Heterogeneous Transformer: A Scale Adaptable Neural Network Architecture for Device Activity Detection

Yang Li, Zhilin Chen, Yunqi Wang, Chenyang Yang, Bo Ai, and Yik-Chung Wu

***Abstract*—To support modern machine-type communications, a crucial task during the random access phase is device activity detection, which is to identify the active devices from a large number of potential devices based on the received signal at the access point. By utilizing the statistical properties of the channel, state-of-the-art covariance based methods have been demonstrated to achieve better activity detection performance than compressed sensing based methods. However, covariance based methods require to solve a high dimensional nonconvex optimization problem by updating the estimate of the activity status of each device sequentially. Since the number of updates is proportional to the device number, the computational complexity and delay make the iterative updates difficult for real-time implementation especially when the device number scales up. Inspired by the success of deep learning for real-time inference, this paper proposes a learning based method with a customized heterogeneous transformer architecture for device activity detection. By adopting an attention mechanism in the architecture design, the proposed method is able to extract features reflecting relevance among device pilots and received signal, permutation equivariant with respect to devices, and its training parameter number is independent of the device number. Simulation results demonstrate that the proposed method achieves better activity detection performance with much shorter computation time than state-of-the-art covariance approach, and generalizes well to different numbers of devices and BS-antennas, different pilot lengths, transmit powers, and cell radii.**

***Index Terms*—Activity detection, attention mechanism, deep learning, Internet-of-Things (IoT), machine-type communications (MTC).**

The work of Y. Li was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62101349 and Grant 62231019, and in part by the State Key Laboratory of Rail Traffic Control and Safety (Contract No. RCS2022K010), Beijing Jiaotong University. The work of C. Yang was supported by the NSFC under Grant 61731002. *(Corresponding authors: Yang Li; Bo Ai.)*

Y. Li is with Shenzhen Research Institute of Big Data, Shenzhen 518172, China, and also with State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China (e-mail: liyang@sribd.cn).

Z. Chen is with The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: zchen@comm.utoronto.ca).

Y. Wang and Y.-C. Wu are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: {yunqi9@connect, ycwu@eee}.hku.hk).

C. Yang is with the School of Electronics and Information Engineering, Beihang University, Beijing 100191, China (e-mail: cyyang@buaa.edu.cn).

B. Ai is with State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China, with Peng Cheng Laboratory, Shenzhen 518055, China, and with Henan Joint International Research Laboratory of Intelligent Networking and Data Analysis, Zhengzhou University, Zhengzhou 450001, China (e-mail: boai@bjtu.edu.cn).

I. Introduction

To meet the dramatically increasing demand for wireless connectivity of Internet-of-Things (IoT), machine-type communications (MTC) have been recognized as a new paradigm in the fifth-generation and beyond wireless systems. Different from the traditional human-to-human communications, MTC scenarios commonly involve a large number of IoT devices connecting to the network, but only a small portion of the devices are active at any given time due to the sporadic traffics [1]–[3].

Due to the large number of devices in MTC, conventional grant-based access schemes will induce high access latency and signalling overheads [4]. To reduce the latency, a grant-free random access scheme was advocated in [5], [6], where each active device sends data without permissions from the base station (BS). In order to know which devices are active, each device is assigned a unique pilot sequence and the BS detects which pilot sequences are received with activity detection. However, the pilot sequences for device activity detection have to be nonorthogonal, due to the large number of devices but limited coherence time. The nonorthogonality of the pilot sequences inevitably induces interference among different devices, and hence complicates the task of device activity detection in MTC.

By exploiting the sporadic nature of MTC, compressed sensing based methods have been adopted to identify the active devices through joint device activity detection and channel estimation [7]–[22]. Specifically, [7]–[9] proposed approximate message passing (AMP) based algorithms to jointly recover the device activity and the instantaneous channel state information. Furthermore, AMP was extended to include data detection [10]–[12] and to multi-cell systems [13]–[15], respectively. In addition to AMP, other compressed sensing based methods, such as Bayesian sparse recovery [16]–[18] and regularization based sparse optimization [19]–[22] have also been investigated for joint device activity detection and channel estimation.

Different from the compressed sensing based methods, another approach utilizes the statistical distribution of the channel without the need of estimating the instantaneous channel state information. This approach is referred to as the covariance based methods, since they are based on the sample covariance matrix of the received signal [23]–[31]. The covariance based methods have recently drawn a lot of attention due to the superiority of activity detection perfor-

mance. In particular, the analytical results in [32], [33] show that the required pilot sequence length of the covariance based methods for reliable activity detection is much shorter than that of the compressed sensing based methods. While the covariance based methods outperform the compressed sensing based methods due to the advantage of utilizing the statistical properties of the channel, the covariance approach requires to solve a high dimensional nonconvex optimization problem [23]–[31], where the estimate of the activity status of each device is updated sequentially using the coordinate descent method. The sequential nature of the coordinate descent method implies that the number of updates is proportional to the total number of devices. Consequently, the resulting computational complexity and delay make it unsuitable for real-time implementation, especially when the device number is very large.

Recently, deep learning has been exploited to avoid the high computational cost caused by iterative algorithms [34], [35]. Instead of solving each optimization problem instance-by-instance, deep learning utilizes neural networks to represent a mapping function from many problem instances to the corresponding solutions based on a large number of training samples. Once the mapping function is obtained, the neural network can infer the solution of any new problem in a real-time manner. Moreover, by unifying different system modules into an end-to-end manner, deep learning also has the opportunity to learn a better solution than the conventional methods for complex problems [36]–[38].

For device activity detection, a pioneer work [39] proposed a deep learning based method, where each layer of the neural network is constructed by approximating an iteration of the compressed sensing or covariance based methods. Lately, deep unrolling was also applied to mimic various versions of AMP [40]–[42] and sparse optimization methods [43], [44]. While these deep learning based methods achieve better detection performance than the conventional methods, they keep the device pilots fixed during the training stage and only take the received signal as the changeable input of the neural networks. Therefore, the learned mapping function is from the received signal to the device activities without considering the variations of the device pilots. This means that whenever the device pilots (including the device number of the system) are changed, these neural networks need to be retrained, which is a big hurdle to their wide deployment to different scenarios or in changing environment.

Instead of restricting the neural network architecture by an existing iterative algorithm and learning a mapping function from the received signal to the device activities under a fixed configuration of the device pilots, this paper strives to design the neural network architecture that accepts the received signal and device pilots as a pair of changeable inputs, and learn different combinations of the received signal and device pilots. In particular, while the generic multi-layer perceptrons (MLPs) have been widely applied for function approximations, they lack some key properties of the activity detection problem. For example,

- To detect the device activities, the BS should perceive which device pilots are received from the received signal. Therefore, it is beneficial to incorporate a computation mechanism into the neural network to learn the relevance among the received signal and device pilots. However, MLPs do not have such dedicated mechanism for relevance extraction.
- The device activity detection has an inherent permutation equivariance property with respect to devices. To be specific, if the indices of any two devices are exchanged, the neural network should output a corresponding permutation. Incorporating permutation equivariance into the neural network architecture can reduce the parameter space and also avoid a large number of unnecessary permuted training samples [45]–[47]. Unfortunately, the permutation equivariance property is not inherently incorporated in MLPs.
- As the device number scales up, it is highly expected that the neural network is generalizable to larger numbers of devices than the setting in the training procedure. Nevertheless, MLPs are designed for a pre-defined problem size with fixed input and output dimensions, and thus the well-trained MLPs are no longer applicable to a different number of devices.

To incorporate the properties of device activity detection mentioned above, this paper proposes for the first time a heterogeneous transformer architecture. The concept of transformer originates from natural language processing (NLP) [48], in which an attention mechanism is exploited to extract the relevance among different words within a sentence. Based on the relevance extraction, transformer can decide which parts of the source sentence to pay attention to. We observe that there is an analogy between the relevance extraction in NLP and the activity detection problem, since the BS should perceive which device pilots contribute to the received signal by evaluating the relevance among the received signal and device pilots.

Yet different from the NLP tasks where different words belong to the same class of features, and hence are processed by the same set of trainable parameters in vanilla transformer, the received signal and device pilots in device activity detection have different physical meanings. Thus, we use two different sets of parameters to process the representations of the received signal and device pilots, respectively. In this way, the two sets of parameters provide the freedom to represent the received signal and device pilots in different spaces, making the proposed heterogeneous transformer architecture more expressive.

The overall deep neural network consists of an initial embedding layer, multiple heterogeneous transformer encoding layers, and a decoding layer. The initial embedding layer takes the received signal and device pilots as the inputs and produces the initial embeddings. The initial embeddings are further processed through the encoding layers, where the heterogeneous attention mechanism is applied to extract the relevance among the received signal and device pilots. Finally, the decoding layer decides the activity status of each device based on the extracted relevance.

The main contributions of this work are summarized as follows.

1) We provide a novel perspective on how device activity detection can be formulated as a classification problem with

the received signal and device pilots as inputs. Different from the existing works [39]–[44] that learn a mapping function from the received signal to the device activities under a fixed configuration of the device pilots, we learn the mapping function from the pair of the received signal and device pilots to the device activities. By learning different combinations of the received signal and device pilots, the trained heterogeneous transformer is suitable for different configurations of the pilots (and also different numbers of devices). Moreover, instead of iteratively solving an optimization problem instance-by-instance, the proposed learning based method can infer the solution of any new problem in a real-time manner.

2) We further show how the transformer model in NLP can be appropriately extended to work in the device activity detection problem. Instead of restricting the neural network architecture by an existing iterative algorithm, we judiciously design a heterogeneous transformer architecture by leveraging the powerful attention mechanism to approximate the input-output relation of the activity detection problem. In this sense, the designed architecture has more powerful ability for function approximations than the algorithm-driven unrolled architectures [49]. Moreover, by sharing the parameters for producing the representations of different device pilots, the proposed heterogeneous transformer is permutation equivariant with respect to devices, and the dimensions of parameters that require to be optimized during the training procedure are independent of the number of devices. This scale adaptability makes the proposed architecture generalizable to different numbers of devices.
3) Simulation results show that the proposed learning based method using heterogeneous transformer achieves better activity detection performance with much shorter computation time than state-of-the-art covariance approach. The proposed method also generalizes well to different numbers of devices and BS-antennas, different pilot lengths, transmit powers, and cell radii.

The remainder of this paper is organized as follows. System model and existing approaches are introduced in Section II. A novel deep learning perspective on device activity detection is proposed in Section III. A heterogeneous transformer architecture is designed in Section IV. Simulation results are provided in Section V. Finally, Section VI concludes the paper.

Throughout this paper, scalars, vectors, and matrices are denoted by lower-case letters, lower-case bold letters, and upper-case bold letters, respectively. The real and complex domains are denoted by $\mathbb{R}$ and $\mathbb{C}$, respectively. We denote the transpose, conjugate transpose, inverse, real part, and imaginary part of a vector/matrix by $(\cdot)^T$, $(\cdot)^H$, $(\cdot)^{-1}$, $\Re(\cdot)$, and $\Im(\cdot)$, respectively. The $N\times N$ identity matrix and the length-N all-one vector are denoted as $\mathbf{I}_N$ and $\mathbf{1}_N$, respectively. The trace, determinant, and the column vectorization of a matrix are represented as $\mathrm{Tr}(\cdot)$, $|\cdot|$, and $\mathrm{vec}(\cdot)$, respectively. The notation $\odot$ denotes the element-wise product, $\mathbb{I}(\cdot)$ denotes the indicator function, $\mathrm{ReLu}(\cdot)$ denotes the function $\max(\cdot,0)$, and $\mathcal{CN}(\cdot,\cdot)$ denotes the complex Gaussian distribution.

II. System Model and Existing Approaches

A. System Model

Consider an uplink multiple-input multiple-output (MIMO) system with one M-antenna BS and N single-antenna IoT devices. We adopt a block-fading channel model, where the channel from each device to the BS remains unchanged within each coherence block[1]. Let $\sqrt{g_n}\mathbf{h}_n$ denote the channel from the n-th device to the BS, where $\sqrt{g_n}$ and $\mathbf{h}_n \in \mathbb{C}^M$ are the large-scale and small-scale Rayleigh fading components, respectively. Due to the sporadic traffics of MTC, only $K \ll N$ devices are active in each coherence block. If the n-th device is active, we denote the activity status as $a_n = 1$ (otherwise, $a_n = 0$).

