Binary Neural Network Aided CSI Feedback in Massive MIMO System

Zhilin Lu
Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University
Beijing, China
luzl18@mails.tsinghua.edu.cn

Jintao Wang
Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University
Beijing, China
wangjintao@tsinghua.edu.cn

Jian Song
Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University
Beijing, China
jsong@tsinghua.edu.cn

***Abstract*—In massive multiple-input multiple-output (MIMO) system, channel state information (CSI) is essential for the base station to achieve high performance gain. Recently, deep learning is widely used in CSI compression to fight against the growing feedback overhead brought by massive MIMO in frequency division duplexing system. However, applying neural network brings extra memory and computation cost, which is non-negligible especially for the resource limited user equipment (UE). In this paper, a novel binarization aided feedback network named BCsiNet is introduced. Moreover, BCsiNet variants are designed to boost the performance under customized training and inference schemes. Experiments shows that BCsiNet offers over 30× memory saving and around 2× inference acceleration for encoder at UE compared with CsiNet. Furthermore, the feedback performance of BCsiNet is comparable with original CsiNet. The key results can be reproduced with https://github.com/Kylin9511/BCsiNet.**

***Index Terms*—Massive MIMO, CSI feedback, deep learning, binary neural network, lightweight network**

I. Introduction

Massive multiple-input multiple-output (MIMO) is a promising technique for 5G wireless communication systems, providing better the spectrum and energy efficiency [1] [2]. However, the base station (BS) needs realtime channel state information (CSI) to acquire performance gain. In frequency division duplexing (FDD) system, downlink CSI must be fed back from user equipment (UE) due to the asymmetry of uplink and downlink channel. With the growing numbers of transmitting antennas in massive MIMO system, the feedback overhead of the CSI matrix becomes unbearable.

Recently, deep learning (DL) is widely adopted to wireless communication scenarios including downlink CSI feedback. Generally, the CSI matrix is compressed by a neural network (NN) based encoder at UE to reduce feedback overhead. CsiNet [3] is the first to prove the effectiveness of the DL based CSI feedback scheme over traditional compressed sensing (CS) algorithms [4].

Many works make remarkable contributions by extending the basic scenario proposed in [3]. Time-varying CSI is considered in [5] while correlation between uplink and downlink CSI is utilized in [6], [7]. Entropy quantizer is adopted in [8]. A denoise module is added to deal with imperfect feedback in [9], [10]. At the mean while, series of papers introduce novel network design for performance boosting. CRNet [11] and CsiNetPlus [12] improve the feedback capacity with multi-resolution network and network expansion, respectively. Pseudo-3D convolution is utilized in [13] and fully connected (FC) layers is removed in [14]. Squeeze and excitation network is used in [15] while non-local block is applied in [16].

However, most of these works trade higher performance with extra memory and computation cost. This can be impractical for many user equipments whose hardware resources are strictly limited. Existing papers devoted to lightweight network design mainly based on larger network like CsiNetPlus [12] and ConvCsiNet [14]. Therefore many of the final lightweight networks are actually heavier than CsiNet in computation complexity or parameters size. [17] introduces FC pruning and network quantization and gives impressive results. However, pruning with unstructured weights brings difficulty to inference at UE and vanilla model quantization harms the performance deeply when the quantized bits is lower than five.

In this paper, we propose a novel lightweight feedback network named binary CsiNet (BCsiNet). Following basic principles proposed in [18], [19], we successfully binarize the FC layer at UE. Moreover, we expand the BCsiNet design to boost the performance without much extra cost. Simulation shows that BCsiNet has comparable performance against the original CsiNet while offering over 30× memory saving and around 2× acceleration at UE.

The main contributions of this paper is listed as follows.

- Binary neural network (BNN) technique is introduced to CSI feedback task and proved to be effective. To the author's best knowledge, we are the first to apply BNN into extremely lightweight feedback encoder design.
- Neat and valid manner is given to generate BCsiNet variants. The expanded BCsiNet has better performance with little extra cost.
- Special training and inference schemes are designed for BCsiNet, which is essential for training convergence and inference acceleration.

The rest of the paper is structured as follows. System model is proposed in section II while detailed design of BCsiNet is given in section III. Section IV presents the numerical results and analysis. Section V concludes the paper in the end.

