HAL open science

mpNet: variable depth unfolded neural network for massive MIMO channel estimation

Taha Yassine, Luc Le Magoarou

▶ To cite this version:

Taha Yassine, Luc Le Magoarou. mpNet: variable depth unfolded neural network for massive MIMO channel estimation. 2020. hal-02913045v1

HAL Id: hal-02913045

https://hal.science/hal-02913045v1

Preprint submitted on 7 Aug 2020 (v1), last revised 9 Dec 2021 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

mpNet: variable depth unfolded neural network for massive MIMO channel estimation

Taha Yassine, Luc Le Magoarou

***Abstract*—Massive MIMO communication systems have a huge potential both in terms of data rate and energy efficiency, although channel estimation becomes challenging for a large number of antennas. Using a physical model allows to ease the problem by injecting a priori information based on the physics of propagation. However, such a model rests on simplifying assumptions and requires to know precisely the configuration of the system, which is unrealistic in practice. In this paper we present mpNet, an unfolded neural network specifically designed for massive MIMO channel estimation. It is trained online in an unsupervised way. Moreover, mpNet is computationally efficient and automatically adapts its depth to the SNR. The method we propose adds flexibility to physical channel models by allowing a base station to automatically correct its channel estimation algorithm based on incoming data, without the need for a separate offline training phase. It is applied to realistic millimeter wave channels and shows great performance, achieving a channel estimation error almost as low as one would get with a perfectly calibrated system. It also allows incident detection and automatic correction, making the base station resilient and able to automatically adapt to changes in its environment.**

***Index Terms*—Deep unfolding, MIMO channel estimation, online learning.**

I. Introduction

DATA processing techniques are often based on the *manifold assumption*: Meaningful data (signals) lie near a low dimensional manifold, although their apparent dimension is much larger [1], [2] [3, Section 5.11.3] [4, Section 9.3]. This fact has classically been exploited in two different ways.

On the one hand, for decades if not centuries, scientists have handcrafted *analytical models*. This amounts to come up with a mathematical description of the manifold, based on domain knowledge and careful observation of the phenomena of interest. This approach reaches its limits for complex phenomena that are difficult to model with a reasonable number of parameters, in which case simplifying assumptions have to be made, hindering model relevance.

On the other hand, thanks to the advent of modern computers, *machine learning* techniques have emerged and led to tremendous successes in various domains [5], [3]. One of their main feature is to avoid any explicit mathematical description of the manifold at hand, which is taken into account via a large amount of training data sampling it. Such an approach is particularly successful in application domains for which building analytical models is difficult, since it is much more flexible. However, flexibility comes at the price of computationally heavy learning and difficulties to inject a priori knowledge on the phenomena at hand.

Recently, a promising approach meant to combine the advantages of the two aforementioned approaches has been proposed under the name of *deep unfolding* (also known as deep unrolling). It amounts to unfold iterative algorithms initially based on analytical models so as to express them as neural networks that can be optimized [6], [7], [8]. This has the advantage of adding flexibility to algorithms based on classical models, and amounts to constrain the search space of neural networks by using domain knowledge. Moreover, this leads to inference algorithms of controlled complexity (see [9] and references therein for a complete survey).

Channel estimation is of paramount importance for communication systems in order to optimize the data rate/energy consumption tradeoff. In modern systems, the possibly large number of transmit/receive antennas and subcarriers makes this task difficult. For example, it has been recently proposed to use massive multiple input multiple output (massive MIMO) wireless systems [10], [11], [12] with a large number of antennas in the millimeter-wave band [13], [14], where a large bandwidth can be exploited. In that case the channel comprises hundreds or even thousands of complex numbers, whose estimation is a very challenging signal processing problem [15]. Fortunately, despite the high dimension of the channel, realistic channels are often well approximated by only a few dominant propagation paths [16] (typically less than ten). Such channels are said *sparse*. For massive MIMO channel estimation, it is thus customary to use an analytical model based on the physics of propagation in order to ease the problem. This amounts to parameterize a manifold by physical parameters such as the directions, delays and gains of the dominant propagation paths, the dimension of the manifold being equal to the number of real parameters considered in the model. Physical channel models allow injecting strong a priori knowledge based on solid principles [17], [18], [19], but necessarily make simplifying assumptions (e.g. the plane wave assumption [20]) and require knowing exactly the system configuration (positions of the antennas, gains, etc.). Such requirements being unrealistic in practice, the massive MIMO channel estimation task could perfectly benefit from the flexibility offered by the deep unfolding approach.

Contributions. In this paper, we introduce mpNet, an unfolded neural network specifically designed for massive MIMO channel estimation. The unfolded algorithm is matching pursuit [21], which is a greedy computationally efficient algorithm taking advantage of the massive MIMO channel sparsity. The network takes as input the least squares channel

Taha Yassine and Luc Le Magoarou are with bcom, Rennes, France. Contact addresses: taha.yassine@b-com.com, luc.lemagoarou@b-com.com.

estimates, with the objective to denoise them. The weights of mpNet are initialized using an imperfect physical model. The main concerns while designing the method were to make it:

- **Unsupervised**: no need for clean channels to train mpNet, the method uses the least squares channel estimates as training data and a reconstruction cost function. It is thus trained as an autoencoder [22].
- **Online**: no need for a separate offline training phase, mpNet is initialized with an imperfect physical model and trained incrementally using online gradient descent [23].
- **SNR adaptive**: no need for several networks trained at different SNRs, mpNet automatically adapts its depth to incoming data.
- **Computationally efficient**: backpropagation through mpNet is cheaper than classical channel estimation using a greedy sparse recovery algorithm (which corresponds to the forward pass of mpNet).

To the best of the authors' knowledge, the proposed method is the only one meeting all these requirements. Such a method is particularly suited to imperfectly known or non-calibrated systems. Indeed, starting from an imperfect physical channel model, our method allows a base station to automatically correct its channel estimation algorithm based on incoming data.

Note that this paper is partly based on a previously published work [24]. This paper is however much more exhaustive, since the method is exposed in a more comprehensive way, SNR adaptivity is totally new, and a much more extensive set of experiments is performed in order to empirically assess the method.

Related work. Machine learning holds promise for wireless communications (see [25], [26], [27] for exhaustive surveys). More specifically, since the physics of propagation provides pretty accurate analytical models, model-driven machine learning approaches [28] seem particularly suited. The method we propose can be seen as an instance of the model-driven approach.

In the context of massive MIMO channel estimation, it has recently been proposed to use adaptive data representations using dictionary learning techniques [29]. However, classical dictionary learning employing algorithms such as K-SVD [30], as proposed in [29], is very computationally heavy, and thus not suited to online learning, as opposed to the mpNet approach.

Deep unfolding has also been considered by communications researchers (see [31] for a survey). It has mainly been used for symbol detection, unfolding projected gradient descent [32] or approximate massage passing [33] algorithms.

Regarding MIMO channel estimation, it has been proposed in [34] to unfold a sparse recovery algorithm named denoising-based approximate message passing (DAMP) [35]. However, the method is directly adapted from image processing [36] and does not make use of a physical channel model as initialization as we propose here. A recent work also proposes to use deep unfolding of the DAMP algorithm for MIMO channel estimation [37], using a physical model to optimize the shrinkage functions used by DAMP. However, these previously proposed methods all require collecting a database of clean channels and an offline training phase, due to their intrinsic supervised nature. This may hinder their practical applicability. Moreover, the proposed unfolded neural networks are much more computationally complex than mpNet, and do not comprise an automatic way to adapt to the SNR.