To detect the activities of the IoT devices at the BS, we assign each device a unique pilot sequence $\mathbf{s}_n \in \mathbb{C}^{L_\mathrm{p}}$, where L_p is the length of the pilot sequence[2]. Device n transmits the pilot sequence $\mathbf{s}_n$ with transmit power p_n if it is active. Assuming that the transmission from different devices are synchronous, we can model the received signal at the BS as

$$\mathbf{Y} = \sum_{n=1}^{N} \mathbf{s}_n \sqrt{p_n g_n} a_n \mathbf{h}_n^T + \mathbf{W} = \mathbf{S}\mathbf{G}^{\frac{1}{2}}\mathbf{A}\mathbf{H} + \mathbf{W}, \quad (1)$$

where $\mathbf{S} \triangleq [\mathbf{s}_1, \ldots, \mathbf{s}_N] \in \mathbb{C}^{L_\mathrm{p}\times N}$, $\mathbf{G} \triangleq \mathrm{diag}\{p_1 g_1, \ldots, p_N g_N\}$, $\mathbf{A} \triangleq \mathrm{diag}\{a_1, \ldots, a_N\}$, $\mathbf{H} \triangleq [\mathbf{h}_1, \ldots, \mathbf{h}_N]^T \in \mathbb{C}^{N\times M}$, and $\mathbf{W} \in \mathbb{C}^{L_\mathrm{p}\times M}$ is the Gaussian noise at the BS.

This paper aims to detect the activity status $\{a_n\}_{n=1}^N$ based on the received signal at the BS and the device pilots. In many practical deployment scenarios, the devices are stationary, so their large-scale fading channels are fixed [25]–[27] and can be obtained in advance using conventional channel estimation methods [50]. In order to reduce the channel gain variations among different devices, the transmit power of each device can be controlled based on the large-scale channel gain [10]. This is especially beneficial to the devices with relatively weak channel gains.

B. Existing Approaches

Existing approaches for device activity detection can be roughly divided into three categories.

1) Compressed Sensing Based Methods: Denoting $\mathbf{B} \triangleq \mathbf{S}\mathbf{G}^{\frac{1}{2}} \in \mathbb{C}^{L_\mathrm{p}\times N}$ and $\mathbf{X} \triangleq \mathbf{A}\mathbf{H} \in \mathbb{C}^{N\times M}$, compressed sensing based methods obtain the activity status by recovering the row-sparse matrix $\mathbf{X}$ from $\mathbf{Y} = \mathbf{B}\mathbf{X} + \mathbf{W}$. However, since a large amount of instantaneous channel state information requires to be estimated simultaneously, the activity detection performance of compressed sensing based methods cannot compete with that of covariance based methods.

2) Covariance Based Methods: Covariance based methods treat the small-scale fading channel matrix $\mathbf{H}$ as a complex

[1]When the coherence time and signal bandwidth are 1 ms and 200 kHz [26], the channels will remain roughly constant over 200 symbols.

[2]The pilot length L_p is set to be shorter than the coherence block length, so that the channels remain unchanged during the activity detection. Consequently, the performance of the proposed method is affected by the coherence block length only through the pilot length. The detection performance under different pilot lengths is shown in Section V-C.

Gaussian random variable. Specifically, each column of $\mathbf{H}$ is independent and identically distributed (i.i.d.) and follows $\mathcal{CN}(\mathbf{0}, \mathbf{I}_N)$. Together with the fact that each column of the noise $\mathbf{W}$ is i.i.d. and follows $\mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I}_{L_\mathrm{p}})$, the covariance approach models the received signal at each receive antenna $\mathbf{y}_m$ as $\mathcal{CN}(\mathbf{0}, \mathbf{\Sigma})$, where $\mathbf{\Sigma} = \mathbf{SGAS}^H + \sigma^2\mathbf{I}_{L_\mathrm{p}}$. Consequently, the activity status $\{a_n\}_{n=1}^N$ can be detected by maximizing the likelihood function

$$\begin{aligned} p\left(\mathbf{Y}; \{a_n\}_{n=1}^N\right) &= \prod_{m=1}^{M} p\left(\mathbf{y}_m; \{a_n\}_{n=1}^N\right) \\ &= \frac{1}{|\pi\mathbf{\Sigma}|^M} \exp\left(-\mathrm{Tr}\left(\mathbf{\Sigma}^{-1}\mathbf{Y}\mathbf{Y}^H\right)\right), \quad (2) \end{aligned}$$

which is solved by the coordinate descent method that iteratively updates each a_n with $\{a_j\}_{j\neq n}$ fixed. However, since the coordinate descent method requires to update each a_n sequentially, the total iteration number is proportional to the number of devices N, which induces tremendous computational complexity and delay, especially when the device number is massive. Moreover, due to the non-convexity of the optimization problems, the covariance approach can only obtain a stationary point.

3) Deep Learning Based Methods: Existing deep learning based methods tackle the device activity detection problem by approximating an iteration of the conventional methods as a layer of the neural network. In particular, the device pilot matrix $\mathbf{B} = \mathbf{SG}^{\frac{1}{2}}$ is assumed to be fixed, and the unrolled neural network represents a mapping function from the received signal $\mathbf{Y}$ to the activity status $\{a_n\}_{n=1}^N$. With $\mathbf{B}$ unchanged in the test stage, these methods achieve better detection performance than the conventional methods.

In the following two sections, we propose a deep learning based method that takes $\mathbf{Y}$ and $\mathbf{B}$ as a pair of changeable inputs, so that the trained neural network can work for a different set of device pilots. The proposed method consists of interpreting device activity detection as a classification problem (Section III), and a customized neural network architecture (Section IV).

III. A Novel Deep Learning Perspective

A. Device Activity Detection as Classification Problem

In this paper, we strive to learn the activity status $\{a_n\}_{n=1}^N$ without estimating the instantaneous channel $\mathbf{H}$. We can see from (1) that the received signal $\mathbf{Y}$ is actually a weighted sum of the active device pilots $\mathbf{SG}^{\frac{1}{2}}\mathbf{A} = \mathbf{BA}$. To find out which columns of $\mathbf{B}$ contribute to $\mathbf{Y}$, we need to build a computation mechanism to evaluate the relevance between $\mathbf{Y}$ and $\mathbf{B}$. Moreover, since each $a_n \in \{0, 1\}$ is a discrete variable, we can view the activity detection as a classification problem, i.e., classifying each a_n as 0 or 1 based on $\mathbf{Y}$ and $\mathbf{B}$.

Specifically, denote the training data set as $\mathcal{D}$, where the i-th training sample is composed of $\left(\mathbf{Y}^{(i)}, \mathbf{B}^{(i)}, \left\{\tilde{a}_n^{(i)}\right\}_{n=1}^N\right)$, and $\tilde{a}_n^{(i)}$ is the ground-truth label of the n-th device's activity status. We learn a classifier to infer the active probability of each device P_n from $\mathbf{Y}$ and $\mathbf{B}$. Let $f : \mathbb{C}^{L_\mathrm{p}\times M} \times \mathbb{C}^{L_\mathrm{p}\times N} \to [0,1]^N$ denote the mapping function from $(\mathbf{Y}, \mathbf{B})$ to $\mathbf{p} \triangleq [P_1, \ldots, P_N]^T$. We strive to optimize the mapping function $f(\cdot,\cdot)$ such that the difference between the output of the mapping function $\left\{P_n^{(i)}\right\}_{n=1}^N$ and the ground-truth label $\left\{\tilde{a}_n^{(i)}\right\}_{n=1}^N$ is as close as possible. For this purpose, we adopt cross entropy [51] for measuring the discrepancy between $\left\{P_n^{(i)}\right\}_{n=1}^N$ and $\left\{\tilde{a}_n^{(i)}\right\}_{n=1}^N$, and learn the classifier by minimizing the following cross entropy based loss function:

$$\begin{aligned} \min_{f(\cdot,\cdot)} \sum_{i=1}^{|\mathcal{D}|} \Bigg(\frac{2}{N} \sum_{n=1}^{N} \Bigg(& \frac{N-K}{N} \tilde{a}_n^{(i)} \log P_n^{(i)} \\ & + \frac{K}{N} (1 - \tilde{a}_n^{(i)}) \log\left(1 - P_n^{(i)}\right) \Bigg) \Bigg). \quad (3) \end{aligned}$$

Notice that when the number of active devices is equal to that of inactive devices, i.e., $K = N - K$, the loss function (3) will reduce to the standard binary cross-entropy loss [51], which is widely used for balanced classification in machine learning. However, due to the sporadic traffics of MTC, the number of active devices is commonly much less than that of inactive devices, i..e., $K \ll N - K$, and thus the standard binary cross-entropy loss will induce overfitting to the inactive class. In order to avoid overfitting, we put a much larger weight $(N-K)/N$ on the loss corresponding to the sporadic active devices while setting a smaller weight K/N on the loss corresponding to the more common inactive devices in (3).

B. Parametrization by Neural Network

To solve problem (3), we train a neural network (the detailed architecture is given in Section IV) for parameterizing the mapping function $f(\cdot,\cdot)$. During the training procedure, the neural network learns to adjust its parameters for minimizing the loss function (3), so that the neural network can mimic the optimal mapping function from the pair of $\mathbf{Y}$ and $\mathbf{B}$ to $\mathbf{p}$. After training, by inputting any $\mathbf{Y}$ and $\mathbf{B}$ into the neural network, we can compute the corresponding output $\mathbf{p}$ via computationally cheap feed-forward operations. Once $\mathbf{p}$ is obtained, we can use Bernoulli sampling to obtain the activity status of each device. Alternatively, we can adopt a threshold ξ to determine the activity status as $a_n = \mathbb{I}(P_n > \xi)$.

Notice that the training data set is constructed based on the received signal model (1), where the ground-truth labels of the device activities are given. This allows the neural network to mimic the optimal mapping function $f(\cdot,\cdot)$ directly from the ground-truth labels [39]–[41], [43], [44]. Therefore, there is an opportunity to achieve better detection performance than state-of-the-art covariance based methods that only obtain a stationary point.

C. Limitations of MLPs

Although MLPs have been widely used for function representations, they are not suitable for the activity detection problem mainly due to three reasons. First, the active probability of each device P_n should be learned based on the the relevance between the received signal $\mathbf{Y}$ and the scaled pilot matrix $\mathbf{B}$. However, MLPs have no specialized mechanism to extract the

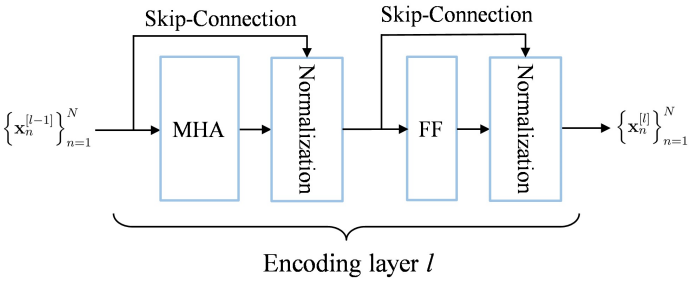


Fig. 1. The l-th encoding layer of the transformer model.

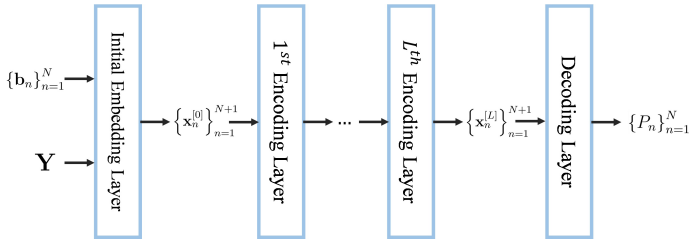


Fig. 2. The overall architecture of the proposed heterogeneous transformer.

relevance between $\mathbf{Y}$ and $\mathbf{B}$. Second, when any two columns of $\mathbf{B}$ are exchanged and $\mathbf{Y}$ is unchanged, $f(\cdot,\cdot)$ should output a corresponding permutation of the original $\mathbf{p}$. Nevertheless, MLPs cannot guarantee the permutation equivariance for the activity detection problem. Last but not the least, as the number of devices scales up, it is highly expected that the neural network is scale adaptable to the device number (i.e., the training parameter number is independent of the device number) and generalizable to larger numbers of devices than the setting in the training procedure. Unfortunately, as the input and output dimensions of MLPs are fixed, they are designed for a pre-defined problem size. Once the number of devices N has changed, the well-trained MLPs are no longer applicable.

IV. Proposed Heterogeneous Transformer for Representing $f(\cdot,\cdot)$

In this section, we propose a customized neural network architecture for representing the mapping function $f(\cdot,\cdot)$. Instead of directly applying MLPs, we strive to incorporate properties of the activity detection problem into the neural network architecture. In particular, the proposed architecture is capable of extracting the relevance between the inputs $\mathbf{Y}$ and $\mathbf{B}$, permutation equivariant with respect to devices, and scale adaptable to different numbers of devices. Before presenting the proposed architecture, we first briefly review the basic idea of the transformer model.

A. Transformer Model

Transformer adopts an encoder-decoder architecture, where the encoder converts the input into a hidden representation by a sequence of encoding layers, while the decoder recovers the output from the hidden representation by a sequence of decoding layers. Since both the encoder and decoder have a similar structure, we only review the architecture of the encoder as follows.

The transformer encoder consists of several sequential encoding layers, where each encoding layer extracts the relevance among the input components. The generated output of each encoding layer is then passed to the next encoding layer as the input. Specifically, each encoding layer mainly consists of two blocks: a multi-head attention (MHA) block that extracts the relevance among different input components, and a component-wise feed-forward (FF) block for additional processing. Each block further adopts a skip-connection [52], which adds an identity mapping to bypass the gradient exploding or vanishing problem for ease of optimization, and a normalization step [53], which re-scales the hidden representations to deal with the internal covariate shift in collective optimization of multiple correlated features.

