Communication-aware in-memory wireless neural networks

Zai-Zheng Yang[1], Cong Wang[1*], Yichen Zhao[1], Gong-Jie Ruan[1], Xing-Jian Yangdong[1], Yuekun Yang [2], Chen Pan[3], Bin Cheng[3], Shi-Jun Liang[1*] and Feng Miao[1*]

[1]Institute of Brain-inspired Intelligence, National Laboratory of Solid State Microstructures, School of Physics, Collaborative Innovation Center of Advanced Microstructures, Nanjing University, Nanjing, China.

[2]School of Intelligence Science and Technology, Nanjing University, Suzhou, China

[3]Institute of Interdisciplinary Physical Sciences, School of Science, Nanjing University of Science and Technology, Nanjing, China.

*Correspondence Email: cong@nju.edu.cn; sjliang@nju.edu.cn; miao@nju.edu.cn

Abstract:

Edge-cloud wireless collaborative computing is critical for energy-constrained edge devices to perform complex intelligent tasks surpassing their computational capacities. However, traditional wireless collaborative systems encounter significant challenges in energy efficiency and latency due to several “separation” issues (i.e., the separation of memory and computing, the separation of signal processing and transmission/reception, and the separation of neural networks and wireless communication). In this work, we propose communication-aware in-memory wireless neural networks, a novel wireless collaborative computing paradigm that employs analog in-memory computing (AIMC) technology to implement wireless collaborative systems. In such a system, the AIMC-based wireless communication is integrated as a trainable component and is optimized together with neural networks through our proposed communication-aware training approach. To validate the scheme, we built an AIMC-based prototype system comprising an edge inference accelerator and a wireless communication system. The prototype achieves an experimental inference accuracy of 93.71% on Street View House Number dataset. Moreover, we show that our design maintains nondegraded inference accuracy even using low-resolution analog-to-digital converters (*e.g.*, 1-bit) in wireless communication, which is unachievable in conventional wireless collaborative systems. Finally, we demonstrate that the proposed approach not only makes the wireless collaborative system adaptable to various wireless conditions but also further reduces communication costs, such as the programming precision of AIMC hardware and transmit power. Our work highlights the potential of using in-memory computing hardware in edge-cloud wireless collaborative systems.

Main Text:

The flourishing development of deep neural networks has enabled a variety of intelligent edge applications, such as object detection, text generation, and smart healthcare. Edge devices in these scenarios generate vast amounts of data that require processing by computation-intensive neural networks. Given the substantial

computational demands of these neural networks, efficient edge-cloud wireless collaboration[1-4] has become essential for extending the computational capacity of resource-constrained edge devices. However, conventional edge-cloud wireless collaborative systems face critical limitations in energy efficiency and latency. The employed von Neumann processors suffer substantial energy dissipation due to frequent data transfers between separated memory and processing units, causing performance bottlenecks in edge neural networks. Additionally, the separation of signal processing and transmission/reception in wireless systems hampers the information flow between spatially distributed neural networks, increasing system latency. Last but not least, wireless communication often operates independently from the neural networks and endeavors to transport intermediate data without error, which neglects the energy efficiency potential enabled by the error tolerance capacity of neural networks. These limitations underscore the pressing demand for alternative wireless collaboration paradigms that can fully exploit the constrained energy budgets of edge devices while maintaining high system performance.

Analog in-memory computing (AIMC) emerges as a promising technology to address these “separation” issues. Instead of separating memory and processing units, AIMC enables in-situ processing directly within the analog domain, bypassing the memory wall of von Neumann architecture. By leveraging physical laws (*i.e.*, Ohm’s law and Kirchhoff’s law) to perform parallel vector-matrix multiplications (VMM, the core operation in neural network computations) directly within memory, AIMC significantly enhances the energy efficiency of edge computing[5-21]. Moreover, AIMC's seamless integration with analog processes such as sensing[22,23] and wireless transmission[24-36] enables it to perform signal processing within the analog domain, accelerating the information flow. With these advantages, AIMC possesses enormous potential for achieving edge-cloud wireless collaborative intelligent systems with ultrahigh energy efficiency and ultralow latency.

In this work, we present a communication-aware in-memory wireless neural network. This scheme leverages the AIMC technology to implement both edge computing and wireless communication, and integrates wireless communication as a learnable module of the wireless neural network. The AIMC-based neural networks and the AIMC-based wireless communication are trained together through the proposed communication-aware training (CAT) approach. A prototype of the communication-aware in-memory wireless neural network is built to validate our design. When performing the wireless collaborative inference of a convolutional neural network (CNN), our prototype achieves an experimental inference accuracy of 93.71% on the SVHN (Street View House Numbers) dataset, with only a 0.3% fluctuation compared to ideal accuracy. Furthermore, we show that our design maintains undegraded inference accuracy even when employing 1-bit analog-to-digital converters (ADCs) for wireless communication. Simulation results based on the ImageNet dataset demonstrate that our training approach significantly reduces the required transmit power as well as precision demands of AIMC chips, and enhances the applicability of wireless neural

networks to complex and varying wireless channels, as well as multiple modulation schemes.

Communication-aware in-memory wireless neural networks

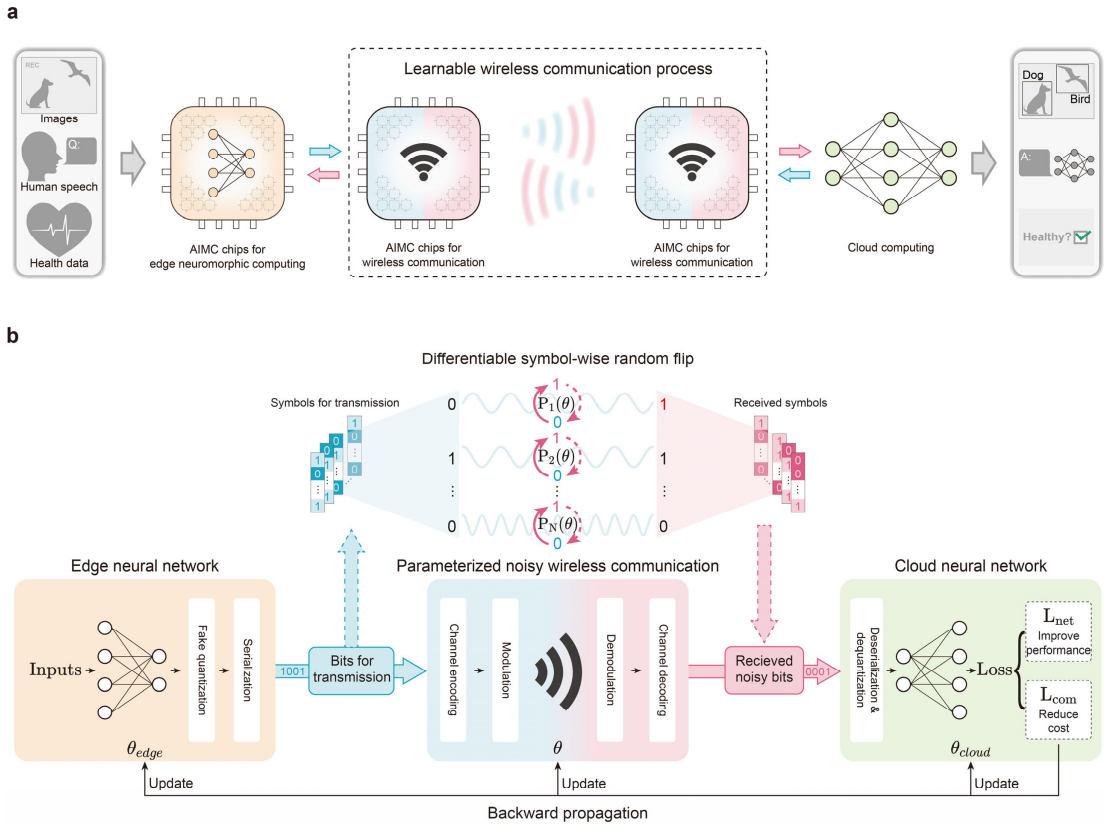

Fig.1 | An overview of communication-aware in-memory wireless neural networks. a, The schematic diagram of communication-aware in-memory wireless neural networks. AIMC technology is employed in both neuromorphic computing and wireless communication. Wireless communication is considered as a learnable module within the neural network, and is trained together with the neural network through the communication-aware training approach. **b,** The flowchart of communication-aware training. The wireless communication is modeled as a differentiable symbol-wise bit flip process during training. The flipping probabilities are determined by the learnable wireless communication parameters θ. Since the training objective comprises the original tasks-specific objective and the communication-relavant objective, the neural network is trained towards improving the task performance while reducing the communication cost.

