Wireless Coded Computation with Error Detection

Borui Fang, Li Chen, *Senior Member, IEEE*, Yunfei Chen, *Senior Member, IEEE*, Changsheng You, *Member, IEEE*, Xiaohui Chen, *Member, IEEE*, and Weidong Wang

***Abstract*—In wireless networks with distributed computing, the computational performance is limited by stragglers. To mitigate the stragglers' effect, coded computation is adopted through computational redundancy. Moreover, in wireless transmission, transmission errors may occur due to noise, channel fading and so on. Existing works design coded computation and error detection separately. However, this leads to frequent encoding and inefficient allocation. In this paper, we propose a joint computation and transmission coding (JCTC) scheme to design coded computation and error detection jointly. The coded computation is based on Luby transform (LT) code and linear error-detecting codes are applied for the re-transmission mechanism. To achieve the low dynamic encoding, two-layer encoding is adopted. Then, the performances of JCTC scheme are analyzed in terms of latency and computation reliability. Finally, in order to achieve efficient task and redundancy allocation, the wireless LT coded computation with error detection (WLTCC-ED) algorithm is given from both iterative and low-complexity perspectives respectively. Through theoretical analysis and numerical simulation, it shows that our proposed JCTC scheme has significant advantages over separate designs.**

***Index Terms*—Coded computation, distributed computing, error detection, transmission errors, wireless networks.**

I. Introduction

With the explosive number of mobile and Internet of Things (IoT) devices, distributed computing has aroused great interest to perform large-scale computational tasks. In distributed computing [1], [2], mobile and IoT devices are connected to solve computational tasks, such as mobile edge computation [3], [4] and federated edge learning [5]. Despite the advantage of efficient computation for distributed computing, its performance is limited by stragglers, which slow down the execution time for the whole distributed networks.

To address the stragglers' effect, a new framework named coded computation [6] was proposed. Inspired by classical coding theory, the authors in [6] applied maximum distance separable (MDS) code to speed up the distributed matrix multiplications by introducing necessary computational redundancy in the homogeneous networks with nodes of uniform computation capabilities. Without waiting for the responses from all the nodes, the desired computational results could be recovered only using some fast-responding nodes. It implied that MDS coding scheme could reduce the computation time significantly and achieve an order-wise improvement over the original uncoded scheme. The authors in [7], [8] further studied the corresponding scheme to allocate the optimal computational task for nodes in the heterogeneous networks with disparate computation capabilities.

Compared to MDS code with a fixed rate, Luby transform (LT) code offers the rateless property and lower decoding complexity. Using the rateless property, the corresponding LT coding scheme was proposed in [9]. Through sub-block division, this scheme could exploit the computed results from all the nodes including stragglers. It led to negligible redundant computation and maximum straggler tolerance for a lower latency. Moreover, the authors in [10] showed the LT coding scheme could further reduce the computation latency at the expense of an increased communication load.

In wireless distributed networks, transmission latency also has an important effect on the performance. For homogeneous wireless networks, the authors in [11] analyzed the performance of MDS coding scheme from the total latency's point of view. With packet losses due to channel fading, the work of [12] investigated the performance of total latency and provided guidelines to design optimal MDS code. For heterogeneous wireless networks, the authors in [13] proposed wireless coded computation scheme to deal with both computation and transmission stragglers. Then, the authors in [14] further exploited the computed results of stragglers in wireless networks based on block-division. As for LT coding scheme, the work of [15] proposed block-design based wireless LT coded computation scheme to balance both computation and transmission latency.

Using coded computation discussed above, various complicated computational tasks and distributed computing scenarios have been studied in [16]–[29]. In order to keep the master's data private and secure from workers, the works of [16]–[18] studied the private and secure distributed matrix multiplication. To speed up more complex distributed computational tasks using codes, the authors in [19], [20] discussed the convolution and the regression problem respectively, and the distributed computing problem of arbitrary functions was studied in [21]. For the scenario where the exact computational result was not required, the authors in [22]–[24] provided a strategy of approximating coded distributed computing to realize a tradeoff between accuracy and speed. As for a more practical distributed network setup, the heterogeneous multi-hop network was considered in [25] and the work of [26] studied multiple distributed matrix multiplication tasks in a multi-master heterogeneous-worker scenario. The deployment of

This work was supported by the National Key Research and Development Program of China under Grant 2021YFB2900302. This paper was presented in part at the 2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall), Hong Kong, China, 10-13 October 2023. *(Corresponding author: Li Chen.)*

Borui Fang, Li Chen, Xiaohui Chen and Weidong Wang are with Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, Anhui 230027 (e-mail: fangbr@mail.ustc.edu.cn; chenli87@ustc.edu.cn; cxh@ustc.edu.cn; wdwang@ustc.edu.cn).

Yunfei Chen is with the Department of Engineering, University of Durham, DH1 3LE Durham, U.K. (e-mail: Yunfei.Chen@durham.ac.uk).

Changsheng You is with the Department of Electronic and Electrical Engineering, Southern University of Science and Technology (SUSTech), Shenzhen 518055, China (e-mail: youcs@sustech.edu.cn).

coded computation in wireless edge computing was discussed in [27]–[29].

Most existing works on coded computation disregarded the transmission errors in wireless networks, which may lead to a severe performance degradation. For example, although the works of [11], [13], [15] considered the transmission latency in wireless distributed networks, they ignored the negative effects caused by wireless channel. The works of [12], [14] only analyzed the performance of wireless coded computation with packet losses and did not discuss data errors, which were more complex and practical. Also, they did not give a design of wireless coded computation with high reliability. As for the accuracy-sensitive computational tasks, the results with low reliability are intolerable. The error-detecting mechanism based on re-transmission [30] was applied to address this. However, coded computation and error detection were designed separately. Such separate design has the following problems.

1) **Frequent Encoding.** Each time the input data changes, the corresponding computed result has to be encoded again before transmission, which leads to excessive encoding tasks and huge encoding latency. For the delay-sensitive computational tasks, the high latency causes a severe performance degradation and it is intolerable. So such scheme is not suitable for high dynamic scenarios.

2) **Inefficient Allocation.** The error-detecting redundancy is designed locally without the global network parameters, while the computational tasks are allocated in the fusion center without considering the re-transmission overhead and reliability. For example, if a worker with good computation and transmission capabilities is in a very bad channel condition, this worker will be still allocated lots of computational tasks in the existing separate designs. It may leads to a huge re-transmission latency or an unsatisfied reliability. Thus, this allocation is not efficient or optimal.

To address the above issues, we propose to design coded computation and error detection jointly. Specifically, we first give the new joint computation and transmission coding (JCTC) scheme. The coded computation is based on LT code, while error-detecting mechanism is based on re-transmission using linear codes. Then, its performances are analyzed for latency and computation reliability. Finally, within the required computation reliability, the sub-optimal efficient task and redundancy allocation strategies based on iterative optimization algorithm and low-complexity algorithm are obtained respectively. The main contributions of this paper are summarized as follows:

- **Joint Coding Design.** The two-layer encoding is performed at the fusion center so that both computation coding and transmission coding can be done offline. Also, the low dynamic encoding can be achieved no matter how frequently the input data changes.
- **Performance Benefits.** Compared with the separate designs with the same computation reliability, our JCTC scheme can achieve less encoding and lower computation latency. Besides, the performance of latency and computation reliability can be balanced well in the proposed scheme.
- **Efficient Task and Redundancy Allocation.** Within the required computation reliability, sub-optimal efficient task and redundancy allocation can be obtained at the fusion center by wireless LT coded computation with error detection (WLTCC-ED) algorithm based on iterative optimization to realize a tradeoff between computation and transmission latency. As for a scenario of low error rate, an approximate algorithm is proposed to simplify the solving process with a lower complexity.

Organization: The rest of this paper is organized as follows. In Section II, the wireless LT coded computation is reviewed and the drawbacks of the existing separate designs are discussed. The proposed JCTC scheme is presented in Section III. Then, the performances of latency and computation reliability for the JCTC scheme are analyzed in Section IV. In Section V, the sub-optimal task and redundancy allocation strategy is obtained through iterative optimization, and a low-complexity algorithm is given for the scenario of low error rate. Simulation results are shown in Section VI and conclusion is finally presented in Section VII.

Notation: The set $\{1, 2, \ldots, n\}$ is denoted as $[n]$ for $n \in \mathbb{N}$. We denote $f(n) = \mathcal{O}(g(n))$ if there exist constants $v > 0$ and $n_0 \in \mathbb{N}$ such that $f(n) \leq v \cdot g(n)$ for $\forall n > n_0$; and $f(n) = \Theta(g(n))$ if $f(n) = \mathcal{O}(g(n))$ and $g(n) = \mathcal{O}(f(n))$. The indicator function is denoted as $\mathbb{1}_{\{\cdot\}}$. For any $\lambda \in \mathbb{R}$, if $\lambda > 0$, $\mathbb{1}_{\{\lambda\}} = 1$; otherwise, $\mathbb{1}_{\{\lambda\}} = 0$. As for a function $f_i(t_i, b_i, r_i)$ with respect to variables t_i, b_i and r_i, we denote $f_{i|[0]} = f_i(t_0, b_0, r_0)$, $f_{i|[t_0]} = \partial f_i/\partial t_i|_{t_i=t_0}$, $f_{i|[b_0]} = \partial f_i/\partial b_i|_{b_i=b_0}$ and $f_{i|[r_0]} = \partial f_i/\partial r_i|_{r_i=r_0}$ with the given point t_0, b_0 and r_0.

II. System Model

We consider a classical distributed master-worker setup [6], [7] in a wireless network, as shown in Fig. 1. The whole network consists of one master and n workers that have different computation and transmission capabilities. The goal is to compute a matrix-vector multiplication $\mathbf{y} = \mathbf{A}\mathbf{x}$ wirelessly and reliably at the master with the help of the workers, where $\mathbf{A} \in \mathbb{F}_{2^q}^{m\times d}$ is a pre-stored matrix in this distributed network, $\mathbf{x} \in \mathbb{F}_{2^q}^{d}$ is an input vector that is broadcast to each worker by the master, and $\mathbf{y} \in \mathbb{F}_{2^q}^{m}$ is the output vector.

To speed up the computational tasks in heterogeneous wireless networks, the rateless LT coded computation [9] is applied. In LT coding approach, the master first generates the encoded matrix $\tilde{\mathbf{A}} \in \mathbb{F}_{2^q}^{\alpha m\times d}$ ($\alpha > 1$, and α can be very large to achieve the rateless property) by treating the m rows of $\mathbf{A}$ as source symbols according to the robust soliton degree distribution [31]. Dividing $\tilde{\mathbf{A}}$ equally by rows, the data block $\tilde{\mathbf{A}}_i \in \mathbb{F}_{2^q}^{l\times d}$ ($l = \alpha m/n$) will be assigned to worker i, $i \in [n]$. Then, in order to further utilize the rateless property of LT code and the partial works done by stragglers, the rows of $\tilde{\mathbf{A}}_i$ are divided again by the master into sub-blocks of the same size as $\{\tilde{\mathbf{A}}_{i,j} \in \mathbb{F}_{2^q}^{b_i\times d}\}_{j=1}^{\lceil l/b_i\rceil}$ and each data sub-block will be stored in the corresponding workers, where b_i denotes the data sub-block size for worker i, i.e., each data sub-block includes b_i inner products to be calculated. For the traditional LT coding approach, the size of data sub-blocks cannot be too large and a fine-grained dividing strategy is usually adopted, i.e., $b_i = 1, i \in [n]$.

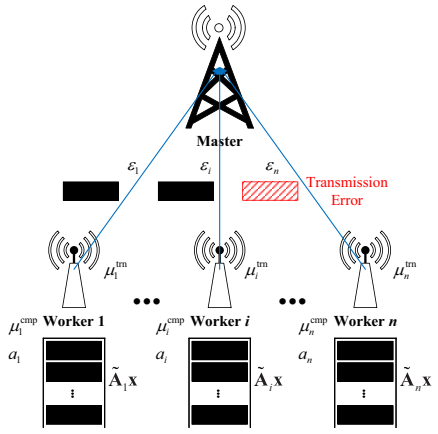
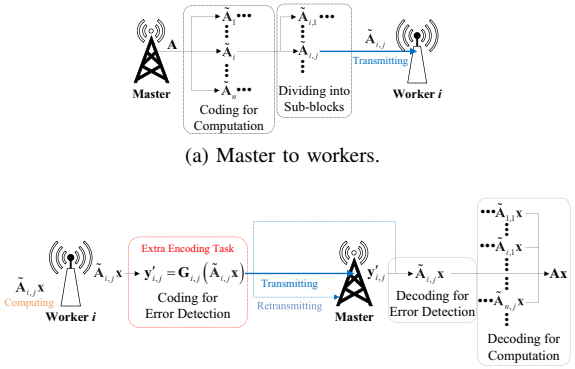


Fig. 1. Distributed coded computation with transmission error in wireless networks. Each square represents a sub-block, which is transmitted through a binary symmetric channel with a bit error transition probability $\{\varepsilon_i\}_{i=1}^{n}$. For worker i, the computation capability is evaluated by $\left(\mu_i^{\mathrm{cmp}}, a_i\right)$ and the transmission capability is measured by μ_i^{trn}.



(a) Master to workers.

(b) Workers to master.

Fig. 2. The workflow of separate designs. Coding for computation is performed by the master, while coding for error detection is done by each worker. It causes that the whole computational tasks for worker i contain the original computational tasks $\{\tilde{\mathbf{A}}_{i,j}\mathbf{x}\}_{j=1}^{\lceil l/b_i \rceil}$ and the extra encoding tasks $\{\mathbf{G}_{i,j}(\tilde{\mathbf{A}}_{i,j}\mathbf{x})\}_{j=1}^{\lceil l/b_i \rceil}$.

After receiving the input $\mathbf{x}$, worker i starts to compute $\{\tilde{\mathbf{A}}_{i,j}\mathbf{x}\}_{j=1}^{\lceil l/b_i \rceil}$. When a partial result $\tilde{\mathbf{A}}_{i,j}\mathbf{x}$ is done, worker i can transmit it to the master as soon as possible instead of waiting for the complete result $\tilde{\mathbf{A}}_i\mathbf{x}$. The worker will transmit its data sub-block early if it finishes the computation early and only one worker can transmit its result at each time. Due to the severe channel fading, noise and so on, different data errors may occur during the transmission. We model these transmission errors as a binary symmetric channel with a fixed bit error probability ε_i for worker i and the channel transition probability matrix $\mathbf{H}_i$ of worker i is given as

$$\mathbf{H}_i = \begin{bmatrix} 1-\varepsilon_i & \varepsilon_i \\ \varepsilon_i & 1-\varepsilon_i \end{bmatrix}, \tag{1}$$

where ε_i can be obtained on basis of the number of error symbols by transmitting the reference signal or other channel estimation techniques [32], [33].

As for an inner product transmitted by worker i, assume that each bit error occurs independently and there is an error in the inner product if at least one bit is erroneous [30]. Then, the error probability for the inner product can be obtained by $\varepsilon_{q,i} = 1-(1-\varepsilon_i)^q$, where each inner product is represented by q bits. In order to avoid these transmission errors, the re-transmission mechanism [30] is considered. Through introducing the error-detecting redundancy, some transmission errors can be detected and the re-transmission is required for the corresponding sub-block to ensure the reliability of transmission. Furthermore, because of the limited bandwidth of wireless channel, a uniform maximum number of sub-blocks that can be transmitted successfully from each worker is pre-allocated to avoid the frequent interaction between workers and master, which is denoted as a constant k. In other words, each worker can send up to k sub-blocks to the master to prevent excessive occupation of channel resources.

