RoemNet: Robust Meta Learning based Channel Estimation in OFDM Systems

Hengxi Mao, Hancheng Lu, Yujiao Lu, Daren Zhu
University of Science and Technology of China, Hefei, China, 230027
Email: mhx6056@mail.ustc.edu.cn, hclu@ustc.edu.cn,
lyj66@mail.ustc.edu.cn, darenzhu@mail.ustc.edu.cn

***Abstract*—Recently, in order to achieve performance improvement in scenarios where the channel is either unknown, or too complex for an analytical description, Neural Network (NN) based channel estimation is introduced in Orthogonal Frequency Division Multiplexing (OFDM) systems. However, this kind of learning method is not reliable enough when the conditions of online deployment of the common NNs are not consistent with the channel models used in the training stage. Furthermore, common NNs need plenty of data as well as time to be trained, which is not suitable for the OFDM communication network with time varying channels. To tackle these challenges, we propose a novel meta learning based channel estimation approach called RoemNet. The most distinctive characteristic of RoemNet is that it involves a meta-learner that can learn from the environment of different channels. With the update of meta-learner, RoemNet is robust enough to solve new channel learning tasks using only a small number of pilots. Furthermore, RoemNet can alleviate the effect of Doppler spread and significantly improve the Bit Error Ratio (BER) performance under different channel environments. Experiment results demonstrate that the proposed RoemNet outperforms existing channel estimation methods including existing learning methods under various scenarios.**

***Index Terms*—Channel Estimation, Meta Learning, Orthogonal Frequency Division Multiplexing, Wireless Multipath Fading Channel**

I. INTRODUCTION

In the past few decades, Orthogonal Frequency Division Multiplexing (OFDM), as an amenable technology with excellent anti-multipath fading ability and spectrum utilization, has attracted extensive attention and research. However, since OFDM is sensitive to frequency offset and phase noise generated by the channel [1], the Channel State Information (CSI) is necessary to be tracked and estimated for coherent detection and decoding. The accuracy and efficiency of channel estimation method will directly affect the transmission performance of the entire OFDM communication system [2].

Historically, designing channel estimators with both low complexity and good channel tracking ability has always been an important issue. Some traditional pilot symbol assisted estimation approaches, e.g., Least Squares (LS) and Minimum Mean-Square Error (MMSE), have been widely used to estimate the channel properties and correct the received signal [3] [4]. Among them, LS is the simplest. It doesn't require priori information of the channel. However, it has a high mean square error rate and is easy to be affected by noise and Inter-Carrier Interference (ICI) [3], so that it performs badly in terms of accuracy. The ideal MMSE algorithm yields much better performance than LS estimators using the second-order statistics of the channel, but it suffers high computational complexity. Actually, most of the practical applications are modified LS or simplified MMSE based algorithms, e.g., Linear Minimum Mean-squared Error (LMMSE) [5] and low-rank MMSE [6], making a compromise between complexity and Signal-to-Noise Ratio (SNR) performance [7].

Recently, deep learning has been proved to be a promising way to be applied in communication systems in an end-to-end manner [8]–[12]. As far as we know, there also exist a few deep learning based approaches for channel estimation [13] [14]. Authors in [13] propose an OFDM-Autoencoder based on the full connected Deep Neural Network (DNN), which could be embedded into an OFDM with Cyclic Prefix (CP) system. In this architecture, channel estimation is modeled as a classification problem, the NN is trained to work with exsting MMSE equalizer. Regrettably, this deep learning based approach will bring two main problems. Firstly, the proposed classification network in [13] can only imitate Quadrature Phase Shift Keying (QPSK). This is due to that the Autoencoder is essentially a 4-way classification network. When the classification categories grows, so does the network complexity, and the performance will not be as good as that using QPSK. Secondly, an MMSE equalizer is required following the Autoencoder to offer subsequent correction, which means such an estimator can not achieve better performance than MMSE estimators in a full OFDM system.

Different from [13], the deep learning estimator in [14] is trained with the received OFDM samples. The samples are generated with various information sequences and under diverse channel conditions with certain statistical properties [14]. Thus, this approach can work better than traditional MMSE estimators under certain circumstances. However, every 16 bits of the transmitted data in [14] has to be estimated based on a single model that is trained independently. Since every NN model needs a large number of training data together with the burden of a long training period to be fully trained [14], this deep learning based approach is still not suitable for practical transmission in wireless multipath fading environments varying over time.

