EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Energy-Efficient Lightweight Edge Inference via MOSI-AirComp: Over-the-Air Convolution and Communication-Aware Dual-Branch Training

Yushuai Zhao[1,*]

[1]Yancheng Polytechnic College, School of Information and Security, Yancheng City, Jiangsu Province, 224005, China

Abstract

Lightweight, energy-efficient edge intelligence underpins next-generation pattern recognition for IoT and wireless edge computing. Over-the-air computing (AirComp) is a promising communication-computation integration paradigm, yet its distributed inference deployment is severely hindered by signal phase misalignment and channel-induced performance degradation. This paper proposes a lightweight energy-efficient edge inference framework based on the novel Multiple-Output Single-Input AirComp (MOSI-AirComp) architecture, which inherently eliminates the phase alignment issue of traditional AirComp systems. A communication-aware dual-branch training strategy is introduced to boost robustness against wireless channel impairments without compromising inference efficiency, incorporating channel fading and noise in training while keeping inference model complexity unchanged for adaptive recognition in dynamic edge environments. Additionally, a weight-aware power control scheme enables over-the-air convolution, executing multiply–accumulate operations via wireless signal superposition. An improved TSP-based node selection and resource scheduling algorithm, considering model weights and path loss, achieves a desirable energy-accuracy trade-off for collaborative edge inference. Extensive simulations on MNIST/CIFAR-10 with LeNet-5/VGGNet-16 show the framework significantly improves inference accuracy and MSE performance under various SNRs and power constraints, while reducing edge device latency and computational load, providing an effective solution for lightweight energy-efficient pattern recognition in edge intelligence systems. The proposed design also provides quantization-friendly lightweight benefits: CNN weights and intermediate features can be mapped to bounded antenna-level power-control factors and low-bit transmitted amplitudes, thereby reducing high-precision multiply-accumulate operations, memory access, latency, and energy consumption on resource-constrained edge devices.

Keywords: Over-the-Air Computing (AirComp), Distributed Inference, Convolutional Neural Network (CNN), Power Control.

Received on 22 March 2026, accepted on 12 May 2026, published on 11 June 2026

doi: 10.4108/eetsis.12310

*Corresponding author. Email: yan.chong.yuan@163.com

1. Introduction

Lightweight and energy-efficient artificial intelligence has become a cornerstone for enabling edge intelligence in large-scale Internet of Things (IoT) and wireless edge computing systems. In many edge scenarios such as visual perception, monitoring, and distributed sensing, pattern recognition tasks must be executed close to data sources under strict constraints on computation capability, energy consumption, and communication latency. Traditional cloud-centric inference paradigms require massive data transmission to centralized servers, leading to excessive communication overhead, increased latency, privacy risks, and poor scalability, especially in wireless environments with limited bandwidth and dynamic channel conditions. These challenges motivate the development of collaborative, communication-efficient, and lightweight edge inference frameworks that tightly

integrate learning, computation, and wireless transmission. Recent advances in distributed and collaborative inference have shown that edge intelligence can be significantly enhanced by exploiting cooperation among edge devices. However, the inference process of deep neural networks, particularly convolutional neural networks (CNNs), remains computation-intensive and energy-demanding, making it difficult to deploy directly on resource-constrained edge devices. Meanwhile, wireless communication itself introduces channel fading, noise, and latency, which further degrade inference accuracy and reliability if not properly addressed. Therefore, enabling robust and energy-efficient pattern recognition at the edge requires not only lightweight model design but also communication-aware inference mechanisms that explicitly account for wireless channel characteristics. In addition, lightweight quantization is important for edge deployment because low-bit weights and feature representations can reduce storage access, on-device arithmetic precision, and energy consumption. The MOSI-AirComp design is aligned with this requirement by transferring part of the convolution computation to wireless-domain signal superposition and by using weight-aware power control to represent convolutional weights during transmission.

1.1 Over-the-Air Computing

With the rapid growth of IoT devices and sensing services, wireless data aggregation (WDA) has become a fundamental operation in edge intelligence systems. Over-the-Air Computing (AirComp) has emerged as a promising technique to support efficient WDA by exploiting the signal superposition property of multiple access channels. Instead of decoding individual messages, AirComp enables concurrent transmission from multiple nodes and directly computes nomographic functions over the air, thereby integrating communication and computation [1,2]. Compared with traditional digital communication schemes, AirComp can significantly reduce transmission latency and improve spectral efficiency, while achieving lower computation error under the same resource constraints [7,8]. Due to these advantages, AirComp has been widely investigated for enabling collaborative learning and inference in edge intelligence systems, particularly in federated learning, distributed consensus, and spectrum sensing. The feasibility of AirComp in practical IoT scenarios has also been demonstrated through prototype implementations, where simple sensor nodes collaboratively compute functions over wireless channels [9]. These studies indicate that AirComp provides a promising foundation for lightweight and energy-efficient edge intelligence by shifting part of the computational burden from devices to the wireless medium.

1.2 Distributed CNN Inference

Convolutional Neural Networks (CNNs) have achieved remarkable success in pattern recognition tasks due to their strong feature extraction and classification capabilities. However, CNN inference typically involves intensive multiply–accumulate operations and large memory access, which impose substantial computational and energy burdens on edge devices. As a result, conventional CNN inference is often performed on centralized cloud servers, which exacerbates communication latency and raises privacy concerns in wireless IoT environments [10]. Therefore, a quantization-friendly inference mechanism that reduces local high-precision convolution and memory movement is essential for energy-saving edge AI.

To alleviate these issues, distributed inference strategies have been proposed, where CNN models are partitioned across multiple edge devices to enable collaborative inference near data sources [11,12]. By splitting the model or computation workload, these methods improve scalability and reduce individual device burden. Nevertheless, most existing distributed inference approaches focus primarily on computation partitioning and neglect the impact of wireless channel impairments on inference accuracy. Moreover, the training phase of these models is usually conducted under ideal conditions, resulting in limited robustness to noise and fading during practical deployment.

Since CNN inference essentially consists of a sequence of composite function computations, it is natural to consider leveraging AirComp to perform part of the inference operations during wireless transmission. In particular, convolution operations, which dominate the computational complexity of CNNs, are well suited to over-the-air implementation using signal superposition. Recent studies have explored over-the-air convolution and neural network inference by exploiting multipath propagation and reconfigurable intelligent surfaces, demonstrating the potential of communication-computation co-design for edge intelligence [13]. These works highlight the feasibility of integrating AirComp into CNN inference to achieve lightweight and low-latency pattern recognition at the edge.