Figure 1a shows the proposed communication-aware in-memory wireless neural network, which comprises spatially distributed neural networks. In edge-cloud collaboration scenarios, the edge neural network is deployed on edge devices, while the cloud neural network runs on powerful computing nodes. The in-memory edge neural network offloads most calculations to AIMC chips, which are able to perform the VMM operation in one-shot manner. During the intermediate data transmission between the edge and the cloud, the AIMC chips are used to perform wireless signal processing (modulation/demodulation) within the analog domain to efficiently transform data into modulated analog radio signals and extract data from received analog signals. In the traditional scenarios of wireless communication, the error-free transmission rate over a

noisy channel is theoretically constrained by the Shannon limit[37], approaching which generally requires substantial resources (e.g. high transmit power for a sufficient SNR) and complex coding strategies[38-40]. Fortunately, the intermediate feature data of neural networks possess intrinsic redundancy (i.e., a few errors of intermediate data would not impact the systemic performance), enabling tolerance to transmission errors to a certain degree. Hence, in the scenario of the edge-cloud wireless collaborative neural networks, the Shannon limit can be temporarily bypassed through error-tolerant transmission in exchange for resource savings in wireless communication. To this end, we propose the communication-aware training approach that integrates the wireless communication as a learnable process within the neural network and jointly optimizes their parameters, automatically digging the potential in efficiency improvement enabled by the neural network's error tolerance capacity.

Figure 1b depicts the schematic diagram of communication-aware training. The wireless neural network consists of distributed neural networks interconnecting through parameterized lossy wireless communication. In the forward phase, intermediate data generated from the edge neural network undergo fake quantization, a common technique in quantization-aware training[41-44] (QAT). The quantized data are then serialized into bitstream and divided into symbols for transmission. We model the noisy wireless communication by simulating random bit flips in each symbol, with probabilities equivalent to the corresponding bit error rates (BERs) determined by learnable wireless communication parameters. Reparameterization techniques make this random flipping process differentiable (Details are presented in Methods and Supplementary Text Section 2.1). By properly adjusting the flipping probabilities, we can emulate the effects of various wireless conditions (*e.g.*, varying SNR, diverse channel fading characteristics, and multiple modulation schemes) on transmission. This customized method, termed channel augmentation, is introduced in Supplementary Text Section 5.3. The received perturbed bits are deserialized, then dequantized, and finally serve as the inputs to the cloud neural network. In the backward phase, the learnable wireless parameters (denoted by θ) are updated together with the neural network parameters (denoted by $\theta_{edge}, \theta_{cloud}$). The training objective is defined as follows:

$$\underset{\theta_{edge},\theta_{cloud},\theta}{minimize}\ L_{net}(\theta_{edge},\theta_{cloud},\theta) + \beta L_{com}(\theta) \quad (1)$$

It contains two components: the original task-specific objective function, represented by L_{net}, and the communication-relavant one, represented by L_{com}. The contribution of the second objective is adjusted by a scaling factor, denoted by β. With the communication-relevant objective defined as minimizing the overheads of wireless communication, parameters are optimized toward improving the task performance (e.g. classification accuracy) while reducing the concerned costs. Moreover, since the potential transmission errors are considered during training, the co-optimized wireless neural network's robustness to these errors will be specifically more strengthened than that of independently optimized one, similar to the effects of dropout regularization[45] and adversarial training[46]. Note that this approach can be applied to wireless neural

networks implemented by various platforms.

Implementation of communication-aware in-memory wireless neural networks

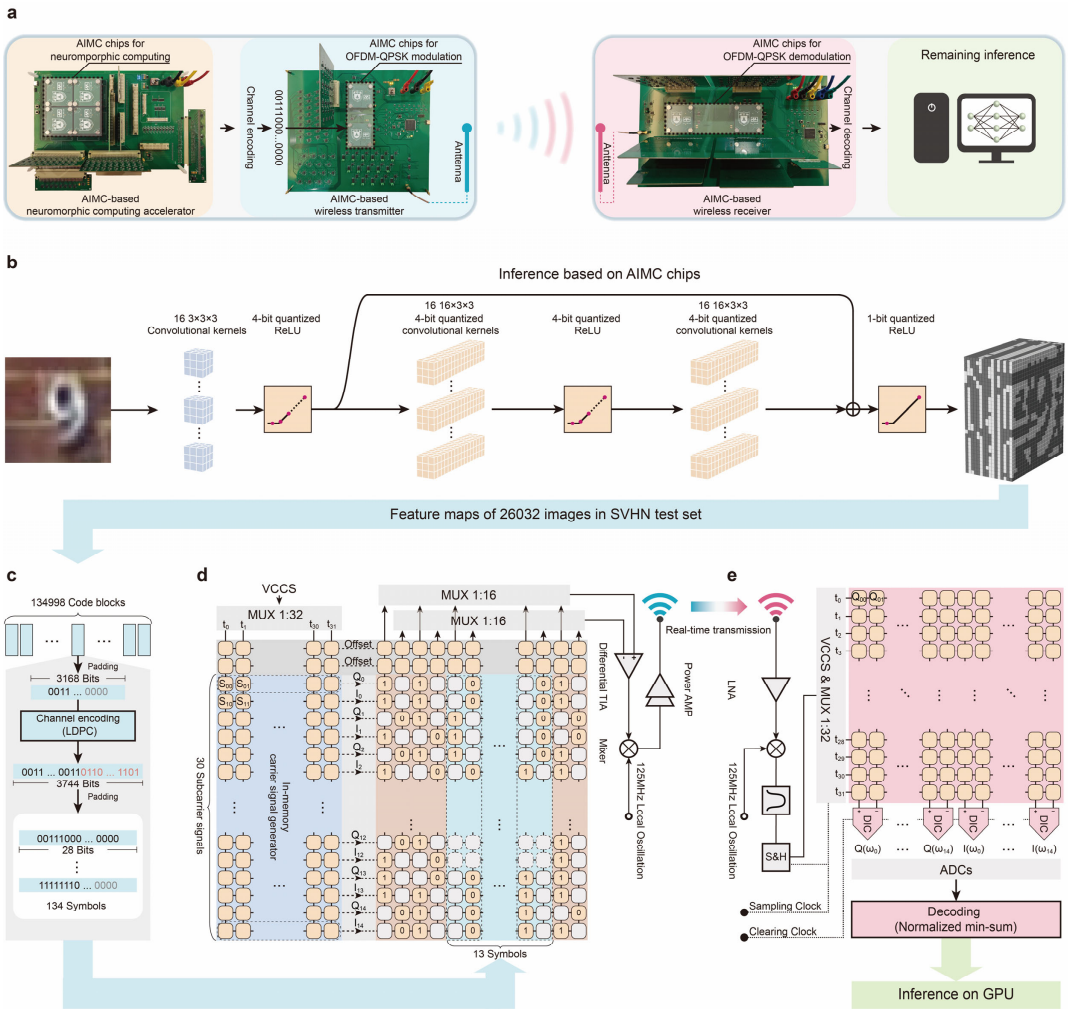

Fig. 2 | The implementation of communication-aware in-memory wireless neural networks. a, The implemented AIMC-based prototype system. **b,** Computational graph of the edge CNN after computational graph optimization and quantization. The edge CNN extracts features from 26032 images in the SVHN test dataset. **c,** Preparations before transmission. The feature data are serialized and divided into 134998 3168-bit code blocks. The bits in each block are encoded through LDPC, padded with zeros and then divided into 134 28-bit symbols. **d,** Schematic diagram of the AIMC-based wireless transmitter. The first AIMC chip, along with peripheral circuits, functions as a subcarrier signal generator. It generates in-phase (I) subcarrier signals and their quadrature (Q) counterparts, which serve as the inputs to the second AIMC chip. In the second AIMC chip, each symbol is stored in a differential pair of columns for modulation. Specifically, 13 pairs of columns (surrounded by dash rectangles) store the symbols for transmission, with 28 valid bits in each symbol. The remaining 3 pairs store 30-bit symbols for controlling the transmission process. The AIMC cells plot with dashed lines are not used in modulation. The modulated signal, carrying the bits stored within the columns, undergoes up-conversion and amplification, finally is transmitted by an antenna. **e,** Schematic diagram of the AIMC-based receiver. The signal collected by the antenna undergoes amplification, down-conversion, and filtering. Two AIMC chips are concatenated for the demodulation of in-phase (I) and quadrature (Q) frequency components. A multiplexer sequentially selects a row to receive the sampled signal as input. Differential integrator circuits (DICs) accumulate the differential outputs of column pairs at each time step. After 32 steps, the accumulated

voltages are digitized by ADCs. Finally, the accumulated voltages of DICs are reset to zeros. The demodulation results of a code block are decoded and used for the remaining inference on a GPU.

