be resolved. Hence, employing data-driven methods, particularly neural networks (NNs), for equalization is a focus of current research [2]–[12]. These NNs, which are known to be universal function approximators [13], should approximate the optimal data estimator function. Ideally, the developed NN-based data estimators exhibit a low computational complexity and require a low amount of training data. However, this is a major challenge, since most of the currently known standard NNs usually have a large number of trainable parameters, leading to a large amount of required training data and a high inference complexity. One approach for tackling this problem is to design the layer architecture of an NN based on the model of the data transmission. In the context of such model-inspired NN layer structures, the concept of deep unfolding [14] is worth mentioning. There, the iterations of an iterative model-based algorithm are unfolded to layers of an NN, where free parameters or even whole parts of the model-based inference structure are replaced by trainable parameters or modules, respectively, that are optimized with the help of training data. NN-based data estimators which are deduced by deep unfolding are, e.g., DetNet [3], OAMP-Net2 [4], ViterbiNet [8], or DeepSIC [9], to name just a few. Besides deep unfolding, there are also other approaches to obtain model knowledge aided data-driven methods for data estimation. In [15], a deep learning aided sphere decoder is proposed, where the radius of the decoding hypersphere is learned. Further, in [10], an interesting NN-based equalizer, referred to as RE-MIMO, is proposed, which can be applied for equalization in multiple-input multiple-output (MIMO) systems, using an NN architecture that is designed by considering properties of MIMO systems.

An important distinction between NN-based equalizers can be made by considering their generalization ability regarding different channels. While the aforementioned DetNet, RE-MIMO, and OAMP-Net2 are trained with an ensemble of different channels, representing samples from a statistical channel

This work has been supported by the "University SAL Labs" initiative of Silicon Austria Labs (SAL) and its Austrian partner universities for applied fundamental research for electronic based systems, and the funded Project TKP2021-NVA-29. Project no. TKP2021-NVA-29 has been implemented with the support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021-NVA funding scheme. This paper was presented in part at the 55th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, USA, 2021 [1].

Stefan Baumgartner and Mario Huemer are with the JKU LIT SAL eSPML Lab, Johannes Kepler University Linz, Austria, and with the Institute of Signal Processing, Johannes Kepler University Linz, Austria (e-mails: {stefan.baumgartner, mario.huemer}@jku.at).

Gergő Bognár is with the Department of Numerical Analysis, ELTE Eötvös Loránd University, Budapest, Hungary (email: bognargergo@staff.elte.hu).

Oliver Lang is with the Institute of Signal Processing, Johannes Kepler University Linz, Austria (email: oliver.lang@jku.at).

model, and use the actual channel realization as an input, MMNet [7], ViterbiNet, and DeepSIC are trained for a single channel realization. The former NNs can be trained completely offline (assuming the channel statistics do not change), but generally feature a higher inference complexity than the latter. The latter have to be (re-)trained in an online manner, however, they usually contain fewer trainable parameters, leading to a faster training convergence and a smaller amount of required training data. In this paper, we focus on NNs that are trained offline with a multitude of different channels and use the current channel realization as an input during inference.

We consider NN-based equalization for communication systems employing the so-called unique word orthogonal frequency division multiplexing (UW-OFDM) signaling scheme [16]. UW-OFDM is an alternative to the popular cyclic prefix (CP)-OFDM, which allows, among others, achieving a better bit error ratio (BER) performance than with CP-OFDM, however, at the cost of a higher equalization complexity. That is, while a low-complex single-tap (per subcarrier) equalizer provides already optimal performance for CP-OFDM, this is not the case for UW-OFDM. More specifically, for UW-OFDM, e.g., the linear minimum mean square error (LMMSE) estimator [17] requires a full estimator matrix. The performance can be improved even further by nonlinear data estimators like a noise interpolator [18], sphere decoder [19], or decision feedback equalizers (DFEs) in different variants [20]. This motivates the investigation of NNs as an alternative to model-based equalizers in UW-OFDM systems. This work gives a comprehensive investigation of NN approaches for data estimation in UW-OFDM systems. However, many of the ideas are quite general, and can also be transferred to other digital communication waveforms. We naturally highlight the advantages of the ideas presented, but also intentionally discuss the downsides and open challenges that still need to be overcome in NN-based data estimation for future digital communication systems, which we regard as an important contribution for other researchers in this field.

Contribution: In this work, we suggest and investigate different approaches for obtaining model-inspired NN-based data estimation methods, namely, by model-aware pre-processing, by choosing NN architectures motivated by the properties of the UW-OFDM waveform, and by deducing an NN layer structure based on model knowledge. For presenting the latter approach, we choose to utilize DetNet, whose layer structure is inferred by unfolding a model-based gradient descent method. We apply DetNet, which has originally been proposed for a MIMO system, to the considered UW-OFDM systems, and make adaptions required specifically for UW-OFDM. Further, we show that also fully-connected NNs (FCNNs) can solve the problem of data estimation for non-static environments, whereby we suggest – motivated by an investigation of the nonlinear minimum mean square error (MMSE) estimator – a model-aware data pre-processing scheme applied to its input data. Investigations on the input data of the FCNN revealed that its input data exhibit correlations. We aim to exploit these correlations by choosing an appropriate NN architecture for achieving an improved equalization performance and/or lower complexity. Hence, we propose as a third NN-based equalizer a novel NN, which is – as one of the first proposed NN-based data estimators – based on the Transformer architecture [21], and utilizes self-attention.

The majority of available publications on NN-based data estimators assume a MIMO system model – often with data transmission over an uncorrelated Rayleigh fading channel. It turns out that applying DetNet for data estimation in UW-OFDM systems as it is suggested for MIMO systems, does not provide satisfactory performance. This issue can be overcome by introducing a data normalization scheme specifically for UW-OFDM and applying preconditioning. This should point out the importance of investigating the properties of a communication system even for data-driven NN approaches, since system-specific adaptions of existing state-of-the-art (SOTA) NN-based data estimators may be required. We compare the NN-based approaches with model-based methods in terms of performance and complexity.

We conduct our investigations for both channel coded and uncoded data transmission, where the former case has rarely been covered in publications on NN-based data estimators yet. For channel coded transmission, the equalizers have to provide reliability information about their estimates. It turns out, that NN-based data estimators tend to be overconfident in their decisions, which impairs the overall system performance including channel decoding. We suggest a simple yet effective measure that can be conducted to counteract the overconfidence of the NNs, which allows achieving approximately the same BER performance as with an optimal equalizer. Furthermore, we present results for imperfect channel knowledge, and we plot the empirical distributions of the estimates of model-based and NN-based equalizers, which highlights peculiarities of some of the considered equalizers.

To summarize, in our paper [1], on which this manuscript is based on, we introduced a data normalization scheme for UW-OFDM systems, we proposed a preconditioning method for DetNet, which boosts its performance, and we analyzed the complexity of this NN-based equalizer and some model-based equalizers. In this work, we extend [1] by the following main contributions:

- The proposal of a model-aware data pre-processing, which we validate by applying an FCNN as a data estimator using the pre-processed data as input data.
- A novel NN-based data estimator, referred to as Attention Detector, which utilizes the self-attention mechanism to exploit correlations in the input data for enhancing its estimation performance.
- An analysis of the impact of imperfect channel knowledge on the performance of the presented NN equalizers.
- An investigation of the performance of NN-based data estimators for channel coded data transmission with a convolutional channel code, including a suggestion for the selection of the training data to achieve approximately optimal performance.
- A detailed complexity analysis of the presented model-based and NN-based equalizers.
- An in-depth investigation of some model-based linear and nonlinear estimators, and a visualization of their decision boundaries.

The remainder of this paper is structured as follows: we start by reviewing the UW-OFDM signaling scheme in Sec. II. In Sec. III, we present optimal and suboptimal model-based equalizers. We address the NN-based equalizers, as well as the utilized data normalization scheme, in Sec. IV. In Sec. V, we provide BER performance results for both channel coded and uncoded data transmission, we conduct a complexity analysis, and we compare the distributions of the estimates provided by model-based and NN-based equalizers.

Notation: Throughout this paper, the ith element of a vector $\mathbf{x}$, the element in the ith row and the jth column of a matrix $\mathbf{X}$, and the ith row of a matrix $\mathbf{X}$ are denoted as x_i, $[\mathbf{X}]_{ij}$, and $[\mathbf{X}]_{i,*}$, respectively. The operators Re$\{.\}$ and Im$\{.\}$ deliver the real and the imaginary part of a complex-valued quantity, $(.)^T$ and $(.)^H$ indicate the transposition and the conjugate transposition of a vector/matrix, respectively. Furthermore, $p(.)$, $p[.]$, Pr$(.)$, $p[a|b]$, $p[a=\tilde{a}]$, and $E_a[.]$ describe the probability density function (PDF) of a continuous random variable, the probability mass function (PMF) of a discrete random variable, the probability operator, a conditional PMF of the random variable a given b, a PMF evaluated at the value $\tilde{a}$, and the expectation operator averaging over the PDF/PMF of a, respectively. The subscript of the expectation operator is omitted, when the averaging PDF/PMF is clear from context.

II. Unique Word OFDM System Model

In this section, we describe the basics of UW-OFDM. The UW-OFDM signaling scheme mainly exhibits two differences from CP-OFDM. Firstly, a deterministic sequence, the so-called UW, is employed as a guard interval. Secondly, the guard interval is part of a UW-OFDM time domain symbol resulting from an inverse discrete Fourier transform (IDFT) operation. That is, a guard interval is not removed on receiver side, but is transformed to frequency domain together with the preceding payload. With this approach, redundancy in frequency domain is introduced, which can be exploited beneficially for spectral shaping [22], and for achieving a better BER performance [16] than with CP-OFDM, however, at the cost of receiver complexity. For more detailed information on UW-OFDM, we refer to [16], [17], [23], [24]. In the following, we elucidate the data transmission in a UW-OFDM system and its associated system model.

As in CP-OFDM, the data symbols, drawn from a phase-shift keying (PSK) or quadrature amplitude modulation (QAM) alphabet[1] $\mathbb{S}'$, are defined in frequency domain. In contrast to a CP-OFDM symbol, a UW-OFDM symbol $\tilde{\mathbf{x}} \in \mathbb{C}^N$, containing N_d data symbols $d' \in \mathbb{S}'$, has to fulfill specific conditions. To reveal the conditions on a UW-OFDM symbol, we consider the structure and the generation of a UW-OFDM time domain symbol $\mathbf{x}_\mathrm{t} \in \mathbb{C}^N$ of length N. In a first step, a time domain symbol $\mathbf{x}$ is generated that consists of payload data $\mathbf{x}_\mathrm{pl}$, and a succeeding sequence of zeros with length N_u, i.e., $\mathbf{x} = [\mathbf{x}_\mathrm{pl}^T \quad \mathbf{0}^T]^T$. The requested structure of $\mathbf{x}$ imposes the condition $\mathbf{F}_N^{-1}\tilde{\mathbf{x}} = [\mathbf{x}_\mathrm{pl}^T \quad \mathbf{0}^T]^T$ on the corresponding UW-OFDM symbol $\tilde{\mathbf{x}}$ in frequency domain, where $\mathbf{F}_N^{-1}$ is the N-point IDFT matrix. To fulfill this constraint, the number of data symbols N_d per UW-OFDM symbol has to be at least by N_u smaller than the length N of a UW-OFDM symbol, reduced by the number of zero subcarriers N_z, i.e., $N_\mathrm{d} \leq N - N_\mathrm{z} - N_\mathrm{u}$. Throughout this paper, we consider the case[2] $N_\mathrm{d} = N - N_\mathrm{z} - N_\mathrm{u}$. The generation of a UW-ODFM symbol is described by $\tilde{\mathbf{x}} = \mathbf{B}\mathbf{G}\mathbf{d}'$, where $\mathbf{d}' \in \mathbb{S}'^{N_\mathrm{d}}$ is the data vector, $\mathbf{B} \in \{0,1\}^{N \times (N_\mathrm{d}+N_\mathrm{u})}$ models the optional insertion of zero subcarriers, and $\mathbf{G} \in \mathbb{C}^{(N_\mathrm{d}+N_\mathrm{u}) \times N_\mathrm{d}}$ is the so-called generator matrix. The generator matrix $\mathbf{G}$ can be decomposed into $\mathbf{G} = \mathbf{A}\left[\mathbf{I}^T \quad \mathbf{T}^T\right]^T$, with the $N_\mathrm{d} \times N_\mathrm{d}$ identity matrix $\mathbf{I}$, and an appropriately chosen matrix $\mathbf{T} \in \mathbb{C}^{N_\mathrm{u} \times N_\mathrm{d}}$, ensuring N_u trailing zeros in the UW-OFDM time domain symbol. The matrix $\mathbf{A} \in \mathbb{R}^{(N_\mathrm{d}+N_\mathrm{u}) \times (N_\mathrm{d}+N_\mathrm{u})}$, in turn, can be any non-singular matrix, which can be chosen according to the so-called systematic or non-systematic UW-OFDM signaling scheme. In this work, the non-systematic approach is used, where $\mathbf{A}$ is optimized for the BER performance of the linear minimum mean square error (LMMSE) data estimator as in [23]. In case $\mathbf{A}$ is chosen to be a permutation matrix placing the data symbols and the redundant values on their intended subcarrier position, the signaling scheme is termed systematic UW-OFDM. For further details on systematic and non-systematic UW-OFDM, we refer to [16], [17], [23], [24].

The last step on transmitter side is generating a transmit symbol $\mathbf{x}_\mathrm{t}$ by inserting the deterministic UW $\mathbf{x}_\mathrm{u} \in \mathbb{C}^{N_\mathrm{u}}$ at the position of the zero sequence of the UW-OFDM time domain symbol, i.e., $\mathbf{x}_\mathrm{t} = \mathbf{x} + [\mathbf{0}^T \quad \mathbf{x}_\mathrm{u}^T]^T$. After transmission of $\mathbf{x}_\mathrm{t}$ over a multipath channel and additional disturbance by additive white Gaussian noise (AWGN), the corresponding received vector is transformed to frequency domain, and the zero subcarriers are removed. The resulting downsized vector $\mathbf{y}_\mathrm{d}$ follows to

$$\mathbf{y}_\mathrm{d} = \widetilde{\mathbf{H}}\mathbf{G}\mathbf{d}' + \widetilde{\mathbf{H}}\mathbf{B}^T\tilde{\mathbf{x}}_\mathrm{u} + \mathbf{B}^T\mathbf{F}_N\mathbf{n}\,, \tag{1}$$

where the diagonal matrix $\widetilde{\mathbf{H}} \in \mathbb{C}^{(N_\mathrm{d}+N_\mathrm{u}) \times (N_\mathrm{d}+N_\mathrm{u})}$ contains the sampled channel frequency response excluding the positions of the zero subcarriers, $\mathbf{F}_N$ is the N-point discrete Fourier transform (DFT) matrix, $\tilde{\mathbf{x}}_\mathrm{u} = \mathbf{F}_N\,[\mathbf{0}^T \quad \mathbf{x}_\mathrm{u}^T]^T$ denotes the UW in frequency domain, and $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma_\mathrm{n}^2\mathbf{I})$ is circularly symmetric complex white Gaussian noise, where σ_n^2 is the variance of the AWGN in time domain.