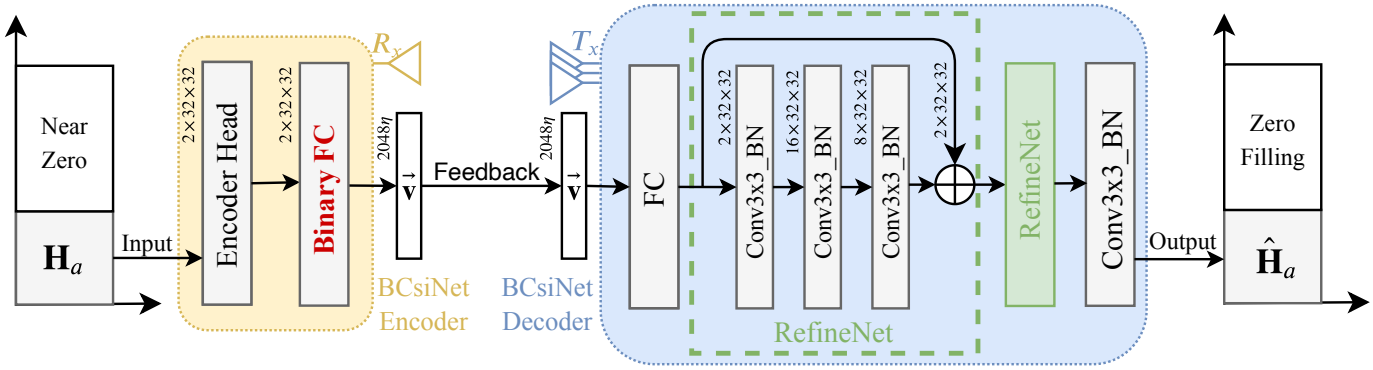

Fig. 1. Overview of BCsiNet aided downlink CSI feedback workflow. The fully connected layer of the encoder is binarized.

II. System Model

We consider a single-cell massive MIMO FDD system with N_c orthogonal frequency division multiplexing (OFDM) sub-carriers. There are $N_t (N_t \gg 1)$ transmitting antennas at BS and N_r receiving antennas at UE. We take $N_r = 1$ for simplicity. The received signal can be expressed as follows:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}, \tag{1}$$

where $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{C}^{N_c \times 1}$ are the transmitted signal, received signal and additive noise in one OFDM period. $\mathbf{A} = \text{diag}(\tilde{\mathbf{h}}_1^H \mathbf{p}_1, \cdots, \tilde{\mathbf{h}}_{N_c}^H \mathbf{p}_{N_c})$ is a diagonal matrix, where $\tilde{\mathbf{h}}_i, \mathbf{p}_i \in \mathbb{C}^{N_t \times 1}, i \in \{1, \cdots, N_c\}$ are the downlink channel vector and precoding vector at sub-carrier i, respectively.

The base station needs to obtain the channel vector $\tilde{\mathbf{h}}_i$ in order to design the precoding vector $\mathbf{p}_i$, for which all the $\tilde{\mathbf{h}}_i$ must be fed back from UE. We define the overall downlink channel matrix as $\tilde{\mathbf{H}} \triangleq [\tilde{\mathbf{h}}_1, \cdots, \tilde{\mathbf{h}}_{N_c}]^H$, which contains $2N_cN_t$ elements.

The overhead is too large to directly feed all $2N_cN_t$ elements back. Like it is shown in [3], the channel matrix is sparse in the angular-delay domain. We can take advantage of it and transfer the CSI matrix from spatial-frequency domain to angular-delay domain via discrete Fourier transform (DFT).

$$\mathbf{H} = \mathbf{F}_c \tilde{\mathbf{H}} \mathbf{F}_t^H, \tag{2}$$

where F_c and F_t are $N_c \times N_c$ and $N_t \times N_t$ DFT matrices, respectively. Since the time delay of multi-path arrivals is limited, only the first N_a rows of CSI matrix $\mathbf{H}$ contains large components. The rest of the rows with near-zero elements can be truncated. We denote the first N_a rows of the CSI matrix with $\mathbf{H}_a$ for simplicity.

However, the matrix $\mathbf{H}_a$ is still too heavy in massive MIMO system where N_t is large. That is why $\mathbf{H}_a$ needs to be further compressed before feedback. Traditional compressed sensing algorithms require $\mathbf{H}_a$ to be sparse enough, but the sparsity of $\mathbf{H}_a$ is guaranteed only when $N_t \to \infty$ which is impractical [20]. Neural network based encoder-decoder can work without such limitation and achieve a better CSI compressing and reconstructing performance.

The overall feedback workflow is demonstrated in Fig. 1. At the beginning, the BCsiNet encoder takes the truncated CSI matrix $\mathbf{H}_a$ as input at UE. The compressed channel feature $\mathbf{v}$ is generated and fed back to BS. Then the BCsiNet decoder reconstruct $\mathbf{H}_a$ and the original CSI matrix $\tilde{\mathbf{H}}$ is achieved by zero filling and inverse DFT. We can summarize the aforementioned feedback workflow with equation (3).

$$\hat{\mathbf{H}}_a = F_d(F_e(\mathbf{H}_a, \Theta_e), \Theta_d), \tag{3}$$

where F_e represents BCsiNet encoder with parameters Θ_e and F_d represents BCsiNet decoder with parameters Θ_d. With Θ_e and Θ_d properly optimized, distance between $\mathbf{H}_a$ and $\hat{\mathbf{H}}_a$ can be minimized. Our purpose here is to lighten Θ_e using FC binarization to reduce the memory and power consumption together with computational complexity.