Once receiving a sub-block, the master detects whether there are any transmission errors. If the transmission errors are detected, the corresponding sub-block will be re-transmitted; otherwise, the master will accept this sub-block and decode it. According to the decoding features of LT code, the master can recover the desired computational result $\mathbf{y}$ successfully once any $(1+\eta)\,m$ accepted data inner products are received from all the workers, where η is a small decoding overhead ($\eta \to 0$ as $m \to \infty$).

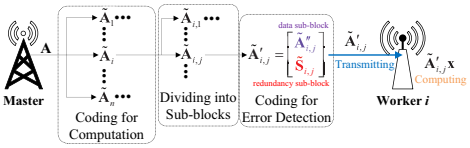
From the above discussion, the existing schemes design coded computation and error detection separately, as illustrated in Fig. 2. This has the following drawbacks.

1) *Frequent and high dynamic encoding.* The pre-stored model matrix $\mathbf{A}$ has the characteristic of low dynamic, while the input vector $\mathbf{x}$ is highly dynamic in many machine learning and big data applications [34]. Each time the input data changes, the corresponding computed result has to be encoded again by each worker before transmission, which causes burdensome encoding tasks and a huge encoding latency. Such a severe performance degradation is intolerable for the delay-sensitive computational tasks.

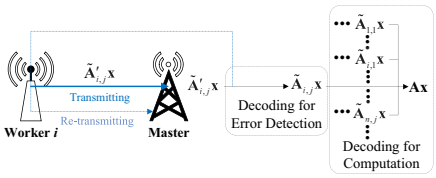
2) *Inefficient task and redundancy allocation.* The computational task is allocated in the fusion center without considering the re-transmission latency and reliability, while the error-detecting redundancy is designed by each worker locally without the global network parameters at the fusion center. In other words, the data sub-block size $b_i, i \in [n]$ is designed by the master but the error-detecting redundancy r_i is decided by worker i. The design of the whole sub-block size is fragmented and inefficient. For example, if a worker with good computation and transmission capabilities is in a very bad channel condition, this worker will be still allocated a large sub-block size in the existing separate designs, which leads to a huge re-transmission latency or an unsatisfied reliability. So this allocation strategy is not optimal.

III. Joint Computation and Transmission Coding Design

In order to overcome the drawbacks of separate designs, the JCTC scheme is proposed. It performs both coded computation and error-detecting coding in the fusion center, as shown in Fig. 3, to achieve low dynamic encoding and efficient allocation. The specific process can be described as follows.



(a) Master to workers.



(b) Workers to master.

Fig. 3. The workflow of JCTC scheme. Both coding for computation and error detection are performed by the master. Each worker calculates matrix multiplications $\{\tilde{\mathbf{A}}'_{i,j}\mathbf{x}\}_{j=1}^{\lceil l/b_i\rceil}$ including data matrix multiplications $\{\tilde{\mathbf{A}}''_{i,j}\mathbf{x}\}_{j=1}^{\lceil l/b_i\rceil}$ and redundancy matrix multiplications $\{\tilde{\mathbf{S}}_{i,j}\mathbf{x}\}_{j=1}^{\lceil l/b_i\rceil}$.

1) *Two-Layer Encoding.* As shown in Fig. 3a, the master first encodes $\mathbf{A}$ for computation to speed up matrix multiplication. After that, the encoded and divided sub-block $\tilde{\mathbf{A}}_{i,j}$ is encoded again for error detection to ensure the reliability of inner products during transmission. In this paper, the linear error-detecting code is applied. The corresponding data sub-block after error-detecting encoding is denoted as $\{\tilde{\mathbf{A}}''_{i,j} \in \mathbb{F}_{2^q}^{b_i\times d}\}_{j=1}^{\lceil l/b_i\rceil}$, and $\{\tilde{\mathbf{S}}_{i,j} \in \mathbb{F}_{2^q}^{r_i\times d}\}_{j=1}^{\lceil l/b_i\rceil}$ represents the additional redundancy for error detection, where r_i is the size of redundancy in each sub-block for worker i. Both $\tilde{\mathbf{A}}''_{i,j}$ and $\tilde{\mathbf{S}}_{i,j}$ together constitute the two-layer encoded sub-matrix $\{\tilde{\mathbf{A}}'_{i,j} \in \mathbb{F}_{2^q}^{(b_i+r_i)\times d}\}_{j=1}^{\lceil l/b_i\rceil}$. Then, the master will send $\tilde{\mathbf{A}}'_{i,j}$ to worker $i \in [n]$.

2) *Distributed Computing and Serial Transmitting.* After receiving $\tilde{\mathbf{A}}'_{i,j}$, the worker i computes matrix multiplication $\tilde{\mathbf{A}}'_{i,j}\mathbf{x}$ and then sends the corresponding computed results back to the master. The total computation time for worker i is denoted as a random variable T_i^{cmp}.

3) *Error Detection and Re-transmission.* The transmission from workers to the master may incur transmission errors. Once receiving the transmitted sub-blocks by workers, the master will perform error detection. For a received sub-block, if it contains no error, the master will accept it directly; if it contains a detectable error pattern, the corresponding sub-block will be re-transmitted; if it contains an undetected error pattern, the master will also accept it with the undetected transmission error, which means the master commits a decoding error and decreases the reliability of the whole networks. For worker i, the total time spent on transmitting computed sub-blocks until accepted by the master is denoted as $T_i^{\mathrm{trn}}(t_{\mathrm{c}})$, where t_{c} is the given computation time. For the whole networks, the number of undetected error data inner products is represented as N_{un}.

4) *Recovering Desired Result.* After receiving enough accepted data inner products, the master is able to recover the desired result $\mathbf{Ax}$.

To facilitate the understanding, a simple example is given as follows:

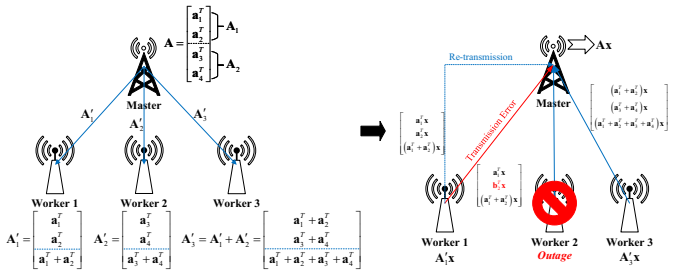


Fig. 4. A simple example of JCTC scheme with a master and 3 workers. After the two-layer encoding, $\mathbf{A}'_i$ is allocated to worker i. During the first transmission for worker 1, here is a transmission error. Thus, worker 1 re-transmits its computed result. Worker 2 is a straggler, which slows down the whole networks. With the help of coded computation, the master can recover $\mathbf{Ax}$ without waiting for worker 2.

Example 1. As illustrated in Fig. 4, a wireless distributed network with one master and three workers is considered. The corresponding steps in JCTC scheme can be described in the following.

1) *Two-Layer Encoding.* Matrix $\mathbf{A}$ is partitioned into 2 submatrixes: $\mathbf{A}_1$ and $\mathbf{A}_2$. Each submatrix contains two row vectors. Then, the two-layer encoded matrixes $\mathbf{A}'_1$, $\mathbf{A}'_2$ and $\mathbf{A}'_3 = \mathbf{A}'_1 + \mathbf{A}'_2$ are generated and each will be sent to a corresponding worker by the master. These encoded matrixes contain three row vectors, where the third one is used for error detection by summing up the first two row vectors.

2) *Distributed Computing and Serial Transmitting.* After receiving the input vector $\mathbf{x}$ broadcast from the master, each worker multiplies $\mathbf{x}$ with the two-layer encoded matrix and transmits the computed result back to the master.

3) *Error Detection and Re-transmission.* The master then will check whether transmission errors occur. For instance, there is a transmission error during the first transmission of worker 1 if $\mathbf{a}_1^T\mathbf{x} + \mathbf{b}_1^T\mathbf{x} \neq \left(\mathbf{a}_1^T + \mathbf{a}_2^T\right)\mathbf{x}$. Thus, worker 1 re-transmits the computed result and the second transmitted result is accepted by the master.

4) *Recovering Desired Result.* The master can only receive $\mathbf{A}'_1$ and $\mathbf{A}'_3$ from worker 1 and 3 respectively because of the outage for worker 2. By subtracting $\mathbf{A}'_1$ from $\mathbf{A}'_3$, the master can recover $\mathbf{A}'_2$ and hence $\mathbf{Ax}$ without waiting for the slowest worker.

This example implies that our JCTC scheme can not only mitigate the stragglers' effect, but also achieve the low dynamic encoding because of the two-layer encoding strategy for $\mathbf{A}$. No matter how the input vector $\mathbf{x}$ changes, the master can still detect the transmission errors. And workers never perform the error-detecting coding before transmission.

As matrix multiplication is one of the key and fundamental computational tasks underlying machine learning and big data analytics, our proposed JCTC scheme also has potential applications in those areas. For example, convolutional neural networks (CNN) convolve their input data with kernels in each layer [35]. With regard to m kernels, m convolutions need to be computed and each convolution operation can be performed as an inner product of two vectors. In other words, the matrix $\mathbf{A}$ is consisted of m kernels and the vector $\mathbf{x}$ represents the input to the neural network. For another example, the

encoders in Transformer perform the matrix calculations of self-attention [36]. The system matrix $\mathbf{A}$ represents the weight matrixes which have been trained and the input represents the embeddings. Then, the output query, key, and value matrixes can be produced through multiplications. Also, the proposed JCTC scheme can be extended to more complex wireless environments, once the transition probability (or the channel bit error rate) of each worker is obtained by the reference signal or other channel estimation techniques.

According to the proposed JCTC scheme, we can define the following metrics to evaluate the performances of the whole networks.

Definition 1 (**Computation Latency**)**.** The computation latency, denoted as T_{cmp}, is the time spent on calculating $(1+\eta)\,m$ data inner products for the whole networks. T_{cmp} is a random variable and can be given as:

$$T_{\mathrm{cmp}} = \max_{i\in[n]} T_i^{\mathrm{cmp}}, \tag{2}$$

where all the random variables $T_1^{\mathrm{cmp}}, T_2^{\mathrm{cmp}}, ..., T_n^{\mathrm{cmp}}$ are assumed to be mutually independent.

Definition 2 (**Transmission Latency**)**.** The transmission latency, denoted as T_{trn}, is the time spent on transmitting $(1+\eta)\,m$ accepted data inner products for the whole networks. T_{trn} is a random variable related to T_{cmp} and can be given as:

$$T_{\mathrm{trn}} = \sum\nolimits_{i=1}^{n} T_i^{\mathrm{trn}}\left(T_{\mathrm{cmp}}\right). \tag{3}$$

Definition 3 (**Computation Reliability**)**.** The computation reliability for the whole networks, denoted as R_{cmp}, represents the ratio of correct data inner products to all the $(1+\eta)\,m$ data inner products accepted by the master, which can be given as:

$$R_{\mathrm{cmp}} = 1 - \frac{N_{\mathrm{un}}}{(1+\eta)\,m}. \tag{4}$$

IV. Performance Analysis

In this section, we will analyze the performances of the JCTC scheme. First, the bounds of expected computation latency and the expectation of transmission latency will be obtained. Then, we will discuss the factors that influence the computation reliability and present the constraint of computation reliability. At last, the superiority of JCTC scheme will be shown.

A. Latency Analysis

1) Computation Latency: Due to the sub-block division and the error-detecting coding at the master, the time of computing j sub-blocks, i.e. $j\left(b_i+r_i\right)$ inner products, is denoted as a random variable $T_{i,j}^{\mathrm{cmp}}$. The cumulative distribution function (CDF) of $T_{i,j}^{\mathrm{cmp}}$ can be described as a shifted exponential distribution [6]:

$$\Pr\left[T_{i,j}^{\mathrm{cmp}} \le t\right] = 1 - e^{-\frac{\mu_i^{\mathrm{cmp}}}{j\left(b_i+r_i\right)}\left(t - j\left(b_i+r_i\right)a_i\right)}, \tag{5}$$

for $t \ge j\left(b_i+r_i\right)a_i$ and $j \le k$, where μ_i^{cmp} and a_i denotes the straggling and shift parameters, respectively, determined by the computation capability of worker i. This latency model fits the distribution of computation time in cloud computing environments well. From Eq. (5), we can observe that

$$T_i^{\mathrm{cmp}} = c_i\left(b_i+r_i\right)\left(\hat{T}_i^{\mathrm{cmp}} + a_i\right), \tag{6}$$

where the random variable $\hat{T}_i^{\mathrm{cmp}}$ is exponentially distributed with rate parameter μ_i^{cmp} representing the initial setup time at worker i before actually beginning computing an inner product, and c_i is the number of sub-blocks computed by worker i completely before completing a total of $(1+\eta)\,m$ accepted data inner products in the network. In Eq. (6), $\hat{T}_i^{\mathrm{cmp}} + a_i$ indicates the time spent on computing one inner product by worker i.

As mentioned in our JCTC scheme, workers perform their computations with sub-block division. The number of sub-blocks calculated by worker i till the given computation time t_{c} is denoted as $x_i\left(t_{\mathrm{c}}\right)$ in the following lemma.

Lemma 1. With a given computation time t_{c}, the average number of sub-blocks calculated by worker i can be derived as

$$\mathbb{E}\left[x_i\left(t_{\mathrm{c}}\right)\right] = \sum_{j=1}^{k}\left(1 - e^{-\frac{\mu_i^{\mathrm{cmp}}}{j\left(b_i+r_i\right)}\left(t_{\mathrm{c}} - j\left(b_i+r_i\right)a_i\right)}\right). \tag{7}$$

Proof. See Appendix A. ∎

According to Lemma 1, we can find that $c_i = x_i\left(T_{\mathrm{cmp}}\right)$ and $\sum_{i=1}^{n} b_i \mathbb{E}\left[x_i\left(T_{\mathrm{cmp}}\right)\right] \ge (1+\eta)\,m$ should be satisfied so that the master can recover the desired result successfully.

For the heterogeneous networks, the order statistics cannot be used to describe the computation latency due to sub-block division and disparate capabilities of workers, so that the exact expression of $\mathbb{E}\left[T_{\mathrm{cmp}}\right]$ is hard to obtain. So the lower and upper bounds of computation latency are discussed in the following.

Lemma 2. Set $\mu_{\mathrm{g}}^{\mathrm{cmp}} = \max_i \mu_i^{\mathrm{cmp}}$ and $a_{\mathrm{g}} = \min_i a_i$, $i \in [n]$. The lower bound of T_{cmp} is given as

$$T_{\mathrm{cmp}} \ge \frac{(1+\eta)\,m}{n}\left(\hat{T}_{(1)}^{\mathrm{cmp}} + a_{\mathrm{g}}\right), \tag{8}$$

where $\hat{T}_{(1)}^{\mathrm{cmp}}$ is the first order statistic that follows exponential distribution with rate parameter $\mu_{\mathrm{g}}^{\mathrm{cmp}}$.

Proof. See Appendix B. ∎

As a result, the expected lower bound can be described in the following proposition.