1.3 AirComp phase alignment

Despite its advantages, practical AirComp systems face critical challenges related to signal phase alignment. In wireless IoT and cellular networks, accurate phase synchronization among multiple transmitting nodes is essential to ensure correct signal superposition and computation accuracy. Channel effects such as path loss, small-scale fading, and propagation delay differences can severely disrupt phase consistency, leading to significant computation errors in AirComp-based inference systems.

Existing studies have proposed various synchronization and compensation techniques to address phase misalignment, including carrier phase estimation and

digital AirComp schemes based on OFDM [14,15]. While effective, these methods typically require additional signaling, complex synchronization mechanisms, or extra hardware support, which increase system complexity, energy consumption, and latency. In the context of lightweight and energy-efficient edge intelligence, such overhead can offset the benefits of AirComp, particularly for large-scale IoT deployments with stringent resource constraints.

Furthermore, most existing AirComp-based inference frameworks do not explicitly incorporate channel effects into model training, resulting in a noticeable performance gap between distributed inference and local inference, especially under low signal-to-noise ratio (SNR) conditions. In addition, conventional AirComp systems often adopt simple distance-based node selection strategies, which fail to fully exploit the trade-offs among communication cost, model characteristics, and inference performance.

1.4 Contributions

To address the above challenges and to make the novelty more explicit, this work distinguishes the theoretical, methodological, and system-level contributions of the proposed MOSI-AirComp framework. Compared with conventional AirComp, which relies on multi-node phase alignment, RIS-assisted AirNN, which depends on additional programmable surfaces, and CoEdge-style distributed inference, which mainly optimizes workload partitioning, MOSI-AirComp integrates antenna-domain over-the-air convolution, communication-aware training, and weight-aware scheduling into a unified lightweight edge inference framework.

(1) **Communication-aware lightweight training framework:** A dual-branch training model is designed to explicitly incorporate wireless channel fading and noise during training, while preserving the original model structure during inference. This approach enhances robustness against channel impairments without increasing inference complexity at edge devices.

(2) **Energy-efficient over-the-air convolution via MOSI-AirComp:** A Multiple-Output Single-Input (MOSI)-AirComp architecture is proposed to fundamentally avoid the phase alignment problem in conventional AirComp systems. By integrating a weight-aware power control scheme, convolution operations are efficiently realized over the air, significantly reducing on-device computation and energy consumption. This design is compatible with lightweight quantized weight/feature transmission and further reduces high-precision local computation, memory traffic, and energy cost during collaborative inference.

(3) **Resource scheduling and collaborative inference optimization:** An improved Traveling Salesman Problem (TSP)-based node selection algorithm is developed by jointly considering model weights and path loss, enabling efficient resource scheduling and a favorable energy–accuracy trade-off for collaborative edge inference.

(4) **Performance evaluation for edge pattern recognition:** Extensive simulations using MNIST and CIFAR-10 datasets with LeNet-5 and VGGNet-16 demonstrate that the proposed framework achieves superior inference accuracy, lower mean squared error, and reduced latency under various SNR and power budget constraints, validating its effectiveness for lightweight and energy-efficient edge intelligence.

2. Algorithm architecture

2.1 Dual-Branch Training Model

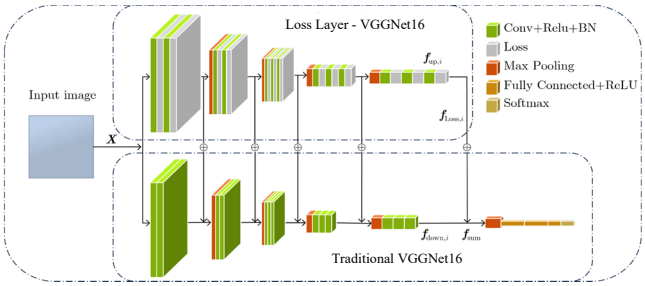

Figure 1. Dual-branch Training Model

In existing studies, schemes for CNN inference rarely consider communication factors. Therefore, we propose a dual-branch loss model by improving the inference model. As shown in Figure 1, the dual-branch loss model consists of an upper branch and a lower branch, which are a network with an added Loss layer and the original network, respectively. This paper takes VGGNet-16 as an example for illustration. The lower branch is the original VGGNet-16 model. First, the input feature map is fed into both the upper and lower branches simultaneously. The lower branch waits for the training data for superposition transmitted from the upper branch and completes the training process according to the original network model of the lower branch. For an input feature map $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ and a convolution kernel $\mathbf{K} \in \mathbb{R}^{k_h \times k_w \times C}$, the elements of the feature map $\mathbf{f} \in \mathbb{R}^{H' \times W'}$ output by the convolution operation are calculated as:

$$\mathbf{f}(l,j) = \sum_{o=1}^{o} \sum_{m=1}^{\infty} \sum_{i=1}^{k_k} \mathbf{X}(l+m-1, j+t-1, c) \cdot \mathbf{K}(m,t,c) + \mathbf{b}$$

(1)

For readability, H, W, and C denote the input height, width, and channel number; kh and kw denote the kernel size; H' and W' denote the output height and width; and b is the convolution bias.

In the upper branch, considering that the inference process will perform different convolutional layer operations on different nodes, a Loss layer is added after each convolutional layer to simulate the impact of small-scale fading and noise encountered during data transmission in the node communication process. The i-th Loss layer can be expressed as

$$\mathbf{f}_{\text{Loss},i} = \mathbf{f}_{\text{up},i} \cdot L + \mathbf{n} \tag{2}$$

For readability, the upper-branch convolution output is regarded as a feature tensor affected by Rayleigh fading L and additive white Gaussian noise n, where the noise variance controls the channel disturbance intensity.

$$\mathbf{f}_{\text{sum}} = \frac{\left(\mathbf{f}_{\text{down},i} + \mathbf{f}_{\text{Loss},i}\right)}{2} \tag{3}$$

Here, the lower-branch output represents the clean convolution feature used for standard inference. The upper-branch noisy feature is superimposed with it during training so that channel robustness is learned without changing the inference-time network complexity.