Removing the influence of the known UW on $\mathbf{y}_\mathrm{d}$ yields the equivalent complex baseband system model

$$\mathbf{y}' = \mathbf{y}_\mathrm{d} - \widetilde{\mathbf{H}}\mathbf{B}^T\tilde{\mathbf{x}}_\mathrm{u} = \mathbf{H}'\mathbf{d}' + \mathbf{w}'\,, \tag{2}$$

with $\mathbf{H}' = \widetilde{\mathbf{H}}\mathbf{G}$ and $\mathbf{w}' \sim \mathcal{CN}(\mathbf{0}, N\sigma_\mathrm{n}^2\mathbf{I})$.

In case of channel coded data transmission, reliability information of the estimates, also referred to as soft information or soft decision estimates, has to be provided, e.g., in form of log-likelihood ratios (LLRs)

$$L_{ji} = \ln\left(\frac{\Pr(b_{ji}=1|\mathbf{y}')}{\Pr(b_{ji}=0|\mathbf{y}')}\right)\,, \tag{3}$$

with L_{ji} being the LLR of the jth bit b_{ji} of the ith data symbol, $j \in \{0,...,\log_2(|\mathbb{S}'|-1)\}$, $i \in \{0,...,N_\mathrm{d}-1\}$. The LLRs serve as input for the channel decoder. For uncoded data transmission, the data symbol estimates are sliced to the nearest symbol in the symbol alphabet, which is also termed hard decision estimation.

III. Model-Based Data Estimation

In this section, we review some traditional, model-based equalizers, that aim to estimate the data vector $\mathbf{d}'$, given the received vector $\mathbf{y}'$ and channel state information in form of the matrix $\mathbf{H}'$, using the system model (2). We start by elaborating on optimal estimators, which are, however, in general computationally infeasible. Consequently, one usually

[1] In this paper, the alphabet is assumed to be QPSK.

[2] In case of additionally employing N_p pilot subcarriers, N_d has to be further reduced by N_p. For simplicity, we omit the inclusion of pilot subcarriers in this derivation, and refer to [25] and [26] for details on pilot subcarrier inclusion.

has to resort to suboptimal estimation methods in practice. We describe two SOTA suboptimal estimators, where one is a linear and the other one is a nonlinear estimator. Further, we present the decision boundaries of the aforementioned equalizers in a toy example to visualize their differences.

A. Bit-Wise Maximum A-Posteriori Estimator

The optimal estimator in terms of the BER performance is the bit-wise maximum a-posteriori (MAP) estimator [27], yielding the bit value featuring the highest probability for a given received vector $\mathbf{y}'$ according to

$$\hat{b}_{ji} = \underset{\tilde{b}\in\{0,1\}}{\arg\max}\, p[b_{ji} = \tilde{b}|\mathbf{y}'] = \underset{\tilde{b}\in\{0,1\}}{\arg\max} \sum_{\mathbf{d}''\in\mathcal{S}_{ji}^{(\tilde{b})}} p(\mathbf{y}'|\mathbf{d}'') \,, \tag{4}$$

with $p(\mathbf{y}'|\mathbf{d}'') = \kappa \exp\left(-\frac{1}{N\sigma_\text{n}^2}||\mathbf{y}' - \mathbf{H}'\mathbf{d}''||_2^2\right)$, κ being a constant that does not affect the maximization, and $\mathcal{S}_{ji}^{(\tilde{b})} \subset \mathbb{S}'^{N_\text{d}}$ denoting the set of data vectors with the bit b_{ji} fixed to the value $\tilde{b} \in \{0,1\}$. In the second step, we assume independent and identically distributed (i.i.d.) data symbols in the data vector with a uniform prior probability.

B. Vector Maximum Likelihood Estimator

The estimated data vector $\hat{\mathbf{d}}'$ produced by the vector maximum likelihood (ML) estimator maximizes the likelihood function $p(\mathbf{y}'|\mathbf{d}'')$, $\mathbf{d}'' \in \mathbb{S}'^{N_\text{d}}$. Since we assume to have i.i.d. data symbols in the data vector, the vector ML estimator coincides with the vector MAP estimator. The vector ML estimator is given by

$$\hat{\mathbf{d}}' = \underset{\mathbf{d}''\in\mathbb{S}'^{N_\text{d}}}{\arg\max}\, p(\mathbf{y}'|\mathbf{d}'') = \underset{\mathbf{d}''\in\mathbb{S}'^{N_\text{d}}}{\arg\min}\, ||\mathbf{y}' - \mathbf{H}'\mathbf{d}''||_2^2 \,. \tag{5}$$

In literature, this estimator is often considered to be the optimal equalizer. In fact, it is optimal with respect to the error probability of the data vector estimate [27], but not with respect to the BER, which is the usual figure of merit in communications. By examining (5), a noteworthy peculiarity of the vector ML estimator can be observed, namely, this estimator does not depend on the noise variance σ_n^2, which is in contrast to the bit-wise MAP estimator (4).

C. Minimum Mean Square Error Estimator

The nonlinear MMSE estimator is, in contrast to the ML and the MAP estimators, very rarely regarded in communications literature. Especially when it comes to NN-based data estimators, which try to approximate the nonlinear MMSE estimator, we believe that a detailed consideration of the nonlinear MMSE estimator is quite meaningful.

When employing the Bayesian mean square error $E_{\mathbf{y}',\mathbf{d}'}[||\hat{\mathbf{d}}' - \mathbf{d}'||_2^2]$ as a performance measure, MMSE estimator is the optimal estimator. The MMSE estimator is obtained by computing the mean of the posterior PMF [28], i.e.,

$$\begin{aligned}\hat{\mathbf{d}}' &= E_{\mathbf{d}'|\mathbf{y}'}[\mathbf{d}'|\mathbf{y}'] = \sum_{\mathbf{d}''\in\mathbb{S}'^{N_\text{d}}} \mathbf{d}'' p[\mathbf{d}''|\mathbf{y}'] \\ &= \frac{\sum\limits_{\mathbf{d}''\in\mathbb{S}'^{N_\text{d}}} \mathbf{d}''\, \exp(-\frac{1}{N\sigma_\text{n}^2}||\mathbf{y}' - \mathbf{H}'\mathbf{d}''||_2^2)}{\sum\limits_{\mathbf{d}''\in\mathbb{S}'^{N_\text{d}}} \exp(-\frac{1}{N\sigma_\text{n}^2}||\mathbf{y}' - \mathbf{H}'\mathbf{d}''||_2^2)} \,,\end{aligned} \tag{6}$$

where again a uniform prior probability distribution of the data vectors is assumed.

Interestingly, as shown in Appendix A, for a QPSK modulation alphabet (which is employed as modulation alphabet in this paper) the hard decision estimates of the MMSE estimator coincide with those of the bit-wise MAP estimator. Hence, the MMSE estimator also serves as a benchmark for the best BER performance achievable. For higher-order modulation alphabets, e.g., 16-QAM or 64-QAM, the MMSE has to be formulated for the transmitted bit vector (instead of the complex-valued data symbol vector) for obtaining optimal BER performance.

Reliability Information for MMSE Estimates: As obvious from (3), the posterior probabilities $\text{Pr}(b_{ji} = 1|\mathbf{y}')$ and $\text{Pr}(b_{ji} = 0|\mathbf{y}')$ have to be determined to obtain the desired LLRs L_{ji}. For the employed QPSK modulation alphabet, the LLRs L_{0i} and L_{1i}, corresponding to the zeroth and the first bit of the ith data symbol d_i', respectively, can be computed on basis of the MMSE estimates $\hat{d}_i'$ with low complexity, which is presented in the following. To this end, let us consider the QPSK bit-to-symbol mapping $(b_{1i}b_{0i}) \mapsto d_i'$, where the bits b_{0i} and b_{1i} are mapped to the real part and the imaginary part of d_i', respectively. The bit values 0 and 1 are mapped to the symbol values $-\rho$ and ρ, respectively, with the energy normalization factor $\rho = 1/\sqrt{2}$. Hence, as given in (37), the real part of the ith MMSE estimate follows to

$$\begin{aligned}\text{Re}\{\hat{d}_i'\} &= E_{d_i'|\mathbf{y}'}[\text{Re}\{d_i'\}|\mathbf{y}'] \\ &= \rho\text{Pr}(b_{0i} = 1|\mathbf{y}') - \rho\text{Pr}(b_{0i} = 0|\mathbf{y}') \,.\end{aligned} \tag{7}$$

Since $\text{Pr}(b_{0i} = 0|\mathbf{y}') + \text{Pr}(b_{0i} = 1|\mathbf{y}') = 1$, (7) can be expressed as

$$\begin{aligned}\text{Re}\{\hat{d}_i'\} &= \rho(2\text{Pr}(b_{0i} = 1|\mathbf{y}') - 1) \,\text{, or as} \\ \text{Re}\{\hat{d}_i'\} &= \rho(1 - 2\text{Pr}(b_{0i} = 0|\mathbf{y}')) \,.\end{aligned} \tag{8}$$

By rearranging the two expressions in (8) with respect to the posterior probabilities, and plugging the results into the LLR definition (3) yields

$$L_{0i} = \ln\left(\frac{\rho + \text{Re}\{\hat{d}_i'\}}{\rho - \text{Re}\{\hat{d}_i\}}\right),\ L_{1i} = \ln\left(\frac{\rho + \text{Im}\{\hat{d}_i'\}}{\rho - \text{Im}\{\hat{d}_i\}}\right), \tag{9}$$

where for obtaining L_{1i} the same steps as above have to be conducted for the imaginary part of $\hat{d}_i'$.

D. Linear Minimum Mean Square Error Estimator

The aforementioned optimal equalizers all suffer from a complexity that is exponential in the length of the data vector. To obtain low-complex equalizers, one can constrain the estimator to be linear. The best linear estimator in terms of the Bayesian mean square error is the LMMSE estimator [28]

$$\hat{\mathbf{d}}' = \mathbf{E}_\text{LMMSE}\mathbf{y}' = \left(\mathbf{H}'^H\mathbf{H}' + \frac{N\sigma_\text{n}^2}{\sigma_\text{d}^2}\mathbf{I}\right)^{-1}\mathbf{H}'^H\mathbf{y}' \,, \tag{10}$$

where σ_d^2 is the variance of the data symbols, and $\mathbf{E}_\text{LMMSE}$ is the LMMSE estimator matrix.

Reliability Information for LMMSE Estimates: With LMMSE estimates $\hat{d}_i'$ at hand, the LLRs can be computed by evaluating the alternative LLR definition [29]

$$L_{ji}^{\text{LMMSE}} = \ln\left(\frac{\Pr(b_{ji}=1|\hat{d}_i')}{\Pr(b_{ji}=0|\hat{d}_i')}\right). \tag{11}$$

By assuming a Gaussian conditional distribution $p(\hat{d}_i'|d_i')$ for the LMMSE estimates (which is valid for large N_{d} following central limit theorem arguments), it can be shown [30], that (11) is equivalent to the LLR definition (3). Following the derivation described in [26], the LLRs for the zeroth and the first bit are given by

$$L_{0i} = \frac{4\text{Re}\{\hat{d}_i\}\alpha_i\rho}{\sigma_i^2} \quad \text{and} \quad L_{1i} = \frac{4\text{Im}\{\hat{d}_i\}\alpha_i\rho}{\sigma_i^2}, \tag{12}$$

respectively, where $\sigma_i^2 = \mathbf{e}_i^H(\sigma_{\text{d}}^2\bar{\mathbf{H}}_i'\bar{\mathbf{H}}_i'^H + N\sigma_{\text{n}}^2\mathbf{I})\mathbf{e}_i$, $\alpha_i = \mathbf{e}_i^H\mathbf{h}_i'$, $\mathbf{e}_i^H$ is the ith row of $\mathbf{E}_{\text{LMMSE}}$, $\mathbf{h}_i'$ denotes the ith column of $\mathbf{H}'$, and $\bar{\mathbf{H}}_i'$ is $\mathbf{H}'$ without the ith column.

E. Decision-Feedback Equalizer

A performance-complexity trade-off is provided by the decision-feedback equalizer (DFE). In this iterative method, LMMSE estimation of a single data symbol is conducted in every iteration. As a decision criterion which data symbol is estimated in the kth iteration, we use the diagonal of the LMMSE error covariance matrix $\mathbf{C}_{\text{ee},k}$, containing the error variances of the LMMSE estimates. That is, in a single iteration the data symbol corresponding to the smallest error variance is estimated, followed by updating the system model in form of removing the influence of the hard decision estimate $\lfloor\hat{d}_i\rceil$ from the received vector, and by deleting the appropriate column from the system matrix $\mathbf{H}_k'$ of the kth iteration. For further details we refer to [20].

Reliability Information for the DFE: Due to the nonlinear iterative equalization process, the best BER results for channel coded data transmission are obtained by incorporating channel decoding into the iterations of the DFE. However, in this work, we do not consider information feedback from the channel decoder to any of the regarded equalizers. As a circumvention, we utilize the LLRs of the LMMSE data symbol estimation in every iteration as reliability information of the DFE. Hence, the LLRs L_{0i} and L_{1i} corresponding to the data symbol estimate $\hat{d}_i'$ estimated in the kth iteration are computed as given in (12), whereby $\mathbf{H}'$ is replaced by $\mathbf{H}_k'$.

F. Decision Boundaries of Model-Based Equalizers

To illustrate the differences between the model-based data estimators elaborated above, we plot the decision boundaries of their hard decision estimates for a small toy example

$$\mathbf{y} = \mathbf{H}\mathbf{d} + \mathbf{w} = \begin{bmatrix} 0.9 & 0.6 \\ -0.3 & 0.5 \end{bmatrix}\begin{bmatrix} d_0 \\ d_1 \end{bmatrix} + \mathbf{w}, \tag{13}$$

where $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. The data vector $\mathbf{d}$, with $d_0, d_1 \in \{-1, 1\}$, corresponds to a block of two bits (b_1b_0), where the bit values 0 and 1 are mapped to the symbols -1 and 1, respectively. The decision boundaries are plotted for different noise power levels, i.e., for $\sigma^2 = 0.5$ and $\sigma^2 = 0.05$ in Fig. 1.