For better understanding, the l-th encoding layer is illustrated in Fig. 1, where the inputs $\left\{\mathbf{x}_n^{[l-1]}\right\}_{n=1}^N$ are passed to the MHA block and the component-wise FF block successively. The most important module in Fig. 1 is the MHA block, which evaluates the relevance of every pair of components in $\left\{\mathbf{x}_n^{[l-1]}\right\}_{n=1}^N$ by scoring how well they match in multiple attention spaces. By combining all the matching results from different attention spaces, each layer's output is able to capture the relevance among the input components. In the context of NLP, this relevance information reflects the importance of each source word and intuitively decides which parts of the source sentence to pay attention to.

B. Architecture of Proposed Heterogeneous Transformer

The relevance extraction of transformer is appealing for representing the mapping function $f(\cdot,\cdot)$, as the device activity should be detected based on the relevance between the received signal $\mathbf{Y}$ and the scaled pilot matrix $\mathbf{B}$. The overall architecture of the proposed heterogenous transformer is shown in Fig. 2, which is composed of an initial embedding layer, L encoding layers, and a decoding layer. For ease of presentation, the scaled pilot matrix is expanded as $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_N]$, with the column $\mathbf{b}_n$ corresponding to the n-th device. The initial embedding layer takes the N device pilots $\{\mathbf{b}_n\}_{n=1}^N$ and the received signal $\mathbf{Y}$ as the inputs and produces the initial representations $\left\{\mathbf{x}_n^{[0]}\right\}_{n=1}^{N+1}$. Then, the initial representations are updated through L encoding layers to produce $\left\{\mathbf{x}_n^{[l]}\right\}_{n=1}^{N+1}$, $\forall l \in \{1,\ldots,L\}$. Finally, the decoding layer takes $\left\{\mathbf{x}_n^{[L]}\right\}_{n=1}^{N+1}$ as the inputs and decides the active probability $\{P_n\}_{n=1}^N$.

1) Initial Embedding Layer

The initial embedding layer takes the N device pilots $\{\mathbf{b}_n\}_{n=1}^N$ and the received signal $\mathbf{Y}$ as the input features and then transforms them into an initial representation for the subsequent encoding layers. The input features are expressed in real-vector forms by separating the real and imaginary parts. Specifically, the input features corresponding to $\{\mathbf{b}_n\}_{n=1}^N$ are given by

$$\mathbf{x}_n^{\text{in}} = \left[\Re\{\mathbf{b}_n\}^T, \Im\{\mathbf{b}_n\}^T\right]^T \in \mathbb{R}^{2L_{\text{p}}}, \quad \forall n = 1,\ldots,N. \quad (4)$$

On the other hand, to make the proposed neural network architecture scale adaptable to the number of antennas M, we represent the input features corresponding to $\mathbf{Y}$ by vectorizing the sample covariance matrix $\mathbf{C} \triangleq \mathbf{Y}\mathbf{Y}^H/M$:

$$\mathbf{x}_{N+1}^{\text{in}} = \left[\Re\left\{\text{vec}\left(\mathbf{C}\right)\right\}^T, \Im\left\{\text{vec}\left(\mathbf{C}\right)\right\}^T\right]^T \in \mathbb{R}^{2L_{\text{p}}^2}, \quad (5)$$

whose dimension is independent of M. In section V, simulation results will be provided to demonstrate the generalizability with respect to different numbers of antennas M.

Given the input features $\left\{\mathbf{x}_n^{\text{in}}\right\}_{n=1}^{N+1}$, the initial embedding layer applies linear projections to produce the initial embeddings. Let d denote the dimension of the initial embeddings. The linear projections are given by

$$\mathbf{x}_n^{[0]} = \begin{cases} \mathbf{W}_{\text{B}}^{\text{in}}\mathbf{x}_n^{\text{in}} + \mathbf{b}_{\text{B}}^{\text{in}}, & \forall n = 1, \ldots, N, \\ \mathbf{W}_{\text{Y}}^{\text{in}}\mathbf{x}_{N+1}^{\text{in}} + \mathbf{b}_{\text{Y}}^{\text{in}}, & n = N+1, \end{cases} \quad (6)$$

where $\mathbf{W}_{\text{B}}^{\text{in}} \in \mathbb{R}^{d\times 2L_{\text{p}}}$ and $\mathbf{b}_{\text{B}}^{\text{in}} \in \mathbb{R}^d$ are the parameters for projecting the input features $\left\{\mathbf{x}_n^{\text{in}}\right\}_{n=1}^{N}$, while $\mathbf{W}_{\text{Y}}^{\text{in}} \in \mathbb{R}^{d\times 2L_{\text{p}}^2}$ and $\mathbf{b}_{\text{Y}}^{\text{in}} \in \mathbb{R}^d$ are the parameters for projecting the input feature $\mathbf{x}_{N+1}^{\text{in}}$. In (6), the same set of parameters $\left\{\mathbf{W}_{\text{B}}^{\text{in}}, \mathbf{b}_{\text{B}}^{\text{in}}\right\}$ is shared among all devices' pilots, so that the initial embedding layer is scale adaptable to the number of devices in the actual deployment. Furthermore, the input feature corresponding to the received signal is processed heterogeneously by another set of parameters $\left\{\mathbf{W}_{\text{Y}}^{\text{in}}, \mathbf{b}_{\text{Y}}^{\text{in}}\right\}$. The obtained initial embeddings $\left\{\mathbf{x}_n^{[0]}\right\}_{n=1}^{N+1}$ from (6) are subsequently passed to L encoding layers as follows.

2) Encoding Layers

In each encoding layer $l \in \{1, \ldots, L\}$, we adopt the general transformer encoding layer structure in Fig. 1. However, the architectures of MHA, FF, and normalization blocks in this work are different from those of the standard transformer, where all the inputs in a particular layer are processed using the same set of parameters. In contrast, since the device pilots and the received signal have different physical meanings, we use one set of parameters to process the inputs $\left\{\mathbf{x}_n^{[l-1]}\right\}_{n=1}^{N}$ (corresponding to the device pilots), and we process $\mathbf{x}_{N+1}^{[l-1]}$ (corresponding to the received signal) heterogeneously using another set of parameters.

Specifically, the computation in the l-th encoding layer is described by (7) and (8) on the top of the next page, where MHA_{B}^l and MHA_{Y}^l denote the MHA computations, FF_{B}^l and FF_{Y}^l denote the component-wise FF computations, BN_{B}^l and BN_{Y}^l represent the batch normalization (BN) steps [54], and the plus signs represent the skip-connections. The superscript l indicates that different layers do not share parameters, while the subscripts B and Y mean that the representations corresponding to the device pilots and the received signal are computed heterogeneously. In (7), $\mathbf{x}_n^{[l-1]}$ and $\mathbf{x}_{N+1}^{[l-1]}$ are put outside of the set $\left\{\mathbf{x}_j^{[l-1]}\right\}_{j=1,j\neq n}^{N}$, which implies that each $\mathbf{x}_j^{[l-1]}$ is processed in the same way, while $\mathbf{x}_n^{[l-1]}$ and $\mathbf{x}_{N+1}^{[l-1]}$ are processed in a different way from $\left\{\mathbf{x}_j^{[l-1]}\right\}_{j=1,j\neq n}^{N}$. Next, we explain the computations of MHA_{B}, MHA_{Y}, FF_{B}, FF_{Y}, BN_{B}, and BN_{Y} in detail. For notational simplicity, we omit the superscript with respect to l in the following descriptions.

a) MHA Computations: First, we present the MHA computations in (7), where we use T attention heads to extract the relevance among the input components (see Fig. 3(a)). To describe each attention head, we define six sets of parameters $\mathbf{W}_{\text{B},t}^{\text{q}} \in \mathbb{R}^{d'\times d}$, $\mathbf{W}_{\text{Y},t}^{\text{q}} \in \mathbb{R}^{d'\times d}$, $\mathbf{W}_{\text{B},t}^{\text{k}} \in \mathbb{R}^{d'\times d}$, $\mathbf{W}_{\text{Y},t}^{\text{k}} \in \mathbb{R}^{d'\times d}$, $\mathbf{W}_{\text{B},t}^{\text{v}} \in \mathbb{R}^{d'\times d}$, and $\mathbf{W}_{\text{Y},t}^{\text{v}} \in \mathbb{R}^{d'\times d}$, where d' is the dimension of each attention space and $t \in \{1, \ldots, T\}$. For the t-th attention head, it computes a query $\mathbf{q}_{n,t}$, a key $\mathbf{k}_{n,t}$, and a value $\mathbf{v}_{n,t}$ for each $\mathbf{x}_n$ (see Fig. 3(b)):

$$\mathbf{q}_{n,t} = \begin{cases} \mathbf{W}_{\text{B},t}^{\text{q}}\mathbf{x}_n, & \forall n = 1, \ldots, N, \\ \mathbf{W}_{\text{Y},t}^{\text{q}}\mathbf{x}_{N+1}, & n = N+1, \end{cases} \quad (9)$$

$$\mathbf{k}_{n,t} = \begin{cases} \mathbf{W}_{\text{B},t}^{\text{k}}\mathbf{x}_n, & \forall n = 1, \ldots, N, \\ \mathbf{W}_{\text{Y},t}^{\text{k}}\mathbf{x}_{N+1}, & n = N+1, \end{cases} \quad (10)$$

$$\mathbf{v}_{n,t} = \begin{cases} \mathbf{W}_{\text{B},t}^{\text{v}}\mathbf{x}_n, & \forall n = 1, \ldots, N, \\ \mathbf{W}_{\text{Y},t}^{\text{v}}\mathbf{x}_{N+1}, & n = N+1, \end{cases} \quad (11)$$

where the heterogeneity is reflected in using $\left\{\mathbf{W}_{\text{B},t}^{\text{q}}, \mathbf{W}_{\text{B},t}^{\text{k}}, \mathbf{W}_{\text{B},t}^{\text{v}}\right\}_{t=1}^{T}$ to project $\{\mathbf{x}_n\}_{n=1}^{N}$ (corresponding to the device pilots) and using $\left\{\mathbf{W}_{\text{Y},t}^{\text{q}}, \mathbf{W}_{\text{Y},t}^{\text{k}}, \mathbf{W}_{\text{Y},t}^{\text{v}}\right\}_{t=1}^{T}$ to project $\mathbf{x}_{N+1}$ (corresponding to the received signal) to different attention spaces. Then, each attention head computes an attention compatibility α_{njt} for evaluating how much $\mathbf{x}_n$ is related to $\mathbf{x}_j$:

$$\alpha_{n,j,t} = \frac{\mathbf{q}_{n,t}^T\mathbf{k}_{j,t}}{\sqrt{d'}}, \quad \forall n = 1, \ldots, N+1, \quad \forall j = 1, \ldots, N+1, \quad \forall t = 1, \ldots, T, \quad (12)$$

and the corresponding attention weight is computed by normalizing $\alpha_{n,j,t}$ in $[0, 1]$:

$$\beta_{n,j,t} = \frac{e^{\alpha_{n,j,t}}}{\sum_{j'=1}^{N+1} e^{\alpha_{n,j',t}}}, \quad \forall n = 1, \ldots, N+1, \quad \forall j = 1, \ldots, N+1, \quad \forall t = 1, \ldots, T. \quad (13)$$

With the attention weight $\beta_{n,j,t}$ scoring the relevance between $\mathbf{x}_n$ and $\mathbf{x}_j$, the attention value of $\mathbf{x}_n$ at the t-th attention head is computed as a weighted sum[3]:

$$\mathbf{x}'_{n,t} = \sum_{j=1}^{N+1} \beta_{n,j,t}\mathbf{v}_{j,t}, \quad \forall n = 1, \ldots, N+1. \quad (14)$$

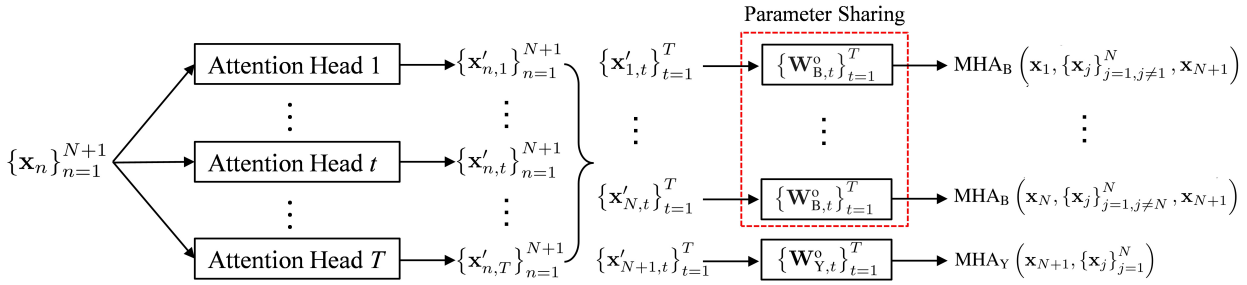
Finally, by combining the attention values from T attention heads with $\left\{\mathbf{W}_{\text{B},t}^{\text{o}} \in \mathbb{R}^{d\times d'}\right\}_{t=1}^{T}$ and $\left\{\mathbf{W}_{\text{Y},t}^{\text{o}} \in \mathbb{R}^{d\times d'}\right\}_{t=1}^{T}$, $\left\{\mathbf{x}'_{n,t}\right\}_{n=1}^{N+1}$ are projected back to d-dimensional vectors and we obtain the MHA computation results (see Fig. 3(a)):

$$\text{MHA}_{\text{B}}\left(\mathbf{x}_n, \{\mathbf{x}_j\}_{j=1,j\neq n}^{N}, \mathbf{x}_{N+1}\right) = \sum_{t=1}^{T} \mathbf{W}_{\text{B},t}^{\text{o}}\mathbf{x}'_{n,t}, \quad \forall n = 1, \ldots, N, \quad (15)$$