We built up a prototype of the proposed in-memory wireless neural network (Fig. 2a). The prototype comprises an AIMC-based neuromorphic computing accelerator, an AIMC-based wireless transmitter, an AIMC-based wireless receiver, and a digital computer. Multiple AIMC chips along with peripheral circuits are assembled together to perform neuromorphic computing and wireless communication. The AIMC-based neuromorphic computing accelerator first extracts features from the raw data through performing the computations of the edge neural network. These feature bits are divided into several symbols and undergo channel encoding. The AIMC-based wireless transmitter modulates and transmits radio signals according to the encoded bits. The transmitted radio signals are then captured by an antenna and demodulated by the AIMC-based wireless receiver. Finally, the decoded data that represent the received features are used to complete the remaining inference of the neural network.

We successfully implemented an in-memory wireless CNN and evaluated its performance on the SVHN test dataset using the AIMC-based prototype. We trained the wireless CNN through the aforementioned communication-aware training approach. During training, the resolution of ADCs is selected as the communication related learnable parameter (Supplementary Text Section 2.1 provides the training details). After training, common model compression approaches were used to streamline the computations of the edge neural network, including computational graph optimization[47] and quantization methods[44,48] (Supplementary Text Section 2.2). The simplified computational graph of the edge neural network is shown in Figure 2b. We quantized all weights and activations to 4 bits, except for the unquantized weights of the first convolutional layer and the final rectified linear unit (ReLU) activation function in the graph. The quantization bit width of the final ReLU (exactly the partition point) plays a crucial role in the trade-off between transmission burdens and neural network performance (Supplementary Fig. S5). We chose 1-bit quantization in this case to minimize transmission burdens. The computations of the edge neural network are performed by the AIMC-based neuromorphic computing accelerator, where a 54×16 array concatenated by two AIMC chips was used to parallelly perform the VMM operations on 4-bit integers in the convolutional process (Detailed in Methods). Through processing of the accelerator, a 24K-bit image was transformed into 16-channel feature maps with a reduced data size of 16×32×32 bits.

The feature bits were transmitted and received in real time through the AIMC-based wireless communication system, which comprises a transmitter and a receiver. During the transmission, the serialized feature bits were encoded using channel coding technologies, such as low-density parity-check code[38] (LDPC, refer to Methods for more details). Subsequently, baseband signals carrying the encoded bits were generated through AIMC-based modulation, and then were upconverted and transmitted in real time (Fig. 2c, d). As depicted in Fig. 2d, the AIMC-based modulation design contains two AIMC chips. The first chip, along with peripheral circuits, generates 30 subcarrier

signals for orthogonal frequency division multiplexing (OFDM), including 15 frequency components and their orthogonal counterparts. These subcarrier signals serve as inputs to the second AIMC chip for quadrature amplitude modulation (QAM), specifically, 4-QAM. Within the second chip, each differential pair of AIMC cells stores one bit for transmission. The differential outputs of these column pairs represent the modulated signals carrying the stored bits. Among these modulated signals, the multiplexers (MUXs) sequentially select one signal as the output. This output signal is mixed with a local oscillation signal, subsequently amplified and transmitted by an antenna. The AIMC-based wireless receiver was used to receive and process these transmitted signals (Fig. 2e). The analog signals collected by the receiver's antenna were amplified, down-converted, filtered, and finally processed by the AIMC-based demodulator. A 32×60 array concatenated by two AIMC chips was used to perform the discrete Fourier transform (DFT) algorithm for the demodulation of in-phase (I) frequency components and quadrature (Q) components. During the demodulation process, one row of the array was sequentially selected by a 1:32 MUX to receive the sampled signal as input at each time step, while leaving other rows without input. For each pair of columns, a differential integrator circuit (DIC) accumulates the differential output of the two columns. The accumulated voltages after 32 steps represent the demodulation results, which then undergo a belief-propagation-like decoding process[49] for error correction, as outlined in Methods. Finally, the feature maps reconstructed from the decoded bits were utilized to complete the remaining inference on a graphics processing unit (GPU).

Performance of in-memory wireless communication

In-memory wireless communication enables reliable feature data transmission. During the wireless collaborative inference on the SVHN test dataset (26032 images), we transmitted and received feature data of 507 megabits (Mb) using the AIMC-based wireless communication system. Figure 3a demonstrates the normalized demodulation results of the received signals by a heatmap-style constellation diagram. The well divided demodulation results of the received 507 Mb features illustrate the reliability of in-memory wireless communication. These demodulation results are decoded and transformed into bits. As shown in Figure 3b, the received bits exhibit a BER of 0.175% before channel decoding (namely, only hard decision decoding is used). With the error correction capability of LDPC, the BER can be reduced to 0.045% after channel decoding. In addition to the errors caused by noisy wireless channels, error bits arising from the noise in AIMC chips can also be effectively corrected through channel decoding (Supplementary Text Section 3.3.3).

Employing AIMC technology for wireless signal processing streamlines the information transmission in wireless neural networks. Since the analog wireless signals have been processed directly within the analog domain by AIMC chips, only the processed results require digitization instead of the complex wireless signals. Therefore, the required resolution of ADCs is significantly reduced for digitalizing the received

information. This feature enables AIMC-based architecture to achieve higher energy efficiency than that of digital solutions (the power consumption estimation is provided in Supplementary Text Section 5.1.2). We compared the in-memory wireless communication and the traditional digital counterpart in terms of their BER's dependence on ADC resolution (Fig. 3c). The in-memory scheme maintains a low BER when employing 1-bit ADCs, while the traditional digital scheme exhibits a severely degraded BER near 15%. More details about the comparison are provided in Methods and Supplementary Text Section 5.1.

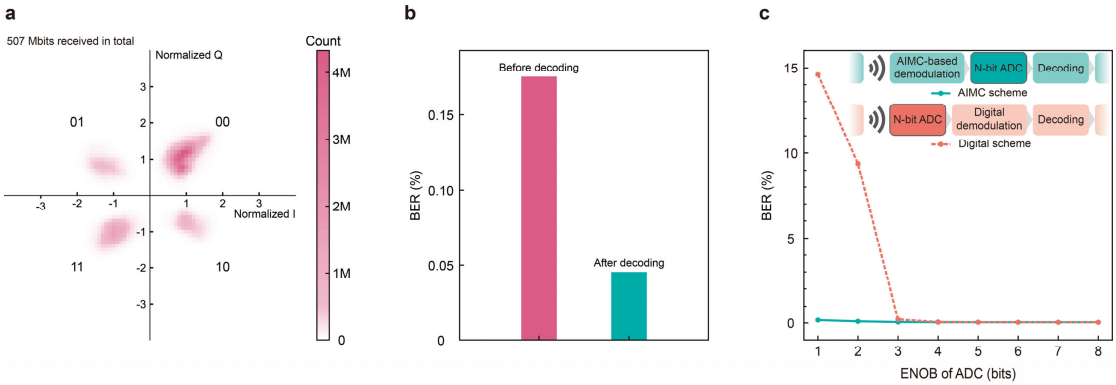

Fig. 3 | Performance of in-memory wireless communication. a, A heatmap-style constellation diagram of the received signals carrying the feature bits. In the diagram, the abscissas and ordinates represent the I and Q components at all the subcarrier frequencies, respectively. The color of an area represents the count of bit pairs (each carried by a pair of orthogonal subcarriers) within it. **b,** Bit error rate of the received 507 Mb features. After decoding, the BER is reduced to 0.045% compared to the raw BER of 0.175%. The real-time transmission of the feature map data is shown in the Supplementary Video. **c,** Comparison between in-memory wireless communication and the digital scheme, in terms of the BER's dependence on the effective number of bits (ENOB) of employed ADCs.

Performance of the communication-aware in-memory wireless neural networks

We next demonstrate the collaborative inference results using the in-memory wireless CNN and evaluate its performance. As shown in Figure 4a, the images in SVHN test dataset are well classified, achieving an inference accuracy of 93.71% with a little fluctuation compared to the reference accuracy (Fig. 4b). Furthermore, based on the features received by employing ADCs of varying resolutions (Fig. 3c), we evaluate the respective inference accuracies and plot the results (Fig. 4c). In comparison to the severe accuracy degradation in traditional scheme, in-memory wireless neural network maintains a nondegraded accuracy even when the 1-bit ADCs is employed, leading to hardware cost reductions in wireless communication. In conjunction with the energy efficiency of in-memory neuromorphic computing, in-memory wireless neural networks offer a reliable and efficient solution for implementing wireless collaborative systems.