Proposition 1 (*The Expected Lower Bound of Computation Latency*). In the JCTC scheme, the expected lower bound of computation latency $\mathcal{L}_{\mathrm{cmp}}$ can be given as

$$\mathbb{E}\left[T_{\mathrm{cmp}}\right] \ge \mathcal{L}_{\mathrm{cmp}} = \frac{(1+\eta)\,m}{n}\left(\frac{1}{n\mu_{\mathrm{g}}^{\mathrm{cmp}}} + a_{\mathrm{g}}\right). \tag{9}$$

Proof. Based on Lemma 2 and the characteristics of order statistic, $\mathcal{L}_{\mathrm{cmp}}$ can be obtained by taking the expectation of (8). ∎

Lemma 3. Set $\mu_{\mathrm{b}}^{\mathrm{cmp}} = \min_i \mu_i^{\mathrm{cmp}}$, $a_{\mathrm{b}} = \max_i a_i$ and $r_{\mathrm{m}} = \max_i r_i$, $i \in [n]$. The upper bound of T_{cmp} is given as

$$T_{\mathrm{cmp}} \le \left(\frac{2\alpha m}{n} + (k+1)\,r_{\mathrm{m}}\right)\left(\bar{\hat{T}}_w^{\mathrm{cmp}} + a_{\mathrm{b}}\right), \tag{10}$$

for $w \in W_{\mathrm{e}}$, where

$$\bar{\hat{T}}_w^{\mathrm{cmp}} = \frac{\sum_{w \in W_{\mathrm{e}}} \hat{T}_w^{\mathrm{cmp}}}{\sum_{w \in W_{\mathrm{e}}} 1},$$

the set of workers that have not completed all their computational tasks until T_{cmp} is denoted as W_{e}, i.e. $W_{\mathrm{e}} = \{\, i |\, c_i b_i < \alpha m/n\}$, and $\hat{T}_w^{\mathrm{cmp}}$ is a exponential random variable with rate parameter $\mu_{\mathrm{b}}^{\mathrm{cmp}}$.

Proof. See Appendix C. ∎

As a result, the expected upper bound can be described in the following proposition.

Proposition 2 (*The Expected Upper Bound of Computation Latency*). In the JCTC scheme, the expected upper bound of computation latency $\mathcal{U}_{\mathrm{cmp}}$ can be given as

$$\mathbb{E}\left[T_{\mathrm{cmp}}\right] \leq \mathcal{U}_{\mathrm{cmp}} = \left(\frac{2\alpha m}{n} + (k+1)\, r_{\mathrm{m}}\right)\left(\frac{1}{\mu_{\mathrm{b}}^{\mathrm{cmp}}} + a_{\mathrm{b}}\right). \tag{11}$$

Proof. Based on Lemma 3, $\mathcal{U}_{\mathrm{cmp}}$ can be obtained by taking the expectation of (10). ∎

Remark 1 (*Special Cases*). Assuming $m = \Theta(n)$, $\alpha = \Theta(1)$, $\eta = \Theta(1)$, $\mu_{\mathrm{g}}^{\mathrm{cmp}} = \Theta(1)$, $a_{\mathrm{g}} = \Theta(1)$, $\mu_{\mathrm{b}}^{\mathrm{cmp}} = \Theta(1)$, $a_{\mathrm{b}} = \Theta(1)$, $k = \Theta(1)$ and $r_{\mathrm{m}} = \Theta(1)$ [11], one has

$$\mathbb{E}\left[T_{\mathrm{cmp}}\right] = \Theta(1),$$

from Eq. (9) and Eq. (11). Consider a scenario where the channel condition is so good that $\varepsilon_i \to 0, i \in [n]$. Then, the upper bound of $\mathbb{E}\left[T_{\mathrm{cmp}}\right]$ can be simplified to

$$\mathbb{E}\left[T_{\mathrm{cmp}}\right] \leq \mathcal{U}_{\mathrm{cmp}} = \frac{2\alpha m}{n}\left(\frac{1}{\mu_{\mathrm{b}}^{\mathrm{cmp}}} + a_{\mathrm{b}}\right),$$

since the error-detecting redundancy is not required.

2) Transmission Latency: Due to the instability of wireless channel and the disparate transmission capabilities, it is assumed that the transmission time for a single inner product follows a mutually independent exponential distribution [12], [37] with the rate parameter μ_i^{trn}, which represents the transmission capability for worker i. In our JCTC scheme, a sub-block with detectable transmission errors is required to be re-transmitted. We denote the total number of times for a sub-block transmitted by worker i as $k_{\mathrm{re},i}$, which follows a geometric distribution that can be given as:

$$\Pr\left[k_{\mathrm{re},i} = j\right] = p_{\mathrm{s},i}(1 - p_{\mathrm{s},i})^{j-1}, \tag{12}$$

where $p_{\mathrm{s},i}$ is assumed as the success probability and its detailed expression will be discussed in Section IV-B. Then, the time $T_i^{\mathrm{trn}}(T_{\mathrm{cmp}})$ spent on transmitting c_i sub-blocks can be obtained by

$$\begin{aligned} T_i^{\mathrm{trn}}(T_{\mathrm{cmp}}) &= \sum_{u=1}^{k_{\mathrm{re},i}} \sum_{\kappa=1}^{c_i(b_i+r_i)} T_{i,(\kappa^{\mathrm{th}})}^{\mathrm{trn}} = \sum_{\kappa=1}^{c_i(b_i+r_i)} \sum_{u=1}^{k_{\mathrm{re},i}} T_{i,(\kappa^{\mathrm{th}})}^{\mathrm{trn}} \\ &= \sum_{\kappa=1}^{c_i(b_i+r_i)} T_{i,\mathrm{re},(\kappa^{\mathrm{th}})}^{\mathrm{trn}}, \end{aligned} \tag{13}$$

for $c_i > 0$, where $T_{i,(\kappa^{\mathrm{th}})}^{\mathrm{trn}}$ is the time for the result of the κ^{th} inner product transmitted by worker i and $T_{i,\mathrm{re},(\kappa^{\mathrm{th}})}^{\mathrm{trn}} = \sum_{u=1}^{k_{\mathrm{re},i}} T_{i,(\kappa^{\mathrm{th}})}^{\mathrm{trn}}$ is the transmission time of the κ^{th} inner product until it is accepted by the master. Obviously, $T_i^{\mathrm{trn}}(T_{\mathrm{cmp}}) = 0$ if $c_i = 0$, which means that worker i has no completed sub-block to transmit by the time T_{cmp}. In the following lemma, we state the statistical property of $T_{i,\mathrm{re},(\kappa^{\mathrm{th}})}^{\mathrm{trn}}$.

Lemma 4. The random variable $T_{i,\mathrm{re},(\kappa^{\mathrm{th}})}^{\mathrm{trn}}$ follows an exponential distribution with rate parameter $p_{\mathrm{s},i}\mu_i^{\mathrm{trn}}$, i.e.,

$$\Pr\left[T_{i,\mathrm{re},(\kappa^{\mathrm{th}})}^{\mathrm{trn}} \leq t\right] = 1 - e^{-p_{\mathrm{s},i}\mu_i^{\mathrm{trn}} t}. \tag{14}$$

Proof. See Appendix D. ∎

Based on Lemma 4, we present an another lemma to show the expectation of $T_i^{\mathrm{trn}}(t_{\mathrm{c}})$ in the following.

Lemma 5. The expected random variable $\mathbb{E}\left[T_i^{\mathrm{trn}}(t_{\mathrm{c}})\right]$ can be given as

$$\mathbb{E}\left[T_i^{\mathrm{trn}}(t_{\mathrm{c}})\right] = \frac{b_i + r_i}{p_{\mathrm{s},i}\mu_i^{\mathrm{trn}}} \sum_{j=1}^{k} \left(1 - e^{-\frac{\mu_i^{\mathrm{cmp}}}{j(b_i+r_i)}(t_{\mathrm{c}} - j(b_i+r_i)a_i)}\right). \tag{15}$$

Proof. See Appendix E. ∎

As a result, the expectation of transmission latency can be given in the following proposition.

Proposition 3 (*The Expectation of Transmission Latency*). In the JCTC scheme, the expectation of transmission latency $\mathbb{E}\left[T_{\mathrm{trn}}\right]$ for the whole networks can be given as

$$\begin{aligned} \mathbb{E}\left[T_{\mathrm{trn}}\right] &= \sum_{i=1}^{n} \mathbb{E}\left[T_i^{\mathrm{trn}}(T_{\mathrm{cmp}})\right] \\ &= \sum_{i=1}^{n} \frac{b_i + r_i}{p_{\mathrm{s},i}\mu_i^{\mathrm{trn}}} \sum_{j=1}^{k} \left(1 - e^{-\frac{\mu_i^{\mathrm{cmp}}}{j(b_i+r_i)}(T_{\mathrm{cmp}} - j(b_i+r_i)a_i)}\right). \end{aligned} \tag{16}$$

Proof. According to Lemma 5 and Eq. (3), $\mathbb{E}\left[T_{\mathrm{trn}}\right]$ can be obtained by substituting $t_{\mathrm{c}} = T_{\mathrm{cmp}}$ into Eq. (15) and summing it over all $i \in [n]$. ∎

B. Computation Reliability Analysis

After the wireless transmission from workers to the master, several scenarios may occur at the master:

- *No error.* The probability that the master receives a sub-block with no error from worker i is denoted as $p_{\mathrm{c},i}$. From the channel model, we know that

$$p_{\mathrm{c},i} = (1 - \varepsilon_i)^{q(b_i+r_i)}. \tag{17}$$

- *Undetected errors.* The probability that the master receives a sub-block with an undetected error pattern from worker i is denoted as $p_{\mathrm{e},i}$. With regard to all the $(b_i + r_i, b_i)$ linear error-detecting codes, the average probability of undetected errors has been proved [38]–[40] that $\bar{p}_{\mathrm{e},i} = (1 - (1 - \varepsilon_i)^{qb_i})2^{-qr_i}$. In this paper, it is assumed that $p_{\mathrm{e},i} \approx \bar{p}_{\mathrm{e},i}$, i.e.

$$p_{\mathrm{e},i} = \left(1 - (1 - \varepsilon_i)^{qb_i}\right) 2^{-qr_i}. \tag{18}$$

• *Detectable errors.* The probability that the master receives a sub-block with a detectable error pattern from worker i is denoted as $p_{\mathrm{d},i}$. It can be obtained by $p_{\mathrm{c},i}$ and $p_{\mathrm{e},i}$, i.e.

$$p_{\mathrm{d},i} = 1 - p_{\mathrm{c},i} - p_{\mathrm{e},i}. \tag{19}$$

A received sub-block is accepted by the master only if it either contains no error or an undetected error pattern. Otherwise, if the master detects the transmission errors, the corresponding sub-block will be re-transmitted until it is accepted. Notice that the number of transmission for a single sub-block follows a geometric distribution with the success probability $p_{\mathrm{s},i} = p_{\mathrm{c},i} + p_{\mathrm{e},i}$ in Eq. (12).

For wireless coded computation, the accepted sub-blocks with undetected errors will affect the accuracy of the desired result $\mathbf{Ax}$ and decrease the computation reliability for the whole distributed networks. In the JCTC scheme, we require the ratio of the number of data inner products with undetected errors to $(1+\eta)\,m$ accepted data inner products in $\mathbf{Ax}$ does not exceed p_{r}, where p_{r} is the tolerable maximum error inner product rate. It means that there is at most $(1+\eta)\,mp_{\mathrm{r}}$ undetected error inner products in the desired result.

As a result, the expected computation reliability for the whole networks can be given in the following proposition.

Proposition 4 (*The Expectation of Computation Reliability*). In the JCTC scheme, the expectation of computation reliability $\mathbb{E}\left[R_{\mathrm{cmp}}\right]$ for the whole networks can be obtained by

$$\mathbb{E}\left[R_{\mathrm{cmp}}\right] = 1 - \frac{\sum\limits_{i=1}^{n} \frac{b_i p_{\mathrm{e},i}}{p_{\mathrm{c},i}+p_{\mathrm{e},i}} \mathbb{E}\left[x_i\left(T_{\mathrm{cmp}}\right)\right]}{(1+\eta)\,m}. \tag{20}$$

Proof. Due to the re-transmission under detectable errors, the expected total number of transmission for worker i sending $x_i\left(t_{\mathrm{c}}\right)$ sub-blocks is denoted as $\mathbb{E}\left[z_i\left(t_{\mathrm{c}}\right)\right]$ with the given computation time t_{c}. Then, $\mathbb{E}\left[z_i\left(t_{\mathrm{c}}\right)\right]$ can be obtained by

$$\mathbb{E}\left[z_i\left(t_{\mathrm{c}}\right)\right] = \frac{\mathbb{E}\left[x_i\left(t_{\mathrm{c}}\right)\right]}{p_{\mathrm{c},i}+p_{\mathrm{e},i}}. \tag{21}$$

With regard to a sub-block transmitted by worker i, it can be accepted on the initial transmission or any re-transmissions. Although it is re-transmitted for many times, there can still be errors for an accepted sub-block because of the limited error-detecting ability. We denote the probability that an accepted sub-block contains undetected errors as $p_{\mathrm{u},i}$ and it can be given by

$$\begin{aligned} p_{\mathrm{u},i} &= p_{\mathrm{e},i} + p_{\mathrm{d},i}p_{\mathrm{e},i} + p_{\mathrm{d},i}^2 p_{\mathrm{e},i} + \cdots \\ &= \sum_{j=1}^{\infty} p_{\mathrm{d},i}^{j-1} p_{\mathrm{e},i} = \frac{p_{\mathrm{e},i}}{p_{\mathrm{c},i}+p_{\mathrm{e},i}}, \end{aligned} \tag{22}$$

where $1 - p_{\mathrm{d},i} = p_{\mathrm{c},i} + p_{\mathrm{e},i}$ has been used.

From Eq. (21) and Eq. (22), we know that the average number of data inner products with undetected error patterns can be given as

$$\begin{aligned} \mathbb{E}\left[N_{\mathrm{un}}\right] &= \sum_{i=1}^{n} b_i p_{\mathrm{e},i} \mathbb{E}\left[z_i\left(T_{\mathrm{cmp}}\right)\right] \\ &= \sum_{i=1}^{n} b_i \mathbb{E}\left[x_i\left(T_{\mathrm{cmp}}\right)\right] p_{\mathrm{u},i} \\ &= \sum_{i=1}^{n} \frac{b_i p_{\mathrm{e},i}}{p_{\mathrm{c},i}+p_{\mathrm{e},i}} \mathbb{E}\left[x_i\left(T_{\mathrm{cmp}}\right)\right]. \end{aligned} \tag{23}$$

Then, $\mathbb{E}\left[R_{\mathrm{cmp}}\right]$ can be obtained by taking the expectation of Eq. (4) and substituting Eq. (23) into it. ■

Remark 2 (*The Constraint of Computation Reliability*). From Proposition 4, the corresponding constraint of computation reliability in our JCTC scheme can be shown as:

$$\mathbb{E}\left[R_{\mathrm{cmp}}\right] \geq 1 - p_{\mathrm{r}}, \tag{24}$$

where the right-hand side of the constraint (24) implies the ratio of the minimum number of correct inner products to the total $(1+\eta)\,m$ inner products. Our design must satisfy the constraint in order to obtain the desired result and meet the requirement of computation reliability at the same time.

C. Comparison with Separate Designs

In the existing separate designs, each worker needs to encode its computed sub-block for error detection by itself before sending to the master, as shown in Fig. 2. In other words, coded computation is independent of error detection and they are designed separately. It implies that each worker not only computes a data sub-block but also spends some time encoding it for error detection. The encoding task performed by worker i can be described by a matrix multiplication as follows:

$$\mathbf{G}_{i,j}\left(\tilde{\mathbf{A}}_{i,j}\mathbf{x}\right), \tag{25}$$

where $\mathbf{G}_{i,j} \in \mathbb{F}_{2^q}^{(b_i+r_i)\times b_i}$ is the coding matrix used for error detection. For JCTC scheme, the worker i only calculates the matrix multiplication $\tilde{\mathbf{A}}'_{i,j}\mathbf{x}$ including the data matrix multiplication $\tilde{\mathbf{A}}''_{i,j}\mathbf{x}$ and the redundancy matrix multiplication $\tilde{\mathbf{S}}_{i,j}\mathbf{x}$. So the computational task for error detection in JCTC scheme can be represented by the redundancy matrix multiplication. With the same computation reliability, it costs less calculated amount for error detection in JCTC scheme than the one in separate designs. The comparison of calculated amount for workers is shown in the following proposition.

