Fundamental Limits of Distributed Computing for Linearly Separable Functions

K. K. Krishnan Namboodiri, Elizabath Peter, Derya Malak, and Petros Elia

Abstract

This work addresses the problem of distributed computation of linearly separable functions, where a master node with access to K datasets, employs N servers to compute L user-requested functions, each defined over the datasets. Servers are instructed to compute subfunctions of the datasets and must communicate computed outputs to the user, who reconstructs the requested outputs. The central challenge is to reduce the per-server computational load and the communication cost from servers to the user, while ensuring recovery for any possible set of L demanded functions.

We here establish the fundamental communication–computation tradeoffs for arbitrary K and L, through novel task-assignment and communication strategies that, under the linear-encoding and no-subpacketization assumptions, are proven to be either exactly optimal or within a factor of three from the optimum. In contrast to prior approaches that relied on fixed assignments of tasks – either disjoint or cyclic assignments – our key innovation is a nullspace-based design that jointly governs task assignment and server transmissions, ensuring exact decodability for all demands, and attaining *optimality over all assignment and delivery methods*. To prove this optimality, we here uncover a duality between nullspaces and sparse matrix factorizations, enabling us to recast the distributed computing problem as an equivalent factorization task and derive a sharp information-theoretic converse bound. Building on this, we establish an additional converse that, for the first time, links the communication cost to the covering number from the theory of *general covering designs*.

Index Terms

Coded distributed computation, linearly separable functions, covering designs, matrix factorization, communication-computation tradeoff.

This research was partially supported by European Research Council ERC-StG Project SENSIBILITÉ under Grant 101077361, the ERC-PoC Project LIGHT under Grant 101101031, by the Huawei France-Funded Chair Toward Future Wireless Networks, and by the Program "PEPR Networks of the Future" of France 2030.

The authors are with the Communication Systems Department, EURECOM, Sophia Antipolis, FRANCE ({karakkad,peter,malak,elia}@eurecom.fr). Manuscript last revised: December 8, 2025.

I. Introduction

Recent advances in machine learning have rendered distributed computing indispensable for managing increasingly intensive computing workloads. This distributed paradigm – as exemplified by frameworks such as MapReduce [1] and Spark [2] – asks for decomposition of large-scale computational tasks into smaller subtasks and their execution in parallel across multiple servers, thereby overcoming the limited capacity of individual nodes to process massive datasets.

Despite their success, distributed computing systems face several fundamental challenges such as having to contend with unreliable or slow computing nodes (so-called stragglers [3]–[8]), protecting the privacy of sensitive datasets, and guarding against adversarial behavior by malicious nodes [9]–[12]. Beyond these issues of reliability and security, two intrinsic constraints dominate system design: servers have finite computational power, and communication links have finite capacity. These intertwined constraints bring to the fore the well-known *communication–computation tradeoff*, which lies at the heart of distributed computing across diverse settings such as MapReduce [13]–[19], gradient coding [20]–[22], and distributed matrix multiplication [6], [7], [23], [24], to mention just a few. These same constraints have inspired the development of various coding-theoretic techniques, proposed to reduce communication load, indeed often at the expense of increased computation or storage at the servers [13], [14], [19], [25].

This same fundamental tradeoff lies at the core of our study, where we examine the (K, L, M) distributed linearly separable function computation problem over a finite[1] field $\mathbb{F}_q$. This problem motivates us as it arises in many domains such as signal processing, machine learning, control theory [26]–[28], etc. In this setting, a user requests L function values from a master node, each corresponding to the output of an independent function $F_\ell, \ell \in [L]$, acting on a dataset library $\mathcal{W}$ consisting of K independent and identically distributed (i.i.d.) datasets. As we clarify later, these functions are linearly separable over K basis subfunctions $f_j, j \in [K]$ of the datasets, enabling the master node to parallelize the computations across N distributed servers. Each server is assigned a subset of no more than M chosen subfunctions to compute, each evaluated on a dataset stored by that server. Each server is thus said to enjoy a finite computational capability M. After completing their computations, the servers transmit linear combinations of their computed values to the user, who then must recover the L requested function outputs. Thus, the objective, for any given (K, L, M) instance of the problem, is to design a distributed computing scheme

[1] We will show later on that many of our results hold for any field including the field of real and complex numbers.

that best allocates tasks across the servers, and best provides communication schemes, in order to minimize the communication cost needed for delivering the desired functions.

To put this objective into context, and to get a sense of the corresponding tradeoff between computation (M) and communication, we quickly note that as one might expect, having $M = K$ maps to a centralized setting, where the single server makes L transmissions to serve the L functions. On the other extreme, having $M = 1$ corresponds to a scenario where each server can compute only one subfunction, thus forcing $N = K$ transmitting servers, and a maximum communication cost of K. We aim to identify and meet the optimal tradeoff in between these two extreme scenarios, by properly assigning the subfunctions to the servers in a way that minimizes the total amount of data transferred between the servers and the user.

A. Related Works

Our setting enjoys numerous connections with various themes of distributed computing. We proceed to mention some of these connections, while also clarifying key differentiating factors.

Our work is most closely related to [25], [29], which study a similar problem of linearly separable function computation with N servers, a single user, multiple requested functions over K datasets, and an objective of minimizing communication cost for a given M. The main distinction with our work is that [25], [29] emphasize on straggler mitigation, and on the specific case of the cyclic task assignment; based on this cyclic structure – and always with a focus on accounting for stragglers – they proceed to design novel communication schemes as well as useful information-theoretic converses. Related work can also be found in the more recent [30], [31], which consider the *multi-user version* of the problem in [29] without stragglers, where now each server is connected to multiple, but not all, users, each asking for their own desired function. Both these works [30], [31] aim to reduce the communication and computation cost, the first by exploiting the properties of covering codes, and the second by employing optimal tilings from tessellation theory. In this multi-user setting, the first work provides optimality under the restrictive assumption that all possible demands are simultaneously requested by the users, and the second work – once translated onto our single-user setting – operates under various restrictive assumptions of disjoint tasks across servers. These assumptions do not appear in our work.

Our problem also has strong natural links to *distributed source coding for function computation*, which often relates to our setting; Multiple sources (that play the role of servers with access to datasets) communicate via parallel channels to a destination (our user) who wishes to compute various functions of the data at the sources. Such works (cf. [32]–[36]) make significant progress on the challenging case of correlated source data (which, in our setting, corresponds to datasets or task outputs), by designing efficient compression schemes that exploit these correlations while taking into account the structure of the requested functions. In particular, various works focus on identifying the *function computation capacity*, corresponding to the maximum rate at which distributed encoders (servers) compress their observations so that a decoder recovers a function of sources. Key works include [37], which introduced cut-set bounds for independent sources, and [38], which refined the approach to yield tighter bounds. More recent works include [39] on the two-source arithmetic sum under channel asymmetry, and [40] (see also [41]) on multiple sources and general functions. Another related direction concerns the compression of vector-linear functions. For instance, [42] studies the problem where a single user aims to recover multiple linear combinations of the sources (losslessly or with loss), designing schemes based on nested linear codes, while [43] provides capacity characterizations under specific source–decoder connectivity, emphasizing on specific instances (e.g., $K = 3$, $N = 2$). The related literature is broad, and while our overview is necessarily selective, all prior approaches remain fundamentally different from ours; In addition to the fact that such studies often (though not always) focus on the special case $M = 1$ (one dataset per source/server), the key distinction from our work lies in source design: in distributed source coding the sources are fixed, whereas in our setting they are shaped through careful task assignment. It is this key overarching distinction, together with having larger M, that introduces challenging combinatorial optimization elements in designing schemes and converses, which we address in our study.

Another related and well-studied direction is *Coded MapReduce* [13], which focuses on distributed computation of separable functions under a framework where servers compute intermediate function values and exchange these, via coded messages, to obtain the inputs needed for their final assigned tasks. The challenge here is to jointly design the task assignment and internode communication strategy in order to best mitigate the communication bottleneck. Variants have been proposed to handle stragglers [44], and to account for predefined task assignments [18], heterogeneous channels [17], [45], and different network connectivities [16], often borrowing techniques from device-to-device coded caching. Related problems of function retrieval with communication minimization have also been studied in cache-aided broadcast systems [46] and broadcast networks with side information [47], [48]. While the Coded MapReduce setting carries certain similarities to our setting, such as for example the need to carefully assign datasets and tasks across the servers, the unifying distinction between these works and our line of work – beyond a deep dependence on subpacketization, the fundamentally different topology and the presence of data-exchange among the servers – is that they critically rely on side information and focus on its use in alleviating the rank-one communication bottleneck.

Further connections to our setting can be found in the rich line of work that studies the application of distributed computing in certain high-impact tasks. Associated to our specific case $L = 1$ (a single requested function), this includes *gradient coding* [20], [22], [49], which develops techniques to mitigate stragglers in distributed learning. Here, the master computes a single gradient with the help of N servers (N depends on K, M, and the number of tolerated stragglers), each evaluating partial gradients. Most schemes use cyclic or fractional repetition task assignment strategies [20], [22], [49], with additional variants to be found in [4], [21], [50], [51]. Another related direction – motivated in part by the growing computational and communication demands of machine learning – concerns *distributed matrix–vector and matrix–matrix multiplication*, with the majority of works emphasizing on straggler-prone settings [5]–[7], [23], [24]. In these works, the input matrices are encoded, and subsets of the encoded submatrices are assigned to servers for computation. Most existing schemes adopt either erasure-coding techniques or polynomial-based methods for encoding. Related studies [52], [53] focus on computing linear transforms of high-dimensional vectors under both straggler effects and explicit computational load constraints (see also the survey [8]). Several extensions have incorporated additional privacy and security guarantees as well [9], [11], [54]–[60]. A key distinction from our work is the highly specific nature of the target function and the primary focus on mitigating stragglers, which typically directs the problem toward coded task assignments inspired by error-correcting codes.

B. Contributions

In this work, we study distributed linearly separable function computation with the goal of minimizing the total communication cost R from servers to the user. In particular, the user requests L linearly separable functions of the K basis subfunctions $f_j(W_j) : j \in [K]$, captured by a demand matrix $\mathbf{D} \in \mathbb{F}_q^{L \times K}$, where $\mathcal{W} = \{W_1, \dots, W_K\}$ is the dataset library at the master node. The challenge arises from the limited computating capability of each server, which can compute at most $M \leq K$ subfunctions. We seek the fundamental limits of the communication–computation (R–M) tradeoff for arbitrary K and L, under the assumption that servers transmit only linear combinations of their computed subfunctions, without subpacketization. The problem reduces to jointly designing (i) the dataset/task assignment under a strict per-server budget M, and (ii) the transmissions that allow the user to recover all requested functions without error. As we clarify later on, performance will be measured by the worst-case communication cost over all possible demands.

The technical contributions of this work are summarized in the following.

- In terms of achievability, we first introduce a novel technique that provides a sufficient condition for designing both task assignments and server transmissions. The core idea (Lemma 1) leverages designed left nullspaces of carefully selected submatrices of the demand matrix to guarantee exact decodability when $K \leq L + M - 1$. In this regime, we construct a 'nullspace-based matrix' (cf. (7)) for any task assignment to test feasibility, and, if feasible, the same matrix specifies the transmissions. Moreover, for all K, L, M satisfying $K \leq L+M-1$, we explicitly provide feasible assignments and transmissions achieving the globally optimal communication cost. While prior works [25], [29] employed nullspace ideas to design transmissions, our contribution extends this principle to task assignment itself. The defining distinction of our approach is that the nullspace design dictates the task allocation, in contrast to [25], [29], where cyclic assignment predetermined the nullspace structure. This extension not only allows for reaching the global optimal over all task assignments, but also ensures exact decodability of all demands, going beyond the probabilistic guarantees in [25], [29][2].

  A few additional design elements appear in the subsequent Schemes 1 and 2, designed for the case of $K > L + M - 1$.

  - Scheme 1 (Theorem 1) applies a column-wise partition of the demand matrix into submatrices, on each of which the nullspace-based design for task assignment and transmissions is applied independently. We further identify the optimal partition that minimizes communication cost. The scheme works for any (K, L, M), and the corresponding cost takes the form $R = \min(K, L\lceil K/(L + M - 1)\rceil)$.
  - Scheme 2 (Theorem 2) is distinguished by admitting a task assignment independent of the demand matrix. It applies in the regime of $M \geq K/2$, where performance is paradoxically improved by augmenting the demand matrix with carefully designed virtual demands,

[2] Indeed, for small field sizes q, those schemes may fail on a significant fraction of demand matrices (see Table I in [25]).

enabling a more efficient fusion of the partitioning and nullspace approaches. To further broaden its applicability, the scheme also incorporates a row-wise partitioning technique.