As already mentioned in Sec. III-B, the vector ML estimator, which is optimal regarding the estimation error probability of the whole data symbol vector, does not depend on the noise variance. This is visible in the identical decision boundaries of the vector ML estimator in Figs. 1b and 1f. The decision boundaries of the MMSE estimator (Figs. 1a and 1e) change with the noise variance, whereby the decision boundaries (and thus also the performance) of the MMSE estimator converge towards those of the vector ML estimator for $\sigma^2 \to 0$. That is, only for higher values of the noise variance a BER performance difference between these two equalizers might be observable. As shown in Sec. V-C, the BER performance difference between the vector ML estimator and the MMSE estimator is negligible for the considered UW-OFDM system. Clearly, the decision boundaries of the LMMSE estimator (Figs. 1c and 1g), can only be straight lines. The decision boundaries of the LMMSE estimator distinctly deviate from those of the MMSE estimator, indicating a considerable performance degradation for hard decision estimation due to the linearity constraint. With the DFE, in turn, in each iteration a symbol is estimated using a linear estimation step, leading to a smaller deviation of the decision boundaries to the optimal ones, which is visible in Figs. 1d and 1h.

IV. Neural Network Based Data Estimation

We start this section by presenting a data normalization scheme specifically designed for UW-OFDM systems, which is essential to achieve well-performing NNs. Then, we introduce three NN-based data estimation methods for UW-OFDM systems. As already mentioned, the three presented approaches for NN-based data estimation can be regarded as different options for utilizing model knowledge when conducting data estimation with NNs. In DetNet, model knowledge is incorporated by designing the structure of a layer by unfolding a model-based gradient descent method. When using an FCNN or our proposed Attention Detector as an equalizer, model knowledge is incorporated into data pre-processing, which is inspired by the MMSE estimator and sufficient statistics. The utilization of an encoder in the Attention Detector, which contains so-called self-attention layers [21], stems from the knowledge of correlated input data. The similarities and differences of the three approaches are visualized in Fig. 2. Interestingly, although the three NN-based approaches are inspired by different model-based concepts, for all NNs the quantities $\mathbf{H}^T\mathbf{H}$ and $\mathbf{H}^T\mathbf{y}$ (for their definition we refer to (14)) are used, however, at different positions of the NNs (DetNet: in all layers; FCNN and Attention Detector: at their input). This observation is also highlighted in Fig. 2.

To use existing knowledge of NN architectures and NN training methods, real-valued input data of the NNs are generated. Hence, we map the complex-valued system model (2) to an equivalent real-valued model of double dimension

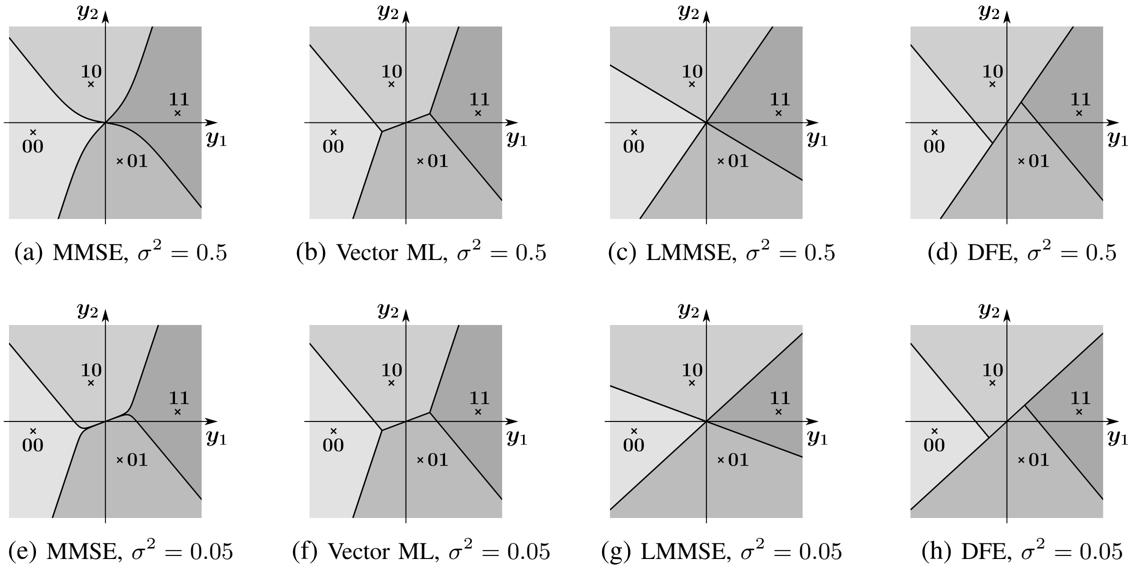
$$\mathbf{y} = \mathbf{H}\mathbf{d} + \mathbf{w}, \tag{14}$$

where

$$\mathbf{y} = \begin{bmatrix} \text{Re}\{\mathbf{y}'\} \\ \text{Im}\{\mathbf{y}'\} \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \text{Re}\{\mathbf{H}'\} & -\text{Im}\{\mathbf{H}'\} \\ \text{Im}\{\mathbf{H}'\} & \text{Re}\{\mathbf{H}'\} \end{bmatrix},$$

$$\mathbf{d} = \begin{bmatrix} \text{Re}\{\mathbf{d}'\} \\ \text{Im}\{\mathbf{d}'\} \end{bmatrix}, \text{ and } \quad \mathbf{w} = \begin{bmatrix} \text{Re}\{\mathbf{w}'\} \\ \text{Im}\{\mathbf{w}'\} \end{bmatrix} \sim \mathcal{N}\Big(\mathbf{0}, \frac{N\sigma_{\text{n}}^2}{2}\mathbf{I}\Big).$$

Assuming a symmetric alphabet $\mathbb{S}'$, $\mathbf{d} \in \mathbb{S}^{2N_{\text{d}}}$ contains data symbols d_i, $i \in \{0, ..., 2N_{\text{d}}-1\}$, drawn from the real-



(a) MMSE, $\sigma^2 = 0.5$

(b) Vector ML, $\sigma^2 = 0.5$

(c) LMMSE, $\sigma^2 = 0.5$

(d) DFE, $\sigma^2 = 0.5$

(e) MMSE, $\sigma^2 = 0.05$

(f) Vector ML, $\sigma^2 = 0.05$

(g) LMMSE, $\sigma^2 = 0.05$

(h) DFE, $\sigma^2 = 0.05$

Fig. 1. Decision boundaries of model-based equalizers for $\sigma^2 = 0.5$ and $\sigma^2 = 0.05$.

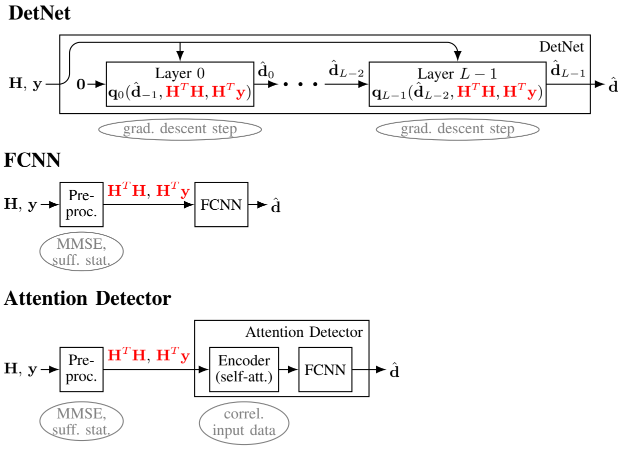


Fig. 2. High-level visualization of the three presented NN-based equalizers. Ideas from model-based methods and concepts that are used in some blocks are given in ellipses below the blocks.

valued symbol alphabet $\mathbb{S} = \mathrm{Re}\{\mathbb{S}'\} = \mathrm{Im}\{\mathbb{S}'\}$. The NN-based data estimators are, however, not trained to directly estimate the data symbols d_i, but to estimate the corresponding so-called one-hot vectors $\mathbf{d}_{\mathrm{oh},i} \in \{0,1\}^{|\mathbb{S}|}$. Let $s_j \in \mathbb{S}$, $j \in \{0, ..., |\mathbb{S}|-1\}$ be the uniquely numbered symbols of the symbol alphabet $\mathbb{S}$. Then, a one-hot vector $\mathbf{d}_{\mathrm{oh},i}$ corresponding to a data symbol d_i that exhibits the value s_j contains all zeros but a one at the jth position. The one-hot vectors $\mathbf{d}_{\mathrm{oh},i}$ are stacked to a vector $\mathbf{d}_{\mathrm{oh}}$, serving as ground truth for training the NNs. Further, a quadratic loss function $\ell(\mathbf{d}_{\mathrm{oh}}, \hat{\mathbf{d}}_{\mathrm{oh}})$ is employed to quantify the error between the output $\hat{\mathbf{d}}_{\mathrm{oh}} \in \mathbb{R}^{2N_{\mathrm{d}}|\mathbb{S}|}$ of an NN and $\mathbf{d}_{\mathrm{oh}}$. It can be shown (cf., e.g., [3]), that with this approach the estimates $\hat{\mathbf{d}}_{\mathrm{oh},i}$ of a properly trained NN approximately contain the posterior probabilities $\Pr(d_i = s_j|\mathbf{y})$, i.e., $\hat{\mathbf{d}}_{\mathrm{oh},i} \approx [\Pr(d_i = s_0|\mathbf{y}), \ ..., \ \Pr(d_i = s_{|\mathbb{S}|-1}|\mathbf{y})]^T$. With the approximate posterior probabilities, LLRs can be computed using (3). Hence, soft information of the data symbol estimates required for coded data transmission is available. A hard decision estimate, in turn, is the symbol corresponding to the maximum entry in $\hat{\mathbf{d}}_{\mathrm{oh},i}$.

A. Data Normalization

Proper normalization of the input data of an NN is generally very important for well-behaved training, and thus the performance of trained NNs [31], [32]. Interestingly, in the majority of currently available publications on NNs for data estimation in MIMO systems (e.g., [3]–[5], [10]), the input data of the NNs are not normalized. As we show in Sec. V-C, applying DetNet [3] as a data estimator in a UW-OFDM system without any data normalization (which is done in [3] for general MIMO systems) leads to poor BER performance. A major reason for this issue can be found by investigating the relation between the noise variance σ_{n}^2 and the signal-to-noise ratio (SNR) on receiver side. The performance of an equalizer is typically determined by evaluating the achieved BER at a specified E_{b}/N_0, which is a measure for the SNR, where E_b is the mean energy per bit, and N_0 is the noise power spectral density. For the following considerations, we define an SNR measure

$$\gamma = \frac{E_{\mathbf{d}}[||\mathbf{Hd}||_2^2]}{E_{\mathbf{w}}[||\mathbf{w}||_2^2]} = \frac{\sigma_{\mathrm{d}}^2 \mathrm{tr}(\mathbf{H}^T\mathbf{H})}{N^2\sigma_{\mathrm{n}}^2}, \tag{15}$$

which is proportional to E_{b}/N_0. For a specified SNR γ at the input of the equalizer, the noise variance in time domain σ_{n}^2 can therefore be expressed as

$$\sigma_{\mathrm{n}}^2 = \frac{\frac{1}{N}\sigma_{\mathrm{d}}^2 \mathrm{tr}(\mathbf{H}^T\mathbf{H})}{N\gamma}. \tag{16}$$

In case of a general MIMO system over an uncorrelated Rayleigh fading channel, which is mainly used in, e.g., [3]–[5], [10] for the performance comparison of different NN-based data estimators, the elements of $\mathbf{H} \in \mathbb{R}^{2N \times 2N_{\mathrm{d}}}$ are drawn independently from a standard normal distribution, i.e., $[\mathbf{H}]_{lm} \sim \mathcal{N}(0,1)$, $l \in \{0, \ ..., \ 2N-1\}$, $m \in \{0, \ ..., \ 2N_{\mathrm{d}}-1\}$. This leads to $E_{\mathbf{H}}\left[[\mathbf{H}^T\mathbf{H}]\right] = 2N\mathbf{I}$. Due to central limit theorem arguments, for large N, $\mathrm{tr}(\mathbf{H}^T\mathbf{H})$ can be approximated as $\mathrm{tr}(\mathbf{H}^T\mathbf{H}) \approx 2N_{\mathrm{d}}E_{\mathbf{H}}\left[[\mathbf{H}^T\mathbf{H}]_{ll}\right] = 4N_{\mathrm{d}}N$. Plugging this approximation into (16) results in

$$\sigma_{\mathrm{n}}^2 \approx \frac{4\sigma_{\mathrm{d}}^2 N_{\mathrm{d}}}{N\gamma}. \tag{17}$$

This article has been accepted for publication in IEEE Transactions on Vehicular Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVT.2023.3325367

That is, for a general MIMO system over an uncorrelated Rayleigh fading channel, the noise variances var$(w_l) = N\sigma_{\text{n}}^2$ of the elements w_l of the noise vector $\mathbf{w}$ are independent of the current channel realization, and for a fixed SNR, they are constant. Hence, the data is implicitly normalized for this communication system. This is not the case for UW-OFDM systems, and thus we normalize the data such that the variances of the elements of the noise vector become independent of the channel realization. The data normalization is conducted by multiplying the real-valued system model (14) by the normalization factor $\sqrt{N}/||\mathbf{H}||_F$, with $||\mathbf{H}||_F = \sqrt{\text{tr}(\mathbf{H}^T\mathbf{H})}$ denoting the Frobenius norm of $\mathbf{H}$. Consequently, every element of the noise vector after normalization has a variance $\text{var}\big((\sqrt{N}w_l)/||\mathbf{H}||_F\big) = (N^2\sigma_{\text{n}}^2)/(2||\mathbf{H}||_F^2) = \sigma_{\text{d}}^2/(2\gamma)$, which is independent of the channel realization. This data normalization is implemented by multiplying both $\mathbf{y}$ and $\mathbf{H}$ by the above-given normalization factor, which is conducted as a pre-processing step for all the NN-based data estimators presented subsequently. In the remainder of this paper, we omit the normalization factor for the sake of better readability.