To tackle the issues above, we propose a two-stage estimation approach called Robust channel Estimation with Meta neural Networks (RoemNet) to estimate the CSI and recover

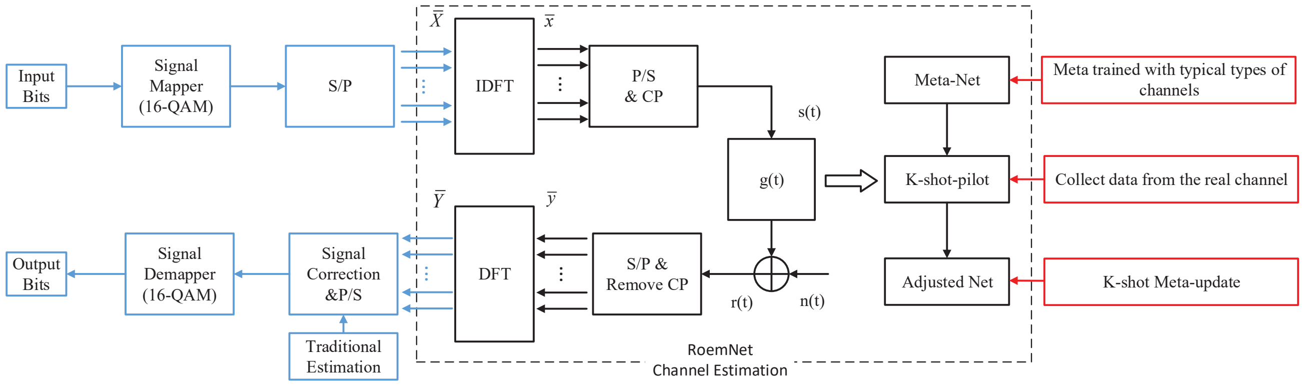

Fig. 1. Baseband OFDM system with Channel Estimation and RoemNet process

the transmitted symbols in OFDM systems in wireless multi-path fading channels. With meta learning (or learning-to-learn) [15] [16], RoemNet is able to acquire many skills and adapt to many environments. Therefore, it can make a quick solution to new learning tasks using only a small number of training samples. Unlike traditional learning methods that learn an end-to-end mapping with trained weights, meta-net is trained to learn meta knowledge, i.e., various basic parameters during neural network training, such as initialization of parameters [17], the choice of optimizers [18], even the structure of models [19], etc.

In RoemNet, we consider multiple channels as multiple tasks, and we train a meta-learner that can learn the general characteristics of these tasks to get a good initialization for the network. Then the meta-learner will be used to guide the adjustment of a new network for its deployment in unknown channels, where few pilot symbols are available for fine-tuning. This process is called meta-update. The prior guidance together with meta-update make RoemNet possible to be robust enough to achieve rapid convergence with few pilots.

The main achievements in RoemNet are summarized as follows. Firstly, RoemNet introduces meta learning for channel estimation, taking an integrated OFDM system into consideration. Secondly, in RoemNet, better Bit Error Rate(BER) and SNR performance can be achieved compared to the method in [14] because of the meta-training process. Thirdly, RoemNet is designed to alleviate the weakness of common NNs in channel estimation, which means that RoemNet can work well with less pilot data and use less time to achieve convergence. Furthermore, using pilots for fine-tuning at transmission stage, the meta learning-based RoemNet is capable of dealing with the situations that online deployment are not consistent with the channel models used in the training stage (e.g., when untrained Doppler frequency exists). From an implementation perspective, this is of great significance for practical transmission.

The rest of the paper is organized as follows. The proposed approach RoemNet is presented in Section II. Section III details the architecture and learning process of meta-net. In Section IV we present and discuss the experiment results. The conclusions of our work are drawn in Section V.

II. ROEMNET CHANNEL ESTIMATION

Fig.1 shows the baseband OFDM system with RoemNet channel estimation module embedded. RoemNet is designed to work with existing baseband OFDM system. However, it takes the place of traditional estimation methods to get better performance and to reduce the computational complexity when estimating CSI for each block. Compared to existing learning based estimation methods, RoemNet has a sense of generalization ability which means it can be adaptable to the untrained channel quickly. The baseband OFDM along with RoemNet are described in detail as follows.

A. Baseband Model of OFDM System

The baseband OFDM system is the same as the conventional ones. On the transmitter side, the binary information is first encoded into multiamplitude-multiphase signals. After converted to the paralleled data stream (already with pilots inserted), the Inverse Discrete-time Fourier Transform (IDFT) unit is used to modulate them on different subcarriers. Following the IDFT unit, a cyclic extension of time length, or CP is inserted to mitigate the Inter-Symbol Interference (ISI). Note that the length of the CP should be no shorter than the maximum delay spread of the channel.