- A key contribution of this work is the development of new converses for distributed linearly separable function computation. We first establish an information-theoretic converse via a novel degrees-of-freedom argument on sparse matrix factorizations, showing Scheme 1 to be within a small constant factor from the optimal. Furthermore, we present an additional converse that, for the first time, identifies the general covering number from combinatorial design theory as a fundamental limiting factor for communication cost in distributed computing. Furthermore, our work provides a new class of covering designs, that yield even tighter converses. All together, these new connections yield converses that are proven tight, matching the performance of our designed achievability schemes, thus revealing a surprising and deep structural equivalence.
  - To achieve the above, we first reformulate our problem as a sparse matrix factorization problem $\mathbf{D}_{L\times K} = \mathbf{C}_{L\times R}\mathbf{A}_{R\times K}$ where the challenge is to minimize the inner dimension R under the constraint that each row of $\mathbf{A}$ has at most M non-zero elements. This formulation yields a novel information-theoretic lower bound (Theorem 3) on the communication cost of the (K, L, M) problem, derived under the no-subpacketization assumption. The converse stems from the idea that, in the factorization $\mathbf{D} = \mathbf{C}\mathbf{A}$, the total degrees of freedom available in choosing $\mathbf{A}$ must exceed those in $\mathbf{C}$ for a given demand matrix $\mathbf{D}$. These degrees of freedom are then bounded using combinatorial counting arguments and matrix entropic inequalities to obtain the lower bound. This derived lower bound is subsequently used to show that under basic divisibility conditions, the communication cost of Scheme 1 is within a factor of 2 from the optimal when the underlying field size $q \geq eK/M$, while when the divisibility condition is not met, the optimality gap is at most 3 (cf. Theorem 4).
  - In Theorem 5, we derive another lower bound for the (K, L, M) distributed linearly separable function computation problem. This bound establishes a novel connection between the distributed computing setting and combinatorial design theory. Specifically, for given K, L, and $q > K$, we show that the existence of a novel combinatorial structure, here called a *multi-level covering design*[3] with appropriate parameters, is necessary for

[3] This new design will be a restricted version of the classical covering design.

the existence of a sufficiently sparse matrix factorization for some $\mathbf{D} \in \mathbf{F}_q^{L \times K}$, and thus is necessary for any given rate R to be achievable. This relation allows the communication cost of the distributed computing problem to be lower bounded by the multi-level covering number of the associated design, as well as by the general covering number of classical general covering designs. Although determining the multi-level and general covering numbers is difficult, we derive a lower bound for the case $L = 2$ using non-trivial counting arguments, which, interestingly, coincide with the communication cost of Scheme 1 when $(M + 1)|K$.

- Of importance is also Corollary 1 which clarifies that the achievable schemes described in Theorem 1 and Theorem 2, as well as the converse in Theorem 5, remain valid even when the computation is performed over any field, including the field of real numbers.
- Finally, in addition to the main results, we also study the increase in communication rate when, for any fixed M, we reduce the number of computing servers. Our tradeoff analysis, reflected in Theorem 7, shows that under our assumptions, the communication rate penalty remains small across a wide range of parameters, while the reduction in server resources is significant. Viewed in reverse, this means that nearly the same cumulative communication cost can be sustained even with substantially reduced overall computing power, simply by using fewer servers.

C. Organization

Section II formally defines the distributed computing problem of linearly separable functions. Section III describes the achievable schemes proposed for different settings, and Section IV presents the converse bounds for the communication cost achieved by a distributed linearly separable function computing scheme. Before we conclude the paper, Section V explores the tradeoff between the communication cost and the number of servers used in the system. To help the reader follow the exposition, we intersperse various examples throughout the paper.

D. Notations and Definitions

For a positive integer n, we let $[n] = \{1, 2, \ldots, n\}$. For m, $n \in \mathbb{Z}^+$ such that $m < n$, $[m : n]$ denotes the set $\{m, m + 1, \ldots, n\}$, and $m \mid n$ denotes m divides n. All vectors are assumed to be column vectors. For a vector $\mathbf{x}$, the number of non-zero entries in $\mathbf{x}$ is denoted by $||\mathbf{x}||_0$. A vector $\mathbf{x}$ is said to be $k-$sparse if $||\mathbf{x}||_0 \leq k$. Binomial coefficients are denoted by $\binom{n}{k}$, where

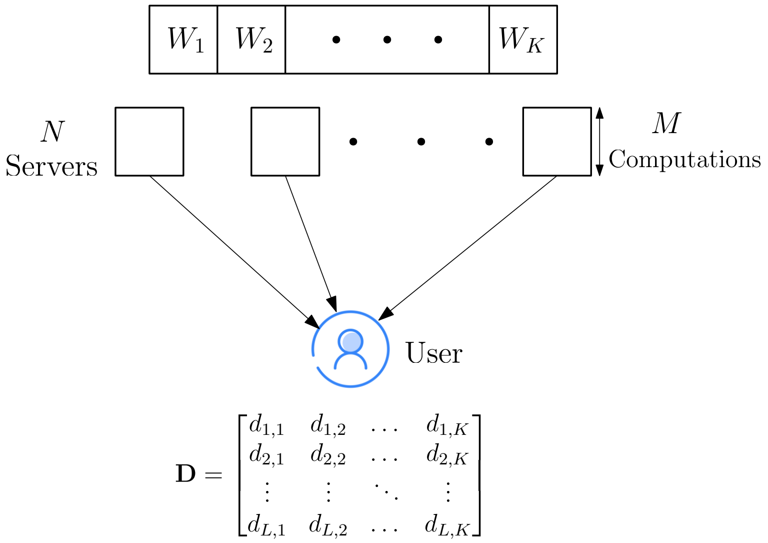

W_1
W_2
W_K
N Servers
M Computations
User
$\mathbf{D} = \begin{bmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,K} \\ d_{2,1} & d_{2,2} & \dots & d_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ d_{L,1} & d_{L,2} & \dots & d_{L,K} \end{bmatrix}$

Fig. 1: System model: (K, L, M) distributed linearly separable function computation.

$\binom{n}{k} = \frac{n!}{k!(n-k)!}$ and $\binom{n}{k} = 0$ for $n < k$. For $N, k \in \mathbb{Z}^+$, $\binom{[N]}{k}$ denotes the set of all $k-$sized subsets of $[N]$. For any set $\mathcal{S}$, the cardinality of $\mathcal{S}$ is denoted by $|\mathcal{S}|$. The complement of set $\mathcal{S}$ is denoted by $\mathcal{S}^c$. The empty set is denoted by $\varnothing$. The vertical concatenation of two matrices $\mathbf{A}_{m_1 \times n}$ and $\mathbf{B}_{m_2 \times n}$ are denoted by $[\mathbf{A}; \mathbf{B}]$. The i-th row and the j-th column of a matrix $\mathbf{A}_{m \times n}$ are denoted by $\mathbf{A}(i, :)$ and $\mathbf{A}(j, :)$, respectively. For a set $\mathcal{S} \subseteq [n]$ and a matrix $\mathbf{A}$ with n columns, $\mathbf{A}_{\mathcal{S}}$ denotes the submatrix of $\mathbf{A}$ formed by the columns indexed by $\mathcal{S}$. For any $x \in \mathbb{R}^+$, $\lceil x \rceil$ denotes the smallest positive integer greater than or equal to x, and $\lfloor x \rfloor$ denotes the largest positive integer less than or equal to x. The finite field of $q \geq 2$ elements is denoted by $\mathbb{F}_q$. A vector of length n with all zeros is denoted by $\mathbf{0}_{n \times 1}$. An $n-$length unit vector with a one at the i-th position and zeroes elsewhere is denoted by $\mathbf{e}_i$. The support of a vector $\mathbf{x} \in \mathbb{F}_q^n$ is defined as $supp(\mathbf{x}) = \{i \in [n] : x_i \neq 0\}$.

II. System Model and Problem Formulation

Consider a distributed computing system with a master node having a library of K i.i.d. datasets $\mathcal{W} = \{W_1, W_2, \dots, W_K\}$, where $W_k \in \mathbb{F}_q^B$, $\forall k \in [K]$.[4] In this setting, a single user

[4] We will for now consider any q. As we will see later on, the achievable schemes are applicable for any field, finite or not. The converse of Theorem 3 is applicable for any finite field $\mathbf{F}_q$, while the converse in Theorem 5 presented later requires $q > K$ and also applies over the field of real and complex numbers. The optimality gap of Theorem 4 (based on Theorem 1 and Theorem 3), holds for all finite fields. The gap reduces to 2 or 3 for any $q \geq eK/M$.

wishes to compute L independent functions $F_1, F_2, \ldots, F_L$ on $\mathcal{W}$, where $F_\ell : (\mathbb{F}_q^B)^K \to \mathbb{F}_q^T$, $\forall \ell \in [L]$ and T denotes the size of each function value. Furthermore, each function F_ℓ is linearly decomposable as

$$F_\ell(\mathcal{W}) = d_{\ell,1} f_1(W_1) + d_{\ell,2} f_2(W_2) + \cdots + d_{\ell,K} f_K(W_K), \quad \forall \ell \in [L]$$

where $d_{\ell,j} \in \mathbb{F}_q$, and where the subfunction $f_j : \mathbb{F}_q^B \to \mathbb{F}_q^T$, $j \in [K]$, can be linear or non-linear and can be computationally intensive. Consequently, the L requested functions can be represented as

$$\begin{bmatrix} F_1(\mathcal{W}) \\ F_2(\mathcal{W}) \\ \vdots \\ F_L(\mathcal{W}) \end{bmatrix} = \underbrace{\begin{bmatrix} d_{1,1} & d_{1,2} & \ldots & d_{1,K} \\ d_{2,1} & d_{2,2} & \ldots & d_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ d_{L,1} & d_{L,2} & \ldots & d_{L,K} \end{bmatrix}}_{\mathbf{D}} \begin{bmatrix} f_1(W_1) \\ f_2(W_2) \\ \vdots \\ f_K(W_K) \end{bmatrix} \tag{1}$$

where – under the worst-case assumption – we consider the case of $\text{rank}_q(\mathbf{D}) = L$. Since the L functions are linearly separable, the computation of the $F_\ell(\mathcal{W})$'s can be performed in a distributed manner across the N servers. We consider the case where the computational capability of the servers is limited, in that each server can compute a maximum of $M \leq K$ subfunctions. This model is illustrated in Figure 1.

The system operates in three phases: the *demand phase*, the *computing phase*, and the *communication phase*.

Demand phase: In the demand phase, the user informs the master node of its L desired functions, which are naturally described by the matrix $\mathbf{D} \in \mathbb{F}_q^{L \times K}$.

Computing phase: Subject to the constraint that each server computes at most M subfunctions, the master uses the demand matrix $\mathbf{D}$ to assign to server n a subset $\mathcal{M}_n \subset [K]$ of subfunctions and the associated datasets. At the end of this phase, server n evaluates $f_j(W_j) \in \mathbb{F}_q^T$ for all $j \in \mathcal{M}_n$. We often refer to the collection $\mathcal{M} = \{\mathcal{M}_n : n \in [N]\}$ as the *task assignment*.