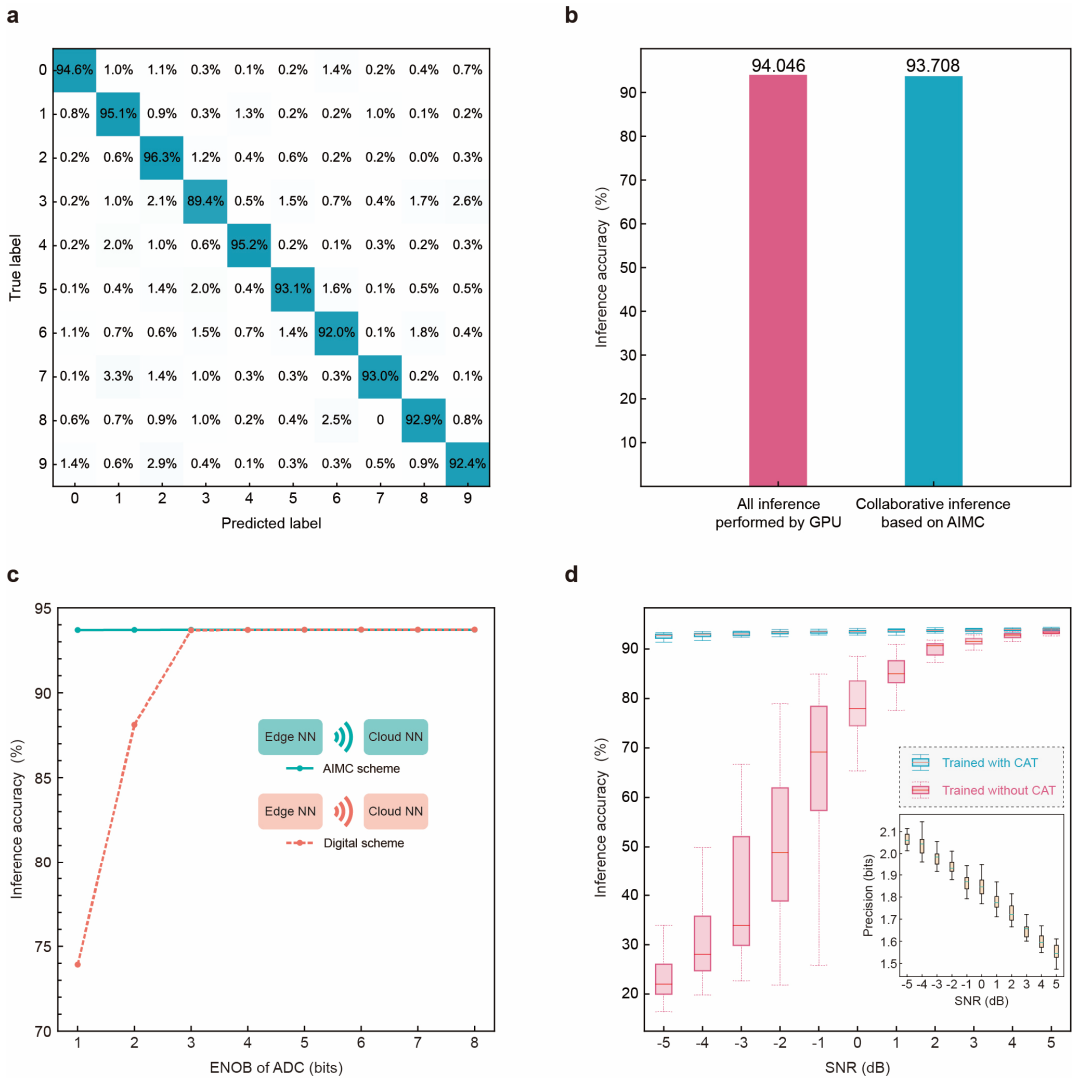

Fig. 4 | Performance of the communication-aware in-memory wireless CNN. a, Confusion matrix of the classification results on the SVHN test dataset. **b,** Inference accuracy on the SVHN test dataset. The implemented in-memory wireless CNN achieves an inference accuracy of 93.71%, close to the ideal inference accuracy. **c,** Respective inference accuracies of the in-memory wireless CNN and the digital scheme when employing ADCs of varying ENOBs in wireless communication. **d,** The reduced hardware requirements of communication-aware in-memory wireless neural networks. We conducted performance comparison between in-memory wireless neural networks trained with and without communication-aware training approach (CAT) based on SVHN dataset. The programming precision of AIMC chips used for wireless communication is selected as a learnable parameter during training. The wireless neural networks are trained and tested under the condition of a constant parameter wireless channel (AWGN channel). A set of simulations based on 20 random seeds (1999, 2009, …, 2189) are conducted for each SNR ranging from -5dB to 5dB. The learned precisions are shown in the inset. The wireless neural networks trained with CAT achieve high inference accuracies even when using low-precision AIMC chips, whereas those non-CAT wireless neural networks experience severe accuracy degradation under the same conditions.

We demonstrate that the proposed communication-aware training approach is of crucial importance to further reducing the hardware requirements of in-memory wireless neural networks. We conducted simulations to evaluate the hardware cost reductions enabled by communication-aware training. Since achieving a high programming precision of the AIMC chips generally requires much programming

cost[50,51] or hardware resources[52], we chose the precision of AIMC chips used for wireless signal processing as a learnable parameter during communication-aware training. Additionally, we employed 1-bit ADCs for digitalization and considered Additive White Gaussian Noise (AWGN) wireless channels with varying SNRs in the simulations. More simulation details are provided in Methods and Supplementary Text Section 5.2. The simulation results (Fig. 4d) show that the learned programming precision of AIMC chips can be reduced to 2 bits (The inset of Fig. 4d) without sacrificing inference accuracy. By contrast, those in-memory wireless neural networks trained without the CAT approach suffer catastrophic accuracy drops under the same conditions. Hence, the CAT approach enables significant further hardware cost reductions for in-memory wireless neural networks.

Furthermore, we demonstrate that the proposed communication-aware training approach enables in-memory wireless neural networks to be applicable to complex and varying wireless conditions. To this end, we add a channel augmentation technique into CAT to emulate the specific impacts of modulator/demodulator hardware, wireless channel characteristics, and modulation schemes on wireless transmission performance (Figs. 5a-c). We conducted simulations based on the ImageNet-1K dataset to validate the applicability of the communication-aware in-memory wireless neural network to various wireless conditions. In the simulation, the wireless neural network is trained once and then tested in various conditions, including employing multiple modulation schemes (4-, 16-, 64-QAM) and transmitting through diverse time-variant wireless channels. During communication-aware training, the average SNR is selected as a learnable parameter, and the wireless-communication-relevant training objective is to minimize the SNR requirements. More details about the simulations and the channel augmentation technique are provided in Methods and Supplementary Text Section 5.3. The evaluation on the training cost is provided in Supplementary Text Section 5.4. The simulation results (Figs. 5d-f) show that the CA-trained wireless neural network achieves high accuracies when using various modulation schemes and transmitting through diverse time-variant wireless channels, which is essential for practical applications, especially for those in mobile scenarios. Moreover, our training approach enables significant reductions in the transmit power of wireless signals, which is quantified as the minimum average SNR (SNR_{min}) for reaching acceptable inference accuracies since the transmit power is proportional to SNR. In comparison with the baseline counterpart (*i.e.*, the model trained without CAT), the required SNR_{min} of the CAT-trained wireless neural network is reduced by 16 dB when transmitting through random time-variant flat fading wireless channels, decreasing the transmit power of wireless signals to 2.5% of that in the traditional approach. Remarkably, in the challenging wireless condition of frequency-selective fading channels, the model trained by our approach maintains high accuracies, whereas the baseline model trained by traditional methods suffers catastrophic accuracy drops despite sufficient SNRs.

