Distributed Linearly Separable Computation

Kai Wan, *Member, IEEE,* Hua Sun, *Member, IEEE,* Mingyue Ji, *Member, IEEE,* and Giuseppe Caire, *Fellow, IEEE*

***Abstract*—This paper formulates a distributed computation problem, where a master asks N distributed workers to compute a linearly separable function. The task function can be expressed as K_c linear combinations of K messages, where each message is a function of one dataset. Our objective is to find the optimal tradeoff between the computation cost (number of uncoded datasets assigned to each worker) and the communication cost (number of symbols the master must download), such that from the answers of any N_r out of N workers the master can recover the task function with high probability, where the coefficients of the K_c linear combinations are uniformly i.i.d. over some large enough finite field. The formulated problem can be seen as a generalized version of some existing problems, such as distributed gradient coding and distributed linear transform.**

In this paper, we consider the specific case where the computation cost is minimum, and propose novel achievability schemes and converse bounds for the optimal communication cost. Achievability and converse bounds coincide for some system parameters; when they do not match, we prove that the achievable distributed computing scheme is optimal under the constraint of a widely used 'cyclic assignment' scheme on the datasets. Our results also show that when $\mathsf{K} = \mathsf{N}$, with the same communication cost as the optimal distributed gradient coding scheme proposed by Tandon *et al*. from which the master recovers one linear combination of K messages, our proposed scheme can let the master recover any additional $\mathsf{N}_\mathrm{r} - 1$ independent linear combinations of messages with high probability.

***Index Terms*—Distributed computation; linearly separable function; cyclic assignment**

I. Introduction

Enabling large-scale computations for a large dimension of data, distributed computation systems such as MapReduce [1] and Spark [2] have received significant attention in recent years [3]. The distributed computation system divides a computational task into several subtasks, which are then assigned to some distributed workers. This reduces significantly the computing time by exploiting parallel computing procedures and thus enables handling of the computations over large-scale big data. However, while large scale distributed computing schemes have the potential for achieving unprecedented levels of accuracy and providing dramatic insights into complex phenomena, they also present some technical issues/bottlenecks. First, due to the presence of stragglers, a subset of workers may take excessively long time or fail to return their computed sub-tasks, which leads to an undesirable and unpredictable latency. Second, data and computed results should be communicated among the master who wants to compute the task, and the workers. If the communication bandwidth is limited, the communication cost becomes another bottleneck of the distributed computation system. In order to tackle these two bottlenecks, coding techniques were introduced to the distributed computing algorithms [4]–[6], with the purpose of increasing tolerance with respect to stragglers and reducing the master-workers communication cost. More precisely, for the first bottleneck, using ideas similar to Minimum Distance Separable (MDS) codes, the master can recover the task function from the answers of the fastest workers. For the second bottleneck, inspired by concepts from coded caching networks [7], [8], network coding techniques are used to save significant communication cost exchanged in the network.

In this paper, a master aims to compute a linearly separable function f (such as linear MapReduce, Fourier Transform, convolution, etc.) on K datasets $(D_1, \dots, D_\mathsf{K})$, which can be written as

$$f(D_1, \dots, D_\mathsf{K}) = g\big(f_1(D_1), \dots, f_\mathsf{K}(D_\mathsf{K})\big) = g(W_1, \dots, W_\mathsf{K}).$$

$W_k = f_k(D_k)$ for all $k \in \{1, \dots, \mathsf{K}\}$ is the outcome of the component function $f_k(\cdot)$ applied to dataset D_k, and it is represented as a string of L symbols on an appropriate sufficiently large alphabet. For example, W_k can be the intermediate value in linear MapReduce, an input signal in Fourier Transform, etc. We consider the case where $g(\cdot)$ is a linear map defined by K_c linear combinations of the messages $W_1, \dots, W_\mathsf{K}$ with uniform i.i.d. coefficients over some large enough finite field; i.e., $g(W_1, \dots, W_\mathsf{K})$ can be seen as the matrix product $\mathbf{FW}$, where $\mathbf{F}$ is the coefficient matrix and $\mathbf{W} = [W_1; \dots; W_\mathsf{K}]$.[1] We consider the distributed computation scenario, where $f(D_1, \dots, D_\mathsf{K})$ is computed in a distributed way by a group of N workers. Each dataset is assigned in an uncoded manner to a subset of workers and the number of datasets assigned to each worker cannot be larger than M,

[1] As matrix multiplication is one of the key building blocks underlying many data analytics, machine learning algorithms and engineering problems, the considered model also has potential applications in those areas, where $f_1, \dots, f_\mathsf{K}$ represent the pretreatment of the datasets. For example, each dataset D_k where $k \in \{1, \dots, \mathsf{K}\}$ represents a raw dataset and needs to be processed through some filters, where W_k represents the filtered dataset of D_k. For the sake of linear transforms (e.g., Wavelet Transform, Discrete Fourier Transform), we need to compute multiple linear combinations of the filtered datasets, which can be expressed as $g(W_1, \dots, W_\mathsf{K})$. For another example, $D_1, \dots, D_\mathsf{K}$ are the K "input channels" of a Convolutional Neural Networks (CNN) stage. Each input channel D_k where $k \in \{1, \dots, \mathsf{K}\}$ is filtered individually by a convolution operation yielding W_k. Then the convolutions are linearly mixed by the coefficients of $g(W_1, \dots, W_\mathsf{K})$ producing K_c new layers in the feature space. Moreover, if $\mathbf{F}$ represents a MIMO precoding matrix, our considered model can also be used in the MIMO systems.

K. Wan and G. Caire are with the Electrical Engineering and Computer Science Department, Technische Universität Berlin, 10587 Berlin, Germany (e-mail: kai.wan@tu-berlin.de; caire@tu-berlin.de). The work of K. Wan and G. Caire was partially funded by the European Research Council under the ERC Advanced Grant N. 789190, CARENET.

H. Sun is with the Department of Electrical Engineering, University of North Texas, Denton, TX 76203, USA (email: hua.sun@unt.edu). The work of Hua Sun was supported in part by funding from NSF grants CCF-2007108 and CCF-2045656.

M. Ji is with the Electrical and Computer Engineering Department, University of Utah, Salt Lake City, UT 84112, USA (e-mail: mingyue.ji@utah.edu). The work of M. Ji was supported in part by NSF Awards 1817154 and 1824558.

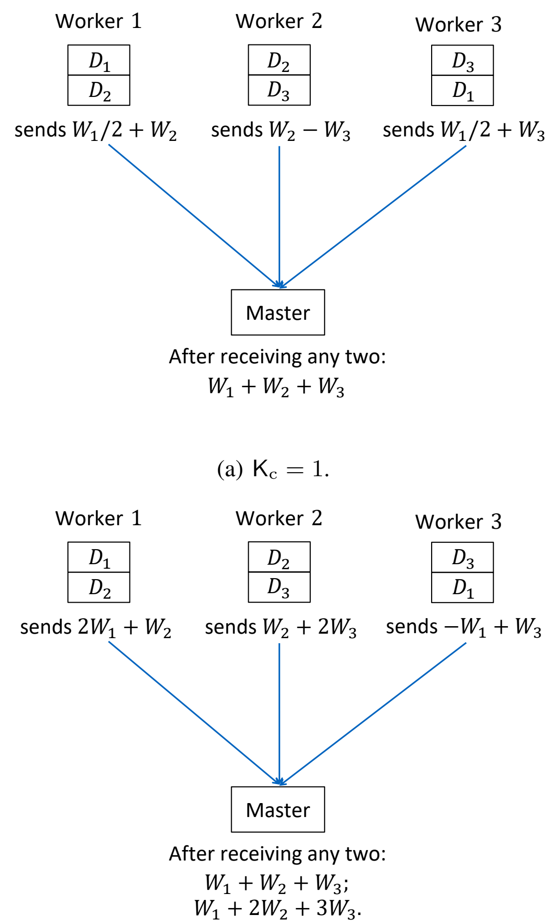
which is referred to as the computation cost.[2] Each worker should compute and send coded messages in terms of the datasets assigned to it, such that from the answers of any $\mathsf{N_r}$ workers, the master can recover the task function with high probability. Given $(\mathsf{K}, \mathsf{N}, \mathsf{N_r}, \mathsf{K_c}, \mathsf{M})$, we aim to find the optimal distributed computing scheme with *data assignment*, *computing*, and *decoding* phases, which leads to the minimum communication cost (i.e., the number of downloaded symbols by the master, normalized by L).

We illustrate two examples of the formulated distributed scenario in Fig. 1 where $\mathsf{K_c} = 1$ and $\mathsf{K_c} = 2$, respectively. In both examples, we consider that $\mathsf{K} = \mathsf{N} = 3, \mathsf{N_r} = 2$, and that the number of datasets assigned to each worker is $\mathsf{M} = 2$. Assume that the characteristic of $\mathbb{F}_\mathsf{q}$ is larger than 3.

- When $\mathsf{K_c} = 1$, the considered problem (as shown in Fig. 1a) is equivalent to the distributed gradient coding problem in [9], which aims to compute the sum of gradients in learning tasks by distributed workers. The gradient coding proposed in [9] assigns the datasets to the workers in a cyclic way, where D_1 and D_2 are assigned to worker 1, D_2 and D_3 are assigned to worker 2, and D_3 and D_1 are assigned to worker 3. Worker 1 then computes and sends $\frac{W_1}{2} + W_2$. Worker 2 sends $W_2 - W_3$, and worker 3 sends $\frac{W_1}{2} + W_3$. From any two sent coded messages, the master can recover the task function $W_1 + W_2 + W_3$. By the converse bound in [10], it can be proved that the gradient coding scheme [9] is optimal under the constraint of linear coding in terms of communication cost. Note that in our paper, from a novel converse bound, we prove the optimality of the gradient coding scheme [9] when $\mathsf{K_c} = 1$ by removing the constraint of linear coding.
- When $\mathsf{K_c} = 2$, besides $W_1 + W_2 + W_3$ we let the master also request another linear combination of the messages, e.g., $W_1 + 2W_2 + 3W_3$. Here, we propose a novel distributed computing scheme (as shown in Fig. 1a), which can compute this additional sum but with the same number of communicated symbols as the gradient coding scheme. With the same cyclic assignment, we let worker 1 send $2W_1 + W_2$, worker 2 send $W_2 + 2W_3$, worker 3 send $-W_1 + W_3$. It can be checked that from any two sent coded messages, the master can recover both of the two requested sums. Hence, with the same communication cost as the gradient coding scheme [9], the proposed distributed computing scheme allows the master recover the two requested linear combinations.

Since the seminal works on using coding techniques in distributed computing [4]–[6], different coded distributed computing schemes were proposed to compute various tasks in machine learning applications. The detailed comparison between the considered distributed linearly separable computation problem and each of the related existing works will be provided in Section II-B. In short,

[2]We assume that each function $f_k(\cdot)$ is arbitrary such that in general it does not hold that computing less symbols for the result W_k is less costly in terms of computation. Hence, each worker n computes the whole $W_k = f_k(D_k)$ if D_k is assigned to it. We also assume that the complexity of computing the messages from the datasets is much higher than computing the desired linear combinations of the messages. So we denote the computation cost by M.



(a) $\mathsf{K_c} = 1$.

(b) $\mathsf{K_c} = 2$.

Fig. 1: Distributed linearly separable computation with $\mathsf{K} = \mathsf{N} = 3$ and $\mathsf{N_r} = 2$. The number of datasets assigned to each worker is $\mathsf{M} = 2$.

- the distributed gradient coding problem considered in [9], [11], [12] is a special case of the considered problem in this paper with $\mathsf{K_c} = 1$ (i.e., the master requests one linear combination of the messages);
- the distributed linear transform problem considered in [13] is a special case of the considered problem in this paper where $\mathsf{L} = 1$ (i.e., each message contains one symbol) and each worker sends one symbol;
- in the distributed matrix-vector multiplication problem considered in [14]–[16], the distributed matrix-matrix multiplication problem considered in [4], [17]–[23], and the distributed multivariate polynomial computation problem considered in [24], coded assignments are allowed, i.e., linear combinations of all input datasets can be assigned to each worker. Instead, in the considered problem the data assignment phase is uncoded, such that each worker can only compute functions of the datasets which are assigned to it.

Contributions

In this paper, we formulate the distributed linearly separable computation problem and consider the case where N divides K and the computation cost is minimum, i.e., $\mathsf{M} = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N} - \mathsf{N}_{\mathrm{r}} + 1)$ by Lemma 1. Our main contributions on this case are as follows.

- We first propose an information theoretic converse bound on the minimum communication cost, inspired by the converse bound for the coded caching problem with uncoded cache placement [25], [26].
- With the cyclic assignment, widely used in the existing works on the distributed gradient coding problem such as [9]–[11],[3] we propose a novel distributed computing scheme based on the linear space intersection and prove its decodability by the Schwartz-Zippel lemma [28]–[30].[4]
- Compared to the proposed converse bound, the achievable scheme is proved to be optimal when $\mathsf{N} = \mathsf{K}$, or $\mathsf{K}_{\mathrm{c}} \in \left\{1, \ldots, \left\lceil \frac{\mathsf{K}}{\frac{\mathsf{N}}{(\mathsf{N}-\mathsf{N}_{\mathrm{r}}+1)}} \right\rceil \right\}$, or $\mathsf{K}_{\mathrm{c}} \in \left\{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_{\mathrm{r}}, \ldots, \mathsf{K}\right\}$. In addition, the proposed achievable scheme is proved to be optimal under the constraint of the cyclic assignment for all system parameters. The optimality results are listed in Table I at the top of the next page.
- By the derived optimality results, we obtain an interesting observation: when $\mathsf{K} = \mathsf{N}$, for any $\mathsf{K}_{\mathrm{c}} \in \{1, \ldots, \mathsf{N}_{\mathrm{r}}\}$, the optimal communication cost is always N_{r}. Thus by taking the same communicatoin cost as the optimal gradient coding scheme in [9] for the distributed gradient coding problem (which is the case $\mathsf{K}_{\mathrm{c}} = 1$ of our problem), with high probability our propose scheme can let the master recover any additional $\mathsf{N}_{\mathrm{r}} - 1$ linear combinations with uniformly i.i.d. coefficients over $\mathbb{F}_{\mathsf{q}}$.

Moreover, for the case where N does not divide K, the cyclic assignment cannot be directly used and we propose modified cyclic assignment and computing phases.

Paper Organization

The rest of this paper is organized as follows. Section II formulates the distributed linearly separable computation problem and explains the differences from the existing distributed computation problem in the literature. Section III provides the main results in this paper. Section IV describes the proposed achievable distributed computing scheme. Section V discusses the extensions of the proposed results. Section VI concludes the paper and some of the proofs are given in the Appendices.

[3] The main advantages of the cyclic assignment are that it can be used for any case where N divides K regardless of other system parameters, and its simplicity. According to our knowledge, the other existing assignments, such as the repetition assignments in [9], [27], can only be used for limited number of cases. In addition, the cyclic assignment is independent of the task function; thus if the master has multiple tasks in different times, we need not assign the datasets in each time.

[4] Note that the proposed computing is decodable with high probability; it will be explained in Remark 3 that for some specific tasks, additional communication cost is needed.

Notation Convention

Calligraphic symbols denote sets, bold symbols denote vectors and matrices, and sans-serif symbols denote system parameters. We use $|\cdot|$ to represent the cardinality of a set or the length of a vector; $[a:b] := \{a, a+1, \ldots, b\}$, $(a:b] := \{a+1, a+2, \ldots, b\}$, $[a:b) := \{a, a+1, \ldots, b-1\}$, $(a,b) = \{a+1, a+2, \ldots, b-1\}$ and $[n] := [1:n]$; $\oplus$ represents bit-wise XOR; $\mathbb{E}[\cdot]$ represents the expectation value of a random variable; $a! = a \times (a-1) \times \ldots \times 1$ represents the factorial of a; $\mathbb{F}_{\mathsf{q}}$ represents a finite field with order q; $\mathbf{M}^{\mathrm{T}}$ and $\mathbf{M}^{-1}$ represent the transpose and the inverse of matrix $\mathbf{M}$, respectively; the matrix $[a;b]$ is written in a Matlab form, representing $[a,b]^{\mathrm{T}}$; $\mathrm{rank}(\mathbf{M})$ represents the rank of matrix $\mathbf{M}$; $\mathbf{I}_n$ represents the identity matrix with dimension $n \times n$; $\mathbf{0}_{m \times n}$ represents the zero matrix with dimension $m \times n$; $(\mathbf{M})_{m \times n}$ represents that the dimension of matrix $\mathbf{M}$ is $m \times n$; $\mathbf{M}^{(\mathcal{S})_{\mathrm{r}}}$ represents the sub-matrix of $\mathbf{M}$ which is composed of the rows of $\mathbf{M}$ with indices in $\mathcal{S}$ (here r represents 'rows'); $\mathbf{M}^{(\mathcal{S})_{\mathrm{c}}}$ represents the sub-matrix of $\mathbf{M}$ which is composed of the columns of $\mathbf{M}$ with indices in $\mathcal{S}$ (here c represents 'columns'); $\det(\mathbf{M})$ represents the determinant matrix $\mathbf{M}$; $\mathrm{Mod}(b,a)$ represents the modulo operation on b with integer divisor a and in this paper we let $\mathrm{Mod}(b,a) \in \{1, \ldots, a\}$ (i.e., we let $\mathrm{Mod}(b,a) = a$ if a divides b); we let $\binom{x}{y} = 0$ if $x < 0$ or $y < 0$ or $x < y$. In this paper, for each set of integers $\mathcal{S}$, we sort the elements in $\mathcal{S}$ in an increasing order and denote the i^{th} smallest element by $\mathcal{S}(i)$, i.e., $\mathcal{S}(1) < \ldots < \mathcal{S}(|\mathcal{S}|)$.

The main network parameters and notations are given in Table II at the top of the next page.

II. System Model

A. Problem formulation

We formulate a $(\mathsf{K}, \mathsf{N}, \mathsf{N}_{\mathrm{r}}, \mathsf{K}_{\mathrm{c}}, \mathsf{M})$ distributed linearly separable computation problem over the canonical master-worker distributed system, as illustrated in Fig. 1. The master wants to compute a function

$$f(D_1, \ldots, D_{\mathsf{K}})$$

on K independent datasets $D_1, \ldots, D_{\mathsf{K}}$. As the data sizes are large, we distribute the computing task to a group of N workers. For distributed computation to be possible, we assume the function is *separable* to some extent. As the simplest case, we assume the function is separable to each dataset,

$$\begin{aligned} f(D_1, \ldots, D_{\mathsf{K}}) &= g\big(f_1(D_1), \ldots, f_{\mathsf{K}}(D_{\mathsf{K}})\big) && \text{(1a)} \\ &= g(W_1, \ldots, W_{\mathsf{K}}), && \text{(1b)} \end{aligned}$$

where we model $f_k(D_k)$, $k \in [\mathsf{K}]$ as the k-th message W_k and $f_k(\cdot)$ is an arbitrary function. We assume that the K messages are independent and that each message is composed of L uniformly i.i.d. symbols over a finite field $\mathbb{F}_{\mathsf{q}}$ for some large enough prime-power q, where L is large enough such that any sub-message division is possible.[5] We consider the

[5] In this paper, the basis of logarithm in the entropy terms is q.

TABLE I: Optimality results for the distributed linearly separable computation problem where $\mathsf{M} = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N} - \mathsf{N}_{\rm r} + 1)$ and N divides K.