Communication phase: After computing, server n sends r_n linearly encoded messages based on its local outputs $f_j(W_j)$, where each message is of the form[5]

$$x_{n,r} = \sum_{j \in \mathcal{M}_n} \alpha_{n,r,j} f_j(W_j) \tag{2}$$

[5] We adopt the standard no-subpacketization assumption, where servers do not divide subfunction outputs into smaller parts.

for $r \in [r_n]$, with coefficients $\alpha_{n,r,j} \in \mathbb{F}_q$. This means that each server n transmits $\mathbf{x}_n = [x_{n,1}, x_{n,2}, \ldots, x_{n,r_n}]^\intercal$, where

$$\begin{bmatrix} x_{n,1} \\ x_{n,2} \\ \vdots \\ x_{n,r_n} \end{bmatrix} = \underbrace{\begin{bmatrix} \alpha_{n,1,1} & \alpha_{n,1,2} & \ldots & \alpha_{n,1,K} \\ \alpha_{n,2,1} & \alpha_{n,2,2} & \ldots & \alpha_{n,2,K} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{n,r_n,1} & \alpha_{n,r_n,2} & \ldots & \alpha_{n,r_n,K} \end{bmatrix}}_{\mathbf{A}_n \in \mathbb{F}_q^{r_n \times K}} \begin{bmatrix} f_1(W_1) \\ f_2(W_2) \\ \vdots \\ f_K(W_K) \end{bmatrix} . \tag{3}$$

Since $x_{n,r}$ is a linear combination of at most M subfunctions, then $||\mathbf{A}_n(r,:)||_0 \leq M, \forall r \in [r_n]$ and $n \in [N]$. For $\mathbf{x} = [\mathbf{x}_1; \mathbf{x}_2; \ldots; \mathbf{x}_N]$ denoting the entire set of transmissions across all the servers, for $R = \sum_{n \in [N]} r_n$, and for $\mathbf{A} = [\mathbf{A}_1; \mathbf{A}_2; \ldots; \mathbf{A}_N] \in \mathbb{F}_q^{R \times K}$ denoting the *encoding matrix*, then the entire set of transmissions can be represented as

$$\mathbf{x} = \mathbf{A}[f_1(W_1), f_2(W_2), \ldots, f_K(W_K)]^\intercal$$

where each row of $\mathbf{A}$ has a non-empty support of cardinality at most M.

The communication link between each of the N servers and the user is assumed to be error-free and non-interfering. The user is allowed to perform linear operations on the received messages, in order to decode the L function values, and thus this decoding operation can be represented as

$$\begin{bmatrix} F_1(\mathcal{W}) \\ F_2(\mathcal{W}) \\ \vdots \\ F_L(\mathcal{W}) \end{bmatrix} = \underbrace{\begin{bmatrix} c_{1,1} & c_{1,2} & \ldots & c_{1,R} \\ c_{2,1} & c_{2,2} & \ldots & c_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ c_{L,1} & c_{L,2} & \ldots & c_{L,R} \end{bmatrix}}_{\mathbf{C} \in \mathbb{F}_q^{L \times R}} \underbrace{\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}}_{\mathbf{x} \in \mathbb{F}_q^{R \times 1}} = \mathbf{CA} \begin{bmatrix} f_1(W_1) \\ f_2(W_2) \\ \vdots \\ f_K(W_K) \end{bmatrix} \tag{4}$$

where the matrix $\mathbf{C}$ is composed of the combining coefficients. From (1) and (4), and from the fact that the scheme must hold independent of the dataset realizations and subfunction outputs, it is evident that the demand matrix $\mathbf{D} \in \mathbb{F}_q^{L \times K}$ must be decomposed as

$$\mathbf{D} = \mathbf{CA}. \tag{5}$$

Because of this equivalence between the distributed computing problem and matrix decomposition, our aim becomes that of designing two matrices $\mathbf{C} \in \mathbb{F}_q^{L \times R}$ and $\mathbf{A} \in \mathbb{F}_q^{R \times K}$, where $R \geq L$, that factorize $\mathbf{D}$ subject to an M-sparsity constraint on every row of $\mathbf{A}$.

The communication cost here represents the total number of transmissions, from the servers to the user, required for recovery of the L desired function outputs at the user. In particular, for a given $\mathbf{D} \in \mathbb{F}_q^{L\times K}$, the communication cost takes the form

$$R_{\mathbf{D}}(M) = \sum_{n\in[N]} r_n$$

where we assume unit-length file messages, i.e., we assume $T = |x_{n,r}| = 1,\ n \in [N], r \in [r_n]$.[6] The rate of interest in our work will represent the worst-case communication cost

$$R(K, L, M) = \max_{\mathbf{D}\in\mathbb{F}_q^{L\times K}} R_{\mathbf{D}}(M)$$

over all full-rank matrices $\mathbf{D}$, and thus our interest is in identifying the optimal rate

$$R^*(K, L, M) = \inf\{R(K, L, M) : R(K, L, M) \text{ is achievable}\}$$

where the infimum is over all task assignments and linear transmission and decoding policies. Our objective is to design the assignment and communication scheme that approaches this optimum.

III. Achievability

In this section, we present two achievable schemes for the distributed linearly separable function computation problem. The first scheme (Theorem 1) applies to all values of M, K, while the second scheme (Theorem 2) is designed to improve the performance of the first scheme in the region of $M \geq K/2$. Additionally, this second scheme enjoys a demand-agnostic task assignment. We remind the reader that the parameter range of interest is naturally that for which both M and L remain less than K.

A. Scheme 1: A General Achievability Scheme

Theorem 1. *For the (K, L, M) distributed linearly separable function computation problem, the rate*

$$R_1(K, L, M) = \min\left\{K, L\left\lceil\frac{K}{L+M-1}\right\rceil\right\} \tag{6}$$

is achievable.

Proof. The proof of the achievability of the rate expression in (6) is divided into two cases. Case 1 for $K \leq L + M - 1$, and Case 2 for $K > L + M - 1$. The crux of the proof of Case 1 lies in Lemma 1 presented next, while for Case 2, we combine the idea presented in Lemma 1 with a demand matrix partitioning technique.

[6] A subfunction output/coded transmission comprising T symbols is treated as one unit.

*1) **Case 1** $(K \leq L+M-1)$:*

We first consider the case $K \leq L+M-1$. When doing so, it is sufficient to focus on the scenario $K = L+M-1$, because if $K < L+M-1$, each server uses only $M' = K-L+1 < M$ of its computing capability, reducing the problem to the case $K = L+M'-1$. Consider a demand matrix $\mathbf{D} \in \mathbb{F}_q^{L \times K}$, where $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K]$. Recall that $\mathcal{M}_n \subset [K]$ is the set of indices of datasets known to server n, where $|\mathcal{M}_n| = M = K-L+1$ for every $n \in [N]$. We now define a submatrix

$$\mathbf{D}_{\mathcal{M}_n} = [\mathbf{d}_j : j \in \mathcal{M}_n], \quad n \in [N]$$

of the demand matrix $\mathbf{D}$ by choosing the columns indexed with the set $\mathcal{M}_n$. Consider the following matrix

$$\mathbf{N}_{\mathbf{D},\mathcal{M}} = [\mathcal{N}(\mathbf{D}^\intercal_{\mathcal{M}_1^c}), \mathcal{N}(\mathbf{D}^\intercal_{\mathcal{M}_2^c}), \ldots, \mathcal{N}(\mathbf{D}^\intercal_{\mathcal{M}_N^c})]^\intercal \tag{7}$$

where $\mathbf{D}_{\mathcal{M}_n^c}$ denotes the matrix obtained by choosing the columns of $\mathbf{D}$ indexed with the set $[K] \backslash \mathcal{M}_n$ and where $\mathcal{N}(.)$ is the nullspace operator, which outputs a set of basis vectors of the nullspace of the matrix on which the operator acts. For any $\mathcal{M}_n$ with $|\mathcal{M}_n| = M = K-L+1$, the submatrix $\mathbf{D}_{\mathcal{M}_n^c}$ is of size $L \times (L-1)$, and therefore $\mathbf{D}_{\mathcal{M}_n^c}$ has a non-trivial left nullspace. Consequently, the rank of the left nullspace of $\mathbf{D}_{\mathcal{M}_n^c}$ is at least one, and $\mathcal{N}(\mathbf{D}^\intercal_{\mathcal{M}_n^c})$ contains at least one vector for every $n \in [N]$. Thus, the number of rows in the matrix $\mathbf{N}_{\mathbf{D},\mathcal{M}}$ is at least N. We now have the following lemma.

Lemma 1. *For a given $\mathbf{D} \in \mathbb{F}_q^{L \times K}$ with $\text{rank}_q(\mathbf{D}) = L$, and $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_N\}$, the rate L is achievable if*

$$\text{rank}_q(\mathbf{N}_{\mathbf{D},\mathcal{M}}) = L. \tag{8}$$

Furthermore, for any given $K, L,$ and M with $K = L+M-1$, and for any $\mathbf{D} \in \mathbb{F}_q^{L \times K}$, the rate

$$R_1(K, L, M) = L$$

is achievable.

Proof of Lemma 1. Consider a demand matrix $\mathbf{D} \in \mathbb{F}_q^{L \times K}$, where $\text{rank}_q(\mathbf{D}) = L$. Furthermore, assume that the task assignment $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_N\}$ is such that the condition

$$\text{rank}_q(\mathbf{N}_{\mathbf{D},\mathcal{M}}) = L$$

holds. Then, the matrix $\mathbf{N}_{\mathbf{D},\mathcal{M}}$ has at least a set of L independent rows. Let $\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}$ be an $L \times L$ matrix constructed by choosing a set of L independent rows from $\mathbf{N}_{\mathbf{D},\mathcal{M}}$. In the communication phase, there is a transmission corresponding to each row in $\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}$. Let the ℓ-th row of $\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}$, denoted by $\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}(\ell,:)$, corresponds to the task assignment $\mathcal{M}_n \in \mathcal{M}$ for some $n \in [N]$. Then, server n makes the following transmission

$$\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}(\ell,:)\mathbf{D}[f_1(W_1), f_2(W_2), \dots, f_K(W_K)]^\intercal. \tag{9}$$

Now, it remains to show the following two claims.

- *Claim 1:* Server n can construct the transmission in (9) from the available datasets in $\mathcal{M}_n$.
- *Claim 2:* The user can decode the L required functions using the L transmissions received from the servers.

Claim 1 essentially means that the message in (9) is a linear combination of the subfunctions $\{f_j(W_j) : j \in \mathcal{M}_n\}$. In other words, we need to show that

$$supp(\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}(\ell,:)\mathbf{D}) \subseteq \mathcal{M}_n. \tag{10}$$

However notice that

$$\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}(\ell,:) \in \mathcal{N}(\mathbf{D}^\intercal_{\mathcal{M}_n^c})$$

and thus that

$$\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}(\ell,:)\mathbf{D}_{\mathcal{M}_n^c} = \mathbf{0}_{1\times K-M}$$

which implies that (10) is true, and thus Claim 1 follows.

From (9), the L transmitted messages can be written in matrix form as

$$\mathbf{x}_{L\times 1} = \tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}\ \mathbf{D}[f_1(W_1), f_2(W_2), \dots, f_K(W_K)]^\intercal. \tag{11}$$

Since $\text{rank}_q(\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}) = L$, the matrix $\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}$ is invertible, and thus the user can decode the requested functions from $\mathbf{x}$ as

$$\mathbf{D}[f_1(W_1), f_2(W_2), \dots, f_K(W_K)]^\intercal = (\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}})^{-1}\mathbf{x}_{L\times 1}. \tag{12}$$

To show the achievability of $R_1(K, L, M) = L$ for any given $\mathbf{D} \in \mathbb{F}_q^{L\times K}$ with $K = L + M - 1$, we give an explicit construction of $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_L\}$ satisfying the condition $\text{rank}_q(\mathbf{N}_{\mathbf{D},\mathcal{M}}) = L$. Note that here we have $N = L$. Without loss of generality, we assume that $\text{rank}_q(\mathbf{D}) = L$. If this is not the case, we can replace some $L - \text{rank}_q(\mathbf{D})$ linearly dependent rows of $\mathbf{D}$ with a different set of $L - \text{rank}_q(\mathbf{D})$ vectors, so that the resulting matrix has full row rank. If the user can decode the new functions, the removed $L - \text{rank}_q(\mathbf{D})$ functions can still be recovered as linear combinations of the original $\text{rank}_q(\mathbf{D})$ functions that were retained. From the full-row-rank matrix $\mathbf{D}$, we find an invertible submatrix $\mathbf{D}_{\mathcal{L}}$ of size $L \times L$, where $\mathcal{L} = \{i_1, i_2, \dots, i_L\}$ denotes the index set of columns in $\mathbf{D}$ chosen to form $\mathbf{D}_{\mathcal{L}}$. Then, we form the task assignment set

$$\mathcal{M}^\star = \{\mathcal{M}_1^\star, \mathcal{M}_2^\star, \dots, \mathcal{M}_L^\star\} \tag{13}$$

where, for every $\ell \in [L]$

$$\mathcal{M}_\ell^\star = \{i_\ell\} \bigcup \left([K] \backslash \mathcal{L}\right). \tag{14}$$