It is worth mentioning that the uplink feedback is assumed to be ideal in this paper. Besides, the COST2100 [21] channel model is used to simulate the CSI matrix.

III. Binary Neural Network in CSI Feedback

A. Complexity Analysis of Existing Feedback Networks

Before explaining how binary neural network works, we give a glance at complexity of several existing feedback networks to provide a contrastive point of view.

As it is presented in Table I, later feedback networks mainly focus on trading better performance with larger cost. Therefore, CsiNet actually has the lightest encoder among all these networks. By further compressing the CsiNet encoder, we are able to provide an extremely lightweight encoder that can works under much tougher resource restriction at UE.

In order to further compress the CsiNet encoder, we need to analyze the distribution of its complexity. As we can see in Table I and Table II, the complexity proportion of FC goes higher as the encoder gets simpler. With simplest encoder, FC layer takes up 96.61% and 99.996% of computation and memory resources of original CsiNet, respectively.

From the aforementioned observation, it is clear that a lighter fully connected layer is the key to the extremely lightweight CsiNet encoder design. Note that the flops counting in Table I and Table II ignores the batch normalization

TABLE I
PARAMS AND FLOPS OF SEVERAL EXISTING FEEDBACK NETWORKS

Methods[a]	Encoder at UE		Decoder at BS	
	FLOPs[b]	params	FLOPs[b]	params
CsiNet [3]	1.09M	1.05M	4.33M	1.05M
CRNet [11]	1.20M	1.05M	3.92M	1.05M
CsiNetPlus [12]	1.45M	1.05M	23.12M	1.07M
ConvCsiNet [14]	60.16M	2.14M	166.07M	2.07M
DeepCMC [8]	173.54M	3.32M	278.40M	9.87M

[a] The compression ratio η is $1/4$ for all methods.
[b] FLOPs is total number of "multiply then add" operation.

TABLE II
ENCODER COMPLEXITY DISTRIBUTION OF FEEDBACK NETWORKS

Methods[a]	UE FLOPs[b]		UE params	
	FC	others[c]	FC	others[c]
CsiNet [3]	96.60%	3.40%	99.996%	0.004%
CRNet [11]	87.07%	12.93%	99.984%	0.016%
CsiNetPlus [12]	72.32%	27.68%	99.962%	0.038%

[a] The compression ratio η is $1/4$ for all methods.
[b] FLOPs is the total number of "multiply then add" operation.
[c] all other layers including convolution, batch normalization, etc.

(BN) layer. BN is cost free in inference since it can be merged into the corresponding convolution layer.

B. Binarization Algorithm Design for Fully Connected Layer

In this subsection, we will explain how to slim the FC layer at UE with network binarization. Compared with standard NN, BNN is able to save around $32\times$ memory since the parameters change from 32bits float point numbers to 1bit binary numbers.

Moreover, the "multiply-accumulate" operation in standard NN is replaced by simple addition in BNN as it is shown in equation (4). Generally in one "multiply-accumulate" operation, multiplication takes more time than addition on chips. For instance, the latency of float multiplication and addition is 4 and 2 in VIA Nano 2000 series of CPU [22]. Therefore, the speed of BNN inference is over $2\times$ faster than standard NN since the time consuming multiplications are gone.

$$\begin{aligned} &\begin{bmatrix}\alpha_1 & \alpha_2\\ \alpha_3 & \alpha_4\end{bmatrix}\cdot\begin{bmatrix}\beta_1\\ \beta_2\end{bmatrix}=\begin{bmatrix}\alpha_1\times\beta_1+\alpha_2\times\beta_2\\ \alpha_3\times\beta_1+\beta_4\times\beta_2\end{bmatrix} && \text{for NN}\\ &\begin{bmatrix}1 & -1\\ 1 & 1\end{bmatrix}\cdot\begin{bmatrix}\beta_1\\ \beta_2\end{bmatrix}=\begin{bmatrix}\beta_1-\beta_2\\ \beta_1+\beta_2\end{bmatrix} && \text{for BNN}\end{aligned} \tag{4}$$