Finally, deep learning has also been applied to channel estimation in an orthogonal frequency-division multiplexing (OFDM) context, which is mathematically very close to the one studied here. In [38], a neural network is used as a post-treatment of the least squares channel estimates in order to denoise them, and included in a joint channel estimation and detection framework. In [39], the noisy time-frequency response of the channel is viewed as an image and is denoised using classical denoising neural networks. Once again, these approaches are supervised, and are of high complexity compared to classical methods [40].

Organization. The remaining of the paper is organized as follows. First, section II introduces the problem at hand and describes the physical model on which mpNet is based. Then, the motivations behind the proposed solution is presented in section III. Section IV introduces in details mpNet: the deep unfolding based strategy we propose for MIMO channel estimation. In section V, different experiments are conducted in order to assess and validate the potential of our approach. Finally, section VI discusses the contributions and concludes the paper.

II. Problem formulation

A. System settings

We consider in this paper a massive MIMO system, also known as multi-user MIMO (MU-MIMO) system [10], [11], [12], in which a base station equipped with N antennas communicates with K single antenna users ($K < N$). Let us consider for ease of presentation a transmission on a single subcarrier, even though everything presented in the paper can obviously be generalized to the multicarrier case. The system operates in time division duplex (TDD) mode, so that channel reciprocity holds and the channel is estimated in the uplink: each user sends a pilot sequence $\mathbf{p}_k \in \mathbb{C}^T$ of duration T (orthogonal to the sequences of the other users, $\mathbf{p}_k^H \mathbf{p}_l = \delta_{kl}$) for the base station to estimate the channel. The received signal is thus expressed

$$\mathbf{R} = \sum_{k=1}^{K} \mathbf{h}_k \mathbf{p}_k^H + \mathbf{N}, \tag{1}$$

where $\mathbf{N} \in \mathbb{C}^{N \times T}$ is Gaussian noise. After correlating the received signal with the pilot sequences, and assuming no pilot contamination from adjacent cells for simplicity, the base station gets noisy measurements of the channels of all users, taking the form

$$\mathbf{x}_l \triangleq \mathbf{R}\mathbf{p}_l = \underbrace{\sum\nolimits_{k=1}^{K} \mathbf{h}_k \mathbf{p}_k^H \mathbf{p}_l}_{\mathbf{h}_l} + \underbrace{\mathbf{N}\mathbf{p}_l}_{\mathbf{n}_l}, \tag{2}$$

for $l = 1, \ldots, K$, with $\mathbf{n}_l \sim \mathcal{CN}(0, \sigma^2 \mathbf{Id})$, $\forall l$. In order to simplify notations in the remaining parts of the paper, we

drop the user index and denote such measurements with the canonical expression

$$\mathbf{x} = \mathbf{h} + \mathbf{n}, \tag{3}$$

where $\mathbf{h}$ is the channel of the considered user and $\mathbf{n}$ is noise, with $\mathbf{n} \sim \mathcal{CN}(0, \sigma^2 \mathbf{Id})$. Such a dropping of the user index does not harm the description of our approach, since it treats the channels of all users indifferently. Note that $\mathbf{x}$ is already an unbiased estimator of the channel, obtained by solving a least squares estimation problem, so that we call it the least squares (LS) estimator in the sequel. Its performance can be assessed by the input signal-to-noise ratio

$$\text{SNR}_{\text{in}} \triangleq \frac{\|\mathbf{h}\|_2^2}{N\sigma^2}.$$

However, one can get better channel estimates using a physical model which allows to denoise the least squares estimate, as is explained in the next subsection.

B. Physical model

Let us denote $\{g_1, \ldots, g_N\}$ the complex gains of the base stations antennas and $\{\overrightarrow{a_1}, \ldots, \overrightarrow{a_N}\}$ their positions with respect to the centroid of the antenna array. Then, under the plane wave assumption and assuming omnidirectional antennas (isotropic radiation patterns), the channel resulting from a single propagation path with direction of arrival (DoA) $\overrightarrow{u}$ is proportional to the *steering vector*

$$\mathbf{e}(\overrightarrow{u}) \triangleq (g_1 \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda}\overrightarrow{a_1}.\overrightarrow{u}}, \ldots, g_N \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda}\overrightarrow{a_N}.\overrightarrow{u}})^T$$

which reads $\mathbf{h} = \beta \mathbf{e}(\overrightarrow{u})$, with $\beta \in \mathbb{C}$. In that case, a sensible estimation strategy [17], [18], [19] is to build a dictionary of steering vectors corresponding to A potential DoAs: $\mathbf{E} \triangleq \big(\mathbf{e}(\overrightarrow{u_1}), \ldots, \mathbf{e}(\overrightarrow{u_A})\big)$ and to compute a channel estimate with the procedure

$$\begin{aligned} \overrightarrow{v} &= \text{argmax}_{\overrightarrow{u_i}} \ |\mathbf{e}(\overrightarrow{u_i})^H \mathbf{x}|, \\ \hat{\mathbf{h}} &= \mathbf{e}(\overrightarrow{v})\mathbf{e}(\overrightarrow{v})^H \mathbf{x}. \end{aligned} \tag{4}$$

The first step of this procedure amounts to find the column of the dictionary the most correlated with the observation (least square channel estimate) to estimate the DoA and the second step amounts to project the observation on the corresponding steering vector. The SNR at the output of this procedure reads

$$\text{SNR}_{\text{out}} \triangleq \frac{\|\mathbf{h}\|_2^2}{\mathbb{E}\big[\|\mathbf{h} - \hat{\mathbf{h}}\|_2^2\big]},$$

and we have at best $\text{SNR}_{\text{out}} = N\text{SNR}_{\text{in}}$ (neglecting the discretization error), if the selected steering vector is collinear to the actual channel. This is a direct consequence of the Cauchy-Schwarz inequality, and is intuitively explained by the fact that from the N complex dimensions of $\mathbf{x}$ corrupted by noise, only one is kept when projecting on the best steering vector, so that the effective noise variance is divided by N. The potential gain of using such a physical model can be huge, especially for massive MIMO systems in which the number of antennas N is large.

Moreover, this strategy can be generalized to estimate sparse multipath channels of the form

$$\mathbf{h} = \sum\nolimits_{p=1}^{P} \beta_p \mathbf{e}(\overrightarrow{u_p}), \tag{5}$$

by iterating the procedure (4) until some predefined stopping criterion is met. This leads to greedy sparse recovery algorithms such as matching pursuit (MP) [21] or orthogonal matching pursuit (OMP) [41]. Since the method proposed in this paper is based on the unfolding of the matching pursuit algorithm, a high level overview of it applied to channel estimation, with dictionary $\mathbf{E}$ and input $\mathbf{x}$ is given in algorithm 1.

Algorithm 1 Matching pursuit [21] (high level overview)

Input: Dictionary $\mathbf{E}$, input $\mathbf{x}$ (noisy channel)
1: $\mathbf{r} \leftarrow \mathbf{x}$
2: **while** Stopping criterion not met **do**
3: Find the most correlated atom: $s \leftarrow \underset{i}{\text{argmax}} \ |\mathbf{e}_i^H \mathbf{r}|$
4: Update the residual: $\mathbf{r} \leftarrow \mathbf{r} - \mathbf{e}_s \mathbf{e}_s^H \mathbf{r}$
5: **end while**
Output: $\hat{\mathbf{h}} \leftarrow \mathbf{x} - \mathbf{r}$ (denoised channel)

III. Motivation: imperfect models

Basing an estimation strategy on a physical model, as suggested in the previous section, requires knowing precisely the physical parameters of the system (in particular the positions and gains of the antennas) in order to build an appropriate dictionary. Then, even in the case of perfect system knowledge, some simplifying hypotheses (such as the plane wave assumption considered in the previous section) have to be made in order to keep the model mathematically tractable. Consequently, every model, regardless of its sophistication, is necessarily imperfect. Such a situation is well-known and summarized by the aphorism *"All models are wrong"* [42].