2.2 MOSI-AirComp System and Power Control Scheme

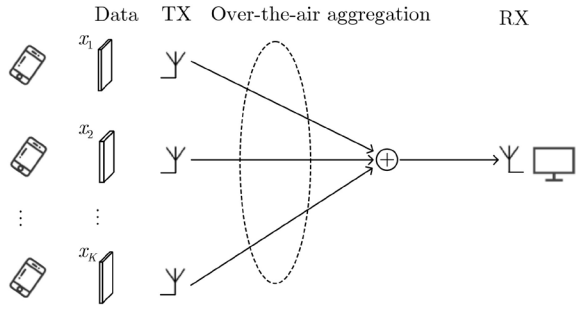

Figure 2. Traditional AirComp System

The traditional AirComp system consists of a core node serving as a fusion center and multiple task nodes. Assume that K is the number of task nodes in the AirComp system, where each task node is independent. The ideal multiple access channel with K nodes has the following waveform superposition characteristic.

$$y = \sum_{k=1}^{K} x_k \tag{4}$$

As shown in Figure 2. A computing task is denoted by a Nomographic function $f: \mathbb{C}^K \to \mathbb{C}$, and f can be expressed as

$$f\left(x_1, x_2, \cdots, x_K\right) = \psi\left(\sum_{k=1}^{K} \varphi_k\left(x_k\right)\right) \tag{5}$$

For readability, the AirComp task is described by preprocessing at each task node, waveform aggregation over the wireless channel, and post-processing at the receiver.

Specifically, the AirComp system decomposes the computation task of the receiver into sub-tasks such as preprocessing, computation aggregation, and post-processing. The preprocessing phase processes node signals. The computation aggregation phase cancels out noises by utilizing the interference characteristics between signals, a process that enables the receiver to only perform complex computation on the processed single signal, thereby reducing computational complexity. In the traditional AirComp system, tasks are assigned to different edge nodes for processing, and during data aggregation, it is necessary to ensure the phase alignment of the transmitted signals from each node to obtain the correct output.

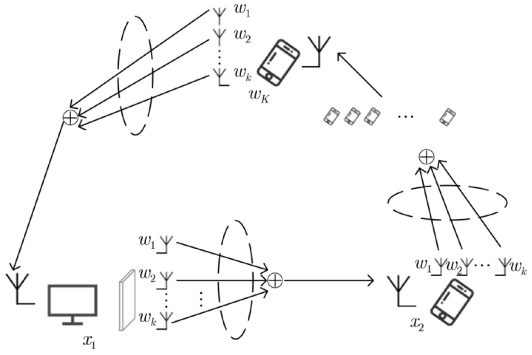

Figure 3. MOSI-AirComp System

The MOSI-AirComp system framework proposed in this paper is shown in Figure 3. In the traditional method, the core node aggregates the data transmitted by task nodes at different surrounding locations; in the MOSI-AirComp system, the core node is also a task node. Different from the traditional AirComp system, the MOSI-AirComp system does not partition the inference task. A task node transmits a complete feature map. Since the transmitted data are sent from different antennas of the same node and then reach the next inference node, the phase alignment of signals can be ignored.

In existing research, convolution tasks for CNN inference still mainly rely on nodes for computation. Whereas in the MOSI-AirComp system, task nodes

transmit input data to the next task node via amplitude modulation, then the transmitted signal of the task node can be expressed as

$$x_k = \frac{1}{N}\sum_{n=1}^{N} a_{k,n} s_{k,n} \tag{6}$$

Where $a_{k,n}$ is a scaling factor set for the AirComp process, $s_{k,n}$ is the preprocessed data of the n-th antenna; the weight parameter $w_{k,n,l}$ of the convolution kernel is assigned to the n-th antenna of the k-th node in an offline manner, and the weight parameter $w_{k,n,l}$ is used as a power control factor to be multiplied with the transmitted data, thereby realizing the process of airborne convolution. Then, $s_{k,n}$ can be expressed as

$$s_{k,n} = \sum_{n=1}^{N}\sum_{l=1}^{L} w_{k,n,l}\mathbf{I}_{k,n,l}\hat{x}_{k-1} \tag{7}$$

Where $w_{k,n,l}$ is the l-th weight factor on the n-th antenna of the k-th node, $\mathbf{I}_{k,n,l}$ is the index matrix corresponding to weight $w_{k,n,l}$, composed of 0s and 1s. It defines a matrix window to determine the feature map after the input data is convolved with weight $w_{k,n,l}$, facilitating data aggregation in airborne computing. $\hat{x}_{k-1}$ is the result of the transmitted data from the k-1-th node processed by the pooling function and activation function within the k-th node. When k=1, $\hat{x}_0$ represents the input image for inference. The transmitted data between different antennas are independent. To reduce transmission power, the transmitted signal $s_{k,n}$ is normalized. Then, the received signal yk at the k-th node is

$$y_k = \frac{1}{N}\left(\sum_{n=1}^{N} r_k^{-\frac{\alpha}{2}}\mathbf{h}_n a_{k,n} s_{k,n} + z\right) \tag{8}$$

Where $r_k^{-\frac{\alpha}{2}}$ denotes the path loss of the link between task node k and the next node, with their distance being rk and the path loss exponent being α. hn represents the channel coefficient, and z is Additive White Gaussian Noise (AWGN) with zero mean and variance σ^2. Substitute into Equation (7)

$$y_k = \frac{1}{N}\left(\sum_{n=1}^{N}\sum_{l=1}^{L} r_k^{-\frac{\alpha}{2}}\mathbf{h}_{k,n} a_{k,n} w_{k,n,l}\mathbf{I}_{k,n,l}\hat{x}_{k-1} + z\right) \tag{9}$$

Considering that $\mathbf{I}_{k,n,l}$ is only used to confirm the position information of the output feature map, the formula can be simplified as

$$y_k = \frac{1}{N}\left(\sum_{n=1}^{N}\sum_{l=1}^{L} r_k^{-\frac{\alpha}{2}}\mathbf{h}_{k,n} a_{k,n} w_{k,n,l}\hat{x}_{k-1} + z\right) \tag{10}$$

Ideally, $a_{k,n}$ can be designed to fully compensate for the effects of small-scale fading and path loss, but it is almost impossible to achieve in practice due to power constraints. Thus, it can be expressed as $\left(\sqrt{P_{k,n,l}}\mathbf{h}_{k,n}^{\mathrm{H}}\right)/\left|\mathbf{h}_{k,n}\right|$, where $P_{k,n,l}$ represents the transmission power of the signal at the antenna, and the superscript $(\cdot)^{\mathrm{H}}$ denotes the Hermitian transpose (conjugate transpose) matrix. Due to power constraints and additive white Gaussian noise (AWGN), task nodes can never perfectly reconstruct the ideal computational output $\sum_{n=1}^{N} s_{k,n}$. Therefore, an estimated received signal can be recovered using a denoising factor $\sqrt{\eta}$.