However, binarization of the FC layer may harm the network performance. In order to reduce the information loss of binarization, a positive scale factor $\alpha\in\mathbb{R}^+$ is attached to the binary FC layer. For standard FC weight $\mathbf{W}\in\mathbb{R}^{m\times n}$ and binary FC weight $\mathbf{B}\in\{-1,1\}^{m\times n}$, the distance between $\mathbf{W}$ and $\mathbf{W}_b=\alpha\mathbf{B}$ should to be minimized. Denoting the vectorized matrix as $\mathbf{w}\triangleq\text{vec}(\mathbf{W})\in\mathbb{R}^{mn}$ and $\mathbf{b}\triangleq\text{vec}(\mathbf{B})\in\{-1,1\}^{mn}$, we can formulate the optimization problem as follows.

$$(\mathbf{b}^{opt},\alpha^{opt})=\arg\min_{\mathbf{b},\alpha}\|\mathbf{w}-\alpha\mathbf{b}\|^2 \tag{5}$$

Following the derivation in [19], we expand the original distance in (5).

$$\begin{aligned}\|\mathbf{w}-\alpha\mathbf{b}\|^2&=\left(\mathbf{b}^T\mathbf{b}\alpha^2-2\mathbf{w}^T\mathbf{b}\alpha+\mathbf{w}^T\mathbf{w}\right)\\&=\left(mn\alpha^2-2\mathbf{w}^T\mathbf{b}\alpha+\mathbf{w}^T\mathbf{w}\right)\\&\triangleq\left(mn\alpha^2-2\mathbf{w}^T\mathbf{b}\alpha+Z\right),\end{aligned} \tag{6}$$

where $Z=\mathbf{w}^T\mathbf{w}$ is a const since $\mathbf{w}$ is given. It is obvious that $\mathbf{w}^T\mathbf{b}$ should be maximized to minimize equation (6).

$$\mathbf{b}^{opt}=\arg\max_{\mathbf{b}}\mathbf{w}^T\mathbf{b}=\text{sign}(\mathbf{w}) \tag{7}$$

Then α^{opt} can be deduced from the derivative of (6).

$$\begin{aligned}\alpha^{opt}&=-\frac{-2\mathbf{w}^T\mathbf{b}^{opt}}{2mn}\\&=\frac{\mathbf{w}^T\text{sign}(\mathbf{w})}{mn}\\&=\frac{1}{mn}\sum_{i=1}^{mn}|\mathbf{w}_i|=\frac{1}{mn}\|\mathbf{W}\|_1\end{aligned} \tag{8}$$

Therefore the final binarized $\mathbf{W}_b$ can be derived as follows.

$$\mathbf{W}_b=\alpha\cdot\text{sign}(\mathbf{W})=\frac{1}{mn}\|\mathbf{W}\|_1\text{sign}(\mathbf{W}) \tag{9}$$

A demonstration of the FC binarization with scale is given in Fig. 2 for better intuitive understanding. It is worth mentioning that we only binarize the weight in FC layer while the bias remains unchanged. This enhances the capacity of Binary FC layers with little cost.

C. Design of the proposed BCsiNet

Vanilla BCsiNet can be obtained by directly applying binarization algorithm in section III-B to the encoder of CsiNet. As it is shown in Fig. 1, the encoder of BCsiNet is made up of an encoder head and a binary FC layer while the decoder consists of two RefineNets and a 3×3 convolution layer.

In order to enhance the performance of proposed BCsiNet, we design several variants for it. The original encoder head of CsiNet is called "head A", which is single 3×3 convolution layer. Head B and head C are designed to replace head A for better performance. As we can see in Fig. 3, head B adds a concatenated 3×3 convolution layer to head A, while head C

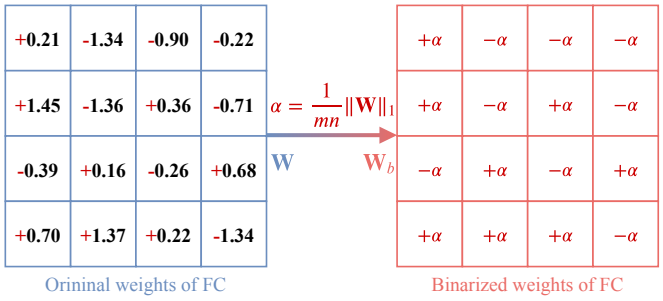

Fig. 2. A demonstration of fully connected layer binarization with scale.