B. DetNet

DetNet is proposed in [3] for data estimation in MIMO systems. Its network architecture is deduced by deep unfolding [14] a projected gradient descent method applied to the optimization problem of the vector ML estimator for the model (14). The kth step of the iterative optimization method can be expressed as

$$\begin{aligned}\hat{\mathbf{d}}_k &= \Pi\Bigg(\hat{\mathbf{d}}_{k-1} - \delta_k \frac{\partial||\mathbf{y}-\mathbf{H}\mathbf{d}||_2^2}{\partial\mathbf{d}}\bigg|_{\mathbf{d}=\hat{\mathbf{d}}_{k-1}}\Bigg) \\ &= \Pi\big(\hat{\mathbf{d}}_{k-1} + 2\delta_k\mathbf{H}^T\mathbf{y} - 2\delta_k\mathbf{H}^T\mathbf{H}\hat{\mathbf{d}}_{k-1}\big)\,, \qquad (18)\end{aligned}$$

where $\Pi(.)$ denotes a nonlinear projection to a convex subspace $\mathcal{D}$ containing all possible data vectors $\mathbf{d}$, i.e., $\mathbb{S}^{2N_{\text{d}}} \subset \mathcal{D} \subset \mathbb{R}^{2N_{\text{d}}}$, and δ_k is the step width in the kth iteration.

The structure of the kth layer of the L DetNet layers is inspired by a projected gradient descent iteration step (18). Firstly, the affine mapping

$$\mathbf{q}_k = \hat{\mathbf{d}}_{k-1} + \delta_{k1}\mathbf{H}^T\mathbf{y} - \delta_{k2}\mathbf{H}^T\mathbf{H}\hat{\mathbf{d}}_{k-1} \qquad (19)$$

is applied to the layer input $\hat{\mathbf{d}}_{k-1}$ to obtain the temporal variable $\mathbf{q}_k$, where δ_{k1} and δ_{k2} are learned parameters. Secondly, the temporal variable is forwarded to a fully-connected neural network (FCNN) with a single hidden layer consisting of d_{h} hidden neurons and ReLU activation, which replaces the (unknown) nonlinear projection $\Pi(.)$. To ease the training of DetNet, weighted residual connections [33] with weighting factor α, as well as an auxiliary loss inspired by the loss function employed for the training of GoogLeNet [34] are utilized. Further, d_{v}-dimensional auxiliary variables $\mathbf{v}_k$ passing unconstrained information from layer to layer are used to improve the performance of DetNet. We refer to [3] for more detailed information.

Preconditioning: Due to the deduction of the layer structure of DetNet by deep unfolding, the number of layers corresponds to the number of required iterations of the underlying projected gradient descent method. It is well known, that the condition number of the Hessian matrix in an optimization problem influences the number of iterations required for an iterative optimization method to converge. Hence, preconditioning the system model (14) may reduce the number of required DetNet layers and thus the number of trainable parameters, which, in turn, enhances both the training behavior and the inference complexity. As also stated in [7], we have observed [1] that for ill-conditioned channel matrices NN-based equalizers suffer from severe performance degradation. We showed in [1] that preconditioning distinctly narrows the eigenvalue spectrum of the Hessian matrix $\mathbf{S} \in \mathbb{R}^{P\times P}$, $[\mathbf{S}]_{rs} = \frac{\partial \ell(\mathbf{d}_{\text{oh}},\hat{\mathbf{d}}_{\text{oh}})}{\partial p_r\, \partial p_s}$ of the NN learning problem, where p_r and p_s are two of the P trainable parameters of the NN. This, in turn, allows using higher learning rates, which leads to a faster and probably better optimization of the NN parameters. We show the influence of preconditioning on the DetNet performance in Sec. V-C.

In the following, we show that preconditioning does only add a further processing step of the layer input data, while the layer structure of DetNet remains unchanged. To this end, let us rewrite the optimization problem of the vector ML estimator in form of

$$\min_{\mathbf{d}\in\mathbb{S}^{2N_{\text{d}}}} ||\mathbf{y} - \mathbf{H}\mathbf{L}^{-1}\mathbf{L}\mathbf{d}||_2^2\,, \qquad (20)$$

where $\mathbf{L} \in \mathbb{R}^{2N_{\text{d}}\times 2N_{\text{d}}}$ is an invertible matrix. Neglecting temporarily the projection operator, a gradient descent step for the linearly transformed vector $\mathbf{d}_{\text{pr}} = \mathbf{L}\mathbf{d}$ is given by

$$\begin{aligned}\hat{\mathbf{d}}_{\text{pr},k} &= \hat{\mathbf{d}}_{\text{pr},k-1} - \delta_k \frac{\partial||\mathbf{y}-\mathbf{H}\mathbf{L}^{-1}\mathbf{d}_{\text{pr}}||_2^2}{\partial\mathbf{d}_{\text{pr}}}\bigg|_{\mathbf{d}_{\text{pr}}=\hat{\mathbf{d}}_{\text{pr},k-1}} \\ &= \hat{\mathbf{d}}_{\text{pr},k-1} + 2\delta_k\mathbf{L}^{-T}\mathbf{H}^T\left(\mathbf{y} - \mathbf{H}\mathbf{L}^{-1}\hat{\mathbf{d}}_{\text{pr},k-1}\right)\,, \qquad (21)\end{aligned}$$

with $\hat{\mathbf{d}}_{\text{pr},k/k-1} = \mathbf{L}\hat{\mathbf{d}}_{k/k-1}$, and $\mathbf{L}^{-T} = \big(\mathbf{L}^{-1}\big)^T = \big(\mathbf{L}^T\big)^{-1}$. Hence, the kth iteration of the projected gradient descent for $\mathbf{d}$ follows to

$$\begin{aligned}\hat{\mathbf{d}}_k &= \Pi\big(\mathbf{L}^{-1}\hat{\mathbf{d}}_{\text{pr},k}\big) \\ &= \Pi\big(\hat{\mathbf{d}}_{k-1} + 2\delta_k\mathbf{P}^{-1}\mathbf{H}^T\mathbf{y} - 2\delta_k\mathbf{P}^{-1}\mathbf{H}^T\mathbf{H}\hat{\mathbf{d}}_{k-1}\big)\,, \qquad (22)\end{aligned}$$

where $\mathbf{P} = \mathbf{L}^T\mathbf{L}$ is the so-called preconditioning matrix. In this paper, we utilize a Jacobi preconditioning matrix, which is a diagonal matrix containing diag$(\mathbf{H}^T\mathbf{H})$ on its main diagonal. Hence, the computation of $\mathbf{P}^{-1}$, $\mathbf{P}^{-1}\mathbf{H}^T\mathbf{y}$, and $\mathbf{P}^{-1}\mathbf{H}^T\mathbf{H}$ can be carried out with low complexity. A comparison of a projected gradient descent step (18) and its preconditioned version (22) reveals that preconditioning does not change the structure of the equation. Hence, the layer architecture of DetNet remains unchanged, while $\mathbf{H}^T\mathbf{y}$ and $\mathbf{H}^T\mathbf{H}$ have to be replaced by $\mathbf{P}^{-1}\mathbf{H}^T\mathbf{y}$ and $\mathbf{P}^{-1}\mathbf{H}^T\mathbf{H}$, respectively.

C. Fully-Connected Neural Network

According to the universal approximation theorem [13], an FCNN with a single hidden layer and sufficiently many hidden neurons can approximate any function, and thus should also be able to accomplish the task of data estimation. However, as stated in [3], it is challenging to employ an FCNN for equalization under changing channel realizations when using the columns of $\mathbf{H}$ concatenated with $\mathbf{y}$ as input data. That is, training an FCNN for different channels might be a hard task. One reason for this issue might be that no model knowledge is included in the structure of an FCNN. We therefore suggest to include model knowledge in data pre-processing.

To motivate the choice of the proposed data pre-processing with the purpose of reducing redundant information, we elucidate three observations. Firstly, the FCNN should approximate the estimator function of the optimal MMSE estimator (6). An inspection of (6) reveals, that an MMSE estimate is a sum of exponential terms, where the exponents contain[3]

$$||\mathbf{y}-\mathbf{H}\mathbf{d}||_2^2=\mathbf{y}^T\mathbf{y}-2\mathbf{d}^T\mathbf{H}^T\mathbf{y}+\mathbf{d}^T\mathbf{H}^T\mathbf{H}\mathbf{d},\quad \forall\mathbf{d}\in\mathbb{S}^{2N_\mathrm{d}}.$$

That is, the MMSE estimator does not use the isolated data $\mathbf{y}$ and $\mathbf{H}$, but the terms $\mathbf{y}^T\mathbf{y}$, $\mathbf{H}^T\mathbf{y}$, and $\mathbf{H}^T\mathbf{H}$. Secondly, it can be shown with the help of the Fisher-Neyman factorization theorem [35] that $\mathbf{H}^T\mathbf{y}$ provides a sufficient statistic for the data estimation problem. Consequently, multiplying $\mathbf{y}$ by $\mathbf{H}^T$, which modifies the system model to

$$\mathbf{H}^T\mathbf{y}=\mathbf{H}^T\mathbf{H}\mathbf{d}+\mathbf{H}^T\mathbf{w}\,, \tag{23}$$

preserves all the relevant information contained in $\mathbf{y}$ for the estimation of $\mathbf{d}$, while reducing the dimension of the available data. Thirdly, the matched filter equalizer for the system model (14) is given by $\hat{\mathbf{d}}_\mathrm{MF}=\mathbf{H}^T\mathbf{y}$, which is the linear filter designed for maximizing the output SNR [36].

With the above-given arguments, we conclude that multiplying both $\mathbf{H}$ and $\mathbf{y}$ by $\mathbf{H}^T$ before using them as inputs of an FCNN compresses the input data while preserving all the information required for data estimation. Interestingly, also for DetNet the quantities $\mathbf{H}^T\mathbf{y}$ and $\mathbf{H}^T\mathbf{H}$ are utilized instead of $\mathbf{y}$ and $\mathbf{H}$, however, due to a different motivation, and in a different manner. Since $\mathbf{H}^T\mathbf{H}$ is a symmetric matrix, the dimension of the input data is further reduced by utilizing only the upper triangular matrix of $\mathbf{H}^T\mathbf{H}$ including its main diagonal. That is, the input vector of the FCNN data estimator is

$$\big[[\mathbf{H}^T\mathbf{H}]_{00},[\mathbf{H}^T\mathbf{H}]_{0:1,1}^T,\,[\mathbf{H}^T\mathbf{H}]_{0:2,2}^T,\,\cdots,\\ [\mathbf{H}^T\mathbf{H}]_{0:2N_\mathrm{d}-1,2N_\mathrm{d}-1}^T,\,(\mathbf{H}^T\mathbf{y})^T\big]^T,$$

where $[\mathbf{H}^T\mathbf{H}]_{0:l,l}$ denotes the vector containing the first $l+1$ entries of the lth column of $\mathbf{H}^T\mathbf{H}$.

The utilized FCNNs for equalization are comprised of L layers, d_h neurons per hidden layer, and weighted residual connections with weighting factor α. The employed activation functions $\varphi(.)$ are stated in Tab. I.

D. Attention Detector

Due to the arguments given in Sec. IV-C, we use the compressed system model (23) for defining the inputs of the so-called Attention Detector. Investigations on the compressed system model (23) revealed that the entries in $\mathbf{H}^T\mathbf{y}$ are correlated. This observation motivates the use of an NN architecture that exploits these correlations for enhancing the estimation performance and/or reducing the required computational complexity. In order to exploit long-range correlations in large-scale communication systems as well, we decide to utilize the self-attention mechanism [21] instead of convolutional layers (which only capture local dependencies) for the NN-based data estimator. The architecture of the Attention Detector is inspired by the Vision Transformer [37]. The Vision Transformer solely relies on the self-attention mechanism and does not employ convolutional layers for capturing spatial correlations. Although Transformer architectures do not exhibit some of the inductive biases of convolutional NNs (CNNs), for image classification tasks the Vision Transformer shows similar performance as SOTA CNNs, especially when being pretrained on large amounts of data. For further elaborations on the network architecture of the Attention Detector, let us start by defining its inputs, which are the rows $\mathbf{m}_k^T$, $k\in\{0,\ ...,\ 2N_\mathrm{d}\ -\ 1\ \}$, of the matrix

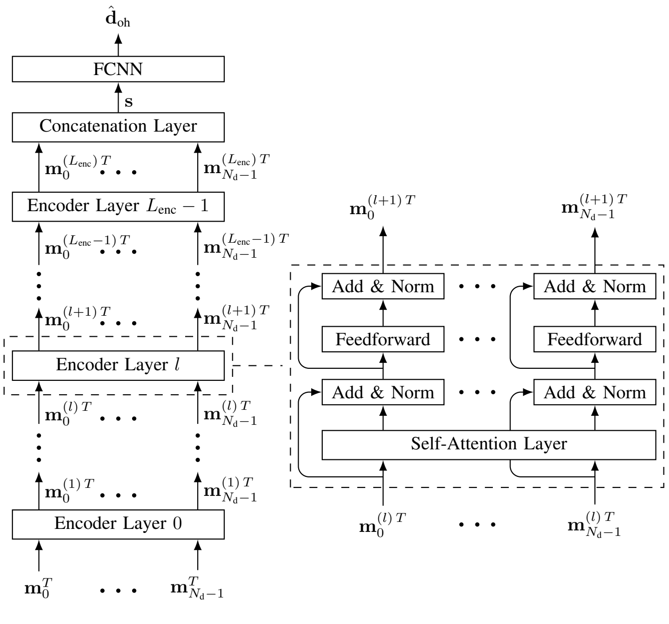

$\hat{\mathbf{d}}_\mathrm{oh}$
FCNN
$\mathbf{s}$
Concatenation Layer
$\mathbf{m}_0^{(L_\mathrm{enc})\,T}$ ⋯ $\mathbf{m}_{N_\mathrm{d}-1}^{(L_\mathrm{enc})\,T}$
Encoder Layer $L_\mathrm{enc}-1$
$\mathbf{m}_0^{(L_\mathrm{enc}-1)\,T}$ ⋯ $\mathbf{m}_{N_\mathrm{d}-1}^{(L_\mathrm{enc}-1)\,T}$
$\mathbf{m}_0^{(l+1)\,T}$ ⋯ $\mathbf{m}_{N_\mathrm{d}-1}^{(l+1)\,T}$
Encoder Layer l
$\mathbf{m}_0^{(l)\,T}$ ⋯ $\mathbf{m}_{N_\mathrm{d}-1}^{(l)\,T}$
$\mathbf{m}_0^{(1)\,T}$ ⋯ $\mathbf{m}_{N_\mathrm{d}-1}^{(1)\,T}$
Encoder Layer 0
$\mathbf{m}_0^{T}$ ⋯ $\mathbf{m}_{N_\mathrm{d}-1}^{T}$
$\mathbf{m}_0^{(l+1)\,T}$ $\mathbf{m}_{N_\mathrm{d}-1}^{(l+1)\,T}$
Add & Norm ⋯ Add & Norm
Feedforward ⋯ Feedforward
Add & Norm ⋯ Add & Norm
Self-Attention Layer
$\mathbf{m}_0^{(l)\,T}$ ⋯ $\mathbf{m}_{N_\mathrm{d}-1}^{(l)\,T}$