Constraint of system parameters	Optimality
$\mathsf{N} = \mathsf{K}$	optimal
$\mathsf{N} \neq \mathsf{K}$, $\mathsf{K}_{\rm c} \in \left\{1, \ldots, \left\lceil \frac{\mathsf{K}}{\frac{\mathsf{N}}{(\mathsf{N}-\mathsf{N}_{\rm r}+1)}} \right\rceil \right\}$	optimal
$\mathsf{N} \neq \mathsf{K}$, $\mathsf{K}_{\rm c} \in \left\{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_{\rm r}, \ldots, \mathsf{K}\right\}$	optimal
$\mathsf{N} \neq \mathsf{K}$, $\mathsf{K}_{\rm c} \in \left\{\left\lceil \frac{\mathsf{K}}{\frac{\mathsf{N}}{(\mathsf{N}-\mathsf{N}_{\rm r}+1)}} \right\rceil + 1, \ldots, \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_{\rm r} - 1\right\}$	optimal under the cyclic assignment

TABLE II: Main notations

Notations	**Semantics**
K	number of datasets
N	number of workers
$\mathsf{N}_{\rm r}$	number of workers the master should wait for
$\mathcal{Z}_n$	set of datasets assigned to worker n
M	computation cost (i.e., number of datasets assigned to each worker)
X_n	transmission of worker n
T_n	number of symbols in X_n
$X_{\mathcal{A}}$	$\{X_n : n \in \mathcal{A}\}$
R	communication cost
$\mathsf{R}^\star$	minimum communication cost over all achievable computing schemes
$\mathsf{R}^\star_{\rm cyc}$	minimum communication cost over all achievable computing schemes with the cyclic assignment
D_n	the $n^{\rm th}$ dataset
$W_n = f_n(D_n)$	the $n^{\rm th}$ message
L	number of symbols of each message
$g(W_1, \ldots, W_{\mathsf{N}}) = \mathbf{F}[W_1; \ldots; W_{\mathsf{N}}]$	task function (i.e., demanded linear combinations of messages)
$\mathsf{K}_{\rm c}$	number of demanded linear combinations of messages (i.e., number of rows in $\mathbf{F}$)

simplest case of the function $g(\cdot)$, the linear mapping. So we can rewrite the task function as

$$g(W_1, \ldots, W_{\mathsf{K}}) = \mathbf{F} \begin{bmatrix} W_1 \\ \vdots \\ W_{\mathsf{K}} \end{bmatrix} = \begin{bmatrix} F_1 \\ \vdots \\ F_{\mathsf{K}_{\rm c}} \end{bmatrix}, \tag{2a}$$

where $\mathbf{F}$ is a matrix known by the master and the workers with dimension $\mathsf{K}_{\rm c} \times \mathsf{K}$, whose elements are uniformly i.i.d. over $\mathbb{F}_{\mathsf{q}}$. In other words, $g(W_1, \ldots, W_{\mathsf{K}})$ contains $\mathsf{K}_{\rm c}$ linear combinations of the K messages, whose coefficients are uniformly i.i.d. over $\mathbb{F}_{\mathsf{q}}$. In this paper, we consider the case where $\mathsf{K}_{\rm c} \leq \mathsf{K}$.[6] Note that each component function f_k where $k \in [\mathsf{K}]$ is not restricted to be linear. We also assume that $\frac{\mathsf{K}}{\mathsf{N}}$ is an integer.[7]

A computing scheme for our problem contains three phases, *data assignment*, *computing*, and *decoding*.

Data assignment phase: We assign each dataset D_k where $k \in [\mathsf{K}]$ to a subset of N workers in an uncoded manner. The set of datasets assigned to worker $n \in [\mathsf{N}]$ is denoted by $\mathcal{Z}_n$, where $\mathcal{Z}_n \subseteq [\mathsf{K}]$. The assignment constraint is that

$$|\mathcal{Z}_n| \leq \mathsf{M}, \ \forall n \in [\mathsf{N}], \tag{3}$$

where M represents the computation cost as explained in Footnote 2. The assignment function of worker n is denoted by φ_n, where

$$\mathcal{Z}_n = \varphi_n(\mathbf{F}) \subseteq [\mathsf{K}], \tag{4}$$

$$\varphi_n : [\mathbb{F}_{\mathsf{q}}]^{\mathsf{K}_{\rm c}\mathsf{K}} \to \Omega_{\mathsf{M}}(\mathsf{K}), \tag{5}$$

and $\Omega_{\mathsf{M}}(\mathsf{K})$ represents the set of all subsets of $[\mathsf{K}]$ of size not larger than M. In other words, the data assignment phase is uncoded.

Computing phase: Each worker $n \in [\mathsf{N}]$ first computes the message $W_k = f_k(D_k)$ for each $k \in \mathcal{Z}_n$. Then it computes

$$X_n = \psi_n(\{W_k : k \in \mathcal{Z}_n\}, \mathbf{F}) \tag{6}$$

where the encoding function ψ_n is such that

$$\psi_n : [\mathbb{F}_{\mathsf{q}}]^{|\mathcal{Z}_n|\mathsf{L}} \times [\mathbb{F}_{\mathsf{q}}]^{\mathsf{K}_{\rm c}\mathsf{K}} \to [\mathbb{F}_{\mathsf{q}}]^{\mathsf{T}_n}, \tag{7}$$

and T_n represents the length of X_n. Finally, worker n sends X_n to the master.

Decoding phase: The master only waits for the $\mathsf{N}_{\rm r}$ fastest workers' answers to compute $g(W_1, \ldots, W_{\mathsf{K}})$. Hence, the computing scheme can tolerate $\mathsf{N} - \mathsf{N}_{\rm r}$ stragglers. Since the master does not know a priori which workers are stragglers, the computing scheme should be designed so that from the answers of any $\mathsf{N}_{\rm r}$ workers, the master can recover $g(W_1, \ldots, W_{\mathsf{K}})$. More precisely, for any subset of workers $\mathcal{A} \subseteq [\mathsf{N}]$ where $|\mathcal{A}| = \mathsf{N}_{\rm r}$, with the definition

$$X_{\mathcal{A}} := \{X_n : n \in \mathcal{A}\}, \tag{8}$$

there exists a decoding function $\phi_{\mathcal{A}}$ such that

$$\hat{g}_{\mathcal{A}} = \phi_{\mathcal{A}}\big(X_{\mathcal{A}}, \mathbf{F}\big), \tag{9}$$

where the decoding function $\phi_{\mathcal{A}}$ is such that

$$\phi_{\mathcal{A}} : [\mathbb{F}_{\mathsf{q}}]^{\sum_{n \in \mathcal{A}} \mathsf{T}_n} \times [\mathbb{F}_{\mathsf{q}}]^{\mathsf{K}_{\rm c}\mathsf{K}} \to [\mathbb{F}_{\mathsf{q}}]^{\mathsf{K}_{\rm c}\mathsf{L}}. \tag{10}$$

[6] For the case where $\mathsf{K}_{\rm c} > \mathsf{K}$, it is straightforward to use the same code for the case where $\mathsf{K}_{\rm c} = \mathsf{K}$, since all K messages can be decoded individually.

[7] The case N does not divide K will be specifically considered in Section V-A where we extend the proposed distributed computing scheme to the general case.

The worst-case probability of error is defined as

$$\varepsilon := \max_{\mathcal{A}\subseteq[\mathsf{N}]:|\mathcal{A}|=\mathsf{N}_\mathrm{r}} \Pr\{\hat{g}_{\mathcal{A}} \neq g(W_1,\ldots,W_\mathsf{K})\}. \tag{11}$$

In addition, we denote the communication cost by,

$$\mathsf{R} := \max_{\mathcal{A}\subseteq[\mathsf{N}]:|\mathcal{A}|=\mathsf{N}_\mathrm{r}} \frac{\sum_{n\in\mathcal{A}}\mathsf{T}_n}{\mathsf{L}}, \tag{12}$$

representing the maximum normalized number of symbols downloaded by the master from any N_r responding workers. The communication cost R is achievable if there exists a computing scheme with assignment, encoding, and decoding functions such that

$$\lim_{\mathsf{q}\to\infty} \varepsilon = 0. \tag{13}$$

The minimum communication cost over all possible achievable computing schemes is denoted by $\mathsf{R}^\star$. Since the elements of $\mathbf{F}$ are uniformly i.i.d. over larger enough field, $\mathbf{F}$ is full-rank with high probability. By the simple cut-set bound, we have

$$\mathsf{R}^\star \geq \mathsf{K}_\mathrm{c}. \tag{14}$$

The following lemma provides the minimum number of workers to whom each dataset should be assigned.

Lemma 1. *Each dataset must be assigned to at least* $\mathsf{N}-\mathsf{N}_\mathrm{r}+1$ *workers.* □

Proof: Assume there exists one dataset (assumed to be D_k) assigned to only ℓ workers where $\ell < \mathsf{N}-\mathsf{N}_\mathrm{r}+1$. It can be seen that there exist at least N_r workers which does not know D_k. Hence, the answers of these N_r workers do not have any information of W_k, and thus cannot reconstruct $g(W_1,\ldots,W_\mathsf{K})$ (recall that $g(W_1,\ldots,W_\mathsf{K})$ depends on W_k with high probability). ■

In this paper, we consider the case where the computation cost is minimum, i.e., each dataset is assigned to $\mathsf{N}-\mathsf{N}_\mathrm{r}+1$ workers and

$$\mathsf{M} = |\mathcal{Z}_1| = \cdots = |\mathcal{Z}_\mathsf{N}| = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_\mathrm{r}+1).$$

The objective of this paper is to characterize the minimum communication cost for the case where the computation cost is minimum.

We then review the cyclic assignment, which was widely used in the existing works on the distributed gradient coding problem in [9] (which is a special case of the consdered problem as explained in the next subsection), such as the gradient coding schemes in [9]–[12]. For each dataset D_k where $k \in [\mathsf{K}]$, we assign D_k to worker j, where $j \in \{\mathrm{Mod}(k,\mathsf{N}), \mathrm{Mod}(k-1,\mathsf{N}), \ldots, \mathrm{Mod}(k-\mathsf{N}+\mathsf{N}_\mathrm{r},\mathsf{N})\}$.[8] In other words, the set of datasets assigned to worker $n \in [\mathsf{N}]$ is

$$\begin{aligned}\mathcal{Z}_n = \bigcup_{p\in\left[0:\frac{\mathsf{K}}{\mathsf{N}}-1\right]} \big\{&\mathrm{Mod}(n,\mathsf{N})+p\mathsf{N}, \mathrm{Mod}(n+1,\mathsf{N})+p\mathsf{N},\ldots,\\ &\mathrm{Mod}(n+\mathsf{N}-\mathsf{N}_\mathrm{r},\mathsf{N})+p\mathsf{N}\big\}\end{aligned} \tag{15}$$

[8]By convention, we let $\mathrm{Mod}(b,a) \in [1:a]$, and let $\mathrm{Mod}(b,a) = a$ if a divides b.

with cardinality $\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_\mathrm{r}+1)$. For example, if $\mathsf{K}=\mathsf{N}=4$ and $\mathsf{N}_\mathrm{r}=3$, by the cyclic assignment with $p=0$ in (15), we assign

$$\begin{aligned}&D_1, D_2, D_3 \text{ to woker } 1;\\ &D_2, D_3, D_4 \text{ to woker } 2;\\ &D_3, D_4, D_1 \text{ to woker } 3;\\ &D_4, D_1, D_2 \text{ to woker } 4.\end{aligned}$$

The minimum communication cost under the cyclic assignment in (15) is denoted by $\mathsf{R}^\star_{\mathrm{cyc}}$.

B. Connection to existing problems

Distributed gradient coding*:* When $f_k(D_k)$, $k \in [\mathsf{K}]$, represents the partial gradient vector of the loss at the current estimate of the dataset D_k and $\mathbf{F} = [1,\ldots,1]$, we have

$$f(D_1,\ldots,D_\mathsf{K}) = f_1(D_1) + \cdots + f_\mathsf{K}(D_\mathsf{K}), \tag{16}$$

representing the gradient of a generic loss function. In this case, our problem reduces to the distributed gradient coding problem in [9]. Hence, the distributed gradient coding problem in [9] is a special case of the distributed linearly separable computation problem with $\mathsf{K}_\mathrm{c} = 1$. For the case where the computation cost is minimum, based on the cyclic assignment in (15) and a random code construction, the authors in [9] proposed a gradient coding scheme which lets each worker compute and send one linear combination of the messages related to its assigned datasets, while the achieved communication cost of this scheme is optimal under the constraint of linear coding [10]. Instead of random code construction, a deterministic code construction was proposed in [11]. The authors in [12] improved the decoding delay/complexity by using Reed–Solomon codes.

The authors in [10] characterized the optimal tradeoff between the computation cost and communication cost for the distributed gradient coding problem. A distributed computing scheme achieving the same optimal computation-communication costs tradeoff as in [10] but with lower decoding complexity, was recently proposed in [31].

Some other extensions on the distributed gradient coding problem in [9] were also considered in the literature. For instance, the authors in [32] extended the gradient coding strategy to a tree-topology where the workers are located, and a fixed fraction of children nodes per parent node may be straggler. The case where the number of stragglers is not given in prior was considered in [33]. In [34], each worker sends multiple linear combinations such that the master does not always need to wait for the answers of N_r workers (i.e., from some 'good' subset of workers with the cardinality less than N_r, the master can recover the task function). It can be seen that these extended models are different from the considered problem in this paper.

Distributed linear transform*:* The distributed linear transform problem in [13] aims to compute the linear transform $\mathbf{A}\mathbf{x}$ where $\mathbf{x}$ is the input vector and $\mathbf{A}$ is a given matrix with dimension $\mathsf{K}_\mathrm{c}\times\mathsf{K}$. We should design a coding vector $\mathbf{c}_n$ for each worker $n \in [\mathsf{N}]$ (which then computes $\mathbf{c}_n\mathbf{x}$)

such that from the computation results of any $\mathsf{N_r}$ workers we can reconstruct $\mathbf{Ax}$. Meanwhile, in order to have low computation cost, each coding vector should be sparse and the number of its non-zero elements should be no more than M, where M should be minimized. Hence, the distributed linear transform problem in [13] can be seen a special case of the distributed linearly separable computation problem with $\mathsf{T}_n = \mathsf{L} = 1$ for each $n \in [\mathsf{N}]$ (recall that T_n represents the number of symbols transmitted by worker n). In other words, in this paper we consider the case where the computation cost is minimum and search for the minimum communication cost, while the authors in [13] considered the case where $\mathsf{L} = 1$ and the communication cost is minimum, and searched for the minimum computation cost. A computing scheme was proposed in [13] which needs $\mathsf{M} = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N} - \mathsf{N_r} + \mathsf{K_c})$. The authors in [35] further improved the distributed linear transform scheme in [13] by proposing a computing scheme to let each worker $n \in [\mathsf{N}]$ only access M'_n elements in $\mathbf{x}$, where $\mathsf{K}(\mathsf{N} - \mathsf{N_r} + \mathsf{K_c}) - \mathsf{N}\mathsf{N_r} < \sum_{n\in[\mathsf{N}]} \mathsf{M}'_n < \mathsf{N}\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N} - \mathsf{N_r} + \mathsf{K_c})$.

The authors in [36] considered another distributed linear transform problem with a different sparsity constraint compared to [13]. The distributed linear transform problem in [36] can be seen as a special case of the distributed linearly separable computation problem with $\mathsf{T}_n = \mathsf{L} = 1$ and $\mathsf{K_c} = \mathsf{K}$.

Distributed matrix-vector and matrix-matrix multiplications*:* Distributed computing techniques against stragglers were also used to compute matrix-vector multiplication as $\mathbf{Ab}$ [14]–[16] and matrix-matrix multiplication as $\mathbf{AB}$ [4], [17]–[23]. The general technique is to partition each input matrix into sub-matrices and assign some linear combinations of all sub-matrices (from MDS codes, polynomial codes, etc.) to the workers without considering the sparsity of the coding vectors/matrices. Thus, the assignment phase is coded.

Distributed multivariate polynomial computation*:* Similar difference as above also appears between the considered distributed linearly separable computation problem and the distributed multivariate polynomial computation problem in [24]. It was shown in [24] that the gradient descent can be computed distributedly by using a coding scheme based on the Lagrange polynomial. However, the assignment phase of the Lagrange distributed computing scheme in [24] is coded.

In summary, compared to the distributed computing schemes with coded assignment phase, the main challenge of designing computing schemes with uncoded assignment phase is that besides the decodability constraint, we should additionally guarantee that in the transmitted linear combination(s) by each worker, the coefficients of the unassigned elements are 0.

III. Main Results

We first propose a converse bound on the minimum communication cost in the following theorem, which will be proved in Appendix A inspired by the converse bound for the coded caching problem with uncoded cache placement [25], [26].

Theorem 1 (Converse)**.** *For the* $(\mathsf{K}, \mathsf{N}, \mathsf{N_r}, \mathsf{K_c}, \mathsf{M})$ *distributed linearly separable computation problem with* $\mathsf{M} = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N} - \mathsf{N_r} + 1)$,

- *when* $\mathsf{K_c} \in \left[\left[\frac{\mathsf{K}}{\binom{\mathsf{N}}{\mathsf{N}-\mathsf{N_r}+1}} \right]\right]$, *we have*
$$\mathsf{R}^\star \geq \mathsf{N_r}\mathsf{K_c}. \tag{17a}$$
- *when* $\mathsf{K_c} \in \left(\left\lceil \frac{\mathsf{K}}{\binom{\mathsf{N}}{\mathsf{N}-\mathsf{N_r}+1}} \right\rceil : \mathsf{K}\right]$, *we have*
$$\mathsf{R}^\star \geq \max\left\{ \mathsf{N_r} \left\lceil \frac{\mathsf{K}}{\binom{\mathsf{N}}{\mathsf{N}-\mathsf{N_r}+1}} \right\rceil, \mathsf{K_c} \right\}. \tag{17b}$$

□

For the case with $\mathsf{K_c} = 1$ and $\mathsf{M} = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N} - \mathsf{N_r} + 1)$ which reduces to the distributed gradient coding problem in [9], from Theorem 1 and the gradient coding scheme in [9] (each worker sends one linear combination of the assigned messages), we can directly prove the following corollary.

Corollary 1. *For the* $(\mathsf{K}, \mathsf{N}, \mathsf{N_r}, \mathsf{K_c}, \mathsf{M})$ *distributed linearly separable computation problem with* $\mathsf{M} = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N} - \mathsf{N_r} + 1)$ *and* $\mathsf{K_c} = 1$, *we have*
$$\mathsf{R}^\star = \mathsf{N_r}. \tag{18}$$

□

Note that the optimality of the gradient coding scheme in [9] for the distributed gradient coding problem was proved in [10], but under the constraint that the encoding functions in (7) are linear. In Corollary 1, we remove this constraint.

With the cyclic assignment in Section II-A, we then propose a novel achievable distributed computing scheme whose detailed proof could be found in Section IV.

Theorem 2 (Proposed distributed computing scheme)**.** *For the* $(\mathsf{K}, \mathsf{N}, \mathsf{N_r}, \mathsf{K_c}, \mathsf{M})$ *distributed linearly separable computation problem with* $\mathsf{M} = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N} - \mathsf{N_r} + 1)$, *the communication cost* $\mathsf{R_{ach}}$ *is achievable, where*

- *when* $\mathsf{K_c} \in \left[1 : \frac{\mathsf{K}}{\mathsf{N}}\right)$,
$$\mathsf{R_{ach}} = \mathsf{N_r}\mathsf{K_c}; \tag{19a}$$
- *when* $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}} : \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N_r}\right]$,
$$\mathsf{R_{ach}} = \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N_r}; \tag{19b}$$
- *when* $\mathsf{K_c} \in \left(\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N_r} : \mathsf{K}\right]$,
$$\mathsf{R_{ach}} = \mathsf{K_c}. \tag{19c}$$

□

In Theorem 2, we consider three regimes with respect to the value of $\mathsf{K_c}$ and the main ingredients are as follows.