In the context of MIMO channel estimation, what is the impact of an imperfect knowledge of the physical parameters and/or of the invalidity of some hypotheses? In order to address this question, let us perform a simple experiment. Consider an antenna array of $N = 64$ antennas at the base station, whose known *nominal* configuration is a uniform linear array (ULA) of unit gain antennas separated by half-wavelengths and aligned with the x-axis. This nominal configuration corresponds to gains and positions $\{\tilde{g}_i, \tilde{\overrightarrow{a_i}}\}_{i=1}^N$. Now, suppose the knowledge of the system configuration is imperfect, meaning that the unknown *true* configuration of the system is given by the gains and positions $\{g_i, \overrightarrow{a_i}\}_{i=1}^N$, with

$$\begin{aligned} &g_i = \tilde{g}_i + n_{g,i},\ n_{g,i} \sim \mathcal{CN}(0, \sigma_g^2), \\ &\overrightarrow{a_i} = \tilde{\overrightarrow{a_i}} + \lambda \mathbf{n}_{p,i},\ \mathbf{n}_{p,i} = (e_{p,i}, \quad 0, \quad 0)^T, e_{p,i} \sim \mathcal{N}(0, \sigma_p^2). \end{aligned} \tag{6}$$

This way, σ_g (resp. σ_p) quantifies the uncertainty about the antenna gains (resp. spacings). Moreover, let

$$\tilde{\mathbf{e}}(\overrightarrow{u}) \triangleq (\tilde{g}_1 \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda}\tilde{\overrightarrow{a_1}}.\overrightarrow{u}}, \ldots, \tilde{g}_N \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\lambda}\tilde{\overrightarrow{a_N}}.\overrightarrow{u}})^T$$

be the nominal steering vector and $\tilde{\mathbf{E}} \triangleq \left(\tilde{\mathbf{e}}(\overrightarrow{u_1}), \ldots, \tilde{\mathbf{e}}(\overrightarrow{u_A})\right)$ be a dictionary of nominal steering vectors. The experiment consists in comparing the estimation strategy of (4) using the true (perfect but unknown) dictionary $\mathbf{E}$ with the exact same strategy using the nominal (imperfect but known) dictionary $\tilde{\mathbf{E}}$. To do so, we generate measurements according to (3) with channels of the form $\mathbf{h} = \mathbf{e}(\overrightarrow{u})$ where $\overrightarrow{u}$ corresponds to azimuth angles chosen uniformly at random, and SNR_{in} is set to $10\,\text{dB}$. Then, the dictionaries $\mathbf{E}$ and $\tilde{\mathbf{E}}$ are built by choosing $A = 32N$ directions corresponding to evenly spaced azimuth angles. Let $\hat{\mathbf{h}}_{\mathbf{E}}$ be the estimate obtained using $\mathbf{E}$ in (4), and $\hat{\mathbf{h}}_{\tilde{\mathbf{E}}}$ the estimate obtained using $\tilde{\mathbf{E}}$. The SNR loss (performance decrease) caused by using the imperfect but known dictionary $\tilde{\mathbf{E}}$ instead of the perfect but unknown dictionary $\mathbf{E}$ is measured by the quantity

$$\frac{\|\hat{\mathbf{h}}_{\tilde{\mathbf{E}}} - \mathbf{h}\|_2^2}{\|\hat{\mathbf{h}}_{\mathbf{E}} - \mathbf{h}\|_2^2}.$$

Results in terms of SNR loss, in average over 10 antenna array realizations and 1000 channel realizations per antenna array realization are shown on figure 1. From the figure, it is obvious that even a relatively small uncertainty about the system configuration can cause a great SNR loss. For example, an uncertainty of 0.03λ on the antenna spacings and of 0.09 on the antenna gains leads to an SNR loss of more than $10\,\text{dB}$, which means that the mean squared error undergoes a more than tenfold increase.

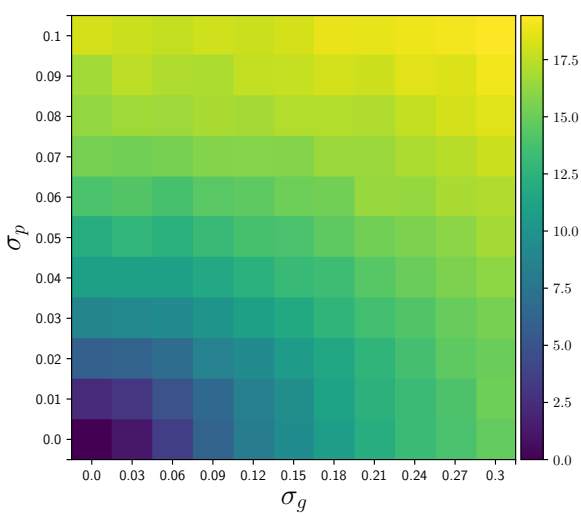

Fig. 1: SNR loss in decibels (dB) due to imperfect knowledge of the system.

This short experiment simply highlights the fact that using imperfect models can severely harm MIMO channel estimation performance. Note that we took here as an example of imperfection the uncertainty about antenna positions and gains, but many other sources of imperfection will impact real world practical systems. The main contribution of this paper is to propose a method that takes into account and corrects to some extent imperfect physical models using machine learning, and more specifically deep unfolding.

IV. Proposed approach: mpNet

Let us now propose a strategy based on deep unfolding allowing to correct a channel estimation algorithm based on an imperfect physical model incrementally, via online learning.

A. Unfolding matching pursuit

The estimation strategy summarized in algorithm 1 can be unfolded as a neural network taking the observation $\mathbf{x}$ as input and outputting a channel estimate $\hat{\mathbf{h}}$. Indeed, the first step in the while loop amounts to perform a linear transformation (multiplying the input by the matrix $\mathbf{E}^H$) followed by a nonlinear one (finding the inner product of maximum amplitude and setting all the others to zero) and the second step corresponds to a linear transformation (multiplying by the matrix $\mathbf{E}$). Such a strategy is parameterized by the dictionary of steering vectors $\mathbf{E}$. In the case where the optimal dictionary $\mathbf{E}$ is unknown (or imperfectly known), we propose to learn the dictionary matrix used in (4) directly on data via backpropagation [22].

Neural network structure. This is done by considering the dictionary matrix as weights of the neural network we introduce, called mpNet, whose forward pass is given in algorithm 2. The notation HT_1 refers to the hard thresholding operator which keeps only the entry of greatest modulus of its input and sets all the others to zero. The parameters of this neural network are the weights $\mathbf{W} \in \mathbb{C}^{N \times A}$, where A is an hyperparameter denoting the number of considered atoms in the dictionary. Note that complex weights and inputs are handled classically by stacking the real and imaginary parts for vectors and using the real representation for matrices. The forward pass of mpNet can be seen as a sequence of K iterations whose schematic description is shown on figure 2. The stopping criterion determining the number K of replications of the aforementioned structure, which corresponds to the number of estimated paths, is studied in section IV-C, with the objective to make the depth of mpNet adaptive to the SNR.