We choose a multipath fading channel. The channel is modeled as an impulse response $g(t)$ followed by the complex Additive White Gaussian Noise (AWGN) $n(t)$

$$g(t,\tau) = \sum_{i=1}^{M} h_i\, \delta(\tau - \tau_i) e^{j2\pi\, f_{Di} t} \tag{1}$$

where M denotes the number of fading-paths, $\{h_i\}$ denotes the set of complex path gains, τ_i is the path delay for the i-th path, f_{Di} is the i-th path Doppler frequency shift which causes ICI.

At the receiver side, the pilot-based signal correction is performed after DFT. Next, the data is passed to the signal demapper unit for the demodulation and channel decoding.

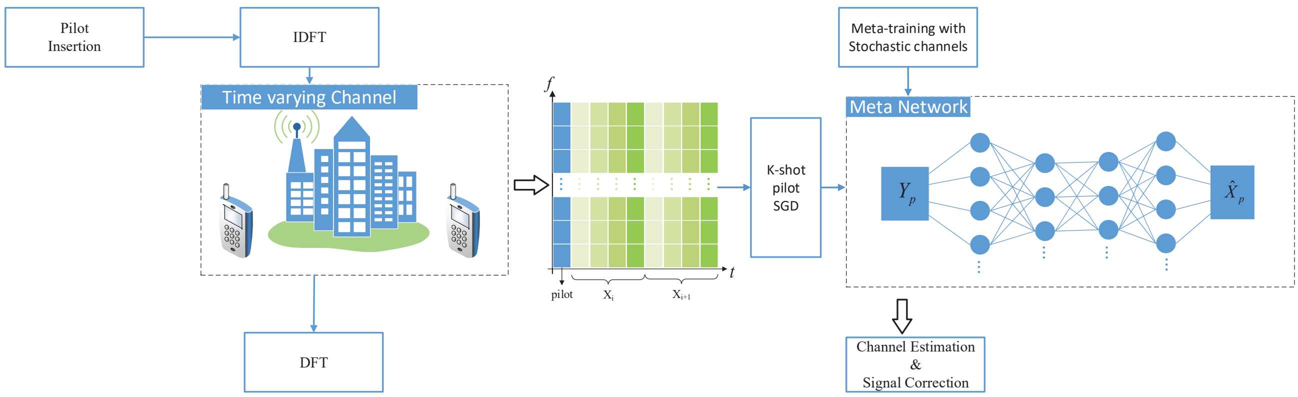

Fig. 2. RoemNet Architecture

Lastly, the binary information data is obtained back. Let $\overline{X}$ and $\overline{Y}$ denote the input data of IDFT unit at the transmitter and the output data of DFT unit at the receiver, respectively. Let $\overline{g}$ and $\overline{n}$ denote the sampled channel impulse response and AWGN, respectively. Then $\overline{Y}$ can be expressed as

$$\overline{Y} = DFT_N(IDFT_N(\overline{X}) \otimes \overline{g} + \overline{n}) \tag{2}$$

Define the input matrix $\mathbf{X} = diag(\overline{X})$. Assuming that the interferences are completely eliminated, we can derive

$$\overline{Y} = \mathbf{XF}\overline{g} + \overline{N} = \mathbf{X}\overline{H} + \overline{N} \tag{3}$$

where $\mathbf{F}$ is the DFT-matrix defined in [2].

Eq.(3) demonstrates that an OFDM system is equivalent to a transmission of data over a set of parallel channels, which makes it possible to train a NN to imitate the channels so that we can recover the transmitted $\overline{X}$ in an end-to-end manner.

B. RoemNet Architecture

Machine learning based signal processing and communication systems are believed to hold the potential to improve some of existing communication algorithms in terms of reliability, generality, latency, and energy efficiency [20] [21]. However, since hardly any of previous applications focus on a full end-to-end system, it is challenging to involve estimation in an integrated OFDM system with learning methods, and there are two main problems to be considered beforehand.

Firstly, the channel is time varying, which means it is not effective if we simply train a NN and then employ it for continuous data transmission. This is because the mapping learned during training process may change along with time, depending on the variation of channel (e.g. with Doppler spread or nonlinear clipping noise [22]). A compromising solution is to assume the estimated channel stays unchanged until the next sequence of pilots is received, which has actually been adopted by IEEE 802.11a/b/g. However, another problem arises: since the time between interval transmission of blocks is too short to train a new NN, can we find a neural model that is able to make fast adaptation to changes over the channel, while retaining prior experience during training period?