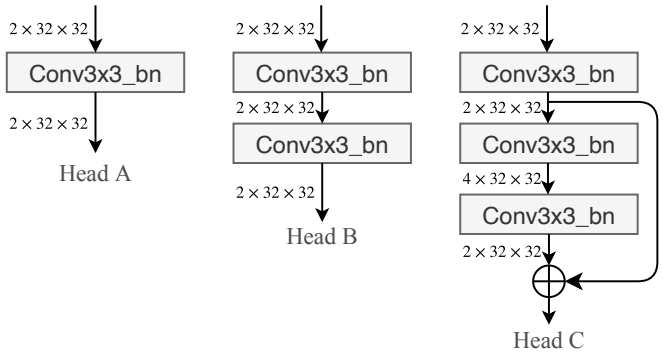

Fig. 3. Design of BCsiNet encoder heads.

includes an extra residual architecture. Deeper encoder head like head B or head C can extract the channel feature better since the available resolution is larger.

In fact, extra cost of head B or head C is tiny and BCsiNet encoder is extremely light with all these heads. Compared with original CsiNet encoder, BCsiNet encoder saves $31.49\times$, $31.48\times$ and $31.34\times$ memory with head A, B and C, respectively. Similarly, applying different heads has little impact on the $2\times$ inference acceleration of BCsiNet.

On the other hand, we expand the original CsiNet decoder by adding one more RefineNet. Three concatenated RefineNets work better without extra cost at UE. The computational complexity rises from 4.32M to 5.95M with negligible parameter size increase, which is completely acceptable for the base station.

Note that all the convolution layers in BCsiNet are followed by BN layer and activation layer. LeakyReLU with negative slope of 0.3 is used as activation function. Specially, a sigmoid function replaces the original LeakyReLU at the end of BCsiNet decoder to limit the output range.

D. Training and Inference of BCsiNet

Special training and inference scheme must be designed in order to train the proposed BCsiNet variants well. The most obvious problem is that the sign function introduced in binary FC layer is not derivable. In order to train an end-to-end network, we use a gate filter as [19] to calculate the gradient of sign function. The gate filter is shown in equation (10).

$$\frac{\partial \text{sign}(x)}{\partial x} = \begin{cases} x & \text{if } |x| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Based on (10), the gradient of the scaled sign function in (9) is derived as follows.

$$\begin{aligned} \frac{\partial C}{\partial \mathbf{W}} &= \frac{\partial C}{\partial \mathbf{W}_b} \cdot \frac{\partial \mathbf{W}_b}{\partial \mathbf{W}} \\ &= \frac{\partial C}{\partial \mathbf{W}_b} \cdot \frac{\partial \left(\alpha \cdot \text{sign}(\mathbf{W})\right)}{\partial \mathbf{W}} \\ &= \frac{\partial C}{\partial \mathbf{W}_b} \left(\frac{1}{mn} + \alpha \frac{\partial \text{sign}(\mathbf{W})}{\partial \mathbf{W}} \right), \end{aligned} \tag{11}$$

where C is the cost function of the network, and $\frac{\partial C}{\partial \mathbf{W}}$ is the gradient of the encoder FC layer in back-propagation.

It is also important to notice that all the parameter updates are based on float point weights. Typically, the parameters change in a single iteration is so small that the weights remain the same if they are binarized. Nothing can be learned if the weights are not updated. Therefore, float point weights are essential during training for network convergency. Note that the float point parameters of binary FC layer do no harm to inference since they only exist during training. The detailed training pipeline of BCsiNet is listed in Algorithm 1.

When training completes, the binarized FC weight $\mathbf{B} = \text{sign}(\mathbf{W})$ and the attached scale $\alpha = \frac{1}{mn}\|\mathbf{W}\|_1$ are calculated and saved for inference. Note that α is a float point number. Therefore, the input feature should be multiplied with the binary weight $\mathbf{B}$ in advance for higher inference efficiency. The correct computing order is shown in equation (12).

$$\mathbf{x}_{i+1} = \mathbf{W}_b \mathbf{x}_i + \mathbf{b} = \alpha \mathbf{B} \mathbf{x}_i + \mathbf{b} = \alpha(\mathbf{B}\mathbf{x}_i) + \mathbf{b}, \tag{12}$$

where $\mathbf{x}_i$, $\mathbf{x}_{i+1}$ and $\mathbf{b}$ are input feature, output feature and bias of the binary FC layer, respectively.

IV. Simulation Results and Analysis

A. Experiment Settings

The proposed BCsiNet is trained with two different channel scenarios. The indoor scenario works under 5.3GHz band while the outdoor scenario works under 300MHz band. For fair comparison, all the parameter settings of COST2100 channel model are inherited from CsiNet [3]. The base station adopts a uniform linear array (ULA) system with $N_t = 32$. A FDD system with $N_c = 1024$ is considered and the angular-delay domain matrix is truncated to $N_a = 32$. The training,

Algorithm 1 A training iteration of proposed BCsiNet

Input: Input CSI data $\mathbf{D}^t$ of current mini-batch, current learning rate γ^t, current weights of BCsiNet $\mathcal{W}^t$.