1) $\mathsf{K_c} \in \left[1 : \frac{\mathsf{K}}{\mathsf{N}}\right)$. By some linear transformations on the request matrix $\mathbf{F}$, we treat the considered problem as $\mathsf{K_c}$ sub-problems in each of which the master requests one linear combination of messages. Thus by using the coding scheme in Corollary 1 for each sub-problem, we can let the master recover the general task function.
2) $\mathsf{K_c} \in \left[\frac{\mathsf{K}}{\mathsf{N}} : \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N_r}\right]$. This is the most interesting case, where we propose a computing scheme based on the

linear space intersection (see Remark 2 for further explanations), with the communication cost equal to the case where $\mathsf{K}_c = \frac{\mathsf{K}}{\mathsf{N}}$. We generate $\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_r - \mathsf{K}_c$ virtually requested linear combinations of messages such that the master totally recover $\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_r$ effective linear combinations of messages from the responses of any N_r workers. Each worker transmits $\frac{\mathsf{K}}{\mathsf{N}}$ linear combinations of messages which lie in the intersection of the linear spaces of its known messages and the effective demanded linear combinations. From a highly non-trivial proof based on the Schwartz-Zippel lemma [28]–[30], where the main challenge is to prove that the multivariate polynomials are generally non-zero (see Appendix D), we show that the responses of any N_r workers are linearly independent with high probability, and thus are able to reconstruct the effective demanded linear combinations.

3) $\mathsf{K}_c \in \left(\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_r : \mathsf{K}\right]$. To recover K_c linear combinations of the K messages, we propose a computing scheme to let the master totally receive K_c coded messages with L symbols each, i.e., $\mathsf{R}^\star = \mathsf{K}_c$ is achieved.

Remark 1. *Note that, when the operations are on the field of real numbers, the proposed computing scheme in Theorem 2 can work with high probability if each element in* $\mathbf{F}$ *is uniformly i.i.d. over a large enough finite set of real numbers or over an interval of real numbers.* □

By comparing the proposed converse bound in Theorem 1 and the achievable scheme in Theorem 2, we can directly derive the following optimality results.

Theorem 3 (Optimality)**.** *For the* $(\mathsf{K}, \mathsf{N}, \mathsf{N}_r, \mathsf{K}_c, \mathsf{M})$ *distributed linearly separable computation problem with* $\mathsf{M} = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_r+1)$*,*

- *when* $\mathsf{K} = \mathsf{N}$*, we have*
$$\mathsf{R}^\star = \begin{cases} \mathsf{N}_r, & \text{if } \mathsf{K}_c \in [\mathsf{N}_r]; \\ \mathsf{K}_c, & \text{if } \mathsf{K}_c \in (\mathsf{N}_r : \mathsf{K}]; \end{cases} \tag{20a}$$
- *when* $\mathsf{K}_c \in \left[\left\lceil \frac{\frac{\mathsf{K}}{\mathsf{N}}}{(\mathsf{N}-\mathsf{N}_r+1)} \right\rceil\right]$*, we have*
$$\mathsf{R}^\star = \mathsf{N}_r\mathsf{K}_c; \tag{20b}$$
- *when* $\mathsf{K}_c \in \left[\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_r : \mathsf{K}\right]$*, we have*
$$\mathsf{R}^\star = \mathsf{K}_c. \tag{20c}$$

□

From Theorem 3, it can be seen that when $\mathsf{K} = \mathsf{N}$ and $\mathsf{K}_c \in [\mathsf{N}_r]$, the optimal communication cost is always N_r (i.e., each worker sends one linear combination of the messages from its assigned datasets). Thus we prove that with the same communication cost as the optimal gradient coding scheme in [9] for the distributed gradient coding problem (from which the master recovers $W_1 + \cdots W_{\mathsf{K}}$), our propose scheme can let the master recover any additional $\mathsf{N}_r - 1$ linear combinations of the K messages whose coefficients are uniformly i.i.d. over $\mathbb{F}_{\mathsf{q}}$ with high probability.

In general, the minimum communication cost in the regime where $\mathsf{K}_c \in \left(\left\lceil \frac{\frac{\mathsf{K}}{\mathsf{N}}}{(\mathsf{N}-\mathsf{N}_r+1)} \right\rceil : \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_r\right)$ is still open. The following theorem claims that the proposed achievable scheme is optimal under the constraint of the cyclic assignment in [9], whose proof is in Appendix B.

Theorem 4 (Optimality under the cyclic assignment in [9])**.** *For the* $(\mathsf{K}, \mathsf{N}, \mathsf{N}_r, \mathsf{K}_c, \mathsf{M})$ *distributed linearly separable computation problem with* $\mathsf{M} = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N} - \mathsf{N}_r + 1)$*, the minimum communication cost under the cyclic assignment is*

$$\mathsf{R}^\star_{\text{cyc}} = \mathsf{R}_{\text{ach}}, \tag{21}$$

where R_{ach} *is given in* (19). □

IV. Achievable Distributed Computing Scheme

In this section, we introduce the proposed distributed computing scheme with the cyclic assignment in [9]. As shown in Theorem 2, we divide the range of K_c (which is $[\mathsf{K}]$) into three regimes, and present the corresponding scheme in the order, $\mathsf{K}_c \in \left[\frac{\mathsf{K}}{\mathsf{N}} : \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_r\right]$, $\mathsf{K}_c \in \left[1 : \frac{\mathsf{K}}{\mathsf{N}}\right)$, and $\mathsf{K}_c \in \left(\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_r : \mathsf{K}\right]$.

A. $\mathsf{K}_c \in \left[\frac{\mathsf{K}}{\mathsf{N}} : \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_r\right]$

We first illustrate the main idea in the following example.

Example 1 $(\mathsf{N} = 3, \mathsf{K} = 6, \mathsf{K}_c = 4, \mathsf{N}_r = 2,\ \mathsf{M} = 4)$**.** In this example, it can be seen that $\mathsf{K}_c = \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_r$. For the sake of simplicity, in the rest of this paper while illustrating the proposed schemes through examples, we assume that the field is a large enough prime field. It will be proved that in general this assumption is not necessary in our proposed schemes where we only need the field size q is large enough. Assume that the task function is

$$f(D_1, \ldots, D_6) = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} = \mathbf{F} \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \end{bmatrix} = \begin{bmatrix} 1,1,1,1,1,1 \\ 1,2,3,4,5,6 \\ 1,0,2,3,5,4 \\ 1,2,1,4,4,0 \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \end{bmatrix}.$$

Data assignment phase: By the cyclic assignment described in Section II-A, we assign that

Worker 1	Worker 2	Worker 3
D_1	D_2	D_1
D_2	D_3	D_3
D_4	D_5	D_4
D_5	D_6	D_6

Computing phase: We first focus on worker 1, who first computes W_1, W_2, W_4, and W_5 based on its assigned datasets. In other words, W_i where $i \in \{3, 6\}$ cannot be computed by worker 1. We retrieve the i^{th} column of $\mathbf{F}$ where $i \in \{3,6\}$, to obtain

$$\mathbf{F}^{(\{3,6\})_{\mathrm{c}}} = \begin{bmatrix} 1,1 \\ 3,6 \\ 2,4 \\ 1,0 \end{bmatrix}. \quad (22)$$

We then search for a vector basis for the left-side null space of $\mathbf{F}^{(\{3,6\})_{\mathrm{c}}}$. Note that $\mathbf{F}^{(\{3,6\})_{\mathrm{c}}}$ is a full-rank matrix with dimension 4×2. Hence, a vector basis for its left-side null space contains $4-2=2$ linearly independent vectors with dimension 1×4, where the product of each vector and $\mathbf{F}^{(\{3,6\})_{\mathrm{c}}}$ is $\mathbf{0}_{1\times2}$ (i.e., the zero matrix with dimension 1×2). A possible vector basis could be the set of vectors $(-6,1,0,3)$ and $(0,-2,3,0)$. It can be seen that

$$-6F_1 + 1F_2 + 0F_3 + 3F_4 = -2W_1 + 2W_2 + 10W_4 + 11W_5, \quad (23a)$$

$$0F_1 - 2F_2 + 3F_3 + 0F_4 = W_1 - 4W_2 + W_4 + 5W_5, \quad (23b)$$

both of which are independent of W_3 and W_6. Hence, the two linear combinations in (23) could be computed and then sent by worker 1.

For worker 2, who can compute W_2, W_3, W_5, and W_6, we search for the a vector basis for the left-side null space of $\mathbf{F}^{(\{1,4\})_{\mathrm{c}}}$. A possible vector basis could be the set of vectors $(0,-1,0,1)$ and $(-1,-2,3,0)$. Hence, we let worker 2 compute and send

$$0F_1 - 1F_2 + 0F_3 + 1F_4 = -2W_3 - W_5 - 6W_6, \quad (24a)$$

$$-1F_1 - 2F_2 + 3F_3 + 0F_4 = -5W_2 - W_3 + 4W_5 - W_6. \quad (24b)$$

For worker 3, who can compute W_1, W_3, W_4, and W_6, we search for the a vector basis for the left-side null space of $\mathbf{F}^{(\{2,5\})_{\mathrm{c}}}$. A possible vector basis could be the set of vectors $(-2,-2,0,3)$ and $(10,-5,3,0)$. Hence, we let worker 3 compute and send

$$-2F_1 - 2F_2 + 0F_3 + 3F_4 = -W_1 - 5W_3 + 2W_4 - 14W_6, \quad (25a)$$

$$10F_1 - 5F_2 + 3F_3 + 0F_4 = 8W_1 + W_3 - W_4 - 8W_6. \quad (25b)$$

In summary, each worker sends two linear combinations of (F_1,F_2,F_3,F_4).

Decoding phase: Assuming the set of responding workers is $\{1,2\}$. The master receives

$$\mathbf{X}_{\{1,2\}} = \begin{bmatrix} -6,1,0,3 \\ 0,-2,3,0 \\ 0,-1,0,1 \\ -1,-2,3,0 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} := \mathbf{C}_{\{1,2\}} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix}. \quad (26)$$

Since matrix $\mathbf{C}_{\{1,2\}}$ is full-rank, the master can recover $[F_1;F_2;F_3;F_4]$ by computing $\mathbf{C}_{\{1,2\}}^{-1}\mathbf{X}_{\{1,2\}}$.

Similarly, it can be checked that the four linear combinations sent from any two workers are linearly independent. Hence, by receiving the answers of any two workers, the master can recover task function.

Performance: The needed communication cost is $\frac{2\mathsf{L}+2\mathsf{L}}{\mathsf{L}} = 4$, coinciding with the converse bound $\mathsf{R}^\star \geq \mathsf{K}_{\mathrm{c}} = 4$. $\square$

We are now ready to generalize the proposed scheme in Example 1. First we focus on $\mathsf{K}_{\mathrm{c}} = \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_{\mathrm{r}}$. During the data assignment phase, we use the cyclic assignment described in Section II-A.

Computing phase: Recall that by the cyclic assignment, the set of datasets assigned to worker $n \in [\mathsf{N}]$ is

$$\mathcal{Z}_n = \bigcup_{p\in\left[0:\frac{\mathsf{K}}{\mathsf{N}}-1\right]} \big\{\mathrm{Mod}(n,\mathsf{N})+p\mathsf{N}, \mathrm{Mod}(n+1,\mathsf{N})+p\mathsf{N},\ldots, \mathrm{Mod}(n+\mathsf{N}-\mathsf{N}_{\mathrm{r}},\mathsf{N})+p\mathsf{N}\big\}$$

as defined in (15). We denote the set of datasets which are not assigned to worker n by $\overline{\mathcal{Z}_n} := [\mathsf{K}] \setminus \mathcal{Z}_n$. We retrieve columns of $\mathbf{F}$ with indices in $\overline{\mathcal{Z}_n}$ to obtain $\mathbf{F}^{(\overline{\mathcal{Z}_n})_{\mathrm{c}}}$. It can be seen that the dimension of $\mathbf{F}^{(\overline{\mathcal{Z}_n})_{\mathrm{c}}}$ is $\mathsf{K}_{\mathrm{c}}\times\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_{\mathrm{r}}-1) = \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_{\mathrm{r}}\times\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_{\mathrm{r}}-1)$, and the elements in $\mathbf{F}^{(\overline{\mathcal{Z}_n})_{\mathrm{c}}}$ are uniformly i.i.d. over $\mathbb{F}_{\mathsf{q}}$. Hence, a vector basis for the left-side null space $\mathbf{F}^{(\overline{\mathcal{Z}_n})_{\mathrm{c}}}$ is the set of $\frac{\mathsf{K}}{\mathsf{N}}$ linearly independent vectors with dimension $1\times\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_{\mathrm{r}}$, where the product of each vector and $\mathbf{F}^{(\overline{\mathcal{Z}_n})_{\mathrm{c}}}$ is $\mathbf{0}_{1\times\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_{\mathrm{r}}-1)}$.

We assume that a possible vector basis contains the vectors $\mathbf{u}_{n,1},\ldots,\mathbf{u}_{n,\frac{\mathsf{K}}{\mathsf{N}}}$. For each $j\in\left[\frac{\mathsf{K}}{\mathsf{N}}\right]$, we focus on

$$\mathbf{u}_{n,j}\mathbf{F}\begin{bmatrix} W_1 \\ \vdots \\ W_{\mathsf{K}} \end{bmatrix}. \quad (27)$$

Since $\mathbf{u}_{n,j}\mathbf{F}^{(\overline{\mathcal{Z}_n})_{\mathrm{c}}} = \mathbf{0}_{1\times\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_{\mathrm{r}}-1)}$, it can be seen that (27) is a linear combination of W_i where $i\in\mathcal{Z}_n$, which could be computed by worker n.

After computing $W_i = f_i(D_i)$ for each $i\in\mathcal{Z}_n$, worker n then computes

$$\mathbf{X}_{\{n\}} = \begin{bmatrix} \mathbf{u}_{n,1} \\ \vdots \\ \mathbf{u}_{n,\frac{\mathsf{K}}{\mathsf{N}}} \end{bmatrix} \mathbf{F} \begin{bmatrix} W_1 \\ \vdots \\ W_{\mathsf{K}} \end{bmatrix} := \mathbf{C}_{\{n\}}\mathbf{F}\begin{bmatrix} W_1 \\ \vdots \\ W_{\mathsf{K}} \end{bmatrix}, \quad (28)$$

which is then sent to the master. It can be seen that $\mathbf{X}_{\{n\}}$ contains $\frac{\mathsf{K}}{\mathsf{N}}$ linear combinations of the messages in $\mathcal{Z}_n$, each of which contains L symbols. Hence, worker n totally sends $\frac{\mathsf{K}}{\mathsf{N}}\mathsf{L}$ symbols, i.e.,

$$T_n = \frac{\mathsf{K}}{\mathsf{N}}\mathsf{L}. \quad (29)$$

Decoding phase: We provide the following lemma which will be proved in Appendix C based on the Schwartz-Zippel lemma [28]–[30].

Lemma 2. *For any set $\mathcal{A}\subseteq[\mathsf{N}]$ where $|\mathcal{A}|=\mathsf{N}_{\mathrm{r}}$, the vectors $\mathbf{u}_{n,j}$ where $n\in\mathcal{A}$ and $j\in\left[\frac{\mathsf{K}}{\mathsf{N}}\right]$ are linearly independent (i.e., $\mathbf{C}_{\mathcal{A}}$ is full-rank) with high probability.* $\square$

Assume that the set of responding workers is $\mathcal{A} = \{\mathcal{A}(1),\ldots,\mathcal{A}(\mathsf{N}_{\mathrm{r}})\}$ where $\mathcal{A}\subseteq[\mathsf{N}]$ and $|\mathcal{A}|=\mathsf{N}_{\mathrm{r}}$. Hence, the master receives

$$\mathbf{X}_{\mathcal{A}} = \begin{bmatrix} \mathbf{X}_{\mathcal{A}(1)} \\ \vdots \\ \mathbf{X}_{\mathcal{A}(\mathsf{N}_{\mathrm{r}})} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{\mathcal{A}(1)} \\ \vdots \\ \mathbf{C}_{\mathcal{A}(\mathsf{N}_{\mathrm{r}})} \end{bmatrix} \mathbf{F} \begin{bmatrix} W_1 \\ \vdots \\ W_{\mathsf{K}} \end{bmatrix}$$

$$:= \mathbf{C}_{\mathcal{A}}\mathbf{F}\left[\begin{array}{c} W_1 \\ \vdots \\ W_{\mathsf{K}} \end{array}\right]. \tag{30}$$

By Lemma 2, matrix $\mathbf{C}_{\mathcal{A}}$ is full-rank. Hence, the master can recover the task function by taking

$$\mathbf{C}_{\mathcal{A}}^{-1}\mathbf{X}_{\mathcal{A}} = \mathbf{F}\left[\begin{array}{c} W_1 \\ \vdots \\ W_{\mathsf{K}} \end{array}\right].$$

Performance: From (29), the number of symbols sent by each worker is $\frac{\mathsf{K}}{\mathsf{N}}\mathsf{L}$. Hence, the communication cost is $\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_{\rm r}$.

Remark 2. *The proposed scheme can be explained from the viewpoint on linear space. The request matrix* $\mathbf{F}$ *can be seen as a linear space composed of* $\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_{\rm r}$ *linearly independent vectors, each of which has the size* $1 \times \mathsf{K}$*. The assigned datasets to each worker* $n \in [\mathsf{N}]$*, are* D_i *where* $i \in \mathcal{Z}_n$*. Thus all the linear combinations which can be sent by worker* n *are located at a linear space composed of the vectors* $(0,\dots,0,1,0,\dots,0)$ *where* 1 *is at* i^{th} *position for* $i \in \mathcal{Z}_n$*. The intersection of these two linear spaces contains* $\frac{\mathsf{K}}{\mathsf{N}}$ *linearly independent vectors. In other words, the product of each of the* $\frac{\mathsf{K}}{\mathsf{N}}$ *vectors and* $[W_1;\dots;W_{\mathsf{K}}]$ *can be sent by worker* n*. In addition, considering any set of* $\mathsf{N}_{\rm r}$ *workers, Lemma 2 shows that the total* $\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_{\rm r}$ *vectors are linearly independent, such that the master can recover the whole linear space generated by* $\mathbf{F}$. $\square$

For each $\mathsf{K}_{\rm c} \in \left[\frac{\mathsf{K}}{\mathsf{N}} : \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_{\rm r}\right)$, the master generates a matrix $\mathbf{G}$ with dimension $\left(\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_{\rm r} - \mathsf{K}_{\rm c}\right) \times \mathsf{K}$, whose elements are uniformly i.i.d. over $\mathbb{F}_{\mathsf{q}}$. The master then requests $\mathbf{F}'[W_1;\dots;W_{\mathsf{K}}]$, where $\mathbf{F}' = [\mathbf{F};\mathbf{G}]$. Hence, we can then use the above distributed computing scheme with $\mathsf{K}_{\rm c} = \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_{\rm r}$ to let the master recover $\mathbf{F}'[W_1;\dots;W_{\mathsf{K}}]$, and the communication cost is also $\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_{\rm r}$, which coincides with (19b).

As stated in Footnote 2, the computation complexity of each worker is mainly due to the computation on the messages from the assigned datasets. Recall that L is large enough. For the proposed computing scheme in this case, the decoding complexity (i.e., the number of multiplications) of the master is $\mathcal{O}\left(\mathsf{K}_{\rm c}\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_{\rm r}\mathsf{L}\right)$.

B. $\mathsf{K}_{\rm c} \in \left[1 : \frac{\mathsf{K}}{\mathsf{N}}\right)$

We also begin with an example to illustrate the main idea.