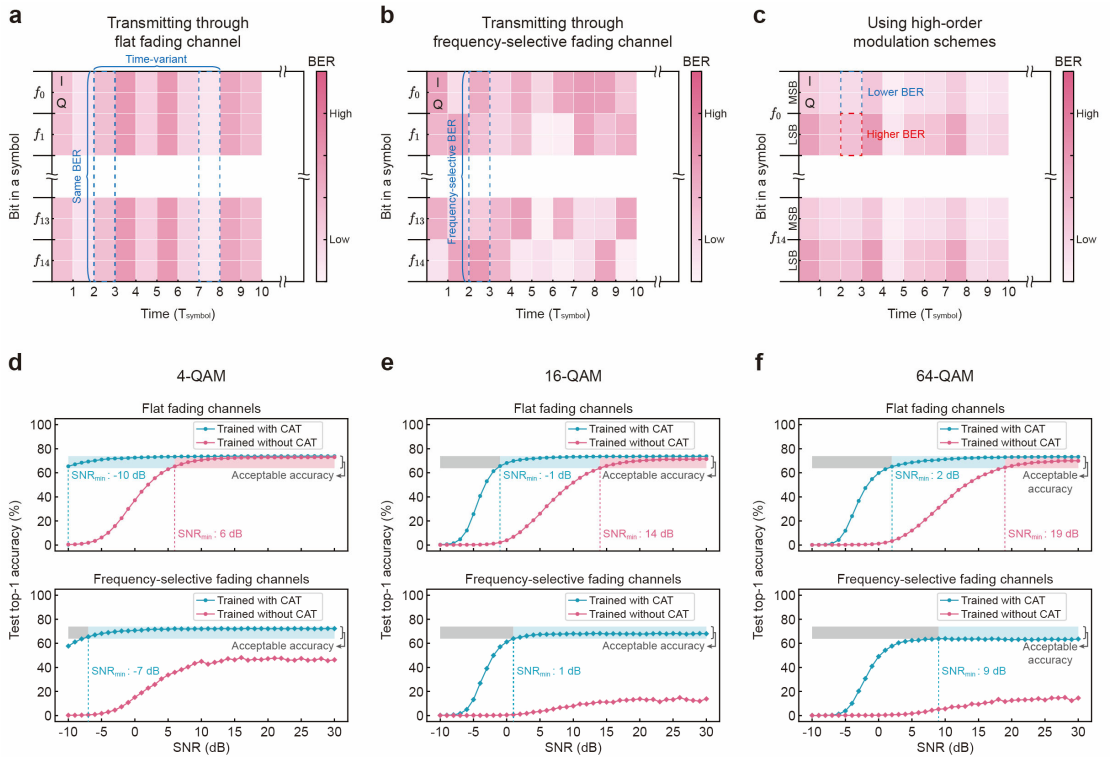

Fig. 5 | Applicability of communication-aware in-memory wireless neural networks to various wireless conditions. a-c, The effects of wireless channel characteristics and modulation schemes on the BER pattern. T_{symbol} denotes the symbol period. The symbol bits share the same time-variant BER when transmitting through a time-variant wireless channel characterized by flat fading (Fig. 5a). By contrast, the bits carried by different subcarriers exhibit different BERs when transmitting through a time-variant frequency-selective channel (Fig. 5b). Additionally, the BER pattern is also influenced by the modulation schemes (Fig. 5c), where the least significant bit (LSB) carried by a subcarrier is more likely to flip compared to the most significant bit (MSB) when using high-order modulation schemes (*e.g.*, 16-QAM). **d-f,** Performance comparison between in-memory wireless neural networks trained with and without the CAT approach. The simulation is conducted on the ImageNet-1K dataset, in which the average SNR is selected as the learnable wireless parameter during communication-aware training. The neural network (ResNet-50) is trained once and then tested in various wireless conditions, including multiple modulation orders (4-, 16-, 64-QAM) and random time-variant wireless channels with diverse channel characteristics (*e.g.*, varying average SNRs, flat fading, and frequency-selective fading). Meanwhile, low-precision AIMC chips (2-, 3-, and 4-bit) and low-precision ADCs (1-, 2-, and 3-bit) are used for modulator/demodulator (4-, 16-, and 64-QAM, respectively). The wireless neural network trained with CAT achieves high accuracies in all test conditions and offers significant benefits in reducing wireless transmit power consumption due to the minimal required average SNR (SNR is proportional to the transmit power) for reaching acceptable inference accuracies (*i.e.*, the top-1 accuracy degradation is less than 10%).

Conclusions

We have reported a novel wireless collaborative computing paradigm based on analog in-memory computing technology, termed communication-aware in-memory wireless neural networks. This paradigm utilizes AIMC technology to reduce the data conversion overhead of modulator/demodulator, and employs a communication-aware training approach to reduce the required BER of wireless transmission, thereby further reducing the energy consumption and hardware overhead of wireless communication systems. We built up a prototype system using the proposed scheme and evaluated its

performance on the SVHN dataset. We show that the prototype system is capable of achieving a nondegraded inference accuracy even when using 1-bit ADCs in wireless communication. Furthermore, we demonstrate that the proposed communication-aware training approach not only improves the applicability of wireless neural networks to various wireless conditions but also further reduces the communication costs such as the required programming precision of AIMC chips and transmit power. Our proposed paradigm paves the way for reliable and efficient intelligent wireless networks covering widespread lower-power edge devices.

Methods

Model structure and dataset

The implemented model is a convolutional neural network composed of 7 convolutional layers and a fully connected layer at the end, as shown in Supplementary Fig. S1. The CNN is partitioned into two parts: the edge neural network and cloud neural network. The first residual block[53] comprising 2 convolutional layers is implemented by AIMC chips, while the computations of other layers behind are performed by a GPU device.

We train and evaluate the model using the SVHN dataset, which comprises RGB images categorized into 10 classes. The dataset is divided into a train set with 73257 images and a test set with 26302 images. The test set is used to evaluate the performance of the implemented in-memory wireless neural network.

Training and quantization of CNN

The wireless collaborative CNN is trained through the proposed communication-aware training approach. During training, the resolution of ADCs used in wireless communication is selected as a learnable parameter. The noisy wireless communication process is modelled as a random bit-flipping process with parameterized probabilities. We use reparameterization tricks to make the bit-flipping process differentiable. The differentiable bit flipping is described as follows. Denotes a bit in a symbol as b_k, it is flipped with a probability of $P_k(\theta)$, where θ represents the learnable parameters of the wireless communication system. The received bit is denoted by $\hat{b}_k$, and is produced as follows:

$$\begin{cases} \hat{b}_k = \dfrac{1 + \left((2b_k - 1) \cdot sgn\left(u_k - P_k(\theta)\right)\right)}{2} \\ u_k \sim U(0,1) \end{cases} \tag{2}$$

The received bit is flipped when the sampled number (denoted by u_k) is less than $P_k(\theta)$. Note that the derivative of the non-differentiable sign function is taken as 1, similar to the trick employed in the straight-through estimator[41] (STE) approach. The detailed illustration to the employed communication-aware training approach is presented in Supplementary Text Section 2.1.

We quantize the parameters of convolutional layers in the edge neural network through several quantization methods. Except for the unquantized weights of the first

convolutional layer and the final 1-bit quantized ReLU activation function, all other weights and activation functions are quantized to 4 bits. In detail, the activation values of the last ReLU in the edge neural network are quantized to 1-bit integers using quantization-aware training, specifically the differentiable soft quantization[44] (DSQ) method. After training, the batch normalization folding method[47] is employed to integrate the parameters of convolutional layers with those of the following batch normalization layers[54]. Other activations are quantized to 4 bits through the analytical clipping for integer quantization (ACIQ) method[48]. The weights of the convolutional layers are quantized to 4-bit integers and then corrected using the bias correction method[48]. More details for these employed quantization methods are presented in Supplementary Text Section 2.2.

Analog in-memory computing chips

The employed AIMC chips are manufactured using 180nm CMOS technology. Each chip comprises 34×32 AIMC cells. The AIMC cell is structured similarly to current mirrors, whose produced output current is proportional to the input current. Each AIMC cell contains a series of metal-oxide-semiconductor field-effect transistors (MOSFETs) with gates of varying dimensions. These MOSFETs share one source MOSFET, which receives an input current I_{in}. Denote the output current of the *i*th MOSFET as I_i. The gain ratio from I_{in} to I_i is determined by the gate dimension ratio of the *i*th MOSFET and the source MOSFET.

$$I_i = \frac{\frac{W_i}{L_i}}{\frac{W_{source}}{L_{source}}} I_{in} = c_i I_{in} \tag{3}$$

W and L represent the width and length of the MOSFET's gate, respectively. There are 8 output MOSFETs in an AIMC cell, each of which is switched by a static random-access memory (SRAM) cell. The *i*th MOSFET copies the input current with a scale of 2^{i-4}, indicating the *i*th bit of an 8-bit integer. The summed output current I_{out} of each cell is proportional to its input current I_{in}, and the proportion factor corresponds to the weight w stored in the cell:

$$I_{out} = I_{in} \times \frac{w}{8} \tag{4}$$

AIMC cells in the same row share the same inputs. The output currents of AIMC cells in a column are summed using Kirchhoff's law. Thus, the AIMC chip is able to perform VMM operations in parallel.