Note that, $|\mathcal{M}_\ell^\star| = 1 + K - L = M$ for every $\ell \in [L]$. To complete the proof of achievability of the rate $R_1(K, L, M) = L$, it remains to verify that $\text{rank}_q(\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}) = L$. Suppose $\text{rank}_q(\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}) < L$, then there exists a row $\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell', :)$ of $\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}$ which can be represented as

$$\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell', :) = \sum_{\ell=1, \ell \neq \ell'}^{L} \alpha_\ell \mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell, :) \tag{15}$$

where $\alpha_\ell \neq 0$ for some ℓ. On one hand, $\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell', :)$ is a vector in the left nullspace of $\mathbf{D}_{\mathcal{M}_{\ell'}^c} = \mathbf{D}_{\mathcal{L}\backslash\{i_{\ell'}\}}$, and thus

$$\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell', :)\mathbf{D}_{\mathcal{L}} = \beta_{\ell'} \mathbf{e}_{\ell'}^\mathsf{T} \tag{16}$$

where $\beta_{\ell'} \in \mathbb{F}_q \backslash \{0\}$ and $\mathbf{e}_{\ell'} \in \mathbb{F}_q^{L \times 1}$ is an L-length vector with a 1 in the ℓ'-th position and 0-s elsewhere. On the other hand, from (15), we have

$$\begin{aligned} \mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell', :)\mathbf{D}_{\mathcal{L}} &= \left(\sum_{\ell=1, \ell \neq \ell'}^{L} \alpha_\ell \mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell, :)\right) \mathbf{D}_{\mathcal{L}} \\ &= \sum_{\ell=1, \ell \neq \ell'}^{L} \alpha_\ell \Big(\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell, :)\mathbf{D}_{\mathcal{L}}\Big) \\ &= \sum_{\ell=1, \ell \neq \ell'}^{L} \alpha_\ell \left(\beta_\ell \mathbf{e}_\ell^\mathsf{T}\right) \end{aligned} \tag{17}$$

where $\beta_\ell \in \mathbb{F}_q \backslash \{0\}$, for every $\ell \in [L] \backslash \{\ell'\}$. Clearly, (16) and (17) contradict. Therefore, $\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}$ cannot have rank less than L. In other words, $\text{rank}_q(\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}) = L$, and therefore, for any given K, L, and M with $K = L + M - 1$, and for any $\mathbf{D} \in \mathbb{F}_q^{L \times K}$, the rate

$$R_1(K, L, M) = L$$

is achievable. This completes the proof of Lemma 1. ■

We continue with the proof of Theorem 1, by proceeding to Case 2. Before doing so, we provide a simple example of Scheme 1 – as it pertains to Case 1 – and demonstrate its achievable rate.

Example 1 (Scheme 1, Case 1)**.** *Consider the* $(K = 4, L = 3, M = 2)$ *distributed linearly separable function computation problem, where the demand matrix is*

$$\mathbf{D} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 2 \\ 1 & 4 & 1 & 2 \end{bmatrix}. \tag{18}$$

We now design the task assignment and the server transmissions based on Lemma 1. The number of servers required to employ the task assignment in Lemma 1 (cf. (8)*) is* $N = L = 3$*. Note that the first three columns of* $\mathbf{D}$ *are linearly independent. Thus, following the nullspace-based approach outlined in* (8)*, we get*

$$\mathcal{M}_1 = \{1, 4\}, \quad \mathcal{M}_2 = \{2, 4\}, \quad \mathcal{M}_3 = \{3, 4\}.$$

Note that the first server does not have access to W_2 *and* W_3*. Now, we find a vector that resides in the left nullspace of the submatrix* $\mathbf{D}_{\mathcal{M}_1^c} = \mathbf{D}_{\{2,3\}}$*. For instance,* $[10, -3, -1]$ *is a vector in the left nullspace of* $\mathbf{D}_{\{2,3\}}$*. Then, the transmission made by the first server is*

$$\begin{aligned} \mathbf{x}_1 &= [10, -3, -1][F_1(\mathcal{W}), F_2(\mathcal{W}), F_3(\mathcal{W})]^\intercal \\ &= 6f_1(W_1) + 2f_4(W_4). \end{aligned} \tag{19}$$

Similarly, the vectors $[-1, 0, 1]$ *and* $[-2, 3, -1]$ *are in the left nullspaces of* $\mathbf{D}_{\mathcal{M}_2^c} = \mathbf{D}_{\{1,3\}}$ *and* $\mathbf{D}_{\mathcal{M}_3^c} = \mathbf{D}_{\{1,2\}}$*, respectively. Therefore, the transmissions from server 2 and server 3 are*

$$\begin{aligned} \mathbf{x}_2 &= [-1, 0, 1][F_1(\mathcal{W}), F_2(\mathcal{W}), F_3(\mathcal{W})]^\intercal \\ &= 3f_2(W_2) + f_4(W_4) \end{aligned} \tag{20}$$

and

$$\begin{aligned} \mathbf{x}_3 &= [-2, 3, -1][F_1(\mathcal{W}), F_2(\mathcal{W}), F_3(\mathcal{W})]^\intercal \\ &= 6f_3(W_3) + 2f_4(W_4) \end{aligned} \tag{21}$$

respectively. Since the matrix constructed using (7)

$$\mathbf{N}_{\mathbf{D},\mathcal{M}} = \begin{bmatrix} 10 & -3 & -1 \\ -1 & 0 & 1 \\ -2 & 3 & -1 \end{bmatrix}$$

is invertible, the user can decode $F_\ell(\mathcal{W})$ for $\ell = 1, 2, 3$, from $\mathbf{x}_1, \mathbf{x}_2$ and $\mathbf{x}_3$. Therefore, from (19)*,* (20)*, and* (21)*, the demanded functions in* (18) *can be decoded as*

$$\begin{aligned} F_1(\mathcal{W}) &= \frac{1}{6}(\mathbf{x}_1 + 2\mathbf{x}_2 + \mathbf{x}_3), \\ F_2(\mathcal{W}) &= \frac{1}{6}(\mathbf{x}_1 + 4\mathbf{x}_2 + 3\mathbf{x}_3), \\ F_3(\mathcal{W}) &= \frac{1}{6}(\mathbf{x}_1 + 8\mathbf{x}_2 + \mathbf{x}_3) \end{aligned}$$

and thus the rate achieved is

$$R_1(K = 4, L = 3, M = 2) = 3.$$

Since rank$_q(\mathbf{D}) = 3$, even a centralized server with $M = 4$, having access to all the datasets, would require 3 transmissions to meet the user's requests. Therefore, the optimal rate of the $(K = 4, L = 3, M = 2)$ distributed linearly separable function computation is

$$R^*(4, 3, 2) = R_1(4, 3, 2) = 3.$$

This completes the example.

We continue with the proof, moving on to Case 2. The nullspace-based condition in Lemma 1 essentially ensures the existence of L linearly independent vectors, each with a Hamming weight (or support size) less than or equal to M, in the rowspace of $\mathbf{D}$. However, now, for $K > L+M-1$, for any $\mathcal{M}_n \subseteq [K]$ with $|\mathcal{M}_n| \leq M$, the number of columns in the submatrix $\mathbf{D}_{\mathcal{M}_n^c}$ is greater than or equal to $K - M$, where $K - M > L - 1$. Subsequently, $\mathbf{D}_{\mathcal{M}_n^c}$ need not have a non-trivial left nullspace in general. To combat this, we devise two techniques. In the next subsection, we present an appropriate column-wise partition of the demand matrix into submatrices (calling them sub-demand matrices), and use the technique presented in the proof of Lemma 1 in each of the sub-demand matrices separately. Later in Theorem 2, we present another technique to build nullspaces by augmenting the rows of the demand matrix.

*2) **Case 2** ($K > L + M - 1$):*

When $K > L + M - 1$, we partition the demand matrix $\mathbf{D} \in \mathbb{F}_q^{L \times K}$ into $\lceil K/K' \rceil$ sub-demand matrices, for some positive integer $K' \leq L + M - 1$, as follows

$$\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_{\lceil K/K' \rceil}]$$

where $\mathbf{D}_\nu \in \mathbb{F}_q^{L \times K'}$ for all $\nu \in [\lceil K/K' \rceil - 1]$, and where $\mathbf{D}_{\lceil K/K' \rceil} \in \mathbb{F}_q^{L \times (K - K'(\lceil K/K' \rceil - 1))}$. Rewriting (1), we have

$$\begin{aligned}
\left[F_1(\mathcal{W}), F_2(\mathcal{W}), \dots, F_L(\mathcal{W})\right]^\mathsf{T} &= \mathbf{D}\left[f_1(W_1), f_2(W_2), \dots, f_K(W_K)\right]^\mathsf{T} && \text{(22a)}\\
&= \sum_{\nu=1}^{\lceil K/K' \rceil - 1} \mathbf{D}_\nu \left[f_{(\nu-1)K'+1}(W_{(\nu-1)K'+1}), \dots, f_{\nu K'}(W_{\nu K'})\right]^\mathsf{T} \\
&\quad + \mathbf{D}_{\lceil K/K' \rceil} \left[f_{(\lceil K/K' \rceil - 1)K'+1}(W_{(\lceil K/K' \rceil - 1)K'+1}), \dots, f_K(W_K)\right]^\mathsf{T} && \text{(22b)}\\
&= \sum_{\nu=1}^{\lceil K/K' \rceil - 1} \left[F_1^{(\nu)}(W_{[(\nu-1)K'+1:\nu K']}), \dots, F_L^{(\nu)}(W_{[(\nu-1)K'+1:\nu K']})\right]^\mathsf{T} \\
&\quad + \left[F_1^{(\lceil K/K' \rceil)}(W_{[(\lceil K/K' \rceil - 1)K'+1:K]}), \dots, F_L^{(\lceil K/K' \rceil)}(W_{[(\lceil K/K' \rceil - 1)K'+1:K]})\right]^\mathsf{T}. && \text{(22c)}
\end{aligned}$$

Consequently, for every $\ell \in [L]$, we get

$$F_\ell(\mathcal{W}) = \sum_{\nu=1}^{\lceil K/K' \rceil} F_\ell^{(\nu)}(W_{[(\nu-1)K'+1:\min(\nu K', K)]}) \tag{23}$$

where

$$F_\ell^{(\nu)}(W_{[(\nu-1)K'+1:\min(\nu K', K)]}) = \mathbf{D}_\nu \left[f_{(\nu-1)K'+1}(W_{(\nu-1)K'+1}), \dots, f_{\min(\nu K', K)}(W_{\min(\nu K', K)})\right]^\mathsf{T}. \tag{24}$$

For every $\ell \in [L]$ and $\nu \in [\lceil K/K' \rceil]$, the function $F_\ell^{(\nu)}(.)$ is linearly separable over datasets $W_{[(\nu-1)K'+1:\min(\nu K', K)]} \subseteq \mathcal{W}$. In addition, the supports of these functions (the domain of the functions) for different values of ν are disjoint. Since the sub-demand matrix $\mathbf{D}_\nu$, $\nu \in [\lceil K/K' \rceil]$, satisfies the condition $K' \leq L + M - 1$, the coding scheme described in Lemma 1 applies, and is employed for $\mathbf{D}_\nu$. Thus, from Lemma 1, the user can retrieve $F_\ell^{(\nu)}$, for every $\ell \in [L]$, with a communication cost L using a set of L servers. This coding strategy is applied separately for every $\nu \in [\lceil K/K' \rceil]$ using distinct sets of L servers. Consequently, using (23), the user can retrieve the demanded functions $\{F_\ell(\mathcal{W}) : \ell \in [L]\}$. Therefore, the rate

$$R_1(K, L, M) = L\left\lceil \frac{K}{K'} \right\rceil \tag{25}$$

is achievable with $N = L\left\lceil \frac{K}{K'} \right\rceil$ servers. To minimize the rate expression in (25), we can choose the largest possible value of K', which is $L + M - 1$. Therefore, we get the required rate expression

$$R_1(K, L, M) = L\left\lceil \frac{K}{L + M - 1} \right\rceil. \tag{26}$$

This completes the proof of Theorem 1. ■

Remark 1. *To make the connection with the subsequent Section V, it is worth quickly observing here that by choosing $K' = L + M - \gamma$, $\gamma \in [2 : L]$ and by designing an alternative task assignment technique discussed in Section V, it will be possible to achieve the rate in* (25) *using $N = \left\lceil \frac{L}{\gamma} \right\rceil \left\lceil \frac{K}{K'} \right\rceil$ servers. Such a chosen value of $K' < L+M-1$ would entail a higher rate, thus bringing to the fore the tradeoff between the rate and the number of servers, which is discussed in detail in Section V.*

We next present an example pertaining to Case 2 ($K > L + M - 1$).