Algorithm 2 Forward pass of mpNet

Input: Weight matrix $\mathbf{W} \in \mathbb{C}^{N \times A}$, input $\mathbf{x}$

$\mathbf{r} \leftarrow \mathbf{x}$
while Stopping criterion not met **do**
  $\mathbf{r} \leftarrow \mathbf{r} - \mathbf{W}\text{HT}_1(\mathbf{W}^H\mathbf{r})$
end while

Output: $\texttt{FW}(\mathbf{W}, \mathbf{x}) \leftarrow \mathbf{x} - \mathbf{r}$

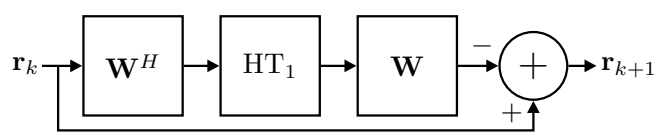

Fig. 2: One layer of mpNet.

B. Training mpNet

The method we propose to jointly estimate channels while simultaneously correcting an imperfect physical model is summarized in algorithm 3. Note that mpNet is fed with

normalized inputs of the form $\frac{\mathbf{x}}{\|\mathbf{x}\|_2}$, since we noticed it improved its performance. The training strategy amounts to initialize the weights of mpNet with a dictionary of nominal steering vectors $\tilde{\mathbf{E}}$ and then to perform a minibatch gradient descent [43] on the weights $\mathbf{W}$ to minimize the risk

$$R \triangleq \mathbb{E}\left[\frac{1}{2}\frac{\|\mathbf{x}-\hat{\mathbf{h}}\|_2^2}{\|\mathbf{x}\|_2^2}\right]. \tag{7}$$

Denoting $\mathcal{B}$ the current minibatch of size B, the expectation involved in the risk is approximated by computing an average over the minibatch observations, leading to the cost function

$$C \triangleq \frac{1}{2B}\sum_{\mathbf{x}\in\mathcal{B}}\frac{\|\mathbf{x}-\hat{\mathbf{h}}\|_2^2}{\|\mathbf{x}\|_2^2}. \tag{8}$$

Note that this cost function evolves with time, since all minibatches are made of different observations, thus allowing real-time adaptation of mpNet to changes in the channel distribution. The network is trained to minimize the average discrepancy between its inputs and outputs, exactly as a classical autoencoder. It operates *online*, on streaming observations $\mathbf{x}_t$, $t = 1, \ldots, \infty$ of the form (3) acquired over time (coming from all users indifferently). Note that, as opposed to the classical unfolding strategies [6], [7], [8], the proposed method is totally *unsupervised*, meaning that it requires only noisy channel observations and no database of clean channels to run. Note that in all the experiments performed in this letter, we use minibatches of 200 observations and the Adam optimization algorithm [44] with an exponentially decreasing learning rate starting at 0.001 and being multiplied by 0.9 every 200 minibatchs. By abuse of notation, we denote $\texttt{Adam}(\mathbf{W}, \frac{\partial C}{\partial \mathbf{W}}, \alpha)$ the update of the weights $\mathbf{W}$ by the Adam algorithm on cost function C using the learning rate α.

Algorithm 3 Online training of mpNet

Input: Nominal dictionary $\tilde{\mathbf{E}} \in \mathbb{C}^{N\times A}$, minibatch size B, learning rate α

1: Initialize the weights: $\mathbf{W} \leftarrow \tilde{\mathbf{E}}$
2: Initialize the cost function: $C \leftarrow 0$
3: **for** $t = 1, \ldots, \infty$ **do**
4: Get observation $\mathbf{x}_t$ following (3)
5: Estimate the channel (forward pass): $\hat{\mathbf{h}}_t \leftarrow \|\mathbf{x}_t\|_2 \texttt{FW}\left(\mathbf{W}, \frac{\mathbf{x}_t}{\|\mathbf{x}_t\|_2}\right)$
6: Increment the cost function: $C \leftarrow C + \frac{1}{2B\|\mathbf{x}_t\|_2^2}\|\mathbf{x}_t - \hat{\mathbf{h}}_t\|_2^2$
7: **if** $t \mod B = 0$ **then**
8: Update the weights (backward pass): $\mathbf{W} \leftarrow \texttt{Adam}(\mathbf{W}, \frac{\partial C}{\partial \mathbf{W}}, \alpha)$
9: Reset the cost function: $C \leftarrow 0$
10: **end if**
11: **end for**

Computational complexity. Let us denote K the number of times line 3 of algorithm 2 is executed, which corresponds to the number of estimated channel paths. This number depends on the chosen stopping criterion, which is studied in section IV-C. The forward pass of mpNet costs $\mathcal{O}(KNA)$ arithmetic operations, its complexity being dominated by the multiplication of the input by the matrix $\mathbf{W}^H$ (first block of figure 2). The backpropagation step costs only $\mathcal{O}(KN)$ arithmetic operations (A times less), since the error flows through only one columns of the weight matrix $\mathbf{W}$ at each step, due to the hard thresholding operation done during the forward pass. This short complexity analysis means that jointly estimating the channel (forward pass) and learning the model (backward pass) is done at a cost that is overall the same order as the one of simply estimating the channel with a greedy algorithm (MP or OMP), without adapting the model at all to data (which corresponds to computing only the forward pass). This very light computational cost makes the method particularly well suited to online learning, as opposed to previously proposed channel estimation strategies based on deep unfolding [34], [37], [40].

C. Choosing the stopping criterion

mpNet is the unfolded version of the matching pursuit algorithm, and its number K of layers corresponds to the number of iterations of the said algorithm. It also represents the number of estimated channel paths. In fact, K is nothing else but an hyperparameter that needs to be optimized. But how can we determine this number appropriately? In the preliminary version of this study [24], this was done by testing different values of K and choosing the one yielding the best results in terms of relative error in average over channel observations, by cross-validation. In that case, the number of estimated paths was the same for every channel observation. For practical systems, this strategy is suboptimal. Indeed, the number of estimated paths should ideally depend on the SNR: the higher the SNR the more paths can be estimated reliably. The SNR depending on the distance separating the users and the base station, the path loss and gains of the different propagation paths, the depth of mpNet should be allowed to vary in order to estimate a number of path adapted to each channel observation.

In order to do so, let us take advantage of the adaptive stopping criteria proposed for greedy sparse recovery algorithms. In [45], [46], the authors show that OMP with the stopping criterion

$$\texttt{SC}_1 : \|\mathbf{r}\|_2^2 \leq \tilde{\sigma}^2(N + 2\sqrt{N\log N}), \tag{9}$$

with $\tilde{\sigma}^2 \triangleq \frac{\sigma^2}{\|\mathbf{x}\|_2^2}$ is optimal in a support recovery sense. Moreover, for a small number of iterations and an incoherent enough dictionary, MP and OMP give very close results. Hence, we propose to use $\texttt{SC}_1$ as stopping criterion for mpNet. Implementing this stopping criterion requires knowing the noise variance σ^2 or at least having an estimate $\hat{\sigma}^2$. Fortunately, the noise variance can be estimated quite reliably in MIMO systems [47]. In the sequel, we assume that a perfect noise variance estimate is available ($\hat{\sigma}^2 = \sigma^2$).

