

Beamforming Design for Large-Scale Antenna Arrays Using Deep Learning

Tian Lin *Student Member, IEEE*, and Yu Zhu, *Member, IEEE*

Abstract—Beamforming (BF) design for large-scale antenna arrays with limited radio frequency chains and the phase-shifter-based analog BF architecture, has been recognized as a key issue in millimeter wave communication systems. It becomes more challenging with imperfect channel state information (CSI). In this letter, we propose a deep learning based BF design approach and develop a BF neural network (BFNN) which can be trained to learn how to optimize the beamformer for maximizing the spectral efficiency with hardware limitation and imperfect CSI. Simulation results show that the proposed BFNN achieves significant performance improvement and strong robustness to imperfect CSI over the traditional BF algorithms.

Index Terms—Deep learning (DL), millimeter wave (mmWave), beamforming (BF) design, large-scale antenna arrays, neural network (NN), beamforming neural network (BFNN).

I. INTRODUCTION

A. Background and Motivations

Recently, hybrid analog and digital beamforming (HBF) design for millimeter wave (mmWave) communication systems with large-scale antenna arrays has been receiving much attention for its advantage of providing high beamforming (BF) gains to compensate for the severe path loss at affordable hardware cost and power consumption [1], [2]. It has been recognized that the most difficult part in the HBF optimization problem is the constant modulus constraint on analog BF due to its phase-shifter-based architecture [3]-[5]. In the existing works using the model-based design approach to handle this difficulty, an orthogonal matching pursuit (OMP) based algorithm was proposed in [1]. However, the analog beamformer is limited to a pre-defined codebook. To enhance the performance of OMP, the manifold optimization method was applied in [3], [4] for the analog BF optimization. An element wise iterative algorithm was proposed in [5] to optimize the analog beamformer. However, all of these algorithms either require some approximations to simplify the original objective function, or a lot of serial time consuming iterations to obtain a solution. Moreover, perfect channel state information (CSI) is assumed in all of these algorithms.

In another aspect, recent works on intelligent communications have shown the great potential of the data-based deep learning (DL) method in dealing with the traditional challenging problems [6]-[9],[11]. Inspired by these works, in this letter, we devote to applying the DL method to solve the complex BF design problem for mmWave systems with

hardware limitation and imperfect CSI. Our motivations can be explained from the following three aspects:

- First, it is well known that as the BF design is quite a complicated non-convex problem due to the joint optimization of multiple variables and the constant modulus constraint, it is unlikely to find a closed-form optimal solution [1]. As DL has been regarded as an efficient method to deal with intractable problems [12], it would be interesting to see what could be obtained if using DL to solve the BF optimization problem.
- Second, through a large number of training iterations with a lot of samples, the DL-based schemes have been shown to have the ability to understand the complicated characteristics of wireless channels [6]. Compared with the conventional works assuming perfect CSI [1], [3]-[5], the DL based approach is expected to possess strong robustness to imperfect CSI.
- Most efficient traditional BF algorithms require time-consuming serial iterations with high-complexity [4], [5]. However, the neural network (NN) after offline training has low complexity with limited matrix multiplications and additions when deployed online. Besides, thanks to the acceleration of parallel computation, the DL-based schemes can operate fast and thus be more applicable for high speed communications.

There have been some recent works on the DL-based HBF design [13][14]. However, the assumption of perfect CSI is made in [13], and the output beamformer does not directly meet the constant modulus constraint. In [14], the analog beamformer is limited to a pre-defined codebook, which normally leads to certain performance loss.

B. Novelty and Contributions

In this letter, we propose a DL-based BF design approach and develop a BF neural network (BFNN) which can be trained to learn how to optimize the beamformer for maximizing the spectral efficiency (SE) with hardware limitation and imperfect CSI. The contributions can be summarized as follows:

- *New design approach:* As the analog beamformer is implemented with analog phase shifters, we cannot follow the traditional full-digital design approach [6], [7] to replace the analog beamformer by a digital NN and train it in the whole communication link. Instead, we propose a novel DL-based design approach by developing the BFNN which directly outputs the optimized beamformer based on the input of the estimated CSI.
- *Novel Loss function:* Different from the traditional NNs, where the loss function is normally defined as the mean

This work was supported by National Natural Science Foundation of China under Grant No. 61771147.