Besides, there is one more problem: the number of pilots can be used to train NN for adaption is limited in each OFDM block [23] (e.g. frequency domain pilots will reduce the spectral efficiency). Assuming that there are K pilots in one block, then NN should correct all the remaining symbols with the aid of K shot pilots, which is similar to few shot learning problems in the domain of Machine Learning [24]. In the past two years, meta-learning has been rediscovered as an amenable approach for learning from small amounts of data [17] [25]. Recent breakthroughs are achieved in continuous learning under nonstationary conditions [26], taking a solid step towards artificial general intelligence. To sum up, a model with generalization ability is required to adapt to changes over multipath fading channel of OFDM with K shot pilots.

To this end, we introduce RoemNet and formulate the channel estimation problem as a K-shot recovery problem with meta learning. As shown in Fig.2, the working procedure of RoemNet is divided into three parts. First of all, we train a meta-net to learn a general character of typical types of channels. Its parameters are not initialized randomly with traditional supervised learning method, but learned through what we called meta-learner, which will be described in detail in Section III. Secondly, frames are sent and received from an OFDM system over the time varying channel. Each block in one frame contains two parts, the pilots and the information. The received and transmitted pilot data of one block is collected and saved as the input and output of the meta net, respectively, for further adjustment. Lastly, RoemNet uses the collected data to update the parameters and fine-tune the previous network by performing certain steps of Stochastic Gradient Descent (SGD). This step can be done in a quite short time because we have initialized the net with weights that are easy to converge. Then we get a network which is suitable for the transmission of this block, where the channel can be treated as constant spanning over the pilot and the data blocks, but changes from one block to another. Given proper initialization and fine-tuning, RoemNet is more suitable for transmission over a time varying channel than the method in [14], which needs to train a new network for the transmission of a new block if the channel condition have changed.

In RoemNet, the training data of typical channels is acquired as multiple tasks to learn the general characteristics. When facing an unknown channel, RoemNet can use the K pilots to fine-tune the network through K-shot SGD. The core of RoemNet is to train the initialization parameters, and update them through several gradient steps with the help of a small amount of data from new tasks, so that the adaptation happens in the right space for fast learning on new channels.

III. LEARNING PROCESS OF ROEMNET

This section describes the detailed architecture of DNN used in RoemNet and illustrates how the estimation approach learns in RoemNet.

A. Training the RoemNet

Algorithm 1 Meta Training of RoemNet

Input: $\{\boldsymbol{T}_i\}_{i=1}^{N_1}$: Tasks of training channels; $\boldsymbol{N}_2$: Number of training slices in one channel; $\boldsymbol{\alpha}_1$ $\boldsymbol{\alpha}_2$: step size hyper-parameters

Output: $\boldsymbol{h}_{\boldsymbol{\phi}}$: Optimal model with parameters of $\boldsymbol{\phi}$

- randomly initialize $\boldsymbol{H}_{\boldsymbol{\Phi}}$ with parameters of $\boldsymbol{\Phi}$
- **repeat**
  - **for** $i = 1, 2....N_1$ **do**
    - Sample $2N_2$ slices of K_1 DFT-output $\overline{Y}$ along with corresponding $\overline{X}$
    - Divide the $2N_2$ slices into B_1 B_2
    - **for** $j = 1, 2....N_2$ **do**
      - Obtain $\nabla_{\phi_i^j} L(h_{\Phi_i})$ using B_1 based on Eq. (6)
      - Evaluate adapted parameters ϕ_i^j based on Eq. (5)
    - **end for**
    - Update ϕ_i using $\boldsymbol{\alpha}_2$ B_2 based on Eq. (7)
  - **end for**
  - Update $\boldsymbol{h}_{\boldsymbol{\phi}}$ with $\boldsymbol{\alpha}_1$ based on Eq. (7)
- **until** Convergence
- **return** $\boldsymbol{h}_{\boldsymbol{\phi}}$

During the meta-training process, we set the received raw data as the input of RoemNet, and the output is set as the estimated $\hat{\overline{X}}$ to imitate original transmission sequence labels $\overline{X}$. The frame structure is shown in Fig.3. Let N_1 denote the number of different channels for the meta-training, which means N_1 different tasks. Note that the tasks should subject to the same distribution $\boldsymbol{p(T)}$. The task set is then defined as $\{T_1, T_2, ..., T_i, ..., T_{N_1}\}$. For each task, let N_2 denote the number of slices in training a single channel. Here we apply Mean Square Error (MSE) as the loss function

$$L(h_{\phi_i}) = \sum_{\overline{X}^{(j)}\overline{Y}^{(j)} \sim T_i} \| h_{\phi_i^j}(\overline{Y}_i^{(j)}) - \overline{X}_i^{(j)} \|_2^2 \tag{4}$$

where L is the loss function, h is the model, ϕ denotes the value of weights.