Example 2 ($\mathsf{N}=3, \mathsf{K}=9, \mathsf{K}_{\rm c}=2, \mathsf{N}_{\rm r}=2$, $\mathsf{M}=6$)**.** Assume that the task function is

$$f(D_1,\dots,D_9) = \left[\begin{array}{c} F_1 \\ F_2 \end{array}\right] = \mathbf{F}\left[\begin{array}{c} W_1 \\ \vdots \\ W_9 \end{array}\right]$$
$$= \left[\begin{array}{c} 1,1,1,1,1,1,1,1,1 \\ 1,2,3,4,5,6,7,8,9 \end{array}\right]\left[\begin{array}{c} W_1 \\ \vdots \\ W_9 \end{array}\right].$$

By the cyclic assignment described in Section II-A, we assign that

Worker 1	Worker 2	Worker 3
D_1	D_2	D_1
D_2	D_3	D_3
D_4	D_5	D_4
D_5	D_6	D_6
D_7	D_8	D_7
D_8	D_9	D_9

Note that by the cyclic assignment, we can divide the $\mathsf{K}=9$ datasets into $\mathsf{N}=3$ groups, where in each group there are $\frac{\mathsf{K}}{\mathsf{N}} = 3$ datasets. The first group contains D_1, D_4, D_7, which are assigned to workers 1 and 3. The coefficients of (W_1, W_4, W_7) in F_1 are $(1,1,1)$ and in F_2 are $(1,4,7)$. We define that

$$W'_{1,1} = W_1 + W_4 + W_7, \tag{31a}$$
$$W'_{2,1} = W_1 + 4W_4 + 7W_7, \tag{31b}$$

which are computed by workers 1 and 3. Similarly, the second group contains D_2, D_5, D_8, which are assigned to workers 1 and 2. The coefficients of (W_2, W_5, W_8) in F_1 are $(1,1,1)$ and in F_2 are $(2,5,8)$. We define that

$$W'_{1,2} = W_2 + W_5 + W_8, \tag{32a}$$
$$W'_{2,2} = 2W_2 + 5W_5 + 8W_8, \tag{32b}$$

which are computed by workers 1 and 2. The third group contains D_3, D_6, D_9, which are assigned to workers 2 and 3. The coefficients of (W_3, W_6, W_9) in F_1 are $(1,1,1)$ and in F_2 are $(3,6,9)$. We define that

$$W'_{1,3} = W_3 + W_6 + W_9, \tag{33a}$$
$$W'_{2,3} = 3W_3 + 6W_6 + 9W_9, \tag{33b}$$

which are computed by workers 2 and 3.

Now we treat this example as two separated sub-problems, where each sub-problem is a $(\mathsf{K}', \mathsf{N}', \mathsf{N}'_{\rm r}, \mathsf{K}'_{\rm c}, \mathsf{M}') = (3,3,2,1,2)$ distributed linearly separable computation problem. In the first sub-problem, the three messages are $W'_{1,1}$, $W'_{1,2}$, and $W'_{1,3}$, and the master aims to compute $W'_{1,1} + W'_{1,2} + W'_{1,3}$. In the second sub-problem, the three messages are $W'_{2,1}$, $W'_{2,2}$, and $W'_{2,3}$, and the master aims to compute $W'_{2,1} + W'_{2,2} + W'_{2,3}$. Hence, each sub-problem can be solved by the proposed scheme in Section IV-A with communication cost equal to $\frac{\mathsf{K}'}{\mathsf{N}'}\mathsf{N}'_{\rm r} = 2$. The total communication cost is 4. $\square$

We are now ready to generalize Example 2. For each integer $n \in [\mathsf{N}]$, we focus on the set of messages $\left\{W_{n+p\mathsf{N}} : p \in \left[0 : \frac{\mathsf{K}}{\mathsf{N}} - 1\right]\right\}$. We define

$$W'_{j,n} = \sum_{p \in \left[0 : \frac{\mathsf{K}}{\mathsf{N}} - 1\right]} f_{j,n+p\mathsf{N}} W_{n+p\mathsf{N}}, \ \forall j \in [\mathsf{K}_{\rm c}], \tag{34}$$

where $f_{j,n+p\mathsf{N}}$ is the element located at the $j^{\rm th}$ row and $(n+p\mathsf{N})^{\rm th}$ column of matrix $\mathbf{F}$. Note that each message W_{n+pN} can be computed by workers in $[n : {\rm Mod}(n-\mathsf{N}+\mathsf{N}_{\rm r})]$. Hence, $W'_{j,n}$ can also be computed by workers in $[n : {\rm Mod}(n-\mathsf{N}+\mathsf{N}_{\rm r})]$.

We can re-write the task function as

$$f(D_1,\ldots,D_{\mathsf{K}}) = \begin{bmatrix} F_1 \\ \vdots \\ F_{\mathsf{K}_c} \end{bmatrix} = \begin{bmatrix} W'_{1,1}+\cdots+W'_{1,\mathsf{N}} \\ \vdots \\ W'_{\mathsf{K}_c,1}+\cdots+W'_{\mathsf{K}_c,\mathsf{N}} \end{bmatrix}. \tag{35a}$$

We then treat the problem as K_c separate sub-problems, where in the j^{th} sub-problem, the master requests $W'_{j,1}+\cdots+W'_{j,\mathsf{N}}$. Hence, each sub-problem is equivalent to the $(\mathsf{K}',\mathsf{N}',\mathsf{N}'_r,\mathsf{K}'_c,\mathsf{M}') = (\mathsf{N},\mathsf{N},\mathsf{N}_r,1,\mathsf{N}-\mathsf{N}_r+1)$ distributed linearly separable computation problem. Each sub-problem can be solved by the proposed scheme in Section IV-A with communication cost equal to $\frac{\mathsf{K}'}{\mathsf{N}'}\mathsf{N}'_r = \mathsf{N}_r$. Hence, considering all the K_c sub-problems, the total communication cost is $\mathsf{K}_c\mathsf{N}_r$, which coincides with (19a).

For the proposed computing scheme in this case, the decoding complexity of the master is $\mathcal{O}(\mathsf{K}_c\mathsf{N}_r\mathsf{L})$.

C. $\mathsf{K}_c \in \left(\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_r : \mathsf{K}\right]$

We still use an example to illustrate the main idea.

Example 3 $(\mathsf{N}=3,\mathsf{K}=3,\mathsf{K}_c=3,\mathsf{N}_r=2,\ \mathsf{M}=2)$**.** Assume that the task function is

$$f(D_1,\ldots,D_3) = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = \mathbf{F}\begin{bmatrix} W_1 \\ W_2 \\ W_3 \end{bmatrix} = \begin{bmatrix} 1,1,1 \\ 1,2,3 \\ 1,4,9 \end{bmatrix}\begin{bmatrix} W_1 \\ W_2 \\ W_3 \end{bmatrix}.$$

By the cyclic assignment described in Section II-A, we assign that

Worker 1	Worker 2	Worker 3
D_1	D_2	D_1
D_2	D_3	D_3

For each message W_k where $k\in[\mathsf{K}]$, we divide W_k into 2 non-overlapping and equal-length sub-messages, denoted by $W_{k,1}$ and $W_{k,2}$. We then use a $(3,2)$ MDS (Maximum Distance Separable) code to obtain 3 MDS-coded packets:

$$W_{k,\{1,2\}} = W_{k,1},\ W_{k,\{1,3\}} = W_{k,2},\ W_{k,\{2,3\}} = W_{k,1}+W_{k,2}.$$

Next we treat this example as 3 sub-problems, where each sub-problem is a $(\mathsf{K}',\mathsf{N}',\mathsf{N}'_r,\mathsf{K}'_c,\mathsf{M}') = (3,3,2,2,2)$ distributed linearly separable computation problem. In the first sub-problem, the three messages are $W_{1,\{1,2\}},W_{2,\{1,2\}},W_{3,\{1,2\}}$, and the master requests

$$\mathbf{F}^{(\{1,2\})_r}\begin{bmatrix} W_{1,\{1,2\}} \\ W_{2,\{1,2\}} \\ W_{3,\{1,2\}} \end{bmatrix} = \begin{bmatrix} W_{1,\{1,2\}}+W_{2,\{1,2\}}+W_{3,\{1,2\}} \\ W_{1,\{1,2\}}+2W_{2,\{1,2\}}+3W_{3,\{1,2\}} \end{bmatrix}.$$

In the second sub-problem, the three messages are $W_{1,\{1,3\}},W_{2,\{1,3\}},W_{3,\{1,3\}}$, and the master requests

$$\mathbf{F}^{(\{1,3\})_r}\begin{bmatrix} W_{1,\{1,3\}} \\ W_{2,\{1,3\}} \\ W_{3,\{1,3\}} \end{bmatrix} = \begin{bmatrix} W_{1,\{1,3\}}+W_{2,\{1,3\}}+W_{3,\{1,3\}} \\ W_{1,\{1,3\}}+4W_{2,\{1,3\}}+9W_{3,\{1,3\}} \end{bmatrix}.$$

In the third sub-problem, the three messages are $W_{1,\{2,3\}},W_{2,\{2,3\}},W_{3,\{2,3\}}$, and the master requests

$$\mathbf{F}^{(\{2,3\})_r}\begin{bmatrix} W_{1,\{2,3\}} \\ W_{2,\{2,3\}} \\ W_{3,\{2,3\}} \end{bmatrix} = \begin{bmatrix} W_{1,\{2,3\}}+2W_{2,\{2,3\}}+3W_{3,\{2,3\}} \\ W_{1,\{2,3\}}+4W_{2,\{2,3\}}+9W_{3,\{2,3\}} \end{bmatrix}.$$

Each sub-problem can be solved by the proposed scheme in Section IV-A, where each worker sends $\frac{\mathsf{K}'}{\mathsf{N}'}=1$ linear combination of sub-messages with $\frac{\mathsf{L}}{2}$ symbols. Hence, each worker totally sends $\frac{3\mathsf{L}}{2}$ symbols, and thus the communication cost equal to $\frac{3\mathsf{L}\mathsf{N}_r}{2\mathsf{L}}=3$.

Now we show that by solving the three sub-problems, the master can recover the task, i.e., $F_1=W_1+W_2+W_3$, $F_2=W_1+2W_2+3W_3$, and $F_3=W_1+4W_2+9W_3$.

From the first and second sub-problems, the master can recover

$$W_{1,\{1,2\}}+W_{2,\{1,2\}}+W_{3,\{1,2\}} = W_{1,1}+W_{2,1}+W_{3,1}, \tag{36a}$$

$$W_{1,\{1,3\}}+W_{2,\{1,3\}}+W_{3,\{1,3\}} = W_{1,2}+W_{2,2}+W_{3,2}. \tag{36b}$$

Hence, by concatenating (36a) and (36b), the master can recover F_1.

From the first and third sub-problems, the master can recover

$$W_{1,\{1,2\}}+2W_{2,\{1,2\}}+3W_{3,\{1,2\}} = W_{1,1}+2W_{2,1}+3W_{3,1}, \tag{37a}$$

$$W_{1,\{2,3\}}+2W_{2,\{2,3\}}+3W_{3,\{2,3\}} = (W_{1,1}+W_{1,2})+ 2(W_{2,1}+W_{2,2})+3(W_{3,1}+W_{3,2}). \tag{37b}$$

From (37a) and (37b), the master can first recover $W_{1,2}+2W_{2,2}+3W_{3,2}$, which is then concatenated with (37a). Hence, the master can recover F_2.

From the second and third sub-problems, the master can recover

$$W_{1,\{1,3\}}+4W_{2,\{1,3\}}+9W_{3,\{1,3\}} = W_{1,2}+4W_{2,2}+9W_{3,2}, \tag{38a}$$

$$W_{1,\{2,3\}}+4W_{2,\{2,3\}}+9W_{3,\{2,3\}} = (W_{1,1}+W_{1,2}) + 4(W_{2,1}+W_{2,2})+9(W_{3,1}+W_{3,2}). \tag{38b}$$

From (38a) and (38b), the master can first recover $W_{1,1}+4W_{2,1}+9W_{3,1}$, which is then concatenated with (38a). Hence, the master can recover F_3. □

We are now ready to generalize Example 3. We divide each message W_k into $\binom{\mathsf{K}_c-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_r-1}$ equal-length and non-overlapped sub-messages, $W_k = \left(W_{k,1},\ldots,W_{k,\binom{\mathsf{K}_c-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_r-1}}\right)$, which are then encoded by a $\left(\binom{\mathsf{K}_c}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_r},\binom{\mathsf{K}_c-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_r-1}\right)$ MDS code. Each MDS-coded packet is denoted by $W_{k,\mathcal{S}}$ where $\mathcal{S}\subseteq[\mathsf{K}_c]$

where $|\mathcal{S}| = \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}$. Since $W_{k,\mathcal{S}}$ is a linear combination of $\left(W_{k,1}, \dots, W_{k,\binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1}}\right)$, we define that

$$W_{k,\mathcal{S}} = \mathbf{v}_\mathcal{S} \begin{bmatrix} W_{k,1} \\ \vdots \\ W_{k,\binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1}} \end{bmatrix}, \ \forall \mathcal{S} \subseteq [\mathsf{K}_\mathrm{c}] : |\mathcal{S}| = \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}, \tag{39}$$

where $\mathbf{v}_\mathcal{S}$ with $\binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1}$ elements represents the generation vector to generate the MDS-coded packet $W_{k,\mathcal{S}}$. Note that each MDS-coded packet has $\frac{\mathsf{L}}{\binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1}}$ symbols.

Next we treat the problem as $\binom{\mathsf{K}_\mathrm{c}}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}}$ sub-problems, where each sub-problem is a $(\mathsf{K}', \mathsf{N}', \mathsf{N}'_\mathrm{r}, \mathsf{K}'_\mathrm{c}, \mathsf{M}') = \left(\mathsf{K}, \mathsf{N}, \mathsf{N}_\mathrm{r}, \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}, \mathsf{M}\right)$ distributed linearly separable computation problem. For each $\mathcal{S} \subseteq [\mathsf{K}_\mathrm{c}]$ where $|\mathcal{S}| = \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}$, there is a sub-problem. In this sub-problem the messages are $W_{1,\mathcal{S}}, \dots, W_{\mathsf{K},\mathcal{S}}$, and the master requests

$$\mathbf{F}^{(\mathcal{S})_\mathrm{r}} \begin{bmatrix} W_{1,\mathcal{S}} \\ \vdots \\ W_{\mathsf{K},\mathcal{S}} \end{bmatrix}.$$

Each sub-problem can be solved by the proposed scheme in Section IV-A, where each worker sends $\frac{\mathsf{K}}{\mathsf{N}}$ linear combination of sub-messages with $\frac{\mathsf{L}}{\binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1}}$ symbols. Hence, each worker totally sends

$$\binom{\mathsf{K}_\mathrm{c}}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}} \frac{\mathsf{K}}{\mathsf{N}} \frac{\mathsf{L}}{\binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1}} = \frac{\mathsf{L}\mathsf{K}_\mathrm{c}}{\mathsf{N}_\mathrm{r}}$$

symbols, and thus the communication cost equal to $\mathsf{N}_\mathrm{r}\frac{\mathsf{L}\mathsf{K}_\mathrm{c}}{\mathsf{N}_\mathrm{r}\mathsf{L}} = \mathsf{K}_\mathrm{c}$, which coincides with (19c).

Now we show that by solving all the sub-problems, the master can recover the task, i.e., for each $j \in [\mathsf{K}_\mathrm{c}]$ the master can recover

$$F_j = \mathbf{F}^{(\{j\})_\mathrm{r}}[W_1; \dots; W_\mathsf{K}] = f_{j,1}W_1 + \dots + f_{j,\mathsf{K}}W_\mathsf{K} \tag{40a}$$

$$= f_{j,1} \begin{bmatrix} W_{1,1} \\ \vdots \\ W_{1,\binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1}} \end{bmatrix} + \dots + f_{j,\mathsf{K}} \begin{bmatrix} W_{\mathsf{K},1} \\ \vdots \\ W_{\mathsf{K},\binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1}} \end{bmatrix}, \tag{40b}$$

where we define that $\mathbf{F}^{(\{j\})_\mathrm{r}} := [f_{j,1}, \dots, f_{j,\mathsf{K}}]$.

For each $\mathcal{S} \subseteq [\mathsf{K}_\mathrm{c}]$ where $|\mathcal{S}| = \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}$ and $j \in \mathcal{S}$, in the corresponding sub-problem the master has recovered

$$\mathbf{F}^{(\{j\})_\mathrm{r}}[W_{1,\mathcal{S}}; \dots; W_{\mathsf{K},\mathcal{S}}] = f_{j,1}W_{1,\mathcal{S}} + \dots + f_{j,\mathsf{K}}W_{\mathsf{K},\mathcal{S}} \tag{41a}$$

$$= f_{j,1}\mathbf{v}_\mathcal{S} \begin{bmatrix} W_{1,1} \\ \vdots \\ W_{1,\binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1}} \end{bmatrix} + \dots + f_{j,\mathsf{K}}\mathbf{v}_\mathcal{S} \begin{bmatrix} W_{\mathsf{K},1} \\ \vdots \\ W_{\mathsf{K},\binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1}} \end{bmatrix}. \tag{41b}$$

We assume that all the sets $\mathcal{S} \subseteq [\mathsf{K}_\mathrm{c}]$ where $|\mathcal{S}| = \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}$ and $j \in \mathcal{S}$, are $\mathcal{S}_1, \dots, \mathcal{S}_{\binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1}}$. By considering all the sub-problems corresponding to the above sets, the master has recovered

$$f_{j,1} \begin{bmatrix} \mathbf{v}_{\mathcal{S}_1} \\ \vdots \\ \mathbf{v}_{\mathcal{S}_{\binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1}}} \end{bmatrix} \begin{bmatrix} W_{1,1} \\ \vdots \\ W_{1,\binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1}} \end{bmatrix} + \dots + f_{j,\mathsf{K}} \begin{bmatrix} \mathbf{v}_{\mathcal{S}_1} \\ \vdots \\ \mathbf{v}_{\mathcal{S}_{\binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1}}} \end{bmatrix} \begin{bmatrix} W_{\mathsf{K},1} \\ \vdots \\ W_{\mathsf{K},\binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1}} \end{bmatrix} := \mathbf{H}_j. \tag{42}$$

Note that $\begin{bmatrix} \mathbf{v}_{\mathcal{S}_1} \\ \vdots \\ \mathbf{v}_{\mathcal{S}_{\binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1}}} \end{bmatrix}$ is full-rank with size $\binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1} \times \binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1}$, and thus invertible. Hence, the master can recover F_j in (40b) by taking $\begin{bmatrix} \mathbf{v}_{\mathcal{S}_1} \\ \vdots \\ \mathbf{v}_{\mathcal{S}_{\binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1}}} \end{bmatrix}^{-1} \mathbf{H}_j$.

For the proposed computing scheme in this case, the decoding complexity of the master is $\mathcal{O}\left(\mathsf{K}_\mathrm{c}\binom{\mathsf{K}_\mathrm{c}-1}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}-1}\mathsf{L}\right)$.