T. Lin and Y. Zhu are with the Department of Communication Science and Engineering, Fudan University, Shanghai, China (e-mail: lint17@fudan.edu.cn, zhuyu@fudan.edu.cn).

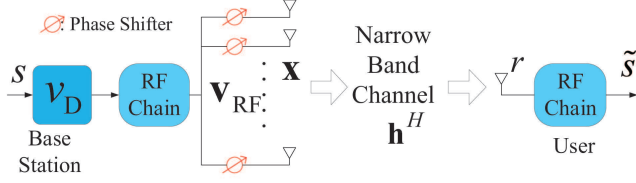


Fig. 1. Diagram of an MISO mmWave system with one RF chain.

square error (MSE) between the transmitted symbols (labels) and the recovered ones [6], [7], [11], in the BFNN, we propose an elegant loss function which is closely related to the performance of SE.

- **Robustness to imperfect CSI:** We propose a two-stage design approach to make the BFNN robust to imperfect CSI. During the first offline training stage, the BFNN learns how to approach the ideal SE when it only has practical channel estimate as its input. By doing so, in the second online deployment stage, the BFNN can adapt itself to imperfect CSI and achieve robust performance to channel estimation errors.

Since in the mmWave HBF design the optimal digital beamformer normally has a closed-form solution [3]-[5], as an initial work and for the easy of presentation, in this letter we focus on the analog BF design and consider the scenario of a large-scale antenna array with only one radio frequency (RF) chain.

II. SYSTEM MODEL

Consider in Fig. 1 the downlink of a narrowband multiple-input and single-output (MISO) mmWave system with the analog BF (precoding) architecture, where a base station (BS) with one RF chain and N_t antennas transmits one data stream to a user with only one receive antenna.¹ Let s denote the transmitted symbol with normalized average symbol energy, i.e., $E\{|s|^2\} = 1$. The symbol is first multiplied by a scalar digital precoder v_D (since there is only one RF chain, v_D is actually a scalar), and then by an $N_t \times 1$ analog precoder \mathbf{v}_{RF} , which is implemented using phase shifters. The final precoded signal is then given by $\mathbf{x} = \mathbf{v}_{RF} v_D s$.

Through the MISO mmWave channel, the received signal at the user can be represented by $r = \mathbf{h}^H \mathbf{v}_{RF} v_D s + n$, where n is the additive noise satisfying the circularly symmetric complex Gaussian distribution with zero mean and covariance σ^2 , and \mathbf{h}^H is the channel vector between the BS and the user. In this letter, a widely used Saleh-Valenzuela mmWave channel model [10][11] for \mathbf{h}^H with one line-of-sight (LoS) path and $(L-1)$ non-LoS (NLoS) paths is adopted, which is represented as

$$\mathbf{h}^H = \sqrt{\frac{N_t}{L}} \sum_{l=1}^L \alpha_l \mathbf{a}_t^H(\phi_t^l), \quad (1)$$

where α_l denotes the complex gain of the l th path, and $\mathbf{a}_t(\phi_t^l)$ denotes the antenna array response vector at the BS, with ϕ_t^l

denoting the azimuth angle of departure associated with the l th path. In particular, the term with $l = 1$ denotes the LoS component in \mathbf{h}^H .

