ArguteDUB: Deep Learning based Distributed Uplink Beamforming in 6G-Based IoV

Xingrui Yi, Jianqiang Li, Yutong Liu, Linghe Kong, Ying Shao, Guihai Chen, *Fellow, IEEE,* Xue Liu, *Fellow, IEEE,* Shahid Mumtaz, and Joel J. P. C. Rodrigues, *Fellow, IEEE*

Abstract—In the last decade, MIMO spatial multiplexing and distributed beamforming play a significant role in improving data throughput through cooperative transmission. It has been widely used in wireless communication, especially in 6G. However, the distributed uplink beamforming is still an open problem in highly dynamic environments. However, the proposed 6G technology represents the further integration of deep learning and wireless communication. In this paper, we propose Argute Distributed Uplink Beamforming (ArguteDUB), which uses a feedback algorithm with an offline-trained deep learning model to implement highly dynamic distributed uplink beamforming for the Internet of Vehicles (IoV) in 6G. Specifically, each vehicle enables the base station (BS)/access point (AP) to separate different channel state information (CSI) by inserting orthogonal sequences into the sending data. The BS adopts deep learning to filter the noise and predict the beamforming weight to achieve phase synchronization. Unlike traditional distributed uplink beamforming, ArguteDUB can be adapted to the highly dynamic time-varying channels. The simple network structure ensures the fast response of ArguteDUB. In addition, we make ArguteDUB Orthogonal Frequency Division Multiplexing (OFDM) compatible so that it can be easily deployed in 6G networks. Our evaluation shows that ArguteDUB has an signal-to-noise ratio (SNR) gain of about 5dB to 5.3dB over the single vehicle transmission mode.

Index Terms—Distributed uplink beamforming, Internet of Vehicles, highly dynamic, deep learning, 6G

1 Introduction

INTERNET of Vehicles (IoV) is a network constructed by communications among intelligent vehicles, which has prospects on solving traffic congestion, vehicle accidents, and many other problems. As one of the main techniques in 6G, distributed uplink beamforming performs as a collaborative data transfer scheme, which can improve data throughput in IoV. Unlike the traditional one-to-one transmission mode in IoV, such multi-vehicle cooperative distributed transmission mode can deal with noise interferences and insufficient transmission distance. It is a technology that a group of vehicles can emulate an antenna array by transmitting a common message to the intended base station (BS) [2]. Here, we start to use distributed beamforming instead of distributed uplink beamforming. All the distributed beamforming mentioned below is uplink. To build an efficient distributed beamforming for 6G-based IoV, the main challenge is how to coordinate the signals from each node so that the transmission reaches the desired destination at the same phase and time. Fortunately, in the context of 6G and smart hardware, we are able to incorporate deep learning techniques to come up with further solutions.

The basic principle of distributed beamforming is to improve the transmission efficiency of signals by superimposing signals to synthesize signals with higher signal-to-noise ratio (SNR). Unlike MIMO technology, which enhances channel bandwidth through multiple channels, the goal of distributed beamforming is to improve the power of effective signals through the collaboration of multiple channels, thereby resisting noise interference and increasing transmission distance. It is well known that a received signal can be considered as a composite signal of an effective signal and noise signal. If we add N signals in this way, since the effective signals are the same, their correlation is 1, and their power can be increased by N^2 times. For noise signals, since noise signals are random signals, they can be considered uncorrelated, and the power after superimposing is N times, so the SNR can theoretically be increased by $10 \log N$ dB. The significance of this collaborative transmission system is that it can help low-power transmitters transmit signals to a greater distance and resist stronger noise interference. It is of great importance in high noise environments and field

This work was supported in part by National Key Research and Development Program of China 2020YFB1710900, NSFC grant 62141220, 61972253, U1908212, 62172276, 61972254, the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, Shanghai Science and Technology Development Funds 23YF1420500, Open Research Projects of Zhejiang Lab No. 2022NL0AB01, FCT/MCTES through national funds and when applicable co-funded EU funds under the Project UIDB/50008/2020, Brazilian National Council for Scientific and Technological Development - CNPq, via Grant No. 313036/2020-9. (Corresponding author: Linghe Kong and Ying Shao.)

- *This paper is extended from our conference version [1].*
- *Xingrui Yi, Yutong Liu, Linghe Kong, and Guihai Chen are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (E-mail: yxr374@sjtu.edu.cn; linghe.kong@sjtu.edu.cn; isabelleliu@sjtu.edu.cn; gchen@sjtu.edu.cn).*
- *Jianqiang Li is with Beihang University, Beijing, and Shanghai Academy of Spaceflight Technology, Shanghai(E-mail: LJQ0124@buaa.edu.cn).*
- *Ying Shao is with Shanghai Technical Institute of Electronics and Information, Shanghai, China (E-mail: dzxshy@163.com)*
- *Xue Liu is with the School of Computer Science, McGill University, Montreal, QC H3A 2A7, Canada (E-mail: xueliu@cs.mcgill.ca).*
- *Shahid Mumtaz is with the Department of Applied Informatics Silesian University of Technology, Akademicka 16 44-100 Gliwice, Poland Nottingham Trent University, Engineering departement (E-mail: smumtaz@av.it.pt).*
- *Joel J. P. C. Rodrigues is with is with College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266555, China, and Instituto de Telecomunicações, 6201-001 Covilhã, Portugal (E-mail: joeljr@ieee.org).*

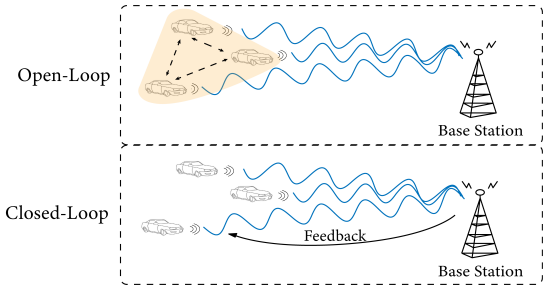


Fig. 1. Two types of distributed beamforming algorithm.

exploration and rescue scenarios.

Generally, the distributed beamforming algorithm can be divided into open-loop and closed-loop, as shown in Figure 1. The open-loop algorithm performs as local coordination among vehicles and calculates the additional phase shift caused by antenna distribution through the acquired position information. After obtaining geometric information and achieving phase synchronization among local vehicles, the open-loop algorithm transforms the problem into a conventional beamforming problem for the antenna array [3]. The advantage is that the direction of the beamforming can be controlled arbitrarily to meet the corresponding requirements. The closed-loop algorithm is also known as the feedback algorithm [4], [5], [6], which aims to help each vehicle adjust its phase using feedback information from the BS. This type of algorithm does not need extra information interactions, such as relevant position or phase information. Compared with the closed-loop algorithm, the open-loop algorithm requires abundant interactions among vehicles. Specifically, the relative positions of each vehicle should be accurately located at each timestamp. The phase of oscillator among vehicles should also be synced, which is a wiener process and impractical in the highly dynamic environment of IoV. Thus, our proposed system adopts the closed-loop algorithm to realize distributed beamforming in highly dynamic environments, which greatly reduces the amount of data interaction among vehicles.

Recent closed-loop researches designed algorithms based on bit feedback to tell the vehicle the next phase adjustment direction [6], [7]. Since bit feedback algorithms are based on continuous phase adjustment, the iterative strategy of continuous positive and negative feedback has been designed to solve the convergence rate problem. Bit feedback algorithms perform well in a static environment. However, high noise and highly dynamic will seriously affect their performance. Due to the rapid change of channel information, the iteration process of the bit feedback algorithm is difficult to keep up with the change of the channel, which leads to serious interference in the direction selection of each phase iteration. Thus, the existing closed-loop algorithms do not work correctly in the low SNR and highly dynamic channel, which we will demonstrate with experiments in Sec.6.

In this paper, we propose a novel closed-loop method based on deep learning called Argute Distributed Uplink Beamforming (ArguteDUB), as shown in Figure 3. ArguteDUB works by inserting orthogonal sequences in the transmit data. Similar to the function of the pilot, these inserted sequences can help to extract the desired phase offset. However, they also have different emphasis. The orthogonal sequence adopted in this paper has the advantage of separating the channels of multiple transmitters effectively, and utilizing the information to calculate and predict the beamforming weights. Unlike traditional pilot design, which prioritizes frequency offset correction and channel estimation in one-to-one transmission, this sequence design focuses on a multi-to-one transmission scenario, aiming to obtain the beamforming weights more accurately with fewer iterations. Next, the BS uses the deep learning strategy to filter noise and predict the beamforming weight of each vehicle then feedback it back. Generally speaking, the time series prediction task requires us to predict the next m time slots through the known n time slots of the current sequence. However, the prediction accuracy of BS and further transmission can be greatly degraded due to the influence of channel noise. In particular, if the wrong beamforming weights are used for transmission, the superposition of multiple waves with different phases will degrade the gain highly. In severe cases, the superposition of peaks and valleys will weaken the amplitude of the signal and not only fail to achieve gain, but also seriously affect the data transmission. To reduce the effect of noise and ensure prediction accuracy, ArguteDUB preprocesses the received information before prediction.

To sum up, our contributions are as follows:

1) We present the design of ArguteDUB, which uses deep learning based closed-loop algorithm that can filter the noise carried in the raw phase offset information and predict the beamforming weights well. ArguteDUB achieves uplink bandwidth scaling in a highly dynamic and noisy environment that is beyond the reach of existing works.
2) We make ArguteDUB OFDM compatible and propose interpolation algorithm to improve the efficiency of data processing.
3) We build a simulation of a multi-vehicle cooperative transmission system. The result shows that under a highly dynamic environment, ArguteDUB has an SNR gain of about 5dB to 5.3dB over the single vehicle transmission mode.