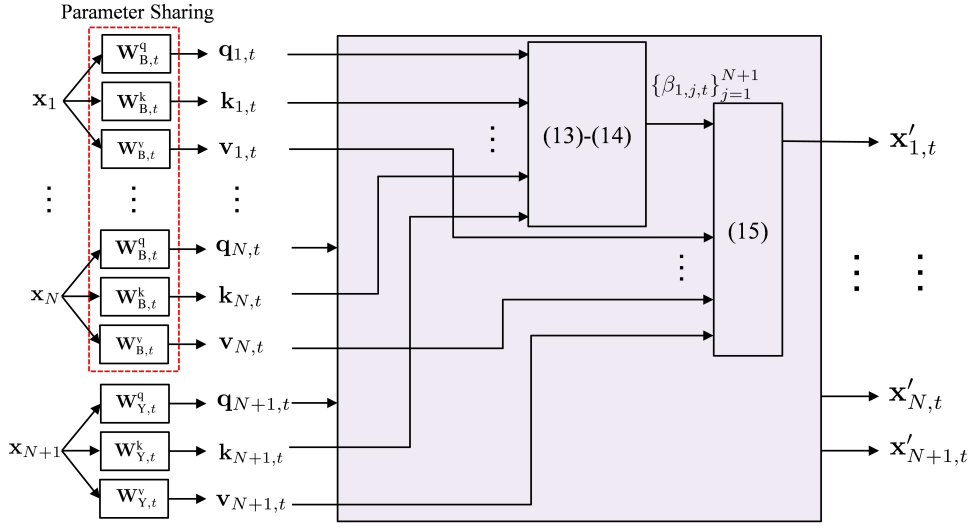
[3]Notice that the summation of (14) is taken over j rather than n. Therefore, $\mathbf{x}'_{n,t}$ serves as the attention value at the t-th attention head corresponding to $\mathbf{x}_n$.

$$\hat{\mathbf{x}}_n^{[l]} = \begin{cases} \mathrm{BN}_{\mathrm{B}}^l\left(\mathbf{x}_n^{[l-1]} + \mathrm{MHA}_{\mathrm{B}}^l\left(\mathbf{x}_n^{[l-1]}, \left\{\mathbf{x}_j^{[l-1]}\right\}_{j=1,j\neq n}^{N}, \mathbf{x}_{N+1}^{[l-1]}\right)\right), & \forall n = 1,\ldots,N, \\ \mathrm{BN}_{\mathrm{Y}}^l\left(\mathbf{x}_{N+1}^{[l-1]} + \mathrm{MHA}_{\mathrm{Y}}^l\left(\mathbf{x}_{N+1}^{[l-1]}, \left\{\mathbf{x}_j^{[l-1]}\right\}_{j=1}^{N}\right)\right), & n = N+1, \end{cases} \tag{7}$$

$$\mathbf{x}_n^{[l]} = \begin{cases} \mathrm{BN}_{\mathrm{B}}^l\left(\hat{\mathbf{x}}_n^{[l]} + \mathrm{FF}_{\mathrm{B}}^l\left(\hat{\mathbf{x}}_n^{[l]}\right)\right), & \forall n = 1,\ldots,N, \\ \mathrm{BN}_{\mathrm{Y}}^l\left(\hat{\mathbf{x}}_{N+1}^{[l]} + \mathrm{FF}_{\mathrm{Y}}^l\left(\hat{\mathbf{x}}_{N+1}^{[l]}\right)\right), & n = N+1. \end{cases} \tag{8}$$



(a) The overall architecture of the MHA block.



(b) The architecture of attention head t of the MHA block.

Fig. 3. The architecture of the MHA block.

$$\mathrm{MHA}_{\mathrm{Y}}\left(\mathbf{x}_{N+1}, \{\mathbf{x}_j\}_{j=1}^N\right) = \sum_{t=1}^{T} \mathbf{W}_{\mathrm{Y},t}^{\mathrm{o}} \mathbf{x}'_{N+1,t}, \tag{16}$$

where (15) and (16) correspond to the projections for the device pilots and the received signal, respectively. Notice that $\mathbf{x}_n$ and $\mathbf{x}_{N+1}$ are put outside of $\{\mathbf{x}_j\}_{j=1,j\neq n}^N$ in (15), because $\mathbf{x}'_{n,t}$ is computed by processing each $\mathbf{x}_j$ using $\left\{\mathbf{W}_{\mathrm{B},t}^{\mathrm{k}}, \mathbf{W}_{\mathrm{B},t}^{\mathrm{v}}\right\}$ in the same way, while by processing $\mathbf{x}_n$ and $\mathbf{x}_{N+1}$ using $\left\{\mathbf{W}_{\mathrm{B},t}^{\mathrm{q}}, \mathbf{W}_{\mathrm{B},t}^{\mathrm{k}}, \mathbf{W}_{\mathrm{B},t}^{\mathrm{v}}\right\}$ and $\left\{\mathbf{W}_{\mathrm{Y},t}^{\mathrm{k}}, \mathbf{W}_{\mathrm{Y},t}^{\mathrm{v}}\right\}$, respectively.

b) FF Computations: Next, we present the computations of FF_{B} and FF_{Y} in (8), which adopt a two-layer MLP with a d_{f}-dimensional hidden layer using the ReLu activation:

$$\mathrm{FF}_{\mathrm{B}}\left(\hat{\mathbf{x}}_n\right) = \mathbf{W}_{\mathrm{B},2}^{\mathrm{f}} \mathrm{ReLu}\left(\mathbf{W}_{\mathrm{B},1}^{\mathrm{f}} \hat{\mathbf{x}}_n + \mathbf{b}_{\mathrm{B},1}^{\mathrm{f}}\right) + \mathbf{b}_{\mathrm{B},2}^{\mathrm{f}}, \quad \forall n = 1,\ldots,N, \tag{17}$$

$$\mathrm{FF}_{\mathrm{Y}}\left(\hat{\mathbf{x}}_{N+1}\right) = \mathbf{W}_{\mathrm{Y},2}^{\mathrm{f}} \mathrm{ReLu}\left(\mathbf{W}_{\mathrm{Y},1}^{\mathrm{f}} \hat{\mathbf{x}}_{N+1} + \mathbf{b}_{\mathrm{Y},1}^{\mathrm{f}}\right) + \mathbf{b}_{\mathrm{Y},2}^{\mathrm{f}}, \tag{18}$$

where $\{\hat{\mathbf{x}}_n\}_{n=1}^{N+1}$ is the output of (7), and $\mathbf{W}_{\mathrm{B},1}^{\mathrm{f}} \in \mathbb{R}^{d_{\mathrm{f}}\times d}$, $\mathbf{b}_{\mathrm{B},1}^{\mathrm{f}} \in \mathbb{R}^{d_{\mathrm{f}}}$, $\mathbf{W}_{\mathrm{B},2}^{\mathrm{f}} \in \mathbb{R}^{d\times d_{\mathrm{f}}}$, $\mathbf{b}_{\mathrm{B},2}^{\mathrm{f}} \in \mathbb{R}^{d}$, $\mathbf{W}_{\mathrm{Y},1}^{\mathrm{f}} \in \mathbb{R}^{d_{\mathrm{f}}\times d}$, $\mathbf{b}_{\mathrm{Y},1}^{\mathrm{f}} \in \mathbb{R}^{d_{\mathrm{f}}}$, $\mathbf{W}_{\mathrm{Y},2}^{\mathrm{f}} \in \mathbb{R}^{d\times d_{\mathrm{f}}}$, and $\mathbf{b}_{\mathrm{Y},2}^{\mathrm{f}} \in \mathbb{R}^{d}$ are the parameters to be optimized during the training procedure. In (17)-(18), the heterogeneity is maintained since we use the same set of parameters $\left\{\mathbf{W}_{\mathrm{B},1}^{\mathrm{f}}, \mathbf{b}_{\mathrm{B},1}^{\mathrm{f}}, \mathbf{W}_{\mathrm{B},2}^{\mathrm{f}}, \mathbf{b}_{\mathrm{B},2}^{\mathrm{f}}\right\}$ to process $\{\hat{\mathbf{x}}_n\}_{n=1}^N$ (corresponding to the device pilots), while we process $\hat{\mathbf{x}}_{N+1}$ (corresponding to the received signal) by another set of parameters $\left\{\mathbf{W}_{\mathrm{Y},1}^{\mathrm{f}}, \mathbf{b}_{\mathrm{Y},1}^{\mathrm{f}}, \mathbf{W}_{\mathrm{Y},2}^{\mathrm{f}}, \mathbf{b}_{\mathrm{Y},2}^{\mathrm{f}}\right\}$.

c) BN Computations: For the BN computations in (7) and (8), it computes the statistics over a batch of training samples. Specifically, let $\left\{\tilde{\mathbf{x}}_n^{(i)} \in \mathbb{R}^d\right\}_{i=1}^{I_{\mathrm{b}}}$ denote a mini-batch of training samples for BN computation. The BN statistics are

calculated as

$$\boldsymbol{\nu}_n = \frac{1}{I_{\rm b}}\sum_{i=1}^{I_{\rm b}}\tilde{\mathbf{x}}_n^{(i)}, \quad \forall n=1,\ldots,N+1, \tag{19}$$

$$\boldsymbol{\Gamma}_n = \left(\frac{1}{I_{\rm b}}\sum_{i=1}^{I_{\rm b}}\boldsymbol{\Lambda}_n^{(i)}\right)^{\frac{1}{2}}, \quad \forall n=1,\ldots,N+1, \tag{20}$$

where $\boldsymbol{\Lambda}_n^{(i)}$ is a diagonal matrix with the diagonal being $\left(\tilde{\mathbf{x}}_n^{(i)}-\boldsymbol{\nu}_n\right)\odot\left(\tilde{\mathbf{x}}_n^{(i)}-\boldsymbol{\nu}_n\right)$. Then, the normalization results corresponding to the device pilots and the received signal are respectively given by

$$\mathrm{BN_B}\left(\tilde{\mathbf{x}}_n^{(i)}\right)=\mathbf{w}_{\rm B}^{\rm bn}\odot\left(\boldsymbol{\Gamma}_n^{-1}\left(\tilde{\mathbf{x}}_n^{(i)}-\boldsymbol{\nu}_n\right)\right)+\mathbf{b}_{\rm B}^{\rm bn}, \quad \forall n=1,\ldots,N, \tag{21}$$

$$\mathrm{BN_Y}\left(\tilde{\mathbf{x}}_{N+1}^{(i)}\right)=\mathbf{w}_{\rm Y}^{\rm bn}\odot\left(\boldsymbol{\Gamma}_{N+1}^{-1}\left(\tilde{\mathbf{x}}_{N+1}^{(i)}-\boldsymbol{\nu}_{N+1}\right)\right)+\mathbf{b}_{\rm Y}^{\rm bn}, \tag{22}$$

where $\mathbf{w}_{\rm B}^{\rm bn}\in\mathbb{R}^d$, $\mathbf{b}_{\rm B}^{\rm bn}\in\mathbb{R}^d$, $\mathbf{w}_{\rm Y}^{\rm bn}\in\mathbb{R}^d$, and $\mathbf{b}_{\rm Y}^{\rm bn}\in\mathbb{R}^d$ are the parameters to be optimized during the training procedure.