Output: Updated weights of BCsiNet $\mathcal{W}^{t+1}$ and updated learning rate γ^{t+1}.

1: Save the real-value weight of encoder FC layer. $\tilde{\mathbf{W}}^t = \mathbf{W}^t$.
2: Calculate the binary scale α and the binarized weight $\mathbf{W}_b^t$ of encoder FC layer according to (9).
3: Assign the encoder FC layer with the binarized weight $\mathbf{W}^t = \mathbf{W}_b^t$, making the forward propagation binary.
4: Execute the forward and backward propagation, getting the gradient of the binarized encoder FC layer $\frac{\partial C}{\partial \mathbf{W}_b^t}$.
5: Restore the original real-value weight of encoder FC layer for parameter update $\mathbf{W}^t = \tilde{\mathbf{W}}^t$.
6: Calculate the real-value weight gradient of the encoder FC layer $\frac{\partial C}{\partial \mathbf{W}^t}$ based on (11).
7: Update all the parameters in BCsiNet with Adam optimizer. $\mathcal{W}^{t+1} = \textbf{AdamOptimizerStep}(\mathcal{W}^t, \frac{\partial C}{\partial \mathcal{W}^t}, \gamma^t)$.
8: Update the learning rate. $\gamma^{t+1} = \textbf{LRScheduler}(\gamma^t, t)$.

validation and test dataset are independently generated with 100,000, 30,000 and 20,000 channel matrices, respectively.

Xavier initialization is used for both convolution and fully connected layers. Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 1e-7$ is adopted while the mean square error (MSE) loss is applied. The batch size is set to 1000 for faster training. The scheduler proposed in CRNet [11] is used as (13).

$$\gamma = \gamma_{ed} + \frac{1}{2}(\gamma_{st} - \gamma_{ed})\left(1 + \cos\left(\frac{i - N_w}{N - N_w}\pi\right)\right), \quad (13)$$

where N, N_w, i are number of total training epochs, number of warm up epochs and index of current epoch, respectively. Initial lr γ_{st} is $1e-2$ while the final lr γ_{ed} is $5e-5$. For BCsiNet training, we set $N = 2500$ and $N_w = 30$. A linear increasing scheduler is applied for warm up. Note that the training of BNN is trapped into bad local minima occasionally, and the problem can be settled by scheduler rebooting. All the training and inference are based on PyTorch.

B. Performance and Complexity of the Proposed BCsiNet

In order to evaluate the performance of CSI reconstruction, normalized mean square error (NMSE) is used to measure the distance between the original $\mathbf{H}_a$ and the recovered $\hat{\mathbf{H}}_a$.

$$\text{NMSE} = \mathsf{E}\left\{\frac{\|\mathbf{H}_a - \hat{\mathbf{H}}_a\|_2^2}{\|\mathbf{H}_a\|_2^2}\right\} \quad (14)$$

As it is explained in section III-C, we design several variants for the proposed BCsiNet. Table III shows an ablation study among different BCsiNet variants under indoor and outdoor scenario. The compression ratio η is fixed to $1/4$.

As we can see from Table III, the performance of BCsiNet-B and BCsiNet-C is higher than vanilla BCsiNet-A under both indoor and outdoor scenario. Moreover, it is clear that B is relatively dominant for indoor scenario while C works better under outdoor scenario. Besides, ablation study in Table III proves that NMSE performance can be further improved by extending an extra RefineNet block at decoder. The extension is cost free for UE and acceptable for BS like it is mentioned in section III-C.

The detailed memory saving multiples of all three heads are depicted in Fig. 4. It is obvious that the memory saving multiple decays with the increase of compression multiple since FC layer becomes less dominant. However, all three BCsiNet variants achieve over $30\times$ memory saving compared with original CsiNet even when $\eta = 1/32$. Therefore, replacing the vanilla head A with head B or C boosts the performance

TABLE III
NMSE (DB) COMPARISON AMONG BCSINET VARIANTS

BCsiNet Head	Two RefineNets		Three RefineNets	
	Indoor	Outdoor	Indoor	Outdoor
A	-17.25	-8.35	-17.49	-8.78
B	**-19.00**	-9.07	**-20.31**	-9.77
C	-18.32	**-9.20**	-19.00	**-9.93**

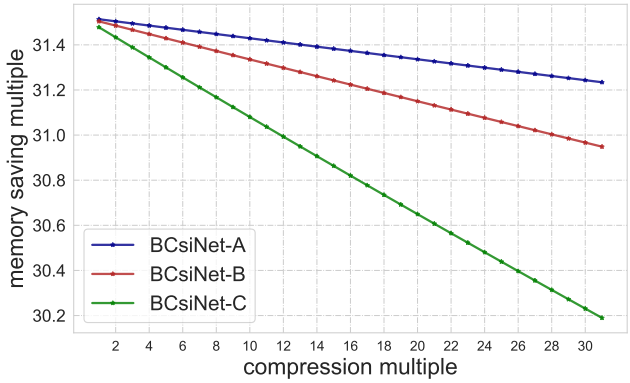

Fig. 4. Memory saving multiple of BCsiNet variants with head A, B and C over original CsiNet with different compression multiple ($1/\eta$).

with little cost. Besides, we can see from Table III and Fig. 4 that head B is more cost efficient than head C.