Remark 3. *By using the Schwartz-Zippel Lemma, we prove that the proposed scheme is decodable with high probability if the elements in the demand matrix* $\mathbf{F}$ *are uniformly i.i.d. over some large field. However, for some specific* $\mathbf{F}$*, the proposed scheme is not decodable (i.e.,* $\mathbf{C}_\mathcal{A}$ *is not full-rank) and we may need more communication load.*

Let us focus on the $(\mathsf{K}, \mathsf{N}, \mathsf{N}_\mathrm{r}, \mathsf{K}_\mathrm{c}, \mathsf{M}) = (3, 3, 2, 2, 2)$ *distributed linearly separable computation problem. In this example, there is only one possible assignment, which is as follows,*

Worker 1	*Worker 2*	*Worker 3*
W_1	W_2	W_1
W_2	W_3	W_3

Noting that in this case we have $\mathsf{N} = \mathsf{K}$ *and* $\mathsf{K}_\mathrm{c} = \mathsf{N}_\mathrm{r}$*. From Theorem 3, the proposed scheme in Section IV-A is decodable with high probability if the elements in the demand matrix* $\mathbf{F}$ *are uniformly i.i.d. over some large field, and achieves the optimal communication cost* 2*.*

In the following, we focus on a specific demand matrix

$$\mathbf{F}' = \begin{bmatrix} 1,1,1 \\ 2,1,1 \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \\ W_3 \end{bmatrix} = \begin{bmatrix} W_1 + W_2 + W_3 \\ 2W_1 + W_2 + W_3 \end{bmatrix}. \tag{43}$$

Note that the demand is equivalent to $(W_1, W_2 + W_3)$*. If we use the proposed scheme in Section IV-A, it can be seen that* $C_{\{1\}} = [1, -1]$*,* $C_{\{2\}} = [2, -1]$*, and* $C_{\{3\}} = [1, -1]$*. So we have* $C_{\{1,3\}} = \begin{bmatrix} 1, -1 \\ 1, -1 \end{bmatrix}$ *is not full-rank, and thus the proposed scheme is not decodable. In the following, we will prove that the optimal communication cost for this demand matrix is* 3*.*

[Converse]: *We now prove that the communication cost is no less than* 3. *Note that from* X_1 *and* X_3*, the master can recover* W_1 *and* $W_2 + W_3$. *Hence, we have*

$$\begin{aligned} 0 &= H(W_2 + W_3|X_1, X_3) && (44a)\\ &\geq H(W_2 + W_3|X_1, X_3, W_1, W_3) && (44b)\\ &= H(W_2 + W_3|X_1, W_1, W_3) && (44c)\\ &= H(W_2|X_1, W_1, W_3) && (44d)\\ &= H(W_2|X_1, W_1), && (44e) \end{aligned}$$

where (44c) *comes from that* X_3 *is a function of* (W_1, W_3) *and* (44e) *comes from that* W_3 *is independent of* (W_1, W_2, X_1). *Since the master can recover* W_1 *from* (X_1, X_3)*,* (44e) *shows that from* (X_1, X_3) *the master can also recover* W_2*, i.e.,*

$$H(W_1, W_2|X_1, X_3) = 0. \tag{45}$$

Moreover, we have

$$\begin{aligned} 0 &= H(W_2 + W_3|X_1, X_3) && (46a)\\ &\geq H(W_2 + W_3|X_1, X_3, W_1, W_2) && (46b)\\ &= H(W_3|X_1, X_3, W_1, W_2) && (46c)\\ &= H(W_3|X_1, X_3), && (46d) \end{aligned}$$

where (46d) *comes from* (45). *Hence, we have*

$$H(W_1, W_2, W_3|X_1, X_3) = 0. \tag{47}$$

Note that from X_1 *and* X_2*, the master can recover* W_1 *and* $W_2 + W_3$. *Since the master can recover* W_1 *from* (X_1, X_2)*,* (44e) *shows that from* (X_1, X_2) *the master can also recover* W_2*, i.e.,*

$$H(W_1, W_2|X_1, X_2) = 0. \tag{48}$$

Moreover, we have

$$\begin{aligned} 0 &= H(W_2 + W_3|X_1, X_2) && (49a)\\ &\geq H(W_2 + W_3|X_1, X_2, W_1, W_2) && (49b)\\ &= H(W_3|X_1, X_2, W_1, W_2) && (49c)\\ &= H(W_3|X_1, X_2), && (49d) \end{aligned}$$

where (49d) *comes from* (48). *From* (48) *and* (49d)*, we have*

$$H(W_1, W_2, W_3|X_1, X_2) = 0. \tag{50}$$

Similarly, we also have

$$H(W_1, W_2, W_3|X_2, X_3) = 0. \tag{51}$$

From (47)*,* (50)*, and* (51)*, it can be seen that for any set of workers* $\mathcal{A} \subseteq [3]$ *where* $|\mathcal{A}| = 2$*, we have (recall that* $X_{\mathcal{A}} := \{X_n : n \in \mathcal{A}\}$*)*

$$H(X_{\mathcal{A}}) \geq 3\mathsf{L}, \tag{52}$$

Hence, we have the communication cost is no less than 3.

[Achievability]: *We can use the proposed scheme in Example 3 to let the master recover* 3 *linearly independent linear combinations of* (W_1, W_2, W_3)*, such that the master can recover each message and then recover* $(W_1, W_2 + W_3)$. *The needed communication cost is* 3 *as shown in Example 3, which coincides with the above converse bound.*

From the above proof, we can also see that for the $(\mathsf{K}, \mathsf{N}, \mathsf{N}_{\rm r}, \mathsf{K}_{\rm c}, \mathsf{M}) = (3, 3, 2, 2, 2)$ *distributed linearly separable computation problem,*

- *if the demand matrix is full-rank and it contains a sub-matrix with dimension* 2×2 *which is not full-rank, the optimal communication cost is* 3*;*
- *otherwise, the optimal communication cost is* 2.

It is one of our on-going works to study the specific demand matrices for more general case. □

V. Extensions

In this section, we will discuss about the extension of the proposed scheme in Section IV. In Section V-A, we propose an extended scheme for the general values of K and N (i.e., N does not necessarily divide K). In Section V-B, we provide an example to show that the cyclic assignment is sub-optimal.

A. General values of K and N

We assume that $\mathsf{K} = \mathsf{a}\mathsf{N} + \mathsf{b}$, where a is a non-negative integer and $\mathsf{b} \in [\mathsf{N} - 1]$. Since we still consider the minimum computation cost and each dataset should be assigned to at least $\mathsf{N} - \mathsf{N}_{\rm r} + 1$ workers, thus now the minimum computation cost is

$$\left\lceil \frac{\mathsf{K}}{\mathsf{N}} (\mathsf{N} - \mathsf{N}_{\rm r} + 1) \right\rceil = \mathsf{a}(\mathsf{N} - \mathsf{N}_{\rm r} + 1) + \left\lceil \frac{\mathsf{b}}{\mathsf{N}} (\mathsf{N} - \mathsf{N}_{\rm r} + 1) \right\rceil. \tag{53}$$

It will be explained later that in order to enable the extension of the cyclic assignment to the general values of K and N, we consider the computation cost

$$\mathsf{M}_1 := \mathsf{a}(\mathsf{N} - \mathsf{N}_{\rm r} + 1) + \left\lceil \frac{\mathsf{N} - \mathsf{N}_{\rm r} + 1}{\left\lfloor \frac{\mathsf{N}}{\mathsf{b}} \right\rfloor} \right\rceil, \tag{54}$$

which may be slightly larger than the minimum computation cost in (53).

We generalize the proposed scheme in Section IV by introducing $\mathsf{N} - \mathsf{b}$ virtual datasets, to obtain the following theorem, which is the generalized version of Theorem 2.

Theorem 5. *For the* $(\mathsf{K}, \mathsf{N}, \mathsf{N}_{\rm r}, \mathsf{K}_{\rm c}, \mathsf{M})$ *distributed linearly separable computation problem with* $\mathsf{K} = \mathsf{a}\mathsf{N} + \mathsf{b}$ *and* $\mathsf{M} = \mathsf{M}_1$ *where* a *is a non-negative integer and* $\mathsf{b} \in [\mathsf{N} - 1]$*, the communication cost* $\mathsf{R}'_{\rm ach}$ *is achievable, where*

- *when* $\mathsf{K}_{\rm c} \in \left[\left\lfloor \frac{\mathsf{K}}{\mathsf{N}} \right\rfloor\right]$*,*
$$\mathsf{R}'_{\rm ach} = \mathsf{N}_{\rm r}\mathsf{K}_{\rm c}; \tag{55a}$$
- *when* $\mathsf{K}_{\rm c} \in \left[\left\lceil \frac{\mathsf{K}}{\mathsf{N}} \right\rceil : \left\lceil \frac{\mathsf{K}}{\mathsf{N}} \right\rceil \mathsf{N}_{\rm r}\right]$*,*
$$\mathsf{R}'_{\rm ach} = \left\lceil \frac{\mathsf{K}}{\mathsf{N}} \right\rceil \mathsf{N}_{\rm r}; \tag{55b}$$
- *when* $\mathsf{K}_{\rm c} \in \left(\left\lceil \frac{\mathsf{K}}{\mathsf{N}} \right\rceil \mathsf{N}_{\rm r} : \mathsf{K}\right]$*,*
$$\mathsf{R}'_{\rm ach} = \mathsf{R}^{\star} = \mathsf{K}_{\rm c}, \tag{55c}$$

  where $\mathsf{R}^{\star}$ *represents the optimal communication cost for this case.*

□

Proof: We first extend the cyclic assignment in Section II-A to the general case by dividing the K datasets into two groups, $[\mathsf{aN}]$ and $[\mathsf{aN}+1:\mathsf{K}]$, respectively.

- For each dataset D_k where $k \in [\mathsf{aN}]$, we assign D_k to worker j, where $j \in \{\mathrm{Mod}(k,\mathsf{N}), \mathrm{Mod}(k-1,\mathsf{N}), \dots, \mathrm{Mod}(k-\mathsf{N}+\mathsf{N}_\mathrm{r},\mathsf{N})\}$. Hence, the assignment on the datasets in the first group is the same as the cyclic assignment in Section II-A. The number of datasets in the first group assigned to each worker is
$$\mathsf{a}(\mathsf{N}-\mathsf{N}_\mathrm{r}+1). \tag{56}$$
- For the second group, we introduce $\mathsf{N}-\mathsf{b}$ virtual datasets and thus there are totally N effective (real or virtual) datasets. We then use the cyclic assignment in Section II-A to assign the N effective datasets to the workers, such that the number of effective datasets assigned to each worker is $\mathsf{N}-\mathsf{N}_\mathrm{r}+1$. To satisfy the assignment constraint (i.e., $|\mathcal{Z}_n| \leq \mathsf{M}$ for each $n \in [\mathsf{N}]$), it can be seen from (54) and (56) that the number of real datasets in the second group assigned to each worker should be no more than $\left\lceil \frac{\mathsf{N}-\mathsf{N}_\mathrm{r}+1}{\lfloor \frac{\mathsf{N}}{\mathsf{b}} \rfloor} \right\rceil$. Hence, our objective is to choose b datasets from N effective datasets as the real datasets, such that by the cyclic assignment on these N effective datasets the number of real datasets assigned to each worker is no more than $\left\lceil \frac{\mathsf{N}-\mathsf{N}_\mathrm{r}+1}{\lfloor \frac{\mathsf{N}}{\mathsf{b}} \rfloor} \right\rceil$. We will propose an allocation algorithm in Appendix E which can generally attain the above objective. Here we provide an example to illustrate the idea, where $\mathsf{K}=\mathsf{b}=3$, $\mathsf{a}=0$, $\mathsf{N}=6$, and $\mathsf{N}_\mathrm{r}=4$. We have totally 6 effective datasets denoted by, $E_1,\dots,E_6$. By the cyclic assignment, the number of effective datasets assigned to each worker is $\mathsf{N}-\mathsf{N}_\mathrm{r}+1=3$. Thus we assign that

Worker 1	Worker 2	Worker 3
E_1	E_2	E_3
E_2	E_3	E_4
E_3	E_4	E_5
Worker 4	**Worker 5**	**Worker 6**
E_4	E_5	E_6
E_5	E_6	E_1
E_6	E_1	E_2

By choosing E_1, E_3, and E_5 as the real datasets, it can be seen that the number of real datasets assigned to each worker is no more than $\left\lceil \frac{\mathsf{N}-\mathsf{N}_\mathrm{r}+1}{\lfloor \frac{\mathsf{N}}{\mathsf{b}} \rfloor} \right\rceil = 2$.

After the data assignment phase, each worker then computes the message for each assigned real dataset. The virtual message which comes from each virtual dataset, is set to be a vector of L zeros. We then directly use the computing phase of the proposed scheme in Section IV for the $(\mathsf{K}',\mathsf{N}',\mathsf{N}'_\mathrm{r},\mathsf{K}'_\mathrm{c},\mathsf{M}') = ((\mathsf{a}+1)\mathsf{N},\mathsf{N},\mathsf{N}_\mathrm{r},\mathsf{K}_\mathrm{c},(\mathsf{a}+1)(\mathsf{N}-\mathsf{N}_\mathrm{r}+1))$ distributed linearly separable computation problem, to achieve the communication cost in Theorem 5. ■

B. Improvement on the cyclic assignment

In the following, we will provide an example which shows the sub-optimality of the cyclic assignment.

Example 4 ($\mathsf{K}=12$, $\mathsf{N}=4$, $\mathsf{N}_\mathrm{r}=3$, $\mathsf{K}_\mathrm{c}=3$, $\mathsf{M}=6$)**.** Consider the example where $\mathsf{K}=12$, $\mathsf{N}=4$, $\mathsf{N}_\mathrm{r}=3$, $\mathsf{K}_\mathrm{c}=3$, and we assign $\mathsf{M}=\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_\mathrm{r}+1)=6$ datasets to each worker. Each dataset is assigned to $\mathsf{N}-\mathsf{N}_\mathrm{r}+1=2$ workers. By the proposed scheme with the cyclic assignment for the case where $\mathsf{K}_\mathrm{c}=\frac{\mathsf{K}}{\mathsf{N}}$ in Theorem 2, the needed communication cost is $\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}=9$, which is optimal under the constraint of the cyclic assignment. However, by the proposed converse bound in Theorem 1, the minimum communication cost is upper bounded by 6. We will introduce a novel distributed computing scheme to achieve the minimum communication cost. As a result, we show the sub-optimality of the cyclic assignment.

Data assignment phase: Inspired by the placement phase of the coded caching scheme in [7], we assign that

Worker 1	Worker 2	Worker 3	Worker 4
D_1	D_1	D_3	D_5
D_2	D_2	D_4	D_6
D_3	D_7	D_7	D_9
D_4	D_8	D_8	D_{10}
D_5	D_9	D_{11}	D_{11}
D_6	D_{10}	D_{12}	D_{12}

More precisely, we partition the 12 datasets into $\binom{4}{2}=6$ groups, each of which is denoted by $\mathcal{H}_\mathcal{T}$ where $\mathcal{T} \subseteq [4]$ where $|\mathcal{T}|=2$ and contains 2 datasets. In this example, we let
$$\mathcal{H}_{\{1,2\}}=\{1,2\},\ \mathcal{H}_{\{1,3\}}=\{3,4\},\ \mathcal{H}_{\{1,4\}}=\{5,6\},$$
$$\mathcal{H}_{\{2,3\}}=\{7,8\},\ \mathcal{H}_{\{2,4\}}=\{9,10\},\ \mathcal{H}_{\{3,4\}}=\{11,12\}.$$
For each set $\mathcal{T} \subseteq [4]$ where $|\mathcal{T}|=2$, we assign dataset D_k where $k \in \mathcal{H}_\mathcal{T}$ to workers in $\mathcal{T}$. Hence, each dataset is assigned to 2 workers, and the number of datasets assigned to each worker is $2\binom{4-1}{2-1}=6$ (e.g., the datasets in groups $\mathcal{H}_{\{1,2\}},\mathcal{H}_{\{1,3\}},\mathcal{H}_{\{1,4\}}$ are assigned to worker k), satisfying the assignment constraint.

Computing phase: We assume that the task function is
$$f(D_1,\dots,D_\mathsf{K}) = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = \mathbf{F} \begin{bmatrix} W_1 \\ \vdots \\ W_{12} \end{bmatrix}$$
$$= \begin{bmatrix} 1,1,1,1,1,1,1,1,1,1,1,1 \\ 1,2,3,4,5,6,7,8,9,10,11,12 \\ 1,0,3,2,8,4,1,2,9,4,5,10 \end{bmatrix} \begin{bmatrix} W_1 \\ \vdots \\ W_{12} \end{bmatrix}.$$
Note that the following proposed scheme works for any request with high probability, where the elements $\mathbf{F}$ are uniformly i.i.d.

We now focus on each group $\mathcal{H}_\mathcal{T}$ where $\mathcal{T} \subseteq [6]$ and $|\mathcal{T}|=2$. When $\mathcal{T}=\{1,2\}$, we have $\mathcal{H}_{\{1,2\}}=\{1,2\}$. We retrieve the sub-matrix
$$\mathbf{F}^{(\{1,2\})_\mathrm{c}} = \begin{bmatrix} 1,1 \\ 1,2 \\ 1,0 \end{bmatrix},$$

i.e., columns with indices in $\mathcal{H}_{\{1,2\}} = \{1,2\}$ of $\mathbf{F}$. Since the dimension of $\mathbf{F}^{(\{1,2\})_c}$ is 3×2, the left-side null-space of $\mathbf{F}^{(\{1,2\})_c}$ contains one vector. Now we choose the vector $(-2,1,1)$, where $(-2,1,1)\mathbf{F}^{(\{1,2\})_c} = (0,0)$. Hence, in the product $(-2,1,1)[F_1; F_2; F_3]$, the coefficients of W_1 and W_2 are 0. We define that

$$U_{\mathcal{T}} = U_{\{1,2\}} := (-2,1,1)[F_1; F_2; F_3] = -2F_1 + 1F_2 + 1F_3 \tag{57a}$$

$$= \mathbf{0W_1} + \mathbf{0W_2} + 4W_3 + 4W_4 + 11W_5 + 8W_6 + 6W_7 + 8W_8 + 16W_9 + 12W_{10} + 14W_{11} + 20W_{12}. \tag{57b}$$

Similarly, when $\mathcal{T} = \{1,3\}$, we have $\mathcal{H}_{\{1,3\}} = \{3,4\}$. By choosing the vector $(-6,1,1)$ as the left-side null-space of $\mathbf{F}^{(\{3,4\})_c}$, and define that

$$U_{\{1,3\}} := (-6,1,1)[F_1; F_2; F_3] = -6F_1 + 1F_2 + 1F_3 \tag{58a}$$

$$= -4W_1 - 4W_2 + \mathbf{0W_3} + \mathbf{0W_4} + 7W_5 + 4W_6 + 2W_7 + 4W_8 + 12W_9 + 8W_{10} + 10W_{11} + 16W_{12}. \tag{58b}$$

When $\mathcal{T} = \{1,4\}$, we have $\mathcal{H}_{\{1,4\}} = \{5,6\}$. By choosing the vector $(-28,4,1)$ as the left-side null-space of $\mathbf{F}^{(\{5,6\})_c}$, and define that

$$U_{\{1,4\}} := (-28,4,1)[F_1; F_2; F_3] = -28F_1 + 4F_2 + 1F_3 \tag{59a}$$

$$= -23W_1 - 20W_2 - 13W_3 - 10W_4 + \mathbf{0W_5} + \mathbf{0W_6} + 1W_7 + 6W_8 + 17W_9 + 16W_{10} + 21W_{11} + 30W_{12}. \tag{59b}$$

When $\mathcal{T} = \{2,3\}$, we have $\mathcal{H}_{\{2,3\}} = \{7,8\}$. By choosing the vector $(6,-1,1)$ as the left-side null-space of $\mathbf{F}^{(\{7,8\})_c}$, and define that

$$U_{\{2,3\}} := (6,-1,1)[F_1; F_2; F_3] = 6F_1 - 1F_2 + 1F_3 \tag{60a}$$

$$= 6W_1 + 4W_2 + 6W_3 + 4W_4 + 9W_5 + 4W_6 + \mathbf{0W_7} + \mathbf{0W_8} + 6W_9 + 0W_{10} + 0W_{11} + 4W_{12}. \tag{60b}$$

When $\mathcal{T} = \{2,4\}$, we have $\mathcal{H}_{\{2,4\}} = \{9,10\}$. By choosing the vector $(-54,5,1)$ as the left-side null-space of $\mathbf{F}^{(\{9,10\})_c}$, and define that

$$U_{\{2,4\}} := (-54,5,1)[F_1; F_2; F_3] = -54F_1 + 5F_2 + 1F_3 \tag{61a}$$

$$= -48W_1 - 44W_2 - 36W_3 - 32W_4 - 21W_5 - 20W_6 - 18W_7 - 12W_8 + \mathbf{0W_9} + \mathbf{0W_{10}} + 6W_{11} + 16W_{12}. \tag{61b}$$

When $\mathcal{T} = \{3,4\}$, we have $\mathcal{H}_{\{3,4\}} = \{11,12\}$. By choosing the vector $(50,-5,1)$ as the left-side null-space of $\mathbf{F}^{(\{11,12\})_c}$, and define that

$$U_{\{3,4\}} := (50,-5,1)[F_1; F_2; F_3] = 50F_1 - 5F_2 + 1F_3 \tag{62a}$$

$$= 46W_1 + 40W_2 + 38W_3 + 32W_4 + 33W_5 + 24W_6 + 16W_7 + 12W_8 + 14W_9 + 4W_{10} + \mathbf{0W_{11}} + \mathbf{0W_{12}}. \tag{62b}$$

Our main strategy is that for any set of two workers $\mathcal{S} \subseteq [4]$ where $|\mathcal{S}| = \mathsf{N} - \mathsf{N}_r + 1 = 2$, from the transmitted coded messages by the workers in $\mathcal{S}$, the master can recover $U_{[4]\setminus\mathcal{S}}$.