Proposition 5 (*Comparison of Calculated Amount for Error Detection*). Assume that the data sub-block size and the error-detecting redundancy obtained by both JCTC scheme and separate designs are the same, which implies that the computation reliability of both schemes is also the same. Compared with separate designs, the JCTC scheme can decrease the calculated amount served as error detection for all n workers by at least $\sum_{i=1}^{n} x_i\left(t_{\mathrm{c}}\right) b_i\left(2b_i - 1\right)$ operations including additions and multiplications with the given computation time t_{c}, when $d \leq \min_i b_i$.

Proof. In the JCTC scheme, worker i calculates each sub-block with a redundancy matrix multiplication $\tilde{\mathbf{S}}_{i,j}\mathbf{x}$. It needs extra $r_i\left(d-1\right)$ additions and $r_i d$ multiplications. So there are a total of $\sum_{i=1}^{n} x_i\left(t_{\mathrm{c}}\right) r_i\left(2d-1\right)$ operations served as error detection for all n workers with the given time t_{c}.

For the separate designs, the encoding task performed by worker i is described as $\mathbf{G}_{i,j}(\tilde{\mathbf{A}}_{i,j}\mathbf{x})$. Worker i encodes each sub-block for error detection with extra $\left(b_i+r_i\right)\left(b_i-1\right)$ additions and $\left(b_i+r_i\right)b_i$ multiplications. Thus, here are a total of $\sum_{i=1}^{n} x_i\left(t_{\mathrm{c}}\right)\left(b_i+r_i\right)\left(2b_i-1\right)$ encoding operations for all n workers in the separate designs.

Notice that when $d \leq \min_i b_i$, the JCTC scheme only needs at most $\sum_{i=1}^{n} x_i\left(t_{\mathrm{c}}\right) r_i\left(2b_i-1\right)$ operations served as error detection for all n workers and can decrease at least $\sum_{i=1}^{n} x_i\left(t_{\mathrm{c}}\right) b_i\left(2b_i-1\right)$ operations, compared with the existing design. ∎

Because each worker encodes its computed data sub-blocks by itself, the whole computation time for the separate designs, denoted as $T_{\mathrm{tot,c}}^{\mathrm{S}}$, contains the original time calculating matrix multiplication and the encoding time for error detection, represented by $T_{\mathrm{cmp}}^{\mathrm{S}}$ and $T_{\mathrm{cc}}^{\mathrm{S}}$ respectively. Since the worker with poor capability for matrix multiplication is also weak in encoding, we assumed that $T_{\mathrm{tot,c}}^{\mathrm{S}}$ can be approximated by the sum of $T_{\mathrm{cmp}}^{\mathrm{S}}$ and $T_{\mathrm{cc}}^{\mathrm{S}}$, i.e. $T_{\mathrm{tot,c}}^{\mathrm{S}} = T_{\mathrm{cmp}}^{\mathrm{S}} + T_{\mathrm{cc}}^{\mathrm{S}}$. Further, $T_{\mathrm{cmp}}^{\mathrm{S}}$ and $T_{\mathrm{cc}}^{\mathrm{S}}$ can be obtained by

$$T_{\mathrm{cmp}}^{\mathrm{S}} = \max_{i\in[n]} T_i^{\mathrm{S,cmp}}, \quad T_{\mathrm{cc}}^{\mathrm{S}} = \max_{i\in[n]} T_i^{\mathrm{S,cc}},$$

where $T_i^{\mathrm{S,cmp}}$ is the original time calculating matrix multiplications $\{\tilde{\mathbf{A}}_{i,j}\mathbf{x}\}_{j=1}^{c_i}$ and $T_i^{\mathrm{S,cc}}$ denotes the encoding time calculating $\{\mathbf{G}_{i,j}(\tilde{\mathbf{A}}_{i,j}\mathbf{x})\}_{j=1}^{c_i}$ for worker i. And the CDF of $T_i^{\mathrm{S,cmp}}$ and $T_i^{\mathrm{S,cc}}$ can be given respectively by

$$\Pr\left[T_i^{\mathrm{S,cmp}} \leq t\right] = 1 - e^{-\frac{\mu_i^{\mathrm{cmp}}}{c_i b_i}\left(t - c_i b_i a_i\right)}, \tag{26}$$

$$\Pr\left[T_i^{\mathrm{S,cc}} \leq t\right] = 1 - e^{-\frac{\mu_i^{\mathrm{cc}}}{c_i\left(b_i+r_i\right)}\left(t - c_i\left(b_i+r_i\right)a_i^{\mathrm{cc}}\right)}, \tag{27}$$

where μ_i^{cc} and a_i^{cc} represent the encoding capability for worker i. In the following proposition, we compare the whole computation time between these two schemes.

Proposition 6 (*Comparison of the Whole Computation Time*). Assume that the data sub-block size and the error-detecting redundancy obtained by both JCTC scheme and separate designs are the same, which implies that the computation reliability of both schemes is also the same. When $\mu_i^{\mathrm{cc}} = \mu_i^{\mathrm{cmp}}$ and $a_i^{\mathrm{cc}} = a_i$, the difference in the expected whole computation time between these two schemes is bounded as

$$\mathcal{L}_{\mathrm{cmp}}^{\mathrm{S}} \leq \mathbb{E}\left[T_{\mathrm{tot,c}}^{\mathrm{S}}\right] - \mathbb{E}\left[T_{\mathrm{cmp}}\right] \leq \mathcal{U}_{\mathrm{cmp}}^{\mathrm{S}}, \tag{28}$$

where $\mathcal{L}_{\mathrm{cmp}}^{\mathrm{S}}$ is the lower bound of $\mathbb{E}\left[T_{\mathrm{cmp}}^{\mathrm{S}}\right]$ and $\mathcal{U}_{\mathrm{cmp}}^{\mathrm{S}}$ is its upper bound.

Proof. From Eq. (26) and Eq. (27), we notice that $T_i^{\mathrm{S,cmp}}$ and $T_i^{\mathrm{S,cc}}$ can be rewritten as $T_i^{\mathrm{S,cmp}} = c_i b_i(\hat{T}_i^{\mathrm{S,cmp}} + a_i)$ and $T_i^{\mathrm{S,cc}} = c_i(b_i+r_i)(\hat{T}_i^{\mathrm{S,cc}} + a_i^{\mathrm{cc}})$, where the random variables $\hat{T}_i^{\mathrm{S,cmp}}$ and $\hat{T}_i^{\mathrm{S,cc}}$ are exponentially distributed with rate parameter μ_i^{cmp} and μ_i^{cc} respectively. Similar to the computation latency analysis of the JCTC scheme in Section IV-A1, the bounds of $\mathbb{E}\left[T_{\mathrm{cmp}}^{\mathrm{S}}\right]$ and $\mathbb{E}\left[T_{\mathrm{cc}}^{\mathrm{S}}\right]$ can be given as

$$\mathcal{L}_{\mathrm{cmp}}^{\mathrm{S}} = \frac{(1+\eta)\,m}{n}\left(\frac{1}{n\mu_{\mathrm{g}}^{\mathrm{cmp}}} + a_{\mathrm{g}}\right), \tag{29}$$

$$\mathcal{U}_{\mathrm{cmp}}^{\mathrm{S}} = \frac{2\alpha m}{n}\left(\frac{1}{\mu_{\mathrm{b}}^{\mathrm{cmp}}} + a_{\mathrm{b}}\right), \tag{30}$$

$$\mathcal{L}_{\mathrm{cc}}^{\mathrm{S}} = \frac{(1+\eta)\,m}{n}\left(\frac{1}{n\mu_{\mathrm{g}}^{\mathrm{cc}}} + a_{\mathrm{g}}^{\mathrm{cc}}\right), \tag{31}$$

$$\mathcal{U}_{\mathrm{cc}}^{\mathrm{S}} = \left(\frac{2\alpha m}{n} + (k+1)\,r_{\mathrm{m}}\right)\left(\frac{1}{\mu_{\mathrm{b}}^{\mathrm{cc}}} + a_{\mathrm{b}}^{\mathrm{cc}}\right), \tag{32}$$

where $\mu_{\mathrm{g}}^{\mathrm{cc}} = \max_i \mu_i^{\mathrm{cc}}$, $a_{\mathrm{g}}^{\mathrm{cc}} = \min_i a_i^{\mathrm{cc}}$, $\mu_{\mathrm{b}}^{\mathrm{cc}} = \min_i \mu_i^{\mathrm{cc}}$, $a_{\mathrm{b}}^{\mathrm{cc}} = \max_i a_i^{\mathrm{cc}}$ and $\mathcal{L}_{\mathrm{cc}}^{\mathrm{S}}$, $\mathcal{U}_{\mathrm{cc}}^{\mathrm{S}}$ represent the lower bound and the upper bound of $\mathbb{E}\left[T_{\mathrm{cc}}^{\mathrm{S}}\right]$ respectively. Then, the expected whole computation time in separate designs can be described as

$$\mathcal{L}_{\mathrm{cmp}}^{\mathrm{S}} + \mathcal{L}_{\mathrm{cc}}^{\mathrm{S}} \leq \mathbb{E}\left[T_{\mathrm{tot,c}}^{\mathrm{S}}\right] \leq \mathcal{U}_{\mathrm{cmp}}^{\mathrm{S}} + \mathcal{U}_{\mathrm{cc}}^{\mathrm{S}}. \tag{33}$$

Thus, when $\mu_i^{\mathrm{cc}} = \mu_i^{\mathrm{cmp}}$ and $a_i^{\mathrm{cc}} = a_i$, we can know that $\mathcal{L}_{\mathrm{cmp}}^{\mathrm{S}} \leq \mathbb{E}\left[T_{\mathrm{tot,c}}^{\mathrm{S}}\right] - \mathbb{E}\left[T_{\mathrm{cmp}}\right] \leq \mathcal{U}_{\mathrm{cmp}}^{\mathrm{S}}$ from Eq. (9), Eq. (11) and Eq. (33). ∎

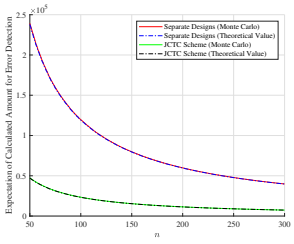


Fig. 5. The expectation of calculated amount for error detection versus the number of workers n, where $m = 5000$, $\alpha = 1.25$, $k = 3$, $b_i = d = \alpha m/kn$, $r_i = b_i/9$, $\mu_i^{\mathrm{cmp}} = \mu_i^{\mathrm{cc}} \sim \mathcal{U}\left(15, 25\right)$ (row/ms) and $a_i = a_i^{\mathrm{cc}} \sim \mathcal{U}\left(0.25, 1\right)$ (ms/row) for $\forall i \in [n]$.

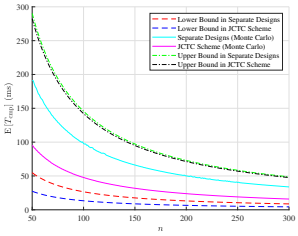


Fig. 6. The expectation of the whole computation latency versus the number of workers n, where $m = 5000$, $\alpha = 1.25$, $k = 3$, $b_i = d = \alpha m/kn$, $r_i = b_i/9$, $\mu_i^{\mathrm{cmp}} = \mu_i^{\mathrm{cc}} \sim \mathcal{U}\left(15, 25\right)$ (row/ms) and $a_i = a_i^{\mathrm{cc}} \sim \mathcal{U}\left(0.25, 1\right)$ (ms/row) for $\forall i \in [n]$.

The above propositions imply that the less calculated amount for workers can also lead to the lower computation latency for the whole networks. Hence, the total latency in JCTC scheme is lower than that in separate designs under the same computation reliability.

Fig. 5 and Fig. 6 show the expectation of calculated amount for error detection and the whole computation latency versus the number of workers respectively. It is observed that Monte Carlo simulation results are in good agreement with the theoretical ones. As n increases, both calculated amount and latency decrease, since more computing resources are utilized with larger n. Moreover, the JCTC scheme performs better than the separate designs, which confirms the theoretical analysis.

V. Optimal Task and Redundancy Allocation

For JCTC scheme, minimizing the upper bound of the expected total latency $\mathbb{E}\left[T_{\mathrm{cmp}} + T_{\mathrm{trn}}\right]$ is considered, which can still lead to a decrease in total latency. Under the condition

that the required computation reliability is satisfied, latency is reduced as much as possible by designing the optimal data sub-block size and the corresponding optimal error-detecting redundancy for each worker. Thus, the optimization problem can be formulated as follows:

$$\mathcal{P}_0: \min_{\boldsymbol{b},\boldsymbol{r}} \mathbb{E}\left[T_{\rm cmp}+T_{\rm trn}\right]$$
$$\text{s.t. } 0\le b_i\le l/k, r_i\ge 0, \forall i\in[n] \tag{34}$$
$$\Pr\left[\sum_{i=1}^{n} b_i x_i\left(T_{\rm cmp}\right) < (1+\eta)\, m\right] = o\left(1/n\right), \tag{35}$$
$$\sum_{i=1}^{n} b_i \mathbb{E}\left[x_i\left(T_{\rm cmp}\right)\right] p_{{\rm u},i} \le (1+\eta)\, m p_{\rm r}. \tag{36}$$

In $\mathcal{P}_0$, the constraint (34) determines the range of data sub-block and error-detecting redundancy. The constraint (35) ensures that the master can aggregate a sufficient number of data inner products to recover the desired computational result successfully, and the computation reliability of the whole networks is guaranteed in (36). For any given $\mu_i^{\rm cmp} > 0$, $a_i > 0$, $\mu_i^{\rm trn} > 0$ and $0 \le \varepsilon_i < 1/2$, $\mathcal{P}_0$ is always feasible because there exists at least one feasible solution, i.e., $b_i = l/k$ and $r_i = +\infty$ for $i \in [n]$, satisfying the constraints of $\mathcal{P}_0$.

However, due to the heavy relation between transmission and computation latency for each worker, it is challenging to obtain the exact expression of $\mathbb{E}\left[T_{\rm cmp}+T_{\rm trn}\right]$, which makes this problem hard to solve. According to [7, Section III-A], we can introduce a new variable $t_{\rm cmp}$ to relax the term $\mathbb{E}\left[T_{\rm cmp}\right]$ and optimize the computation latency $t_{\rm cmp}$, the data sub-block size $\{b_i\}_{i=1}^n$ and the error-detecting redundancy $\{r_i\}_{i=1}^n$ simultaneously when the distribution of the random variable $T_{\rm cmp}$ is unknown. Then, the reformulated problem $\mathcal{P}_1$ can be obtained as follows:

$$\mathcal{P}_1: \min_{t_{\rm cmp},\boldsymbol{t},\boldsymbol{b},\boldsymbol{r}} t_{\rm cmp} + \sum_{i=1}^{n} t_i$$
$$\text{s.t. } 0\le b_i\le l/k, r_i\ge 0, \forall i\in[n]$$
$$(p_{{\rm c},i}+p_{{\rm e},i}) t_i \mu_i^{\rm trn} \le (b_i+r_i)\,\mathbb{E}\left[x_i\left(t_{\rm cmp}\right)\right], \forall i\in[n] \tag{37}$$
$$\sum_{i=1}^{n} (p_{{\rm c},i}+p_{{\rm e},i})\,\frac{b_i}{b_i+r_i} t_i \mu_i^{\rm trn} \ge (1+\eta)\, m, \tag{38}$$
$$\sum_{i=1}^{n} p_{{\rm e},i}\frac{b_i}{b_i+r_i} t_i \mu_i^{\rm trn} \le (1+\eta)\, m p_{\rm r}, \tag{39}$$

where the set $\{t_i\}_{i=1}^n$ is introduced to relax the term $\mathbb{E}\left[T_{\rm trn}\right]$, representing the transmission time of each worker. The constraint (37) implies the relationship between computation and transmission for each worker, which ensures that the number of transmitted sub-blocks is no more than the number of computed sub-blocks. To aggregate sufficient data inner products to recover the desired result, the number of total accepted results should be more than $(1+\eta)m$, which leads to the constraint (38). And the computation reliability required by the whole networks can be denoted as the constraint (39). The solution to $\mathcal{P}_1$ is provably asymptotically optimal when n becomes very large [13].