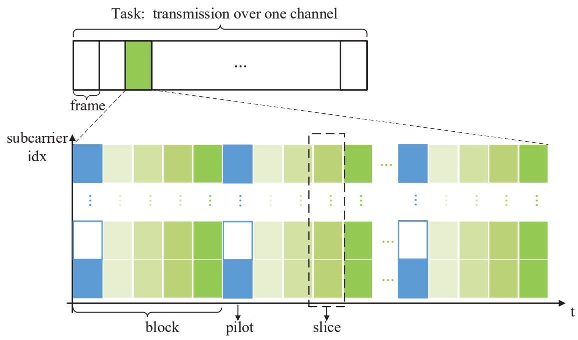

Fig. 3. Frame structure in RoemNet

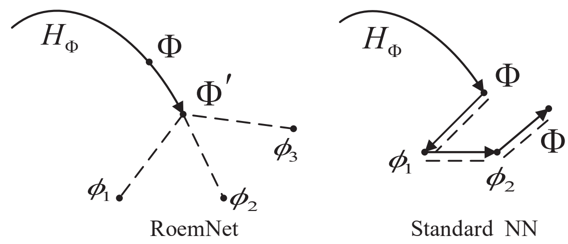

Fig. 4. Different methods of updating the model's parameters in RoemNet and standard NN

Let K_1 denote the number of $\overline{Y}$ and its corresponding labels $\overline{X}$. To get a thorough training of typical channels, K_1 should be no less than the number of sub-carriers of OFDM. In the j-th training slice of each task T_i, the primary parameter ϕ_i^j is updated through

$$\phi_i^j = \Phi_i - \nabla_{\phi_i^j} L(h_{\Phi_i}) \tag{5}$$

where Φ_i is the i-th updated parameters, and the gradient of loss in Eq. (5) for the j-th training slice is obtained through

$$\nabla_{\phi_i^j} L(h_{\Phi_i}) = \frac{\partial L(h_{\Phi_i})}{\partial \phi_i^j} = -\frac{1}{K_1} \sum_{m=1}^{K_1} (\overline{X}^m - h_{\phi_i^j}(\overline{Y}^m)) Y_j^m \tag{6}$$

After all iterations of N_2 slices or N_1 tasks have been traversed, the parameter ϕ is updated through

$$\phi = \Phi - \alpha \nabla_\phi \sum L(h_\Phi) \tag{7}$$

where α is the step size hyper-parameter used in RoemNet.

The update of parameters is not done like common NN with standard supervised learning method, but with all the rules learned from the priori channels, just like priori guidance. This process is called meta-learner. As shown in Fig.4, during training, the standard supervised learning based NN updates the parameters every time in which there is a gradient descent. Differently, RoemNet firstly calculate all the gradients of loss function using Eq. (6). Next with Eq. (7), RoemNet can find a group of appropriate initial parameters that has the best generalization ability for different channel tasks in the parameter space.

Algorithm 2 Meta Update for Time Varying Channel

Input: $\boldsymbol{h}_{\phi}$: Meta trained model with parameters of ϕ; $\boldsymbol{P} = \{\overline{\boldsymbol{Y}}_{\boldsymbol{p}}^{\boldsymbol{i}}, \overline{\boldsymbol{X}}_{\boldsymbol{p}}^{\boldsymbol{i}}\}_{i=1}$: A stream of K_2-shot pilot pairs; $\{\overline{\boldsymbol{Y}}\}_{i=1}$: A stream of output data of DFT block at the receiver

Output: $\{\overline{\boldsymbol{X}}\}_{i=1}$: Estimated stream of input data of IDFT block at the transmitter

1: **while** there are new incoming data **do**
2: Update $\boldsymbol{\phi_i}$ using $\boldsymbol{P}$ based on Eq. (8)
3: Fine-tune $\boldsymbol{h}_{\boldsymbol{\phi_i}}$ for the input of $\overline{\boldsymbol{Y}}_i$
4: Obtain $\overline{\boldsymbol{X}}_i = \boldsymbol{h}_{\boldsymbol{\phi_i}}\overline{\boldsymbol{Y}}_i$
5: **end while**

When faced with blocks for testing, there is an update for meta-learner to make RoemNet adapted to time varying channels. Given that there are only K_2 known pilot signals in one block, we set K_2 pilots as one group. Different from training period, since OFDM communication system calls for real-time transmission, we use SGD for meta-update, which requires relatively few computing

$$\phi_i(T_i) = \Phi + (\overline{X}_p - h_{\phi_i}(\overline{Y}_p))\overline{X}_{p_i} \tag{8}$$

The details for training and deployment of RoemNet is described in Algorithm I and Algorithm II, respectively.