In this letter, the SE, which has been widely used in the existing BF designs [4], [5], is chosen as the optimization objective, which is given as follows for the studied system

$$R = \log_2 \left(1 + \frac{1}{\sigma^2} \|\mathbf{h}^H \mathbf{v}_{RF} v_D\|^2 \right). \quad (2)$$

Considering the constant modulus constraint, $|\mathbf{v}_{RF}|_i|^2 = 1$, for $i = 1, \dots, N_t$, and the maximum transmit power constraint $\|\mathbf{v}_{RF} v_D\|^2 \leq P$, it can be proved that the optimal v_D for maximizing R is given by $\sqrt{P/N_t}$. Then, the BF optimization problem for \mathbf{v}_{RF} is given by

$$\begin{aligned} & \underset{\mathbf{v}_{RF}}{\text{minimize}} && \log_2(1 + \frac{\gamma}{N_t} \|\mathbf{h}^H \mathbf{v}_{RF}\|^2) \\ & \text{subject to} && |\mathbf{v}_{RF}|_i|^2 = 1, \quad \text{for } i = 1, \dots, N_t, \end{aligned} \quad (3)$$

where $\gamma = \frac{P}{\sigma^2}$ denotes the signal to noise ratio (SNR). As the SNR can be generally considered to be more accurately estimated than the CSI, throughout this letter we assume $\gamma_{\text{est}} = \gamma$, where γ_{est} denotes the estimated SNR, and focus on how to deal with imperfect CSI.

III. DL MODEL AND DESIGN OF BFNN

In this section, we first introduce the new challenges when applying the DL method to solve (3) with imperfect CSI. Then, we describe in detail the BFNN architecture. Finally, we analyze its complexity.

A. Challenges

Since the analog beamformer has a specific architecture containing analog phase shifters, we cannot follow the traditional approach to replace it by a multi-layer NN and train it in the BS-user communication link [6]-[8]. Here, we propose a different DL design approach by designing the BFNN that directly outputs \mathbf{v}_{RF} to solve (3). However, this design is not trivial due to the following three challenges.

- **What should the input of the BFNN be?** Most existing DL-based works [6], [7], [14] take the received baseband digital signal as the input. However, this approach cannot be applied here as the received signal itself is a function of the analog precoder to be optimized. Furthermore, the number of dimensions of the received signal is much lower than that of the precoder (N_t) to be optimized.
- **How to guarantee that the output (\mathbf{v}_{RF}) satisfies the constraint?** As it is well known that complex output is not well supported by most DL frameworks (e.g., Tensorflow, pytorch), it would be more difficult to further impose the constant modulus constraint on the output.
- **What should the label be when training the BFNN?** In almost all of the conventional intelligent communication designs [8], [11], the label is set exactly as the transmitted bits or perfect CSI. However, it is difficult to find a proper label for this BF design problem. For example, if we take an optimized analog beamformer based on a traditional algorithm as the label, the resulting BFNN would not perform better than the traditional algorithm.

¹Although this letter focuses on the narrowband analog BF design with the aid of DL, the design approach can be generalized to a broadband MIMO mmWave system with HBF, which will be discussed in Section V.

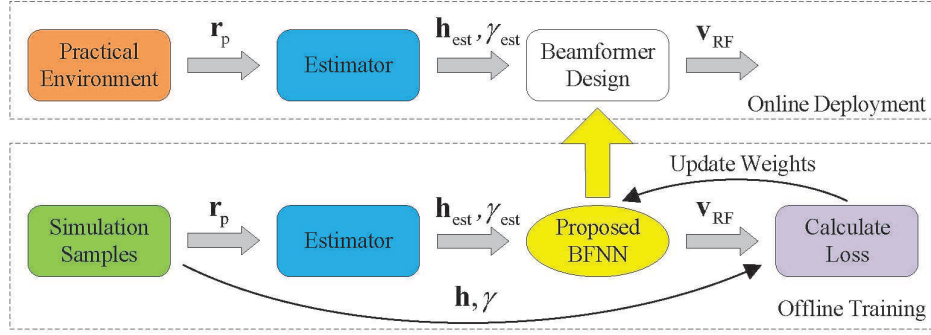


Fig. 2. Illustration of the two-stage DL-based HBF design approach: offline training and online deployment.

B. BFNN Architecture

In this subsection we first elaborate on some considerations for the above three challenges, and then describe the two-stage design approach of the BFNN.