For $\mathcal{P}_1$, there are differences of convex (DC) structure and products of convex functions (PF) structure [41] in the constraint (37), (38) and (39), which makes this problem non-convex. In the following, we solve this problem in two different ways.

A. Iterative Optimization Algorithm

The non-convexity of $\mathcal{P}_1$ is caused by the DC structures and the PF structures in constraints. Using successive convex approximation (SCA) algorithms [42], we can transform such non-convex structures into convex approximations and iteratively solve the relaxed convex optimization problem to get sub-optimal solutions. For the DC structure, we can linearize the concave part by taking the Taylor expansion to obtain the convex upper approximation, while the product of convex functions can first be rewritten as a function with the DC structure according to [41] and then the corresponding convex upper approximation can be obtained by linearizing the concave part in the rewritten function for the PF structure. Hence, the relaxed convex optimization problem can be given as:

$$\mathcal{P}_1': \min_{t_{\rm cmp},\boldsymbol{t},\boldsymbol{b},\boldsymbol{r}} t_{\rm cmp} + \sum_{i=1}^{n} t_i$$
$$\text{s.t.} \quad 0\le b_i\le l/k, r_i\ge 0, \forall i\in[n] \tag{40}$$
$$f_{1,i}\left(t_i,b_i,r_i\right) - k\left(b_i+r_i\right)$$
$$+ (b_i+r_i)\sum_{j=1}^{k} e^{-\frac{\mu_i^{\rm cmp}}{j(b_i+r_i)}\left[t_{\rm cmp}-j(b_i+r_i)a_i\right]} \le 0, \forall i\in[n] \tag{41}$$
$$(1+\eta)\, m + \sum_{i=1}^{n} f_{2,i}\left(t_i,b_i,r_i\right) \le 0, \tag{42}$$
$$\sum_{i=1}^{n} f_{3,i}\left(t_i,b_i,r_i\right) - (1+\eta)\, m p_{\rm r} \le 0, \tag{43}$$

where $f_{1,i}\left(t_i,b_i,r_i\right)$, $f_{2,i}\left(t_i,b_i,r_i\right)$ and $f_{3,i}\left(t_i,b_i,r_i\right)$ are convex functions with respect to t_i, b_i and r_i. See Appendix F for the detailed convex approximations of DC and PF structures and the concrete expressions of $f_{1,i}\left(t_i,b_i,r_i\right)$, $f_{2,i}\left(t_i,b_i,r_i\right)$, $f_{3,i}\left(t_i,b_i,r_i\right)$. In $\mathcal{P}_1'$, note that the objective function and the constraint (40) are linear functions. Besides, the constraint (41) can be rewritten as a convex exponential cone. The constraint (42) and (43) are also convex since they are the sum of some convex functions. Thus, $\mathcal{P}_1'$ is a convex problem and we can solve it iteratively to find the sub-optimal approximate solution to $\mathcal{P}_1$. The wireless LT coded computation with error detection based on SCA (WLTCC-ED(SCA)) algorithm is provided, which is given as Alg. 1.

During each iteration of WLTCC-ED(SCA) algorithm, it is required to deal with $\mathcal{P}_1'$, which falls into a convex exponential cone programming category. It can be solved efficiently to a desired accuracy by using interior-point methods with MOSEK in polynomial time computational complexity of $\mathcal{O}\left(n^{3.5}\right)$.

Remark 3 (*Negligible Channel Transition Probability*). When the channel condition is so good that the sub-blocks transmitted from workers are almost error-free, i.e. $\varepsilon_i \to 0, i \in$

[1]One way to find out the initial points is to choose the optimal solution to $\mathcal{P}_1''$ as the value of $\boldsymbol{t}_0^{(0)}$, $\boldsymbol{b}_0^{(0)}$ and $\boldsymbol{r}_0^{(0)}$.

Algorithm 1 Wireless LT Coded Computation with Error Detection based on SCA

Require: The parameter tuple $(\mu_i^{\rm cmp}, a_i, \mu_i^{\rm trn}, \varepsilon_i, p_{\rm r})$ for each worker $i \in [n]$.

Ensure: The data sub-block size b_i^* and error-detecting redundancy r_i^* for the worker i.

procedure WLTCC-ED(SCA)
  Set the number of iterations $\beta = 0$, the proper initial step-size $\theta^{(0)} \in (0, 1]$ and adopt the proper initial points[1] $\boldsymbol{t}_0^{(0)}$, $\boldsymbol{b}_0^{(0)}$ and $\boldsymbol{r}_0^{(0)}$;
  while $t_i'^{*(\beta)}$, $b_i'^{*(\beta)}$ and $r_i'^{*(\beta)}$ are not a stationary solution **do**
    Solve $\mathcal{P}_1'$ to obtain the optimal solution $b_i'^{*(\beta+1)}$ and $r_i'^{*(\beta+1)}$;
    Update $\boldsymbol{t}_0$, $\boldsymbol{b}_0$ and $\boldsymbol{r}_0$ according to $\boldsymbol{t}_0^{(\beta+1)} = \boldsymbol{t}_0^{(\beta)} + \theta^{(\beta)} \cdot (\boldsymbol{t}'^{*(\beta+1)} - \boldsymbol{t}_0^{(\beta)})$, $\boldsymbol{b}_0^{(\beta+1)} = \boldsymbol{b}_0^{(\beta)} + \theta^{(\beta)} \cdot (\boldsymbol{b}'^{*(\beta+1)} - \boldsymbol{b}_0^{(\beta)})$ and $\boldsymbol{r}_0^{(\beta+1)} = \boldsymbol{r}_0^{(\beta)} + \theta^{(\beta)} \cdot (\boldsymbol{r}'^{*(\beta+1)} - \boldsymbol{r}_0^{(\beta)})$;
    Apply a diminishing step-size rule [42]: $\theta^{(\beta+1)} = \theta^{(\beta)}\left(1 - \delta\theta^{(\beta)}\right), \delta \in (0, 1)$;
    set $\beta \leftarrow \beta + 1$;
  end while
  return $b_i^* = b_i'^{*(\beta)}$ and $r_i^* = r_i'^{*(\beta)}$ for the worker i.
end procedure

$[n]$, the error-detecting redundancy obtained by WLTCC-ED is given as $r_i^* = 0$. In other words, only coded computation is needed, while error-detecting coding is not required in this situation. Moreover, if the transmission capability of each worker $\mu_i^{\rm trn}$ is the same and the bandwidth of wireless channel is unlimited, the data sub-block size obtained by WLTCC-ED is given as $b_i^* = 1$, which degenerates to the fine-grained LT coding approach.

Remark 4 (*Trade-off between Computation and Transmission*). WLTCC-ED realizes a trade-off between computation latency and transmission latency. When the transmission capability and channel condition of each worker are the same, i.e. $\mu_i^{\rm trn} = \mu^{\rm trn}, \varepsilon_i = \varepsilon, i \in [n]$, the workers with more powerful computation capability will complete more computational tasks. In other words, for worker i, the larger value of $\mu_i^{\rm cmp}$ and the smaller value of a_i will lead to the larger value of kb_i^* in WLTCC-ED. If the computation capability of each worker are the same, i.e. $\mu_i^{\rm cmp} = \mu^{\rm cmp}, a_i = a, i \in [n]$, the workers with more powerful transmission capability and better channel condition will complete more computational tasks. In other words, for worker i, the larger value of $\mu_i^{\rm trn}$ and the smaller value of ε_i will lead to the larger value of kb_i^* in WLTCC-ED.

B. Low-Complexity Algorithm

The iterative procedure in Alg. 1 may incur high computational complexity. To simplify it with a lower complexity, an approximate method is provided when the error rate is small.

First, in the scenario of low error rate, the corresponding approximate treatments are done for some terms in $\mathcal{P}_1$ as follows.

• *Approximation 1.* Since the channel condition is pretty good, i.e. the value of ε_i is very small, it does not need to add a large amount of error-detecting redundancy to meet the requirement of computation reliability for the whole networks. In other words, the value of r_i is also very small and satisfies $r_i \ll b_i$ for each worker. Thus, it is approximated that

$$\frac{b_i}{b_i + r_i} \approx 1. \tag{44}$$

• *Approximation 2.* According to [30], $p_{{\rm e},i}$ can be approximated by a weaker upper bound in the scenario of low error rate, i.e.

$$p_{{\rm e},i} \approx 2^{-qr_i}. \tag{45}$$

• *Approximation 3.* For the $(b_i + r_i, b_i)$ linear code, up to r_i error inner products can be detected [38]–[40], i.e. $p_{{\rm d},i} \le \sum_{j=1}^{r_i} C_{b_i+r_i}^{j} \varepsilon_{q,i}^{j} (1 - \varepsilon_{q,i})^{b_i+r_i-j}$. Approximating binomial distribution by Poisson distribution [43] and then using Stirling's approximation, we can obtain

$$\begin{aligned} p_{{\rm c},i} + p_{{\rm e},i} &= 1 - p_{{\rm d},i} \\ &\ge 1 - \sum_{j=1}^{r_i} C_{b_i+r_i}^{j} \varepsilon_{q,i}^{j} (1 - \varepsilon_{q,i})^{b_i+r_i-j} \\ &\overset{(a)}{\approx} 1 - \sum_{j=1}^{r_i} \frac{{\chi_i}^{j} e^{-\chi_i}}{j!} \ge 1 - r_i \frac{{\chi_i}^{\chi_i} e^{-\chi_i}}{\chi_i!} \\ &\overset{(b)}{\approx} 1 - r_i \frac{{\chi_i}^{\chi_i} e^{-\chi_i} e^{\chi_i}}{{\chi_i}^{\chi_i}\sqrt{2\pi\chi_i}} = 1 - \frac{r_i}{\sqrt{2\pi\left(b_i + r_i\right)\varepsilon_{q,i}}}, \end{aligned} \tag{46}$$

where $\chi_i = (b_i + r_i)\,\varepsilon_{q,i}$. The condition (a) represents the approximation between binomial distribution and Poisson distribution, while the condition (b) holds because of the Stirling's approximation.

Then, utilizing arithmetic means and geometric means (AM-GM) inequality, the bounds of PF structures in $\mathcal{P}_1$ are given as:

• Substitute Eq. (45) into the constraint (37) of $\mathcal{P}_1$, and utilize AM-GM inequality as follows:

$$\begin{aligned} &\left((1 - \varepsilon_{q,i})^{b_i+r_i} + 2^{-qr_i}\right) t_i \\ &\qquad \le \frac{1}{2}(1 - \varepsilon_{q,i})^{2(b_i+r_i)} + 2^{-1-2qr_i} + t_i^2; \end{aligned}$$

• Substitute Eq. (44) and Eq. (46) into the constraint (38) of $\mathcal{P}_1$, and utilize AM-GM inequality as follows:

$$\begin{aligned} &\left(1 - \frac{(b_i + r_i)^{-\frac{1}{2}} r_i}{\sqrt{2\pi\varepsilon_{q,i}}}\right) t_i \\ &\qquad \ge t_i - \frac{1}{3\sqrt{2\pi\varepsilon_{q,i}}}\left((b_i + r_i)^{-\frac{3}{2}} + r_i^3 + t_i^3\right); \end{aligned}$$

• Substitute Eq. (44) and Eq. (45) into the constraint (39) of $\mathcal{P}_1$, and utilize AM-GM inequality as follows:

$$2^{-qr_i} t_i \le 2^{-1-2qr_i} + \frac{1}{2} t_i^2.$$

At last, by replacing the original constraints in $\mathcal{P}_1$ with their tighter convex bounds, the approximate optimization problem for scenario of low error rate can be given as

$$\begin{aligned}\mathcal{P}_1'': \min_{t_{\rm cmp},\boldsymbol{t},\boldsymbol{b},\boldsymbol{r}} & t_{\rm cmp}+\sum_{i=1}^{n} t_i \\ \text{s.t.} \quad & 0\le b_i\le l/k, r_i\ge 0, \forall i\in[n] \\ & \mu_i^{\rm trn}\left(\frac{1}{2}(1-\varepsilon_{q,i})^{2(b_i+r_i)}+2^{-1-2qr_i}+t_i^2\right) \\ & \quad \le (b_i+r_i)\sum_{j=1}^{k} e^{-\frac{\mu_i^{\rm cmp}}{j(b_i+r_i)}[t_{\rm cmp}-j(b_i+r_i)a_i]}, \forall i\in[n] \\ & \sum_{i=1}^{n}\mu_i^{\rm trn}\left(t_i-\frac{(b_i+r_i)^{-\frac{3}{2}}+r_i^3+t_i^3}{3\sqrt{2\pi\varepsilon_{q,i}}}\right)\ge(1+\eta)\,m, \\ & \sum_{i=1}^{n}\mu_i^{\rm trn}\left(2^{-1-2qr_i}+\frac{1}{2}t_i^2\right)\le(1+\eta)\,mp_{\rm r}.\end{aligned}$$

Since the objective function and the constraints in $\mathcal{P}_1''$ are composed of the sum of convex functions, this approximate optimization problem is also convex. Based on the Lagrange function with Karush–Kuhn–Tucker (KKT) conditions [44], we can get the optimal computation time as

$$t_{\rm cmp}^{*}=\frac{(1+\eta)\,m}{\sum\limits_{i=1}^{n} h_i/k\gamma_i\cdot\mathbb{1}_{\{\lambda_i\}}}, \tag{47}$$

where $h_i=\sum_{j=1}^{k}\left(1-e^{-\frac{\mu_i^{\rm cmp}k\gamma_i}{j}+\mu_i^{\rm cmp}a_i}\right)$, γ_i is the positive solution to the equation

$$\sum_{j=1}^{k}\left(1+\frac{\mu_i^{\rm cmp}k\gamma_i}{j}\right)e^{-\frac{\mu_i^{\rm cmp}k\gamma_i}{j}}=ke^{-\mu_i^{\rm cmp}a_i}, \tag{48}$$

and

$$\lambda_i=(1-\varepsilon_{q,i})\,\mu_i^{\rm trn}+\sum_{u=1}^{n}\left(\frac{(1-\varepsilon_{q,i})\,\mu_i^{\rm trn}}{(1-\varepsilon_{q,u})\,\mu_u^{\rm trn}}-1\right)\frac{h_u}{k\gamma_u} \tag{49}$$

is the straggling factor that indicates whether the worker i is a straggler or not. Moreover, the optimal transmission time, error-detecting redundancy and data sub-block size can be obtained as follows:

$$t_i^{*}=\sqrt{\frac{2\,(1-p_{\rm r})\,t_{\rm cmp}^{*}h_i}{k\gamma_i\mu_i^{\rm trn}}-(1-\varepsilon_{q,i})^{\frac{2t_{\rm cmp}^{*}}{k\gamma_i}}}\cdot\mathbb{1}_{\{\lambda_i\}}, \tag{50}$$

$$r_i^{*}=\frac{-1-\log_2\left(\frac{(2p_{\rm r}-1)t_{\rm cmp}^{*}h_i}{k\gamma_i\mu_i^{\rm trn}}+\frac{(1-\varepsilon_{q,i})^{\frac{2t_{\rm cmp}^{*}}{k\gamma_i}}}{2}\right)}{2q}\cdot\mathbb{1}_{\{\lambda_i\}}, \tag{51}$$

$$b_i^{*}=\left(\frac{t_{\rm cmp}^{*}}{k\gamma_i}-r_i^{*}\right)\cdot\mathbb{1}_{\{\lambda_i\}}. \tag{52}$$

Then, the wireless LT coded computation with error detection in the scenario of low error rate (WLTCC-ED(LER)) is provided, which is given as Alg. 2.