B. DNN

For initial evaluation, we employ a DNN model in RoemNet. The DNN consists of five layers, three of which are hidden layers of size 64,128,64. To accelerate the converging procedure, we apply the Rectified Linear Unit (ReLU) as activation function in these three hidden layers. The general activation function of ReLU is defined as

$$ReLU(S_i) = max(S_i, 0) \tag{9}$$

where S_i is the input signal of the activation on the i-th layer.

When training with the DNN, we set K_1 = 64 with an initial step size $\alpha_1 = 0.003$ and a fixed step size $\alpha_2 = 0.001$, and use Adam as the meta-optimizer.

It should be noted that we use the group of received complex values of slices $\overline{Y}$ and their corresponding ground truth sent symbols $\overline{X}$ as training pairs. However, none of the previously described machine learning libraries currently support operating on complex values. Therefore, the 2n-complex-valued vector of $\overline{Y}$ is cast to 2n-real numbers by considering one half as the real part and the other half as the imaginary part (see [21]), thus the input number corresponds to the number of real parts plus the number of imaginary parts of pilots in OFDM blocks.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance enhanced by RoemNet in an OFDM system. Several experiments are performed to compare it with other channel estimation approaches including DNN in [14] under various scenarios. Implementation is firstly presented, then results are given and discussed. The values of parameters of the system are presented in TABLE I.

TABLE I
SUMMARY OF SYSTEM PARAMETERS

Parameters	Values
Modulation scheme	16QAM
Number of subcarriers N_s	64
DFT size N_d	512
Size cyclic prefix L_{CP}	$8T_s$
Number of pilots N_p	16
Carrier frequency f_c	1 GHz
Bandwidth B_w	2 MHz
Frame size B	10 blocks
Number of symbols in each block N_s	10
Training SNR E_b/N_0	5:5:30 dB
Meta-Training Doppler frequency range f_d	[0,50] Hz
Meta-Training fading-paths of channels M	[1,16]
Meta-Training maximum delay spread range τ_m	[0.2,8.0) μs
Meta-Training amounts of channels N_1	100
Meta-Training amounts of slices in one channel N_2	100

TABLE II
PARAMETERS OF CHANNEL FOR TESTING

Path Number	Average Path Gain(dB)	Delay(μs)
1	-3.0	0.0
2	0.0	0.2
3	-2.0	0.5
4	-6.0	1.6
5	-8.0	2.3
6	-10.0	5.0

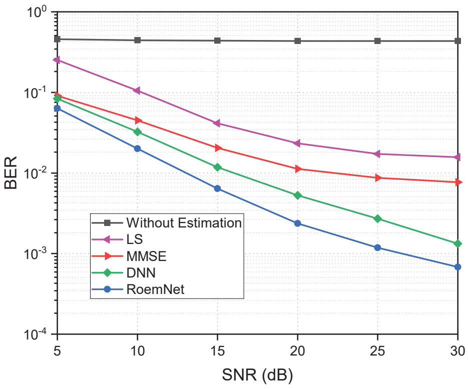

Fig. 5. BER of channel estimation using different approaches under different SNR in slow Rayleigh fading channel

A. Implementation

For the dataset, with these channel models under different parameters in Eq. (1), the training data can be obtained by simulation. In each simulation, random channel parameters are first generated, then several frames of random data sequence are generated as the transmitted symbol labels and the corresponding OFDM blocks of these frames are formed with a sequence of pilot symbols. The OFDM signal is received in the form of OFDM frames undergoing the current channel

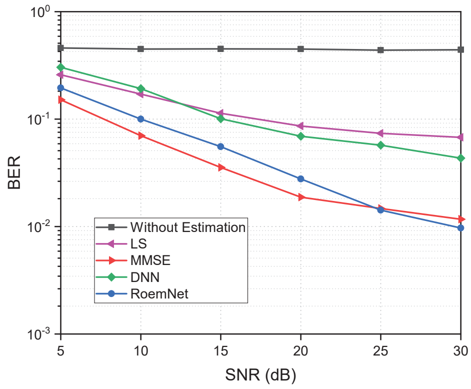

Fig. 6. BER of channel estimation using different approaches under different SNR with untrained Doppler frequency