2 Problem Statement

In this section, we present the relevant problem statement for this paper. We show our channel model followed with problem formulation.

The phase offset of each signal received by the BS is mainly due to two points: the difference phase of each vehicle's local oscillator, and the channel response between different vehicles and the BS. Thus, we consider a cluster of N cooperating vehicles which want to send the same message to a distant destination. Then, the received signal at the BS in the i-th timeslot can be expressed as:

$$y[i] = \sum_{n=1}^{N} \sqrt{P_{T_n}} a_n[i] e^{j(\psi_n[i]+\gamma_n[i]+\phi_n[i])} x_n[i] + w[i], \quad (1)$$

where $x[i]$ represents the transmitted signal. $a_n e^{j\psi_n}$ represents amplitude gain from the n-th vehicle; γ_n is the phase offset from vehicle n to the BS; $e^{j\phi_n}$ is the beamforming weight we need to control; and $w[i]$ is an Additive White Gaussian Noise (AWGN) at the distant BS. We assume that

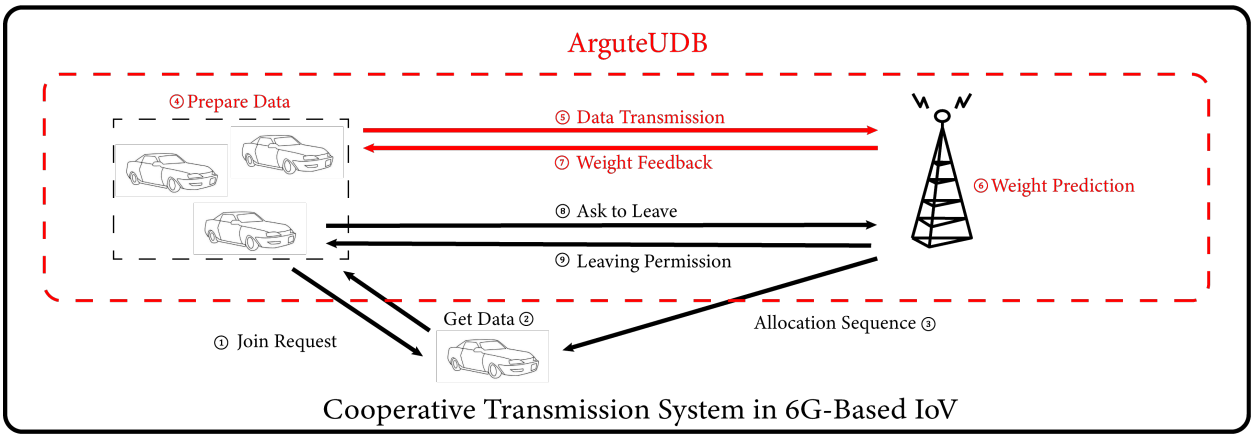


Fig. 2. Distributed cooperative transmission system contains three parts: the join part, the transmission part, and the exit part. ArguteDUB focuses on the transmission part.

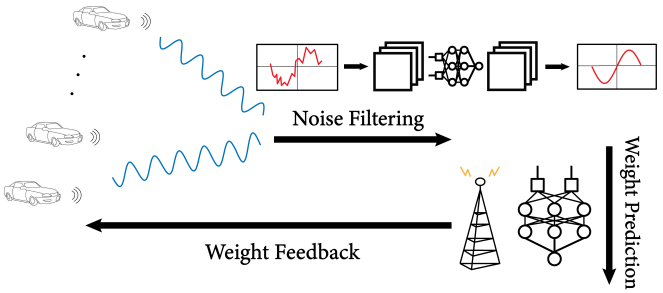


Fig. 3. ArguteDUB consists of three main parts: Data Preprocessing, Weight Prediction and Weight Feedback.

each vehicle uses a unit of transmitted power (e.g., $P_{T_n} = 1$), and transmitted signal can be set at a constant value during training stage (e.g., $x[i] = 1$).

Assume that two signals carry the same information with different AWGN. If these two signals can be aligned and superimposed, then since the correlation of the real signal is 1, the superimposed amplitude doubles and the power becomes a squared multiple, while the noise power is simply summed. Therefore, we want all transmitted signals to be aligned so that a higher SNR signal can be received at the BS. Thus, to achieve distributed beamforming, we need to adjust the beamforming weight $e^{j\phi_n}$ to synchronize each vehicle phase, in other words, to maximize the receive signal strength (RSS), which can be expressed as:

$$RSS[i] = \left| \sum_{n=1}^{N} a_n[i] e^{j(\psi_n[i]+\gamma_n[i]+\phi_n[i])} + w[i] \right|. \quad (2)$$

Because of the special nature of distributed beamforming that the addition of two signals with a small phase offset can also increase RSS, as long as the phases between vehicles remain in a relatively aligned state, despite the existence of multiple fading paths, the RSS will eventually increase.

ArguteDUB is the system we present for the transmission part as shown in Figure 2. Thus, ArguteDUB needs to solve the following problems:

1) Noise and Doppler Shift on phase synchronization.
2) Information delay problem caused by feedback algorithm.
3) The trade-off between prediction accuracy and latency.

3 Raw Data Acquisition

In this section, we first show how we obtained the raw phase offset information. Next, we give our analysis, and finally we discuss the limitations of the raw phase offset to show the importance of weight prediction.

3.1 Phase Offset Extraction

Unlike previous closed-loop methods, we want to get the beamforming weight in constant time. Our key insight is to extract the phase offset information via differential measurement, by sending orthogonal data to separate different channels. Using orthogonal sequence on different vehicles, the BS can extract each vehicle's raw phase offset information from the receiving signal.

There are many options for orthogonal sequences such as discrete Fourier transform (DFT). It is important to note that the length of the sequence we choose should be greater than the number of vehicles so that the sequence is orthogonal to each other. Assume that we use DFT to generate the sequence and we have N vehicles in the system, and L represent the length of the sequence, the set of the sequence can be written as a matrix:

$$\mathbf{X} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_N \end{bmatrix}, \quad (3)$$

where $\mathbf{a}_n$, $n \in \{1, 2, \cdots, N\}$ are orthogonal vectors of length L and $\mathbf{a}_n^H \mathbf{a}_n = L$. We send orthogonal sequences through different vehicles, according to Equation 1, the matrix form of observed signal at the BS can be expressed as follows:

$$\mathbf{y} = \mathbf{X}\mathbf{h} + \mathbf{w}, \quad (4)$$

where $\mathbf{h}$ is an $N \times 1$ channel complex vector which contains the phase offset information $arg(\mathbf{h})$ and $\mathbf{w} \sim \mathcal{CN}(0, N_w \mathbf{I})$ is AWGN. Thus, We can use the least square estimation method to estimate our channel vector from the receiving signal:

$$\hat{\mathbf{h}} = \left(\mathbf{X}^H \mathbf{X}\right)^{-1} \mathbf{X}^H \mathbf{y}. \quad (5)$$

Obviously, the rank of $\mathbf{X}$ is N, so that for every vehicle, we can compute the inner product of the corresponding rows of

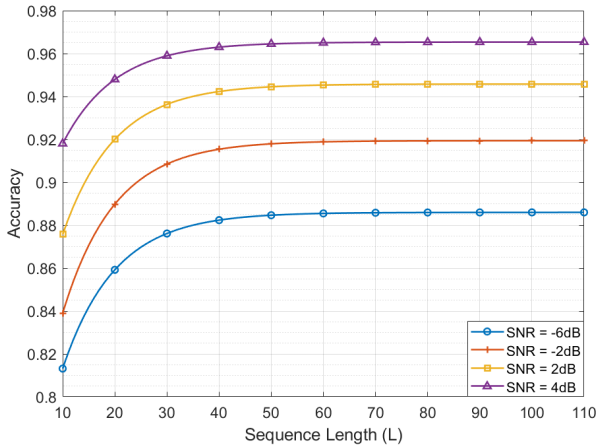


Fig. 4. We tested the effect of sequence length on the estimation accuracy under different SNR conditions. Note that we use 12 independent vehicles, and the length of the sequence increases from 12 to 112.

the matrix $\left(\mathbf{X}^H\mathbf{X}\right)^{-1}\mathbf{X}^H$ to obtain the corresponding CSI. Note that $\mathbf{a}_n^H\mathbf{a}_n = L$, thus we have $\left(\mathbf{X}^H\mathbf{X}\right) = L\mathbf{I}$ and

$$\hat{\mathbf{h}}_n = \frac{1}{L}\mathbf{a}_n^H\mathbf{y}, \tag{6}$$

where $\hat{\mathbf{h}}_n$ represents each channel response. Therefore, it can be noted that each vehicle can obtain the corresponding beamforming weight independently. More importantly, unlike the previous closed-loop algorithm, the BS can separate the information of each vehicle from the received data, and it will not be difficult to determine the direction of phase iteration due to the entry and exit of vehicles as in the previous algorithm.

3.2 Raw Data Limitation

For the problem of parameter estimation, Cramer-Rao [8] establishes a lower bound for the variance of any unbiased estimator:

$$C_r = -\frac{1}{E\left[\frac{\partial^2 \ln P(\mathbf{y}|\mathbf{h})}{\partial \mathbf{h}^2}\right]} = \frac{1}{L}N_w\mathbf{I}. \tag{7}$$

By calculating the lower bound of the variance of the unbiased estimation, we can know the bound of the covariance for each vehicle:

$$Var\left(\hat{\mathbf{h}}_n\right) \geq \left(N_w\left(\mathbf{X}^H\mathbf{X}\right)^{-1}\right)_{n,n} = \frac{N_w}{L}. \tag{8}$$

In Figure 4, we can easily find the impact of noise on the extract phase offset data. As the length of the sequence increases, the improvement in accuracy decreases. Although the variance of parameter estimation is inversely proportional to the length of the sequence, we cannot reduce the impact of noise by simply increasing L. The increase of sequence length is accompanied by the change of transmission time. Because the channel changes dynamically during transmission, the increase of transmission time will affect the system performance. Most importantly, the packet size is finite. Therefore, we need to further process the phase offset data to reduce the effect of noise.