Compared with SCA algorithm with iterations, Alg. 2 can be carried out in the constant time. It has low-complexity and can obtain approximate solutions faster in the scenario of low error rate.

Algorithm 2 Wireless LT Coded Computation with Error Detection in the scenario of Low Error Rate

Require: The parameter tuple $(\mu_i^{\rm cmp}, a_i, \mu_i^{\rm trn}, \varepsilon_i, p_{\rm r})$ for each worker $i\in[n]$.

Ensure: The data sub-block size b_i^{*} and error-detecting redundancy r_i^{*} for the worker i.

procedure WLTCC-ED(LER)
  for $i=1$ to n **do**
    Obtain γ_i in Eq. (48) and λ_i in Eq. (49) for the worker i;
    if $\lambda_i>0$ **then**
      The worker i is chosen;
    else
      The worker i is abandoned;
    end if
  end for
  Obtain the optimal $t_{\rm cmp}^{*}$ in Eq. (47);
  return b_i^{*} and r_i^{*} for the worker i according to Eq. (52) and Eq. (51).
end procedure

Remark 5 (*Stragglers Recognition*). There are not only computation stragglers with the poor computation capability, but also transmission stragglers with the weak transmission capability or bad channel condition. In Alg. 2, a worker can be decided as a straggler or not by $\lambda_i, i\in[n]$. When $\lambda_i\le 0$, worker i is a straggler, which implies that it will lead to a severe performance degradation for the whole networks. Thus, worker i will not compute or transmit any inner products, i.e. $b_i^{*}=0$.

VI. Simulation Results and Discussion

In this section, we will present some numerical results to show the performances of our proposed JCTC scheme.

Similar to [7], [13], [32], [33], we choose the number of rows in $\mathbf{A}$ as $m=5000$, the number of workers as $n=100$, the tolerable maximum error inner product rate as $p_{\rm r}=0.005$, and the maximum number of sub-blocks that can be transmitted by each worker as $k=4$. Also, we assume that $\alpha=2.8$ and $q=1$. The value of decoding parameter η in LT coding approach can be determined as $\eta=0.0326$ [15], which implies the master can recover the desired result successfully once receiving $(1+\eta)m=5163$ data inner products. For error detection, MDS code is applied and the encoding capability of workers in the separate designs is chosen as $a_i^{\rm cc}\sim\mathcal{U}(0.1,2)$, $\mu_i^{\rm cc}\sim\mathcal{U}(10,30)$, $i\in[n]$. The schemes studied are given as follows.

1) **UUA (Uniform Uncoded Allocation).** Computation and error detection are designed separately. Each worker is assigned the same number of rows and does not divide the local data block into sub-blocks, i.e., $l=m/n, b_i=l=m/n$ for $\forall i\in[n]$. The error-detecting redundancy r_i, $i\in[n]$ is obtained by [33, Eq. (15)];
2) **MG-MDSCA (Maximum-Grained MDS Coding Approach).** Coded computation and error detection are

TABLE I
PARAMETERS OF THREE SCENARIOS.

Scenario 1	Group 1: 20 workers	Group 2: 30 workers	Group 3: 40 workers	Group 4: 10 workers
	$a_i = 3$, $\mu_i^{\rm cmp} = 6$, $\mu_i^{\rm trn} = 1500$, $\varepsilon_i = 0$	$a_i = 6$, $\mu_i^{\rm cmp} = 10$, $\mu_i^{\rm trn} = 1500$, $\varepsilon_i = 0$	$a_i = 5$, $\mu_i^{\rm cmp} = 1$, $\mu_i^{\rm trn} = 1500$, $\varepsilon_i = 0$	$a_i = 12$, $\mu_i^{\rm cmp} = 2$, $\mu_i^{\rm trn} = 1500$, $\varepsilon_i = 0$
Scenario 2	Group 1: 20 workers	Group 2: 30 workers	Group 3: 40 workers	Group 4: 10 workers
	$a_i = 3$, $\mu_i^{\rm cmp} = 6$, $\mu_i^{\rm trn} = 60$, $\varepsilon_i = 0.03$	$a_i = 6$, $\mu_i^{\rm cmp} = 10$, $\mu_i^{\rm trn} = 2000$, $\varepsilon_i = 0.07$	$a_i = 5$, $\mu_i^{\rm cmp} = 1$, $\mu_i^{\rm trn} = 3000$, $\varepsilon_i = 0.11$	$a_i = 12$, $\mu_i^{\rm cmp} = 2$, $\mu_i^{\rm trn} = 500$, $\varepsilon_i = 0.01$
Scenario 3	100 workers			
	$a_i \sim \mathcal{U}(0.5, 12)$, $\mu_i^{\rm cmp} \sim \mathcal{U}(2, 20)$, $\mu_i^{\rm trn} \sim \mathcal{U}(100, 1200)$, $\varepsilon_i \sim \mathcal{U}(0.01, 0.1)$			
Scenario 4	Group 1: 12 workers	Group 2: 6 workers	Group 3: 12 workers	Group 4: 20 workers
	$a_i = a_i^{\rm cc} = 0.012$, $\mu_i^{\rm cmp} = \mu_i^{\rm cc} = 10.4907$, $\mu_i^{\rm trn} = 464.155$, $\varepsilon_i = 0.0076$	$a_i = a_i^{\rm cc} = 0.5178$, $\mu_i^{\rm cmp} = \mu_i^{\rm cc} = 3.8685$, $\mu_i^{\rm trn} = 153.77$, $\varepsilon_i = 0.0286$	$a_i = a_i^{\rm cc} = 0.1877$, $\mu_i^{\rm cmp} = \mu_i^{\rm cc} = 5.3052$, $\mu_i^{\rm trn} = 218.68$, $\varepsilon_i = 0.017$	$a_i = a_i^{\rm cc} = 0.0108$, $\mu_i^{\rm cmp} = \mu_i^{\rm cc} = 12.3772$, $\mu_i^{\rm trn} = 646.75$, $\varepsilon_i = 0.0055$

designed separately. Each worker is assigned the same number of rows with maximum-grained sub-block division [6] based on MDS code, i.e., l is obtained by setting the first derivative of [6, Eq. (11)] to zero and $b_i = l/k$. The error-detecting redundancy r_i, $i \in [n]$ is obtained by [33, Eq. (15)];

3) **MG-LTCA (Maximum-Grained LT Coding Approach).** Coded computation and error detection are designed separately. Each worker is assigned the same number of rows with maximum-grained sub-block division [9, Sec. 3.2] based on LT code, i.e., $l = \alpha m/n, b_i = \alpha m/kn$ for $\forall i \in [n]$. The error-detecting redundancy r_i, $i \in [n]$ is obtained by [33, Eq. (15)];
4) **BD-WLTCC (Block-Design Based Wireless LT Coded Computation).** Coded computation and error detection are designed separately. With the given $\{\mu_i^{\rm cmp}\}$, $\{a_i\}$, $\{\mu_i^{\rm trn}\}$, $\{\varepsilon_i\}$ and $p_{\rm r}$, each worker is assigned the data sub-block size b_i based on [15, Alg. 1], while the error-detecting redundancy r_i is obtained by [33, Eq. (15)], for $\forall i \in [n]$;
5) **WLTCC-ED(SCA).** Coded computation and error detection are designed jointly through JCTC scheme. With the given $\{\mu_i^{\rm cmp}\}$, $\{a_i\}$, $\{\mu_i^{\rm trn}\}$, $\{\varepsilon_i\}$ and $p_{\rm r}$, the data sub-block size b_i and error-detecting redundancy r_i of worker i can be obtained by Alg. 1, where latency is optimized using SCA;
6) **WLTCC-ED(LER).** Coded computation and error detection are designed jointly through JCTC scheme. With the given $\{\mu_i^{\rm cmp}\}$, $\{a_i\}$, $\{\mu_i^{\rm trn}\}$, $\{\varepsilon_i\}$ and $p_{\rm r}$, the data sub-block size b_i and error-detecting redundancy r_i of worker i can be obtained by Alg. 2, where latency is optimized using approximations.

In order to compare the performances of different schemes, four scenarios are considered as in Table I[2]. For Scenario 1, considering no transmission error, 100 workers are divided into four groups with the different computation capabilities and the same transmission capabilities, whereas each group in Scenario 2 has disparate computation capabilities, transmission capabilities and channel conditions. Scenario 3 is the case of heterogeneous wireless networks where the parameters of each worker are drawn from the corresponding random sources. Scenario 4 is based on a practical wireless distributed computing system with 50 workers. First, we observe the computation time, transmission time and channel conditions of these workers to get the statistics data. Then, through fitting the statistics data on computation and transmission time to the exponential model, we get the computation and transmission capabilities of workers. Also, the channel conditions can be obtained by using the reference signal. Finally, the workers can be divided into 4 groups as shown in Table I.

[2]The unit of $1/\mu_i^{\rm cmp}$, a_i, $1/\mu_i^{\rm cc}$ and $a_i^{\rm cc}$ is milliseconds per row, and the unit of $\mu_i^{\rm trn}$ is the number of inner products per millisecond.

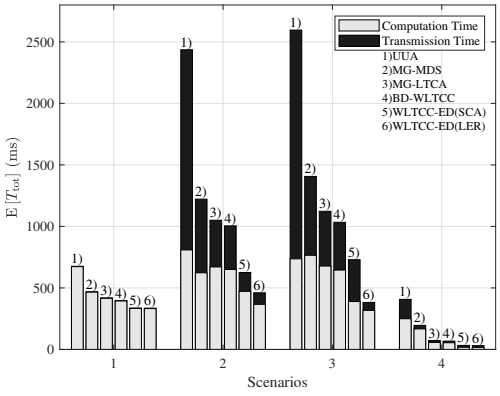


Fig. 7. Latency comparison between four separate designs and our JCTC schemes in four different scenarios, where $p_{\rm r} = 0.005$.

Performance comparisons in the above four scenarios between the implemented schemes are shown as Fig. 7 and Fig. 8 for latency and computation reliability, respectively. We can observe that WLTCC-ED(SCA) and WLTCC-ED(LER) can avoid encoding for error detection in workers and minimize the total latency to achieve a sub-optimal trade-off between computation and transmission latency compared with separate designs. For the computation reliability, WLTCC-ED(SCA) can always satisfy the required reliability but the low-complexity algorithm is only applicable to the scenario of low error rate, like Scenario 1 and Scenario 4, because of the approximations.

For Scenario 2, the performance changes over ε_i including latency and computation reliability are shown in Fig. 9 and Fig. 10. We can note that the total latency of our JCTC

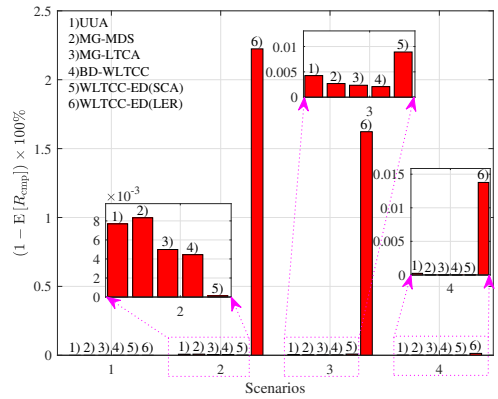


Fig. 8. Computation reliability comparison between four separate designs and our JCTC schemes in four different scenarios, where $p_{\mathrm{r}} = 0.005$.

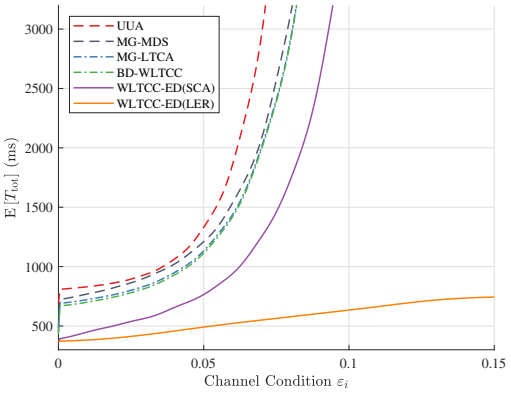


Fig. 9. The expected total latency $\mathbb{E}[T_{\mathrm{tot}}]$ versus channel condition ε_i, where $p_{\mathrm{r}} = 0.005$ and other parameters are chosen from Scenario 2.

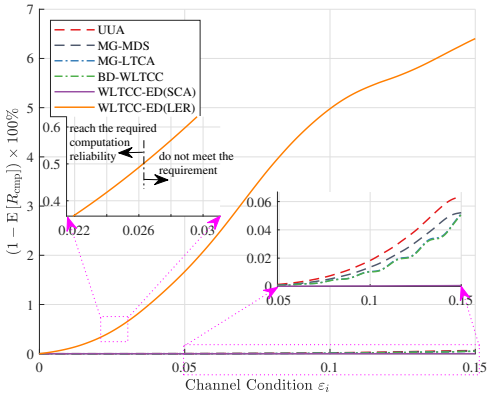


Fig. 10. The expected error inner product rate $1-\mathbb{E}[R_{\mathrm{cmp}}]$ versus channel condition ε_i, where $p_{\mathrm{r}} = 0.005$ and other parameters are chosen from Scenario 2.

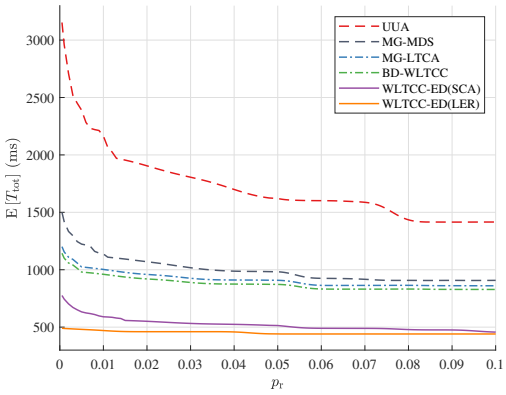


Fig. 11. The expected total latency $\mathbb{E}[T_{\mathrm{tot}}]$ versus the tolerable maximum error inner product rate p_{r}, where the parameters of workers are chosen from Scenario 2.

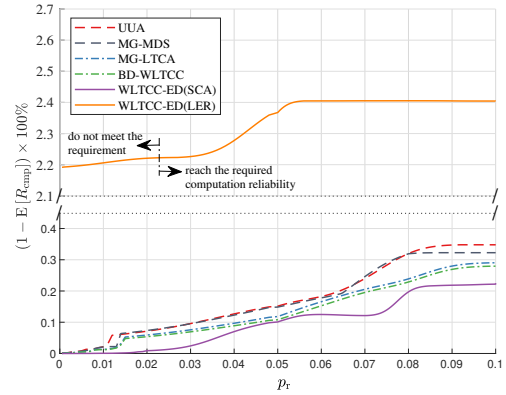


Fig. 12. The expected error inner product rate $1-\mathbb{E}[R_{\mathrm{cmp}}]$ versus the tolerable maximum error inner product rate p_{r}, where the parameters of workers are chosen from Scenario 2.

schemes is always lower than the separate designs regardless of the channel condition. Moreover, in order to guarantee the computation reliability, the implemented schemes needs lots of re-transmissions especially when the channel condition is not satisfactory, which makes them sensitive with ε_i. For computation reliability, WLTCC-ED(LER) does not satisfy the required computation reliability when the channel condition is bad, since some approximations for the scenario of low error rate are applied in this scheme. And other schemes keep the value of expected computation reliability above $1-p_{\mathrm{r}}$.

Fig. 11 and Fig. 12 show the effect of p_{r} on the performances from latency's and computation reliability's points of view respectively. The expected total latency will be reduced as the required computation reliability is decreased. Specially, if an extremely high computation reliability is required for a scenario with bad channel condition, the total latency may increase to infinity due to the constant re-transmission. From the reliability's point of view, we notice that the low-complexity algorithm cannot guarantee the final computational result with no error regardless of p_{r} due to the bad channel condition, and the expected computation reliability of WLTCC-ED(LER) is worse than other schemes because of the approximation. But

WLTCC-ED(LER) can also perform well in a scenario with the low requirement of computation reliability and good channel condition.