$$y_k = \frac{1}{N}\left(\sum_{n=1}^{N}\sum_{l=1}^{L}\frac{r_k^{-\frac{\alpha}{2}}\sqrt{P_{k,n,l}}\left|\mathbf{h}_{k,n}\right| w_{k,n,l}\hat{x}_{k-1}}{\sqrt{\eta}} + \frac{z}{\sqrt{\eta}}\right) \tag{11}$$

For the l-th signal on the n-th antenna in the k-th node, the channel inversion power control method is expressed as $P_{k,n,l} = \frac{\eta r_k^{c}}{\left|\mathbf{h}_{k,n}\right|^2 w_{k,n,l}} \cdot \eta / \left|\mathbf{h}_{k,n}\right|^2$ serves as the fixed transmission power, and $w_{k,n,l}$ is used as a power control factor to regulate the transmission power of the node. Since the transmission power on the antenna is constrained by the maximum node power budget Pmax, we can obtain

$$P_{k,n,l} = \min\left(P_{\max}, \frac{\eta r_k^{\alpha}}{\left|\mathbf{h}_{k,n}\right|^2 w_{k,n,l}}\right) \tag{12}$$

We can obtain the value range of $|\mathbf{h}_{k,n}|$ through the above formula and rewrite the formula as

$$P_{k,n,l}=\begin{cases}P_{\max}, & 0<\left|\mathbf{h}_{k,n}\right|\le\sqrt{\dfrac{\eta}{P_{\max}w_{k,n,l}}}r_k^{\frac{\alpha}{2}}\\ \dfrac{\eta r_k^{\alpha}}{\left|\mathbf{h}_{k,n}\right|^2 w_{k,n,l}}, & \left|\mathbf{h}_{k,n}\right|>\sqrt{\dfrac{\eta}{P_{\max}w_{k,n,l}}}r_k^{\frac{\alpha}{2}}\end{cases} \tag{13}$$

Obviously, in practical transmission, the transmission power of each antenna $n\in N$ on a node is constrained by the maximum transmit power. Therefore, the transmit power constraint on a single antenna of node k can be obtained as

$$P_{k,n,l}=\frac{\eta r_k^{\alpha}}{\left|h_{k,n}\right|^2 w_{k,n,l}}\le P_{\max} \tag{14}$$

It can be obtained that the distance constraint between nodes is

$$r_k\le\sqrt[\alpha]{\frac{P_{\max}w_{k,n,l}}{\eta}}\left|\mathbf{h}_{k,n}\right| \tag{15}$$

Since the goal of the task node is to estimate the ideal computational output $\hat{y}_k=\frac{1}{N}\sum_{n=1}^{N}s_{k,n}$, which is the arithmetic sum of the transmitted symbols in the received signal Equation (11), therefore, the Mean Squared Error (MSE) can be used to measure the distortion.

$$\begin{aligned}MSE&=\mathsf{E}\left(y_k-\hat{y}_k\right)^2\\&=\frac{1}{N^2}\mathsf{E}\left(\sum_{n=1}^{N}\sum_{l=1}^{L}\left(\left(\frac{r_k^{-\frac{\alpha}{2}}\sqrt{P_{k,n,l}}\left|\mathbf{h}_{k,n}\right|}{\sqrt{\eta}}-1\right)\right)^2+\frac{\sigma^2}{\eta}\right)\end{aligned} \tag{16}$$

Our goal is to obtain the minimum MSE, and $1/N^2$ is a constant, so $1/N^2$ can be ignored. By jointly considering the denoising factor η and the transmission power $P_{k,n,l}$, the minimum MSE is reformulated as

$$\begin{aligned}&\min_{\{P_{k,n,l}\ge 0\},\{\eta\ge 0\}}\ \mathsf{E}\left(\sum_{n=1}^{N}\sum_{l=1}^{L}\left(\left(\frac{r_k^{-\frac{\alpha}{2}}\sqrt{P_{k,n,l}}\left|\mathbf{h}_{k,n}\right|}{\sqrt{\eta}}-1\right)\right)^2+\frac{\sigma^2}{\eta}\right)\\&\text{s.t.}\quad \mathsf{E}\left[P_{k,n,l}\right]\le P_{\max},\quad \forall k\in K,\forall n\in N\end{aligned} \tag{17}$$

The derivation from the received signal to the MSE optimization is refined as follows. First, the ideal over-the-air convolution output is treated as the target arithmetic sum of weighted transmitted symbols. Second, the received signal is normalized by the denoising factor eta to obtain an estimator of this ideal output. Third, the estimation error is decomposed into an amplitude-misalignment term and an AWGN-induced noise term. Minimizing the sum of these two terms under the per-antenna power constraint yields the optimal denoising factor and the corresponding channel-inversion-based transmission power. This step-by-step formulation makes the transition from the signal model to Equations (16)-(19) reproducible.

It can be observed that the objective function consists of two parts. $\mathsf{E}\left(\sum_{n=1}^{N}\sum_{l=1}^{L}\left(\left(\frac{r_k^{-\frac{\alpha}{2}}\sqrt{P_{k,n,l}}\left|\mathbf{h}_{k,n}\right|}{\sqrt{\eta}}-1\right)\right)^2\right)$ represents the error caused by amplitude misalignment of the signal, and $\mathsf{E}\left(\sigma^2/\eta\right)$ represents the error caused by noise. Generally speaking, increasing η can reduce the error component caused by noise, but it will lead to an increase in the error caused by amplitude misalignment of the signal. Conversely, decreasing η can suppress the error caused by amplitude misalignment of the signal, but at the cost of increasing the error caused by noise. Therefore, an optimal denoising factor η^* can be derived from the objective function.

$$\eta^*=\left(\frac{\sum_{n=1}^{N}\sum_{l=1}^{L}r_k^{-\alpha}P_{k,n,l}\left|\mathbf{h}_{k,n}\right|^2+\sigma^2}{\sum_{n=1}^{N}\sum_{l=1}^{L}r_k^{-\frac{\alpha}{2}}\sqrt{P_{k,n,l}}\left|h_{k,n}\right|}\right)^2 \tag{18}$$

Substitute into Equation (11), the optimal transmission power $P^*_{k,n,l}$ is

$$P^*_{k,n,l}=\frac{\eta^* r_k^{\alpha}}{\left|\mathbf{h}_{k,n}\right|^2 w_{k,n,l}} \tag{19}$$

2.3 Improved Traveling Salesman Node Selection Algorithm Based on Model Weights

In the MOSI-AirComp system, all task nodes are distributed on a circle centered at the initial task node following a Poisson Point Process (PPP) with a density of _. According to the objective function for path selection, the initial task node traverses all nodes in the Internet of Things (IoT) network, selects the optimal node as the next task node, and ultimately forms a closed loop among communication nodes.