1) Three Specific Considerations:

- **Input of the BFNN:** As the analog beamformer is implemented with analog devices, it cannot be replaced by a full-digital NN and trained in the whole communication link. Instead, the BFNN is designed to generate an optimized analog BF vector \mathbf{v}_{RF} based on the input of the channel estimate \mathbf{h}_{est} and the SNR estimate γ_{est} .
- **Lambda Layer:** To ensure that the output of the BFNN, i.e., \mathbf{v}_{RF} , is a complex-valued vector satisfying the constant modulus constraint, a self-defined Lambda layer is added at the end of the BFNN. Specifically, letting θ denote its real-valued input (the output of the last dense unit), its complex-valued output is given by

$$\mathbf{v}_{RF} = \exp(j \cdot \theta) = \cos(\theta) + j \cdot \sin(\theta), \quad (4)$$

where $j = \sqrt{-1}$. It can be seen from (4) that θ has a clear physical meaning that each element of θ corresponds to the phase of each analog BF coefficient in \mathbf{v}_{RF} . Compared with an alternative approach which is to first generate the real and imaginary parts of a complex value and then normalize that complex value on the unit circle, the proposed method directly generates the phase component, which automatically guarantees the constant modulus constraint with less neurons and makes the NN more elegant.

- **Loss Function:** Different from traditional supervised learning designs [6], [7], in our design, there is no need of labels and the BFNN is trained with the following new loss function directly related to the objective in (3)

$$\text{Loss} = -\frac{1}{N} \sum_{n=1}^N \log_2 \left(1 + \frac{\gamma_n}{N_t} \|\mathbf{h}_n^H \mathbf{v}_{RF,n}\|^2 \right), \quad (5)$$

where N denotes the total number of training samples, and γ_n , \mathbf{h}_n and $\mathbf{v}_{RF,n}$ represent the SNR, CSI and output analog beamformer associated with the n th sample. Note that the reduction of the loss exactly corresponds to the increase of the average SE.

2) **Two-stage Design Approach:** With the above considerations on BFNN, the two-stage DL-based BF design approach is illustrated in Fig. 2. In the offline training stage, channel samples, transmit pilot symbols, noise samples are generated via simulation according to the system model in Section II. Then, a practical mmWave channel estimator is applied for the BS to obtain partial CSI. In this letter, we apply the classical mmWave channel estimator proposed in [2], where the BS estimates the channel by sending pilot symbols with beamformers in a hierarchical codebook and receiving the feedback of the user's decision based on its received signal \mathbf{r}_p . The channel estimate \mathbf{h}_{est} and the SNR estimate γ_{est} are sent into the BFNN as input. Note that as mentioned in Section II, it is assumed $\gamma_{est} = \gamma$. Then, the BFNN generates the optimized analog beamforming vector $\mathbf{v}_{RF,n}$ by minimizing the loss function defined in (5). Since the channel samples and SNR values are generated via simulation, they (perfect CSI and SNR information) can be directly used in the calculation of the loss, as shown in Fig. 2. As the DL-based technique is essentially a gradient-descend method, the loss can be guaranteed to converge to a local optimal with a proper learning rate [12]. By taking the channel estimate as the input of BFNN and using the perfect CSI in the loss function, the BFNN can be trained to learn how to approach the ideal SE with perfect CSI as much as possible and be robust to channel estimation errors.

In the online deployment stage, the same mmWave channel estimator is applied for the BS to obtain partial CSI. The channel estimate is then sent into the BFNN to obtain the optimized beamformer. It is worth noting that the perfect CSI is only required to compute the loss during the offline training stage. When deployed online, all parameters of the BFNN have already been fixed and the well-trained BFNN only accepts the imperfect CSI as the input and directly outputs the analog beamformer, with no need to take the perfect CSI to calculate the loss.