3) Decoding Layer

After the L encoding layers, the produced hidden representations $\left\{\mathbf{x}_n^{[L]}\right\}_{n=1}^{N+1}$ are further passed to a decoding layer to output the final mapping result. The proposed decoding layer consists of a contextual block and an output block. The contextual block applies an MHA block to compute a context vector $\mathbf{x}^{\rm c}$, which is a weighted sum of the components in $\left\{\mathbf{x}_n^{[L]}\right\}_{n=1}^{N+1}$:

$$\mathbf{x}^{\rm c}=\mathrm{MHA_C}\left(\mathbf{x}_{N+1}^{[L]},\left\{\mathbf{x}_n^{[L]}\right\}_{n=1}^{N}\right), \tag{23}$$

where $\mathrm{MHA_C}$ is similar to $\mathrm{MHA_Y}$ in (16), but using different parameters $\mathbf{W}_t^{\rm q,c}\in\mathbb{R}^{d'\times d}$, $\mathbf{W}_{{\rm B},t}^{\rm k,c}\in\mathbb{R}^{d'\times d}$, $\mathbf{W}_{{\rm Y},t}^{\rm k,c}\in\mathbb{R}^{d'\times d}$, $\mathbf{W}_{{\rm B},t}^{\rm v,c}\in\mathbb{R}^{d'\times d}$, $\mathbf{W}_{{\rm Y},t}^{\rm v,c}\in\mathbb{R}^{d'\times d}$, $\mathbf{W}_t^{\rm o,c}\in\mathbb{R}^{d\times d'}$, $t\in\{1,\ldots,T\}$. In particular, $\left\{\mathbf{W}_{{\rm B},t}^{\rm k,c},\mathbf{W}_{{\rm B},t}^{\rm v,c}\right\}_{t=1}^{T}$ is used to process $\left\{\mathbf{x}_n^{[L]}\right\}_{n=1}^{N}$ (corresponding to the device pilots), and $\left\{\mathbf{W}_t^{\rm q,c},\mathbf{W}_{{\rm Y},t}^{\rm k,c},\mathbf{W}_{{\rm Y},t}^{\rm v,c},\mathbf{W}_t^{\rm o,c}\right\}_{t=1}^{T}$ is used to process $\mathbf{x}_{N+1}^{[L]}$ (corresponding to the received signal). The specific expression of $\mathrm{MHA_C}$ is shown in Appendix A. Each weight in $\mathbf{x}^{\rm c}$ reflects the importance of each component in $\left\{\mathbf{x}_n^{[L]}\right\}_{n=1}^{N+1}$. Therefore, the context vector $\mathbf{x}^{\rm c}$ intuitively decides which device pilots to pay attention to based on the received signal.

With $\mathbf{x}^{\rm c}$, the output block decides the final output, i.e., the active probability of each device, by scoring how well the context vector $\mathbf{x}^{\rm c}$ and each $\mathbf{x}_n^{[L]}$, $n\in\{1,\ldots,N\}$ match. The relevance between the context vector $\mathbf{x}^{\rm c}$ and each $\mathbf{x}_n^{[L]}$ is evaluated by

$$\alpha_n^{\rm out}=C\tanh\left(\frac{\left(\mathbf{x}^{\rm c}\right)^T\mathbf{W}_{\rm out}\mathbf{x}_n^{[L]}}{\sqrt{d}}\right), \quad \forall n=1,\ldots,N, \tag{24}$$

where $\mathbf{W}_{\rm out}\in\mathbb{R}^{d\times d}$ is a parameter to be optimized during the training procedure, and C is a tuning hyperparameter that controls $\alpha_n^{\rm out}$ in a reasonable range. Finally, the active probability of each device is computed by normalizing $\alpha_n^{\rm out}$ in $[0,1]$:

$$P_n=\mathrm{OUT}\left(\mathbf{x}^{\rm c},\mathbf{x}_n^{[L]}\right)=\frac{1}{1+e^{-\alpha_n^{\rm out}}}, \quad \forall n=1,\ldots,N. \tag{25}$$

C. Key Properties and Insights

The proposed heterogenous transformer for representing $f(\cdot,\cdot)$ has been specified as an initial embedding layer, L encoding layer, and a decoding layer as shown in (4)-(25). We examine some key properties of the proposed architecture for the activity detection problem as follows.

a) *Relevance Extraction Among Device Pilots and Received Signal:* Both the proposed encoding and decoding layers are built on MHA as shown in (7) and (23), respectively. The MHA computation is naturally a weighted sum as shown in (14). The attention weight $\beta_{n,j,t}$ is the normalization of the attention compatibility $\alpha_{n,j,t}$ in (12), which scores how well each pair of $\mathbf{x}_n$ and $\mathbf{x}_j$ match. Therefore, with the attention weight $\beta_{n,j,t}$ reflecting the importance of each $\mathbf{x}_j$ with respect to $\mathbf{x}_n$, each encoding layer learns the relevance among different device pilots and the received signal. In the decoding layer, the captured relevance is further used to compute the context vector $\mathbf{x}^{\rm c}$ in (23), which finally extracts the relevance between each device pilot and the received signal, and decides which device pilots to pay attention to based on the extracted relevance.

b) *Permutation Equivariant with Respect to Devices:* The proposed heterogeneous transformer architecture enjoys the following permutation equivariance property.

Proposition 1. *(Permutation Equivariance in Heterogeneous Transformer) Viewing the input-output mapping of the proposed heterogeneous transformer in Section IV-B as* $\mathbf{p}=f(\mathbf{Y},\mathbf{B})$ *and letting* $\boldsymbol{\Pi}$ *denote a column permutation matrix, we have* $\boldsymbol{\Pi}^T\mathbf{p}=f(\mathbf{Y},\mathbf{B}\boldsymbol{\Pi})$.

Proof: See Appendix B. ∎

Proposition 1 implies that the proposed architecture is inherently incorporated with the permutation equivariance property. This is in sharp contrast to the generic MLPs, which require all permutations of each training sample to approximate this property. In this sense, the proposed architecture highly reduces the sample complexity and training difficulty.

c) *Scale Adaptable and Generalizable to Different Numbers of Devices:* In all the layers of the proposed heterogeneous transformer, the representations of different device pilots are produced with the same architecture using the same set of parameters. Therefore, the dimensions of parameters that require to be optimized during the training procedure are independent of the number of devices. This scale adaptability empowers the whole architecture to be readily applied to scenarios with any number of devices, and hence generalizable to different numbers of devices.

Remark 1: Graph neural networks (GNNs) have been recently exploited to model resource allocation problems in wireless networks and demonstrated superior performance, scalability,

Algorithm 1 Learning Procedure for Activity Detection

1: **Training Procedure:**
2: **Input:** number of epochs N_{e}, steps per epoch N_{s}, batch size N_{b}, and learning rate decay epoch N_{d} and factor β
3: **Initialize:** learning rate η
4: **for** epoch $= 1, \ldots, N_{\mathrm{e}}$
5: **for** step $= 1, \ldots, N_{\mathrm{s}}$
6: a) Generate a batch of N_{b} samples
b) Compute the mini-batch gradient of the loss function (3) over the parameters of heterogeneous transformer
c) Update the parameters by a gradient descent step using the Adam optimizer with learning rate η
7: **end**
8: **if** epoch $== N_{\mathrm{d}}$
9: $\eta \leftarrow \beta\eta$
10: **end**
11: **end**
12: **Output:** heterogeneous transformer with optimized parameters
13: **Test Procedure:**
14: **Input:** N_{t} test samples
15: Compute the output of the trained heterogeneous transformer $P_n^{(i)}$, $n = 1, \ldots, N$, $i = 1, \ldots, N_{\mathrm{t}}$
16: Determine the activity status of each device as $a_n^{(i)} = \mathbb{I}(P_n^{(i)} > \xi)$, $n = 1, \ldots, N$, $i = 1, \ldots, N_{\mathrm{t}}$
17: **Output:** (PM, PF) pairs under different ξ

and generalization ability [45]–[47]. Transformer is in fact a special case of GNNs on a complete graph [55]. We consider a complete graph because in the activity detection problem, it is more helpful to extract the relevance among the received signal and all device pilots, which is achieved by the incorporated attention mechanism.

D. Learning Procedure

So far, we have presented the architecture and key properties of the proposed heterogeneous transformer. Next, we show the learning procedure to optimize the parameters of heterogeneous transformer for device activity detection in Algorithm 1, which consists of a training procedure and a test procedure. As shown in lines 8-10, we adopt a learning rate decay strategy to accelerate the training procedure [56]. In particular, the learning rate η is decreased by a factor of β after N_{d} training epochs. During the test procedure, we adopt two metrics to assess the performance of device activity detection, i.e., the probability of missed detection (PM) and the probability of false alarm (PF) [3], [5]–[7], which are respectively given by

$$\mathrm{PM} = 1 - \frac{\sum_{n=1}^{N} a_n \tilde{a}_n}{\sum_{n=1}^{N} \tilde{a}_n}, \quad \mathrm{PF} = \frac{\sum_{n=1}^{N} a_n \left(1 - \tilde{a}_n\right)}{\sum_{n=1}^{N} (1 - \tilde{a}_n)}. \quad (26)$$

In (26), $\tilde{a}_n$ is the ground-truth device activity, and the detected activity status $a_n = \mathbb{I}(P_n > \xi)$, where ξ is a threshold that increases in $[0, 1]$ to realize a trade-off between PM and PF.

V. Simulation Results

In this section, simulation results are provided to demonstrate the benefits of the proposed learning based method.

A. Simulation Setting

We consider an uplink MIMO system with IoT devices uniformly distributed within a cell with a 250-meter radius, and the ratio of the active devices to the total devices is 0.1. Both the training and test samples are generated as follows. The pilot sequence of each device is an independently generated complex Gaussian distributed vector with i.i.d. elements and each element is with zero mean and unit variance. The large-scale fading coefficient is generated according to the path-loss model $128.1 + 37.6 \log_{10} D_n$ in dB, where D_n is the distance in kilometers between the n-th device and the BS, and the small-scale Rayleigh fading coefficient follows $\mathcal{CN}(0, 1)$. In order to reduce the channel gain variations among different devices especially for the cell-edge devices, the transmit power of each device is controlled as $p_n = p_{\max} \frac{g_{\min}}{g_n}$ [10], where $p_{\max}$ is the maximum transmit power and $g_{\min}$ is the minimum large-scale channel gain in the cell. The maximum transmit power $p_{\max}$ is set from 11 dBm to 23 dBm, and the background Gaussian noise power at the BS is -99 dBm. Based on these settings, the SNR at the BS varies from 2.19 dB to 14.19 dB as $p_{\max}$ increases from 11 dBm to 23 dBm. The received signal is generated according to (1), where the activity of each device is generated from Bernoulli distribution and used as the ground-truth label for the training samples. The hyper-parameters of the proposed heterogeneous transformer are summarized in Table I.

B. Performance Evaluation

First, we show the training loss of Algorithm 1 for updating the parameters of heterogeneous transformer. During the training procedure, the number of devices is set as $N = 100$ and the maximum transmit power is $p_{\max} = 23$ dBm. The length of each pilot sequence is set as $L_{\mathrm{p}} = 7$ or 8, and the number of BS-antennas is set as $M = 32$ or 64, respectively. The training losses versus epochs under different settings are illustrated in Fig. 4. It can be seen that the training losses generally decrease as the training epoch increases. In particular, due to the learning rate decay, the training losses have a sudden decrease in the 90-th training epoch, which demonstrates the effectiveness of learning rate decay in speeding up the training procedure. We also observe from Fig. 4 that the training performance can be improved by increasing the the length of pilot sequence or equipping with a larger number of antennas at the BS.

Then, we test the corresponding activity detection performance of the well-trained heterogeneous transformer in terms of PM and PF. For comparison, we also provide the simulation results of state-of-the-art covariance approach [25], [26] and the MLP based method in Fig. 5, where the proposed learning based method using heterogeneous transformer is termed as HT, the covariance approach is termed as Covariance, and the MLP based method is termed as MLP. In particular, each MLP has 4 hidden layers with batch normalization and ReLu activation, where the first hidden layer has 1024 neurons, and each of the other hidden layers has 512 neurons. Other settings (4-10 hidden layers and 256-1024 neurons in each layer) lead to similar performance and hence are omitted. The number of training samples is the same as that of heterogeneous transformer. It can be seen that the proposed method always achieves better PM-PF trade-offs than those of the covariance approach under different settings. In particular, when $L_{\mathrm{p}} = 8$

TABLE I
HYPER-PARAMETERS OF THE PROPOSED HETEROGENEOUS TRANSFORMER

Parameters	Values	Parameters	Values
Number of encoding layers L	5	Number of training epochs N_{e}	100
Encoding size d	128	Number of steps in each epoch N_{s}	5000
Number of attention heads T	8	Batch size N_{b}	256
Dimension of attention space d'	32	Learning rate η	10^{-4}
Hidden size of the component-wise FF block d_{f}	512	Decay factor β	0.1
Tuning parameter C	10	Number of test samples N_{t}	5000

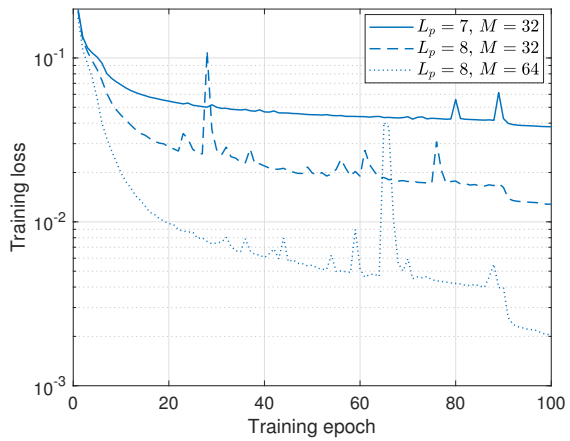


Fig. 4. The training loss of Algorithm 1 for updating the parameters of heterogeneous transformer.