Example 2 (Scheme 1, Case 2)**.** *Consider the $(K = 8, L = 2, M = 3)$ distributed linearly separable function computation problem, and consider the demand matrix*

$$\mathbf{D} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{bmatrix}.$$

Since each server can compute $M = |\mathcal{M}_n| = 3$ subfunctions, the size of submatrix $\mathbf{D}_{\mathcal{M}_n^{\mathrm{c}}}$ is 2×5 for any $\mathcal{M}_n \subseteq [8]$. Therefore, it is not guaranteed to have a left nullspace for $\mathbf{D}_{\mathcal{M}_n^{\mathrm{c}}}$, and subsequently, it is not possible to design a transmission vector for the servers using the technique associated with Lemma 1. To that end, we partition the demand matrix into two submatrices of equal size, $\mathbf{D}_1 = \mathbf{D}_{(:,[1:4])}$ and $\mathbf{D}_2 = \mathbf{D}_{(:,[5:8])}$, that take the form

$$\mathbf{D}_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{bmatrix}, \quad \mathbf{D}_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 4 & 5 & 6 & 7 \end{bmatrix}.$$

Notice that $\mathbf{D}_1$ and $\mathbf{D}_2$ have $K' = K/2 = 4$ columns, and that $K' = L+M-1$. Now, for each submatrix, we can design separate task assignment and transmission strategies using Lemma 1. Therefore, with $R = L = 2$ transmissions (using Lemma 1), the user can obtain the following linear combinations of the subfunctions $f_j(W_j)$, $j \in [4]$

$$F_1^{(1)}(W_{[1:4]}) = f_1(W_1) + f_2(W_2) + f_3(W_3) + f_4(W_4),$$

$$F_2^{(1)}(W_{[1:4]}) = f_2(W_2) + 2f_3(W_3) + 3f_4(W_4).$$

Similarly, with a different set of two servers, the following linear combinations of $f_j(W_j)$, $j \in \{5, 6, 7, 8\}$ can also be obtained by the user

$$F_1^{(2)}(W_{[5:8]}) = f_5(W_5) + f_6(W_6) + f_7(W_7) + f_8(W_8),$$
$$F_2^{(2)}(W_{[5:8]}) = 4f_5(W_5) + 5f_6(W_6) + 6f_7(W_7) + 7f_8(W_8).$$

Recall that the required functions are

$$F_1(\mathcal{W}) = F_1^{(1)}(W_{[1:4]}) + F_1^{(2)}(W_{[5:8]}),$$
$$F_2(\mathcal{W}) = F_2^{(1)}(W_{[1:4]}) + F_2^{(2)}(W_{[5:8]}).$$

Therefore, with $N = 4$ servers, we achieve the rate

$$R_1(K = 8, L = 2, M = 3) = 4.$$

Now, we argue – drawing from our converse in Section IV-B – that $R_1(8, 2, 3) = 4$ is, in fact, the optimal rate for the considered parameter setting. When $K = 8$ and $M = 3$, at least 3 servers are required to store all the datasets. We now show that it is not possible to achieve a rate less than 4 with $N = 3$ servers. Since each dataset has to be assigned to at least one server, out of $K = 8$ datasets, only one will be assigned to two servers, while the remaining seven will each be assigned to a single server. Consequently, each server will be assigned at least two datasets that are not assigned to the remaining two servers. Assume that $\{k_1^{(n)}, k_2^{(n)}\}$ are the indices of the datasets that are assigned exclusively to server n, for every $n \in [3]$. That is, $W_{k_1^{(1)}}$ and $W_{k_2^{(1)}}$ are assigned to the first server, but not to the second and the third server. Since the submatrix $\mathbf{D}_{\{k_1^{(n)}, k_2^{(n)}\}}$ is full rank for every choice of $\{k_1^{(n)}, k_2^{(n)}\}$, server n has to make at least two transmissions, as no other server can make transmissions involving the subfunctions $f_{k_1^{(n)}}(W_{k_1^{(n)}})$ and $f_{k_2^{(n)}}(W_{k_2^{(n)}})$. Then, the achievable rate with $N = 3$ servers must satisfy $R \geq 6$. Therefore, the rate achieved by our scheme $R_1(8, 2, 3) = 4$ with $N = 4$ servers is optimal. Note that, if there are more than 4 servers transmitting, the achievable rate is strictly more than 4, since we do not consider subpacketization of the computed subfunction values.

The converse idea discussed in Example 2 is generalized and connected to a combinatorial structure called the multi-level covering number, which is discussed in detail in Section IV-B.

Before proceeding with Scheme 2, we provide an additional example of Scheme 1 for Case 2. The example highlights the reduction in rate compared to a scheme having a disjoint task assignment policy.

Example 3 (Scheme 1, Case 2)**.** *Consider the* $(K = 6, L = 2, M = 2)$ *distributed linearly separable function computation problem. First, consider the disjoint task assignment strategy* $\mathcal{M}^{(\textit{disjoint})} = \{\mathcal{M}_1^{(\textit{disjoint})}, \mathcal{M}_2^{(\textit{disjoint})}, \mathcal{M}_3^{(\textit{disjoint})}\}$*, where*

$$\mathcal{M}_1^{(\textit{disjoint})} = \{1, 2\}, \quad \mathcal{M}_2^{(\textit{disjoint})} = \{3, 4\}, \quad \mathcal{M}_3^{(\textit{disjoint})} = \{5, 6\}.$$

Let the demand matrix be

$$\mathbf{D} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 \end{bmatrix} \in \mathbb{F}_7^{2\times 6}.$$

In the case of disjoint task assignment, where each server makes two transmissions, the transmissions made by the respective servers are

$$\mathbf{x}_{1,1} = f_1(W_1) + f_2(W_2), \quad \mathbf{x}_{1,2} = f_2(W_2),$$
$$\mathbf{x}_{2,1} = f_3(W_3) + f_4(W_4), \quad \mathbf{x}_{2,2} = 2f_3(W_3) + 3f_4(W_4),$$
$$\mathbf{x}_{3,1} = f_5(W_5) + f_6(W_6), \quad \mathbf{x}_{3,2} = 4f_5(W_5) + 5f_6(W_6).$$

Thus, the rate achieved is

$$R^*_{\textit{disjoint}}(6, 2, 2) = 6$$

with $N = 3$ *servers.*

Corresponding to the scheme in Theorem 1, we have the task assignment $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4\}$*, where*

$$\mathcal{M}_1 = \{1, 3\}, \quad \mathcal{M}_2 = \{2, 3\},$$
$$\mathcal{M}_3 = \{4, 6\}, \quad \mathcal{M}_4 = \{5, 6\}.$$

Thus, from Theorem 1, we have the server transmissions

$$\mathbf{x}_1 = f_1(W_1) - f_3(W_3),$$
$$\mathbf{x}_2 = f_2(W_3) + 2f_3(W_3),$$
$$\mathbf{x}_3 = f_4(W_4) - f_6(W_6),$$
$$\mathbf{x}_4 = f_5(W_5) + 2f_6(W_6).$$

Thus, the rate achieved by our proposed scheme is

$$R_1(6,2,2) = 4.$$

We note that until now, in Scheme 1, after establishing the task assignment, the delivery follows closely in the footsteps of [29] in designing the nullspaces that govern transmission. The nullspaces are naturally different from [29] which relies on fixed task assignment (disjoint or cyclic), but the design method – after establishing our task assignment – is similar. In our setting though, this nullspace approach guarantees decodability for all demand matrices.

We now proceed with Scheme 2 which includes an additional design that carefully augments the demand matrix to build non-trivial nullspaces for the sub-demand matrices.

B. Scheme 2: Alternate Scheme Better Suited for Larger M

In Section III-A2, we devised a matrix partitioning technique to build the nullspace condition in (8) for a given demand matrix. Instead, in this section, we propose another technique for building a target nullspace and satisfying the nullspace condition in Lemma 1, where this now involves augmenting the demand matrix. The proposed technique is applicable for large values of M, specifically when $M \geq K/2$. Since we already have an optimal scheme for the case $L > K - M$ (Case 1 in Scheme 1), we focus on the scenario where $L \leq K - M$, and we have the following theorem.

Theorem 2. *For the (K, L, M) distributed linearly separable function computation problem with $L < \lfloor K/(K-M) \rfloor$, the rate*

$$R_2(K, L, M) = L + 1 \tag{27}$$

is achievable with $N = L + 1$ servers. Furthermore, for any $L \leq K - M$, the rate

$$R_2(K, L, M) = L + \left\lceil \frac{L}{\left\lfloor \frac{M}{K-M} \right\rfloor} \right\rceil \tag{28}$$

is achievable with $N = \lfloor K/(K-M) \rfloor$, if $M \geq K/2$.

Proof. Let $\mathbf{D} \in \mathbb{F}_q^{L \times K}$ be the demand matrix. First, we consider the case $L < \tau$, where $\tau = \lfloor K/(K-M) \rfloor = 1 + \lfloor M/(K-M) \rfloor$ is a positive integer. We set the number of servers $N = L + 1 \leq \tau$. To design the task assignment $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_N\}$, we first partition the set $[K]$ as

$$\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_\tau\}$$

where $|\mathcal{P}_t| = K - M$, for every $t \in [\tau - 1]$, and $|\mathcal{P}_\tau| = K - (\tau - 1)(K - M)$. Note that

$$\begin{aligned}|\mathcal{P}_\tau| &= K - (\tau - 1)(K - M)\\ &= K - \lfloor M/(K - M)\rfloor(K - M)\\ &\geq K - M.\end{aligned}$$

Without loss of generality, we assume that for every $t \in [\tau - 1]$ then

$$\mathcal{P}_t = \{(t - 1)(K - M) + 1, (t - 1)(K - M) + 2, \ldots, t(K - M)\} \tag{29}$$

and that

$$\mathcal{P}_\tau = \{(\tau - 1)(K - M) + 1, (\tau - 1)(K - M) + 2, \ldots, K\} \tag{30}$$

which means that the task assignment set $\mathcal{M}$ takes the form

$$\mathcal{M} = \{\mathcal{P}_1^c, \mathcal{P}_2^c, \ldots, \mathcal{P}_N^c\}. \tag{31}$$

Note that $|\mathcal{M}_n| \leq M$, for every $n \in [N]$. Now, our objective is to design the server transmissions using the nullspace approach presented in Lemma 1. However, the left nullspace of $\mathbf{D}_{\mathcal{M}_n^c}$, for any $n \in [N]$, does not need to have a non-trivial vector, since $L \leq K - M$. Considering that, we augment the demand matrix by an additional row such that the augmented demand matrix, denoted by $\tilde{\mathbf{D}}$, satisfies the required condition in (8). Then

$$\tilde{\mathbf{D}} = \begin{bmatrix}\mathbf{D}\\ \tilde{\mathbf{d}}\end{bmatrix} \in \mathbb{F}_q^{(L+1)\times K}$$

where the construction of $\tilde{\mathbf{d}} \in \mathbb{F}_q^{1\times K}$ is explained in the sequel. First, we define a matrix

$$\mathbf{T} = \begin{bmatrix}1 & 0 & \ldots & 0 & 1\\ 0 & 1 & \ldots & 0 & 1\\ \vdots & \vdots & \ddots & \vdots & 1\\ 0 & 0 & \ldots & 1 & 1\\ 0 & 0 & \ldots & 0 & 1\end{bmatrix} \in \mathbb{F}_q^{N\times N} \tag{32}$$

which we refer to as the 'target nullspace matrix'. Note that $\text{rank}_q(\mathbf{T}) = N$ irrespective of the field size q. We now construct the vector $\tilde{\mathbf{d}}$ as a concatenation of several vectors, where

$$\tilde{\mathbf{d}} = [\tilde{\mathbf{d}}_1, \tilde{\mathbf{d}}_2, \ldots, \tilde{\mathbf{d}}_N]$$

if $\tau = \frac{K}{K-M}$ (when $(K-M)$ divides K) and $L = \tau - 1$. In that case, $\tilde{\mathbf{d}}_n \in \mathbb{F}_q^{1\times(K-M)}$, for every $n \in [N]$. If either $\tau < \frac{K}{K-M}$ or $L < \tau - 1$, we have

$$\tilde{\mathbf{d}} = [\tilde{\mathbf{d}}_1, \tilde{\mathbf{d}}_2, \ldots, \tilde{\mathbf{d}}_N, \tilde{\mathbf{d}}_{N+1}]$$

where $\tilde{\mathbf{d}}_n \in \mathbb{F}_q^{1\times(K-M)}$, for every $n \in [N]$, and $\tilde{\mathbf{d}}_{N+1} \in \mathbb{F}_q^{1\times K-N(K-M)}$. For every $n \in [N-1]$, we now define

$$\tilde{\mathbf{d}}_n = -\mathbf{D}_{\mathcal{P}_n}(n, :) \tag{33}$$

where $\mathbf{D}_{\mathcal{P}_n}(n, :)$ is the n-th row of the submatrix $\mathbf{D}_{\mathcal{P}_n}$. Further, we define

$$\tilde{\mathbf{d}}_N = \mathbf{0}_{1\times(K-M)}. \tag{34}$$

Furthermore, $\tilde{\mathbf{d}}_{N+1}$ can be set to any row vector of length $K - N(K-M)$. Now, using Lemma 1 on the augmented demand matrix $\tilde{\mathbf{D}}$ and the task assignment $\mathcal{M}$ in (31), we show the achievability of the rate $R_2 = N = L + 1$. That is, we show that

$$\text{rank}_q(\mathbf{N}_{\tilde{\mathbf{D}},\mathcal{M}}) = L + 1$$

where

$$\mathbf{N}_{\tilde{\mathbf{D}},\mathcal{M}} = [\mathcal{N}(\tilde{\mathbf{D}}^{\intercal}_{\mathcal{M}_1^c}), \mathcal{N}(\tilde{\mathbf{D}}^{\intercal}_{\mathcal{M}_2^c}), \ldots, \mathcal{N}(\tilde{\mathbf{D}}^{\intercal}_{\mathcal{M}_N^c})]^{\intercal}.$$