TABLE 1
PERFORMANCE OF DENOISING ALGORITHMS

Method	Bias (Euclidean Distance)
Origin Data	39.80
Savitzky Golay	33.38
Kalman	31.25
Movemean	29.86
Our Method (Deep Learning)	7.36

4 Noise Filtering and Weight Prediction

In the last section, we showed limitation of raw phase offset data. Since the raw data is extracted from the previous round, it is not wise to apply the weight information of the previous round to the next directly. Here, we use the prediction method to predict the weights of the next few rounds of beamforming to improve the accuracy.

Currently, there are many feasible time series prediction methods, most of which are good at predicting the next value of a sequence x based on its previous value. [9], [10], [11], [12]. However, we need to predict the next value of sequence x through sequence y with noise, and these methods are either complicated to calculate or not highly accurate. There is no doubt that training on an unprocessed dataset will cause a series of problems such as overfitting, leading to inaccurate testing results [13]. Therefore, we consider performing rapid denoising processing on the original dataset before making predictions. Figure 5 presents the data processing flow in ArguteDUB to show that how the system works.

4.1 Noise Filtering

The time series denoising methods have been extensively researched in recent times. Traditional methods, such as Savitzky Golay filtering and Kalman filtering, typically employ filters to achieve their objectives. However, these traditional filtering methods have numerous limitations and are unable to learn the underlying features in the data effectively. With the development of neural networks, the implementation of denoising through deep learning algorithms has become more prevalent and has gained popularity in many fields. These methods have several advantages compared to traditional methods, including:

1) Improved accuracy: Deep learning algorithms are able to capture complex and non-linear relationships in data, resulting in higher accuracy in denoising compared to traditional methods.
2) Automated feature selection: Deep learning algorithms are capable of automatically selecting relevant features in the data, making the process more efficient and effective than traditional methods, which relies on manual feature selection.
3) Robustness to outliers: Deep learning algorithms are able to handle outliers in the data, leading to improved robustness compared to traditional methods, which can be sensitive to outliers.

Table 1 illustrates the performance difference between the deep learning algorithm utilized in this study and tradi-

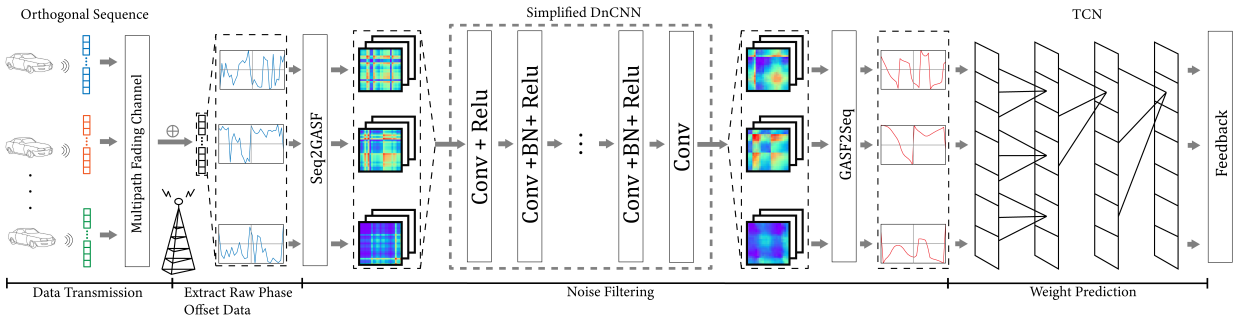


Fig. 5. Illustration of data processing flow in ArguteDUB. In the noise filtering part, we convert the sequence data into 2D images, and use the simplified DnCNN to do noise reduction on the image data, and finally restore the denoised image to the sequence data. Next, we perform sequence prediction through Temporal Convolutional Network (TCN) and feedback the weight to each vehicle.

tional methods (using time series data generated from 400 iterations), and it can be observed that the deep learning algorithm significantly improves denoising performance, enabling subsequent algorithms to make better predictions with its help.

The algorithm presented in this paper was inspired by image denoising algorithms. In the aspect of image processing, the network structure based on Convolutional Neural Networks (CNN) has been continuously optimized and improved and has a very good performance [14], [15]. What's more, image noise reduction is particularly prominent. Therefore, we considered transforming the time series with noise into 2D images and realized a fast noise reduction algorithm by using the excellent performance of CNN with adjustment in computational complexity.

4.1.1 Convert Time Series to 2D Images

In this paper, we use Gramian Angular Summation Field (GASF) [16] to convert the time series to 2D images. GASF encodes time series by mapping time series from Cartesian coordinate to polar coordinate to generate Gramian matrix. In the Gramian matrix, each element is the cosine of the summation of angles. The advantage of GASF is that the temporal relationship is well preserved, and the calculation is simple. Also, it is realizable to restore a 2D image to a sequence. More importantly, processing time series through GASF has a more stable and excellent performance than processing time series directly.

Before GASF, we need to normalize the data. In Sec.3, we obtained the estimates of each channel response $\hat{\mathbf{h}}_n$ through least-squares estimation. We focus on the phase offset of the channel, and rescale it in the interval $[0, 1]$ by:

$$g_n[i] = \frac{angle(\hat{\mathbf{h}}_n[i]) + \pi}{2\pi}. \tag{9}$$

Thus we can represent the rescaled time series G_n in polar coordinates by encoding the value as the angular cosine with the equation below:

$$\phi = \arccos\left(g_n[i]\right), -1 \leq g_n[i] \leq 1, g_n[i] \in G_n, \tag{10}$$

where we use the time stamp as the radius. Next, GASF is defined as follow:

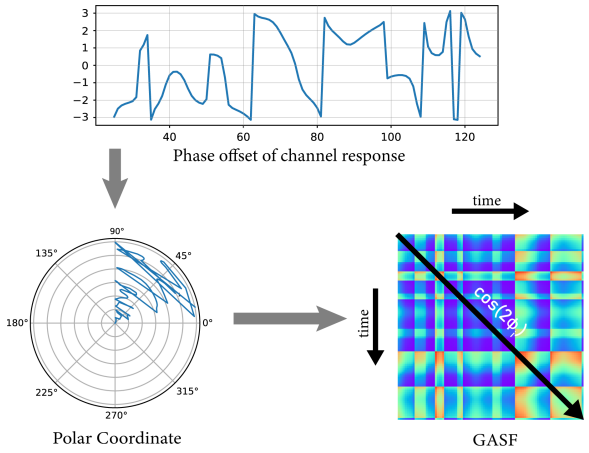


Fig. 6. Illustration of the proposed encoding map of Gramian Angular Summation Fields. In this example we use 120 time slots build the image.

$$\begin{aligned} \text{GASF} &= \left[\cos\left(\phi_i + \phi_j\right)\right] \\ &= {G_n}' \cdot G_n - \sqrt{I - {G_n}^2}' \cdot \sqrt{I - {G_n}^2}. \end{aligned} \tag{11}$$

Figure 6 shows how to convert a time series to 2D images on GASF. Because the mapping functions of $[0, 1]$ rescaled time series are bijections. The main diagonal of GASF, i.e. $\{GASF_{ii}\} = \{\cos\left(2\phi_i\right)\}$ allows us to precisely reconstruct the original time series by

$$\cos(\phi) = \sqrt{\frac{\cos(2\phi) + 1}{2}} \quad \phi \in \left[0, \frac{\pi}{2}\right]. \tag{12}$$

4.1.2 Simplified DnCNN (SDnCNN)

In terms of image noise reduction, there are many different networks, such as Block-matching and 3D filtering (BM3D) [17], Learned Simultaneous Sparse Coding (LSSC) [18], Non-local Centralized Sparse Representation (NCSR) [19], Weighted Nuclear Norm Minimization (WNNM) [20] and Denoising Convolutional Neural Network (DnCNN) [21]. Among them, DnCNN is the most classic. As for the goal we want to achieve, under the condition of achieving a certain accuracy, what we pursue is a lower computational complexity to reduce the computational pressure on the

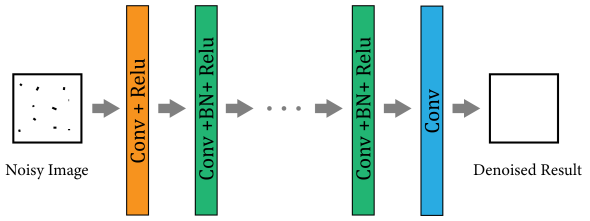


Fig. 7. Illustration of the architecture of the DnCNN. Conv represents the convolution layer, which enables automatic extraction of features. BN represents the batch normalization layer, which is used to avoid the vanishing gradient problem. ReLU is an activation function commonly used in neural networks.

base station, to achieve more efficient and rapid prediction to follow up the changes of the channel.