- Assume that the straggler is worker 4. From workers 1 and 2, the master can recover $U_{\{3,4\}}$; from workers 1 and 3, the master can recover $U_{\{2,4\}}$; from workers 2 and 3, the master can recover $U_{\{1,4\}}$. In addition, it can be seen that $U_{\{1,4\}}$, $U_{\{2,4\}}$, and $U_{\{3,4\}}$ are linearly independent. Hence, the master can recover F_1, F_2, and F_3.
- Assume that the straggler is worker 3. The master can recover $U_{\{1,3\}}$, $U_{\{2,3\}}$, and $U_{\{3,4\}}$, which are linearly independent, such that it can recover F_1, F_2, and F_3.
- Assume that the straggler is worker 2. The master can recover $U_{\{1,2\}}$, $U_{\{2,3\}}$, and $U_{\{2,4\}}$, which are linearly independent, such that it can recover F_1, F_2, and F_3.
- Assume that the straggler is worker 1. The master can recover $U_{\{1,2\}}$, $U_{\{1,3\}}$, and $U_{\{1,4\}}$, which are linearly independent, such that it can recover F_1, F_2, and F_3.

In the following, we provide a code construction such that the above strategy can be achieved.

When $\mathcal{S} = \{1,2\}$, workers 1 and 2 should send cooperatively

$$U_{\{3,4\}} = 46W_1 + 40W_2 + 38W_3 + 32W_4 + 33W_5 + 24W_6 + 16W_7 + 12W_8 + 14W_9 + 4W_{10} + \mathbf{0W_{11}} + \mathbf{0W_{12}}.$$

Between workers 1 and 2, it can be seen that W_3, W_4, W_5, and W_6 can only be computed by worker 1, while W_7, W_8, W_9, and W_{10} can only be computed by worker 2. In addition, both workers 1 and 2 can compute W_1 and W_2. Hence, we let worker 1 send

$$A_{1,\{3,4\}} = x_5W_1 + x_6W_2 + 38W_3 + 32W_4 + 33W_5 + 24W_6,$$

and let worker 2 send

$$A_{2,\{3,4\}} = x_{11}W_1 + x_{12}W_2 + 16W_7 + 12W_8 + 14W_9 + 4W_{10},$$

where $A_{1,\{3,4\}} + A_{2,\{3,4\}} = U_{\{3,4\}}$. Note that x_5, x_6, x_{11}, and x_{12} are the coefficients which we can design. Hence, we have

$$x_5 + x_{11} = 46; \tag{63}$$

$$x_6 + x_{12} = 40. \tag{64}$$

Similarly, by considering all sets $\mathcal{S} \subseteq [4]$ where $|\mathcal{S}| = 2$, the transmissions of worker 1 can be expressed as

$$A_{1,\{2,3\}} = 6W_1 + 4W_2 + 6W_3 + 4W_4 + x_1W_5 + x_2W_6, \tag{65}$$

$$A_{1,\{2,4\}} = -48W_1 - 44W_2 + x_3W_3 + x_4W_4 - 21W_5 - 20W_6, \tag{66}$$

$$A_{1,\{3,4\}} = x_5W_1 + x_6W_2 + 38W_3 + 32W_4 + 33W_5 + 24W_6. \tag{67}$$

The transmissions of worker 2 can be expressed as

$$A_{2,\{1,4\}} = -23W_1 - 20W_2 + x_7W_7 + x_8W_8 + 17W_9 + 16W_{10}, \tag{68}$$

$$A_{2,\{1,3\}} = -4W_1 - 4W_2 + 2W_7 + 4W_8 + x_9W_9 + x_{10}W_{10}, \tag{69}$$

$$A_{2,\{3,4\}} = x_{11}W_1 + x_{12}W_2 + 16W_7 + 12W_8$$

$$+ 14W_9 + 4W_{10}. \tag{70}$$

The transmissions of worker 3 can be expressed as

$$A_{3,\{1,2\}} = 4W_3 + 4W_4 + 6W_7 + 8W_8 + x_{13}W_{11} + x_{14}W_{12}, \tag{71}$$

$$A_{3,\{1,4\}} = -13W_3 - 10W_4 + x_{15}W_7 + x_{16}W_8 + 21W_{11} + 30W_{12}, \tag{72}$$

$$A_{3,\{2,4\}} = x_{17}W_3 + x_{18}W_4 - 18W_7 - 12W_8 + 6W_{11} + 16W_{12}. \tag{73}$$

The transmissions of worker 4 can be expressed as

$$A_{4,\{1,2\}} = 11W_5 + 8W_6 + 16W_9 + 12W_{10} + x_{19}W_{11} + x_{20}W_{12}, \tag{74}$$

$$A_{4,\{1,3\}} = 7W_5 + 4W_6 + x_{21}W_9 + x_{22}W_{10} + 10W_{11} + 16W_{12}, \tag{75}$$

$$A_{4,\{2,3\}} = x_{23}W_5 + x_{24}W_6 + 6W_9 + 0W_{10} + 0W_{11} + 4W_{12}. \tag{76}$$

The coefficients of $(x_1, \ldots, x_{12})$ should satisfy (63), (64), and

$$x_1 + x_{23} = 9; \tag{77}$$
$$x_2 + x_{24} = 4; \tag{78}$$
$$x_3 + x_{17} = -36; \tag{79}$$
$$x_4 + x_{18} = -32; \tag{80}$$
$$x_7 + x_{15} = 1; \tag{81}$$
$$x_8 + x_{16} = 6; \tag{82}$$
$$x_9 + x_{21} = 12; \tag{83}$$
$$x_{10} + x_{22} = 8; \tag{84}$$
$$x_{13} + x_{19} = 14; \tag{85}$$
$$x_{14} + x_{20} = 20. \tag{86}$$

Finally, we will introduce how to choose $(x_1, \ldots, x_{12})$ such that the above constraints are satisfied. Meanwhile, the rank of the transmissions of each worker is 2 (i.e., among the three sent sums by each worker, one sum can be obtained by the linear combinations of the other two sums), such that we can let each worker send only two linear combinations of messages and the needed communication cost is $2\mathsf{N}_{\rm r} = 6$, which coincides with the proposed converse bound in Theorem 1.

We let $A_{1,\{2,3\}} + A_{1,\{2,4\}} = A_{1,\{3,4\}}$. Hence, we have

$$x_1 = 54,\ x_2 = 44,\ x_3 = 32,\ x_4 = 28,\ x_5 = -42,\ x_6 = -40.$$

With $x_5 = -42$ and $x_6 = -40$, from (63) and (64) we can see that

$$x_{11} = 88,\ x_{12} = 80.$$

Since we fix $x_{11} = 88$ and $x_{12} = 80$, if the rank of the transmissions of worker 2 is 2, we should have

$$x_7 = -11,\ x_8 = -29/2,\ x_9 = -89/10,\ x_{10} = -7.$$

With $x_3 = 32$ and $x_4 = 28$, from (79) and (80) we can see that

$$x_{17} = -68,\ x_{18} = -60.$$

Since we fix $x_{17} = -68$ and $x_{18} = -60$, if the rank of the transmissions of worker 3 is 2, we should have

$$x_{13} = 6,\ x_{14} = 192/25,\ x_{15} = 12,\ x_{16} = 41/2.$$

With $x_1 = 54$ and $x_2 = 44$, from (77) and (78) we can see that

$$x_{23} = -45,\ x_{24} = -40.$$

Since we fix $x_{23} = -45$ and $x_{24} = -40$, if the rank of the transmissions of worker 4 is 2, we should have

$$x_{19} = 8,\ x_{20} = 308/25,\ x_{21} = 418/20,\ x_{22} = 15.$$

With the above choice of $(x_1, \ldots, x_{12})$, we can find that
$x_7 + x_{15} = -11 + 12 = 1$, satisfying (81);
$x_8 + x_{16} = -29/2 + 41/2 = 6$, satisfying (82);
$x_9 + x_{21} = -89/10 + 418/20 = 12$, satisfying (83);
$x_{10} + x_{22} = -7 + 15 = 8$, satisfying (84);
$x_{13} + x_{19} = 6 + 8 = 14$, satisfying (85);
$x_{14} + x_{20} = 192/25 + 308/25 = 20$, satisfying (86).

In conclusion the above choice of $(x_1, \ldots, x_{12})$ satisfies all constraints in (63), (64), (77)-(86), while the rank of the transmissions of each worker is 2.

Note that the above assignment based on coded caching can only be used for very limited number of cases in our problem, i.e., when $\binom{\mathsf{N}}{\mathsf{N}-\mathsf{N}_{\rm r}+1}$ divides K. In addition, it is part of on-going works to generalize the above computing phase under the coded caching assignment to the general case where $\binom{\mathsf{N}}{\mathsf{N}-\mathsf{N}_{\rm r}+1}$ divides K. $\square$

VI. Conclusions

In this paper, we introduced a distributed linearly separable computation problem and studied the optimal communication cost when the computation cost is minimum. We proposed a converse bound inspired by coded caching converse bounds and an achievable distributed computing scheme based on linear space intersection. The proposed scheme was proved to be optimal under some system parameters. In addition, it was also proved to be optimal under the constraint of the cyclic assignment on the datasets.

Further works include the extension of the proposed scheme to the case where the computation cost is increased, the design of the distributed computing scheme with some improved assignment rather than the cyclic assignment, and novel achievable schemes on specific demand matrices for general case.

Appendix A
Proof of Theorem 1

Recall that the computation cost is minimum, and thus each dataset is assigned to $\mathsf{N} - \mathsf{N}_{\rm r} + 1$ workers. For each set $\mathcal{S} \subseteq [\mathsf{N}]$ where $|\mathcal{S}| = \mathsf{N} - \mathsf{N}_{\rm r} + 1$, we define $\mathcal{G}_{\mathcal{S}}$ as the set of datasets uniquely assigned to all workers in $\mathcal{S}$. For example, in Example 1, $\mathcal{G}_{\{1,2\}} = \{2,5\}$, $\mathcal{G}_{\{1,3\}} = \{1,4\}$, and $\mathcal{G}_{\{2,3\}} = \{3,6\}$.

Let us focus one worker $n \in [\mathsf{N}]$. Since the number of datasets assigned to each worker is $\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N} - \mathsf{N}_{\rm r} + 1)$, we have

$$\sum_{\mathcal{S} \subseteq [\mathsf{N}] : |\mathcal{S}| = \mathsf{N} - \mathsf{N}_{\rm r} + 1, n \in \mathcal{S}} |\mathcal{G}_{\mathcal{S}}| = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N} - \mathsf{N}_{\rm r} + 1). \tag{87}$$

From (87), it can be seen that

$$\max_{\mathcal{S}\subseteq[\mathsf{N}]:|\mathcal{S}|=\mathsf{N}-\mathsf{N}_\mathrm{r}+1,n\in\mathcal{S}}|\mathcal{G}_\mathcal{S}| \geq \left\lceil \frac{\mathsf{K}(\mathsf{N}-\mathsf{N}_\mathrm{r}+1)}{\mathsf{N}\binom{\mathsf{N}-1}{\mathsf{N}-\mathsf{N}_\mathrm{r}}} \right\rceil \tag{88a}$$

$$= \left\lceil \frac{\mathsf{K}}{\binom{\mathsf{N}}{\mathsf{N}-\mathsf{N}_\mathrm{r}+1}} \right\rceil. \tag{88b}$$

In addition, with a slight abuse of notation we define that

$$\mathcal{S}_{\max} = \arg\max_{\mathcal{S}\subseteq[\mathsf{N}]:|\mathcal{S}|=\mathsf{N}-\mathsf{N}_\mathrm{r}+1,n\in\mathcal{S}}|\mathcal{G}_\mathcal{S}| \tag{89}$$

Consider now the set of responding workers $\mathcal{S}_1 = \{n\}\cup([\mathsf{N}]\setminus\mathcal{S}_{\max})$. Note that among the workers in $\mathcal{S}_1$, each dataset D_k where $k\in\mathcal{G}_{\mathcal{S}_{\max}}$ is only assigned to worker n. In addition, since the elements in $\mathbf{F}$ are uniformly i.i.d. over a large enough field, matrix $\mathbf{F}^{(\mathcal{G}_{\mathcal{S}_{\max}})_\mathrm{c}}$ (representing the sub-matrix containing the columns with indices in $\mathcal{G}_{\mathcal{S}_{\max}}$ of $\mathbf{F}$) has the rank equal to $\min\{\mathsf{K}_\mathrm{c},|\mathcal{G}_{\mathcal{S}_{\max}}|\}$ with high probability. In addition, each message has L uniformly i.i.d. symbols. Hence, we have

$$T_n \geq H(X_n) \geq \min\{\mathsf{K}_\mathrm{c},|\mathcal{G}_{\mathcal{S}_{\max}}|\}\,\mathsf{L}. \tag{90}$$

Now we consider each $\mathcal{A}\subseteq[\mathsf{N}]$ where $|\mathcal{A}|=\mathsf{N}_\mathrm{r}$ as the set of responding worker. From the definition of the communication cost in (12), we have

$$\mathsf{R} \geq \frac{\sum_{n_1\in\mathcal{A}}T_{n_1}}{\mathsf{L}} \tag{91a}$$

$$\geq \frac{\mathsf{N}_\mathrm{r}\min\{\mathsf{K}_\mathrm{c},|\mathcal{G}_{\mathcal{S}_{\max}}|\}\,\mathsf{L}}{\mathsf{L}} \tag{91b}$$

$$\geq \mathsf{N}_\mathrm{r}\min\left\{\mathsf{K}_\mathrm{c}, \left\lceil \frac{\mathsf{K}}{\binom{\mathsf{N}}{\mathsf{N}-\mathsf{N}_\mathrm{r}+1}} \right\rceil\right\}, \tag{91c}$$

where (91b) comes from (90) and (91c) comes from (88b). By the definition of the minimum communication cost and the fact that $\mathsf{R}^\star\geq\mathsf{K}_\mathrm{c}$, from (91c) we prove Theorem 1.

Appendix B
Proof of Theorem 4

We fix an integer $n\in[\mathsf{N}]$. By the cyclic assignment described in Section II-A, each dataset $D_{n+p\mathsf{N}}$ where $p\in\left[0:\frac{\mathsf{K}}{\mathsf{N}}-1\right]$ is assigned to $\mathsf{N}-\mathsf{N}_\mathrm{r}+1$ workers. The set of these $\mathsf{N}-\mathsf{N}_\mathrm{r}+1$ workers is

$$\mathcal{S}_1 = \big\{n, \mathrm{Mod}(n-1,\mathsf{N}),\ldots,\mathrm{Mod}(n-\mathsf{N}+\mathsf{N}_\mathrm{r},\mathsf{N})\big\}.$$

Now we assume the set of the responding workers is $\mathcal{R}_1 = \{n\}\cup([\mathsf{N}]\setminus\mathcal{S}_1)$. It can be seen that among the workers in $\mathcal{R}_1$, each dataset D_k where $k\in\left\{n+p\mathsf{N}:p\in\left[0:\frac{\mathsf{K}}{\mathsf{N}}-1\right]\right\}$ is only assigned to worker n. In addition, since the elements in $\mathbf{F}$ are uniformly i.i.d. over a large enough field, matrix $\mathbf{F}^{(\{n+p\mathsf{N}:p\in[0:\frac{\mathsf{K}}{\mathsf{N}}-1]\})_\mathrm{c}}$ has the rank equal to $\min\left\{\mathsf{K}_\mathrm{c},\frac{\mathsf{K}}{\mathsf{N}}\right\}$ with high probability. In addition, each message has L uniformly i.i.d. symbols. Hence, we have

$$T_n \geq H(X_n) \geq \min\left\{\mathsf{K}_\mathrm{c},\frac{\mathsf{K}}{\mathsf{N}}\right\}\mathsf{L}. \tag{92}$$

Now we consider each $\mathcal{A}\subseteq[\mathsf{N}]$ where $|\mathcal{A}|=\mathsf{N}_\mathrm{r}$ as the set of responding worker. We have

$$\mathsf{R} \geq \frac{\sum_{n_1\in\mathcal{A}}T_{n_1}}{\mathsf{L}} \tag{93a}$$

$$\geq \frac{\mathsf{N}_\mathrm{r}\min\left\{\mathsf{K}_\mathrm{c},\frac{\mathsf{K}}{\mathsf{N}}\right\}\mathsf{L}}{\mathsf{L}}, \tag{93b}$$

where (93b) comes from (92). Hence, when $\mathsf{K}_\mathrm{c}\leq\frac{\mathsf{K}}{\mathsf{N}}$, we have $\mathsf{R}\geq\mathsf{N}_\mathrm{r}\mathsf{K}_\mathrm{c}$; when $\mathsf{K}_\mathrm{c}\geq\frac{\mathsf{K}}{\mathsf{N}}$, we have $\mathsf{R}\geq\mathsf{N}_\mathrm{r}\frac{\mathsf{K}}{\mathsf{N}}$. Together with $\mathsf{R}\geq\mathsf{K}_\mathrm{c}$, we obtain the converse bound in Theorem 4.

Appendix C
Proof of Lemma 2

We first focus one $\mathcal{A}\subseteq[\mathsf{N}]$ where $|\mathcal{A}|=\mathsf{N}_\mathrm{r}$. We assume that $\mathcal{A}=\{\mathcal{A}(1),\ldots,\mathcal{A}(\mathsf{N}_\mathrm{r})\}$ where $\mathcal{A}(1)<\cdots<\mathcal{A}(\mathsf{N}_\mathrm{r})$.

Recall that $\mathsf{K}_\mathrm{c}=\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}$ and that the task function is (recall that $(\mathbf{M})_{m\times n}$ indicates that the dimension of matrix $\mathbf{M}$ is $m\times n$)

$$(\mathbf{F})_{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\times\mathsf{K}}([W_1;\ldots;W_\mathsf{K}])_{\mathsf{K}\times\mathsf{L}},$$

where each element in $\mathbf{F}$ is uniformly i.i.d. over large enough finite field $\mathbb{F}_\mathsf{q}$. By the construction of our proposed achievable scheme, each worker $\mathcal{A}(i)$ where $i\in[\mathsf{N}_\mathrm{r}]$ sends

$$\mathbf{C}_{\{\mathcal{A}(i)\}}\mathbf{F}\begin{bmatrix} W_1\\ \vdots\\ W_\mathsf{K}\end{bmatrix} = \begin{bmatrix}\mathbf{u}_{\mathcal{A}(i),1}\\ \vdots\\ \mathbf{u}_{\mathcal{A}(i),\frac{\mathsf{K}}{\mathsf{N}}}\end{bmatrix}\mathbf{F}\begin{bmatrix} W_1\\ \vdots\\ W_\mathsf{K}\end{bmatrix}, \tag{94}$$

where $\mathbf{u}_{\mathcal{A}(i),j}\mathbf{F}^{\left(\overline{\mathcal{Z}_{\mathcal{A}(i)}}\right)_\mathrm{c}} = \mathbf{0}_{1\times\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_\mathrm{r}-1)}$ for each $j\in\left[\frac{\mathsf{K}}{\mathsf{N}}\right]$, and $\overline{\mathcal{Z}_{\mathcal{A}(i)}}\subseteq[\mathsf{K}]$ represents the set of datasets which are not assigned to worker $\mathcal{A}(i)$. To simplify the notations, we let

$$\overline{\mathbf{F}_{\mathcal{A}(i)}} := \mathbf{F}^{\left(\overline{\mathcal{Z}_{\mathcal{A}(i)}}\right)_\mathrm{c}}, \tag{95}$$

with dimension $\mathsf{K}_\mathrm{c}\times\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_\mathrm{r}-1)=\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\times\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_\mathrm{r}-1)$. By some linear transformations on the rows of $\mathbf{C}_{\{\mathcal{A}(i)\}}$ (we will prove very soon that this transformation exists with high probability), we have (96) at the top of the next page. In other words, we let

$$\begin{bmatrix} c_{\mathcal{A}(i),1,\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1} & \cdots & c_{\mathcal{A}(i),1,\frac{\mathsf{K}}{\mathsf{N}}i}\\ \vdots & \ddots & \vdots\\ c_{\mathcal{A}(i),\frac{\mathsf{K}}{\mathsf{N}},\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1} & \cdots & c_{\mathcal{A}(i),\frac{\mathsf{K}}{\mathsf{N}},\frac{\mathsf{K}}{\mathsf{N}}i}\end{bmatrix} = \mathbf{I}_{\frac{\mathsf{K}}{\mathsf{N}}} \tag{97}$$

where $\mathbf{I}_{\frac{\mathsf{K}}{\mathsf{N}}}$ represents the identity matrix with dimension $\frac{\mathsf{K}}{\mathsf{N}}\times\frac{\mathsf{K}}{\mathsf{N}}$.