Recall that, $\mathcal{M}_n^c = \mathcal{P}_n$, for every $n \in [N]$. Therefore, for every $n \in [N]$, we have

$$\tilde{\mathbf{D}}_{\mathcal{M}_n^c} = \tilde{\mathbf{D}}_{\mathcal{P}_n}.$$

However, in the submatrix $\tilde{\mathbf{D}}_{\mathcal{P}_n}$, we have

$$\tilde{\mathbf{D}}_{\mathcal{P}_n}(n, :) + \tilde{\mathbf{D}}_{\mathcal{P}_n}(N, :) = \tilde{\mathbf{D}}_{\mathcal{P}_n}(n, :) + \tilde{\mathbf{d}}_n = \mathbf{0}_{1\times(K-M)} \tag{35}$$

for every $n \in [N-1]$, where (35) follows from the construction of $\tilde{\mathbf{d}}_n$ in (33). Therefore, for every $n \in [N-1]$, the vector

$$\mathbf{e_n} + \mathbf{e_N} \in \text{span}(\mathcal{N}(\tilde{\mathbf{D}}^{\intercal}_{\mathcal{M}_n^c}))$$

where $\mathbf{e}_i$ is an $N-$length vector with a 1 at the i-th position and 0's elsewhere. Notice that, we also have $\mathbf{e_n} + \mathbf{e_N} = \mathbf{T}(n, :)$, for every $n \in [N-1]$. Therefore, the vector $\mathbf{T}(n, :)$, $n \in [N-1]$, is a row in $\mathbf{N}_{\tilde{\mathbf{D}},\mathcal{M}}$. Furthermore, in the submatrix $\tilde{\mathbf{D}}_{\mathcal{M}_N^c} = \tilde{\mathbf{D}}_{\mathcal{P}_N}$, we have $\tilde{\mathbf{D}}_{\mathcal{P}_N}(N, :) = \tilde{\mathbf{d}}_N = \mathbf{0}$. Thus, the vector

$$\mathbf{e_N} \in \text{span}(\mathcal{N}(\tilde{\mathbf{D}}^{\intercal}_{\mathcal{M}_N^c}))$$

and note that $\mathbf{e_N} = \mathbf{T}(N,:)$. Consequently, the vector $\mathbf{T}(N,:)$ is a row in $\mathbf{N}_{\tilde{\mathbf{D}},\mathcal{M}}$. Therefore, we have

$$\text{rank}_q(\mathbf{N}_{\tilde{\mathbf{D}},\mathcal{M}}) \geq \text{rank}_q(\mathbf{T}) = N = L+1.$$

However, the number of columns in $\mathbf{N}_{\tilde{\mathbf{D}},\mathcal{M}}$ is N. Thus, the rank cannot exceed N. Therefore,

$$\text{rank}_q(\mathbf{N}_{\tilde{\mathbf{D}},\mathcal{M}}) = L+1. \tag{36}$$

The achievability of $R_2(K,L,M) = L+1$ follows from (36).

It remains to show the achievability of (28) for a general L, with $N = \lfloor K/(K-M) \rfloor$. Let $\mathbf{D} \in \mathbb{F}_q^{L \times K}$ be the demand matrix. If $L < \lfloor K/(K-M) \rfloor$, the rate expression in (28) reduces to

$$L + \left\lceil \frac{L}{\left\lfloor \frac{M}{K-M} \right\rfloor} \right\rceil \leq L + \left\lceil \frac{\left\lfloor \frac{K}{K-M} \right\rfloor - 1}{\left\lfloor \frac{M}{K-M} \right\rfloor} \right\rceil = L + \left\lceil \frac{\left\lfloor \frac{M}{K-M} \right\rfloor}{\left\lfloor \frac{M}{K-M} \right\rfloor} \right\rceil = L+1.$$

In order to show the achievability of the expression in (28) for $L \geq \lfloor K/(K-M) \rfloor$, we partition the demand matrix row-wise, and apply the technique presented in the first part of this proof on each of the sub-demand matrices. We have

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \\ \vdots \\ \mathbf{D}_{\lceil L/L' \rceil} \end{bmatrix}$$

where $L' = \lfloor K/(K-M) \rfloor - 1 = \lfloor M/(K-M) \rfloor$. For every $\lambda \in [\lceil L/L' \rceil - 1]$, we have the submatrices $\mathbf{D}_\lambda \in \mathbb{F}_q^{L' \times K}$, and $\mathbf{D}_{\lceil L/L' \rceil} \in \mathbb{F}_q^{(L - (\lceil L/L' \rceil - 1)L') \times K}$. Further,

$$\left[F_1(\mathcal{W}), F_2(\mathcal{W}), \ldots, F_L(\mathcal{W})\right]^\intercal = \mathbf{D}\left[f_1(W_1), f_2(W_2), \ldots, f_K(W_K)\right]^\intercal$$

implies that for every $\lambda \in [\lceil L/L' \rceil - 1]$ it holds that

$$\left[F_{(\lambda-1)L'+1}(\mathcal{W}), F_{(\lambda-1)L'+2}(\mathcal{W}), \ldots, F_{\lambda L'}(\mathcal{W})\right]^\intercal = \mathbf{D}_\lambda\left[f_1(W_1), f_2(W_2), \ldots, f_K(W_K)\right]^\intercal \tag{37}$$

and

$$\left[F_{(\lceil L/L' \rceil -1)L'+1}(\mathcal{W}), F_{(\lceil L/L' \rceil -1)L'+2}(\mathcal{W}), \ldots, F_L(\mathcal{W})\right]^\intercal = \mathbf{D}_{\lceil L/L' \rceil}\left[f_1(W_1), f_2(W_2), \ldots, f_K(W_K)\right]^\intercal. \tag{38}$$

Since $L' < \lfloor K/(K-M) \rfloor$, the task assignment and transmissions based on a target nullspace matrix, presented in the first part of this proof, are applicable in each sub-demand matrix $\mathbf{D}_\lambda$, $\lambda \in [\lceil L/L' \rceil]$. Note that the task assignment set $\mathcal{M}$ is independent of $\mathbf{D}_\lambda$, and we have

$$\mathcal{M} = \{\mathcal{P}_1^c, \mathcal{P}_2^c, \ldots, \mathcal{P}_N^c\}$$

where the sets $\mathcal{P}_n, n \in [N]$, are defined in (29) and (30). For each $\mathbf{D}_\lambda$, the transmission of the servers can be designed using the target nullspace matrix. The functions corresponding to each $\mathbf{D}_\lambda$, $\lambda \in [\lceil L/L' \rceil]$ can thus be decoded by the user (from (37) and (38)).

We now calculate the rate achieved by this distributed computing system. Recall that there are $N = L' + 1$ servers in the system. Corresponding to every $\mathbf{D}_\lambda$, $\lambda \in [\lceil L/L' \rceil - 1]$, each server makes a transmission, and corresponding to $\mathbf{D}_{\lceil L/L' \rceil}$, the first $L - (\lceil L/L' \rceil - 1)L' + 1$ servers make one transmission each and the remaining $N - (L - (\lceil L/L' \rceil - 1)L' + 1) = L' \lceil L/L' \rceil - L$ servers do not make any transmission. Thus, the rate is

$$\begin{aligned}
R_2(K, L, M) &= N \left(\left\lceil \frac{L}{L'} \right\rceil - 1 \right) + L - (\lceil L/L' \rceil - 1)L' + 1 \\
R_2(K, L, M) &= (L' + 1) \left(\left\lceil \frac{L}{L'} \right\rceil - 1 \right) + L - (\lceil L/L' \rceil - 1)L' + 1 \\
&= L + \left\lceil \frac{L}{L'} \right\rceil \\
&= L + \left\lceil \frac{L}{\left\lfloor \frac{M}{K-M} \right\rfloor} \right\rceil .
\end{aligned}$$

This completes the proof of Theorem 2. ■

Remark 2. *Unlike Scheme 1, the task assignment $\mathcal{M}$ in Scheme 2 is independent of the demand matrix. However, if $L < \lfloor K/(K - M) \rfloor - 1$, only $L + 1$ servers need to perform the subfunction computations and the corresponding transmissions; not all servers need to participate in the computation phase.*

Remark 3. *The target nullspace matrix $\mathbf{T}$ need not be the one defined in* (32). *It can be any $N \times N$ matrix over $\mathbb{F}_q$ that satisfies the following two properties.*

1) *$\text{rank}_q(\mathbf{T}) = N$.*
2) *The entries of the N-th column of $\mathbf{T}$ are all non-zero.*

The first property ensures the decodability of the demanded functions, and the second property is required to construct the $L + 1$-th row. The second property also ensures the existence of a nullspace for the corresponding submatrix of the augmented demand matrix $\tilde{\mathbf{D}}$. Having zero as the N-th entry of a row implies the existence of a non-trivial nullspace for the corresponding submatrix of the original demand matrix $\mathbf{D}$, which is not guaranteed.

Example 4 (Scheme 2)**.** *Consider the* $(K=6, L=2, M=4)$ *distributed linearly separable function computation problem with a demand matrix*

$$\mathbf{D} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 \end{bmatrix}$$

and note that $L < K/(K-M) = 3$*. From the scheme described in the proof of Theorem 2, we have* $N = L+1 = 3$*. The task assignment on the three servers is*

$$\mathcal{M}_1 = \{1,2,3,4\},$$

$$\mathcal{M}_2 = \{1,2,5,6\},$$

$$\mathcal{M}_3 = \{3,4,5,6\}.$$

We choose the matrix

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

as the target nullspace matrix. Then the augmented demand matrix $\tilde{\mathbf{D}}$ *is constructed in such a way that the vector* $\mathbf{T}(1,:) = [1,0,-1]$ *falls in the left nullspace of the submatrix* $\tilde{\mathbf{D}}_{\{5,6\}}$*. Similarly, the vectors* $[0,1,-1]$ *and* $[0,0,1]$ *should be in the left nullspaces of* $\tilde{\mathbf{D}}_{\{3,4\}}$ *and* $\tilde{\mathbf{D}}_{\{1,2\}}$*, respectively. Therefore, we get*

$$\tilde{\mathbf{D}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 2 & 3 & 1 & 1 \end{bmatrix}.$$

The following expressions denote the linear combinations of the computed subfunctions sent by the three servers, respectively:

$$\mathbf{x}_1 = f_1(W_1) + f_2(W_2) - f_3(W_3) - 2f_4(W_4)$$

$$\mathbf{x}_2 = f_2(W_2) + 3f_5(W_5) + 4f_6(W_6)$$

$$\mathbf{x}_3 = 2f_3(W_3) + 3f_4(W_4) + f_5(W_5) + f_6(W_6).$$

Note that the user can decode the required functions, since

$$F_1(\mathcal{W}) = \mathbf{x}_1 + \mathbf{x}_3, \quad F_2(\mathcal{W}) = \mathbf{x}_2 + \mathbf{x}_3.$$

Remark 4. *In Example 4, instead of adding the particular row that we have added, it is, in fact, possible to add any random third row and achieve the rate $R_1(6,2,4)=3$ using Scheme 1. This holds because when $L=3$ and $M=4$, we have $K=L+M-1=6$, and in this case, Lemma 1 provides the achievability of $R_1(6,2,4)=3$. However, it is not the case in general. For instance, when $K=9$, $M=6$, and $L=2$, the achievable rate corresponding to Scheme 2 is $R_2=3$, while Scheme 1 requires 2 additional random rows to fall into the regime $K=L+M-1$, and the corresponding rate would be $R_1=4$.*

The achievability result in (28) is obtained by partitioning the demand matrix, row-wise, into sub-demand matrices and applying the achievability result in (27) in each sub-demand matrix. These results in Theorem 2 can also be applied on sub-demand matrices obtained by dividing a demand matrix column-wise, as in the proof of Theorem 1. In certain instances, this could result in a lower rate than the scheme in Theorem 1. One such example is presented below.