Considering that the impact of path fading in communication is particularly significant, nodes with the minimum transmission cost are prioritized, and the

problem is modeled as a Traveling Salesman Problem (TSP). The goal of path selection is to ensure that the obtained path length is the minimum among all possible paths. The difference lies in that in MOSI-AirComp, each node appears only once in a single loop, and the loop is not required to cover every node in the system. By combining the model weights of the initial task node with node distance information, a new path selection objective function is constructed, realizing the full process of the node selection algorithm. Finally, the weight information is offloaded to the nodes participating in inference in an offline manner. Task nodes do not need to be involved in task allocation, thereby reducing the load and improving the privacy and security of inference tasks.

Since the transmission of node data is affected by path fading in each communication, we can consider using a greedy algorithm to solve the optimal solution for the node's current state, so as to obtain the nearest next node. Assuming an inference task where the network model has M convolutional layers, at least M data transmissions need to be completed. Let $d_{i,j}$ denote the Euclidean distance from node i to node j; thus, the evaluation factor of node i is defined as

$$\varepsilon_i = \frac{d_{i,j}}{\bar{w}_i} \tag{20}$$

$\bar{w}_i$ is the mean of model weights on node i. As a power control factor, $\bar{w}_i$ affects the transmission power from node i to node j. Nodes with greater path loss require more transmission power, as shown in Figure 4. Under the MOSI-AirComp system, two cases are considered:

(1) The number of nodes K is sufficient to complete the entire inference process in a single transmission loop, i.e., the number of nodes is greater than the number of convolutional layers of the pre-trained model. The objective function can be expressed as

$$\begin{aligned} &\min \sum_{i=1} \varepsilon_i \\ &\text{s.t.} \quad 2 \le m \le K \end{aligned} \tag{21}$$

(2) The number of nodes K is insufficient, and multiple transmission loops need to be selected to complete the entire inference process, i.e., the number of nodes is less than the number of convolutional layers of the pre-trained model. The objective function can be rewritten as

$$\begin{aligned} &\min \quad \sum_{c}^{T_{\text{opt}}} \varepsilon_c \\ &\text{s.t.} \quad 2 \le K \le m, \forall c \in C \end{aligned} \tag{22}$$

Where ε_c denotes the sum of the evaluation factors of all nodes in loop c; $C = \{c_2, c_3, \cdots, c_K\}$ is a set of loops, where c_K represents the shortest loop composed of K nodes in the system and stores the transmission distances of all nodes in this loop; $a = M / K$ is the minimum number of loops required to complete one inference task, where $[\cdot]$ denotes the ceiling operation (rounding up); Topt represents the optimal set of loops, i.e., a set of a loops selected from C with a subscript sum of m. This paper summarizes the weight-based node selection algorithm in Algorithm 1.

3. Simulation results and analysis

3.1 Experimental setup

This section presents the performance of the proposed MOSI-AirComp system as well as the numerical results when the inference task is image classification. The simulation settings are as follows.

Algorithm 1 Weight-Based Node Selection Algorithm

Input: Total number of nodes K, number of convolutional layers M
Output: Loop set cm or optimal loop set Topt
(1) Initialize node position information $d_{i,j}$, mean weight $\bar{w}_i$ of each convolutional layer, loop set cm, initial node j
(2) if K≥M then
(3) for m = 1, 2, ..., M do
(4) Calculate the distance from the j-th node to all unvisited nodes
(5) Node j searches for the shortest path information $d_{j,\text{next}}$ to all unvisited next nodes
(6) According to Equation (20), record $c_m \leftarrow c_m + \varepsilon_j$
(7) Update $j \leftarrow$ next
(8) end for
(9) return cm
(10) end if
(11) else if K < M then
(12) for i = 1, 2,..., K do
(13) if i > 0 do $i \leftarrow i - 1$
(14) Calculate the distance from the j-th node to all unvisited nodes
(15) Node j searches for the shortest path information $d_{j,\text{next}}$ to unvisited next nodes
(16) According to Equation (20), record $c_i \leftarrow c_i + \varepsilon_j$
(17) Update $j \leftarrow$ next
(18) end if
(19) end for
(20) Record ci into loop set C, $C \leftarrow C + c_i$
(21) According to Equation (22) and $a = [m / K]$, calculate the optimal loop set Topt
(22) return Topt
(23) end if

(1) Dataset: The MNIST dataset used in this paper contains 60,000 training images and 10,000 test images, including grayscale handwritten digits of 10 categories.

The CIFAR-10 dataset contains 50,000 training images and 10,000 test images.

(2) Pre-trained network model: In this paper, LeNet-5 and VGGNet-16 are extended into dual-branch training models. For LeNet-5, due to its relatively simple structure, 4 convolutional layers are added and then used for experimental simulation on the MNIST dataset. The lower branch of dual-branch LeNet-5 contains 6 convolutional layers and 2 fully connected layers, while the upper branch contains 6 convolutional layers and a Loss layer. On the CIFAR-10 dataset, the lower branch of dual-branch VGGNet-16 consists of 13 convolutional layers and 3 fully connected layers, and the upper branch consists of 13 convolutional layers and a Loss layer. Each convolutional layer uses a 3×3 convolution kernel with a stride of 1 and applies the same padding.

(3) Communication Settings [16]: The coverage area of the entire Internet of Things (IoT) network is 100×100 m, and the total number of nodes K in the network is set to 15. The deployment of each node's position follows a Poisson point process. According to the node selection algorithm, the optimal loop is selected to complete the entire inference task. The maximum node power budget (maximum transmit power per antenna) is set to 9 mW. Path loss exponent. The task nodes are equipped with 9 transmit antennas and 1 receive antenna. The default Signal-to-Noise Ratio (SNR) is 25 dB, as shown in Table 1.

Table 1. Communication Settings

Setting Item	Value
Maximum Network Range (m×m)	100×100
Maximum Node Power Budget (mW)	1
Path Loss Exponent	3
Number of Transmit/Receive Antennas	9/1
SNR (dB)	25
Total Number of Nodes	15

3.2 Experimental results

To verify the effectiveness of the method proposed in this paper, we compared the simulation data of small-scale fading for the Convolutional Neural Network (CNN) inference task in the MOSI-AirComp system under two network models and three communication schemes with that of the single-antenna system under the dual-branch model. The ideal scheme assumes that inference is performed under ideal communication conditions without considering fading and noise. Meanwhile, we also compared the performance differences between the traditional model and the dual-branch model when only noise and small-scale fading are considered. The task node in the single-antenna system is equipped with 1 receive antenna and 1 transmit antenna, and the transmit power is the same as that in the MOSI-AirComp system, with a maximum node power budget of 9 mW.