To show the detailed structure of the BFNN, we consider a MISO system with $N_t = 64$ and show in Tab. I the network structure, the output dimension, the activation function (if used) and the number of trainable parameters of each layer. It will be shown in Section IV that such a network can achieve excellent performance in the mmWave system setup there. As shown in Fig. 2, since the input \mathbf{h}_{est} is a complex-valued vector and the BFNN is a real-valued network, the real and imaginary

parts of \mathbf{h}_{est} are concatenated and further with γ_{est} to form a $(2N_t + 1) \times 1$ real-valued input vector. Three dense layers are then applied with 256, 128, 64 neurons, respectively. Similar to most related works [6]-[9][11][13][14], the number of dense layers and that of the neurons in each layer are determined by the empirical trials. Different activation functions (e.g., Relu, tanh and sigmoid) have been tried for the last dense layer, but simulation results suggested that the output without any activation function achieves the best performance and convergence behavior. To enhance the convergence, each dense layer is preceded by a batch normalization layer, which is omitted in the Tab. I. At the end, the Lambda layer imposes the constraint modulus constraint on the final output. To guarantee the generality of the BFNN, multiple samples are required for offline training. In our experiments, the training, validation, and testing sets contain 10^5 , 10^4 , and 10^4 samples, respectively.

C. Complexity Analysis

In this subsection we analyze the computational complexity in terms of the number of floating-point operations (FLOPs) for the proposed BFNN. As the complexity in the offline training stage is normally not counted [8], only that in the online deployment stage is counted. According to [16], the number of FLOPs of a dense layer is given by $(2N_t - 1)N_o$, where N_t and N_o denote the input and output dimensions, respectively. In the case when $N_t = 64$, considering the BFNN in Tab. I, the total number of FLOPs of the BFNN is about 0.15 million.

For conventional model-based HBF designs [1][4][5], the asymptotic computational complexity in terms of the number of complex multiplications is in the order of $\mathcal{O}(N_t^3)$ as they involve the operations such as singular value decomposition and matrix inversion. Even taking the coefficient in the complexity order to be 1, the number of N_t^3 complex multiplications is about 0.26 million with $N_t = 64$. It can be seen that the proposed BFNN has competitive computational complexity when compared with the traditional model-based HBF algorithms. In addition, the main operation of BFNN involves large-scale matrix multiplications and additions, which can be effectively accelerated by the graphics processing unit (GPU). However, most of the traditional model-based HBF algorithms normally involve serial iterations (the optimization of the next iteration depends on the result of the previous iteration) and are not suitable for parallel computing.

IV. SIMULATION RESULTS

Throughout the simulations, a half-wave spaced uniform linear array with $N_t = 64$ is deployed at the BS. The Saleh-Valenzuela mmWave channel model in (1) with exactly the same parameters as those in [10] is considered, where L is set to 3, i.e., the channel contains one LoS path and two NLoS paths. α_l satisfies independently and circularly symmetric Gaussian distribution with zero mean, and the variance of α_l is set to 1 for $l = 1$, and set to $10^{-0.5}$ for $l = 2, 3$. ϕ_l^t satisfies independently uniform distribution in $[-0.5\pi, 0.5\pi]$. Two state-of-the-art HBF algorithms in the special case of one RF chain

TABLE I
IMPLEMENTATION DETAILS OF THE BFNN.

Layer Name	Output Dim.	Activation Func.	Number of Paras.
Input Layer	129×1	\searrow	0
Dense Layer 1	256×1	ReLu	33024
Dense Layer 2	128×1	ReLu	32896
Dense Layer 3	64×1	\searrow	8256
Lambda Layer	64×1	\searrow	0

are considered for comparison, i.e., the manifold-optimization based HBF algorithm in [4] and the iterative HBF algorithm in [5]. The classical channel estimation algorithm in [2] is applied for obtaining \mathbf{h}_{est} . In the traditional HBF algorithms, \mathbf{h} is directly replaced by \mathbf{h}_{est} when computing the BF coefficients. In the proposed BFNN, the hyper-parameter setting is shown in Tab. I and fixed throughout all experiments. The learning rate is initialized at 0.001 and the Adam optimizer is used. The real-valued loss function results in real-valued gradient of trainable parameters of the BFNN, and therefore can be directly implemented with Tensorflow. All source codes, data sets and some trained weights are provided openly in [15].