Figure 7 illustrates the architecture of the DnCNN. DnCNN introduces the residual learning strategy, which can solve the problem of performance degradation to improve performance. Currently, DnCNN is used for suppressing Gaussian noise in images, which can be expressed as:

$$P_y = P_x + P_w, \tag{13}$$

where P_y represents the noisy image which is the sum of the clean image P_x and noise P_w. For DnCNN, the goal is to recover clean image P_x from a noisy observation P_y. DnCNN adopts the residual learning formulation to train a residual mapping $\mathcal{R}(P_y) \approx P_w$. The averaged mean squared error between the desired residual images and estimated ones is chosen to be the loss function:

$$\ell(\mathbf{\Theta}) = \frac{1}{2M}\sum_{i=1}^{M}\left\|\mathcal{R}\left(P_{y_i};\mathbf{\Theta}\right) - \left(P_{y_i} - P_{xi}\right)\right\|_F^2, \tag{14}$$

where $\mathbf{\Theta} = \{W, b\}$ is the parameter of the network, including the weight W and offset b; $\left\{\left(P_{y_i}, P_{xi}\right)\right\}_{i=1}^{M}$ represents M noisy-clean training image pairs. The loss will be small enough after training, and correspondingly, $\mathcal{R}\left(P_{y_i};\mathbf{\Theta}\right)$ will be closer to P_w, then we will have:

$$P_y - \mathcal{R}(P_y;\mathbf{\Theta}) \approx P_x. \tag{15}$$

In this paper, we choose to generate a 2D image every 25 time slots, which has the size of $25 \times 25 \times 3$. The reason why we choose 25 timeslots is to ensure that the image is small enough so that the base station can process the response in a short time. Due to the random nature of dynamic channels, time slots with long time intervals are not highly correlated with each other. Therefore, using more time slots is not helpful for the results. The original DnCNN is mainly used to suppress Gaussian noise in images, which usually requires processing larger size images. For our task, although the channel noise is Gaussian noise, the columns and rows of the image are correlated after being transformed by GASF. Therefore, the noise of the image does not conform to the Gaussian distribution. Moreover, since the size of the image we process is limited, we need to simplify the network to improve the computational efficiency. Thus, we adjust the network structure according to the characteristics of the images.

4.1.2.1 Kernel Size: The convolution layer is mainly used for automatic features extraction and increasing the number of convolution kernels can often extract more features. Note that if a data point in the sequence is biased, then after converted to the corresponding GASF image, the row and column in which it is located will be affected accordingly. To enable the network to learn such a relationship, we use the following configuration. The first convolution layer uses a convolution kernel of size $3 \times 3 \times 1 \times 32$, and the last convolution kernel of size $3 \times 3 \times 32 \times 1$. For the remaining layers, we use convolution kernels of size $3 \times 3 \times 32 \times 32$.

4.1.2.2 Training Set and Test Set: For the dataset, we simulate 12 different vehicles to conduct 10000 times of transmission to generate corresponding 2D images (4800 pieces in total) for training, and test on 3 different vehicles (1200 pieces in total). In terms of transmission parameters, we chose the maximum Doppler shift around 100Hz-300Hz to simulate the dynamic change of the channel and adopted the standard parameters to set the data packet. The specific parameters are listed in Sec.6.

4.1.2.3 Depth: In general, the performance of the deep neural network is better than that of the shallow neural network. However, when the number of layers reaches a certain level, the performance becomes saturated. As the depth of the network increases, the performance does not improve much, while the training time increases dramatically and also causes a longer predicting time. After training the network with different depths when the batch size is 15 and the epoch is 200, we compare the performance of each network. Table 2 shows the performance with different depth of the neural network. According to the observation, although the accuracy of the training set increases with the depth of the network, the improvement on the test set is not significant after the depth is 6. Moreover, we can find that the training time increases faster as the depth. To better trade-off the performance and efficiency, we determined the depth of the network to be 6.

After simplifying, adjusting, and training the network, we combined the network with GASF to preprocess the sequence. We use the sliding window method to process the sequence, and the step of the sliding window can be dynamically adjusted between 1 and k (depends on the prediction step) to match the data processing speed of the base station. When the processing speed of the base station is fast enough, we can appropriately shorten the step size to increase the coincidence degree between windows, to improve the accuracy by processing a certain data point several times. Correspondingly, if the base station processing speed is not enough, we will reduce the step size to reduce the processing times so that the base station can respond quickly when predicting beamforming weight.

Figure 8 shows how the sequence is preprocessed during the operation of the system. We can see that the noise distribution of the sequence with noise is very obvious after it is converted into a GASF image. If the value on the main diagonal has an obvious deviation, the color on the corresponding row and column will also have a relatively obvious deviation. After being processed by Simplified DnCNN (SDnCNN), the corresponding noise is significantly decreased, and the processed image is more similar to the target image. Next, by restoring the corresponding sequence

TABLE 2
NETWORK PERFORMANCE WITH DIFFERENT DEPTHS

Depth	Training Time/Batch [μs]	Train Acc	Train Loss	Test Acc	Test Loss
4	220	0.8866	0.0125	0.8639	0.0171
5	250	0.9188	0.0072	0.8821	0.0140
6	290	0.9422	0.0039	0.8972	0.0114
7	340	0.9529	0.0026	0.9005	0.0105

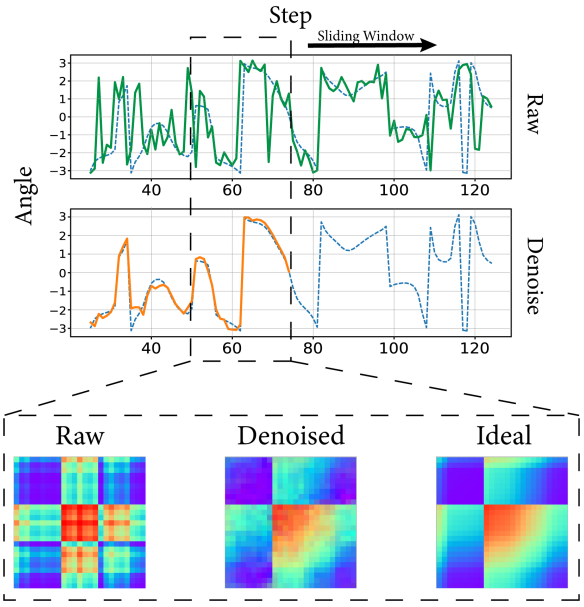


Fig. 8. Illustration of how the sequence is preprocessed. As the sliding window moves, the sequence is progressively optimized, resulting in cleaner data for prediction.

with the value on the main diagonal, we can observe that the processed result is very close to the ideal value.

4.2 Sequence Prediction with TCN

After getting clear sequence data, we need to make a sequence prediction. At present, there are many time series prediction methods like recurrent neural networks (RNN) or long-term short-term memory (LSTM). Despite the many advantages of RNN (i.e. architectures based on RNN), there are two limitations that restrict their applicability in reality. One is the inherent sequential nature in which later time steps must wait for their predecessors to finish during the training and evaluation process, thus eliminating parallelization of the training and evaluation procedure. The other is that as the sequence length grows, RNN become more focused on the local context and are sensitive to the order of words in recent sentences, but ignore the order of words in long-term context. Although LSTM have improved RNN in terms of long-term memory, the limitation of non-parallelizable training remains an issue. In this study, we adopt Temporal Convolutional Networks (TCN) [22] as our method of choice due to its several benefits. Unlike RNN or LSTM, TCN offers parallel processing of time series, instead of the sequential processing implemented by RNN. Furthermore, TCN boasts a stable gradient, mitigating the

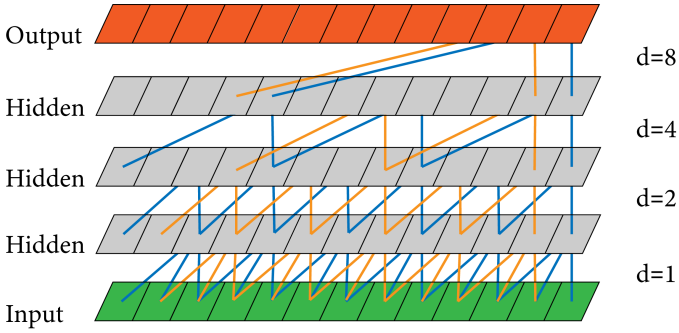


Fig. 9. The architecture of TCN in this paper, which mainly uses causal convolution and dilated convolution. The input of TCN is 25 steps of time slots phase offset data of each vehicle after filtering the noise and it's output is the next 5 prediction steps of phase offset of each vehicles.

potential issues of gradient disappearance and explosion. Additionally, TCN exhibits a low memory footprint and reduces the computational demands on the base station.

Simply speaking, TCN is a variant of CNN, which takes the entire time series as input through convolution and can control different output lengths. TCN mainly realizes the learning of time series features through causal convolution and dilated convolution. Causal convolution means that the value at the time of t of the upper layer only depends on the value at the time of t of the next layer and the value before it. Thus, it is a strict time constraint model. Dilated convolution allows the convolution input to have interval sampling so that the convolution network can obtain a large receptive field with fewer layers. Figure 9 shows the TCN structure used in this paper. It has 5 layers for convolution, with sampling intervals of $1, 2, 4, 8$. The input of TCN has the length of 25, which are previous 25 time slots phase offset data of each vehicle after filtering the noise and it's output is the next 10 prediction steps of phase offset of each vehicles.

5 OFDM Compatibility

Our goal is to extend ArguteDUB to 6G-based IoV applications. The recent 6G new radio (NR) cellular networks adopt sophisticated wireless transmission technologies to enhance the network capacity and can, thus, offer high data rates to IoV applications [23]. As one of the key technologies, Orthogonal Frequency Division Multiplexing (OFDM) has been widely used in various scenarios [24], [25], [26]. The main advantage of OFDM over single-carrier schemes is its ability to cope with severe channel conditions without complex equalization filters. Therefore, combining with OFDM can let ArguteDUB be well used in the 6G network, and at

the same time, it can take advantage of OFDM to increase the transmission bandwidth.