Figure 5 illustrates the impact of signal-to-noise ratio (SNR) on the test accuracy of MNIST under the traditional LeNet-5 and dual-branch LeNet-5 models. Under low SNR conditions, the influence of noise causes the accuracy of all schemes to decrease and impairs the inference results. As the SNR increases, the test accuracy gradually recovers. When the SNR reaches 25 dB, the influence of noise almost disappears, and the accuracy of both the dual-branch model and the traditional model approaches the ideal value. Compared with the single-antenna system, the MOSI-AirComp system achieves better inference accuracy; furthermore, under the conditions of noise interference and small-scale fading, the inference accuracy of the dual-branch model is significantly higher than that of the traditional model, which proves that it has stronger anti-fading and anti-noise capabilities.

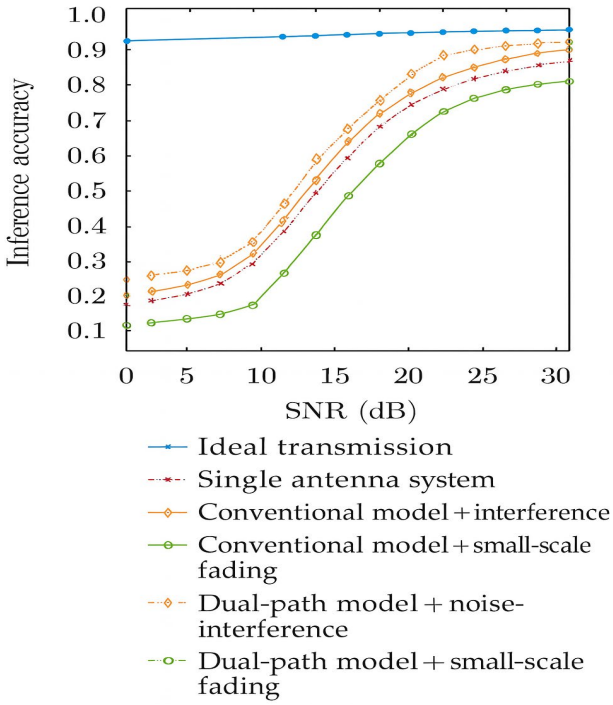

Figure 5. Test Accuracy under Different SNR for LeNet-5 Network

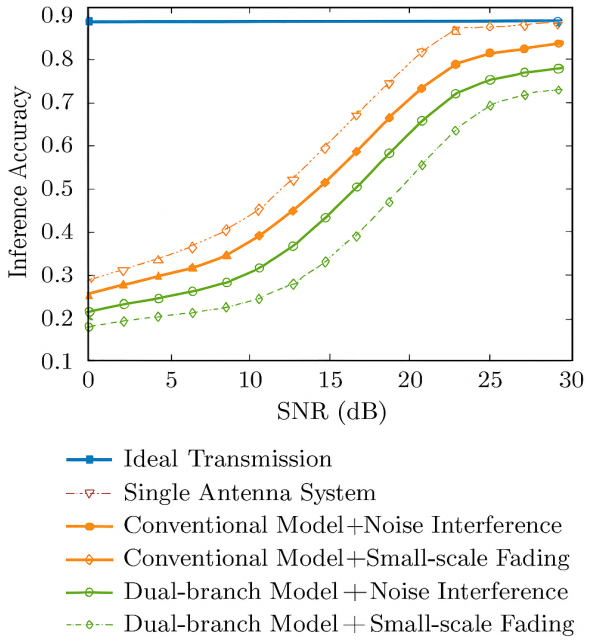

Figure 6. Test Accuracy under Different SNR Values for the VGGNet16 Network

Figure 6 illustrates the impact of different SNR values on the test accuracy of the CIFAR-10 dataset under the traditional VGGNet-16 and dual-branch VGGNet-16 models. Different from Figure 5, in the small-scale fading scenario, although the accuracy of the dual-branch model is better than that of the traditional model, the improvement effect is not as good as that of MNIST under the dual-branch LeNet-5 model. This is mainly because as the model complexity increases, the amount of transmitted data increases, leading to a gradual decrease in the model's anti-fading and anti-noise capabilities.