Note that practical channels $\mathbf{h}$ and learned weights $\mathbf{W}$ do not follow exactly the generative model used in [45], [46] to derive the optimal stopping criterion, so that $\texttt{SC}_1$ may lose its

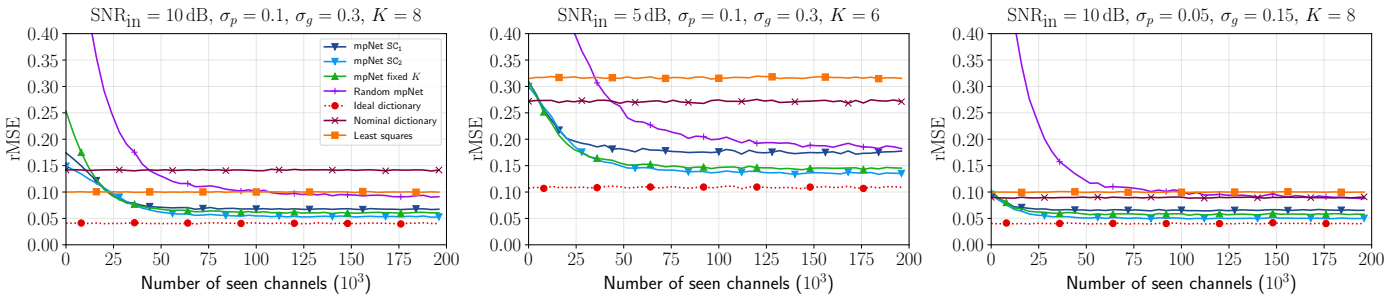

Fig. 3: Channel estimation performance on synthetic realistic channels for various SNRs and model imperfections.

optimality in the case studied here. For this reason, we propose also to use the simpler and more intuitive stopping criterion

$$\texttt{SC}_2 : \|\mathbf{r}\|_2^2 \leq \tilde{\sigma}^2 N. \tag{10}$$

From a neural network perspective, using the stopping criteria $\texttt{SC}_1$ and $\texttt{SC}_2$ means training a neural network whose depth is adaptive and dynamically adjusted during learning and inference. The structure of figure 2 is indeed replicated until the used stopping criterion is met. The two stopping criteria $\texttt{SC}_1$ and $\texttt{SC}_2$ are empirically compared on realistic synthetic channels in section V.

V. Experimental validation

Let us now assess mpNet on realistic synthetic channels. To do so, we consider the SSCM channel model [16] in order to generate non-line-of-sight (NLOS) channels at 28 GHz (see [16, table IV]) corresponding to all users. We consider the same setting as in section III, namely a base station equipped with an ULA of 64 antennas, with an half-wavelength nominal spacing and unit nominal gains used to build the imperfect nominal dictionary $\tilde{\mathbf{E}}$ (with $A = 8N$) which serves as an initialization for mpNet. The actual antenna arrays are generated the same way as in section III, using (6), and are kept fixed for the whole experiment.

A. Fixed SNR

For the first set of experiments, we consider two model imperfections: $\sigma_p = 0.05, \sigma_g = 0.15$ (small uncertainty) and $\sigma_p = 0.1, \sigma_g = 0.3$ (large uncertainty) to build the unknown ideal dictionary $\mathbf{E}$. The input SNR is fixed during the experiment and takes the values $\{5, 10\}$ dB.

We compare the performance of various configurations of mpNet, namely the version with a fixed number of iterations K, with K set to 6 for the SNR of 5 dB and to 8 for the SNR of 10 dB (determined by cross validation) and the versions with an adaptive stopping criterion, using criteria $\texttt{SC}_1$ and $\texttt{SC}_2$ described in section IV-C. In addition, the proposed method is compared to the least squares estimator and to the OMP algorithm using the stopping criterion $\texttt{SC}_2$, with either the imperfect nominal dictionary $\tilde{\mathbf{E}}$ or the unknown ideal dictionary $\mathbf{E}$. Finally, in order to show the interest of the imperfect model initialization, we compare the proposed method to mpNet using the well-known Xavier initialization [48].

Results. The results of this experiment are shown on figure 3 as a function of the number of channels of the form (3) seen by the base station over time. The performance measure is the relative mean squared error (rMSE $= \|\hat{\mathbf{h}} - \mathbf{h}\|_2^2 / \|\mathbf{h}\|_2^2$) averaged over minibatches of 200 channels. Several comments are in order: leftmargin=*

1) The imperfect model is shown to be well corrected by mpNets, the green and the two blue curves being very close to the red one (ideal unknown dictionary) after a certain amount of time. This is true both for a small uncertainty and for a large one and at all tested SNRs. Note that using the nominal dictionary (initialization of mpNet) may be even worse than the least squares method, showing the interest of correcting the model, since mpNet always ends up outperforming the least squares, thanks to the learning process.
2) Comparing the leftmost and center figures, it is interesting to notice that learning is faster and the attained performance is better with a large SNR (the green and the two blue curves get closer to the red one faster), which can be explained by the better quality of data used to train the model.
3) Comparing the leftmost and rightmost figures, it is apparent that a smaller uncertainty, which means a better initialization since the nominal dictionary is closer to the ideal unknown dictionary, leads to a faster convergence, but obviously also to a smaller improvement.
4) Looking at the purple curve on all figures, it is apparent that initialization matters. Indeed, the random initialization performs much worse than the initialization with the nominal dictionary and takes longer to converge.
5) The green and the two blue curves are all close to each other. In terms of performance, the light blue curve (corresponding to the use of $\texttt{SC}_2$ as a stopping criterion) always leads to the best performance, followed by the green curve (corresponding to a fixed depth) and finally the dark blue curve (corresponding to the use of $\texttt{SC}_1$). The central figure shows that with a lower SNR, the gap between the dark blue and the rest of the mpNet curves is more pronounced. To further understand the difference between $\texttt{SC}_1$ and $\texttt{SC}_2$, we show on figures 4 and 5 histograms of the selected depths for the experimental settings corresponding to the leftmost figure in figure 3. We see that, for $\texttt{SC}_1$, the distribution is centered at $K = 5$ which is 3 iterations behind the network with

the best results at a fixed depth. However, for $\mathtt{SC}_2$, the distribution is centered at $K = 8$ which is the optimal value for a fixed depth. Moreover, the $\mathtt{SC}_2$ version outperforming the fixed depth one is a result of its adaptability to each channel observation.

These conclusions are very promising and highlight the applicability of the proposed method.

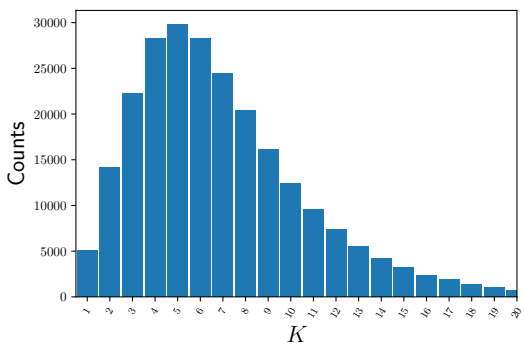

Fig. 4: Histograms of depths selected by mpNet when equipped with $\mathtt{SC}_1$.

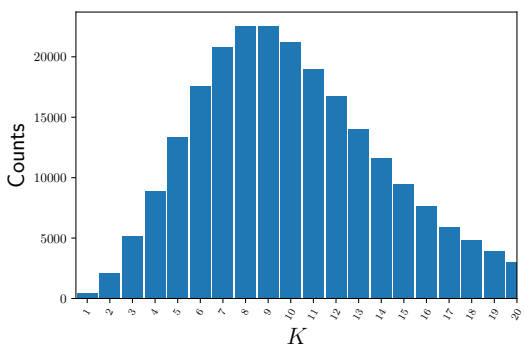

Fig. 5: Histograms of depths selected by mpNet when equipped with $\mathtt{SC}_2$.