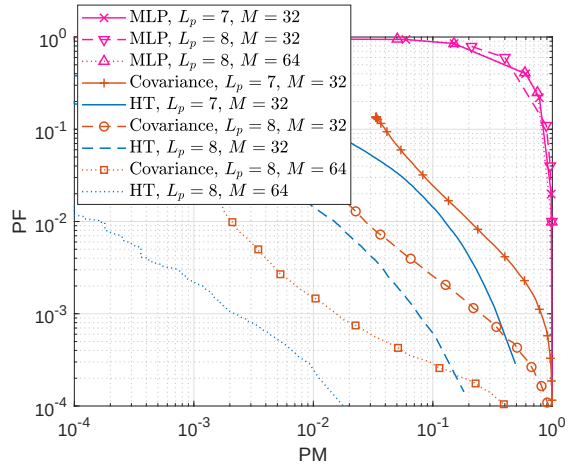


Fig. 5. The test performance comparison in terms of PM and PF.

and $M = 64$, the PM of the proposed method is about 10 times lower than that of the covariance approach under the same PF. This is because the proposed method utilizes neural network to mimic the optimal mapping function directly from the ground-truth training labels, which provide the opportunity to achieve better detection performance than the covariance approach that only finds a stationary point. Moreover, it can be seen that due to the lack of a customized architecture design, MLPs have the worst detection performance, which further verifies the necessity of the proposed heterogeneous transformer architecture. Due to the bad detection performance, we omit the results of MLP in the rest of simulations.

To show the superiority of the proposed method for real-time implementation, we compare the computation time/complexity of the proposed method with that of the covariance approach. The average computation time of the approaches over the test samples are compared in Table II, where the covariance approach is termed as Covariance, and the proposed method is termed as either HT CPU or HT GPU, depending on whether CPU or GPU is used. In particular, both the covariance approach and HT CPU are run on Intel(R) Xeon(R) CPU @ 2.20GHz, while HT GPU is run on Tesla T4. We can see that the average computation time of HT CPU is about 100 times shorter than that of the covariance approach. Moreover, HT GPU achieves a remarkable running speed, with a running time over 10^5 times shorter than that of the covariance approach. Notice that the computation time of the covariance approach is much longer than the coherence time, which is commonly around 1 ms [26]. This makes the detection result of the covariance approach outdated. The computational complexity of the proposed heterogeneous transformer is dominated by the matrix-vector multiplications, especially the multi-head attention and feed-forward computations in the L encoding layers. The corresponding complexity is $\mathcal{O}\left(Ld(N+1)(Td'+d_{\mathrm{f}})\right)$. Notice that in heterogeneous transformer, the above matrix-vector multiplications for different devices, attention heads, and hidden neurons can be executed in parallel. When the computations are fully parallelized, e.g., on powerful GPUs, the time complexity is simply $\mathcal{O}\left(Ld\right)$. In contrast, the computational complexity of the covariance approach is dominated by an $L_{\mathrm{p}} \times L_{\mathrm{p}}$ matrix inversion in each coordinate descent update [23], and hence the complexity of a single update is $\mathcal{O}\left(L_{\mathrm{p}}^3\right)$. Moreover, the covariance approach requires to update the estimate of the activity status of each device in a sequential manner within each iteration, resulting in the overall time complexity $\mathcal{O}\left(INL_{\mathrm{p}}^3\right)$, where I is the number of iterations. Since the covariance approach commonly requires a large number of iterations I for convergence (much larger than the number of encoding layers L in heterogeneous transformer), and the number of devices N is also usually large, its computation time is much longer than that of heterogeneous transformer.

We further compare the performance under different ratios of the active devices to the total devices in Fig. 6. The number of the total devices is set as $N = 100$ and the number of the active devices varies from $K = 8$ to 12. The maximal transmit power is $p_{\max} = 23$ dBm, the pilot length is $L_{\mathrm{p}} = 8$, and the number of BS-antennas is $M = 32$, respectively. It can be seen that as the number of active devices increases, the performance of the two approaches becomes worse. However, the proposed

TABLE II
AVERAGE COMPUTATION TIME COMPARISON AMONG DIFFERENT APPROACHES

	Covariance	HT CPU	HT GPU
$L_{\mathrm{p}}=7,\ M=32$	6.34×10^{-1} s	6.31×10^{-3} s	1.60×10^{-6} s
$L_{\mathrm{p}}=8,\ M=32$	6.24×10^{-1} s	6.32×10^{-3} s	1.63×10^{-6} s
$L_{\mathrm{p}}=8,\ M=64$	6.33×10^{-1} s	6.55×10^{-3} s	2.13×10^{-6} s

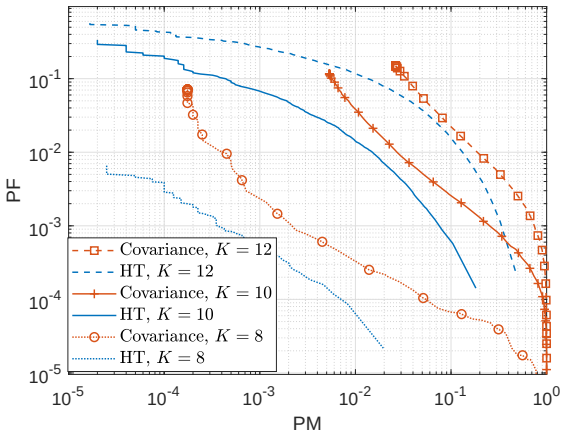
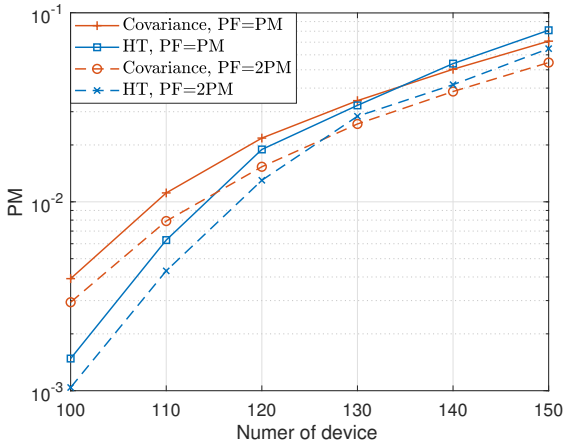


Fig. 6. The performance comparison under different ratios of the active devices to the total devices.

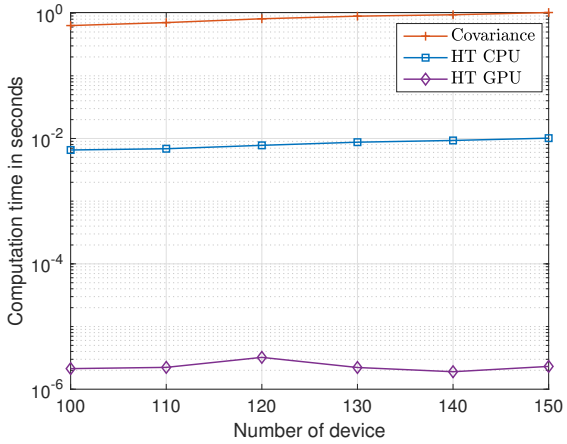
method always outperforms the covariance approach.

C. Generalizability

Next, we demonstrate the generalizability of the proposed method. Unless otherwise specified, the length of pilot sequence is set as $L_{\mathrm{p}}=8$ in the following simulations. We begin by training a heterogeneous transformer, where the device number of the training samples is fixed as $N=100$. However, we test the activity detection performance under different device numbers from 100 to 150. The number of BS-antennas is set as $M=64$ and the maximum transmit power is $p_{\max}=23$ dBm. Due to the trade-off between PM and PF, we provide the PM when PF = PM and PF = 2PM respectively, by appropriately setting the threshold ξ. In the following figures, "Covariance, PF = PM" and "Covariance, PF = 2PM" denote the PM of the covariance approach when PF = PM and PF = 2PM respectively, while "HT, PF = PM" and "HT, PF = 2PM" represent the PM of the proposed method when PF = PM and PF = 2PM respectively. The activity detection performance and average computation time versus number of devices are illustrated in Fig. 7(a) and Fig. 7(b), respectively. We can see from Fig. 7(a) that while the activity detection performances of different approaches become worse as the number of devices N increases, the PM of the proposed method is still comparable with that of the covariance approach when N is increased from 100 to 150. This demonstrates that the proposed method generalizes well to different numbers of devices. On the other hand, Fig. 7(b) shows that as N increases, the average computation times of both the covariance approach and the proposed method on CPU are linearly increased. However, due to the the parallel



(a) PM versus number of devices.

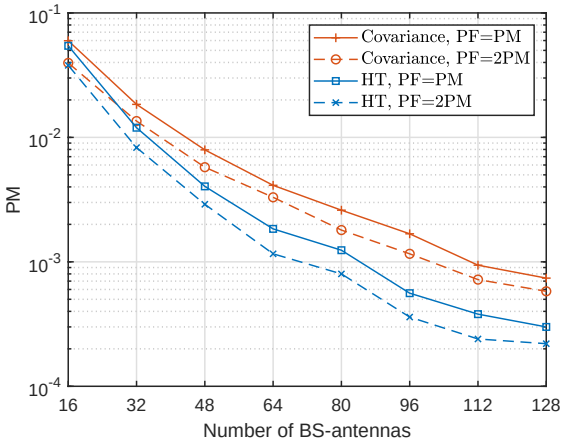


(b) Average computation time versus number of devices.

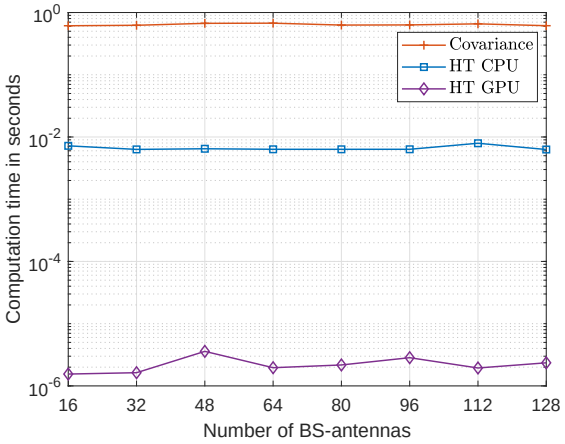
Fig. 7. Generalization to different numbers of devices.

computation of GPU, the average computation time of the proposed method on GPU is nearly a constant, i.e., about 2×10^{-6} second, which is much shorter than that of other approaches.

We further demonstrate the generalizability of the proposed method with respect to different numbers of BS-antennas. To this end, we train a heterogeneous transformer by fixing the number of BS-antennas as $M=32$, and then test its activity detection performance under different numbers of BS-antennas from 16 to 128. The number of devices is $N=100$ and the maximum transmit power is $p_{\max}=23$ dBm. The performance comparisons in terms of PM and average computation time are illustrated in Fig. 8(a) and Fig. 8(b), respectively. Figure 8(a) shows that as M increases from 16 to 128, the proposed method always achieves much lower PM than that of the covariance approach. Although the heterogeneous transformer is trained under $M=32$, when we test the detection performance under $M=128$, the PM of the proposed method is still 2 times lower than that of the covariance approach for both PF = PM and PF = 2PM. This demonstrates that the proposed method generalizes well to larger numbers of BS-antennas. On the other hand, Fig. 8(b) shows that the average computation time of the proposed

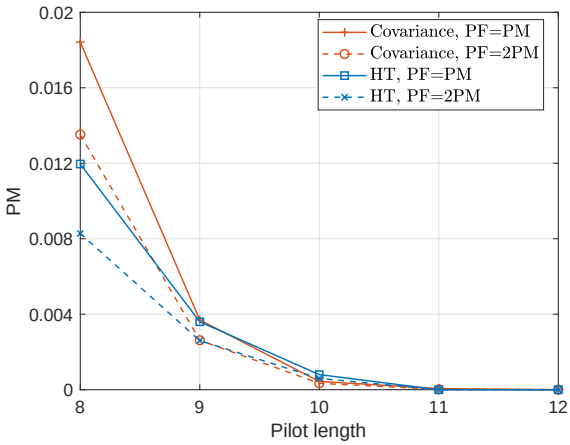


(a) PM versus number of BS-antennas.

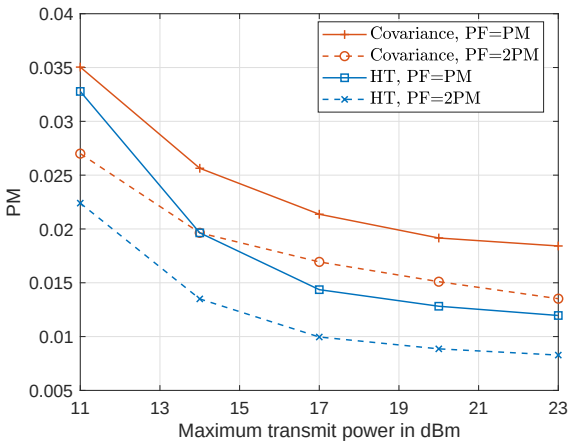


(b) Average computation time versus number of BS-antennas.