5.1 Orthogonal Sequence Insertion

In the OFDM system, we need to choose the appropriate pilot structure to realize the very important functions such as channel estimation [27]. For the block-type pilot, the pilot signal is continuous in the frequency domain, which is suitable for the case that the channel is slow-changing. Block-type pilot can be used for channel estimation by decision feedback. As for comb-type pilot, the pilots are inserted at regular frequency intervals and are continuous in the time domain. Therefore, comb-type pilot is more able to deal with the rapid transformation of time-varying channels. Comb-type pilot can estimate the channel of each subcarrier by interpolation. In this paper, we choose to use comb-type pilot to better deal with highly dynamic time-varying channels.

The adoption of orthogonal sequence in this paper aims to facilitate the acquisition of beamforming weights of various parts of the OFDM resource block. Due to its special design, it cannot be applied in scenarios with similar specific requirements as other pilot designs like pilot designs in MIMO. To ensure the design's universality, a portion of the original data segment is used to accomplish this task, allowing for compatibility in various scenarios. Thus, after determining the pilot structure, we need to consider how to place the orthogonal sequence into it. Simply, we can select a subset of subcarriers to place orthogonal sequences like comb-type pilot. After the beamforming weight is predicted by the base station, the weight on subcarriers without orthogonal sequence can be calculated by interpolation.

Further, we need to consider the effect of time-varying channels. In a communication system using phase modulation, such as QPSK or QAM, the Doppler Effect causes a change in the phase of the received signal, leading to a rotation of the constellation points on the constellation diagram. The rate of rotation is proportional to the Doppler frequency shift, and it can cause significant degradation in the signal-to-noise ratio of the received signal. The same effect applies to subcarriers within an OFDM resource block. If a subcarrier adopts the same beamforming weight, since a subcarrier occupies a period of transmission time, the Doppler effect of this time will affect the subcarrier. Therefore, we need to divide the weight of the subcarrier into a more fine-grained manner to better combat the Doppler effect in high dynamic environments. In order to improve the accuracy and maintain a certain RB size, we can consider dividing the beamforming weight inside RB. We divide RB into N segments in the time domain, and the subcarrier weight in each segment is determined by the weight predicted by internal orthogonal sequences and interpolation.

5.2 Interpolation

Figure 10 shows the illustration of an OFDM RB. Even in the case of insufficient base station processing speed, we can greatly reduce the amount of computation by using interpolation. Specifically, we only need to select some sequences for weight prediction. Due to the continuity of time domain and frequency domain, we can use interpolation to approximate and complete the values of all subcarriers in each weight area. It is worth noting that the selection of key points is particularly critical when doing interpolation. Although the change of angle is a continuous process, since the angle changes in a cycle within a range $[-\pi, \pi]$, the direct interpolation of two key points is likely to get wrong results, while the correct interpolation should be carried out after several cycles of a certain section. Therefore, how to determine the number of cycles is very important. When selecting key points, we need to select several adjacent points as a combination and determine the number of cycles between combinations through the slope of internal points. Of course, using this method is less accurate, but it can greatly reduce the computation of the base station, ensuring that it can continuously give feedback to each target.

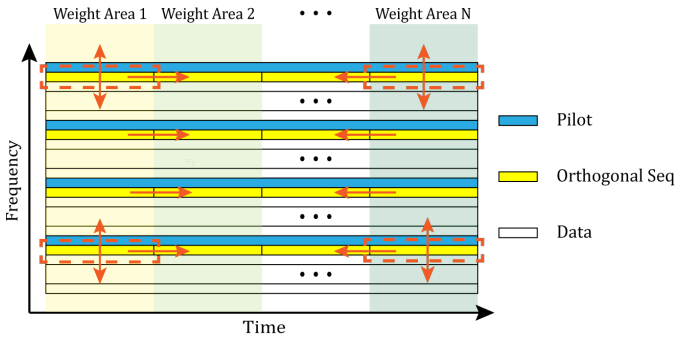


Fig. 10. In an OFDM Resource Block (RB), we can select some sequences for weight prediction and complete them by interpolation.

Many different interpolation strategies can be employed to extend the orthogonal sequence subcarrier channel weights to the rest of the subcarriers, including linear interpolation, second order interpolation, low-pass interpolation, spline cubic interpolation, and time domain interpolation. In fact, the low-pass interpolation algorithm was adopted in our experiments. Research has shown that low-pass interpolation has a significant advantage in combating multi-Doppler frequency shifts and has good results with comb-like insertion methods [28].

6 Evaluation

6.1 Simulation Environments and Parameter Settings

In this section, we provide numerical simulation results to verify the performance of the designed ArguteDUB. We consider a multi-vehicle cooperative transmission mode in a suburban high-speed environment. The choice of channel model is one of our concerns. Multipath fading channels are defined by a combination of multipath delay and maximum puller frequency. Three different multipath fading models (EPA, EVA, and ETU) are defined in the 3GPP standard. In this paper, we use ETU with a high Doppler shift (100-300Hz) as our simulation environment.

After determining the channel model, we consider the design of the simulation experiment combined with 6G. The system parameters shown in Table 3 are used in the simulations and comb type pilots are placed at all OFDM symbols on a subset of 200 equispaced subcarriers.

TABLE 3
PARAMETERS AND SYSTEM SETUP OF SIMULATION

Parameter	Value
Number of Tx antennas	4
Number of Vehicles	5
Number of Rx antennas	64
Bandwidth	20MHz
Number of subcarriers	1200
Number of pilot	200
Sample rate	30.72MHz
Bandwidth	20MHz
Number of subcarriers	1200
Size of FFT	2048
Channel parameter	ETU
Modulation	QPSK
Encoding	Convolutional code
Maximum Doppler shift	10-300Hz

TABLE 4
COMPLEXITY OF NEURAL NETWORKS

Network	Params	FLOPS
Denoising Network (sDnCNN)	132995	166.12M
Prediction Network (TCN)	139393	3.74M

We use an experiment to show why previous distributed beamforming algorithms are not applicable to the high dynamic case. Since the previous algorithms are not applicable in the highly dynamic environments, we use single-vehicle transmission as our baseline and our analysis is based on different measurement studies. First, we compare the performance of different algorithms. Also, we compare the performance of the interpolation-assisted prediction method with the full prediction method. Next, we collect received data and show the performance of different algorithms on BER and FER. In addition, we compare the SNR gain of different methods. To compare the experimental results analytically, we show the SNR gain defined as the SNR achieved by ArguteDUB minus the SNR achieved by the baseline. In the end, we discuss the influence of the number of vehicles on the multi-vehicle cooperative transmission mode. The summary of the evaluation of ArguteDUB and the key findings are as follows:

- ArguteDUB can successfully overcome the impact of the highly dynamic environment.
- ArguteDUB can predict the change of the channel change accurately, in which 90% deviations are no more than 1, 70% deviations are no more than 0.5.
- ArguteDUB can improve the SNR by 5-5.3dB with different types of algorithms in our simulation conditions.
- ArguteDUB can improve performance as the number of participants increases within limits.

6.2 Analysis of Network Complexity and Latency

The hardware platform used in this experiment is as follows:

- CPU: Intel Core i7-8700k (3.7 GHz)
- GPU: NVIDIA GeForce GTX 1070 (6.5 TFLOPS)

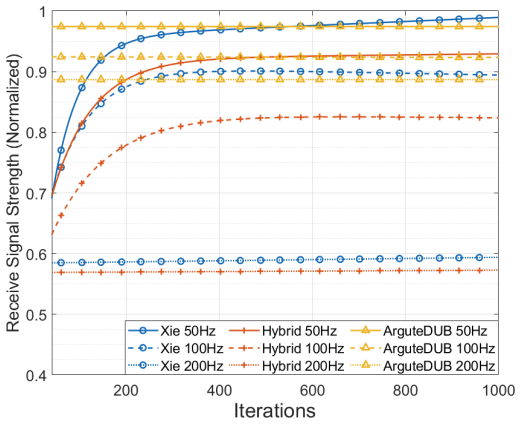


Fig. 11. Comparison the proposed and previous algorithms with time-varying channels. Both Xie [7] and Hybrid [6] can not work well in the highly dynamic environment.

- Memory: 16G (2400 MHz)

The parameters and computational volume of the denoising network and the prediction network used in this paper are shown in Table 4. The parameters of the two networks are basically the same, but the computational volume required by the denoising network is much larger than that of the prediction network due to its high input data dimension, however, they are both at the level of MFLOPs. According to the theoretical calculation of GPU computing power and network complexity, the denoising and prediction time required for data corresponding to an orthogonal sequence is approximately 260μs. However, due to factors such as memory access and data exchange, the actual measured time is around 300μs. For an OFDM resource block, the time required ranges from 1.2ms to 600ms, depending on the degree of interpolation employed. Additionally, due to the presence of prediction algorithms, the data computed in each iteration can provide weighting data for subsequent rounds of information exchange.

In general, the OFDM resource block duration can range from a few microseconds to several milliseconds, depending on the frequency band and the system bandwidth. The time interval between OFDM resource blocks is known as the inter-block time, and it is typically in the range of a few microseconds to several milliseconds. Therefore, under the current experimental platform with a single entry-level GPU, the system is capable of providing the required computational power for normal operation with interpolation methods. Typically, on the server side, the huge computational power is provided by a cluster of multiple professional GPUs, which can completely handle the task with hundreds of times improvement.

6.3 Performance Compared with Previous Works

We compare the performance of different algorithms and demonstrate that the previous algorithms do not work in a highly dynamic environment. Figure 11 shows the RSS of each algorithm at under different Doppler shifts. All algorithms work well at low Doppler shift of 50 Hz. As the Doppler shift increases, the drawbacks of the previous

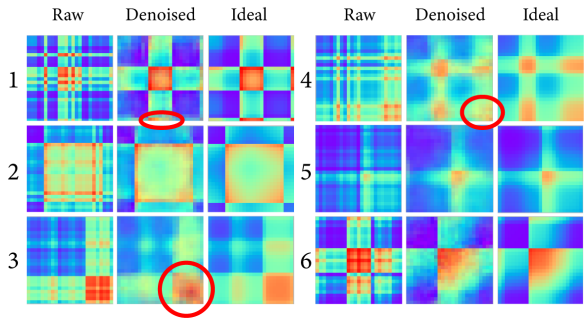


Fig. 12. SDnCNN results. After the network processing, such noise is mostly eliminated, but can't deal well with some places where the noise is very serious.