VII. Conclusion

In this paper, we have proposed the JCTC scheme to design coded computation and error detection jointly. Due to the two-layer encoding strategy, the low dynamic encoding has been achieved. Then, the performances of JCTC scheme, including latency and computation reliability, have been analyzed. Under the same computation reliability, theoretical performance comparisons with separate designs have shown the advantages of JCTC scheme from calculated amount's and latency's points of view. Finally, to achieve the efficient task and redundancy allocation, the WLTCC-ED algorithms have been presented based on both iterative and low-complexity methods. The simulation results have also verified the superiority of our proposed scheme.

Appendix A
Proof of Lemma 1

From Eq. (5), we can know that

$$x_i\left(t_{\mathrm{c}}\right)=\begin{cases}0, & \text{if } T_{i,1}^{\mathrm{cmp}}>t_{\mathrm{c}},\\ j, & \text{if } T_{i,j}^{\mathrm{cmp}}\le t_{\mathrm{c}}<T_{i,j+1}^{\mathrm{cmp}}, \forall j\in[k-1],\\ k, & \text{if } t_{\mathrm{c}}\ge T_{i,k}^{\mathrm{cmp}}.\end{cases}$$

Then, the expectation $\mathbb{E}\left[x_i\left(t_{\mathrm{c}}\right)\right]$ can be obtained by

$$\begin{aligned}\mathbb{E}\left[x_i\left(t_{\mathrm{c}}\right)\right]&=\sum_{j=0}^{k} j\times\Pr\left[x_i\left(t_{\mathrm{c}}\right)=j\right]\\ &=\sum_{j=1}^{k-1} j\times\Pr\left[T_{i,j}^{\mathrm{cmp}}\le t_{\mathrm{c}}<T_{i,j+1}^{\mathrm{cmp}}\right]+k\times\Pr\left[T_{i,k}^{\mathrm{cmp}}\le t_{\mathrm{c}}\right]\\ &=\sum_{j=1}^{k-1} j\times\left(\Pr\left[T_{i,j}^{\mathrm{cmp}}\le t_{\mathrm{c}}\right]-\Pr\left[T_{i,j+1}^{\mathrm{cmp}}\le t_{\mathrm{c}}\right]\right)\\ &\quad+k\times\Pr\left[T_{i,k}^{\mathrm{cmp}}\le t_{\mathrm{c}}\right]\\ &=\sum_{j=1}^{k}\Pr\left[T_{i,j}^{\mathrm{cmp}}\le t_{\mathrm{c}}\right]\\ &=\sum_{j=1}^{k}\left(1-e^{-\frac{\mu_i^{\mathrm{cmp}}}{j(b_i+r_i)}\left(t_{\mathrm{c}}-j(b_i+r_i)a_i\right)}\right).\end{aligned}$$

Appendix B
Proof of Lemma 2

When the computation capability tuple for all n workers is $\left(\mu_{\mathrm{g}}^{\mathrm{cmp}},a_{\mathrm{g}}\right)$, the best computational performance for the whole networks is achieved. It implies that the heterogeneous networks are reduced to the corresponding homogeneous networks with the best computational performance. Thus, we can obtain

$$\begin{aligned}T_{\mathrm{cmp}}&=\max_i T_i^{\mathrm{cmp}}=\max_i c_i\left(b_i+r_i\right)\left(\hat{T}_i^{\mathrm{cmp}}+a_i\right)\\ &\ge\left(\max_i c_i b_i+\min_i c_i r_i\right)\times\left(\min_i \hat{T}_i^{\mathrm{cmp}}+a_{\mathrm{g}}\right)\\ &\ge\frac{(1+\eta)\,m}{n}\left(\hat{T}_{(1)}^{\mathrm{cmp}}+a_{\mathrm{g}}\right).\end{aligned}$$

Appendix C
Proof of Lemma 3

When the computation capability tuple for all n workers is $(\mu_{\mathrm{b}}^{\mathrm{cmp}},a_{\mathrm{b}})$, the worst computational performance for the whole networks is achieved. It implies that the heterogeneous networks are reduced to the corresponding homogeneous networks with the worst computational performance. Thus, we can obtain

$$\begin{aligned}T_{\mathrm{cmp}}&=\max_i T_i^{\mathrm{cmp}}=\max_i c_i\left(b_i+r_i\right)\left(\hat{T}_i^{\mathrm{cmp}}+a_i\right)\\ &\le\left(c_w+1\right)\left(b_w+r_w\right)\left(\hat{T}_w^{\mathrm{cmp}}+a_{\mathrm{b}}\right), w\in W_{\mathrm{e}}.\end{aligned}$$

Summing over all $w\in W_{\mathrm{e}}$, we get

$$\begin{aligned}\sum_w T_{\mathrm{cmp}}&\le\sum_w\left(c_w+1\right)\left(b_w+r_w\right)\left(\hat{T}_w^{\mathrm{cmp}}+a_{\mathrm{b}}\right)\\ &\le\sum_w\left(\max_w c_w b_w+\max_w c_w r_w\right)\left(\hat{T}_w^{\mathrm{cmp}}+a_{\mathrm{b}}\right)\\ &\quad+\sum_w\left(\max_w b_w+\max_w r_w\right)\left(\hat{T}_w^{\mathrm{cmp}}+a_{\mathrm{b}}\right)\\ &=\left(\max_w c_w b_w+\max_w c_w r_w+\max_w b_w+\max_w r_w\right)\\ &\quad\times\sum_w\left(\hat{T}_w^{\mathrm{cmp}}+a_{\mathrm{b}}\right),\end{aligned}$$

$$\begin{aligned}T_{\mathrm{cmp}}&\le\left(\max_w c_w b_w+\max_w c_w r_w+\max_w b_w+\max_w r_w\right)\\ &\quad\times\left(\bar{\hat{T}}_w^{\mathrm{cmp}}+a_{\mathrm{b}}\right)\\ &\le\left(\frac{2\alpha m}{n}+\left(k+1\right)r_{\mathrm{m}}\right)\left(\bar{\hat{T}}_w^{\mathrm{cmp}}+a_{\mathrm{b}}\right).\end{aligned}$$

Appendix D
Proof of Lemma 4

With the certain $k_{\mathrm{re},i}=j$, $T_{i,\mathrm{re},(\kappa^{\mathrm{th}})}^{\mathrm{trn}}$ follows an erlang distribution with shape parameter j and rate parameter μ_i^{trn} according to the convolution formula. On the basis of the total probability theorem, we can get

$$\begin{aligned}&\Pr\left[T_{i,\mathrm{re},(\kappa^{\mathrm{th}})}^{\mathrm{trn}}\le t\right]\\ &=\sum_{j=1}^{\infty}\Pr\left[k_{\mathrm{re},i}=j\right]\Pr\left[T_{i,\mathrm{re},(\kappa^{\mathrm{th}})}^{\mathrm{trn}}\le t\,|k_{\mathrm{re},i}=j\right]\\ &=\sum_{j=1}^{\infty}p_{\mathrm{s},i}(1-p_{\mathrm{s},i})^{j-1}\left(1-\sum_{u=0}^{j-1}\frac{e^{-\mu_i^{\mathrm{trn}}t}(\mu_i^{\mathrm{trn}}t)^u}{u!}\right)\\ &=1-e^{-\mu_i^{\mathrm{trn}}t}\sum_{j=1}^{\infty}\sum_{u=0}^{j-1}p_{\mathrm{s},i}(1-p_{\mathrm{s},i})^{j-1}\frac{(\mu_i^{\mathrm{trn}}t)^u}{u!}\\ &=1-e^{-\mu_i^{\mathrm{trn}}t}\sum_{u=0}^{\infty}\sum_{j=u+1}^{\infty}p_{\mathrm{s},i}(1-p_{\mathrm{s},i})^{j-1}\frac{(\mu_i^{\mathrm{trn}}t)^u}{u!}\\ &=1-e^{-\mu_i^{\mathrm{trn}}t}\sum_{u=0}^{\infty}\frac{\left(\left(1-p_{\mathrm{s},i}\right)\mu_i^{\mathrm{trn}}t\right)^u}{u!}\\ &=1-e^{-p_{\mathrm{s},i}\mu_i^{\mathrm{trn}}t}.\end{aligned}$$

APPENDIX E
PROOF OF LEMMA 5

From Eq. (13) and Eq. (14), we notice that the random variable $T_i^{\mathrm{trn}}(t_{\mathrm{c}}) = \sum_{\kappa=1}^{(b_i+r_i)x_i(t_{\mathrm{c}})} T_{i,\mathrm{re},(\kappa^{\mathrm{th}})}^{\mathrm{trn}}$ follows an erlang distribution with shape parameter $(b_i + r_i)\, x_i(t_{\mathrm{c}})$ and rate parameter $p_{\mathrm{s},i}\mu_i^{\mathrm{trn}}$ with the given time t_{c} and fixed $x_i(t_{\mathrm{c}})$. Then, the CDF of $T_i^{\mathrm{trn}}(t_{\mathrm{c}})$ can be obtained through the total probability theorem as

$$\begin{aligned}
&\Pr[T_i^{\mathrm{trn}}(t_{\mathrm{c}}) \le t] \\
&= \sum_{j=1}^{k} \Pr\left[x_i(t_{\mathrm{c}}) = j\right] \Pr\left[T_i^{\mathrm{trn}}(t_{\mathrm{c}}) \le t \,|\, x_i(t_{\mathrm{c}}) = j\right] \\
&= \sum_{j=1}^{k-1} \left(\Pr\left[T_{i,j}^{\mathrm{cmp}} \le t_{\mathrm{c}}\right] - \Pr\left[T_{i,j+1}^{\mathrm{cmp}} \le t_{\mathrm{c}}\right]\right) \Pr\left[U_{i,j} \le t\right] + 1 \\
&\quad - \Pr\left[T_{i,1}^{\mathrm{cmp}} \le t_{\mathrm{c}}\right] + \Pr\left[T_{i,k}^{\mathrm{cmp}} \le t_{\mathrm{c}}\right] \Pr\left[U_{i,k} \le t\right], \qquad (53)
\end{aligned}$$

where $U_{i,j}$ is a random variable following an erlang distribution with shape parameter $j\,(b_i + r_i)$ and rate parameter $p_{\mathrm{s},i}\mu_i^{\mathrm{trn}}$. According to Eq. (5) and Eq. (53), the expectation of $T_i^{\mathrm{trn}}(t_{\mathrm{c}})$ can be gotten directly by the definition of the mean, i.e.,

$$\begin{aligned}
\mathbb{E}\left[T_i^{\mathrm{trn}}(t_{\mathrm{c}})\right] &= \int_0^{+\infty} t \frac{\partial \Pr\left[T_i^{\mathrm{trn}}(t_{\mathrm{c}}) \le t\right]}{\partial t} dt \\
&= \frac{b_i + r_i}{p_{\mathrm{s},i}\mu_i^{\mathrm{trn}}} \sum_{j=1}^{k} \left(1 - e^{-\frac{\mu_i^{\mathrm{cmp}}}{j(b_i+r_i)}(t_{\mathrm{c}} - j(b_i+r_i)a_i)}\right).
\end{aligned}$$

APPENDIX F
THE CONVEX APPROXIMATIONS OF DC AND PF STRUCTURES IN $\mathcal{P}_1$

In $\mathcal{P}_1$, we assume that

$$\begin{aligned}
f_{1,i}(t_i, b_i, r_i) &= (p_{\mathrm{c},i} + p_{\mathrm{e},i})\, t_i \mu_i^{\mathrm{trn}} \\
&= t_i \mu_i^{\mathrm{trn}} \left((1-\varepsilon_i)^{q(b_i+r_i)} + 2^{-qr_i} - (1-\varepsilon_i)^{qb_i} 2^{-qr_i}\right),
\end{aligned}$$

$$\begin{aligned}
f_{2,i}(t_i, b_i, r_i) &= -(p_{\mathrm{c},i} + p_{\mathrm{e},i}) \frac{b_i}{b_i + r_i} t_i \mu_i^{\mathrm{trn}} \\
&= -\, b_i t_i \mu_i^{\mathrm{trn}} (b_i + r_i)^{-1} \\
&\quad \times \left((1-\varepsilon_i)^{q(b_i+r_i)} + 2^{-qr_i} - (1-\varepsilon_i)^{qb_i} 2^{-qr_i}\right),
\end{aligned}$$

$$\begin{aligned}
f_{3,i}(t_i, b_i, r_i) &= p_{\mathrm{e},i} \frac{b_i}{b_i + r_i} t_i \mu_i^{\mathrm{trn}} \\
&= b_i t_i \mu_i^{\mathrm{trn}} (b_i + r_i)^{-1} 2^{-qr_i} \left(1 - (1-\varepsilon_i)^{qb_i}\right).
\end{aligned}$$

For the DC structures in $f_{1,i}$, $f_{2,i}$ and $f_{3,i}$, we set

$$\begin{aligned}
f_{1,i}^{\mathrm{DC}}(b_i, r_i) &= (1-\varepsilon_i)^{qb_i} 2^{-qr_i}, \\
f_{2,i}^{\mathrm{DC}}(b_i, r_i) &= (1-\varepsilon_i)^{qb_i} 2^{-qr_i} (b_i + r_i)^{-1}.
\end{aligned}$$

Then we can linearize $f_{1,i}^{\mathrm{DC}}$ and $f_{2,i}^{\mathrm{DC}}$ with the given point b_0 and r_0 as follows:

$$\begin{aligned}
f_{1,i}^{\mathrm{DC}} &\approx f_{1,i|[0]}^{\mathrm{DC}} + f_{1,i|[b_0]}^{\mathrm{DC}} \cdot (b_i - b_0) + f_{1,i|[r_0]}^{\mathrm{DC}} \cdot (r_i - r_0) \\
&= f_{l_{1,i}}(b_i, r_i), \\
f_{2,i}^{\mathrm{DC}} &\approx f_{2,i|[0]}^{\mathrm{DC}} + f_{2,i|[b_0]}^{\mathrm{DC}} \cdot (b_i - b_0) + f_{2,i|[r_0]}^{\mathrm{DC}} \cdot (r_i - r_0) \\
&= f_{l_{2,i}}(b_i, r_i).
\end{aligned}$$

For the PF structures in $f_{1,i}$, $f_{2,i}$ and $f_{3,i}$, we can rewrite as follows:

$$\begin{aligned}
b_i t_i &= \frac{1}{2}(b_i + t_i)^2 - \frac{1}{2}\left(b_i^2 + t_i^2\right) \\
&\approx \frac{1}{2}(b_i + t_i)^2 - \frac{1}{2}\left(b_0^2 + t_0^2\right) - t_0(t_i - t_0) - b_0(b_i - b_0) \\
&= f_{p_{1,i}}(t_i, b_i),
\end{aligned}$$

with the given point t_0.