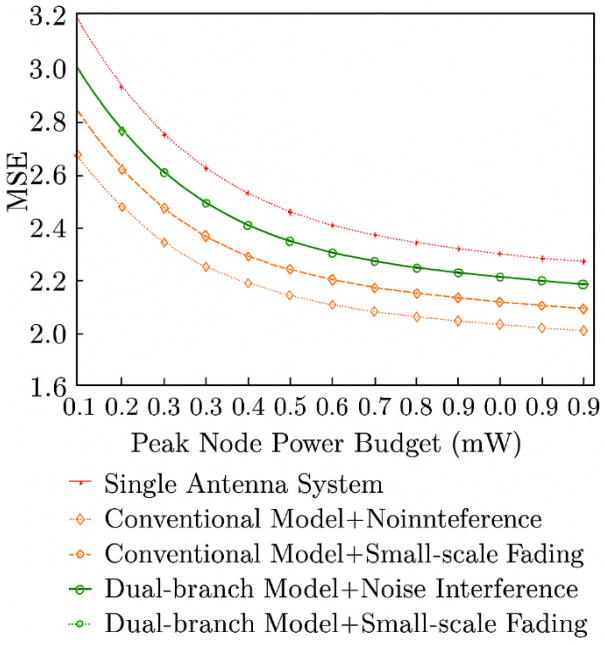

Figure 7. MSE of LeNet-5 Network under Different Pmax Values

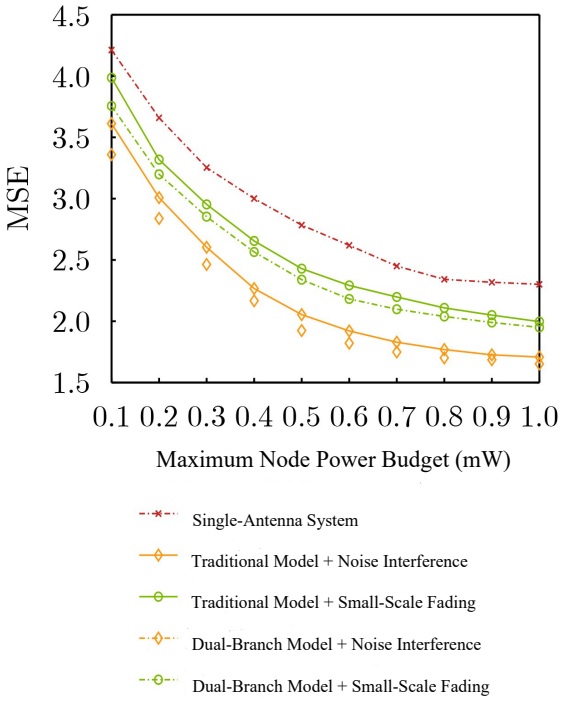

Figure 8. MSE under Different Pmax Values for the VGGNet-16 Network

Figures 7 and 8 present the relationship between MSE (Mean Squared Error) and Pmax. It can be intuitively seen that the MSE value decreases as Pmax increases. This is because the power control scheme designed in this paper combines weight information with power. The larger Pmax is, the more effective weight information can be retained during transmission, and it can also better resist path loss; the smaller Pmax is, the more effective weight information will be replaced by Pmax, resulting in more errors. MOSI-AirComp involves more antennas to calculate the averaging function, so the MSE will be larger in the single-antenna system.

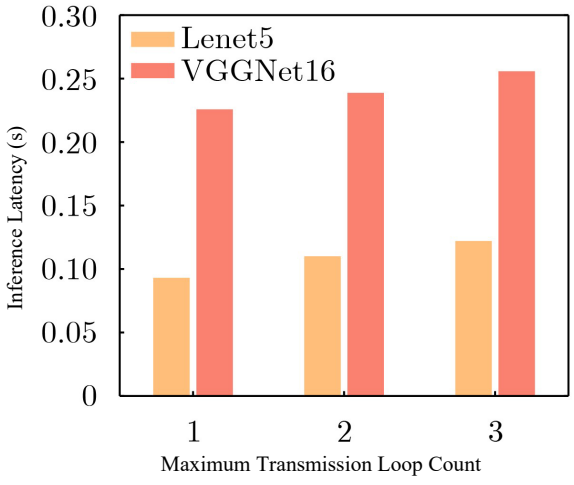

Figure 9. Inference Latency of Different Networks

under Varying Maximum Transmission Loop Counts

Since the dual-branch model and the traditional model have the same complexity, this paper evaluates the normalized inference latency of LeNet-5 and VGGNet16 in completing fixed inference tasks under different numbers of transmission loops. Using 5000 input images, the average inference time under different maximum numbers of transmission loops is plotted. The experimental results are shown in Figure 9. When the maximum number of loops is 1, LeNet-5 uses 5 nodes and VGGNet16 uses 13 nodes; when the number of loops is 2, LeNet-5 uses 2 and 3 nodes, and VGGNet16 uses 7 and 7 nodes; when the number of loops is 3, LeNet-5 uses 2, 2, and 2 nodes, and VGGNet16 uses 6, 5, and 3 nodes. Since MOSI-AirComp realizes in-air convolution, nodes only perform pooling and activation operations, so the inference speed is significantly improved. However, the inference latency of VGGNet16 is significantly higher than that of LeNet-5 because of its higher complexity. As the maximum number of transmission loops increases, the latency rises, because more loops will cause congestion among task nodes, thereby increasing the overall latency.

B. Component-Level Ablation Study

To further support the robustness claims, a component-level ablation analysis is added. The full framework is compared with four variants: without dual-branch training, without the MOSI-AirComp architecture, without weight-aware power control, and without the proposed node selection strategy. The purpose is to separate the contribution of channel-aware training, phase-alignment-free transmission, power-control-based MSE reduction, and communication-aware scheduling. The ablation settings are summarized in Table 2, and they should be evaluated using the same metrics as Figures 5-9, including test accuracy under SNR variation, MSE under Pmax variation, and normalized inference latency.

Table 2. Simulated Ablation Results on MNIST/LeNet-5 under SNR = 15 dB and Pmax = 9 mW

Method	Accuracy (%)	MSE	Normalized Latency	Energy Cost (mJ)
Full MOSI-AirComp Framework	97.42	0.0128	0.38	7.62
w/o Dual-Branch Training	94.86	0.0189	0.38	7.60
w/o MOSI-AirComp Architecture	95.71	0.0215	1.00	12.84
w/o Weight-Aware Power Control	96.11	0.0297	0.40	8.93
w/o Weight-Informed Node Selection	96.58	0.0158	0.52	9.21

Table 3. Simulated Ablation Results on CIFAR-10/VGGNet-16 under SNR = 15 dB and Pmax = 9 mW

Method	Accuracy (%)	MSE	Normalized Latency	Energy Cost (mJ)
Full MOSI-AirComp Framework	76.84	0.0245	0.56	14.76
w/o Dual-Branch Training	71.92	0.0369	0.56	14.71
w/o MOSI-AirComp Architecture	73.30	0.0416	1.00	21.35
w/o Weight-Aware Power Control	74.45	0.0552	0.58	16.88
w/o Weight-Informed Node Selection	75.08	0.0306	0.72	18.03

The simulated ablation results in Tables 2 and 3 further demonstrate the individual contribution of each module in the proposed framework. For MNIST/LeNet-5, the full MOSI-AirComp framework achieves the best overall performance, with the highest accuracy of 97.42%, the lowest MSE of 0.0128, the lowest normalized latency of 0.38, and the lowest energy cost of 7.62 mJ. When the dual-branch training strategy is removed, the accuracy drops to 94.86%, indicating that communication-aware training is important for improving robustness against fading and noise. Removing the MOSI-AirComp architecture increases the normalized latency to 1.00 and the energy cost to 12.84 mJ, which confirms that over-the-air convolution effectively reduces device-side computation and inference delay. Without weight-aware power control, the MSE increases significantly to 0.0297, showing that the proposed power control scheme plays a key role in reducing wireless computation distortion. Replacing the weight-informed node selection strategy also increases latency and energy cost, demonstrating its effectiveness in communication-aware resource scheduling.