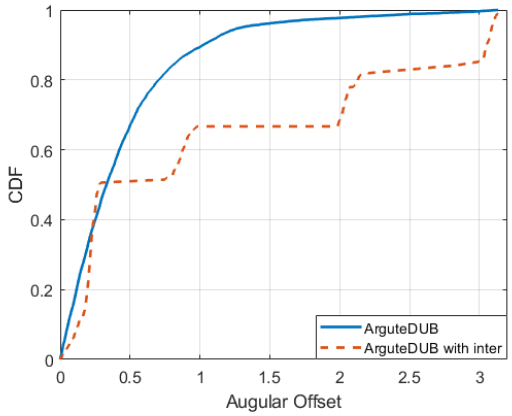


Fig. 13. Angular offset with different method. Since a large number of points are calculated through interpolation, if there is a certain phase error in the value of sampling points, it will lead to a series of points with phase error.

algorithms start to appear. The improvement in RSS is not significant because the update rate of the bit feedback does not catch up with the rate of change of the time-varying channel. When the Doppler shift reaches 200 Hz, the previous algorithm does not work properly. However, our method works well because we do rich feedback with prediction.

We have noticed that under low Doppler frequency shift conditions, the traditional method obtains a slightly higher signal intensity compared to the method proposed in this paper, due to the fact that it receives less interference in low Doppler frequency shift, and its iteration speed can surpass the rate of change of time-varying channels, thus yielding good results. Conversely, in this situation, the operation of the prediction network in the proposed method is unnecessary, which results in redundant computations that would affect accuracy. Nevertheless, in order to ensure uniformity in testing, the prediction module was still retained. However, the proposed method in this paper has the advantage of ensuring high signal strength from the beginning of the transmission without requiring a large number of iterations, thus saving a significant amount of time.

6.4 Performance of Prediction and Interpolation

ArguteDUB's key algorithmic component is to find an appropriate beamforming weight utilizing prediction. Therefore, the prediction accuracy determines the performance of our system. Here, we will show the accuracy of the prediction algorithm for Angle prediction and the loss of accuracy after using interpolation for velocity optimization. Figure 12 shows the difference between the image processed by SDnCNN and the image generated by the ideal value. We can see that since the noise of the original image associates rows and columns, after the network processing this noise is mostly eliminated, but unsmooth transitions can still be seen at the edges of each pixel. In addition, in some cases, such as the red circle in the figure, SDnCNN could not handle this situation well because the values of the original image deviate significantly from the ideal values and there are no good pixels around as a reference. This mostly happens when there is a significant deviation in the continuous part of the sequence. Due to the construction of the GASF matrix, there are large deviations in the whole block of pixels, which makes it difficult for the network to capture the correct information from it.

Figure 13 shows the difference between the Angle values obtained in different ways and the ideal values. We can see that the Angular offset distribution of the full prediction method is relatively concentrated, 70% of the values are less than 0.5 from the accurate value, and 90% of the values are less than 1. But as for the interpolation strategy, 50% of the value less than 0.25, which is a good performance, but the other 50% is greater than 1, which cause bad performance.

6.5 SNR Improvement

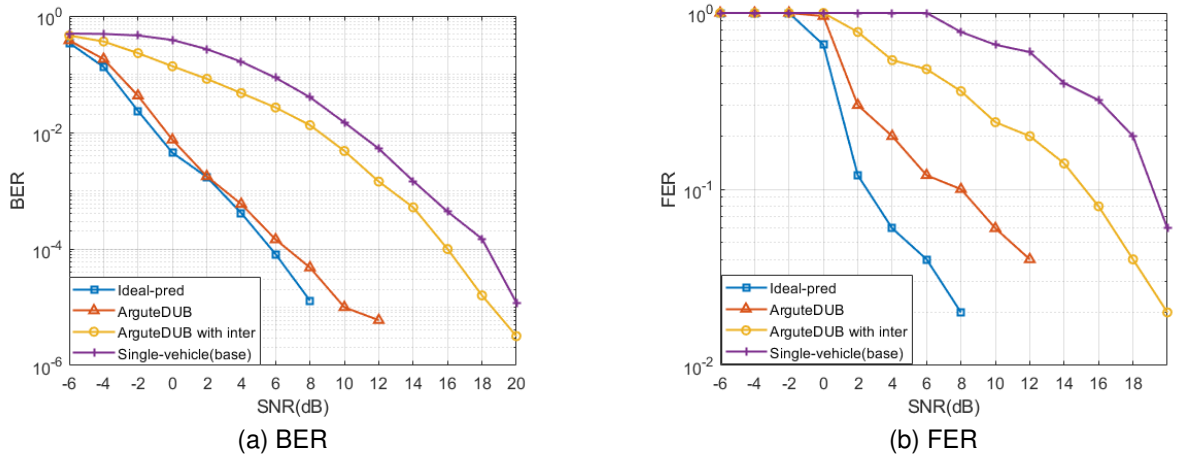
To further understand distributed beamforming spatial multiplexing gains, we analyze the SNR gain received from the base station. By comparing with single-vehicle transmission mode, we obtain the gain of multi-vehicle transmission mode when SNR = 0dB. Figure 16 shows the SNR gain with different transmission mode. About 90% of the values are between 5dB and 5.3dB. It can be found that there is a gap of about 1dB between the interpolation and the ideal prediction, while the performance of the full predictive mode and the ideal prediction mode is closer. The Maximum Doppler Shift has some influence on the SNR gain, but the experiment shows that the effect is not obvious. This confirms that ArguteDUB has a significant SNR gain effect.

6.6 Performance of Transmission

We first compare the bit error rates of different algorithms in different environments. Note that we uniformly selected the 5-vehicle cooperative transmission mode for the test, and the single-vehicle transmission mode was taken as the baseline.

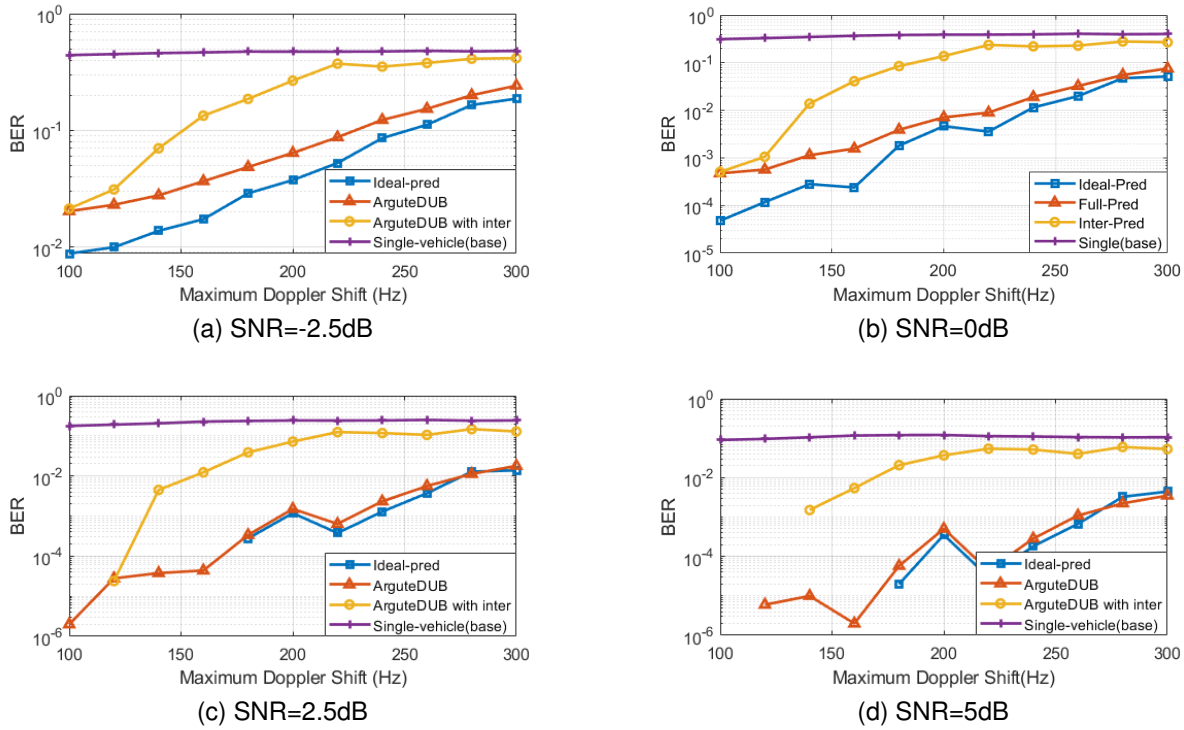
6.6.1 Performance Under Different SNR

First of all, we are concerned about the effectiveness of the multi-vehicle cooperative transmission model. The easiest way to do this is to measure the bit error rate (BER) and frame error rate (FER) of data transmission. We measure the effect of different SNR on BER when the Maximum Doppler Shift is 200Hz. We compared the single-vehicle transmission mode(baseline) with the full prediction mode



(a) BER (b) FER

Fig. 14. BER and FER under different SNR. Here we use 200Hz as Maximum Doppler Shift.