Fig. 3. Structure of the Attention Detector (left) and one of its encoder layers (right).

$$\mathbf{M}=\mathbf{P}^{-1}\big[\mathbf{H}^T\mathbf{y},\,\mathbf{H}^T\mathbf{H}\big]\,, \tag{24}$$

where $\mathbf{P}$ is the Jacobi preconditioning matrix as described in Sec. IV-B. Although the layer architecture of the Attention Detector is not deduced by deep unfolding, we apply preconditioning for obtaining a narrower eigenvalue spectrum of the Hessian matrix of the NN learning problem, cf. Sec. IV-B. The vectors $\mathbf{m}_k^T$ serve as an input sequence of an encoder. Since the rows of the equation system (23) are interchangeable, no positional encoding is applied to the vectors. The encoder is very similar to that of the Transformer [21]. It is comprised of L_enc stacked encoder layers, whereby the lth encoder layer, $l\ \in\ \{0,...,L_\mathrm{enc}-1\}$, is schematically shown in Fig. 3. An encoder layer with inputs[4] $\mathbf{m}_k^{(l)\,T}$ consists of a self-attention layer [21], followed by a batch norm layer, a single hidden layer FCNN with $d_\mathrm{h,enc}$ hidden neurons and ReLU activation function, and another batch norm layer. Around both the self-attention layer and the single hidden layer FCNN residual connections are employed. Further, dropout [38] with a dropout rate D is applied to the outputs of the self-attention layer and the single hidden layer FCNN, as well as to the input layer outputs of the latter. The outputs of the last encoder layer $\mathbf{m}_k^{(L_\mathrm{enc})\,T}$ are concatenated to the input vector $\mathbf{s}=[\mathbf{m}_0^{(L_\mathrm{enc})\,T},\,\cdots,\,\mathbf{m}_{N_\mathrm{d}-1}^{(L_\mathrm{enc})\,T}]$ of a shallow FCNN with L_fcnn hidden layers, $d_\mathrm{h,fcnn}$ neurons per hidden layer, and an

[3] Here, we express the exponent with the real-valued quantities $\mathbf{y}$, $\mathbf{d}$, and $\mathbf{H}$ instead of using the complex-valued $\mathbf{y}'$, $\mathbf{d}'$, and $\mathbf{H}'$ as in (6).

[4] Note that $\mathbf{m}_k^{(0)\,T}=\mathbf{m}_k^T$.

TABLE I
HYPERPARAMETER SETTINGS.

	DetNet					FCNN					Attention Detector						
	η	L	d_{h}	d_{v}	α	η	L	d_{h}	α	$\varphi(.)$	η	L_{enc}	$d_{\text{h,enc}}$	L_{fcnn}	$d_{\text{h,fcnn}}$	D	$\varphi(.)$
System I	$1.9{\cdot}10^{-3}$	10	80	32	0.1	$6.0{\cdot}10^{-4}$	10	300	0.0	ReLU	$1.8{\cdot}10^{-3}$	8	80	2	150	0.0	ReLU
System II	$4.6{\cdot}10^{-4}$	30	250	80	0.9	$1.0{\cdot}10^{-4}$	22	800	0.7	SeLU	$3.0{\cdot}10^{-4}$	10	400	3	500	0.0	SeLU

activation function $\varphi(.)$ specified in Tab. I. The outputs of this shallow FCNN are the final estimation results $\hat{\mathbf{d}}_{\text{oh}}$.

V. RESULTS

In this section, we compare the proposed NN-based data estimators for UW-OFDM with SOTA model-based equalizers in terms of the achieved BER performance over a specified SNR range by means of simulations, and provide an in-depth comparison of the computational complexity of the presented NNs and the model-based methods. For UW-OFDM, no other NN-based equalizers have been presented yet, and thus we regard the model-based LMMSE estimator, the DFE, and – for simulation setups where it is computationally feasible – the MMSE estimator as the SOTA equalizers to be used for comparison. However, we also show the BER performance results of the recently published NN-based data estimators OAMP-Net2 [4] and RE-MIMO [10], which have been proposed for equalization in MIMO systems, for one of our simulation scenarios. OAMP-Net2 is deduced by deep unfolding the OAMP algorithm [39] and replacing a few scalar parameters of the model-based method (four per iteration) by learnable parameters. RE-MIMO, in turn, implements a recurrent learning scheme to conduct iterative symbol estimation. This NN-based equalizer consists of three modules, where the NN architecture of each module is chosen due to model-based considerations. One of the three modules is very similar to the encoder of the Transformer [21], which allows to capture dependencies between a varying number of transmitters (and independent of their ordering in the system model) in a MIMO system. We study the model-based and NN-based equalizers presented in Sec. III and IV, respectively, in different simulation scenarios for channel coded and uncoded data transmission, and we detail how to counteract overconfidence of NN-based data estimators to obtain reliable soft information required for channel coded data transmission. Further, we investigate the influence of imperfect channel knowledge on the BER performance of NN-based equalizers and model-based estimators, and we highlight the peculiar distribution of the estimates provided by NN-based data estimators. Due to the multitude of possible combinations of system settings, only selected simulation cases are presented, while those setups that do not provide further insights are omitted. Since with non-systematic UW-OFDM signaling a better BER performance is achievable [23], we focus on this signaling scheme in our investigations.

A. Simulation Setup

The evaluation is conducted for two different system dimensions. The parameter setup for system I is $N = 12$, $N_{\text{d}} = 8$, $N_{\text{u}} = 4$, $N_{\text{z}} = 0$, and $N_{\text{p}} = 0$, and for system II $N = 64$, $N_{\text{d}} = 32$, $N_{\text{u}} = 16$, $N_{\text{z}} = 12$, and $N_{\text{p}} = 4$. System II should represent a real-world communication system, where N_{p} pilot subcarriers can be utilized for synchronization purposes. However, in our simulations, the pilot subcarriers are unused and do not influence the presented results. Since the computational complexity of the optimal equalizers grows exponentially with the data vector length N_{d}, simulating their BER performance for system II is computationally infeasible. Hence, we also introduced the downsized system I, for which the BER performance of the optimal model-based data estimators can be simulated in a reasonable time, providing insights concerning the gap between the performance achieved with an NN-based equalizer and the lower BER bound.

We assume data transmission over a multipath channel in form of data bursts comprised of a sequence of 1000 UW-OFDM symbols. The channel is assumed to be stationary for a single data burst, but to be changing independently of all other channel realizations from burst to burst. We utilize the statistical channel model [40] of an indoor frequency selective environment, where the channel impulse responses are modeled in form of tapped delay lines. The complex tap values exhibit a uniformly distributed phase and a Rayleigh distributed magnitude with an exponentially decaying power profile. As in the referenced works on UW-OFDM [16], [23], [26], we use for system II a sampling time $T_{\text{s}} = 50\,\text{ns}$, and we choose a channel delay spread of $\tau_{\text{RMS}} = 100\,\text{ns}$. For system I, we specify the sampling time to be $200\,\text{ns}$ while keeping the same channel delay spread as for system II. For all results apart from those in Sec. V-D we assume perfect channel knowledge on receiver side. The presented BER curves are obtained by averaging over 8000 channels.

For channel coded data transmission, a convolutional code with generator polynomials $(133, 171)_8$, constraint length 7, and rate $R = 1/2$ is used, whereby a Viterbi channel decoder is employed. As already mentioned, the data symbols are drawn from a QPSK modulation alphabet.

B. Neural Network Training

The dataset for training the NNs is obtained by simulating sample data transmissions with known payload data over randomly generated multipath channels following the employed channel model described in Sec. V-A. Since data estimation is most challenging for transmissions over deep fading channels, we emphasize those cases by adding a set of sample data transmissions to the training set that solely contains transmissions over deep fading channels. The channels for this subset of the training set are found by creating 5000 times more channels than needed and picking the channels with the most severe fading holes. Including particularly bad channels in the training set turns out to be beneficial for the BER performance of the NN-based data estimators (a similar observation has also been mentioned in [41]). Empirical investigations show that the proportion of the subset of specifically generated bad channels in the training set of $10\,\%$ and $50\,\%$ is a good choice for system I and system II, respectively. Overall, the training

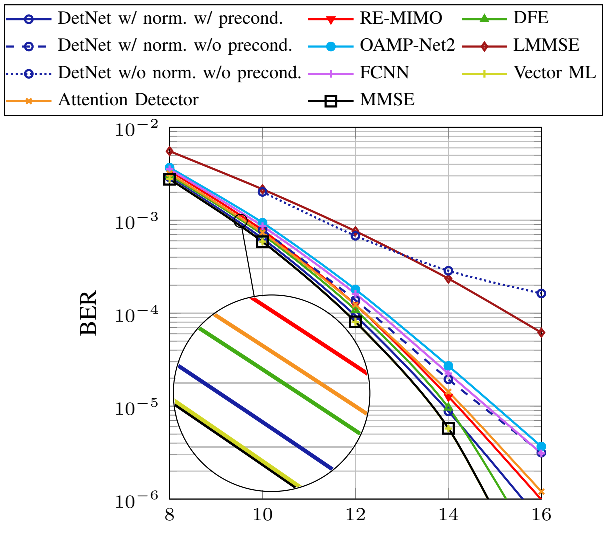


Fig. 4. BER performance comparison for system I, uncoded case.

set consists of 30000 channels and 40000 channels for system I and system II, respectively. The selection of the E_b/N_0 values for the sample data transmissions, which turns out to have a major impact on the performance of the NNs, differs for the simulated system setups, and thus is given with the results for the chosen system setup. Furthermore, we pre-trained the NNs with noiseless data transmissions, i.e., the sent data is only disturbed by a multipath channel, over 2000 different channels, which leads to a faster training convergence.

For the training, we employ an Adam optimizer [42] with default settings. The learning rate is decreased exponentially, such that the learning rate in the final optimization step is $5\,\%$ of the initial learning rate η. All NNs are trained with a batch size of 1024 and for 60 epochs. Further, early stopping is utilized as a regularization technique. The hyperparameters of the NN-based equalizers are found with an extensive grid search by evaluating the trained NNs on a validation set; the best settings found are summarized in Tab. I.

C. Bit Error Ratio Performance – Uncoded Transmission

We start the performance comparison of the NN-based equalizers by highlighting the importance of data preprocessing. Without data normalization, the NNs exhibit even worse performance than the LMMSE estimator, which is exemplarily shown for DetNet in Fig. 4 (dotted line). Utilizing the normalized data leads to a major performance improvement (dashed line in Fig. 4). For DetNet, the BER performance can be further boosted by employing preconditioning, such that with this NN close to optimal MMSE performance can be achieved. The FCNN performs approximately equivalently to the DetNet without preconditioning, while the Attention Detector can outperform the FCNN, which confirms the idea of exploiting correlations for enhancing the estimation performance by utilizing the self-attention mechanism. It turns out, that the SNR utilized for the sample transmission contained in the training set has a large influence on the performance of the NN-based data estimators. Training at too low SNRs leads to flattening out BER curves of the NN-based data estimators at higher SNRs. Training solely at higher SNRs, in turn, impairs the overall performance of the NNs, which probably comes from too few data samples located around the optimal decision boundaries (these samples are very important for the NNs to learn good decision boundaries). Hence, the E_b/N_0 training range is another hyperparameter for the NN-based data estimators, whereby the E_b/N_0 values for the data burst transmissions contained in the training set are chosen randomly, with uniform distribution on a linear scale within the specified range. For system I, all NNs are trained in the E_b/N_0 range $[9\,\text{dB}, 18\,\text{dB}]$. To get an idea how other SOTA NN-based equalizers (however, originally not being developed for UW-OFDM systems) perform, for this evaluation setup we further compare our proposed NN-based data estimators with the SOTA NN-based equalizers OAMP-Net2 [4] and RE-MIMO [10], which have been presented for data estimation in MIMO systems. Both NNs are trained with the same training set (with normalized data) as the aforementioned NN-based equalizers. For OAMP-Net2, the best hyperparameters found are (using the original notation from [4]) $T = 8$ layers and a learning rate $\eta = 10^{-3}$. For RE-MIMO (using the notation from [10]), the best hyperparameter setting found is $T = 10$ layers, $d_s = 102$ (dimensionality of the state variable), $n = 4$ parallel attention heads, $d_{TE} = 24$ (dimensionality of the transmitter encoding vector), and a learning rate $\eta = 10^{-4}$. As shown in Fig. 4, the OAMP-Net2 is the worst performing NN, which could be due to the fact that this NN is designed for unitarily-invariant system matrices $\mathbf{H}$, a condition that is not satisfied in UW-OFDM systems. The RE-MIMO, which also utilizes the self-attention mechanism, performs approximately the same as the Attention Detector.

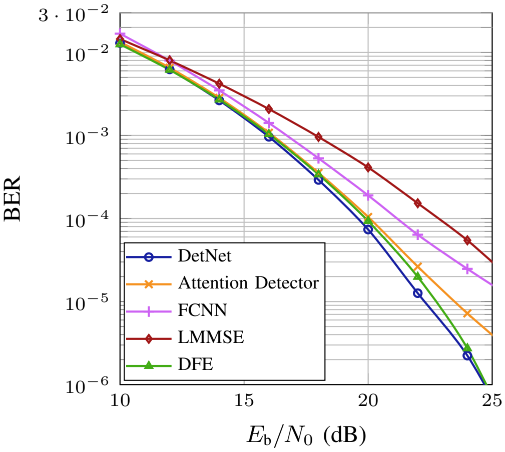


Fig. 5. BER performance comparison for system II, uncoded case.

Regarding the model-based equalizers, we observe a large performance gap between the LMMSE estimator and other estimators. With the DFE, a performance close to the optimal MMSE performance[5] can be achieved, while the BER performance difference between the vector ML estimator and the MMSE estimator is negligible for the considered system.

As illustrated in Fig. 5, for system II, DetNet can slightly outperform the DFE. Similar as for system I, the Attention Detector exhibits a small performance gap compared to the DetNet, while it outperforms the FCNN. All NNs considered clearly outperform the LMMSE baseline performance.

[5] As stated in Sec. III-C, for the regarded setup the hard decision estimates of the MMSE estimator coincide with those of the bit-wise MAP estimator.

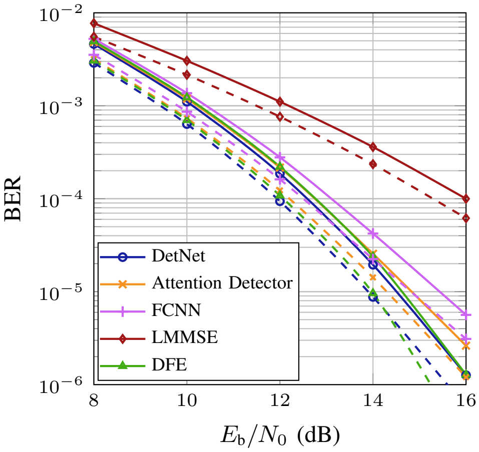


Fig. 6. BER performance comparison for system I, uncoded transmission, imperfect channel knowledge; solid lines: imperfect channel knowledge, dashed lines: perfect channel knowledge.