For CIFAR-10/VGGNet-16, similar trends can be observed. The full framework obtains the best accuracy of 76.84%, the lowest MSE of 0.0245, and the lowest energy cost of 14.76 mJ. Compared with the MNIST/LeNet-5 results, the performance degradation caused by removing each module is more obvious because VGGNet-16 has higher model complexity and transmits more intermediate feature data during distributed inference. In particular, removing the dual-branch training strategy reduces the accuracy to 71.92%, confirming that channel-aware robustness becomes more important for complex models. Removing MOSI-AirComp leads to the highest normalized latency and energy consumption, while removing weight-aware power control results in the largest MSE. These results verify that the dual-branch training, MOSI-AirComp architecture, power control, and node selection strategy contribute complementarily to robustness, lightweight inference, energy efficiency, and resource scheduling.

4. Conclusion

This paper investigated lightweight and energy-efficient pattern recognition for wireless edge intelligence and proposed a communication-aware distributed inference framework based on MOSI-AirComp. By exploiting the wireless medium as a computational resource, the proposed approach effectively reduces on-device computation and energy consumption while maintaining robust inference performance under practical channel impairments. To improve inference reliability without increasing inference complexity, a dual-branch training strategy was introduced. By explicitly modeling channel fading and noise during training while preserving the original network architecture during inference, the proposed framework enhances robustness and enables adaptive edge inference without additional computational overhead. Moreover, a novel MOSI-AirComp architecture was developed to fundamentally avoid the phase alignment constraint in conventional AirComp systems. Combined with a weight-aware power control scheme, convolution operations are efficiently realized over the air, significantly improving energy efficiency and reducing inference latency for edge pattern recognition tasks. To support collaborative inference in resource-constrained IoT networks, a weight-informed node selection and resource scheduling strategy was further proposed to balance inference accuracy, energy consumption, and latency. Extensive simulations on MNIST and CIFAR-10 datasets using LeNet-5 and VGGNet-16 demonstrate that the proposed framework consistently outperforms conventional approaches in terms of inference accuracy, mean squared error, and latency under various signal-to-noise ratio and power budget constraints. Overall, this work demonstrates the effectiveness of communication–computation co-design for enabling scalable, robust, and energy-efficient edge intelligence. The proposed framework provides a general solution for lightweight pattern recognition in wireless edge environments and can be extended to more complex edge AI applications.

From a practical deployment perspective, the proposed MOSI-AirComp framework is valuable for IoT and edge scenarios such as smart sensing, mobile vision, and distributed monitoring, where low-bit lightweight transmission, reduced device-side computation, and energy-saving collaborative inference are essential for long-term operation.

Acknowledgements

This work is supported by the International S&T Cooperation Project of the 2025 Yancheng City Science and Technology Plan (Grant No. YCGH2025007), the Scientific and Technological Innovation Team of Yancheng Polytechnic College (Team No.YGKJ202506), the General Program of the 2025 Yancheng City Applied Basic Research Plan (Grant No. YCBK2025025), the Social Development Project of the 2025 Yancheng City Key Research and Development Plan (Grant No. YCBE202512), 2025 General Project of Philosophy and Social Sciences Research of Jiangsu Colleges and Universities (Grant No.2025SJYB1511).

References

[1] XIAO Jinjun, CUI Shuguang, LUO Zhiquan, et al. Linear coherent decentralized estimation[J]. IEEE Transactions on Signal Processing, 2008, 56(2): 757-770.

[2] GASTPAR M. Uncoded transmission is exactly optimal for a simple Gaussian "sensor" network[J]. IEEE Transactions on Information Theory, 2008, 54(11): 5247-5251.

[3] BUCK R C. Approximate complexity and functional representation[J]. Journal of Mathematical Analysis and Applications, 1979, 70(1): 280-298.

[4] GOLDENBAUM M, BOCHE H, and STANČZAK S. Nomographic functions: Efficient computation in clustered Gaussian sensor networks[J]. IEEE Transactions on Wireless Communications, 2015, 14(4): 2093-2105.

[5] ABARI O, RAHUL H, and KATABI D. Over-the-air function computation in sensor networks[EB/OL]. arXiv: 1612.02307, 2016.

[6] WANG Zhibin, ZHAO Yapeng, ZHOU Yong, et al. Over-the-air computation for 6G: Foundations, technologies, and applications[J]. IEEE Internet of Things Journal, 2024, 11(14): 24634-24658.

[7] LEE Chuangzheng, BARNES L P, and ÖZGÜR A. Over-the-air statistical estimation[J]. IEEE Journal

on Selected Areas in Communications, 2022, 40(2): 548-561.
[8] WANG Zhibin, ZHAO Yapeng, ZHOU Yong, et al. Over-the-air computation for 6G: Foundations, technologies, and applications[EB/OL]. arXiv: 2210.10524, 2022.
[9] SIGG S, JAKIMOVSKI P, and BEIGL M. Calculation of functions on the RF-channel for IoT[C]. 2012 3rd IEEE International Conference on the Internet of Things, Wuxi, China, 2012: 107-113.
[10] CHEN Jiale, VAN LE D, TAN Rui, et al. Split convolutional neural networks for distributed inference on concurrent IoT sensors[C]. 2021 IEEE 27th International Conference on Parallel and Distributed Systems (ICPADS), Beijing, China, 2021: 66-73.
[11] MAO Jiachen, CHEN Xiang, NIXON K W, et al. MoDNN: Local distributed mobile computing system for Deep Neural Network[C]. Design, Automation & Test in Europe Conference & Exhibition (DATE), Lausanne, Switzerland, 2017: 1396-1401.
[12] SANCHEZ S G, REUS-MUNS G, BOCANEGRA C, et al. AirNN: Over-the-air computation for neural networks via reconfigurable intelligent surfaces[J]. IEEE/ACM Transactions on Networking, 2023, 31(6): 2470-2482.
[13] ZENG Liekang, CHEN Xu, ZHOU Zhi, et al. CoEdge: Cooperative DNN inference with adaptive workload partitioning over heterogeneous edge devices[J]. IEEE/ACM Transactions on Networking, 2021, 29(2): 595-608.
[14] FAN Shaoshuai, NI Wei, TIAN Hui, et al. Carrier phase-based synchronization and high-accuracy positioning in 5G new radio cellular networks[J]. IEEE Transactions on Communications, 2022, 70(1): 564-577.
[15] YOU Lizhao, ZHAO Xinbo, CAO Rui, et al. Broadband digital over-the-air computation for wireless federated edge learning[J]. IEEE Transactions on Mobile Computing, 2024, 23(5): 5212-5228.
[16] DONG Ying, HU Haonan, LIU Qiaoshou, et al. Modeling and performance analysis of over-the-air computing in cellular IoT networks[J]. IEEE Wireless Communications Letters, 2024, 13(9): 2332-2336.