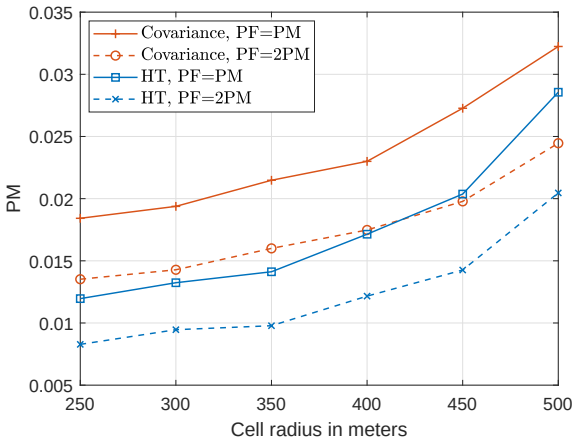
Fig. 8. Generalization to different numbers of BS-antennas.



(a) PM versus pilot length.



(b) PM versus maximum transmit power.



(c) PM versus cell radius.

Fig. 9. Generalization to different pilot lengths, transmit powers, and cell radii.

method on CPU is about 100 times shorter than that of the covariance approach, and the proposed method on GPU even achieves a 10^5 times faster running speed than that of the covariance approach.

Finally, we demonstrate the generalizability with respect to different pilot lengths, transmit powers, and cell radii. In Fig. 9(a), the pilot length of the training samples is fixed as $L_{\mathrm{p}} = 8$, while we test the activity detection performance under different pilot lengths from 8 to 12. The numbers of devices and BS-antennas are $N = 100$ and $M = 32$ respectively, and the maximum transmit power is $p_{\max} = 23$ dBm. To match the input size of the trained heterogeneous transformer, we randomly select 8 time slots of the received signal and device pilots as the input, and obtain the active probability by averaging the outputs over 100 random selections. As shown in Fig. 9(a), heterogeneous transformer generalizes well to different pilot lengths. In particular, when the pilot length increases to 11, the PM decreases to 0.

In Fig. 9(b), the maximum transmit power of the training samples is fixed as $p_{\max} = 23$ dBm, while the activity detection performance is tested under different $p_{\max}$ varying from 11 to 23 dBm. As shown in Fig. 9(b), when the SNR decreases as the transmit power becomes lower, the PM of different approaches becomes higher. However, the proposed method still achieves much lower PM than that of the covariance approach under different SNRs for both PF = PM and PF = 2PM.

In Fig. 9(c), the cell radius of the training samples is fixed as 250 meters, while we test the activity detection performance

under different cell radii varying from 250 meters to 500 meters. The maximum transmit power is fixed as 23 dBm. Consequently, the SNR at the BS varies from 2.87 dB to 14.19 dB as the cell radius increases from 250 meters to 500 meters. It can be seen that the proposed method generalizes well to different cell radii and outperforms the covariance approach.

VI. Conclusions

This paper proposed a deep learning based method with a customized heterogeneous transformer architecture for device activity detection. By adopting an attention mechanism in the neural network architecture design, the proposed heterogeneous transformer was incorporated with desired properties of the activity detection task. Specifically, the proposed architecture is able to extract the relevance among device pilots and received signal, permutation equivariant with respect to devices, and scale adaptable to different numbers of devices. Simulation results showed that the proposed learning based method achieves much better activity detection performance and takes remarkably shorter computation time than state-of-the-art covariance approach. Moreover, the proposed method was demonstrated to generalize well to different numbers of devices and BS-antennas, different pilot lengths, transmit powers, and cell radii.

Appendix A
The expression of $\mathrm{MHA_C}$

Define five sets of parameters $\mathbf{W}_t^{\mathrm{q,c}} \in \mathbb{R}^{d'\times d}$, $\mathbf{W}_{\mathrm{B},t}^{\mathrm{k,c}} \in \mathbb{R}^{d'\times d}$, $\mathbf{W}_{\mathrm{Y},t}^{\mathrm{k,c}} \in \mathbb{R}^{d'\times d}$, $\mathbf{W}_{\mathrm{B},t}^{\mathrm{v,c}} \in \mathbb{R}^{d'\times d}$, and $\mathbf{W}_{\mathrm{Y},t}^{\mathrm{v,c}} \in \mathbb{R}^{d'\times d}$, where d' is the dimension of each attention space and $t \in \{1,\ldots,T\}$. Then, we compute a query $\mathbf{q}_t^{\mathrm{c}}$ for $\mathbf{x}_{N+1}^{[L]}$ at the t-th attention head:

$$\mathbf{q}_t^{\mathrm{c}} = \mathbf{W}_t^{\mathrm{q,c}} \mathbf{x}_{N+1}^{[L]}. \tag{A.1}$$

The key and value corresponding to each $\mathbf{x}_n^{[L]}$ are respectively computed as

$$\mathbf{k}_{n,t}^{\mathrm{c}} = \begin{cases} \mathbf{W}_{\mathrm{B},t}^{\mathrm{k,c}} \mathbf{x}_n^{[L]}, & \forall n = 1,\ldots,N, \\ \mathbf{W}_{\mathrm{Y},t}^{\mathrm{k,c}} \mathbf{x}_{N+1}^{[L]}, & n = N+1, \end{cases} \tag{A.2}$$

$$\mathbf{v}_{n,t}^{\mathrm{c}} = \begin{cases} \mathbf{W}_{\mathrm{B},t}^{\mathrm{v,c}} \mathbf{x}_n^{[L]}, & \forall n = 1,\ldots,N, \\ \mathbf{W}_{\mathrm{Y},t}^{\mathrm{v,c}} \mathbf{x}_{N+1}^{[L]}, & n = N+1. \end{cases} \tag{A.3}$$

To evaluate the relevance between $\mathbf{x}_{N+1}^{[L]}$ and each component of $\left\{\mathbf{x}_n^{[L]}\right\}_{n=1}^{N+1}$, we compute a compatibility $\alpha_{n,t}^{\mathrm{c}}$ using the query $\mathbf{q}_t^{\mathrm{c}}$ and the key $\mathbf{k}_{n,t}^{\mathrm{c}}$:

$$\alpha_{n,t}^{\mathrm{c}} = \frac{\left(\mathbf{q}_t^{\mathrm{c}}\right)^T \mathbf{k}_{n,t}^{\mathrm{c}}}{\sqrt{d'}}, \quad \forall n = 1,\ldots,N+1, \quad \forall t = 1,\ldots,T, \tag{A.4}$$

and the corresponding attention weight is computed by normalizing $\alpha_{n,t}^{\mathrm{c}}$ in $[0,1]$:

$$\beta_{n,t}^{\mathrm{c}} = \frac{e^{\alpha_{n,t}^{\mathrm{c}}}}{\sum_{j=1}^{N+1} e^{\alpha_{j,t}^{\mathrm{c}}}}, \quad \forall n = 1,\ldots,N+1, \quad \forall t = 1,\ldots,T. \tag{A.5}$$

With the attention weight $\beta_{n,t}^{\mathrm{c}}$ scoring the relevance between $\mathbf{x}_{N+1}^{[L]}$ and each component of $\left\{\mathbf{x}_n^{[L]}\right\}_{n=1}^{N+1}$, the attention value of $\mathbf{x}_{N+1}^{[L]}$ at the t-th attention head is computed as

$$\mathbf{x}_t' = \sum_{n=1}^{N+1} \beta_{n,t}^{\mathrm{c}} \mathbf{v}_{n,t}^{\mathrm{c}}. \tag{A.6}$$

The expression of $\mathrm{MHA_C}$ is finally given by a combination of the T attention values:

$$\mathrm{MHA_C}\left(\mathbf{x}_{N+1}^{[L]}, \left\{\mathbf{x}_n^{[L]}\right\}_{n=1}^{N}\right) = \sum_{t=1}^{T} \mathbf{W}_t^{\mathrm{o,c}} \mathbf{x}_t', \tag{A.7}$$

where $\mathbf{W}_t^{\mathrm{o,c}} \in \mathbb{R}^{d\times d'}$ is the parameter for projecting back to a d-dimensional vector.

Appendix B
Proof of Proposition 1

For a device activity detection problem instance, denote the input and output of the heterogeneous transformer as $(\mathbf{Y}, \mathbf{B})$ and $\mathbf{p}$, respectively. The corresponding output of the l-th encoding layer is denoted as $\left\{\mathbf{x}_n^{[l]}\right\}_{n=1}^{N}$. For a permuted problem instance, denote the input and output of the heterogeneous transformer as $\left(\dot{\mathbf{Y}}, \dot{\mathbf{B}}\right) = (\mathbf{Y}, \mathbf{B}\mathbf{\Pi})$ and $\dot{\mathbf{p}}$ respectively, and denote the corresponding output of the l-th encoding layer as $\left\{\dot{\mathbf{x}}_n^{[l]}\right\}_{n=1}^{N}$.

Since $\dot{\mathbf{B}} = \mathbf{B}\mathbf{\Pi}$, there exists a permuted index $\pi(n)$ such that $\dot{\mathbf{b}}_{\pi(n)} = \mathbf{b}_n, \forall n = 1,\ldots,N$. Substituting $\dot{\mathbf{Y}} = \mathbf{Y}$ and $\dot{\mathbf{b}}_{\pi(n)} = \mathbf{b}_n$ into (4)-(6), we have

$$\mathbf{x}_n^{[0]} = \begin{cases} \dot{\mathbf{x}}_{\pi(n)}^{[0]}, & \forall n = 1,\ldots,N, \\ \dot{\mathbf{x}}_{N+1}^{[0]}, & n = N+1. \end{cases} \tag{B.1}$$

Next, we prove that the equation in (B.1) holds for l as long as it holds for $l-1$. Substituting $\mathbf{x}_n^{[l-1]} = \dot{\mathbf{x}}_{\pi(n)}^{[l-1]}$, $\forall n = 1,\ldots,N$ and $\mathbf{x}_{N+1}^{[l-1]} = \dot{\mathbf{x}}_{N+1}^{[l-1]}$ into (7) and (8), we obtain

$$\mathbf{x}_n^{[l]} = \begin{cases} \dot{\mathbf{x}}_{\pi(n)}^{[l]}, & \forall n = 1,\ldots,N, \\ \dot{\mathbf{x}}_{N+1}^{[l]}, & n = N+1. \end{cases} \tag{B.2}$$

Combining (B.1) and (B.2), we can conclude that (B.2) holds $\forall l = 0,\ldots,L$. Substituting (B.2) for $l = L$ into (23)-(25), we obtain $P_n = \dot{P}_{\pi(n)}, \forall n = 1,\ldots,N$, which implies $\dot{\mathbf{p}} = \mathbf{\Pi}^T \mathbf{p}$. Together with $\dot{\mathbf{p}} = f(\dot{\mathbf{Y}}, \dot{\mathbf{B}}) = f(\mathbf{Y}, \mathbf{B}\mathbf{\Pi})$, we have $\mathbf{\Pi}^T \mathbf{p} = f(\mathbf{Y}, \mathbf{B}\mathbf{\Pi})$.

References

[1] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, P. Popovski, and A. Dekorsy, "Massive machine-type communications in 5G: Physical and MAC-layer solutions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 59–65, Sep. 2016.

[2] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, "Toward massive machine type cellular communications," *IEEE Wireless Commun. Mag.*, vol. 24, no. 1, pp. 120–128, Feb. 2017.

[3] L. Liu, E. G. Larsson, W. Yu, P. Popovski, Č. Stefanović, and E. de Carvalho, "Sparse signal processing for grant-free massive connectivity: A future paradigm for random access protocols in the internet of things," *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 88–99, Sep. 2018.

[4] J. Wang, Z. Zhang, and L. Hanzo, "Joint active user detection and channel estimation in massive access systems exploiting reedcmuller sequences," *IEEE J. Sel. Top. Signal Process.*, vol. 13, no. 3, pp. 739–752, Jun. 2019.

[5] L. Liu and W. Yu, "Massive connectivity with massive MIMO-Part I: Device activity detection and channel estimation," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2933–2946, Jun. 2018.

[6] ——, "Massive connectivity with massive MIMO–Part II: Achievable rate characterization," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2947–2959, Jun. 2018.

[7] Z. Chen, F. Sohrabi, and W. Yu, "Sparse activity detection for massive connectivity," *IEEE Trans. Signal Process.*, vol. 66, no. 7, pp. 1890–1904, Apr. 2018.

[8] Z. Sun, Z. Wei, L. Yang, J. Yuan, X. Cheng, and L. Wan, "Exploiting transmission control for joint user identification and channel estimation in massive connectivity," *IEEE Trans. Commun.*, vol. 67, no. 9, pp. 6311–6326, Sep. 2019.

[9] M. Ke, Z. Gao, Y. Wu, X. Gao, and R. Schober, "Compressive sensing-based adaptive active user detection and channel estimation: Massive access meets massive MIMO," *IEEE Trans. Signal Process.*, vol. 68, pp. 764–779, 2020.