While for DetNet the E_b/N_0 training range is chosen to be $[18\,\mathrm{dB}, 27.5\,\mathrm{dB}]$, the Attention Detector and the FCNN exhibit better performance for an E_b/N_0 training range of $[15\,\mathrm{dB}, 27.5\,\mathrm{dB}]$.

D. Bit Error Ratio Performance – Uncoded Transmission with Imperfect Channel Knowledge

As mentioned earlier, for the results presented we assume to have perfect channel knowledge in form of the channel impulse response (CIR). However, in practice, the CIR has to be estimated as well, and the occurring estimation errors naturally degrade the performance of the equalizer. Hence, in this section we compare the influence of channel estimation errors on the performance of the investigated model-based and data-driven equalizers. For estimating the CIR, a known preamble, which is defined in [43], is transmitted prior to a data burst. Based on this preamble, the CIR is estimated with the BLUE as described in [44].

We regard the same system setting as in Sec. V-C and evaluate the BER performance of the equalizers for system I using the estimated CIRs. For the NN-based equalizers, we use the same hyperparameters as in case of perfect channel knowledge, however, the NNs are trained with estimated CIRs instead of true CIRs. The obtained results are shown in Fig. 6, where the BER results for imperfect channel knowledge are plotted with solid lines, and – for comparison – the results with perfect channel knowledge from Sec. V-C are plotted with dashed lines. It can be observed that imperfect channel knowledge degrades the performance of both model-based and NN-based equalizers in the same scale, namely by approximately $0.7\,\mathrm{dB}$.

E. Bit Error Ratio Performance – Coded Transmission

As already described in Sec. IV, the NNs are trained to provide estimates for the posterior probabilities for every data symbol estimate for both coded and uncoded data transmission. That is, we expect a trained NN-based data estimator to be applicable for coded and uncoded transmission without requiring retraining. As detailed in Sec. V-C, for uncoded transmission it is beneficial to train the NNs for different SNRs, where the SNR training range limits can be viewed as hyperparameters – with this approach a good, or even close to optimal BER performance can be achieved. However, employing these trained NNs for coded transmission, their performance is unsatisfactory. As shown in Fig. 9b exemplarily for DetNet, the NN-based equalizer trained in an E_b/N_0 range of $[1\,\mathrm{dB}, 9\,\mathrm{dB}]$ performs distinctly worse than the DFE and the LMMSE estimator, while the same NN outperforms both model-based equalizers for uncoded transmission (Fig. 9a). The reason for this result can be explained by investigating the empirical distribution of the LLRs provided by DetNet. Comparing the LLRs of DetNet trained in an E_b/N_0 range of $[1\,\mathrm{dB}, 9\,\mathrm{dB}]$ (Fig. 7a) with the true LLRs at $E_\mathrm{b}/N_0 = 4\,\mathrm{dB}$ (Fig. 7c) reveals that a vast number of LLRs provided by DetNet has a high absolute value[6], while this is not the case for the true LLRs and also not for the LLRs of the LMMSE estimator (Fig. 7b). That is, the NN is overconfident in many of its decisions, which harms the performance of the Viterbi channel decoder.

To tackle this problem, we investigated treating the data estimation problem as a classification task, i.e., we utilized Softmax as an output activation function of the NNs, combined with using cross-entropy loss for training. Then, so-called label smoothing can be applied, which is a common approach for combating overconfidence of classification NNs [45]. Unfortunately, this approach did not lead to significant performance improvements in our experiments. However, we observed that the training E_b/N_0 range has a large impact on the distribution of the LLRs provided by DetNet. More specifically, the overconfidence of an NN-based equalizer can be highly reduced by training at low SNRs. This highlights the importance of the training SNR as a hyperparameter, which has to be chosen differently for coded and uncoded data transmission.

Investigating solely the distribution of the LLRs, however, is only an indicator of their reliability. We utilize an approach described in [46] for an assessment of the LLR quality of turbo equalizers. To this end, we apply the trained NNs on the validation set, to obtain the estimated LLRs $L_{\mathrm{est},i}$ for all bits b_i contained in the validation set. The estimated LLRs $L_{\mathrm{est},i}$ are grouped according to their value into K bins with the value L_k, $k \in \{0, ..., K-1\}$ (L_k is the mean of the estimated LLRs in bin k). The signs of $L_{\mathrm{est},i}$ are used for a hard decision estimate of the corresponding bits b_i. With these hard decision estimates at hand, the empirical bit error probability

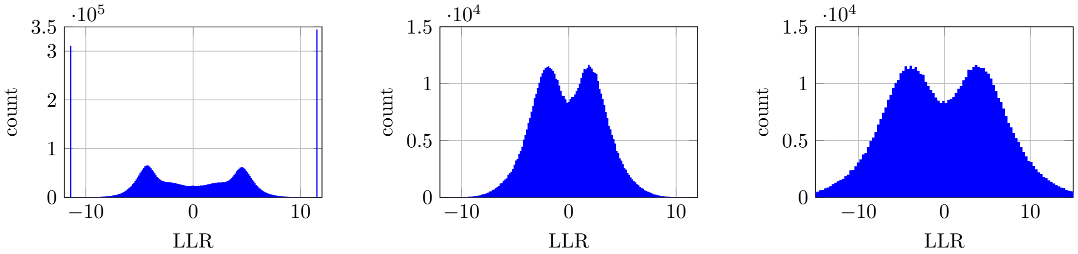
$$P_{\mathrm{emp},k} = \frac{\#\,\text{wrong hard decisions in bin}\,k}{\#\,\text{bits in bin}\,k}$$

can be computed for all K bins. These empirical bit error probabilities, in turn, can be utilized to determine the empirical LLRs $L_{\mathrm{emp},k}$ for all K bins with

$$L_{\mathrm{emp},k} = \mathrm{sign}(L_k) \left| \ln\left(\frac{1-P_{\mathrm{emp},k}}{P_{\mathrm{emp},k}}\right) \right| . \tag{25}$$

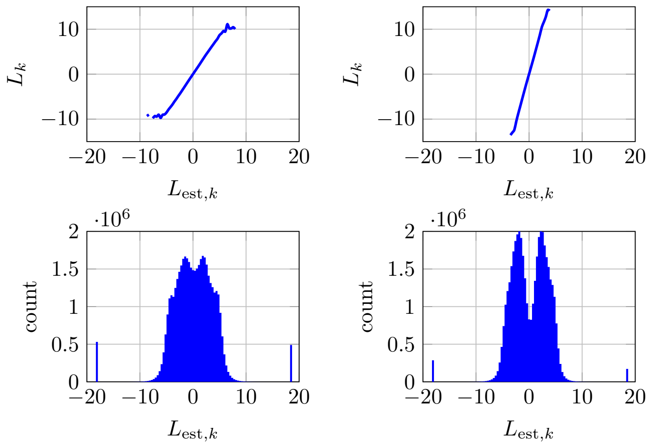
Assuming a sufficiently large number of LLR values per bin, the empirical LLRs $L_{\mathrm{emp},k}$ provide an approximation of the true LLRs. The quality of the estimated LLRs $L_{\mathrm{est},i}$ can be ascertained by plotting $L_{\mathrm{emp},k}$ against L_k. Since the estimated LLRs should match the empirical ones, the plotted graph is

[6] We introduced an upper and a lower limit for the output values of DetNet since, due to imperfect training, its output values can be slightly smaller than 0 or greater than 1, which leads to problems for the computation of the LLRs.



(a) DetNet (b) LMMSE (c) Empirical distrib. true LLRs

Fig. 7. Empirical distribution of the LLRs provided by DetNet trained in an E_b/N_0 range of $[1\,\mathrm{dB}, 9\,\mathrm{dB}]$ (a) and by the LMMSE estimator (b), compared with the empirical distribution of the true LLRs (c) at $E_\mathrm{b}/N_0 = 4\,\mathrm{dB}$.



(a) $E_\mathrm{b}/N_0 = 4\,\mathrm{dB}$ (b) $E_\mathrm{b}/N_0 = 8\,\mathrm{dB}$

Fig. 8. Distribution of the estimated LLRs provided by the DetNet trained at $1.5\,\mathrm{dB}$, and their relation to the empirical LLRs at a test SNR of (a) $E_\mathrm{b}/N_0 = 4\,\mathrm{dB}$ and (b) $E_\mathrm{b}/N_0 = 8\,\mathrm{dB}$.

ideally a linear function with slope one. However, also slopes not equal to one allow optimal channel decoding performance of the Viterbi decoder, since all LLRs are under- or overrated in the same fashion. Nonlinear graphs, in turn, indicate a loss in BER performance, since some estimated LLRs are overrated while others are underrated at the same time. This may lead to wrong decisions of the Viterbi channel decoder when searching the optimum path in the trellis diagram of the convolutional code. As shown in Fig. 8 for the LLRs provided by DetNet when being trained at $1.5\,\mathrm{dB}$, the number of LLRs with too high value could be drastically lowered. Further, the empirical LLRs and the estimated LLRs are related nearly linearly for the majority of the estimated LLRs, i.e., in the regions where the relation is nonlinear, the counts per LLR bin are comparatively small.

As the BER curves in Fig. 9b show, DetNet trained at $1.5\,\mathrm{dB}$ achieves close to optimal BER performance. Interestingly, for uncoded transmission, the DetNet trained at $E_\mathrm{b}/N_0 = 1.5\,\mathrm{dB}$ performs distinctly worse than the DetNet trained in the E_b/N_0 range of $[1\,\mathrm{dB}, 9\,\mathrm{dB}]$, which is depicted in Fig. 9a. This supports the our observation that NN-based equalizers have to be trained differently for channel coded and uncoded data transmission. For the Attention Detector and the FCNN, $E_\mathrm{b}/N_0 = 0.8\,\mathrm{dB}$ is utilized as an SNR for training, all other hyperparameters are chosen as for uncoded data transmission. Both achieve close to optimal BER performance, too.

For system II, we compare the LMMSE estimator, the DFE, and the DetNet, which is trained at $E_\mathrm{b}/N_0 = 4\,\mathrm{dB}$. As shown in Fig. 10, all three investigated equalizers exhibit approximately the same BER performance for coded data transmission. Although simulating the optimal BER performance is computationally infeasible, it can be stated that the achieved performance of the three equalizers is very close to the optimal performance. This statement can be verified by considering the LLRs provided by the LMMSE estimator. They are equivalent to the true LLRs when the conditional distribution $p(\hat{d}'_i|d'_i)$ is Gaussian (cf. Sec. III-D). Since this condition is well fulfilled for the system dimensions of system II, the LLRs of the LMMSE are close to the true LLRs, leading to close to optimal BER performance for coded data transmission.

F. Complexity Analysis

In this section, we provide a brief analysis of the inference complexity of the presented NN-based data estimators as well as of the LMMSE estimator and the DFE in terms of the number of required scalar, real-valued multiplications needed for equalization of one UW-OFDM data symbol. In this paper, we account four real-valued multiplications for one complex-valued multiplication. Data normalization, as well as the complexity required for training the NNs is not regarded in this analysis.

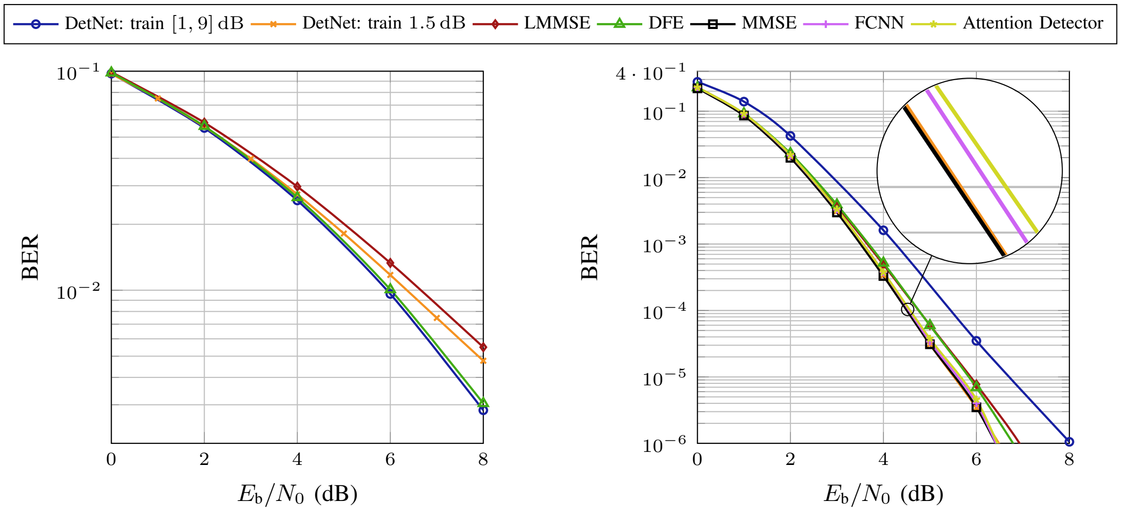
For DetNet, we first determine the complexity of a single layer. Given $\mathbf{H}^T\mathbf{H}$, the number of multiplications carried out in a layer according to (19) including the projection by an FCNN with a single hidden layer, one-hot demapping, and the weighted residual connections is

$$\begin{aligned} M_{\mathrm{DetNet},k} = {} & \underbrace{4N_\mathrm{d}^2}_{\mathbf{H}^T\mathbf{H}\hat{\mathbf{d}}_k} + \underbrace{2d_\mathrm{h}(N_\mathrm{d}(|\mathbb{S}|+1)+d_\mathrm{v})}_{\text{single hid. layer FCNN}} \\ & + \underbrace{2N_\mathrm{d}}_{\delta_{1k}\cdot} + \underbrace{2N_\mathrm{d}}_{\delta_{2k}\cdot} + \underbrace{2N_\mathrm{d}|\mathbb{S}|}_{\text{one-hot demap.}} + \underbrace{2N_\mathrm{d}+d_\mathrm{v}}_{\text{residual}} . \end{aligned} \tag{26}$$

Overall, DetNet has an inference complexity of

$$\begin{aligned} M_{\mathrm{DetNet}} = {} & LM_{\mathrm{DetNet},k} - 2N_\mathrm{d}|\mathbb{S}| + 8N_\mathrm{d}^2(N_\mathrm{d}+N_\mathrm{u}) \\ & + 4N_\mathrm{d}(N_\mathrm{d}+N_\mathrm{u}) + 2N_\mathrm{d}(2N_\mathrm{d}+1) \end{aligned} \tag{27}$$

real-valued multiplications, where we consider with the subtracted term that no one-hot decoding is conducted in the last layer, while with the three added terms the computations of $\mathbf{H}^T\mathbf{H}$ and of $\mathbf{H}^T\mathbf{y}$, and the preconditioning are taken into account.