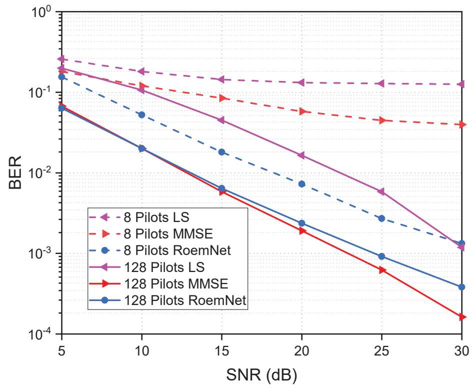

Fig. 7. BER curves with different numbers of pilots

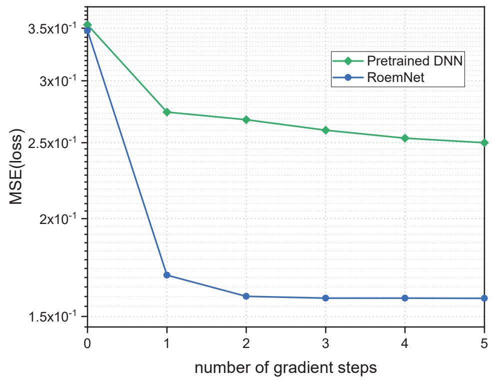

Fig. 8. MSE under 5dB with different gradient steps during fine-tuning

distortion. The received signal and the original transmitted data are collected as the training data.

We compare RoemNet with other three approaches, including:

(1) LS estimator. A low complexity algorithm without using any knowledge of the statistics of the channels but also with no consideration of noise and ICI.

(2) MMSE estimator. One of the state-of-the-art approaches that employs the second-order statistics of the channel conditions to minimize the MSE.

(3) DNN in [14]. Pre-trained on all channels with traditional supervised learning method. At test-time, the DNN is fine-tuned on the same K_2 provided pilots as RoemNet.

B. Results

1) Comparisons in slow Rayleigh fading channel: Fig.5 plots the result of BER curves of four channel estimation methods under different SNR. The slow Rayleigh fading channel used for testing has not been trained in advance for the two learning based methods, with parameter shown in TABLE II. The Doppler frequency f_D is set to 5Hz to imitate the indoor and pedestrian channels, which is within the range of training set. The pilot number is set to 16. It can be recognized from Fig.5 that, with meta-training and meta-update, RoemNet yields the best performance among all methods. The DNN proposed in [14] achieves the inferior performance. Traditional approaches, i.e., LS, MMSE, cannot achieve optimal performance, because the impact of channel on the distortion is nonlinear and difficult to be analyzed and formulated. Differently, RoemNet uses a deep learning network, making it insensitive to wireless multipath fading channels. We also notice that the performance of DNN and RoemNet continue to make an enhancement, while the higher SNR only improves the performance little with traditional approaches. This is due to the fact that during the training process, the data collected under low SNR will mislead the training process.

2) Comparisons in fast Rayleigh fading channel: To further verify the robustness and the estimation ability of RoemNet, we evaluate and compare all approaches in fast Rayleigh fading channel with 150Hz Doppler frequency shift. As can be seen from Fig.6, since the maximum Doppler spread is increased, traditional methods are strongly affected by ICI. As for learning based approaches, the untrained Doppler frequency shift will cause rotation of the QAM constellation (see [3]), which is extremely different from the mapping learned during training period in [14]. Luckily, RoemNet is robust enough to adapt to new channel conditions depending on pilots for meta-update. The result in Fig.6, which validates our analysis, shows that RoemNet can achieve performance comparable to MMSE under high SNR.

3) Impact of the number of pilots: Since inserting pilots will incur energy diversion, in pilot-aided OFDM system, the optimal pilot estimation scheme should use the least number of pilots. From Fig.7, when the number of pilots per block is set to 8 (adopting the same block-type pilot scheme in slow Rayleigh fading channel), there is clearly a 10 dB degradation in performance of traditional methods. However, RoemNet still has the ability to reduce its BER to 10^{-3} with increasing SNR, which demonstrates that the RoemNet is robust to the number of pilots needed for estimation.

4) Impact of different learning approaches: To account for the distinct performance of meta-learning based NN compare with common DNN, we evaluate the MSE under 5dB with

150Hz Doppler shift during fine-tuning in Fig.8. Each gradient step in the horizontal axis is an iteration in Eq. (8), which is computed using the same K pilots. The DNN used for comparison is trained on the same dataset as RoemNet, but with traditional supervised learning method. We observe that RoemNet achieves obvious improvement than the pre-trained DNN in terms of MSE in the first gradient step. The DNN, on the contrary, can not learn the new characteristics of the wireless channel for testing so quickly, and is easy to reach an overfitting. The reason for the superior performance is that the parameters with meta-trained lie in a region that is sensitive to the loss functions from $\boldsymbol{p(T)}$.

V. Conclusion

In this paper, we have demonstrated our initial efforts to employ meta learning for channel estimation called RoemNet in an OFDM system. The model is meta-trained offline based on the simulated data. The most distinctive characteristic of RoemNet is that it involves a meta-learner with characteristic channels and a meta-update before a deep learning network. With meta-learner, RoemNet can solve new channel learning tasks using only a small number of pilots. Besides, the scheme employed by RoemNet leads to efficiency of transmission compared to existing approach, which is to train a new DNN for every block. Experimental results demonstrate that the proposed RoemNet outperforms LS, MMSE and the existing learning based method in terms of BER under certain scenarios.