B. Varying SNR

In practical scenarios, the SNR is not fixed, and rather depends on the distance between the transmitter and the receiver as well as propagation conditions. This variability has to be taken into account for the learning algorithm to be efficiently optimized. Using $\mathtt{SC}_2$, which takes into account both the noise level and the signal intensity, suggests that our model is capable of automatically adapting to the SNR. To verify this claim, we consider the same SSCM channel model described in the previous section, but this time, the distance between the transmitter and the receiver is sampled from a uniform distribution ranging from 60 m to 200 m. Our model is then compared to its fixed depth versions for a selected set of values of K ($\{3, 6, 8, 14\}$), as well as to the other previously presented estimation methods. The results are shown in figure 7. While all versions of mpNet are capable of learning overtime, the adaptive one is clearly outperforming the others throughout the whole experiment, starting once again at the same error level as OMP equipped with the nominal dictionary (the maroon curve). Furthermore, compared to the precedent experiment at a fixed SNR, the gap between the fixed depth and the adaptive version is more pronounced. The performance of mpNet with $\mathtt{SC}_2$ is indeed very close in terms of performance in this more realistic scenario to what is achievable with a perfect knowledge of the physical parameters (here represented by the red curve).

Finally, figure 6 shows how the SNR of the generated channel observations is distributed. Note that we choose not to consider observations with a low SNR (< 1 dB) as the model struggles to learn from very noisy instances. In practice, this could be achieved by setting a threshold on the intensity of received signals as a way to filter out useless observations. This leads to a truncated normal distribution of the SNR centered at 10 dB.

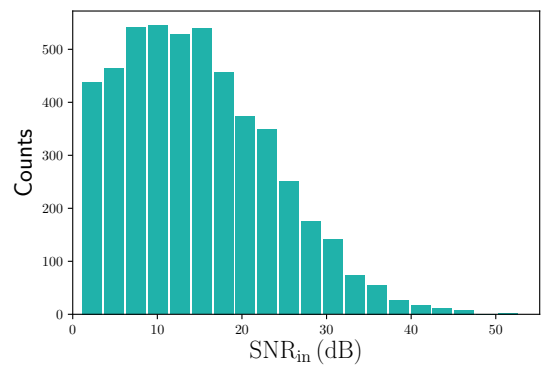

Fig. 6: SNR distribution over 5000 generated channel observations.

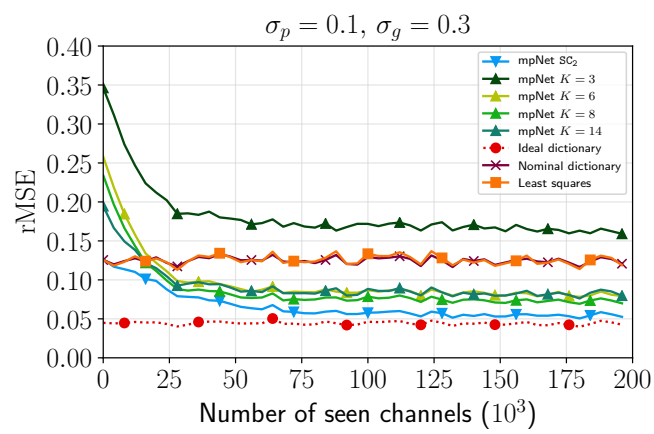

Fig. 7: Channel estimation performance on synthetic realistic channels for a varying SNR.

C. Anomaly detection and recovery

One of the benefits of online learning is the continuous adaptation of the model to incoming data. If this data is

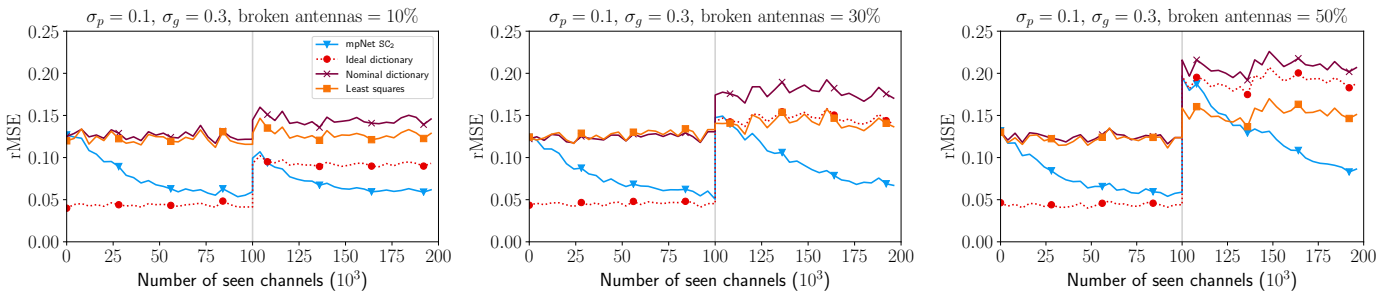

Fig. 8: Adaptation to antenna damage. The horizontal bar at the middle marks the moment at which the break happens.

disturbed, the impact would be observed on the cost function used by the system. Indeed, the distribution of the new data would be different from the one on which the system has learned so far. The speed at which the error increases would be proportional to the rate at which the distribution of the new data shifts from its original state. Furthermore, this increase in the error is simultaneously compensated by the training that is done on the network. This behavior may prove useful for detecting and recovering from anomalies. Anomalies could occur for many reasons in a massive MIMO system, including:

- Bent or broken antennas due to natural causes
- Improperly adjusted antennas after a technical intervention
- Disoriented array due to wind

In this section, we propose to test the ability of our model to detect and adapt to various types of anomalies.

Out of order antennas. Let us first simulate antennas that go out of order, in a base station equipped with 64 antennas. This is done by setting some of the antenna gains to zero at a certain point of time during training. In this case, we consider 3 scenarios: 10%, 30% and 50% of broken antennas (chosen uniformly at random). Similarly to previous experiments, the channels are generated following the SSCM model at a variable SNR and the adaptive version of mpNet with $\texttt{SC}_2$ is compared to the other estimation methods. Figure 8 shows the results. It can be seen that the training starts as usual with our model slowly approaching the performance of OMP with the optimal dictionary. The anomaly can be observed at the middle of training when the number of seen channels reaches $\sim$100000 channels. All the estimation methods see their error jump, with the exception of LS where no change is observed. The amount by which the error increases is proportional to the number of broken antennas. The error starts decreasing again for mpNet as the training resumes, but naturally stays the same for OMP based methods that do not correct the dictionary they use. By the end of the experiment, and depending on the number of broken antennas, mpNet can completely recover from the damage and its error reaches once again the level it successfully attained right before the anomaly.

The LS estimation method does not depend on any physical parameter, which explains the stable error level it displayed throughout the experiment. OMP methods, however, see their error increase because the physical parameters they are based on become less precise and thus induce a bigger error. Those methods are hence capable of detecting the exact moment where the damage happens but are incapable of adapting. On the other hand, and for the same reason, mpNet is also capable of detecting the anomaly but rapidly adapts its parameters (dictionary). In practice, the detection could be implemented via a simple threshold.

Note that the anomaly recovery only concerns the channel estimation performance. Indeed, a base station with broken antennas will always be less efficient than a fully functional one, especially in terms of channel capacity. In summary, the model does the best it can on the channel estimation task with the available means.