[10] K. Senel and E. G. Larsson, "Grant-free massive MTC-enabled massive MIMO: A compressive sensing approach," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6164–6175, Dec. 2018.

[11] S. Jiang, X. Yuan, X. Wang, C. Xu, and W. Yu, "Joint user identification, channel estimation, and signal detection for grant-free NOMA," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6960–6976, Oct. 2020.

[12] Y. Mei, Z. Gao, Y. Wu, W. Chen, J. Zhang, D. W. K. Ng, and M. Di Renzo, "Compressive sensing based joint activity and data detection for grant-free massive IoT access," *IEEE Trans. Wireless Commun.*, to appear 2022, doi:10.1109/TWC.2021.3107576.

[13] Z. Utkovski, O. Simeone, T. Dimitrova, and P. Popovski, "Random access in C-RAN for user activity detection with limited-capacity fronthaul," *IEEE Signal Process. Lett.*, vol. 24, no. 1, pp. 17–21, Jan. 2017.

[14] Z. Chen, F. Sohrabi, and W. Yu, "Multi-cell sparse activity detection for massive random access: Massive MIMO versus cooperative MIMO," *IEEE Trans. Wireless Commun.*, vol. 18, no. 8, pp. 4060–4074, Aug. 2019.

[15] M. Ke, Z. Gao, Y. Wu, X. Gao, and K.-K. Wong, "Massive access in cell-free massive MIMO-based internet of things: Cloud computing and edge computing paradigms," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 3, pp. 756–772, Mar. 2021.

[16] X. Xu, X. Rao, and V. K. Lau, "Active user detection and channel estimation in uplink CRAN systems," in *IEEE ICC*, 2015.

[17] J. Ahn, B. Shim, and K. B. Lee, "EP-based joint active user detection and channel estimation for massive machine-type communications," *IEEE Trans. Commun.*, vol. 67, no. 7, pp. 5178–5189, Jul. 2019.

[18] W. Chen, H. Xiao, L. Sun, and B. Ai, "Joint activity detection and channel estimation in massive MIMO systems with angular domain enhancement," *IEEE Trans. Wireless Commun.*, to appear 2022, doi:10.1109/TWC.2021.3117358.

[19] X. Liu, Y. Shi, J. Zhang, and K. B. Letaief, "Massive CSI acquisition for dense cloud-RANs with spatial-temporal dynamics," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2557–2570, Apr. 2018.

[20] Q. He, T. Q. S. Quek, Z. Chen, Q. Zhang, and S. Li, "Compressive channel estimation and multi-user detection in C-RAN with low-complexity methods," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 3931–3944, Jun. 2018.

[21] Y. Li, M. Xia, and Y.-C. Wu, "Activity detection for massive connectivity under frequency offsets via first-order algorithms," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1988–2002, Mar. 2019.

[22] X. Shao, X. Chen, and R. Jia, "A dimension reduction-based joint activity detection and channel estimation algorithm for massive access," *IEEE Trans. Signal Process.*, vol. 68, no. 1, pp. 420–435, Jan. 2020.

[23] S. Haghighatshoar, P. Jung, and G. Caire, "Improved scaling law for activity detection in massive MIMO systems," in *IEEE ISIT*, 2018.

[24] X. Shao, X. Chen, D. W. K. Ng, C. Zhong, and Z. Zhang, "Cooperative activity detection: Sourced and unsourced massive random access paradigms," *IEEE Trans. Signal Process.*, vol. 68, pp. 6578–6593, 2020.

[25] Z. Chen, F. Sohrabi, and W. Yu, "Sparse activity detection in multi-cell massive MIMO exploiting channel large-scale fading," *IEEE Trans. Signal Process.*, vol. 69, pp. 3768–3781, 2021.

[26] U. K. Ganesan, E. Björnson, and E. G. Larsson, "Clustering based activity detection algorithms for grant-free random access in cell-free massive MIMO," *IEEE Trans. Commun.*, vol. 69, no. 11, pp. 7520–7530, Nov. 2021.

[27] D. Jiang and Y. Cui, "ML estimation and MAP device activity detections for grant-free massive access in multi-cell networks," *IEEE Trans. Wireless Commun.*, to appear 2022, doi: 10.1109/TWC.2021.3125199.

[28] ——, "MAP-based pilot state detection in grant-free random access for mMTC," in *IEEE SPAWC*, 2020.

[29] Z. Wang, Y.-F. Liu, and L. Liu, "Covariance-based joint device activity and delay detection in asynchronous mMTC," *IEEE Signal Process. Lett.*, vol. 29, pp. 538–542, Jan. 2022.

[30] Q. Lin, Y. Li, and Y.-C. Wu, "Sparsity constrained joint activity and data detection for massive access: A difference-of-norms penalty framework," *IEEE Trans. Wireless Commun.*, to appear 2022, doi:10.1109/TWC.2022.3204786.

[31] Y. Li, Q. Lin, Y.-F. Liu, B. Ai, and Y.-C. Wu, "Asynchronous activity detection for cell-free massive MIMO: From centralized to distributed algorithms," *IEEE Trans. Wireless Commun.*, to appear 2022, doi:10.1109/TWC.2022.3211967.

[32] A. Fengler, S. Haghighatshoar, P. Jung, and G. Caire, "Non-Bayesian activity detection, large-scale fading coefficient estimation, and unsourced random access with a massive MIMO receiver," *IEEE Trans. Inf. Theory*, vol. 67, no. 5, pp. 2925–2951, May 2021.

[33] Z. Chen, F. Sohrabi, Y.-F. Liu, and W. Yu, "Phase transition analysis for covariance based massive random access with massive MIMO," *IEEE Trans. Inf. Theory*, to appear 2022, doi: 10.1109/TIT.2021.3132397.

[34] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, Oct. 2018.

[35] M. Zhu, T. Chang, and M. Hong, "Learning to beamform in heterogeneous massive MIMO networks," 2020. [Online]. Available: https://arxiv.org/abs/2011.03971.

[36] F. Sohrabi, K. M. Attiah, and W. Yu, "Deep learning for distributed channel feedback and multiuser precoding in FDD massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4044–4057, Jul. 2021.

[37] T. Jiang, H. V. Cheng, and W. Yu, "Learning to reflect and to beamform for intelligent reflecting surface with implicit channel estimation," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 1931–1945, Jul. 2021.

[38] Y. Li, Z. Chen, G. Liu, Y.-C. Wu, and K.-K. Wong, "Learning to construct nested polar codes: An attention-based set-to-element model," *IEEE Commun. Lett.*, vol. 25, no. 12, pp. 3898–3902, Dec. 2021.

[39] Y. Cui, S. Li, and W. Zhang, "Jointly sparse signal recovery and support recovery via deep learning with applications in MIMO-based grant-free random access," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 3, pp. 788–803, Mar. 2021.

[40] Y. Qiang, X. Shao, and X. Chen, "A model-driven deep learning algorithm for joint activity detection and channel estimation," *IEEE Commun. Lett.*, vol. 24, no. 11, pp. 2508–2512, Nov. 2020.

[41] X. Shao, X. Chen, Y. Qiang, C. Zhong, and Z. Zhang, "Feature-aided adaptive-tuning deep learning for massive device detection," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 1899–1914, Jul. 2021.

[42] J. Huang, H. Zhang, C. Huang, L. Yang, and W. Zhang, "Noncoherent massive random access for inhomogeneous networks: From message passing to deep learning," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 5, pp. 1457–1472, May 2022.

[43] Y. Shi, H. Choi, Y. Shi, and Y. Zhou, "Algorithm unrolling for massive access via deep neural network with theoretical guarantee," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 945–959, Feb. 2022.

[44] Z. Mao, X. Liu, M. Peng, Z. Chen, and G. Wei, "Joint channel estimation and active-user detection for massive access in internet of things-a deep learning approach," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2870–2881, Feb. 2022.

[45] M. Eisen and A. Ribeiro, "Optimal wireless resource allocation with random edge graph neural networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 2977–2991, 2020.

[46] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "Graph neural networks for scalable radio resource management: Architecture design and theoretical analysis," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 101–115, Jan. 2021.

[47] J. Guo and C. Yang, "Learning power allocation for multi-cell-multi-user systems with heterogeneous graph neural network," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 884–897, Feb. 2022.

[48] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.
[49] Y. Shen, J. Zhang, S. Song, and K. B. Letaief, "AI empowered resource management for future wireless networks," in *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, 2021.
[50] M. Mossberg, E. K. Larsson, and E. Mossberg, "Estimation of large-scale fading channels from sample covariances," in *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006.
[51] P.-T. de Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of Operations Research*, vol. 134, no. 1, pp. 19–67, Jan. 2005.
[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE CVPR*, 2016.
[53] J. Ba, J. Kiros, and G. E. Hinton, "Layer normalization," 2016. [Online]. Available: https://arxiv.org/abs/1607.06450.
[54] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015.
[55] Z. Ye, J. Zhou, Q. Guo, Q. Gan, and Z. Zhang. Transformer as a graph neural network. [Online]. Available: https://docs.dgl.ai/tutorials/models/4_old_wines/7_transformer.html
[56] K. You, M. Long, J. Wang, and M. I. Jordan, "How does learning rate decay help modern neural networks?" 2019. [Online]. Available: https://arxiv.org/abs/1908.01878.

Yang Li (Member, IEEE) received the B.E and M.E degrees in electronics engineering from Beihang University (BUAA), Beijing, China, in 2012 and 2015, respectively, and the Ph.D. degree from the Department of Electrical and Electronic Engineering, The University of Hong Kong (HKU) in 2019. From 2019 to 2020, he has been a Senior Research Engineer with Huawei Noah's Ark Laboratory. He is currently a Research Scientist with Shenzhen Research Institute of Big Data. His research interests include radio resource management, learning to optimize, and large-scale optimization. He is the winner of the 2020 Innovation Pioneer Award of Huawei.

Zhilin Chen (S'14) received the B.E. degree in electrical and information engineering and the M.E. degree in signal and information processing from Beihang University (BUAA), Beijing, China, in 2012 and 2015, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2020. He is now with Huawei Technologies, Shenzhen, China. His main research interests include wireless communication, signal processing, and machine learning.

Yunqi Wang received the B.S. degree in Electrical Engineering from the University of Electronic Science and Technology of China, the M.S. degree in Electrical and Computer Engineering from Rutgers University, and Ph.D. degree in Electrical and Electronic Engineering, University of Hong Kong. He is currently a postdoc fellow in Chinese University of Hong Kong. His research interests include deep learning, causal inference, and applications in computer vision and neuroimaging.

Chenyang Yang received her Ph.D. degree in Electrical Engineering from Beihang University, China, in 1997. She has been a full professor with Beihang University since 1999. She has published over 200 papers in the fields of machine learning for wireless communications, URLLC, energy efficient resource allocation, wireless caching, interference management, etc. She was supported by the 1st Teaching and Research Award Program for Outstanding Young Teachers of Higher Education Institutions by Ministry of Education of China. She has served as an associate or guest editor for several IEEE journals. Her recent research interests lie in mobile/wireless AI, and URLLC.

Bo Ai (IEEE Fellow, IET Fellow) is the professor and doctoral supervisor of Beijing Jiaotong University. He is also the deputy director of the State Key Laboratory of Rail Traffic Control and Safety.

Prof. Bo Ai has published 6 Chinese academic books, 3 English books, over 150 IEEE journal articles. He has obtained 13 international paper awards include IEEE VTS Neil Shepherd Memorial Best Propagation Award and IEEE GLOBECOM 2018 Best Paper Award, 36 invention patents, 28 proposals adopted by the ITU, 3GPP, etc., and 9 provincial and ministerial-level science and technology awards. His research results have been involved in 6 national standards. He is mainly engaged in the research and application of the theory and core technology of broadband mobile communication and rail transit dedicated mobile communication systems (GSM-R, LTE-R, 5G-R, LTE-M).

Prof. Bo Ai is the Fellow of Chinese Institute of Electronics, Fellow of China Institute of Communications, Chair of IEEE BTS Xi'an Branch, Vice Chair of IEEE VTS Beijing Branch, IEEE VTS distinguished lecturer, an expert of the 5G Industry Expert Group of the China Mobile Group Technical Advisory Committee, and an expert of the 6G Group in China.

Yik-Chung Wu received the B.Eng. (EEE) degree in 1998 and the M.Phil. degree in 2001 from the University of Hong Kong (HKU). He received the Croucher Foundation scholarship in 2002 to study Ph.D. degree at Texas A&M University, College Station, and graduated in 2005. From August 2005 to August 2006, he was with the Thomson Corporate Research, Princeton, NJ, as a Member of Technical Staff. Since September 2006, he has been with HKU, currently as an Associate Professor. He was a visiting scholar at Princeton University, in summers of 2015 and 2017. His research interests are in general areas of signal processing, machine learning and communication systems. Dr. Wu served as an Editor for IEEE COMMUNICATIONS LETTERS and IEEE TRANSACTIONS ON COMMUNICATIONS. He is currently an editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE WIRELESS COMMUNICATIONS LETTERS, and JOURNAL OF COMMUNICATIONS AND NETWORKS.