Example 5 (Scheme 2 with column partitioning)**.** *Let $K=40, M=15$, and $L=3$. For this setting, the scheme in Theorem 1 achieves the rate*

$$R_1(40,3,15) = L\left\lceil \frac{K}{L+M-1} \right\rceil = 9$$

with $N=9$ servers. However, partitioning $\mathbf{D}$ into two equal-sized (3×20) submatrices $\mathbf{D}_1$ and $\mathbf{D}_2$ and applying Scheme 2 separately on these submatrices will result in a rate

$$R_{ach}(40,3,15) = 2(L+1) = 8$$

with $N=8$ servers.

IV. Converses and Order-Optimality

In this section, we first derive an information-theoretic lower bound on the rate of a general (K,L,M) distributed linearly separable function computation problem. The lower bound assumes that the system can utilize any number of servers, and that each server transmits linear combinations of its computed subfunction outputs, without any subpacketization. With the converse in place, we will then show that our proposed scheme in Theorem 1 is within a constant gap (gap is upper bounded by 2 when a divisibility condition is met, and the gap is upper bounded by 3, in general) from the derived information-theoretic lower bound, when the field size $q \geq eK/M$. We will subsequently derive an additional converse using covering designs, and we will show that for a specific case, the new converse bound matches the achievable rate in Theorem 1.

A. Information-Theoretic Converse

Theorem 3. *For the (K, L, M) distributed linearly separable function computing problem, the optimal rate $R^*(K, L, M)$ satisfies*

$$R^*(K, L, M) \geq \max\Big(L, \frac{LK}{L + M + \log_q \binom{K}{M}}\Big). \tag{39}$$

Proof. Since the user wants L independent linear combinations of $f_j(W_j)$'s, we get

$$R^*(K, L, M) \geq L. \tag{40}$$

To show $R^*(K, L, M) \geq \frac{LK}{L+M+\log_q \binom{K}{M}}$, we resort to the equivalent matrix factorization formulation of the (K, L, M) distributed computing problem. Recall that given a matrix $\mathbf{D} \in \mathbb{F}_q^{L\times K}$ of $\text{rank}_q(\mathbf{D}) = L$, we aim to jointly design two matrices $\mathbf{C} \in \mathbb{F}_q^{L\times R}$ and $\mathbf{A} \in \mathbb{F}_q^{R\times K}$ such that $\mathbf{D} = \mathbf{C}\mathbf{A}$, with the following constraints $||\mathbf{A}(r, :)||_0 \leq M, \forall r \in [R]$, and $\text{rank}_q(\mathbf{A}) = R \geq L$. Assume that there exists such a decomposition $\mathbf{D} = \mathbf{C}\mathbf{A}$. Then, the conditional joint entropy of the matrices $\mathbf{A}$ and $\mathbf{C}$ given $\mathbf{D}$ can be expressed as follows

$$H(\mathbf{A}, \mathbf{C}|\mathbf{D}) = H(\mathbf{A}|\mathbf{D}) + H(\mathbf{C}|\mathbf{D}, \mathbf{A}) \tag{41a}$$

$$= H(\mathbf{A}|\mathbf{D}). \tag{41b}$$

Note that all entropies mentioned in this section are computed to the base q. The transition from (41a) to (41b) is because the matrix $\mathbf{C}$ is deterministic when the matrices $\mathbf{D}$ and $\mathbf{A}$ are known, i.e., given two matrices $\mathbf{D} \in \mathbb{F}_q^{L\times K}$ and $\mathbf{A} \in \mathbb{F}_q^{R\times K}$ with $\text{rank}_q(\mathbf{D}) = L$, $\text{rank}_q(\mathbf{A}) = R \geq L$, and $\text{Rowspace}(\mathbf{D}) \subseteq \text{Rowspace}(\mathbf{A})$, then there exists a unique $\mathbf{C}$ such that $\mathbf{D} = \mathbf{C}\mathbf{A}$. Therefore, we get $H(\mathbf{C}|\mathbf{D}, \mathbf{A}) = 0$. Equation (41) can be equivalently written as

$$H(\mathbf{A}, \mathbf{C}|\mathbf{D}) = H(\mathbf{C}|\mathbf{D}) + H(\mathbf{A}|\mathbf{D}, \mathbf{C}). \tag{42}$$

Since $H(\mathbf{A}|\mathbf{D}, \mathbf{C}) \geq 0$, from (41b) and (42), we have

$$H(\mathbf{C}|\mathbf{D}) \leq H(\mathbf{A}|\mathbf{D}). \tag{43}$$

Equivalently,

$$H(\mathbf{C},\mathbf{D}) \leq H(\mathbf{A},\mathbf{D}). \tag{44}$$

We know that $H(\mathbf{A},\mathbf{D}) = H(\mathbf{A}) + H(\mathbf{D}|\mathbf{A})$, therefore, an upper bound for $H(\mathbf{A},\mathbf{D})$ can be easily computed by upper bounding the terms $H(\mathbf{A})$ and $H(\mathbf{D}|\mathbf{A})$.

Let us first compute $H(\mathbf{A})$. Recall that the rows of the matrix $\mathbf{A} \in \mathbb{F}_q^{R\times K}$ are $M-$sparse. Hence, the total number of possible configurations of $\mathbf{A}(r,:)$, $r \in [R]$, is at most $\binom{K}{M}q^M$, where $\binom{K}{M}$ accounts for the choice of positions of the M non-zero entries, and q^M corresponds to the maximum choices for the values in those M positions. Therefore, we get

$$\begin{aligned} H(\mathbf{A}) &\leq \log_q \left(\binom{K}{M}q^M\right)^R \\ &= R\Big(\log_q \binom{K}{M} + M\Big). \end{aligned} \tag{45}$$

Now, consider $H(\mathbf{D}|\mathbf{A})$, where $\mathbf{D} \in \mathbb{F}_q^{L\times K}$ and $\mathbf{A} \in \mathbb{F}_q^{R\times K}$, $R \geq L$. From the relation $\mathbf{D} = \mathbf{CA}$, we know that the rows of $\mathbf{D}$ form an $L-$dimensional vector space in an $R-$dimensional space spanned by the rows of $\mathbf{A}$. Hence, to upper bound $H(\mathbf{D}|\mathbf{A})$, we need to compute the number of ways in which L linearly independent vectors can be chosen from a given $R-$dimensional space, which is precisely $(q^R-1)(q^R-q)\times\cdots\times(q^R-q^{R-L+1})$. Therefore,

$$H(\mathbf{D}|\mathbf{A}) \leq \log_q\Big((q^R-1)(q^R-q)\cdots(q^R-q^{R-L+1})\Big) \tag{46a}$$

$$< \log_q (q^R-1)^L \tag{46b}$$

$$< RL. \tag{46c}$$

Thus, from (45) and (46c), we have

$$H(\mathbf{A},\mathbf{D}) \leq R\Big(\log_q \binom{K}{M} + M + L\Big). \tag{47}$$

Now, consider $H(\mathbf{C},\mathbf{D})$. We know that

$$H(\mathbf{C},\mathbf{D}) \geq \max\big(H(\mathbf{C}), H(\mathbf{D})\big) \tag{48a}$$

$$= \max(LR, LK) \tag{48b}$$

$$= LK \tag{48c}$$

where the transition from (48a) to (48b) is because $H(\mathbf{C}) \leq LR$ and $H(\mathbf{D}) \leq LK$ (the above inequalities are met with equality if $\mathbf{C}$ and $\mathbf{D}$ are uniform i.i.d. over $\mathbb{F}_q$). Since $L \leq R \leq K$, we get (48c). Thus, using (48c) and (47), equation (44) becomes

$$LK \leq R\Big(\log_q \binom{K}{M} + M + L\Big). \tag{49}$$

Upon rearranging (49), we get

$$R \geq \frac{LK}{L + M + \log_q \binom{K}{M}}$$

which implies

$$R^*(K, L, M) \geq \frac{LK}{L + M + \log_q \binom{K}{M}}. \tag{50}$$

Thus, by combining (40) and (50), we get

$$R^*(K, L, M) \geq \max\Big(L, \frac{LK}{L + M + \log_q \binom{K}{M}}\Big).$$

This completes the proof of Theorem 3. ■

We now characterize the gap to optimal of the performance of Scheme 1 using the converse in (39).

Theorem 4. *For the (K, L, M) distributed linearly separable function computation problem, the achievable rate in Theorem 1 satisfies the following relation when $(L + M - 1)|K$.*

$$\frac{R_1(K, L, M)}{R^*(K, L, M)} \leq 1 + \frac{\log_q \binom{K}{M} + 1}{L + M - 1}. \tag{51}$$

Furthermore, if $q \geq \frac{eK}{M}$, (51) *reduces to*

$$1 \leq \frac{R_1(K, L, M)}{R^*(K, L, M)} \leq 2. \tag{52}$$

When the divisibility condition $(L + M - 1)|K$ does not hold, we have

$$\frac{R_1(K, L, M)}{R^*(K, L, M)} \leq 3. \tag{53}$$

Proof. We let $R_0(K, L, M) = \frac{LK}{L+M-1}$. From (50), we have $R^*(K, L, M) \geq \frac{LK}{L+M+\log_q \binom{K}{M}}$. Therefore,

$$\frac{R_0(K, L, M)}{R^*(K, L, M)} \leq \frac{L + M + \log_q \binom{K}{M}}{L + M - 1} = 1 + \frac{\log_q \binom{K}{M} + 1}{L + M - 1}. \tag{54}$$

Since $\binom{K}{M} \leq \left(\frac{eK}{M}\right)^M$ for all $1 \leq M \leq K$, equation (54) can be written as

$$\frac{R_0(K, L, M)}{R^*(K, L, M)} \leq 1 + \frac{M \log_q \left(\frac{eK}{M}\right) + 1}{L + M - 1} \tag{55a}$$

$$\leq 1 + \frac{M + 1}{L + M - 1} \tag{55b}$$

$$\leq 2. \tag{55c}$$

The transition from (55a) to (55b) is because $\log_q\left(\frac{eK}{M}\right) \leq 1$ for $q \geq \frac{eK}{M}$,[7] and (55c) follows from the fact that $\frac{M+1}{L+M-1} \leq 1$, if $L \geq 2$. Note that, from Theorem 1, we have $R_1(K,L,M) = R_0(K,L,M) = \frac{LK}{L+M-1}$ when $(L+M-1) \mid K$. Therefore, from (55) we obtain

$$R_1(K,L,M) \leq 2R^*(K,L,M). \tag{56}$$

In general, we have

$$\begin{aligned} R_1(K,L,M) &\leq L\left(\frac{K}{L+M-1}+1\right) = \frac{LK}{L+M-1}+L \\ &\leq 2R^*(K,L,M)+L. \end{aligned} \tag{57}$$

Equation (57) follows from (54) and (55). Since $R^*(K,L,M) \geq L$, equation (57) becomes

$$R_1(K,L,M) \leq 3R^*(K,L,M).$$

This completes the proof of Theorem 4. ■

We now derive another lower bound on the rate of a general (K,L,M) distributed linearly separable function computation problem. This lower bound establishes a connection between the distributed computing problem and covering designs. Using the derived lower bound, we show exact optimality results for certain parameter settings.

B. Converse Derived from Multi-Level Covering Designs

We proceed to introduce a new combinatorial design that will help us build our converse.