Aging antenna array. The second type of anomaly we consider is antenna aging. It corresponds to antenna gains slowly shifting away from their initial values. Considering antenna aging, using the initially ideal dictionary will lead to an increase of the error over time since the physical parameters are less and less precisely known. To simulate this phenomenon, we consider the same settings as for the precedent experiment. We then iteratively add noise to the antenna gains over the course of training. This is done for 10 iterations. Antenna gains at iteration t are thus expressed as

$$g_{i,t} = g_{i,t-1} + n_i,\ n_i \sim \mathcal{CN}(0, \sigma_a^2), \tag{11}$$

where σ_a could be seen as a measure of the severity (speed) of aging. We consider 3 levels of aging: $\sigma_a = 0.05$ (mild), $\sigma_a = 0.1$ (medium) and $\sigma_a = 0.2$ (severe). Again, channels are generated following the SSCM model at a variable SNR and the adaptive version of mpNet with $\texttt{SC}_2$ is compared to the other estimation methods. Results are shown on figure 9. We observe that both OMP based estimations see their error progressively increase at a rate proportional to the severity of aging, but performance worsens more rapidly when starting with an ideal dictionary. On the other hand, the neural network continues to learn and the impact of aging is barely noticeable. mpNet online learning compensates for the error induced by the continuous change of physical parameters. Finally, no noticeable change is observed on the LS estimation method.

D. From ULA to UPA

The physical model on which is based mpNet is structure-agnostic, meaning that it is meant to work with any antenna array structure. This suggests that our model, which was initially tested on ULAs, is capable of working with any structure as well. To verify this claim, we propose to adapt

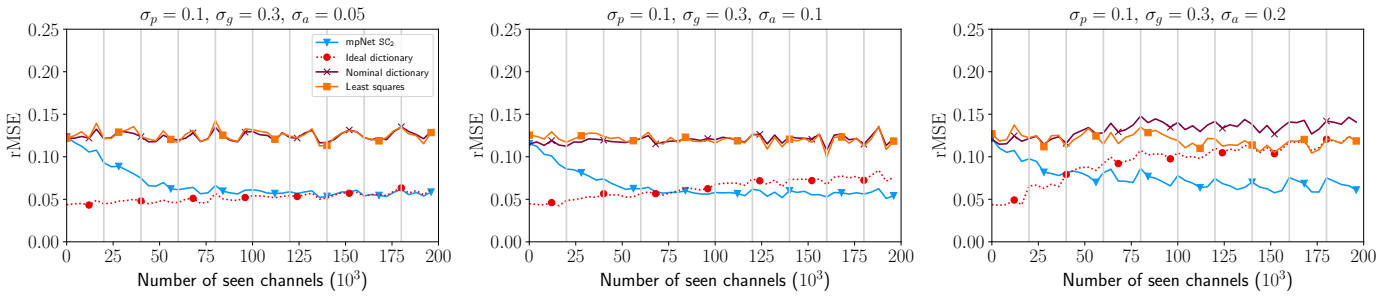

Fig. 9: Adaptation to antenna aging. The horizontal bars mark the different moments at which the aging happens.

it to uniform planar arrays (UPAs). A change in the way steering vectors are generated is required to take into account the rotational symmetry that was verified for ULAs and that no longer holds for UPAs. Therefore, the dictionary of steering vectors has to be built from DoAs sampled from the whole 3D half space, instead of a half-plane in the case of ULAs.

We conducted an experiment similar to the ones described in section V. We consider an UPA consisting of a square grid of 8×8 antennas ($N = 64$) separated by half-wave lengths and placed on the xz-plane . Channels are generated at a fixed SNR of 10 dB. To take into account the additional dimension of UPAs, ideal antenna positions are this time given by

$$\overrightarrow{a_i} = \overrightarrow{\tilde{a}_i} + \lambda \mathbf{n}_{p,i},\ \mathbf{n}_{p,i} = \left(e_{x,p,i}, \quad 0, \quad e_{z,p,i}\right)^T, \tag{12}$$

with $e_{x,p,i}, e_{z,p,i} \sim \mathcal{N}(0, \sigma_p^2)$. We compare the adaptive version of mpNet equipped with SC$_2$ to other estimation methods. The results are shown on figure 10. Once again, the model successfully learns overtime reducing its estimation error while the other classical methods maintain their performances.

This small experiment can be seen as a sanity check to show that, indeed, the model is capable of accepting any antenna structure with minimal changes in the implementation.

VI. Conclusion and Perspectives

In this paper, we introduced mpNet: a neural network allowing adding flexibility to physical models used for MIMO channel estimation. It is based on the deep unfolding strategy that views a classical algorithm (matching pursuit in this case) as a neural network, whose parameters can be trained. The proposed method was shown to correct incrementally (via online learning) an imperfect or imperfectly known physical model in order to make channel estimation as efficient as if the unknown ideal model were known. It is trained in an unsupervised manner as an autoencoder, and mpNet can be seen as a denoiser for channel observations. Training mpNet thus does not necessitate a database of clean channels, nor an offline training phase, which makes it particularly attractive for practical systems and unalike previously proposed methods.

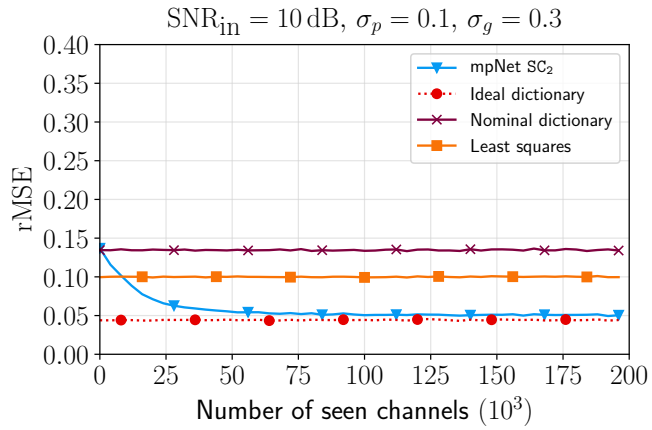

Fig. 10: Channel estimation performance on synthetic realistic channels for a base station equipped with a UPA.

We have shown that initializing the network with a dictionary of imperfect steering vectors (as opposed to using a random initialization) improves performance considerably. In the experimental part of the paper, we simulated the model imperfection by introducing uncertainties on the antenna gains and positions, but it is important to highlight the fact that the method could in principle correct many other model imperfections (such as uncertainties about the antenna diagrams, couplings between neighboring antennas, etc.).

Moreover, we introduced a stopping criterion, inspired by previous work on the OMP algorithm, to dynamically select the optimal depth of mpNet. This was shown to be particularly convenient when working on observations with a varying SNR level, which more accurately resembles real world channel observations. Evaluated on realistic synthetic data, this approach showed great results compared to other methods.

In addition, online learning enabled us to exploit the observed change in data distribution following an anomaly occurrence to detect and recover from it. We simulated two types of anomalies: antenna damage and aging. In both cases, our model was capable of efficiently recovering from the decrease in performance overtime.

Finally, we proved that our model is capable of adapting to any antenna structure with no apparent drop in performance or change in behavior. In particular, we showed that a simple change in the steering vector generation process was required to adapt the model, previously based on a ULA structure, to a UPA structure.

In future work, we could explore the unfolding of more sophisticated sparse recovery algorithms (such as iterative soft thresholding [49] or approximate message passing [50]) using the same strategy. In addition, other stopping criteria could also be integrated to the model and tested. Finally, the deep

unfolding of the matching pursuit algorithm initialized with a dictionary based on an imperfect model is by no means limited to the MIMO channel estimation task and could be exploited for other tasks, as long as an initial model is available.

REFERENCES

[1] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[2] Gabriel Peyré. Manifold models for signals and images. *Computer Vision and Image Understanding*, 113(2):249–260, 2009.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[4] Michael Elad. *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010.

[5] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[6] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 399–406. Omnipress, 2010.