PCB systems for the implementation of in-memory wireless neural networks

We assembled three printed circuit boards (PCBs) functioning as an AIMC-based neuromorphic computing accelerator, an AIMC-based wireless transmitter, and an AIMC-based wireless receiver, respectively.

The AIMC-based neuromorphic computing accelerator consists of digital-analog converters (DACs), voltage-controlled current sources (VCCSs), AIMC chips, transimpedance amplifiers (TIAs), ADCs and a microcontroller. The employed DAC is

the DAC7678 produced by Texas Instruments with 8 channels per chip. Eight DACs are used to provide 54-channel input signals corresponding to the 4-bit quantized feature maps. The used ADC is the ADS131M08 produced by Texas Instruments. ADCs are used to digitize the 16-channel output voltages that represent the results of the VMM operation.

The AIMC-based wireless transmitter is composed of multiplexers (MUXs), VCCSs, two AIMC chips, TIAs, a microcontroller, a mixer and an antenna. The MUXs select the input signal of the AIMC chip used for generating subcarrier signals. The mixer is used for up-conversion of the modulated signals. The final signal is emitted by the antenna.

The AIMC-based wireless receiver contains an antenna, a low noise amplifier (LNA), a mixer, a bandwidth filter (BF), a sample and hold circuit (S/H circuit), MUXs, VCCSs, AIMC chips, TIAs, ADCs, and a microcontroller. The mixer is utilized for down-conversion of the signals received by the antenna. The S/H circuit periodically samples the signals filtered by the BF. MUXs select an input channel of AIMC chips to receive the sampled signal. The used ADC is the ADS7953 produced by Texas Instruments. ADCs digitize the 30-channel results of AIMC-based demodulation.

STM32F407 is selected as the microcontroller in the PCBs, responsible for reading and programming the weights stored in AIMC chips, timing control, and communication with the upper computer.

Hardware implementation of quantized edge neural network

The majority of computation in the quantized edge neural network is simplified to VMM operations on 4-bit integers. A 54×16 AIMC array is employed to perform 4-bit VMM operations. Convolutional kernel weights are vectorized and programmed into the AIMC array. Each column of the AIMC array corresponds to a 6×3×3 section of a convolutional kernel. Every time a 6×3×3 area of the input feature map is vectorized and mapped to the 54-channel outputs of DACs, the 16-channel voltages read by ADCs are exactly the VMM results. The convolutional kernels move across the feature map to obtain the complete convolutional results. After the AIMC-based convolutional process, the calculated integer results undergo dequantization and the addition of bias. Finally, they are activated and quantized by a quantized ReLU activation function. Then the quantized features are used as inputs to the next convolutional layer. More information about the AIMC-based implementation of CNN is provided in Supplementary Text Section 3.1.2.

Encoding process of LDPC

We use LDPC for channel coding, which enables error-correction capabilities through introducing redundant bits. As a linear block code, LDPC linearly maps r bits (denoted by u) in a code block to n extra bits (denoted by v), referred to as parity bits. The parity bits are then concatenated with the raw bits, producing the encoded bits $[u, v]$. The $r \times (r + n)$ matrix G transforming u to $[u, v]$ is called the generator matrix. Meanwhile, a corresponding $n \times (r + n)$ matrix H is named as the parity-check matrix, which is typically sparse. In fact, the n rows of H indicate n

constraint equations of the encoded bits. The encoding and parity-check process are described as follows. Note that binary subtraction and addition are equivalent to the XOR operations.

$$\begin{cases} Gu^T = [u, v]^T \\ H[u, v]^T = 0 \end{cases} \tag{5}$$

In principle, parity bits are produced by solving the above equations. The construction of a quasi-cyclic parity-check matrix H and the detailed encoding process based on H are described in Supplementary Text Section 3.2.2.

Implementation of OFDM-QAM modulation based on AIMC chips

The transmitter contains two AIMC chips for OFDM-QAM modulation (Fig. 2d). The first one, along with peripheral circuits, functions as a subcarrier signal generator. The generator matrix S is programmed into a 30×32 array in the first AIMC chip, where the weight of a single AIMC cell represents an entry in the matrix. The matrix S is described as follows:

$$\begin{cases} S_{2i,j} = round\left(\dfrac{\left(\sin\left(2\pi f_i t_j\right) + 1\right)}{2} \times 255\right) \\ S_{2i+1,j} = round\left(\dfrac{\left(\cos\left(2\pi f_i t_j\right) + 1\right)}{2} \times 255\right) \\ f_i = \dfrac{(i+1)}{2}\ kHz \\ t_j = \dfrac{(j+1)}{f_s} \\ i = 0, 1, \ldots, 14; j = 0, 1, \ldots, 31 \end{cases} \tag{6}$$

Through sequentially selects a column of the 30×32 AIMC array to receive the input current at a sampling frequency f_s (16kHz), 30-channel subcarrier signals are simultaneously generated. Another 2 rows in the AIMC chip are used to compensate the offsets of the generated subcarrier signals in cooperation with the second AIMC chip.

The second AIMC chip is used to generate modulated signals carrying bits stored within it (Fig. 2d). Using a differential scheme, a pair of AIMC cells is used to store a bit. For example, a AIMC cell pair stores the bit 1 through programming the first cell to a constant value while setting the weight in the second cell to 0. The differential output of a column pair represents the modulated signal carrying the stored bits. A 30×32 AIMC array is used for modulation. It takes the 30-channel subcarrier signals as inputs, and generate 16-channel modulated signals, each carrying 30 bits. Thus, the AIMC array is able to store and modulate 16×30 bits. Among the output channels, 13 channels are designated to carry 28-bit symbols for transmission, while the remaining 3 channels are reserved for 30-bit control symbols. The 28-bit symbol for transmission is distinguished from the 30-bit control symbol by leaving the subcarriers Q_{12}, I_{12} not carrying bits. Correspondingly, the cells in the 24th and 25th rows, spanning from the 4th column to the 29th column, are programmed with zeros. The control symbols are used for transmission control, such as the synchronization between transmitter and

receiver. Details about the transmission process and the synchronization are provided in Supplementary Text Section 3.2.5 and Supplementary Text Section 3.4, respectively. Two additional rows in the second AIMC chip are used to compensate the offsets of the subcarriers. Moreover, there is an alternative modulation scheme that does not store the bits but takes them as inputs. Details about the compensation and the alternative scheme are presented in Supplementary Text Section 3.2.4.

Implementation of OFDM-QAM demodulation based on AIMC chips.

As depicted in Fig. 2e, two AIMC chips are used for demodulation. Each pair of AIMC cells corresponds to an entry of the demodulation matrix. The 32×30 matrix programmed into the AIMC chip for the demodulation of in-phase frequency components is denoted as I, and that for the demodulation of quadrature frequency components is denoted as Q. I is described as follows:

$$\begin{cases} value = round\left(abs\left(\cos\left(2\pi f_i t_j\right)\right) \times 255\right) \\ I_{j,2i} = value;\ \ I_{j,2i+1} = 0 \quad (value \geq 0) \\ I_{j,2i} = 0;\ \ I_{j,2i+1} = value \quad (value < 0) \\ f_i = \dfrac{(i+1)}{2}\ kHz \\ t_j = \dfrac{(j+1)}{f_s} \\ i = 0, 1, \dots, 14; j = 0, 1, \dots, 31 \end{cases} \tag{7}$$

Q is defined as:

$$\begin{cases} value = round\left(abs\left(\sin\left(2\pi f_i t_j\right)\right) \times 255\right) \\ Q_{j,2i} = value;\ \ Q_{j,2i+1} = 0 \quad (value \geq 0) \\ Q_{j,2i} = 0;\ \ Q_{j,2i+1} = value \quad (value < 0) \\ f_i = \dfrac{(i+1)}{2}\ kHz \\ t_j = \dfrac{(j+1)}{f_s} \\ i = 0, 1, \dots, 14; j = 0, 1, \dots, 31 \end{cases} \tag{8}$$

The matrices I, Q are illustrated in Supplementary Fig. S15.