We consider that this work is one step toward a practical learning based technique that can be applied to channel estimation in a real OFDM system. For further research, we plan to employ a reinforcement learning network in RoemNet to realize realtime OFDM symbol transmission. In addition, we hope to set up the dataset with samples from the real wireless channels to improve the performance of RoemNet.

References

[1] J. Armstrong, "Ofdm for optical communications," *Journal of lightwave technology*, vol. 27, no. 3, pp. 189–204, 2009.

[2] J.-J. Van De Beek, O. Edfors, M. Sandell, S. K. Wilson, and P. O. Borjesson, "On channel estimation in ofdm systems," in *Vehicular Technology Conference, 1995 IEEE 45th*, vol. 2. IEEE, 1995, pp. 815–819.

[3] S. Coleri, M. Ergen, A. Puri, and A. Bahai, "Channel estimation techniques based on pilot arrangement in ofdm systems," *IEEE Transactions on broadcasting*, vol. 48, no. 3, pp. 223–229, 2002.

[4] M. Morelli and U. Mengali, "A comparison of pilot-aided channel estimation methods for ofdm systems," *IEEE Transactions on signal processing*, vol. 49, no. 12, pp. 3065–3073, 2001.

[5] O. Edfors, M. Sandell, J.-J. Van de Beek, S. K. Wilson, and P. O. Borjesson, "Ofdm channel estimation by singular value decomposition," in *Vehicular Technology Conference, 1996. Mobile Technology for the Human Race., IEEE 46th*, vol. 2. IEEE, 1996, pp. 923–927.

[6] Y. Shen and E. Martinez, "Channel estimation in ofdm systems," *Freescale semiconductor application note*, pp. 1–15, 2006.

[7] B. Yang, Z. Cao, and K. B. Letaief, "Analysis of low-complexity windowed dft-based mmse channel estimator for ofdm systems," *IEEE Transactions on Communications*, vol. 49, no. 11, pp. 1977–1987, 2001.

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

[9] T. Gruber, S. Cammerer, J. Hoydis, and S. ten Brink, "On deep learning-based channel decoding," in *Information Sciences and Systems (CISS), 2017 51st Annual Conference on*. IEEE, 2017, pp. 1–6.

[10] Y.-S. Jeon, S.-N. Hong, and N. Lee, "Blind detection for mimo systems with low-resolution adcs using supervised learning," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

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

[12] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Unsupervised representation learning of structured radio communication signals," in *Sensing, Processing and Learning for Intelligent Machines (SPLINE), 2016 First International Workshop on*. IEEE, 2016, pp. 1–5.

[13] A. Felix, S. Cammerer, S. Dörner, J. Hoydis, and S. t. Brink, "Ofdm-autoencoder for end-to-end learning of communications systems," *arXiv preprint arXiv:1803.05815*, 2018.

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

[15] D. Lei, M. A. Hitt, and R. Bettis, "Dynamic core competences through meta-learning and strategic context," *Journal of management*, vol. 22, no. 4, pp. 549–569, 1996.

[16] R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artificial Intelligence Review*, vol. 18, no. 2, pp. 77–95, 2002.

[17] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *arXiv preprint arXiv:1703.03400*, 2017.

[18] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," in *Advances in Neural Information Processing Systems*, 2016, pp. 3981–3989.

[19] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[20] T. OShea 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.

[21] T. J. OShea and J. Hoydis, "An introduction to machine learning communications systems," *arXiv preprint*, vol. 1702, 2017.

[22] X. Li and L. J. Cimini, "Effects of clipping and filtering on the performance of ofdm," in *Vehicular Technology Conference, 1997, IEEE 47th*, vol. 3. IEEE, 1997, pp. 1634–1638.

[23] W. Zhang, X.-G. Xia, P. Ching, and W.-K. Ma, "On the number of pilots for ofdm system in multipath fading channels," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, vol. 4. IEEE, 2004, pp. iv–iv.

[24] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," 2016.

[25] A. Nichol and J. Schulman, "Reptile: a scalable metalearning algorithm," *arXiv preprint arXiv:1803.02999*, 2018.

[26] M. Al-Shedivat, T. Bansal, Y. Burda, I. Sutskever, I. Mordatch, and P. Abbeel, "Continuous adaptation via meta-learning in nonstationary and competitive environments," *arXiv preprint arXiv:1710.03641*, 2017.