[7] John R Hershey, Jonathan Le Roux, and Felix Weninger. Deep unfolding: Model-based inspiration of novel deep architectures. *arXiv preprint arXiv:1409.2574*, 2014.

[8] Ulugbek S Kamilov and Hassan Mansour. Learning optimal nonlinearities for iterative thresholding algorithms. *IEEE Signal Processing Letters*, 23(5):747–751, 2016.

[9] Vishal Monga, Yuelong Li, and Yonina C Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *arXiv preprint arXiv:1912.10557*, 2019.

[10] Fredrik Rusek, Daniel Persson, Buon Kiong Lau, Erik G Larsson, Thomas L Marzetta, Ove Edfors, and Fredrik Tufvesson. Scaling up mimo: Opportunities and challenges with very large arrays. *IEEE Signal Processing Magazine*, 30(1):40–60, 2013.

[11] Erik G Larsson, Ove Edfors, Fredrik Tufvesson, and Thomas L Marzetta. Massive mimo for next generation wireless systems. *IEEE Communications Magazine*, 52(2):186–195, 2014.

[12] Lu Lu, Geoffrey Ye Li, A Lee Swindlehurst, Alexei Ashikhmin, and Rui Zhang. An overview of massive mimo: Benefits and challenges. *IEEE journal of selected topics in signal processing*, 8(5):742–758, 2014.

[13] Theodore S Rappaport, Shu Sun, Rimma Mayzus, Hang Zhao, Yaniv Azar, Kevin Wang, George N Wong, Jocelyn K Schulz, Mathew Samimi, and Felix Gutierrez. Millimeter wave mobile communications for 5g cellular: It will work! *IEEE access*, 1:335–349, 2013.

[14] A Lee Swindlehurst, Ender Ayanoglu, Payam Heydari, and Filippo Capolino. Millimeter-wave massive mimo: the next wireless revolution? *IEEE Communications Magazine*, 52(9):56–62, 2014.

[15] Robert W Heath, Nuria Gonzalez-Prelcic, Sundeep Rangan, Wonil Roh, and Akbar M Sayeed. An overview of signal processing techniques for millimeter wave mimo systems. *IEEE journal of selected topics in signal processing*, 10(3):436–453, 2016.

[16] Mathew K Samimi and Theodore S Rappaport. 3-d millimeter-wave statistical channel model for 5g wireless system design. *IEEE Transactions on Microwave Theory and Techniques*, 64(7):2207–2225, 2016.

[17] Akbar M Sayeed. Deconstructing multiantenna fading channels. *IEEE Transactions on Signal Processing*, 50(10):2563–2579, 2002.

[18] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak. Compressed channel sensing: A new approach to estimating sparse multipath channels. *Proceedings of the IEEE*, 98(6):1058–1076, June 2010.

[19] Luc Le Magoarou and Stéphane Paquelet. Parametric channel estimation for massive MIMO. In *IEEE Statistical Signal Processing Workshop (SSP)*, 2018.

[20] Luc Le Magoarou, Antoine Le Calvez, and Stéphane Paquelet. Massive mimo channel estimation taking into account spherical waves. In *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5. IEEE, 2019.

[21] S.G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12):3397–3415, Dec 1993.

[22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning Internal Representations by Error Propagation*, page 318362. MIT Press, Cambridge, MA, USA, 1986.

[23] Léon Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142, 1998.

[24] Luc Le Magoarou and Stphane Paquelet. Online unsupervised deep unfolding for massive mimo channel estimation, 2020.

[25] Timothy OShea and Jakob Hoydis. An introduction to deep learning for the physical layer. *IEEE Transactions on Cognitive Communications and Networking*, 3(4):563–575, 2017.

[26] Tianqi Wang, Chao-Kai Wen, Hanqing Wang, Feifei Gao, Tao Jiang, and Shi Jin. Deep learning for wireless physical layer: Opportunities and challenges. *China Communications*, 14(11):92–111, 2017.

[27] Zhijin Qin, Hao Ye, Geoffrey Ye Li, and Biing-Hwang Fred Juang. Deep learning in physical layer communications. *IEEE Wireless Communications*, 26(2):93–99, 2019.

[28] Hengtao He, Shi Jin, Chao-Kai Wen, Feifei Gao, Geoffrey Ye Li, and Zongben Xu. Model-driven deep learning for physical layer communications. *IEEE Wireless Communications*, 26(5):77–83, 2019.

[29] Yacong Ding and Bhaskar D Rao. Dictionary learning-based sparse channel representation and estimation for fdd massive mimo systems. *IEEE Transactions on Wireless Communications*, 17(8):5437–5451, 2018.

[30] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, Nov 2006.

[31] Alexios Balatsoukas-Stimming and Christoph Studer. Deep unfolding for communications systems: A survey and some new directions. *arXiv preprint arXiv:1906.05774*, 2019.

[32] Neev Samuel, Tzvi Diskin, and Ami Wiesel. Deep mimo detection. In *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5. IEEE, 2017.

[33] Hengtao He, Chao-Kai Wen, Shi Jin, and Geoffrey Ye Li. A model-driven deep learning network for mimo detection. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 584–588. IEEE, 2018.

[34] Hengtao He, Chao-Kai Wen, Shi Jin, and Geoffrey Ye Li. Deep learning-based channel estimation for beamspace mmwave massive mimo systems. *IEEE Wireless Communications Letters*, 7(5):852–855, 2018.

[35] Christopher A Metzler, Arian Maleki, and Richard G Baraniuk. From denoising to compressed sensing. *IEEE Transactions on Information Theory*, 62(9):5117–5144, 2016.

[36] Chris Metzler, Ali Mousavi, and Richard Baraniuk. Learned d-amp: Principled neural network based compressive image recovery. In *Advances in Neural Information Processing Systems*, pages 1772–1783, 2017.

[37] Xiuhong Wei, Chen Hu, and Linglong Dai. Knowledge-aided deep learning for beamspace channel estimation in millimeter-wave massive mimo systems. *arXiv preprint arXiv:1910.12455*, 2019.

[38] Xuanxuan Gao, Shi Jin, Chao-Kai Wen, and Geoffrey Ye Li. Comnet: Combination of deep learning and expert knowledge in ofdm receivers. *IEEE Communications Letters*, 22(12):2627–2630, 2018.

[39] Mehran Soltani, Vahid Pourahmadi, Ali Mirzaei, and Hamid Sheikhzadeh. Deep learning-based channel estimation. *IEEE Communications Letters*, 23(4):652–655, 2019.

[40] Michel van Lier, Alexios Balatsoukas-Stimming, Henk Corporaaal, and Zoran Zivkovic. Optcomnet: Optimized neural networks for low-complexity channel estimation. *arXiv preprint arXiv:2002.10493*, 2020.

[41] J.A. Tropp and S.J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, June 2010.

[42] George EP Box. Science and statistics. *Journal of the American Statistical Association*, 71(356):791–799, 1976.

[43] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[44] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[45] T Tony Cai and Lie Wang. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information theory*, 57(7):4680–4688, 2011.

[46] Rui Wu, Wei Huang, and Di-Rong Chen. The exact support recovery of sparse signals with noise via orthogonal matching pursuit. *IEEE signal processing letters*, 20(4):403–406, 2012.

[47] Aniruddha Das and Bhaskar D. Rao. Snr and noise variance estimation for mimo systems. *IEEE Transactions on Signal Processing*, 60:3929–3941, 2012.

[48] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[49] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–1457, 2004.

[50] David L Donoho, Arian Maleki, and Andrea Montanari. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, 2009.