Then, according to [41, Section IV-B], $f_{1,i}(t_i, b_i, r_i)$, $f_{2,i}(t_i, b_i, r_i)$ and $f_{3,i}(t_i, b_i, r_i)$ can be written as the following convex functions:

$$\begin{aligned}
f_{1,i} \approx& \left((1-\varepsilon_i)^{q(b_i+r_i)} + 2^{-qr_i} - f_{l_{1,i}}\right) t_i \mu_i^{\mathrm{trn}} \\
\approx& \frac{\mu_i^{\mathrm{trn}}}{2}\left((1-\varepsilon_i)^{q(b_i+r_i)} + 2^{-qr_i} - f_{l_{1,i}} + t_i\right)^2 \\
&- \frac{\mu_i^{\mathrm{trn}}}{2} f_{p_{2,i}}, \\
f_{2,i} \approx& -\mu_i^{\mathrm{trn}} f_{p_{1,i}} \\
&\times \left((b_i + r_i)^{-1}\left((1-\varepsilon_i)^{q(b_i+r_i)} + 2^{-qr_i}\right) - f_{l_{2,i}}\right) \\
\approx& \frac{\mu_i^{\mathrm{trn}}}{2}\left((b_i + r_i)^{-1}\left((1-\varepsilon_i)^{q(b_i+r_i)} + 2^{-qr_i}\right) - f_{l_{2,i}}\right)^2 \\
&+ \frac{\mu_i^{\mathrm{trn}}}{2} f_{p_{1,i}}^2 - \frac{\mu_i^{\mathrm{trn}}}{2} f_{p_{3,i}}, \\
f_{3,i} \approx& \mu_i^{\mathrm{trn}}\left(2^{-qr_i}(b_i + r_i)^{-1} - f_{l_{2,i}}\right) f_{p_{1,i}} \\
\approx& \frac{\mu_i^{\mathrm{trn}}}{2}\left(2^{-qr_i}(b_i + r_i)^{-1} - f_{l_{2,i}} + f_{p_{1,i}}\right)^2 - \frac{\mu_i^{\mathrm{trn}}}{2} f_{p_{4,i}},
\end{aligned}$$

where $f_{p_{2,i}}(t_i, b_i, r_i)$, $f_{p_{3,i}}(t_i, b_i, r_i)$ and $f_{p_{4,i}}(t_i, b_i, r_i)$ can be obtained by

$$\begin{aligned}
f_{p_{2,i}} =& f_{2,i|[0]}^{\mathrm{PF}} + f_{2,i|[t_0]}^{\mathrm{PF}} \cdot (t_i - t_0) + f_{2,i|[b_0]}^{\mathrm{PF}} \cdot (b_i - b_0) \\
&+ f_{2,i|[r_0]}^{\mathrm{PF}} \cdot (r_i - r_0), \\
f_{p_{3,i}} =& f_{3,i|[0]}^{\mathrm{PF}} + f_{3,i|[t_0]}^{\mathrm{PF}} \cdot (t_i - t_0) + f_{3,i|[b_0]}^{\mathrm{PF}} \cdot (b_i - b_0) \\
&+ f_{3,i|[r_0]}^{\mathrm{PF}} \cdot (r_i - r_0), \\
f_{p_{4,i}} =& f_{4,i|[0]}^{\mathrm{PF}} + f_{4,i|[t_0]}^{\mathrm{PF}} \cdot (t_i - t_0) + f_{4,i|[b_0]}^{\mathrm{PF}} \cdot (b_i - b_0) \\
&+ f_{4,i|[r_0]}^{\mathrm{PF}} \cdot (r_i - r_0),
\end{aligned}$$

and functions $f_{2,i}^{\mathrm{PF}}(t_i, b_i, r_i)$, $f_{3,i}^{\mathrm{PF}}(t_i, b_i, r_i)$, $f_{4,i}^{\mathrm{PF}}(t_i, b_i, r_i)$ are given as follows:

$$\begin{aligned}
f_{2,i}^{\mathrm{PF}} =& \left((1-\varepsilon_i)^{q(b_i+r_i)} + 2^{-qr_i} - f_{l_{1,i}}\right)^2 + t_i^2, \\
f_{3,i}^{\mathrm{PF}} =& \left((b_i + r_i)^{-1}\left((1-\varepsilon_i)^{q(b_i+r_i)} + 2^{-qr_i} - f_{l_{2,i}}\right) + f_{p_{1,i}}\right)^2, \\
f_{4,i}^{\mathrm{PF}} =& \left(2^{-qr_i}(b_i + r_i)^{-1} - f_{l_{2,i}}\right)^2 + \left(f_{p_{1,i}}\right)^2.
\end{aligned}$$

Note that $f_{l_{1,i}}$, $f_{l_{2,i}}$, $f_{p_{2,i}}$, $f_{p_{3,i}}$, $f_{p_{4,i}}$ are composed of linear functions with respect to t_i, b_i and r_i respectively and

$f_{p_{1,i}}$ is also a convex function with respect to t_i and b_i. According to the conclusion on the convexity of composite functions [44], we can know that $f_{1,i}(t_i, b_i, r_i)$, $f_{2,i}(t_i, b_i, r_i)$ and $f_{3,i}(t_i, b_i, r_i)$ are also convex so that $\mathcal{P}_1$ can be relaxed to the convex problem $\mathcal{P}_1'$.

REFERENCES

[1] F. Li, J. Chen, and Z. Wang, "Wireless mapreduce distributed computing," *IEEE Trans. Inf. Theory*, vol. 65, no. 10, pp. 6101–6114, 2019.

[2] Y. He, J. Ren, G. Yu, and Y. Cai, "D2D communications meet mobile edge computing for enhanced computation capacity in cellular networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1750–1763, 2019.

[3] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, 2017.

[4] C. You, Y. Zeng, R. Zhang, and K. Huang, "Asynchronous mobile-edge computation offloading: Energy-efficient resource management," *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7590–7605, 2018.

[5] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, 2020.

[6] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1514–1529, 2017.

[7] A. Reisizadeh, S. Prakash, R. Pedarsani, and A. S. Avestimehr, "Coded computation over heterogeneous clusters," *IEEE Trans. Inf. Theory*, vol. 65, no. 7, pp. 4227–4242, 2019.

[8] B. Wang, J. Xie, K. Lu, Y. Wan, and S. Fu, "On batch-processing based coded computing for heterogeneous distributed computing systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2438–2454, 2021.

[9] A. Mallick, M. Chaudhari, U. Sheth, G. Palanikumar, and G. Joshi, "Rateless codes for near-perfect load balancing in distributed matrix-vector multiplication," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 3, no. 3, dec 2019.

[10] A. Severinson, A. Graell i Amat, and E. Rosnes, "Block-diagonal and LT codes for distributed computing with straggling servers," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 1739–1753, 2019.

[11] A. Reisizadeh and R. Pedarsani, "Latency analysis of coded computation schemes over wireless networks," in *Annu. Allerton Conf. Commun., Control, Comput.*, Oct 2017, pp. 1256–1263.

[12] D.-J. Han, J.-Y. Sohn, and J. Moon, "Coded wireless distributed computing with packet losses and retransmissions," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 8204–8217, Dec 2021.

[13] F. Wu and L. Chen, "Latency optimization for coded computation straggled by wireless transmission," *IEEE Wireless Commun. Lett.*, vol. 9, no. 7, pp. 1124–1128, July 2020.

[14] L. Chen, K. Han, Y. Du, and Z. Wang, "Block-division-based wireless coded computation," *IEEE Wireless Commun. Lett.*, vol. 11, no. 2, pp. 283–287, Feb 2022.

[15] B. Fang, K. Han, Z. Wang, and L. Chen, "Latency optimization for luby transform coded computation in wireless networks," *IEEE Wireless Commun. Lett.*, vol. 12, no. 2, pp. 197–201, Feb 2023.

[16] R. Bitar, P. Parag, and S. El Rouayheb, "Minimizing latency for secure coded computing using secret sharing via staircase codes," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4609–4619, 2020.

[17] R. Bitar, M. Xhemrishi, and A. Wachter-Zeh, "Adaptive private distributed matrix multiplication," *IEEE Trans. Inf. Theory*, vol. 68, no. 4, pp. 2653–2673, 2022.

[18] N. Mital, C. Ling, and D. Gündüz, "Secure distributed matrix computation with discrete fourier transform," *IEEE Trans. Inf. Theory*, vol. 68, no. 7, pp. 4666–4680, 2022.

[19] S. Dutta, V. Cadambe, and P. Grover, "Coded convolution for parallel and distributed computing within a deadline," in *IEEE Int. Symp. Inf. Theor. Proc.*, 2017, pp. 2403–2407.

[20] E. Ozfatura, S. Ulukus, and D. Gündüz, "Coded distributed computing with partial recovery," *IEEE Trans. Inf. Theory*, vol. 68, no. 3, pp. 1945–1959, 2022.

[21] T. Jahani-Nezhad and M. A. Maddah-Ali, "Berrut approximated coded computing: Straggler resistance beyond polynomial computing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 111–122, 2023.

[22] ——, "Codedsketch: A coding scheme for distributed computation of approximated matrix multiplication," *IEEE Trans. Inf. Theory*, vol. 67, no. 6, pp. 4185–4196, 2021.

[23] H. Jeong, A. Devulapalli, V. R. Cadambe, and F. P. Calmon, "ε-approximate coded matrix multiplication is nearly twice as efficient as exact multiplication," *IEEE J. Sel. Area. Inf. Theory*, vol. 2, no. 3, pp. 845–854, 2021.

[24] S. Kiani and S. C. Draper, "Successive approximation coding for distributed matrix multiplication," *IEEE J. Sel. Area. Inf. Theory*, vol. 3, no. 2, pp. 286–305, 2022.

[25] H. Zhu, L. Chen, N. Zhao, Y. Chen, W. Wang, and F. R. Yu, "Hierarchical coded matrix multiplication in heterogeneous multihop networks," *IEEE Trans. Commun.*, vol. 70, no. 6, pp. 3597–3612, 2022.

[26] Y. Sun, F. Zhang, J. Zhao, S. Zhou, Z. Niu, and D. Gündüz, "Coded computation across shared heterogeneous workers with communication delay," *IEEE Trans. Signal Process.*, vol. 70, pp. 3371–3385, 2022.

[27] K. Li, M. Tao, J. Zhang, and O. Simeone, "Coded computing and cooperative transmission for wireless distributed matrix multiplication," *IEEE Trans. Commun.*, vol. 69, no. 4, pp. 2224–2239, 2021.

[28] N. Liu, K. Li, and M. Tao, "Code design and latency analysis of distributed matrix multiplication with straggling servers in fading channels," *China Commun.*, vol. 18, no. 10, pp. 15–29, 2021.

[29] X. He, T. Li, R. Jin, and H. Dai, "Delay-optimal coded offloading for distributed edge computing in fading environments," *IEEE Trans. Wireless Commun.*, vol. 21, no. 12, pp. 10 796–10 808, 2022.

[30] S. Lin, D. J. Costello, and M. J. Miller, "Automatic-repeat-request error-control schemes," *IEEE Commun. Mag.*, vol. 22, no. 12, pp. 5–17, 1984.

[31] M. Luby, "LT codes," in *Annu. Symp. Found. Comput. Sci. Proc.*, Nov 2002, pp. 271–280.

[32] J. Martins and J. Alves, "ARQ protocols with adaptive block size perform better over a wide range of bit error rates," *IEEE Trans. Commun.*, vol. 38, no. 6, pp. 737–739, June 1990.

[33] M. Rice and S. Wicker, "Adaptive error control for slowly varying channels," *IEEE Trans. Commun.*, vol. 42, no. 234, pp. 917–926, February 1994.

[34] Y. Afoudi, M. Lazaar, and M. Al Achhab, "Collaborative filtering recommender system," in *Adv. Intell. Sys. Comput.*, M. Ezziyyani, Ed. Cham: Springer International Publishing, 2019, pp. 332–345.

[35] R. R. Müller, H. Rosenberger, and M. Reichenbach, "Linear computation coding for convolutional neural networks," in *IEEE Workshop Stat. Signal Process. Proc.*, 2023, pp. 562–565.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Adv. neural inf. proces. syst.*, 2017.

[37] H. Park, K. Lee, J.-Y. Sohn, C. Suh, and J. Moon, "Hierarchical coding for distributed computing," in *IEEE Int. Symp. Inf. Theor. Proc.*, 2018, pp. 1630–1634.

[38] J. Wolf, A. Michelson, and A. Levesque, "On the probability of undetected error for linear block codes," *IEEE Trans. Commun.*, vol. 30, no. 2, pp. 317–325, 1982.

[39] Y.-M. Wang and S. Lin, "A modified selective-repeat type-II hybrid ARQ system and its performance analysis," *IEEE Trans. Commun.*, vol. 31, no. 5, pp. 593–608, 1983.

[40] T. Kasami and S. Lin, "On the probability of undetected error for the maximum distance separable codes," *IEEE Trans. Commun.*, vol. 32, no. 9, pp. 998–1006, 1984.

[41] Z. Yu, Y. Gong, S. Gong, and Y. Guo, "Joint task offloading and resource allocation in UAV-enabled mobile edge computing," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3147–3159, 2020.

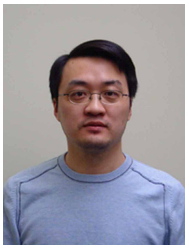
[42] G. Scutari, F. Facchinei, and L. Lampariello, "Parallel and distributed methods for constrained nonconvex optimization—part I: Theory," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1929–1944, April 2017.

[43] H. Poor, "The maximum difference between the binomial and poisson distributions," *Stat. Probabil. Lett.*, vol. 11, no. 2, pp. 103–106, 1991. [Online]. Available: https://www.sciencedirect.com/science/article/pii/016771529190125B

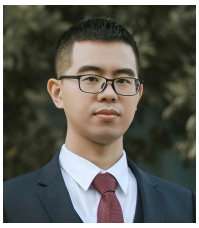
[44] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

Borui Fang received the B.E. degree in communication engineering from Dalian Maritime University, Dalian, China, in 2020. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, China. His research interests include coded distributed computing, wireless networks, and integrated communication and computation.

Li Chen (Senior Member, IEEE) received the B.E. degree in electrical and information engineering from the Harbin Institute of Technology, Harbin, China, in 2009, and the Ph.D. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 2014. He is currently an Associate Professor with the Department of Electronic Engineering and Information Science, University of Science and Technology of China. His research interests include integrated communication and computation, integrated sensing and communication, and wireless IoT networks.



Yunfei Chen (Senior Member, IEEE) received his B.E. and M.E. degrees in electronics engineering from Shanghai Jiaotong University, Shanghai, P.R.China, in 1998 and 2001, respectively. He received his Ph.D. degree from the University of Alberta in 2006. He is currently working as a Professor in the Department of Engineering at the University of Durham, U.K. His research interests include wireless communications, performance analysis, joint radar communications designs.



Changsheng You (Member, IEEE) received the B.Eng. degree from the University of Science and Technology of China (USTC) in 2014 and the Ph.D. degree from The University of Hong Kong (HKU) in 2018.

He was a Research Fellow with the National University of Singapore (NUS). He is currently an Assistant Professor with the Southern University of Science and Technology. His research interests include intelligent reflecting surface, UAV communications, edge learning, and mobile-edge computing. He received the IEEE Communications Society Asia-Pacific Region Outstanding Paper Award in 2019, the IEEE ComSoc Best Survey Paper Award in 2021, and the IEEE ComSoc Best Tutorial Paper Award in 2023. He is listed as a Highly Cited Chinese Researcher and an Exemplary Reviewer of the IEEE Transactions on Communications and IEEE Transactions on Wireless Communications. He is an Editor of IEEE Communications Letters, IEEE Transactions on Green Communications and Networking, and IEEE Open Journal of the Communications Society.

Xiaohui Chen (Member, IEEE) received the B.S. and M.S. degrees in communication and information engineering from the University of Science and Technology of China (USTC), Hefei, China, in 1998 and 2004, respectively. He is currently an Associate Professor with the Department of Electronic Engineering and Information Science, USTC. His current research interests include wireless network QoS, mobile computing, and AI-based communication.

Weidong Wang received the B.S. degree from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 1989, and the M.S. degree from the University of Science and Technology of China, Hefei, China, in 1993. He is currently a Full Professor with the Department of Electronic Engineering and Information Science, University of Science and Technology of China. His research interests include wireless communication, microwave and millimeter-wave, and radar technology. He is a member of the Committee of Optoelectronic Technology, Chinese Society of Astronautics.