Recall that $\mathbf{M}^{(\mathcal{S})_\mathrm{r}}$ represents the sub-matrix of $\mathbf{M}$ which is composed of the rows of $\mathbf{M}$ with indices in $\mathcal{S}$. From

$$\mathbf{C}_{\{\mathcal{A}(i)\}}\overline{\mathbf{F}_{\mathcal{A}(i)}} = \mathbf{0}_{\frac{\mathsf{K}}{\mathsf{N}}\times\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_\mathrm{r}-1)}, \tag{98}$$

we have

$$\mathbf{C}_{\{\mathcal{A}(i)\}}^{\left(\left[\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\right]\setminus\left[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1:\frac{\mathsf{K}}{\mathsf{N}}i\right]\right)_\mathrm{c}}\;\overline{\mathbf{F}_{\mathcal{A}(i)}}^{\left(\left[\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\right]\setminus\left[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1:\frac{\mathsf{K}}{\mathsf{N}}i\right]\right)_\mathrm{r}}$$

$$= -\overline{\mathbf{F}_{\mathcal{A}(i)}}^{\left(\left[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1:\frac{\mathsf{K}}{\mathsf{N}}i\right]\right)_\mathrm{r}} := \begin{bmatrix}\overline{\mathbf{f}_{\mathcal{A}(i),\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1}}\\ \vdots\\ \overline{\mathbf{f}_{\mathcal{A}(i),\frac{\mathsf{K}}{\mathsf{N}}i}}\end{bmatrix}, \tag{99}$$

where each vector $\overline{\mathbf{f}_{\mathcal{A}(i),j}}$, $j\in\left[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1:\frac{\mathsf{K}}{\mathsf{N}}i\right]$, is with dimension $1\times\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_\mathrm{r}-1)$.

By the Cramer's rule, it can be seen that

$$c_{\mathcal{A}(i),j,m} = \frac{\det(\mathbf{Y}_{\mathcal{A}(i),j,m})}{\det\left(\overline{\mathbf{F}_{\mathcal{A}(i)}}^{\left(\left[\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\right]\setminus\left[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1:\frac{\mathsf{K}}{\mathsf{N}}i\right]\right)_\mathrm{r}}\right)}, \tag{100}$$

$$\left(\mathbf{C}_{\{\mathcal{A}(i)\}}\right)_{\frac{\mathsf{K}}{\mathsf{N}}\times\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}} = \begin{bmatrix} c_{\mathcal{A}(i),1,1} & c_{\mathcal{A}(i),1,2} & \cdots & c_{\mathcal{A}(i),1,\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}} \\ \vdots & \vdots & \ddots & \vdots \\ c_{\mathcal{A}(i),\frac{\mathsf{K}}{\mathsf{N}},1} & c_{\mathcal{A}(i),\frac{\mathsf{K}}{\mathsf{N}},2} & \cdots & c_{\mathcal{A}(i),\frac{\mathsf{K}}{\mathsf{N}},\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}} \end{bmatrix} \tag{96a}$$

$$= \begin{bmatrix} c_{\mathcal{A}(i),1,1} & \cdots & c_{\mathcal{A}(i),1,\frac{\mathsf{K}}{\mathsf{N}}(i-1)} & 1 & 0 & \cdots & 0 & c_{\mathcal{A}(i),1,\frac{\mathsf{K}}{\mathsf{N}}i+1} & \cdots & c_{\mathcal{A}(i),1,\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}} \\ c_{\mathcal{A}(i),2,1} & \cdots & c_{\mathcal{A}(i),2,\frac{\mathsf{K}}{\mathsf{N}}(i-1)} & 0 & 1 & \cdots & 0 & c_{\mathcal{A}(i),2,\frac{\mathsf{K}}{\mathsf{N}}i+1} & \cdots & c_{\mathcal{A}(i),2,\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_{\mathcal{A}(i),\frac{\mathsf{K}}{\mathsf{N}},1} & \cdots & c_{\mathcal{A}(i),\frac{\mathsf{K}}{\mathsf{N}},\frac{\mathsf{K}}{\mathsf{N}}(i-1)} & 0 & 0 & \cdots & 1 & c_{\mathcal{A}(i),\frac{\mathsf{K}}{\mathsf{N}},\frac{\mathsf{K}}{\mathsf{N}}i+1} & \cdots & c_{\mathcal{A}(i),\frac{\mathsf{K}}{\mathsf{N}},\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}} \end{bmatrix}. \tag{96b}$$

$\forall m \in \left[\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\right] \setminus \left[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1 : \frac{\mathsf{K}}{\mathsf{N}}i\right]$. Assuming m is the s^{th} smallest value in $\left[\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\right] \setminus \left[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1 : \frac{\mathsf{K}}{\mathsf{N}}i\right]$, we define $\mathbf{Y}_{\mathcal{A}(i),j,m}$ as the matrix formed by replacing the s^{th} row of $\overline{\mathbf{F}_{\mathcal{A}(i)}}^{\left(\left[\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\right]\setminus\left[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1:\frac{\mathsf{K}}{\mathsf{N}}i\right]\right)_\mathrm{r}}$ by $\overline{\mathbf{f}_{\mathcal{A}(i),j}}$.

In addition, $\det\left(\overline{\mathbf{F}_{\mathcal{A}(i)}}^{\left(\left[\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\right]\setminus\left[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1:\frac{\mathsf{K}}{\mathsf{N}}i\right]\right)_\mathrm{r}}\right)$ is the determinant of a $\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_\mathrm{r}-1) \times \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_\mathrm{r}-1)$ matrix, which can be viewed as a multivariate polynomial whose variables are the elements in $\mathbf{F}$. Since the elements in $\mathbf{F}$ are uniformly i.i.d. over $\mathbb{F}_\mathsf{q}$, it is with high probability that the multivariate polynomial $\det\left(\overline{\mathbf{F}_{\mathcal{A}(i)}}^{\left(\left[\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\right]\setminus\left[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1:\frac{\mathsf{K}}{\mathsf{N}}i\right]\right)_\mathrm{r}}\right)$ is a non-zero multivariate polynomial (i.e., a multivariate polynomial whose coefficients are not all 0) of degree $\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_\mathrm{r}-1)$. Hence, by the Schwartz-Zippel Lemma [28]–[30], we have

$$\Pr\{c_{\mathcal{A}(i),j,m} \text{ exsits}\}$$
$$= \Pr\left\{\det\left(\overline{\mathbf{F}_{\mathcal{A}(i)}}^{\left(\left[\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\right]\setminus\left[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1:\frac{\mathsf{K}}{\mathsf{N}}i\right]\right)_\mathrm{r}}\right) \text{ is non-zero}\right\} \tag{101a}$$
$$\geq 1 - \frac{\mathsf{K}(\mathsf{N}_\mathrm{r}-1)}{\mathsf{N}\mathsf{q}}. \tag{101b}$$

Note that the above probability (101b) is over all possible realizations of $\mathbf{F}$ whose elements are uniformly i.i.d. over $\mathbb{F}_\mathsf{q}$.

By the probability union bound, we have

$$\Pr\left\{c_{\mathcal{A}(i),j,m} \text{ exsits}, \ \forall i \in [\mathsf{N}_\mathrm{r}], j \in \left[\frac{\mathsf{K}}{\mathsf{N}}\right], m \in \left[\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\right] \setminus \left[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1 : \frac{\mathsf{K}}{\mathsf{N}}i\right]\right\}$$
$$\geq 1 - \frac{\mathsf{K}(\mathsf{N}_\mathrm{r}-1)}{\mathsf{N}\mathsf{q}}\mathsf{N}\frac{\mathsf{K}}{\mathsf{N}}\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_\mathrm{r}-1) \tag{102a}$$
$$= 1 - \frac{\mathsf{K}^3(\mathsf{N}_\mathrm{r}-1)^2}{\mathsf{N}^2\mathsf{q}} \tag{102b}$$
$$\overset{\mathsf{q}\to\infty}{\longrightarrow} 1. \tag{102c}$$

Hence, we prove that the coding matrix of each worker $\mathcal{A}(i)$ where $i \in [\mathsf{N}_\mathrm{r}]$, $\mathbf{C}_{\mathcal{A}(i)}$ in (94), exists with high probability.

In the following, we will prove that matrix

$$\mathbf{C}_\mathcal{A} := \begin{bmatrix} \mathbf{C}_{\mathcal{A}(1)} \\ \vdots \\ \mathbf{C}_{\mathcal{A}(\mathsf{N}_\mathrm{r})} \end{bmatrix} \tag{103}$$

is full-rank with high probability.

Note that $\mathbf{C}_\mathcal{A}$ is a matrix with dimension $\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r} \times \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}$. We expand the determinant of $\mathbf{C}_\mathcal{A}$ as follows,

$$\det(\mathbf{C}_\mathcal{A}) = \sum_{i \in \left[\left(\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\right)!\right]} \frac{P_i}{Q_i}, \tag{104}$$

which contains $\left(\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\right)!$ terms. Each term can be expressed as $\frac{P_i}{Q_i}$, where P_i and Q_i are multivariate polynomials whose variables are the elements in $\mathbf{F}$. From (100), it can be seen that each element in $\mathbf{C}_\mathcal{A}$ is the ratio of two multivariate polynomials whose variables are the elements in $\mathbf{F}$ with degree $\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_\mathrm{r}-1)$. In addition, each term in $\det(\mathbf{C}_\mathcal{A})$ is a multivariate polynomial whose variables are the elements in $\mathbf{C}_\mathcal{A}$ with degree $\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}$. Hence, P_i and Q_i are multivariate polynomials whose variables are the elements in $\mathbf{F}$ with degree $\left(\frac{\mathsf{K}}{\mathsf{N}}\right)^2\mathsf{N}_\mathrm{r}(\mathsf{N}_\mathrm{r}-1)$.

We then let

$$P_\mathcal{A} := \det(\mathbf{C}_\mathcal{A}) \prod_{i \in \left[\left(\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\right)!\right]} Q_i. \tag{105}$$

If $\mathbf{C}_\mathcal{A}$ exists and $P_\mathcal{A} \neq 0$, we have $\det(\mathbf{C}_\mathcal{A}) \neq 0$ and thus $\mathbf{C}_\mathcal{A}$ is full-rank.

To apply the Schwartz-Zippel lemma [28]–[30], we need to guarantee that $P_\mathcal{A}$ is a non-zero multivariate polynomial. To this end, we only need one specific realization of $\mathbf{F}$ so that $P_\mathcal{A} \neq 0$ (or alternatively $\det(\mathbf{C}_\mathcal{A}) \neq 0$ and $Q_i \neq 0$ at the same time). We construct such specific $\mathbf{F}$ in Appendix D such that the following lemma can be proved.

Lemma 3. *For the* $(\mathsf{K},\mathsf{N},\mathsf{N}_\mathrm{r},\mathsf{K}_\mathrm{c},\mathsf{M}) = \left(\mathsf{K},\mathsf{N},\mathsf{N}_\mathrm{r},\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r},\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_\mathrm{r}+1)\right)$ *distributed linearly separable computation problem,* $P_\mathcal{A}$ *in* (105) *is a non-zero multivariate polynomial.* □

Recall that P_i and Q_i are multivariate polynomials with degree $\left(\frac{\mathsf{K}}{\mathsf{N}}\right)^2\mathsf{N}_\mathrm{r}(\mathsf{N}_\mathrm{r}-1)$. Thus the degree of $P_\mathcal{A}$ is less than $\left(\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\right)^2$. Hence, by the Schwartz-Zippel lemma [28]–[30] we have

$$\Pr\{P_\mathcal{A} \neq 0\} \geq 1 - \frac{\left(\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\right)!\left(\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\right)^2}{\mathsf{q}}. \tag{106}$$

Hence, from (102b) and (106), we have

$$\Pr\{\mathbf{C}_\mathcal{A} \text{ is full-rank}\}$$
$$\geq 1 - \Pr\{\mathbf{C}_\mathcal{A} \text{ does not exist}\} - \Pr\{P_\mathcal{A} = 0\} \tag{107a}$$
$$\geq 1 - \frac{\mathsf{K}^3(\mathsf{N}_\mathrm{r}-1)^2}{\mathsf{N}^2\mathsf{q}} - \frac{\left(\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\right)!\left(\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}\right)^2}{\mathsf{q}}. \tag{107b}$$

Finally, by considering all $\mathcal{A} \subseteq [\mathsf{N}]$ where $|\mathcal{A}| = \mathsf{N}_\mathrm{r}$, we have

$$\Pr\{\mathbf{C}_\mathcal{A} \text{ is full-rank}, \ \forall \mathcal{A} \subseteq [\mathsf{N}] : |\mathcal{A}| = \mathsf{N}_\mathrm{r}\} \tag{108a}$$
$$\geq 1 - \sum_{\mathcal{A} \subseteq [\mathsf{N}]:|\mathcal{A}|=\mathsf{N}_\mathrm{r}} \Pr\{\mathbf{C}_\mathcal{A} \text{ is not full-rank}\} \tag{108b}$$
$$\geq 1 - \binom{\mathsf{N}}{\mathsf{N}_\mathrm{r}} \left(\frac{\mathsf{K}^3(\mathsf{N}_\mathrm{r}-1)^2}{\mathsf{N}^2\mathsf{q}} + \frac{(\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r})!\,(\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r})^2}{\mathsf{q}} \right) \tag{108c}$$
$$\stackrel{\mathsf{q}\to\infty}{\longrightarrow} 1. \tag{108d}$$

Hence, we prove Lemma 2.

Appendix D
Proofs of Lemma 3

A. $\mathsf{N} = \mathsf{K}$

We first consider the case where $\mathsf{N} = \mathsf{K}$. We aim to construct one demand matrix $\mathbf{F}$ where $\det(\mathbf{C}_\mathcal{A}) \neq 0$, such that we can prove Lemma 3 for this case.

Note that when $\mathsf{N} = \mathsf{K}$, we have that $\mathsf{K}_\mathrm{c} = \frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r} = \mathsf{N}_\mathrm{r}$ and that the dimension of $\mathbf{F}$ is $\mathsf{N}_\mathrm{r} \times \mathsf{N}$. We construct an $\mathbf{F}$ such that for each $i \in [\mathsf{N}_\mathrm{r}]$ and $j \in \overline{\mathcal{Z}_{\mathcal{A}(i)}}$, the element located at the i^{th} row and the j^{th} column is 0. Recall that the number of datasets which are not assigned to each worker is $|\overline{\mathcal{Z}_{\mathcal{A}(i)}}| = \mathsf{N}_\mathrm{r} - 1$ and that by the cyclic assignment, the elements in $\overline{\mathcal{Z}_{\mathcal{A}(i)}}$ are adjacent; thus the i^{th} row of $\mathbf{F}$ can be expressed as follows,

$$\mathbf{F}^{(\{i\})_\mathrm{r}} = [*, *, \cdots, *, 0, 0, \cdots, 0, *, *, \cdots, *], \tag{109}$$

where the number of adjacent '0' in (109) is $\mathsf{N}_\mathrm{r} - 1$ and each '$*$' represents a symbol uniformly i.i.d. over $\mathbf{F}_\mathsf{q}$.

To prove that $\mathcal{P}(\mathcal{A})$ in (105) is non-zero, we need to prove

1) $\det\left(\overline{\mathbf{F}_{\mathcal{A}(i)}}^{([\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}]\setminus[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1:\frac{\mathsf{K}}{\mathsf{N}}i])_\mathrm{r}}\right) \neq 0$ for each $i \in [\mathsf{N}_\mathrm{r}]$, such that $\mathbf{C}_\mathcal{A}$ exists (see (100)); thus $\prod_{i \in [(\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r})!]} Q_i \neq 0$.
2) $\det(\mathbf{C}_\mathcal{A}) \neq 0$.

First, we prove that $\mathbf{C}_\mathcal{A}$ exists. We focus on worker $\mathcal{A}(i)$ where $i \in [\mathsf{N}_\mathrm{r}]$. Matrix $\overline{\mathbf{F}_{\mathcal{A}(i)}}^{([\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}]\setminus[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1:\frac{\mathsf{K}}{\mathsf{N}}i])_\mathrm{r}}$ is with dimension $(\mathsf{N}_\mathrm{r} - 1) \times (\mathsf{N}_\mathrm{r} - 1)$. Each row of $\overline{\mathbf{F}_{\mathcal{A}(i)}}^{([\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}]\setminus[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1:\frac{\mathsf{K}}{\mathsf{N}}i])_\mathrm{r}}$ corresponds to one worker in $\mathcal{A} \setminus \{\mathcal{A}(i)\}$. There are three cases:

- if this worker is $\mathrm{Mod}(\mathcal{A}(i) + j, \mathsf{N})$ where $j \in [\mathsf{N}_\mathrm{r} - 2]$, the corresponding row is
$$[*, \cdots, *, 0, \cdots, 0],$$
where the number of '$*$' is j and the number of '0' is $\mathsf{N}_\mathrm{r} - 1 - j$;
- if this worker is $\mathrm{Mod}(\mathcal{A}(i) - j, \mathsf{N})$ where $j \in [\mathsf{N}_\mathrm{r} - 2]$, the corresponding row is
$$[0, \cdots, 0, *, \cdots, *],$$
where the number of '0' is j and the number of '$*$' is $\mathsf{N}_\mathrm{r} - 1 - j$;
- otherwise, the corresponding row is
$$[*, \cdots, *].$$

By the above observation, it can be seen that each column of $\overline{\mathbf{F}_{\mathcal{A}(i)}}^{([\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}]\setminus[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1:\frac{\mathsf{K}}{\mathsf{N}}i])_\mathrm{r}}$ contains at most $(\mathsf{N}_\mathrm{r} - 2)$ '0', and that there does not exist two columns with $(\mathsf{N}_\mathrm{r} - 2)$ '0' where these two columns have the same form (i.e., the positions of '0' are the same). Hence, with some row permutation on rows, we can let the elements located at the right-diagonal of $\overline{\mathbf{F}_{\mathcal{A}(i)}}^{([\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}]\setminus[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1:\frac{\mathsf{K}}{\mathsf{N}}i])_\mathrm{r}}$ are all '$*$'. In other words, $\det\left(\overline{\mathbf{F}_{\mathcal{A}(i)}}^{([\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}]\setminus[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1:\frac{\mathsf{K}}{\mathsf{N}}i])_\mathrm{r}}\right)$ is a non-zero multivariate polynomial where each '$*$' in $\overline{\mathbf{F}_{\mathcal{A}(i)}}^{([\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}]\setminus[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1:\frac{\mathsf{K}}{\mathsf{N}}i])_\mathrm{r}}$ is a variable uniformly i.i.d. over $\mathbb{F}_\mathsf{q}$. By the Schwartz-Zippel lemma [28]–[30], it can be seen that

$$\Pr\left\{\det\left(\overline{\mathbf{F}_{\mathcal{A}(i)}}^{([\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}]\setminus[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1:\frac{\mathsf{K}}{\mathsf{N}}i])_\mathrm{r}}\right) \neq 0\right\} \stackrel{\mathsf{q}\to\infty}{\longrightarrow} 1. \tag{110}$$

By the probability union bound, we have

$$\Pr\left\{\det\left(\overline{\mathbf{F}_{\mathcal{A}(i)}}^{([\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}]\setminus[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1:\frac{\mathsf{K}}{\mathsf{N}}i])_\mathrm{r}}\right) \neq 0, \ \forall i \in [\mathsf{N}_\mathrm{r}]\right\} \stackrel{\mathsf{q}\to\infty}{\longrightarrow} 1. \tag{111}$$

Hence, there must exist some $\mathbf{F}$ such that $\det\left(\overline{\mathbf{F}_{\mathcal{A}(i)}}^{([\frac{\mathsf{K}}{\mathsf{N}}\mathsf{N}_\mathrm{r}]\setminus[\frac{\mathsf{K}}{\mathsf{N}}(i-1)+1:\frac{\mathsf{K}}{\mathsf{N}}i])_\mathrm{r}}\right) \neq 0$ for each $i \in [\mathsf{N}_\mathrm{r}]$; thus we finish the proof on the existence of $\mathbf{C}_\mathcal{A}$.