(a) SNR=-2.5dB (b) SNR=0dB

(c) SNR=2.5dB (d) SNR=5dB

Fig. 15. BER with different Maximum Doppler Shift in the range of 100Hz to 300Hz.

and the interpolation prediction mode. Figure 14 shows the performance of the different algorithms. Although the interpolation prediction mode generates many phase errors, it can still improve transmission efficiency. A lesson can be drawn from this: although misalignment of phases in distributed beamforming can have a negative performance impact, the system can have a high tolerance for this.

Furthermore, we can find that the full predictive mode has a very good performance. However, this requires the hardware to be able to perform high-speed computation. The general base station without optimization is difficult to achieve this performance. At the same time, the interpolation prediction mode is more reasonable. By losing some precision in exchange for a significant reduction in computational complexity, real deployment can be achieved. In the previous subsection, we analyzed the interpolation method with significant Angular offset, but it can still improve performance over the baseline. Moreover, by adding interpolation sampling points, its performance can be further improved, depending on the actual computational efficiency of the BS.

6.6.2 Performance Under Different Maximum Doppler Shift

We also investigate the effect of the Maximum Doppler Shift on the performance of different algorithms. With the increase of Maximum Doppler Shift, the performance of the algorithm decreases differently. Among them, the interpolation strategy is the most sensitive to Doppler shift.

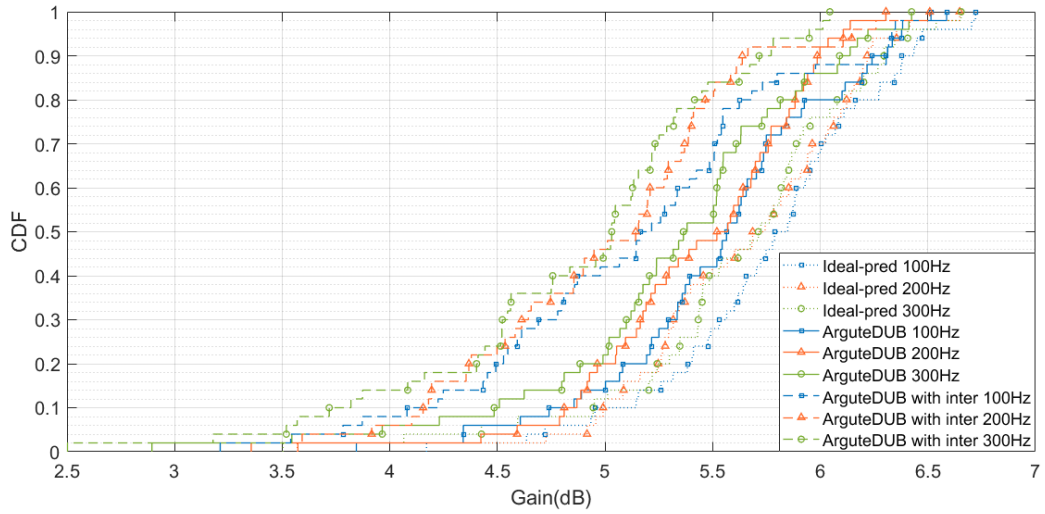


Fig. 16. SNR gain over the baseline with different Maximum Doppler Shift.

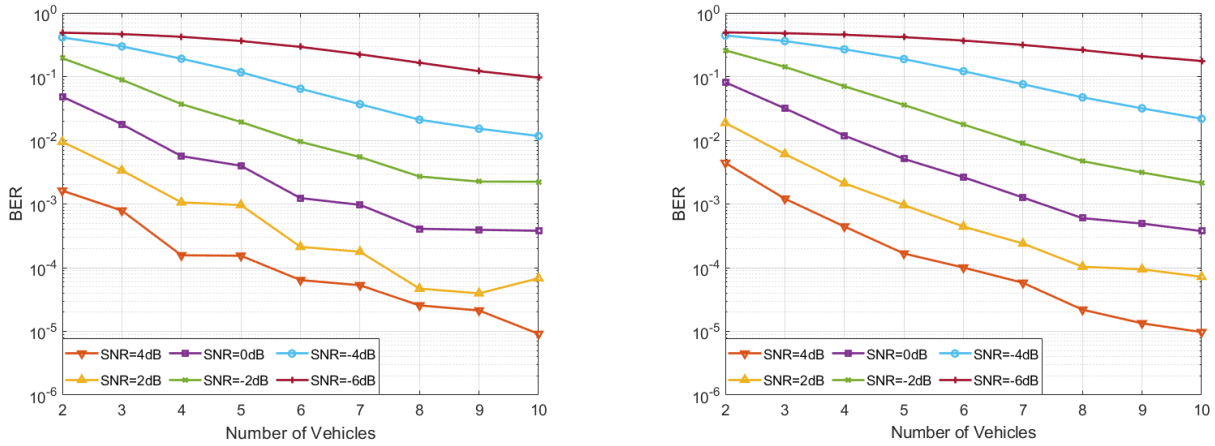


Fig. 17. The performance under different number of vehicles.

When the Maximum Doppler Shift is 100 Hz, the interpolation strategy can maintain the same performance as the full prediction mode. However, with the increase of the Maximum Doppler shift, its performance is seriously affected especially when the Maximum Doppler Shift is around 150Hz to 200Hz. The reason is that with the increase of Doppler Shift, the degree of phase change in RB increases, and the decrease of prediction performance leads to the increase of sampling point error of the interpolation strategy.

6.7 Number of Vehicles

We discuss the effect under the different number of vehicles participating in the ArguteDUB. More does not necessarily mean better, which involves the following factors: the more vehicles we have, the longer orthogonal sequences we should use, making it more time to transmit a packet; each vehicle's transmission will be affected by the noise, it is impossible to perfectly predict the beamforming weight. Thus, no matter how many vehicles carry out the cooperative system, there will still be errors.

We select 2 to 10 vehicles to form different cooperative systems for transmission tests. Figure 17 shows the transmission BER under different numbers of vehicles. Regardless of ideal prediction and truly prediction, we can find that BER decreases obviously within the range of 2 to 6 vehicles, which proves that appropriately increasing the participants can improve transmission efficiency. However, when the number is greater than 7, such improvement is obsolete. Moreover, in some high SNR situations, the transmission efficiency of the system is reduced.

7 Related Work

Our work presents a unique direction where we present a feasible scheme of highly dynamic beamforming. Our work is closely related to multiple avenues as follows:

7.1 6G Intelligent

In recent years, with the development of deep learning and intelligent hardware, the combination of wireless and Artificial Intelligence (AI) is a major research of 6G. AI helps many subjects solve problems that are difficult to deal with traditional methods. For example, in 6G vehicle network, deep learning makes autonomous driving distance control under more efficiency [29]. Also, it can realize intelligent signal detection methods for OFDM systems in 6G [30].

7.2 Closed-loop Distributed Beamforming

R. Mudumbai [5], S.Song [31], et al. proposed different forms of base-station based feedback algorithms, but they had different limitations. Among them, R. Mudumbai proposed a one-bit feedback control method [5] to simplify feedback information and save control mechanisms. In this context, other authors also discuss the influence of Gaussian white noise on the one-bit feedback algorithm and analyze that the performance stability of the algorithm is insufficient [32]. In terms of the algorithm of S.Song [31], the algorithm makes more effective use of one-bit feedback information and improves the convergence speed. However, the algorithm shows the instability of performance and cannot guarantee a relatively fast convergence speed every time. To overcome the instability of the algorithm, subsequent researchers proposed a hybrid algorithm by using continuous positive and negative feedback to improve the performance of the traditional closed-loop feedback algorithm [7]. However, the algorithm is easy to be disturbed by noise, which leads to poor performance, and it is difficult to achieve a good effect in the environment of multipath fading. Also, although the algorithm can adapt to lower dynamic time-varying channels, the performance will highly degrade or even fail in a highly dynamic environment. The advantage of ArguteDUB is that it can deal well with the interference of time-varying channels, noise, and multipath fading. More importantly, ArguteDUB can support the dynamic entry and exit of vehicles, which makes our system more flexible and adaptable to more complex scenarios.

7.3 Cooperative Transmission

Cooperative transmission is widely used in various fields of wireless communication like Internet of Things(IoT), Cellular Networks, satellite communication systems, and etc [33], [34], [35], [36] to concentrate energy and improve transmission distance and performance. We propose a cooperative transmission system based on IoV to improve the efficiency of uplink transmission using distributed beamforming.

8 Conclusion and Future Work

We present a novel distributed beamforming framework to deal with highly dynamic time-varying channels in 6G-based IoV, named as ArguteDUB. It is composed of GASF domain and deep learning methods, which can predict the beamforming weight well and achieve the gain of SNR. Furthermore, we present the design and simulation of ArguteDUB combined with OFDM to optimize the bandwidth. We believe this work paves a way forward for highly dynamic distributed beamforming-based wireless communications and networking especially in 6G-based IoV. To further improve our work in the future, some possible future directions are listed below.

1) **Feedback efficiency**: ArguteDUB uses the help of deep learning, which to some extent requires specialized hardware support. There might have some challenges when deploying the system on existing base stations directly. How to reduce the amount of calculation and improve the accuracy of prediction are future works.
2) **Design of real environment experiment**: The simulation experiment cannot fully simulate the real environment. In the real environment, the channel changes and noise interference will be more complicated. How to achieve the accuracy of the algorithm in the real environment is the next step that needs attention.
3) **Protocol design**: While focusing on how to achieve cooperative transmission, we should also consider the interaction process of the whole system. We need to design a set of protocols to achieve flexible vehicle participation and exit ArguteDUB, which can achieve flexible system interaction.
4) **Transfer to more scenarios**: In addition to realizing the cooperative transmission between vehicles, we can also take a long-term view and apply the system to different high-speed environments, such as the cooperative communication of mobile devices on high-speed trains using 6G.