We denote the vanilla BCsiNet with head A and two RefineNets at BS as "BCsiNet-A2". Similarly, BCsiNet with head B and three RefineNets at BS is denoted as "BCsiNet-B3". Performance comparison among BCsiNet-A2, BCsiNet-B3 and original CsiNet is demonstrated in Table IV.

Experiments show that the encoder of the proposed BCsiNet at UE is over $30\times$ lighter than the original CsiNet. Additionally, the inference speed at UE is around $2\times$ faster since the number of multiplication is significantly reduced. Note that the addition complexity remains the same after FC binarization, for which it is omitted in Table III.

Despite the extremely lightweight encoder, BCsiNet achieves comparable performance over original CsiNet. As a matter of fact, the BCsiNet-B3 scheme outperforms CsiNet with all compression ratio under indoor scenario. For outdoor scenario where channel reconstruction is harder, BCsiNet-B3 dominates only with low compression ratio. Notably, the NMSE performance gap between the lightweight BCsiNet-B3 and the original CsiNet is less that 1dB for high compression

TABLE IV
NMSE (DB) AND COMPLEXITY COMPARISON BETWEEN CSINET AND BCSINET

η	Methods	complexity at UE		NMSE	
		mul[a]	params	indoor	outdoor
1/4	CsiNet	1085K	1049K	-17.36	-8.75
	BCsiNet-A2	37K	**33K**	-17.25	-8.35
	BCsiNet-B3	74K	**33K**	**-20.31**	**-9.77**
1/8	CsiNet	561K	525K	-12.70	**-7.61**
	BCsiNet-A2	37K	**17K**	-12.38	-6.26
	BCsiNet-B3	74K	**17K**	**-12.77**	-6.86
1/16	CsiNet	299K	262K	-8.65	-4.51
	BCsiNet-A2	37K	**8K**	-8.99	-4.17
	BCsiNet-B3	74K	**8K**	**-10.71**	**-4.52**
1/32	CsiNet	168K	131K	-6.24	**-2.81**
	BCsiNet-A2	37K	**4K**	-6.79	-2.69
	BCsiNet-B3	74K	**4K**	**-7.93**	-2.74

[a] "mul" refers to the total number of multiplication.

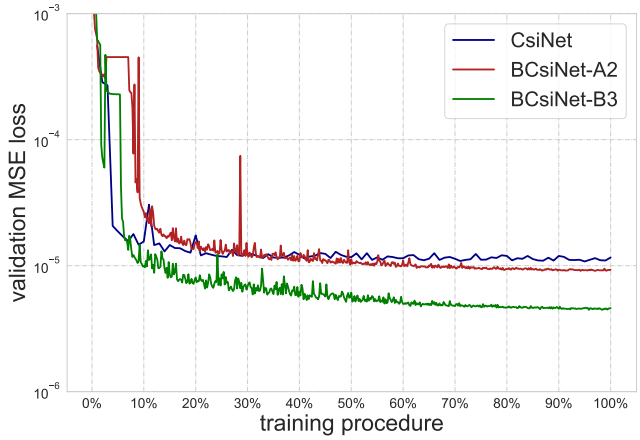

Fig. 5. Validation loss descending trends of BCsiNet-A2, BCsiNet-B3 and original CsiNet. Experiments are based on in door scenario with compression ratio $\eta = 4$.

ratio under outdoor scenario.

When we focus on the performance of BCsiNet-A2 and BCsiNet-B3, it is apparent that the extra complexity in BCsiNet-B3 strengthens the network capacity. Another interesting observation is that parameters size at UE is almost the same for BCsiNet-A2 and BCsiNet-B3 with arbitrary η. This further proves the dominance of FC layer at the encoder.

Finally, we compare the descending trend of validation loss between BCsiNet and CsiNet in Fig. 5. The proposed BCsiNet converges to a lower MSE loss as expected, and the advantage of BCsiNet-B3 is quite conspicuous. Notably, the training of BCsiNet is less stable at early stage due to the gradient deviation from binarization.