Fig. 3 shows the SE versus SNR performance under three channel estimation levels, which are characterized by three pilot-to-noise power ratios (PNRs), i.e., -20dB, 0dB and 20dB. It is assumed that the estimation of the number of channel paths is correct, i.e., $L_{\text{est}} = 3$. The results in Fig. 3 show that with imperfect CSI the two traditional HBF algorithms perform quite similarly, but proposed BFNN significantly outperforms them. For example, at a SE of 8bits/s/Hz, the proposed BFNN achieves around a 1.5dB gain in SNR over the traditional algorithms when $\text{PNR} = 20\text{dB}$, and such a gain becomes larger for smaller PNRs.

In practical systems, there also exist estimation errors in estimating the number of total paths L . Due to the sparsity of the mmWave channels and considering the estimation complexity, L_{est} is preset to a small value [2]. Fig. 4 shows the SE performance for different L_{est} values. It can be seen from this figure that the proposed BFNN outperforms the traditional HBF algorithms with the improvement becoming larger for less accurate L_{est} .

In summary, it can be concluded that the proposed BFNN exhibits much stronger robustness to imperfect CSI than the traditional algorithms. The less accurate the channel estimate is, the larger the performance gap is. This is because through many iterations with large training sets, the BFNN has been trained to learn the characteristics of the mmWave propagation channels, as well as the relationship between the imperfect CSI and the ideal SE with perfect CSI.

V. DISCUSSION OF THE GENERALITY OF BFNN

Although the BFNN is designed specifically for a simple scenario in this paper, it has good generality for other more complex problems. For example, in the broadband scenario, by concatenating the multi-tap channel vectors as the input and redefining the loss function to the one related to the broadband SE [4], the BFNN can be extended to optimize the analog beamformer for broadband mmWave channels. For

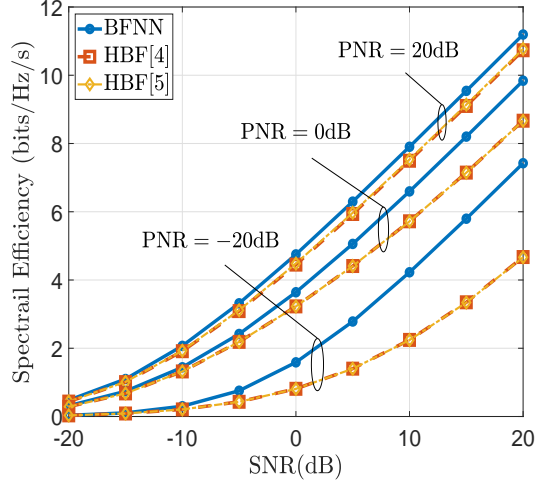


Fig. 3. SE v.s. SNR for different BF algorithms with different PNRs.

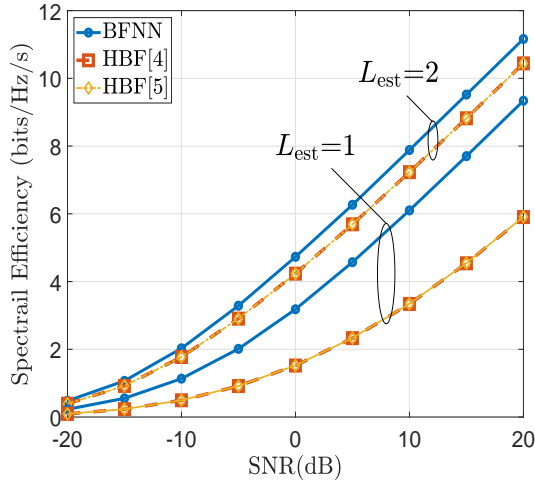


Fig. 4. SE v.s. SNR for different BF algorithms with different L_{est} .

another example, considering the HBF problem with multiple RF chains, while the optimal digital beamformer can be solved with a closed-form solution [3], [5], a simple extension of the current BFNN is to increase the output dimensions from N_t to $N_{RF}N_t$ for the $N_t \times N_{RF}$ analog BF matrix with a new loss function. Following the similar idea, the BFNN can also be considered in the joint transmit and receive BF design or the multi-user BF design.