(a) Uncoded case (b) Coded case

Fig. 9. Comparison uncoded and coded BER performance for system I, non-systematic UW-OFDM. DetNet is once trained in an E_b/N_0 range of $[1\,\mathrm{dB}, 9\,\mathrm{dB}]$, and once at $E_\mathrm{b}/N_0 = 1.5\,\mathrm{dB}$. The FCNN and the Attention Detector are trained at $E_\mathrm{b}/N_0 = 0.8\,\mathrm{dB}$.

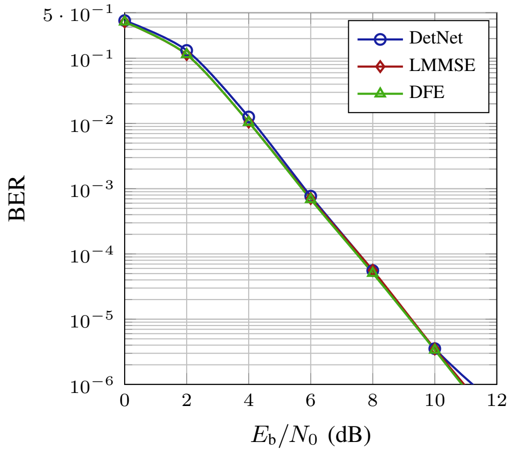


Fig. 10. BER performance comparison for system II, coded case.

Determining the number of multiplications required by the FCNN is straightforward and can be expressed as

$$M_\mathrm{FCNN} = \underbrace{(2N_\mathrm{d}^2 + 2N_\mathrm{d})d_\mathrm{h}}_{\text{input to hidden}} + \underbrace{d_\mathrm{h}^2(L-1)}_{\text{hidden to hidden}} + \underbrace{2d_\mathrm{h}N_\mathrm{d}|\mathbb{S}|}_{\text{hidden to output}} + 2(2N_\mathrm{d}^2 + N_\mathrm{d})(N_\mathrm{d} + N_\mathrm{u}) + 4N_\mathrm{d}(N_\mathrm{d} + N_\mathrm{u})\,, \tag{28}$$

where with the last two terms the computations of the upper triangular matrix (including the main diagonal) of $\mathbf{H}^T\mathbf{H}$ and of $\mathbf{H}^T\mathbf{y}$ are considered.

For the Attention Detector, we start by evaluating the complexity of a single encoder layer, which consists of a self-attention layer, a single hidden layer FCNN, layer normalization, and residual connections. The inputs of the self-attention layer are mapped to so-called queries $\mathbf{q}_i$, keys $\mathbf{k}_i$, and values $\mathbf{v}_i$, $i \in \{0, ..., 2N_\mathrm{d} - 1\}$, by multiplying with learned matrices [21]. Then, self-attention scores between each query and each key are computed, followed by a weighting of the values $\mathbf{v}_i$ by these scores. The number of multiplications conducted in one encoder layer thus is

$$\begin{aligned} M_{\mathrm{AttEnc},k} = {} & \underbrace{6N_\mathrm{d}(2N_\mathrm{d}+1)^2}_{\text{input mappings}} + \underbrace{4N_\mathrm{d}^2(4N_\mathrm{d}+3)}_{\text{score weighting}} \\ & + \underbrace{8N_\mathrm{d}(2N_\mathrm{d}+1)}_{\text{normalization, residual}} + \underbrace{4N_\mathrm{d}d_\mathrm{h,enc}(2N_\mathrm{d}+1)}_{\text{single hid. layer FCNN}}\,. \end{aligned} \tag{29}$$

For the FCNN on top of the encoder, another

$$\begin{aligned} M_\mathrm{AttFcnn} = {} & \underbrace{2N_\mathrm{d}d_\mathrm{h,fcnn}(2N_\mathrm{d}+1)}_{\text{input to hidden}} + \underbrace{(L_\mathrm{fcnn}-1)d_\mathrm{h,fcnn}^2}_{\text{hidden to hidden}} \\ & + \underbrace{2N_\mathrm{d}|\mathbb{S}|d_\mathrm{h,fcnn}}_{\text{hidden to output}} \end{aligned} \tag{30}$$

multiplications have to be performed. Hence, the complexity of the Attention Detector is given by

$$\begin{aligned} M_\mathrm{AttDet} = {} & L_\mathrm{enc}M_{\mathrm{AttEnc},k} + M_\mathrm{Att,fcnn} + 8N_\mathrm{d}^2(N_\mathrm{d}+N_\mathrm{u}) \\ & + 4N_\mathrm{d}(N_\mathrm{d}+N_\mathrm{u}) + 2N_\mathrm{d}(2N_\mathrm{d}+1)\,, \end{aligned} \tag{31}$$

where we consider the computations of $\mathbf{H}^T\mathbf{H}$ and $\mathbf{H}^T\mathbf{y}$, as well as the preconditioning with the last three terms.

For the LMMSE estimator, we first regard the complexity for obtaining the estimator matrix $\mathbf{E}_\mathrm{LMMSE}$. Since the channel is assumed to be stationary for a whole data burst, the estimator matrix has to be computed only once per burst. Assuming that the inversion in (10) is computed by a Cholesky decomposition as of [47], the computation of $\mathbf{E}_\mathrm{LMMSE}$ entails a complexity of

$$\begin{aligned} M_\mathrm{LMMSE,burst} &= \underbrace{\frac{14}{3}N_\mathrm{d}^3 + 4N_\mathrm{d}^2}_{\text{inverse (Cholesky)}} + \underbrace{8N_\mathrm{d}^2(N_\mathrm{d}+N_\mathrm{u})}_{\mathbf{H}'^H\mathbf{H}' \text{ \& multipl. with } \mathbf{H}'^H} \\ &= \frac{38}{3}N_\mathrm{d}^3 + 8N_\mathrm{d}^2N_\mathrm{u} + 4N_\mathrm{d}^2\,. \end{aligned} \tag{32}$$

Then, given $\mathbf{E}_\mathrm{LMMSE}$, the number of required multiplications for the equalization of every received UW-OFDM vector is

$$M_\mathrm{LMMSE,eq} = 4(N_\mathrm{d}+N_\mathrm{u})N_\mathrm{d}\,. \tag{33}$$

TABLE II
NUMBER OF REQUIRED MULTIPLICATIONS OF CONSIDERED EQUALIZERS ROUNDED TO HUNDREDS.

		System I	System II
DetNet	M_{DetNet}	100000	3178300
FCNN	M_{FCNN}	866400	15437800
Attention Detector	M_{AttDet}	614400	49971700
LMMSE	$M_{\text{LMMSE,burst}}$	8800	550200
	$M_{\text{LMMSE,eq}}$	400	6100
DFE	$M_{\text{DFE,burst}}$	11700	1644700
	$M_{\text{DFE,eq}}$	800	12300

We determine the DFE complexity by first considering those computations that have to be done once for every data burst. Namely, this refers to the computations of the estimator vectors $\mathbf{e}_k^H$ and the error covariance matrices $\mathbf{C}_{\text{ee},k} = N\sigma_{\text{n}}^2\mathbf{A}_k$, $\mathbf{A}_k = \left(\mathbf{H}_k'^H\mathbf{H}_k' + \frac{N\sigma_{\text{n}}^2}{\sigma_{\text{d}}^2}\mathbf{I}\right)^{-1}$, for every iteration step. We note that $\mathbf{H}'^H\mathbf{H}$ needs to be computed only once, and then the matrices $\mathbf{H}_k'^H\mathbf{H}_k$ can be retrieved by deleting the appropriate rows and columns. The size of $\mathbf{H}_k'$ decrements in every iteration, and thus we elaborate the complexity of computing $\mathbf{A}_k$ given $\mathbf{H}_k'^H\mathbf{H}_k \in \mathbb{C}^{C\times C}$, with $C \in \{2, ..., N_{\text{d}}\}$. Furthermore, the scaling of $\mathbf{A}_k$ by $N\sigma_{\text{n}}^2$ to obtain $\mathbf{C}_{\text{ee},k}$ can be omitted, since only the minimum value on the diagonal of $\mathbf{C}_{\text{ee},k}$ is needed for finding the data symbol to be estimated. In summary,

$$\begin{aligned} M_{\text{DFE,burst}} = {} & \underbrace{4N_{\text{d}}^2(N_{\text{d}}+N_{\text{u}})}_{\mathbf{H}'^H\mathbf{H}'} + \underbrace{4(N_{\text{d}}+N_{\text{u}}+1)}_{\text{last estimator vector}} \\ & + \sum_{C=2}^{N_{\text{d}}} \underbrace{\frac{14}{3}C^3 + 4C^2}_{\mathbf{A}_k\text{ (Cholesky)}} + \underbrace{4C(N_{\text{d}}+N_{\text{u}})}_{\mathbf{e}_k^H} \\ = {} & \frac{7}{6}N_{\text{d}}^4 + \frac{29}{3}N_{\text{d}}^3 + \frac{31}{6}N_{\text{d}}^2 + 6N_{\text{d}}^2N_{\text{u}} \\ & + \frac{2}{3}N_{\text{d}} + 2N_{\text{d}}N_{\text{u}} - \frac{14}{3} \end{aligned} \tag{34}$$

multiplications have to be carried out once for every data burst to obtain the N_{d} estimator vectors $\mathbf{e}_k^H$. For both the estimation of a single data symbol and the removal of the influence of this estimate on the received vector, $(N_{\text{d}} + N_{\text{u}})$ complex-valued multiplications have to be accounted for. Hence, given the estimator vectors, equalization of every received UW-OFDM vector with the DFE has a complexity of

$$M_{\text{DFE,eq}} = 8N_{\text{d}}^2 + 8N_{\text{d}}N_{\text{u}}\,. \tag{35}$$

The particular complexity numbers of the considered equalizers are stated in Tab. II for both system I and system II. Obviously, the NN-based equalizers exhibit a distinctly higher complexity than the considered model-based ones. However, a comparison of the complexities of the DetNet and the DFE reveals that the complexity of the DFE grows significantly faster with the dimension of the UW-OFDM system model than that of the DetNet. Among the considered NNs, the DetNet is the lowest complex equalizer. That is, incorporating model knowledge directly into the layers structure of an NN seems to be most promising for obtaining well-performing and comparably low complex NN-based data estimators.

G. Distributions of the Data Estimates

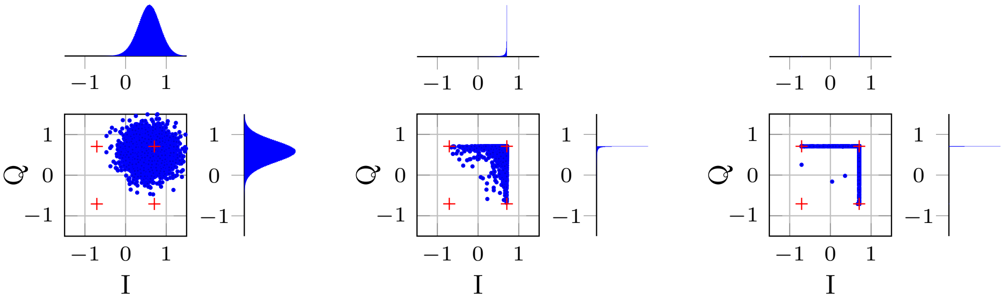
We also want to highlight the differences in the distributions of the estimates of the MMSE estimator, the NN-based estimators (exemplarily shown for DetNet), and the LMMSE estimator. To this end, we visualize the conditional distributions of their estimates, given a transmitted symbol $(1+j)/\sqrt{2}$, for system I at $E_{\text{b}}/N_0 = 4\,\text{dB}$ in in-phase/quadrature-phase (I/Q)-diagrams. The empirical distributions of the data symbol estimates are plotted in histograms along the I-axis and the Q-axis. As shown in Fig. 11a, the conditional LMMSE estimates follow, as expected, (approximately) a Gaussian distribution. However, the MMSE estimates are distributed in a completely different manner. As indicated by the histograms in Fig. 11b, the vast majority of the estimates are located very close to the constellation point. Since the MMSE estimator yields the posterior expectation of a data symbol as an estimate, no estimate can lie outside the square connecting the four constellation points (marked by red crosses). The estimates of DetNet, plotted in Fig. 11c, exhibit a distribution similar to that of the MMSE estimates. This is in fact expected, since, due to training the NNs with a quadratic loss function, the NNs try to minimize the cost metric that the MMSE estimator minimizes, namely the Bayesian mean square error. Hence, the trained NNs approximate the MMSE estimator function.

VI. Conclusion

In this paper, we investigated three NN-based approaches for data estimation in UW-OFDM systems, whereby model knowledge was utilized in different ways. Moreover, we described SOTA model-based equalizers, and we discussed the equivalence of the MMSE estimator and the bit-wise MAP estimator for the considered system setup. We pointed out the importance of proper data normalization for NN-based equalizers and proposed a data normalization scheme specifically for UW-OFDM signaling. With preconditioning, we introduced adaptions for DetNet to boost its BER performance and decrease its computational complexity. Further, we showed a model-inspired approach for data pre-processing, and we proposed an NN-based data estimator inspired by the Transformer network. Among the different approaches considered for exploiting model knowledge when conduction data estimation with NNs, deep unfolding seems to be the most promising approach for the UW-OFDM systems, since the adapted DetNet can achieve the best BER performance with the lowest inference complexity. We highlighted the difficulties when employing NNs for data estimation in channel coded data transmission, and we introduced a measure for obtaining reliable LLRs by NN-based equalizers. Finally, we provided BER performance results, we conducted a complexity analysis, and we visualized the distribution of the estimates of selected model-based and NN-based equalizers.