V. Conclusion

In this paper, binary neural network technique was applied to CSI feedback task and proved to be effective. After analyzing the complexity bottleneck, a novel feedback network with extremely lightweight encoder named BCsiNet was proposed. Additionally, detailed algorithm of BCsiNet training and inference were designed for better convergency and higher efficiency. Experiments showed that BCsiNet achieved over $30\times$ memory saving and around $2\times$ inference speed up for encoder at UE compared with CsiNet. Moreover, the feedback performance of BCsiNet was comparable with original CsiNet.

Acknowledgment

This work was supported in part by the National Key R&D Program of China under Grant 2017YFE0112300 and Beijing National Research Center for Information Science and Technology under Grant BNR2019RC01014 and BNR2019TD01001.

References

[1] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive mimo for next generation wireless systems," *IEEE communications magazine*, vol. 52, no. 2, pp. 186–195, 2014.

[2] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5g," *IEEE communications magazine*, vol. 52, no. 2, pp. 74–80, 2014.

[3] C.-K. Wen, W.-T. Shih, and S. Jin, "Deep learning for massive mimo csi feedback," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 748–751, 2018.

[4] P.-H. Kuo, H. Kung, and P.-A. Ting, "Compressive sensing based channel feedback protocols for spatially-correlated massive antenna arrays," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2012, pp. 492–497.

[5] T. Wang, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based csi feedback approach for time-varying massive mimo channels," *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 416–419, 2018.

[6] Z. Liu, L. Zhang, and Z. Ding, "Exploiting bi-directional channel reciprocity in deep learning for low rate massive mimo csi feedback," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 889–892, 2019.

[7] Y. Yang, F. Gao, G. Y. Li, and M. Jian, "Deep learning-based downlink channel prediction for fdd massive mimo system," *IEEE Communications Letters*, vol. 23, no. 11, pp. 1994–1998, 2019.

[8] Q. Yang, M. B. Mashhadi, and D. Gündüz, "Deep convolutional compression for massive mimo csi feedback," in *2019 IEEE 29th international workshop on machine learning for signal processing (MLSP)*. IEEE, 2019, pp. 1–6.

[9] H. Ye, F. Gao, J. Qian, H. Wang, and G. Y. Li, "Deep learning based denoise network for csi feedback in fdd massive mimo systems," *IEEE Communications Letters*, 2020.

[10] Y. Sun, W. Xu, L. Fan, G. Y. Li, and G. K. Karagiannidis, "Ancinet: An efficient deep learning approach for feedback compression of estimated csi in massive mimo systems," *IEEE Wireless Communications Letters*, 2020.

[11] Z. Lu, J. Wang, and J. Song, "Multi-resolution csi feedback with deep learning in massive mimo system," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[12] J. Guo, C.-K. Wen, S. Jin, and G. Y. Li, "Convolutional neural network-based multiple-rate compressive sensing for massive mimo csi feedback: Design, simulation, and analysis," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2827–2840, 2020.

[13] X. Li and H. Wu, "Spatio-temporal representation with deep neural recurrent network in mimo csi feedback," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 653–657, 2020.

[14] Z. Cao, W.-T. Shih, J. Guo, C.-K. Wen, and S. Jin, "Lightweight convolutional neural networks for csi feedback in massive mimo," *arXiv preprint arXiv:2005.00438*, 2020.

[15] Q. Cai, C. Dong, and K. Niu, "Attention model for massive mimo csi compression feedback and recovery," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2019, pp. 1–5.

[16] X. Yu, X. Li, H. Wu, and Y. Bai, "Ds-nlcsinet: Exploiting non-local neural networks for massive mimo csi feedback," *IEEE Communications Letters*, 2020.

[17] J. Guo, J. Wang, C.-K. Wen, S. Jin, and G. Y. Li, "Compression and acceleration of neural networks for communications," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 110–117, 2020.

[18] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," in *Advances in neural information processing systems*, 2015, pp. 3123–3131.

[19] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European conference on computer vision*. Springer, 2016, pp. 525–542.

[20] C.-K. Wen, S. Jin, K.-K. Wong, J.-C. Chen, and P. Ting, "Channel estimation for massive mimo using gaussian-mixture bayesian learning," *IEEE Transactions on Wireless Communications*, vol. 14, no. 3, pp. 1356–1368, 2014.

[21] L. Liu, C. Oestges, J. Poutanen, K. Haneda, P. Vainikainen, F. Quitin, F. Tufvesson, and P. De Doncker, "The cost 2100 mimo channel model," *IEEE Wireless Communications*, vol. 19, no. 6, pp. 92–99, 2012.

[22] H. Chen, Y. Wang, C. Xu, B. Shi, C. Xu, Q. Tian, and C. Xu, "Addernet: Do we really need multiplications in deep learning?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1468–1477.