The workflow of AIMC-based demodulation is outlined as follows. The down-converted and then filtered signal is sampled at each single time step of the 16 kHz sampling clock. Meanwhile, a 1:32 multiplexer sequentially select one channel to receive the sampled signal s_j as input at each time step, leaving other channels without input. For each pair of columns, a differential integrator circuit accumulates the differential output of the two columns. After 32 steps, the accumulated voltages representing the demodulation result are digitized by ADCs. Finally, the accumulated voltages of differential integrator circuits are reset to zeros for next demodulation cycle. The used differential integrator circuit is illustrated in Supplementary Fig. S16.

Decoding process of LDPC

The decoding process is based on the demodulation results of a received code block. The demodulation results D derive from the voltages digitized by the ADCs in the

AIMC-based receiver:

$$\begin{cases} D^{+} = \dfrac{V^{+} - V_{ref}}{E\left(V^{+} - V_{ref}\right)} \\ D^{-} = \dfrac{V^{-} - V_{ref}}{E\left(V^{-} - V_{ref}\right)} \end{cases} \tag{9}$$

V_{ref} denotes the reference voltage level which equals the voltage accumulated in a demodulation cycle without input signals, and is a preset constant in the experiment. V^{+} and V^{-} denote the voltages that are higher and lower than V_{ref}, respectively. $E\left(V^{+} - V_{ref}\right)$ and $E\left(V^{-} - V_{ref}\right)$ denote the averages of V^{+} and V^{-} in a code block, respectively. The demodulation results of a block are used for subsequent decoding through the normalized min-sum algorithm[49] (NMSA). Further information about the decoding procedures is provided in Supplementary Text Section 3.3.3.

Performance comparison between in-memory wireless neural networks and the digital scheme

The comparison between in-memory wireless neural networks and the digital scheme in terms of the ADC resolution is conducted as follows. ADCs used in the AIMC scheme digitize the demodulation voltages while ADCs used in the digital scheme digitize the sampled wireless signals. In the AIMC scheme, the demodulation voltages are symmetrically quantized by N-bit ADCs to 2^{N} discrete values ranging from $-B$ to B. The value of B is selected based on the demodulation voltage distribution. Afterwards, these quantized voltages are transformed to demodulated signals for decoding and further inference. In contrast to analog demodulation, the digital scheme directly digitizes the sampled signal by N-bit ADCs and then perform demodulation in the digital domain. The sampled signals are calculated from the experimentally collected demodulation voltages as follows:

$$S = V \cdot D_{l}^{+} \tag{10}$$

V is the 1×30 sized vector of demodulation voltages collected by ADCs in a demodulation cycle. D_{l}^{+} is the left inverse matrix of the 32×30 demodulation matrix. The sampled signals are quantized to 2^{N} discrete values ranging from S_{lower} to S_{upper}. The clip bounds S_{lower} and S_{upper} are selected according to the distribution of the sampled signals. Subsequently, the quantized signals are demodulated, then undergo decoding, and finally are used for the remaining inference of the cloud neural network. Further information about the comparison is provided in Supplementary Text Section 5.1.

Analysis of the hardware cost reductions enabled by communication-aware training

We conducted simulations to analyze the hardware cost reductions enabled by the proposed communication-aware training approach. Through simulation, we compared the performance and hardware costs of in-memory wireless neural networks trained with and without communication awareness. The programming precision of AIMC

chips employed for in-memory wireless communication is selected as a learnable wireless communication parameter. Before training, we conducted link-level simulations to evaluate the dependences of BERs on the programming precision, when using additive white Gaussian noise (AWGN) wireless channels with varying SNRs ranging from -5dB to 5dB. Note that 1-bit ADCs are employed in simulations. We fitted these dependences by an extra neural network, whose parameters were frozen (namely, not updated) during communication-aware training. The optimization objective of communication-aware training is:

$$\underset{(\theta_{net},\theta)}{minimize}\ L_{net}(\theta_{net},\theta) + 0.1 \times 2^{\theta-\theta_{max}} \tag{11}$$

where θ_{net} denotes the neural network parameters and θ denotes the programming precision. During training, θ is clamped between θ_{min} (1-bit) and θ_{max} (8-bit).

By contrast, for those wireless neural networks trained without communication awareness, the wireless communication is not integrated into training, and the optimization objective is:

$$\underset{\theta_{net}}{minimize}\ L_{net}(\theta_{net}) \tag{12}$$

We conducted simulations based on 20 random seeds (1999, 2009, …, 2189). The performances of the trained neural networks were evaluated by performing wireless collaborative inference using AWGN wireless channels and AIMC chips with the trained programming precision. Details about the simulations are provided in Supplementary Text Section 5.2.

Applicability of communication-aware wireless neural network to various wireless conditions

We conducted simulations based on ImageNet-1K dataset for validating the applicability of communication-aware wireless neural network to various wireless conditions, as well as demonstrating the energy consumption reduction enabled by the proposed communication-aware training approach. In simulation, we compared the inference accuracies and corresponding energy costs of in-memory wireless neural networks (ResNet-50) trained with and without communication awareness. The average SNRs at the receiver are selected as learnable wireless communication parameters θ. Before training, we conducted link-level simulations to evaluate the dependences of BERs on average SNR, when using various modulation schemes (4-QAM, 16-QAM, 64-QAM) and transmitting through time-variant flat fading wireless channels with varying SNRs ranging from -10 dB to 30 dB. Note that 2-bit, 3-bit, 4-bit AIMC chips are employed for 4-QAM, 16-QAM, 64-QAM, respectively; 1-bit, 2-bit, 3-bit ADCs are used for 4-QAM, 16-QAM, 64-QAM, respectively. We fitted these dependences by an additional neural network, whose parameters were frozen during communication-aware training. The optimization objective of communication-aware training is:

$$\underset{(\theta_{net},\theta)}{minimize}\ L_{net}(\theta_{net},\theta) + \beta \times \sum_{ord=4,16,64} \frac{(\theta^{ord} - \theta_{min}^{ord})}{3(\theta_{max}^{ord} - \theta_{min}^{ord})} \quad (13)$$

where θ_{net} denotes the neural network parameters and θ denotes the average SNRs. During training, θ^{ord} is clamped between θ_{min}^{ord} (-10 dB, -5 dB, 0 dB for 4-QAM, 16-QAM, 64-QAM, respectively) and θ_{max}^{ord} (20 dB, 25 dB, 30 dB for 4-QAM, 16-QAM, 64-QAM, respectively).

By contrast, for those wireless neural networks trained without communication awareness, the wireless communication is not integrated into training, and the optimization objective is:

$$\underset{\theta_{net}}{minimize}\ L_{net}(\theta_{net}) \quad (14)$$

The trained wireless neural networks are tested in various wireless conditions: employing various modulation schemes (4-QAM, 16-QAM, 64-QAM), undergoing wireless channels with various channel characteristics (varying SNRs and different fading characteristics). Supplementary Text Section 5.3 provides details about the simulations, including the link-level simulation of wireless communication, the proposed channel augmentation techniques and their corresponding ablation studies.

Acknowledgements

This work was supported in part by the National Key R&D Program of China (2023YFF1203600, 2023YFF0718400), the National Natural Science Foundation of China (62122036, 12322407, 62034004, 61921005, 12074176, 62305155), the Leading-edge Technology Program of Jiangsu Natural Science Foundation (BK20232004), and the Fundamental Research Funds for the Central Universities (02042103031). F.M. and S.J.L. would like to acknowledge supports from the AIQ foundation and the e-Science Center of Collaborative Innovation Center of Advanced Microstructures. C.W. would like to acknowledge support from the Xiaomi Young Scholar Foundation. The microfabrication center of the National Laboratory of Solid State Microstructures (NLSSM) is also acknowledged for technical support.

Author Contributions

Z.Z.Y., and C.W. conceived the idea and designed the experiments. F.M., S.J.L., and C.W. supervised the whole project. Z.Z.Y. set up the hardware platform and conducted the experiments and simulations. C.W., Y.C.Z., and G.J.R. aided experiment design. Y.C.Z., G.J.R., and X.J.Y. provided help in the device fabrication and circuit assembly. Y.Y., C.P., and B.C. provided help in discussing the results. Z.Z.Y., C.W., S.J.L. and F.M. co-wrote the manuscript.

Competing interests

Authors declare that they have no competing interests.

Data availability

All data are available within the article and the Supplementary Information, and from the corresponding authors upon reasonable request.

Code availability

All algorithms and codes supporting the findings of this study are available in the article and the Supplementary Information, and from the corresponding authors upon reasonable request.