Next, we prove the proposed scheme is decodable. Obviously,

$$\mathbf{F}^{(\{i\})_\mathrm{r}} \begin{bmatrix} W_1 \\ \vdots \\ W_\mathsf{N} \end{bmatrix}$$

can be sent by worker $\mathcal{A}(i)$. With $\mathsf{N} = \mathsf{K}$, each worker sends $\frac{\mathsf{K}}{\mathsf{N}} = 1$ linear combination of messages. By the construction, we can see that for each $i \in [\mathsf{N}_\mathrm{r}]$, the coding matrix is

$$\mathbf{C}_{\mathcal{A}(i)} = [0, \cdots, 0, 1, 0, \cdots, 0], \tag{112}$$

where 1 is located at the i^{th} column and the dimension of $\mathbf{C}_{\mathcal{A}(i)}$ is $1 \times \mathsf{N}_\mathrm{r}$. Hence, it can be seen that

$$\mathbf{C}_\mathcal{A} = \begin{bmatrix} \mathbf{C}_{\mathcal{A}(1)} \\ \vdots \\ \mathbf{C}_{\mathcal{A}(\mathsf{N}_\mathrm{r})} \end{bmatrix} \tag{113}$$

is an identity matrix and is thus full-rank, i.e., $\det(\mathbf{C}_\mathcal{A}) \neq 0$.

B. N *divides* K

Let us then focus on the $(\mathsf{K}, \mathsf{N}, \mathsf{N}_\mathrm{r}, \mathsf{K}_\mathrm{c}, \mathsf{M}) = \big(\mathsf{a}\mathsf{N}, \mathsf{N}, \mathsf{N}_\mathrm{r}, \mathsf{a}\mathsf{N}_\mathrm{r}, \mathsf{a}(\mathsf{N} - \mathsf{N}_\mathrm{r} + 1)\big)$ distributed linearly separable computation problem, where a is a positive integer. Similarly, we also aim to construct one demand matrix $\mathbf{F}$ where $\det(\mathbf{C}_\mathcal{A}) \neq 0$.

More precisely, we let (recall that $\mathbf{0}_{m\times n}$ represents the zero matrix with dimension $m \times n$; $(\mathbf{M})_{m\times n}$ represents the dimension of matrix $\mathbf{M}$ is $m \times n$)

$$\mathbf{F} = \left[\begin{array}{c|c|c|c} (\mathbf{F}_1)_{\mathsf{N}_\mathrm{r}\times\mathsf{N}} & \mathbf{0}_{\mathsf{N}_\mathrm{r}\times\mathsf{N}} & \cdots & \mathbf{0}_{\mathsf{N}_\mathrm{r}\times\mathsf{N}} \\ \hline \mathbf{0}_{\mathsf{N}_\mathrm{r}\times\mathsf{N}} & (\mathbf{F}_2)_{\mathsf{N}_\mathrm{r}\times\mathsf{N}} & \cdots & \mathbf{0}_{\mathsf{N}_\mathrm{r}\times\mathsf{N}} \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline \mathbf{0}_{\mathsf{N}_\mathrm{r}\times\mathsf{N}} & \mathbf{0}_{\mathsf{N}_\mathrm{r}\times\mathsf{N}} & \cdots & (\mathbf{F}_\mathsf{a})_{\mathsf{N}_\mathrm{r}\times\mathsf{N}} \end{array}\right], \tag{114}$$

where each element in $\mathbf{F}_i, i \in [\mathsf{a}]$, is uniformly i.i.d. over $\mathbb{F}_\mathsf{q}$. In the above construction, the $(\mathsf{K}, \mathsf{N}, \mathsf{N}_\mathrm{r}, \mathsf{K}_\mathrm{c}, \mathsf{M}) = \big(\mathsf{a}\mathsf{N}, \mathsf{N}, \mathsf{N}_\mathrm{r}, \mathsf{a}\mathsf{N}_\mathrm{r}, \mathsf{a}(\mathsf{N} - \mathsf{N}_\mathrm{r} + 1)\big)$ distributed linearly separable computation problem is divided into a independent $(\mathsf{K}, \mathsf{N}, \mathsf{N}_\mathrm{r}, \mathsf{K}_\mathrm{c}, \mathsf{M}) = (\mathsf{N}, \mathsf{N}, \mathsf{N}_\mathrm{r}, \mathsf{N}_\mathrm{r}, \mathsf{N} - \mathsf{N}_\mathrm{r} + 1)$ distributed linearly separable computation sub-problems. In each sub-problem, assuming that the coding matrix of the workers in $\mathcal{A}$ is $\mathbf{C}'_\mathcal{A}$, from Appendix D-A, we have $\mathbf{C}'_\mathcal{A} \neq 0$ with high probability. Hence, in the $(\mathsf{K}, \mathsf{N}, \mathsf{N}_\mathrm{r}, \mathsf{K}_\mathrm{c}, \mathsf{M}) = \big(\mathsf{a}\mathsf{N}, \mathsf{N}, \mathsf{N}_\mathrm{r}, \mathsf{a}\mathsf{N}_\mathrm{r}, \mathsf{a}(\mathsf{N} - \mathsf{N}_\mathrm{r} + 1)\big)$ distributed linearly separable computation problem with the constructed $\mathbf{F}$ in (114), we also have that $\det(\mathbf{C}_\mathcal{A}) \neq 0$ with high probability.

Appendix E
An Allocation Algorithm for the Cyclic Assignment in the General Case

Recall that our objective is to choose b datasets from N effective datasets as the real datasets, such that by the cyclic assignment on these N effective datasets the number of real datasets assigned to each worker is no more than $\left\lceil \frac{\mathsf{N}-\mathsf{N}_\mathrm{r}+1}{\lfloor \frac{\mathsf{N}}{\mathsf{b}} \rfloor} \right\rceil$. By the cyclic assignment, each effective dataset (denoted by E_k where $k \in [\mathsf{N}]$) is assigned to workers in $\big\{\mathrm{Mod}(k, \mathsf{N}), \mathrm{Mod}(k-1, \mathsf{N}), \ldots, \mathrm{Mod}(k - \mathsf{N} + \mathsf{N}_\mathrm{r}, \mathsf{N})\big\}$. The set of effective datasets assigned to worker $n \in [\mathsf{N}]$ is $\big\{\mathrm{Mod}(n, \mathsf{N}), \mathrm{Mod}(n+1, \mathsf{N}), \ldots, \mathrm{Mod}(n + \mathsf{N} - \mathsf{N}_\mathrm{r}, \mathsf{N})\big\}$. We propose an algorithm based on the following integer decomposition.

We decompose the integer $\mathsf{N} - \mathsf{b}$ into b parts, $\mathsf{N} - \mathsf{b} = p_1 + \cdots + p_\mathsf{b}$, where $p_1 \leq \cdots \leq p_\mathsf{b}$ and p_i is either $\lceil \frac{\mathsf{N}-\mathsf{b}}{\mathsf{b}} \rceil$ or $\lfloor \frac{\mathsf{N}-\mathsf{b}}{\mathsf{b}} \rfloor$ for each $i \in [\mathsf{b}]$. More precisely, by defining $\alpha = \mathsf{b} \lceil \frac{\mathsf{N}-\mathsf{b}}{\mathsf{b}} \rceil - (\mathsf{N} - \mathsf{b})$, we let

$$p_1 = \cdots = p_\alpha = \left\lfloor \frac{\mathsf{N} - \mathsf{b}}{\mathsf{b}} \right\rfloor; \tag{115a}$$

$$p_{\alpha+1} = \cdots = p_\mathsf{b} = \left\lceil \frac{\mathsf{N} - \mathsf{b}}{\mathsf{b}} \right\rceil. \tag{115b}$$

We then choose datasets

$$E_1,\ E_{2+p_1},\ E_{3+p_1+p_2}, \ldots,\ E_{\mathsf{b}+p_1+\cdots+p_{\mathsf{b}-1}}$$

as the real datasets. It can be seen that between each two real datasets, there are at least $\lfloor \frac{\mathsf{N}-\mathsf{b}}{\mathsf{b}} \rfloor$ virtual datasets. Hence, in each adjacent $\mathsf{N} - \mathsf{N}_\mathrm{r} + 1$ datasets, there are at most

$$\left\lceil \frac{\mathsf{N} - \mathsf{N}_\mathrm{r} + 1}{\lfloor \frac{\mathsf{N}-\mathsf{b}}{\mathsf{b}} + 1 \rfloor} \right\rceil = \left\lceil \frac{\mathsf{N} - \mathsf{N}_\mathrm{r} + 1}{\lfloor \frac{\mathsf{N}}{\mathsf{b}} \rfloor} \right\rceil$$

real datasets. Hence, we prove that by the above choice, the number of real datasets assigned to each worker is no more than $\left\lceil \frac{\mathsf{N}-\mathsf{N}_\mathrm{r}+1}{\lfloor \frac{\mathsf{N}}{\mathsf{b}} \rfloor} \right\rceil$.

References

[1] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[2] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, I. Stoica *et al.*, "Spark: Cluster computing with working sets." *HotCloud*, vol. 10, no. 10-10, p. 95, 2010.

[3] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, Q. V. Le, and A. Y. Ng, "Large scale distributed deep networks," *in Advances in Neural Information Processing Systems (NIPS)*, pp. 1223–1231, 2012.

[4] 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, Mar. 2018.

[5] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, "A fundamental tradeoff between computation and communication in distributed computing," *IEEE Trans. Inf. Theory*, vol. 64, no. 1, pp. 109–128, Jan. 2018.

[6] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "A unified coding framework for distributed computing with straggling servers," *in IEEE Global Communications Conference Workshops (GLOBECOM)*, pp. 1–6, 2016.

[7] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Infor. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[8] M. Ji, G. Caire, and A. Molisch, "Fundamental limits of caching in wireless D2D networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 1, pp. 849–869, 2016.

[9] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3368–3376.

[10] M. Ye and E. Abbe, "Communication-computation efficient gradient coding," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5610–5619.

[11] N. Raviv, R. Tandon, A. Dimakis, and I. Tamo, "Gradient coding from cyclic MDS codes and expander graphs," *in Proc. Int. Conf. on Machine Learning (ICML)*, pp. 4302–4310, Jul. 2018.

[12] W. Halbawi, N. Azizan-Ruhi, F. Salehi, and B. Hassibi, "Improving distributed gradient descent using reed-solomon codes," *in IEEE Int. Symp. Inf. Theory (ISIT)*, pp. 2027–2031, Jun. 2018.

[13] S. Dutta, V. Cadambe, and P. Grover, ""Short-dot": Computing large linear transforms distributedly using coded short dot products," *IEEE Transactions on Information Theory*, vol. 65, no. 10, pp. 6171–6193, 2019.

[14] A. Ramamoorthy, L. Tang, and P. O. Vontobel, "Universally decodable matrices for distributed matrix-vector multiplication," *in IEEE Int. Symp. Inf. Theory (ISIT)*, pp. 1777–1781, Jul. 2019.

[15] A. B. Das and A. Ramamoorthy, "Distributed matrix-vector multiplication: A convolutional coding approach," *in IEEE Int. Symp. Inf. Theory (ISIT)*, pp. 3022–3026, Jul. 2019.

[16] F. Haddadpour and V. R. Cadambe, "Codes for distributed finite alphabet matrix-vector multiplication," *in IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2018.

[17] K. Lee, C. Suh, and K. Ramchandran, "High-dimensional coded matrix multiplication," *in IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017.

[18] S. Wang, J. Liu, and N. Shroff, "Coded sparse matrix multiplication," *in Proc. 35th Intl. Conf. on Mach. Learning (ICML)*, pp. 5139–5147, 2018.

[19] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Polynomial codes: an optimal design for high-dimensional coded matrix multiplication," *in Advances in Neural Information Processing Systems (NIPS)*, pp. 4406–4416, 2017.

[20] ——, "Straggler mitigation in distributed matrix multiplication: Fundamental limits and optimal coding," *IEEE Trans. Infor. Theory*, vol. 66, no. 3, pp. 1920–1933, Mar. 2020.

[21] S. Dutta, M. Fahim, F. Haddadpour, H. Jeong, V. Cadambe, and P. Grover, "On the optimal recovery threshold of coded matrix multiplication," *IEEE Trans. Infor. Theory*, vol. 66, no. 1, pp. 278–301, Jan. 2020.

[22] A. Ramamoorthy, A. B. Das, and L. Tang, "Straggler-resistant distributed matrix computation via coding theory: Removing a bottleneck in large-scale data processing," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 136–145, May 2020.

[23] Z. Jia and S. A. Jafar, "Cross subspace alignment codes for coded distributed batch computation," *IEEE Trans. Infor. Theory*, Mar. 2021.

[24] Q. Yu, S. Li, N. Raviv, S. M. M. Kalan, M. Soltanolkotabi, and S. A. Avestimehr, "Lagrange coded computing: Optimal design for resiliency, security, and privacy," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 1215–1225.

[25] K. Wan, D. Tuninetti, and P. Piantanida, "An index coding approach to caching with uncoded cache placement," *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1318–1332, Mar. 2020.

[26] Q. Yu, M. A. Maddah-Ali, and S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Trans. Infor. Theory*, vol. 64, no. 2, pp. 1281–1296, Feb. 2018.

[27] A. Behrouzi-Far and E. Soljanin, "Efficient replication for straggler mitigation in distributed computing," *available at arXiv:2006.02318*, Jun. 2020.

[28] J. T. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," *Journal of the ACM (JACM)*, vol. 27, no. 4, pp. 701–717, 1980.

[29] R. Zippel, "Probabilistic algorithms for sparse polynomials," in *International symposium on symbolic and algebraic manipulation*. Springer, 1979, pp. 216–226.

[30] R. A. Demillo and R. J. Lipton, "A probabilistic remark on algebraic program testing," *Information Processing Letters*, vol. 7, no. 4, pp. 193–195, 1978.

[31] S. Kadhe, O. O. Koyluoglu, and K. Ramchandran, "Communication-efficient gradient coding for straggler mitigation in distributed learning," *arXiv:2005.07184*, May. 2020.

[32] A. Reisizadeh, S. Prakash, R. Pedarsani, and A. S. Avestimehr, "Tree gradient coding," *in IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2019.

[33] S. Li, S. M. M. Kalan, A. S. Avestimehr, and M. Soltanolkotabi, "Near-optimal straggler mitigation for distributed gradient methods," *in IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 857–866, 2018.

[34] E. Ozfatura, S. Ulukus, and D. Gunduz, "Straggler-aware distributed learning: Communication computation latency trade-off," *Entropy 2020, 22(5), 544*.

[35] G. Suh, K. Lee, and C. Suh, "Matrix sparsification for coded matrix multiplication," *in 55th Allerton Conf. Commun., Control, Comp.*, pp. 1271–1278, Oct. 2017.

[36] S. Wang, J. Liu, N. Shroff, and P. Yang, "Fundamental limits of coded linear transform," *available at arXiv:1804.09791*, Apr. 2018.

Kai Wan (S '15 – M '18) received the B.E. degree in Optoelectronics from Huazhong University of Science and Technology, China, in 2012, the M.Sc. and Ph.D. degrees in Communications from Université Paris-Saclay, France, in 2014 and 2018. He is currently a post-doctoral researcher with the Communications and Information Theory Chair (CommIT) at Technische Universität Berlin, Berlin, Germany. His research interests include information theory, coding techniques, and their applications on coded caching, index coding, distributed storage, distributed computing, wireless communications, privacy and security. He has served as an Associate Editor of IEEE Communications Letters from Aug. 2021.

Hua Sun (S '12 – M '17) received the B.E. degree in Communications Engineering from Beijing University of Posts and Telecommunications, China, in 2011, and the M.S. degree in Electrical and Computer Engineering and the Ph.D. degree in Electrical Engineering from University of California Irvine, USA, in 2013 and 2017, respectively. He is an Assistant Professor in the Department of Electrical Engineering at the University of North Texas, USA. His research interests include information theory and its applications to communications, privacy, security, and storage.

Dr. Sun is a recipient of the NSF CAREER award in 2021, and the UNT College of Engineering Distinguished Faculty Fellowship in 2021. His co-authored papers received the IEEE Jack Keil Wolf ISIT Student Paper Award in 2016, and an IEEE GLOBECOM Best Paper Award in 2016.

Mingyue Ji (S '09 – M '15) received the B.E. in Communication Engineering from Beijing University of Posts and Telecommunications (China), in 2006, the M.Sc. degrees in Electrical Engineering from Royal Institute of Technology (Sweden) and from University of California, Santa Cruz, in 2008 and 2010, respectively, and the PhD from the Ming Hsieh Department of Electrical Engineering at University of Southern California in 2015. He subsequently was a Staff II System Design Scientist with Broadcom Corporation (Broadcom Limited) in 2015-2016. He is now an Assistant Professor of Electrical and Computer Engineering Department and an Adjunct Assistant Professor of School of Computing at the University of Utah. He received the IEEE Communications Society Leonard G. Abraham Prize for the best IEEE JSAC paper in 2019, the best paper award in IEEE ICC 2015 conference, the best student paper award in IEEE European Wireless 2010 Conference and USC Annenberg Fellowship from 2010 to 2014. He has served as an Associate Editor of IEEE Transactions on Communications from 2020. He is interested the broad area of information theory, coding theory, concentration of measure and statistics with the applications of caching networks, wireless communications, distributed storage and computing systems, distributed machine learning, and (statistical) signal processing.

Giuseppe Caire (S '92 – M '94 – SM '03 – F '05) was born in Torino in 1965. He received the B.Sc. in Electrical Engineering from Politecnico di Torino in 1990, the M.Sc. in Electrical Engineering from Princeton University in 1992, and the Ph.D. from Politecnico di Torino in 1994. He has been a post-doctoral research fellow with the European Space Agency (ESTEC, Noordwijk, The Netherlands) in 1994-1995, Assistant Professor in Telecommunications at the Politecnico di Torino, Associate Professor at the University of Parma, Italy, Professor with the Department of Mobile Communications at the Eurecom Institute, Sophia-Antipolis, France, a Professor of Electrical Engineering with the Viterbi School of Engineering, University of Southern California, Los Angeles, and he is currently an Alexander von Humboldt Professor with the Faculty of Electrical Engineering and Computer Science at the Technical University of Berlin, Germany.

He received the Jack Neubauer Best System Paper Award from the IEEE Vehicular Technology Society in 2003, the IEEE Communications Society and Information Theory Society Joint Paper Award in 2004 and in 2011, the Okawa Research Award in 2006, the Alexander von Humboldt Professorship in 2014, the Vodafone Innovation Prize in 2015, an ERC Advanced Grant in 2018, the Leonard G. Abraham Prize for best IEEE JSAC paper in 2019, the IEEE Communications Society Edwin Howard Armstrong Achievement Award in 2020, and he is a recipient of the 2021 Leibinz Prize of the German National Science Foundation (DFG). Giuseppe Caire is a Fellow of IEEE since 2005. He has served in the Board of Governors of the IEEE Information Theory Society from 2004 to 2007, and as officer from 2008 to 2013. He was President of the IEEE Information Theory Society in 2011. His main research interests are in the field of communications theory, information theory, channel and source coding with particular focus on wireless communications.