VI. CONCLUSION AND FUTURE WORK

We have proposed a DL-based BF design approach for mmWave systems with large-scale antenna arrays. With some special designs on the self-defined Lambda layer and the loss function, the proposed BFNN can well handle the challenges of hardware limitation and imperfect CSI in mmWave systems. Simulation results have shown the competitive performance of the BFNN and provided valuable insights for future BF designs.

As for future work, due to the generality of BFNN, it is of interest to extend the BFNN to more complex BF problems. Furthermore, like that in the existing works [6][8][13][14], in this study the number of layers and that of the neurons in each layer in the proposed BFNN are mainly determined by the empirical trials, it is also of interest to investigate the physical meaning of each layer.

REFERENCES

- [1] O. E. Ayach, S. Rajagopal, S. Abu-Surra, Z. Pi, and R. W. Heath *et al.*, "Spatially sparse precoding in millimeter wave MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1499–1513, Mar. 2014.
- [2] A. Alkhateeb, O. El Ayach, G. Leus, and R. W. Heath, "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE J. Sel. Topics in Signal Process.*, vol. 8, no. 5, pp. 831–846, Oct. 2014.
- [3] T. Lin, J. Cong, Y. Zhu, J. Zhang, and K. B. Letaief, "Hybrid beamforming for millimeter wave systems using the MMSE criterion," *IEEE Trans. on Commun.*, vol. 67, no. 5, pp. 3693–3708, May 2019.
- [4] X. Yu, J. Shen, J. Zhang, and K. B. Letaief, "Alternating minimization algorithms for hybrid precoding in millimeter wave MIMO systems," *IEEE J. Sel. Topics in Signal Process.*, vol. 10, no. 3, pp. 485–500, Apr. 2016.
- [5] F. Sohrabi and W. Yu, "Hybrid digital and analog beamforming design for large-scale antenna arrays," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 501–513, Apr. 2016.
- [6] H. Ye, G. Y. Li, and B. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.
- [7] S. Dörner, S. Cammerer, J. Hoydis, and S. Brink, "Deep learning based communication over the air," *IEEE J. Sel. Topics in Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.
- [8] X. Gao, S. Jin, C. Wen and G. Y. Li, "ComNet: combination of deep learning and expert knowledge in OFDM receivers," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2627–2630, Dec. 2018.
- [9] C. Wen, W. Shih, and S. Jin, "Deep learning for massive MIMO CSI feedback," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 748–751, Oct. 2018.
- [10] X. Gao, L. Dai, S. Han, C.-L. I, and X. Wang, "Reliable beamspace channel estimation for millimeter-wave massive MIMO systems with lens antenna array," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 6010–6021, Sep. 2017.
- [11] H. He, C. Wen, S. Jin and G. Y. Li, "Deep learning-based channel estimation for beamspace mmWave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 852–855, Oct. 2018.
- [12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press Cambridge, 2016.
- [13] H. Huang, Y. Song, J. Yang, G. Gui, and F. Adachi, "Deep-learning-based millimeter-wave massive MIMO for hybrid precoding," *IEEE Trans. on Veh. Technol.*, vol. 68, no. 3, pp. 3027–3032, Mar. 2019.
- [14] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, "Deep learning coordinated beamforming for highly-mobile millimeter wave systems," *IEEE Access*, vol. 6, pp. 37328–37348, 2018.
- [15] <https://github.com/TianLin0509/BF-design-with-DL>.
- [16] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," [Online]. Available: <https://arxiv.org/abs/1611.06440>.