Appendix A
Equivalence of the MMSE and the Bit-Wise MAP Hard Decision Estimates for QPSK

When employing a QPSK alphabet, the data symbols d_i' are drawn from $\mathbb{S}' := \rho\{1+j, 1-j, -1+j, -1-j\}$, with $\rho = 1/\sqrt{2}$ for a normalized alphabet or $\rho = 1$ otherwise. For



(a) LMMSE

(b) MMSE

(c) DetNet

Fig. 11. Distribution of the conditional data symbol estimates for system I at $E_\mathrm{b}/N_0 = 4\,\mathrm{dB}$.

deriving the MMSE hard decision estimates, let us consider the MMSE estimate of the ith data symbol in the data vector:

$$\begin{aligned}\hat{d}_i' &= \sum_{\mathbf{d}''\in\mathbb{S}'^{N_\mathrm{d}}} d_i'' p[\mathbf{d}''|\mathbf{y}'] \\ &= \sum_{d''_{i,\mathrm{Re}}\in\mathbb{S}_\mathrm{Re}} \sum_{d''_{i,\mathrm{Im}}\in\mathbb{S}_\mathrm{Im}} (d''_{i,\mathrm{Re}} + j d''_{i,\mathrm{Im}}) p[(d''_{i,\mathrm{Re}} + j d''_{i,\mathrm{Im}})|\mathbf{y}'] \\ &= \sum_{d''_{i,\mathrm{Re}}\in\mathbb{S}_\mathrm{Re}} \sum_{d''_{i,\mathrm{Im}}\in\mathbb{S}_\mathrm{Im}} d''_{i,\mathrm{Re}} p[(d''_{i,\mathrm{Re}} + j d''_{i,\mathrm{Im}})|\mathbf{y}'] \\ &\quad + j \sum_{d''_{i,\mathrm{Re}}\in\mathbb{S}_\mathrm{Re}} \sum_{d''_{i,\mathrm{Im}}\in\mathbb{S}_\mathrm{Im}} d''_{i,\mathrm{Im}} p[(d''_{i,\mathrm{Re}} + j d''_{i,\mathrm{Im}})|\mathbf{y}'] \\ &= \sum_{d''_{i,\mathrm{Re}}\in\mathbb{S}_\mathrm{Re}} d''_{i,\mathrm{Re}} p[d''_{i,\mathrm{Re}}|\mathbf{y}'] + j \sum_{d''_{i,\mathrm{Im}}\in\mathbb{S}_\mathrm{Im}} d''_{i,\mathrm{Im}} p[j d''_{i,\mathrm{Im}}|\mathbf{y}'] \,,\end{aligned} \tag{36}$$

where $\mathbb{S}_\mathrm{Re} = \mathbb{S}_\mathrm{Im} = \{-\rho, \rho\}$, $d''_{i,\mathrm{Re}} := \mathrm{Re}\{d''_i\}$, and $d''_{i,\mathrm{Im}} := \mathrm{Im}\{d''_i\}$. That is, the real and the imaginary part of d_i are estimated independently of each other. Inserting the symbols of the symbol alphabet into (36) leads to

$$\begin{aligned}\hat{d}_i' &= -\rho p[\mathrm{Re}\{d_i'\} = -\rho|\mathbf{y}'] + \rho p[\mathrm{Re}\{d_i'\} = \rho|\mathbf{y}'] \\ &\quad + j(-\rho p[\mathrm{Im}\{d_i'\} = -\rho|\mathbf{y}'] + \rho p[\mathrm{Im}\{d_i'\} = \rho|\mathbf{y}']) \\ &= -\rho p[b_{0i} = 0|\mathbf{y}'] + \rho p[b_{0i} = 1|\mathbf{y}'] \\ &\quad + j(-\rho p[b_{1i} = 0|\mathbf{y}'] + \rho p[b_{1i} = 1|\mathbf{y}']) \,,\end{aligned} \tag{37}$$

where in the last step the QPSK bit-to-symbol mapping described in Sec. III-C is applied. In case of hard decision, $\hat{d}_i'$ is sliced to the closest constellation symbol, i.e., to ρ for $\mathrm{Re}\{\hat{d}_i\} > 0$, and to $-\rho$ otherwise (accordingly for $\mathrm{Im}\{\hat{d}_i\}$). Hence, the real and the imaginary part of an MMSE hard decision estimate $\lfloor \hat{d}_i' \rceil$ follow to

$$\mathrm{Re}\{\lfloor \hat{d}_i' \rceil\} = \begin{cases} \rho & p[b_{0i} = 1|\mathbf{y}'] > p[b_{0i} = 0|\mathbf{y}'] \\ -\rho & \text{otherwise} \end{cases} \tag{38}$$

and

$$\mathrm{Im}\{\lfloor \hat{d}_i' \rceil\} = \begin{cases} \rho & p[b_{1i} = 1|\mathbf{y}'] > p[b_{1i} = 0|\mathbf{y}'] \\ -\rho & \text{otherwise} \end{cases} \,, \tag{39}$$

respectively, coinciding with a hard decision estimate of the bit-wise MAP estimator.

References

[1] S. Baumgartner, G. Bognár, O. Lang, and M. Huemer, "Neural Network Based Data Estimation for Unique Word OFDM," in *Proceedings of the 55th Asilomar Conference on Signals, Systems, and Computers*, 2021, pp. 381–388.

[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] N. Samuel, T. Diskin, and A. Wiesel, "Learning to Detect," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2554–2564, 2019.

[4] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Model-Driven Deep Learning for MIMO Detection," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1702–1715, 2020.

[5] H. He, C. Wen, S. Jin, and G. Y. Li, "A Model-Driven Deep Learning Network for MIMO Detection," in *Proceedings of the 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP 2018)*, 2018, pp. 584–588.

[6] J. Liao, J. Zhao, F. Gao, and G. Y. Li, "A Model-Driven Deep Learning Method for Massive MIMO Detection," *IEEE Communications Letters*, vol. 24, no. 8, pp. 1724–1728, 2020.

[7] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive Neural Signal Detection for Massive MIMO," *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5635–5648, 2020.

[8] N. Shlezinger, N. Farsad, Y. C. Eldar, and A. J. Goldsmith, "ViterbiNet: A Deep Learning Based Viterbi Algorithm for Symbol Detection," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3319–3331, 2020.

[9] N. Shlezinger, R. Fu, and Y. C. Eldar, "DeepSIC: Deep Soft Interference Cancellation for Multiuser MIMO Detection," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1349–1362, 2021.

[10] K. Pratik, B. D. Rao, and M. Welling, "RE-MIMO: Recurrent and Permutation Equivariant Neural MIMO Detection," *IEEE Transactions on Signal Processing*, vol. 69, pp. 459–473, 2021.

[11] T. V. Luong, N. Shlezinger, C. Xu, T. M. Hoang, Y. C. Eldar, and L. Hanzo, "Deep Learning Based Successive Interference Cancellation for the Non-Orthogonal Downlink," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 11, pp. 1–13, 2022.

[12] S. Baumgartner, O. Lang, and M. Huemer, "Neural Network Based Single-Carrier Frequency Domain Equalization," in *Computer Aided Systems Theory - EUROCAST 2022*, ser. Lecture Notes in Computer Science (LNCS). Springer, 2023.

[13] K. Hornik, "Approximation Capabilities of Multilayer Feedforward Networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[14] J. R. Hershey, J. Le Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," *arXiv preprint arXiv:1409.2574*, 2014.

[15] M. Mohammadkarimi, M. Mehrabi, M. Ardakani, and Y. Jing, "Deep Learning-Based Sphere Decoding," *IEEE Transactions on Wireless Communications*, vol. 18, no. 9, pp. 4368–4378, 2019.

[16] M. Huemer, C. Hofbauer, and J. Huber, "The Potential of Unique Words in OFDM," in *Proceedings of the 15th International OFDM-Workshop*, 2010, pp. 140–144.

[17] M. Huemer, A. Onic, and C. Hofbauer, "Classical and Bayesian Linear Data Estimators for Unique Word OFDM," *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 6073–6085, 2011.

[18] A. Onic and M. Huemer, "Noise Interpolation for Unique Word OFDM," *IEEE Signal Processing Letters*, vol. 21, no. 7, pp. 814–818, 2014.

[19] ——, "Sphere Decoding for Unique Word OFDM," in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, 2011, pp. 1–5.

[20] A. Onic, *Receiver Concepts for Unique Word OFDM*, 2013, PhD Dissertation, Alpen-Adria-Universität Klagenfurt.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017, pp. 5998–6008.

[22] M. Rajabzadeh, H. Khoshbin, and H. Steendam, "Sidelobe Suppression for Non-systematic Coded UW-OFDM in Cognitive Radio Networks,"

in *Proceedings of the European Wireless 2014; 20th European Wireless Conference*, 2014, pp. 1–6.

[23] M. Huemer, C. Hofbauer, and J. B. Huber, "Non-Systematic Complex Number RS Coded OFDM by Unique Word Prefix," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 285–299, 2012.

[24] M. Huemer, C. Hofbauer, A. Onic, and J. B. Huber, "Design and Analysis of UW-OFDM Signals," *International Journal of Electronics and Communications (AEÜ)*, vol. 68, no. 10, pp. 958–968, 2014.

[25] C. Hofbauer, W. Haselmayr, and M. Huemer, "Pilot Tone Insertion and Utilization in Unique Word OFDM," in *Proceedings of the 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2020, pp. 1–5.

[26] C. Hofbauer, *Design and Analysis of Unique Word OFDM*, 2016, PhD Dissertation, Alpen-Adria-Universität Klagenfurt.

[27] E. G. Ström, "Chapter 4 - Optimal Detection of Digital Modulations in AWGN," in *Academic Press Library in Mobile and Wireless Communications - Transmission Techniques for Digital Communications*. Academic Press, 2016, pp. 121–169.

[28] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1993, vol. 1.

[29] O. Lang, M. Huemer, and C. Hofbauer, "On the Log-Likelihood Ratio Evaluation of CWCU Linear and Widely Linear MMSE Data Estimators," in *Proceedings of the 2016 50th Asilomar Conference on Signals, Systems and Computers*, 2016, pp. 633–637.

[30] W. Haselmayr and A. Springer, "Extrinsic LLR Computation by the SISO LMMSE Detector: Four Different Approaches," in *Computer Aided Systems Theory – EUROCAST 2015*, ser. Lecture Notes in Computer Science (LNCS), 2015, pp. 529–536.

[31] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, *Efficient BackProp*. Springer Berlin Heidelberg, 2012, pp. 9–48.

[32] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., 1995.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[35] G. Casella and R. L. Berger, *Statistical Inference*, 2nd ed. Thomson Learning, 2002.

[36] S. Yang and L. Hanzo, "Fifty Years of MIMO Detection: The Road to Large-Scale MIMOs," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 1941–1988, 2015.

[37] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *International Conference on Learning Representations*, 2021.

[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[39] J. Ma and L. Ping, "Orthogonal AMP," *IEEE Access*, vol. 5, pp. 2020–2033, 2017.

[40] J. Fakatselis, "Criteria for 2.4 GHz PHY Comparison of Modulation," IEEE Document, 1997, IEEE P802.11-97/157r1.

[41] A. Wiesel, "Tutorial 3 - From Iterative Algorithms to Deep Learning," Jun. 2018, IEEE International Conference on Accoustics, Speech and Signal Processing (ICASSP).

[42] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[43] "IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: High Speed Physical Layer in the 5 GHz band," *IEEE Std 802.11a-1999*, pp. 1–102, 1999.

[44] M. Huemer and O. Lang, "On Component-Wise Conditionally Unbiased Linear Bayesian Estimation," in *Proceedings of the 2014 48th Asilomar Conference on Signals, Systems and Computers*, Nov. 2014, pp. 879–885.

[45] R. Müller, S. Kornblith, and G. Hinton, "When Does Label Smoothing Help?" in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 4694–4703.

[46] G. Bauch, *"Turbo-Entzerrung" und Sendeantennen-Diversität mit "Space-Time-Codes" im Mobilfunk (in German)*, ser. 10, 2001, no. 660.

[47] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Johns Hopkins University Press, 2013.

Stefan Baumgartner (Graduate Student Member, IEEE) was born in Linz, Austria, in 1996. From 2016 to 2021, he studied Electronics and Information Technology at Johannes Kepler University (JKU) Linz, where he obtained his Dipl.-Ing. degree (MSc equivalent). During his studies, he specialized in signal and information processing. Since 2021, he is with the Institute of Signal Processing at JKU, where he is currently pursuing his PhD. His research focuses on machine learning approaches in communications engineering.

Gergő Bognár received his B.Sc., M.Sc., and Ph.D. degrees in computer science from the Eötvös Loránd University (ELTE), Budapest, Hungary, in 2012, 2014, and 2020, respectively. From 2020 to 2021, he was a postdoctoral researcher with the Institute of Signal Processing, Johannes Kepler University Linz, Austria. Currently, he is an assistant professor with the Department of Numerical Analysis, ELTE. His main research interest is signal and image processing by means of adaptive mathematical transformations and machine learning approaches, model-based neural networks, and knowledge-augmented deep learning.

Oliver Lang (Member, IEEE) received the bachelor's degree in electrical engineering and information technology and the master's degree in microelectronics from the Vienna University of Technology, Austria, in 2011 and 2014, respectively. From 2014 to 2018, he was a member of the Institute of Signal Processing (ISP) at the Johannes Kepler University (JKU) Linz, Austria, where he received his Ph.D. in 2018. From 2018 to 2019 he worked at DICE GmbH in Linz, which was a subsidiary company of Infineon Austria GmbH. During this period, he worked on automotive radar MMICs and systems. Since March 2019, he is a university assistant with Ph.D. at the ISP at JKU. He is main inventor of several patents in the field of automotive radar systems and main author of several publications in the field of estimation theory and adaptive filtering.

Mario Huemer (Senior Member, IEEE) received his Dipl.-Ing. and Dr.techn. degrees from the Johannes Kepler University (JKU) Linz, Austria, in 1996 and 1999, respectively. After holding positions in industry and academia he was an associate professor at the University of Erlangen-Nuremberg, Germany, from 2004 to 2007, and a full professor at Klagenfurt University, Austria, from 2007 to 2013. From 2012 to 2013 he served as dean of the Faculty of Technical Sciences. In September 2013 Mario Huemer moved back to Linz, Austria, where he is now heading the Institute of Signal Processing at JKU Linz as a full professor. Since 2017 he is co-head of the "Christian Doppler Laboratory for Digitally Assisted RF Transceivers for Future Mobile Communications". His research focuses on statistical and adaptive signal processing, signal processing architectures and implementations, as well as mixed signal processing with applications in information and communications engineering, radio frequency and baseband integrated circuits, sensor and biomedical signal processing. Within these fields he has published more than 300 scientific papers. Mario Huemer is member of the IEEE Signal Processing, the Circuits and Systems, the Microwave Theory and Techniques, and the Communications societies, the German Society of Information Technology (ITG), and the Austrian Electrotechnical Association (OVE). In 2000 Mario Huemer received the dissertation awards of the ITG and the Austrian Society of Information and Communications Technology (GIT), in 2010 the Austrian Cardinal Innitzer award in natural sciences, and in 2016 the German ITG award. From 2009 to 2015 he was member of the editorial board of the International Journal of Electronics and Communications (AEU), and from May 2017 to April 2019 he served as an associate editor for the IEEE Signal Processing Letters.