References

[1] X. Yi, Y. Liu, L. Kong, G. Chen, X. Liu, M. Shahid, and J. J. P. C. Rodrigues, "6g intelligent distributed uplink beamforming for transport system in highly dynamic environments," in *2022 IEEE Global Communications Conference (GLOBECOM)*.

[2] S. Jayaprakasam, S. K. A. Rahim, and C. Y. Leow, "Distributed and collaborative beamforming in wireless sensor networks: Classifications, trends, and research directions," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2092–2116, 2017.

[3] J. A. Nanzer, R. L. Schmid, T. M. Comberiate, and J. E. Hodkin, "Open-loop coherent distributed arrays," *IEEE Transactions on Microwave Theory and Techniques*, vol. 65, no. 5, pp. 1662–1672, 2017.

[4] R. Mudumbai, J. Hespanha, U. Madhow, and G. Barriac, "Scalable feedback control for distributed beamforming in sensor networks," in *Proceedings. International Symposium on Information Theory, 2005. ISIT*, pp. 137–141.

[5] R. Mudumbai, D. R. B. Iii, U. Madhow, and H. V. Poor, "Distributed transmit beamforming: challenges and recent progress," *IEEE Communications Magazine*, vol. 47, no. 2, pp. 102–110, 2009.

[6] S. Song, J. S. Thompson, P.-J. Chung, and P. M. Grant, "Exploiting negative feedback information for one-bit feedback beamforming algorithm," *IEEE transactions on wireless communications*, vol. 11, no. 2, pp. 516–525, 2011.

[7] N. Xie, K. Xu, and J. Chen, "Exploiting cumulative positive feedback information for one-bit feedback synchronization algorithm," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 5821–5830, 2018.

[8] A. N. D'Andrea, U. Mengali, and R. Reggiannini, "The modified cramer-rao bound and its application to synchronization problems," *IEEE Transactions on Communications*, vol. 42, no. 234, pp. 1391–1399, 1994.

[9] A. B. Gershman, N. D. Sidiropoulos, S. Shahbazpanahi, M. Bengtsson, and B. Ottersten, "Convex optimization-based beamforming," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 62–75, 2010.

[10] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[12] S. J. Taylor and B. Letham, "Forecasting at scale," *The American Statistician*, vol. 72, no. 1, pp. 37–45, 2018.

[13] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[14] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.

[15] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[16] Z. Wang and T. Oates, "Imaging time-series to improve classification and imputation," in *International Joint Conferences on Artificial Intelligence Organization (IJCAI)*, 2015.

[17] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with bm3d?" in *2012 IEEE conference on computer vision and pattern recognition*, pp. 2392–2399.

[18] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *2009 IEEE 12th international conference on computer vision*, pp. 2272–2279.

[19] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE transactions on Image Processing*, vol. 22, no. 4, pp. 1620–1630, 2012.

[20] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2862–2869.

[21] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE transactions on image processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[22] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[23] W. Zhuang, Q. Ye, F. Lyu, N. Cheng, and J. Ren, "Sdn/nfv-empowered future iov with enhanced communication, computing, and caching," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 274–291, 2019.

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

[25] J. Terry and J. Heiskala, *OFDM wireless LANs: A theoretical and practical guide*. Sams publishing, 2002.

[26] G. L. Stuber, J. R. Barry, S. W. Mclaughlin, Y. Li, M. A. Ingram, and T. G. Pratt, "Broadband mimo-ofdm wireless communications," *Proceedings of the IEEE*, vol. 92, no. 2, pp. 271–294, 2004.

[27] 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.

[28] S. Colieri, M. Ergen, A. Puri, and A. Bahai, "A study of channel estimation in ofdm systems," in *Proceedings IEEE 56th vehicular technology conference*, vol. 2. IEEE, 2002, pp. 894–898.

[29] X. Chen, S. Leng, J. He, and L. Zhou, "Deep-learning-based intelligent intervehicle distance control for 6g-enabled cooperative autonomous driving," *IEEE Internet of Things Journal*, vol. 8, no. 20, pp. 15 180–15 190, 2021.

[30] X. Chen, M. Liu, G. Gui, B. Adebisi, H. Gacanin, and H. Sari, "Complex deep neural network based intelligent signal detection methods for ofdm-im systems," in *2021 Joint European Conference on Networks and Communications 6G Summit (EuCNC/6G Summit)*, 2021, pp. 90–94.

[31] S. Song, J. S. Thompson, P.-J. Chung, and P. M. Grant, "Improving the one-bit feedback algorithm for distributed beamforming," in *2010 IEEE Wireless Communication and Networking Conference*, pp. 1–6.

[32] M. F. Gencel, M. E. Rasekh, and U. Madhow, "Distributed transmit beamforming with one bit feedback revisited: How noise limits scaling," in *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 2041–2045.

[33] A. Salem and L. Musavian, "NOMA in cooperative communication systems with energy-harvesting nodes and wireless secure transmission," *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1023–1037, 2021.

[34] C. Jeong and H. Son, "Cooperative transmission of energy-constrained iot devices in wireless-powered communication networks," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3972–3982, 2021.

[35] P. Lin, Q. Song, J. Song, A. Jamalipour, and F. R. Yu, "Cooperative caching and transmission in comp-integrated cellular networks using reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5508–5520, 2020.

[36] H. Liu, N. I. Miridakis, T. A. Tsiftsis, K. J. Kim, and K. S. Kwak, "Coordinated uplink transmission for cooperative NOMA systems," in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6.

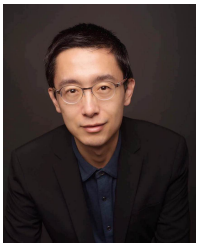
Xingrui Yi received the B.E. degree in computer science from Shanghai Jiao Tong University in 2020 and the master's degree in computer science from Shanghai Jiao Tong University in 2023. He is now a Software Engineer at Alibaba Cloud. His research interests include distributed beamforming, deep learning, and mobile computing.

Jianqiang Li , professor, Member of the Standing Committee of Science and Technology Committee of Shanghai Academy of Aerospace Technology, Chief designer of Long March 2D. He is currently a PhD candidate at School of Astronautics, Beijing University of Aeronautics and Astronautics. Before that, he received a bachelor's degree in Automation from Beijing Institute of Technology and a master's degree in Information and telecommunications from Zhejiang University. His mainly focus on the overall design of carrier rocket, control system design, wireless measurement and so on.

Yutong Liu received the B.E. degree in computer science and technology from the Ocean University of China in 2017 and the Ph.D degree in the same major from Shanghai Jiao Tong University in 2022. She is currently a Research Assistant Professor in computer science with Shanghai Jiao Tong University. Her research interests include wireless communications, artificial intelligence, and mobile computing.

Linghe Kong (Senior Member, IEEE) received the B.Eng. degree in automation from Xidian University in 2005, the master's degree in telecommunication from Telecom SudParis in 2007, and the Ph.D. degree in computer science from Shanghai Jiao Tong University in 2013. He is currently a Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. Before that, he was a Post-Doctoral Researcher with Columbia University, McGill University, and the Singapore University of Technology and Design. His research interests include Internet of things, 5G, blockchain, and mobile computing.

Ying Shao is an Associate Professor in the School of Technology and Engineering at Shanghai Technical Institute of Electronics & Information. She is the member of the academic committee of Shanghai Technical Institute of Electronics & Information. She received Master degree in 2003 and was a Visiting Scholar in Fudan University in 2011. Her research focuses on the embedded system and big data.



Xue Liu (Fellow, IEEE) received the Ph.D. degree (Hons.) in computer science from the University of Illinois at Urbana–Champaign, Champaign, IL, USA. He has also worked as the Samuel R. Thompson Chaired Associate Professor with the University of Nebraska–Lincoln and as a Visiting Faculty with HP Labs, Palo Alto, CA, USA. He is a William Dawson Scholar (Chair Professor) and a Full Professor with the School of Computer Science, McGill University, Montreal, Canada. His research interests include computer and communication networks, real-time and embedded systems, cyber-physical systems and the IoT, green computing, and smart energy technologies. He has published more than 200 research papers in major peer-reviewed international journals and conference proceedings in these areas and received several best paper awards. His research has been covered by various news media.

Guihai Chen (Fellow, IEEE) received the B.S. degree from Nanjing University in 1984, the M.E. degree from Southeast University in 1987, and the Ph.D. degree from The University of Hong Kong in 1997. He is currently a Distinguished Professor with Shanghai Jiao Tong University, China. He was a Visiting Professor with many universities including the Kyushu Institute of Technology, Japan, in 1998, the University of Queensland, Australia, in 2000, and Wayne State University, USA, from 2001 to 2003. He has a wide range of research interests with a focus on sensor network, peer-to-peer computing, and high-performance computer architecture and combinatorics.

Shahid Mumtaz (Senior Member, IEEE) has more than 12 years of wireless industry experience and is currently working as Senior Research Scientist and Technical Manager at GS, Portugal. Dr. Shahid research interests lie in the field of architectural enhancements to 3GPP networks, 5G NR related technologies, green communications, cognitive radio, cooperative networking, radio resource management, Network slicing, LAA/LTU, cross-layer design, Backhaul/fronthaul, heterogeneous networks, M2M and D2D communication, and baseband digital signal processing.

Joel J. P. C. Rodrigues (Fellow, IEEE) is a professor at the Federal University of Piauí (UFPI), Brazil; and senior researcher at the Instituto de Telecomunicações, Portugal. He received the Academic Title of Aggregated Professor in informatics engineering from UBI, the Habilitation in computer science and engineering from the University of Haute Alsace, France, a PhD degree in informatics engineering and an MSc degree from the UBI, and a fiveyear BSc degree in informatics engineering from the University of Coimbra, Portugal. His main research interests include IoT and sensor networks, ehealth technologies vehicular communications, and mobile and ubiquitous computing.