References

1 Zhang, S. *et al.* Towards Real-time Cooperative Deep Inference over the Cloud and Edge End Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **4**, Article 69 (2020).

2 Banitalebi-Dehkordi, A. *et al.* Auto-Split: A General Framework of Collaborative Edge-Cloud AI. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* 2543–2553 (2021).

3 Yang, H. H., Chen, Z., Quek, T. Q. S. & Poor, H. V. Revisiting Analog Over-the-Air Machine Learning: The Blessing and Curse of Interference. *IEEE Journal of Selected Topics in Signal Processing* **16**, 406-419 (2022).

4 Duan, S. *et al.* Distributed Artificial Intelligence Empowered by End-Edge-Cloud Computing: A Survey. *IEEE Communications Surveys & Tutorials* **25**, 591-624 (2023).

5 Prezioso, M. *et al.* Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* **521**, 61-64 (2015).

6 Wang, Z. *et al.* Fully memristive neural networks for pattern classification with unsupervised learning. *Nature Electronics* **1**, 137-145 (2018).

7 Cai, F. *et al.* A fully integrated reprogrammable memristor–CMOS system for efficient multiply–accumulate operations. *Nature Electronics* **2**, 290-299 (2019).

8 Chen, W.-H. *et al.* CMOS-integrated memristive non-volatile computing-in-memory for AI edge processors. *Nature Electronics* **2**, 420-428 (2019).

9 Li, C. *et al.* Long short-term memory networks in memristor crossbar arrays. *Nature Machine Intelligence* **1**, 49-57 (2019).

10 Krestinskaya, O., James, A. P. & Chua, L. O. Neuromemristive Circuits for Edge Computing: A Review. *IEEE Transactions on Neural Networks and Learning Systems* **31**, 4-23 (2020).

11 Liu, Q. *et al.* 33.2 A Fully Integrated Analog ReRAM Based 78.4TOPS/W Compute-In-Memory Chip with Fully Parallel MAC Computing. In *2020 IEEE International Solid-State Circuits Conference - (ISSCC)* 500-502 (2020).

12 Qin, Y.-F. *et al.* Recent Progress on Memristive Convolutional Neural Networks for Edge Intelligence. *Advanced Intelligent Systems* **2**, 2000114 (2020).

13 Yao, P. *et al.* Fully hardware-implemented memristor convolutional neural network. *Nature* **577**, 641-646 (2020).

14 Kiani, F., Yin, J., Wang, Z., Yang, J. J. & Xia, Q. A fully hardware-based memristive multilayer neural network. *Science Advances* **7**, eabj4801 (2021).
15 Xue, C.-X. *et al.* A CMOS-integrated compute-in-memory macro based on resistive random-access memory for AI edge devices. *Nature Electronics* **4**, 81-90 (2021).
16 Huo, Q. *et al.* A computing-in-memory macro based on three-dimensional resistive random-access memory. *Nature Electronics* **5**, 469-477 (2022).
17 Wan, W. *et al.* A compute-in-memory chip based on resistive random-access memory. *Nature* **608**, 504-512 (2022).
18 Ye, W. *et al.* A 28-nm RRAM Computing-in-Memory Macro Using Weighted Hybrid 2T1R Cell Array and Reference Subtracting Sense Amplifier for AI Edge Inference. *IEEE Journal of Solid-State Circuits* **58**, 2839-2850 (2023).
19 Zhang, W. *et al.* Edge learning using a fully integrated neuro-inspired memristor chip. *Science* **381**, 1205-1211 (2023).
20 Aguirre, F. *et al.* Hardware implementation of memristor-based artificial neural networks. *Nature Communications* **15**, 1974 (2024).
21 Huaqiang Wu, Z. L., Jie Mei et al. Memristor chip-enabled adaptive neuromorphic decoder for co-evolutional brain-computer interfaces. *PREPRINT (Version 1) available at Research Square* (2024).
22 Zhou, F. & Chai, Y. Near-sensor and in-sensor computing. *Nature Electronics* **3**, 664-671 (2020).
23 Chen, Y. *et al.* All-analog photoelectronic chip for high-speed vision tasks. *Nature* **623**, 48-57 (2023).
24 Pi, S., Ghadiri-Sadrabadi, M., Bardin, J. C. & Xia, Q. Nanoscale memristive radiofrequency switches. *Nature Communications* **6**, 7519 (2015).
25 Li, C. *et al.* Analogue signal and image processing with large memristor crossbars. *Nature Electronics* **1**, 52-59 (2018).
26 Kim, M. *et al.* Analogue switches made from boron nitride monolayers for application in 5G and terahertz communication systems. *Nature Electronics* **3**, 479-485 (2020).
27 Wang, C. *et al.* Scalable massively parallel computing using continuous-time data representation in nanoscale crossbar array. *Nature Nanotechnology* **16**, 1079-1085 (2021).
28 Kim, M. *et al.* Monolayer molybdenum disulfide switches for 6G communication systems. *Nature Electronics* **5**, 367-373 (2022).
29 Lanza, M. *et al.* Memristive technologies for data storage, computation, encryption, and radio-frequency communication. *Science* **376**, eabj9979 (2022).
30 Qin, Q. *et al.* Hybrid Precoding with a Fully-Parallel Large-Scale Analog RRAM Array for 5G/6G MIMO Communication System. In *2022 International Electron Devices Meeting (IEDM)* 33.32.31-33.32.34 (2022).
31 Ross, A. *et al.* Multilayer spintronic neural networks with radiofrequency connections. *Nature Nanotechnology* **18**, 1273-1280 (2023).

32 Wang, C. *et al.* Parallel in-memory wireless computing. *Nature Electronics* **6**, 381-389 (2023).

33 Zuo, P., Sun, Z. & Huang, R. Extremely-Fast, Energy-Efficient Massive MIMO Precoding With Analog RRAM Matrix Computing. *IEEE Transactions on Circuits and Systems II: Express Briefs* **70**, 2335-2339 (2023).

34 Kim, D. *et al.* Emerging memory electronics for non-volatile radiofrequency switching technologies. *Nature Reviews Electrical Engineering* **1**, 10-23 (2024).

35 Liu, C. *et al.* VO2 memristor-based frequency converter with in-situ synthesize and mix for wireless internet-of-things. *Nature Communications* **15**, 1523 (2024).

36 Zeng, Q. *et al.* Realizing In-Memory Baseband Processing for Ultrafast and Energy-Efficient 6G. *IEEE Internet of Things Journal* **11**, 5169-5183 (2024).

37 Shannon, C. E. A mathematical theory of communication. *The Bell System Technical Journal* **27**, 379-423 (1948).

38 Gallager, R. Low-density parity-check codes. *IRE Transactions on information theory* **8**, 21-28 (1962).

39 Berrou, C., Glavieux, A. & Thitimajshima, P. Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1. In *Proceedings of ICC'93-IEEE International Conference on Communications* 1064-1070 (1993).

40 Arikan, E. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on information Theory* **55**, 3051-3073 (2009).

41 Bengio, Y., Léonard, N. & Courville, A. C. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. *ArXiv* **abs/1308.3432** (2013).

42 Rastegari, M., Ordonez, V., Redmon, J. & Farhadi, A. XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. In *Computer Vision – ECCV 2016* 525-542 (2016).

43 Jacob, B. *et al.* Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition* 2704-2713 (2018).

44 Gong, R. *et al.* Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In *Proceedings of the IEEE/CVF international conference on computer vision* 4852-4861 (2019).

45 Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014).

46 Goodfellow, I. J., Shlens, J. & Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

47 Krishnamoorthi, R. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342* (2018).

48 Banner, R., Nahshan, Y. & Soudry, D. Post training 4-bit quantization of

convolutional networks for rapid-deployment. *Advances in Neural Information Processing Systems* **32** (2019).

49 Chen, J. & Fossorier, M. P. Near optimum universal belief propagation based decoding of low-density parity check codes. *IEEE Transactions on communications* **50**, 406-414 (2002).

50 Zhao, M., Gao, B., Tang, J., Qian, H. & Wu, H. Reliability of analog resistive switching memory for neuromorphic computing. *Applied Physics Reviews* **7**, 011301 (2020).

51 Rao, M. *et al.* Thousands of conductance levels in memristors integrated on CMOS. *Nature* **615**, 823-829 (2023).

52 Song, W. *et al.* Programming memristor arrays with arbitrarily high precision for analog computing. *Science* **383**, 903-910 (2024).

53 He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* 770-778 (2016).

54 Ioffe, S. & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* 448-456 (2015).