Definition 1 (Multi-Level Covering Designs)**.** *Given positive integers v, k, and m where $v \geq k$ and $v \geq m$, we define a (v,k,m) multi-level covering design to be a pair $(X, \mathcal{B})$ where X is a set of v elements (called points) and $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_\Omega\}$ a multiset of subsets of X of size at most k (called blocks) such that, for all $m' \leq m$, every m'-subset of X intersects at least m' blocks of $\mathcal{B}$. In other words, $(X, \mathcal{B})$ is a multi-level covering design if the condition*

$$|\{\omega \in [\Omega] : \mathcal{T} \cap \mathcal{B}_\omega \neq \varnothing\}| \geq m' \tag{58}$$

holds for every $\mathcal{T} \subseteq X$ such that $|\mathcal{T}| = m' \leq m$. A (v,k,m) multi-level covering design $(X, \mathcal{B})$ is said to be optimal if:

$$|\mathcal{B}| = \min\{|\mathcal{A}| : \textit{ there exists a } (v,k,m) \textit{ multi-level covering design } (X, \mathcal{A})\}. \tag{59}$$

[7] By applying the approximation $\binom{K}{M} \leq K^M$, we obtain the condition $q \geq K$ on the field size q to ensure the gap results in Theorem 4. In the case $M = 2$, this condition extends the applicability of the results compared to the requirement $q \geq eK/M$.

In this case, the cardinality of $\mathcal{B}$ is called the multi-level covering number and is denoted by $C(v, k, m)$.

We now proceed with our new converse.

Theorem 5. *For the (K, L, M) distributed linearly separable function computation problem with $q > K$, we have*

$$R^*(K, L, M) \geq C(K, M, L) \tag{60}$$

where $C(K, M, L)$ is the multi-level covering number with parameters $v = K, k = M$, and $m = L$.

Before proceeding with the proof, a small remark is in order.

Remark 5. *The closest design to our own, is the well-known $t - (v, k, m, \lambda)$ general covering design (cf. [61]), after setting $t = 1$ and $\lambda = m$. The observant reader may notice though that in the classical definition [61], the intersection condition is imposed only on m-subsets of X, with a fixed requirement on the level of intersection. In contrast, our definition requires the condition to hold for all $m' \leq m$, with the intersection threshold changing with m'. Consequently, the multi-level covering number $C(v, k, m)$ in our setting is never smaller than the classical general covering number $C_{\mathrm{GCD}}(v, k, m)$.*[8] *Hence we can conclude that the bound* (60) *remains valid under either definition, since our multi-level covering number is always larger. The new definition can provide for a potentially tighter converse.*

We proceed with the proof.

Proof. Let $\mathbf{D} \in \mathbb{F}_q^{L \times K}$ be the demand matrix. Since we are interested in the worst-case communication cost, we consider $\mathbf{D}$ to be a matrix with the property that every submatrix of $\mathbf{D}$ of size $L \times L$ is invertible. We know from the literature on MDS codes, that the existence of such a matrix for a given K and any $L \leq K$ requires the field size $q > K$ [62], [63]. Let R be the total

[8] Indeed, every (v, k, m) multi-level covering design is also a (v, k, m) general covering design, since the case $m' = m$ is included. The reverse need not hold, hence $C(v, k, m) \geq C_{\mathrm{GCD}}(v, k, m)$.

number of transmissions made by the servers. Note that R need not be the number of servers, since for $r \neq r'$, it is possible to have $\mathcal{M}_r = \mathcal{M}_{r'}$. For every $r \in [R]$, we have the transmission[9]

$$x_r = \sum_{\substack{j \in \mathcal{M}_r \\ |\mathcal{M}_r| \leq M}} \alpha_{r,j} f_j(W_j) \tag{61}$$

for $\alpha_{r,j} \in \mathbb{F}_q$, where $\mathcal{M}_r \subseteq [K]$, $|\mathcal{M}_r| \leq M$ is the support of the transmission vector x_r, $r \in [R]$.[10] The user linearly combines the transmissions x_r, $r \in [R]$, to obtain the demanded functions. Recalling the decomposition of $\mathbf{D}$ into transmission and decoding matrices in (3) and (4), we write

$$\begin{aligned}
\begin{bmatrix} F_1(\mathcal{W}) \\ F_2(\mathcal{W}) \\ \vdots \\ F_L(\mathcal{W}) \end{bmatrix} &= \underbrace{\begin{bmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,K} \\ d_{2,1} & d_{2,2} & \dots & d_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ d_{L,1} & d_{L,2} & \dots & d_{L,K} \end{bmatrix}}_{\mathbf{D}} \begin{bmatrix} f_1(W_1) \\ f_2(W_2) \\ \vdots \\ f_K(W_K) \end{bmatrix} \\
&= \underbrace{\begin{bmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,R} \\ c_{2,1} & c_{2,2} & \dots & c_{2,R} \\ \vdots & \vdots & \ddots & \vdots \\ c_{L,1} & c_{L,2} & \dots & c_{L,R} \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \dots & \alpha_{1,K} \\ \alpha_{2,1} & \alpha_{2,2} & \dots & \alpha_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{R,1} & \alpha_{R,2} & \dots & \alpha_{R,K} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} f_1(W_1) \\ f_2(W_2) \\ \vdots \\ f_K(W_K) \end{bmatrix}
\end{aligned} \tag{62}$$

to get

$$\mathbf{D} = \mathbf{C}\mathbf{A}. \tag{63}$$

Note that the positions of nonzero entries in the r-th row of $\mathbf{A}$ are determined by $\mathcal{M}_r$. Consider a matrix $\mathbf{D}_\mathcal{L}$ formed by choosing the columns of $\mathbf{D}$ indexed with the elements of the set $\mathcal{L} \subseteq [K]$, where $|\mathcal{L}| = L' \leq L$. From the matrix decomposition in (63), we have

$$\mathbf{D}_\mathcal{L} = \mathbf{C}\mathbf{A}_\mathcal{L} \tag{64}$$

where $\mathbf{A}_\mathcal{L}$ is formed by choosing the columns of $\mathbf{A}$ indexed with the elements of the set $\mathcal{L}$. From our earlier assumption on the invertibility of the submatrices of $\mathbf{D}$, we have $\text{rank}_q(\mathbf{D}_\mathcal{L}) = L'$ for any $\mathcal{L} \subseteq [K]$ with $|\mathcal{L}| = L' \leq L$. Therefore, we require $\text{rank}_q(\mathbf{A}_\mathcal{L}) = L'$. Further relaxing

[9] With a slight abuse of notation, we denote the R transmissions received at the user with $x_1, x_2, \dots x_R$ instead of the notation x_{n,r_n} used in (2). This mapping can be captured by defining a bijection from the set of possible (n, r_n) pairs to $[R]$.

[10] Note that the set $\{\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_R\}$ differs slightly from the task assignment set used previously.

the requirement, we need a minimum of L' rows in $\mathbf{A}_{\mathcal{L}}$ with at least a non-zero entry. In other words, we need

$$|\{r : \mathcal{L} \cap \mathcal{M}_r \neq \varnothing\}| \geq L'. \tag{65}$$

This condition must be true for every $\mathcal{L} \subseteq [K]$ with $|\mathcal{L}| \leq L$. Comparing (65) with (58), we get that in order to have a matrix decomposition as in (62), the pair $([K], \mathcal{M})$ requires to be a (K, M, L) multi-level covering design, where $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_R\}$. However, the number of blocks in the design

$$|\mathcal{M}| = R \geq C(K, M, L)$$

where

$$C(K, M, L) = \min\{|\mathcal{M}| : \text{ there exists a } (K, M, L) \text{ multi-level covering design } ([K], \mathcal{M})\}$$

is the multi-level covering number. Therefore, the rate incurred by any scheme is lower bounded by

$$R^*(K, L, M) \geq C(K, M, L)$$

if $q > K$. This completes the proof of Theorem 5. ■

Remark 6. *The proof of Theorem 5 does not strictly require the condition $q > K$. Rather, for given $K, L,$ and q, it requires the existence of at least one matrix $\mathbf{D} \in \mathbb{F}_q^{L \times K}$ such that every $L \times L$ submatrix of $\mathbf{D}$ has full rank. For instance, when $K = L + 1$, such a matrix exists even over $\mathbb{F}_2$. As one may notice, this is the same property attributed to the generator matrix of a maximum distance separable (MDS) code [62]. The assumption $q > K$ is often imposed because it guarantees the existence of simple explicit constructions, such as Vandermonde matrices, which possess the required full-rank property [63], [64].*

Remark 7. *A closer inspection of the proof of Theorem 5 shows that any achievable scheme – i.e., any scheme that can successfully treat all full-rank demand matrices – for the (K, L, M) distributed linearly separable function computation problem (when $q > K$) must rely on an underlying multi-level covering design structure. To be more specific, such a scheme can exist only if a $(K, M, L)-$multi-level covering design exists. If such a design did not exist, no scheme would be able to resolve the problem for any demand matrix $\mathbf{D}$ for which every $L \times L$ submatrix*

is invertible[11]*. To elaborate on this point, we note that the task assignments used in Theorem 1 and Theorem 2 naturally give rise to such designs:*
– The task assignment $\mathcal{M}^\star$ in Theorem 1, given in (13)*, forms the block set of a (K, M, L) multi-level covering design $([K], \mathcal{M}^\star)$, where $|\mathcal{M}^\star| = L\left\lceil \frac{K}{L+M-1} \right\rceil$.*[12] *Note that the number of blocks in this design corresponds to the achievable rate of the scheme.*
– Unlike Theorem 1, the multi-level covering design arising from Theorem 2 may contain repeated blocks. Consider the task assignment $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_N\}$ in Theorem 2, given in (31)*. The multiplicity of each block $\mathcal{M}_n$, for $n \in [N]$, is equal to the number of transmissions made by server n. Thus, the corresponding (K, M, L) multi-level covering design is $([K], \mathcal{M}^\star)$, where $\mathcal{M}^\star$ is the multiset obtained from $\mathcal{M}$ by allowing repetitions, and $|\mathcal{M}^\star| = L + \left\lceil \frac{L}{\lfloor M/(K-M) \rfloor} \right\rceil$.*

It can be readily verified that $C_{\text{GCD}}(K, M = 1, L) = K$ and $C_{\text{GCD}}(K, M, L = 1) = \lceil K/M \rceil$. Note that, for these cases, the multi-level covering number coincides with the general covering number. i.e., $C(K, M = 1, L) = K$ and $C(K, M, L = 1) = \lceil K/M \rceil$. For any other values of L and M, the general covering number is not known. Finding the general covering number is a hard problem, in general. Different variants of the covering design problem are studied in the literature [61], [65]–[67]. For most of those problems, closed-form solutions are not available, and existing solutions primarily rely on probabilistic algorithms and extensive computer searches. However, using combinatorial arguments, we find a lower bound on the multi-level covering number for the specific case $L = 2$, and interestingly, the lower bound matches the rate achieved by our scheme in Theorem 1 when $(M + 1)|K$.

Theorem 6. *For the $(K, L = 2, M)$ distributed linearly separable function computation problem, we have*

$$R^*(K, L = 2, M) = \frac{2K}{M + 1}. \tag{66}$$

if $(M + 1)|K$.

Proof. When $(M + 1)|K$, the achievability of the rate $R_1(K, L = 2, M) = \frac{2K}{M+1}$ follows from Theorem 1. In order to show the converse, we use Theorem 5, and prove that $C(K, M, 2) = \frac{2K}{M+1}$. Let $([K], \mathcal{B})$ be a (K, M, L) multi-level covering design. Let $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_\Omega\}$ where Ω is

[11] It is worth recalling that when considering the real (or the complex) number field, such matrices appear with probability 1.

[12] The column partition can produce submatrices of non-uniform size, which may result in some blocks having fewer than M elements. Nevertheless, condition (58) is satisfied for every subset of $[K]$ of size at most L.

the number of blocks in the design, and for every $\omega \in [\Omega]$, $|\mathcal{B}_\omega| \leq M$. For every $k \in [K]$, we now define

$$\mathcal{B}^{(k)} = \{\omega : \mathcal{B}_\omega \ni k\} \tag{67}$$

where $\mathcal{B}^{(k)}$ is the set of indices of the blocks that contain the point k. Further, for every $\omega \in [\Omega]$, we define

$$\mathcal{K}_\omega = \{k : |\mathcal{B}^{(k)}| = \omega\} \tag{68}$$

where $\mathcal{K}_\omega$ is the set of points that appear exactly in ω blocks. We now denote $\kappa_\omega = |\mathcal{K}_\omega|$. Since each point should be present in at least one block (follows from the definition of multi-level covering design with $m' = 1$), the collection $\{\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_\Omega\}$ is clearly a partition of $[K]$. Therefore, we have

$$\kappa_1 + \kappa_2 + \cdots + \kappa_\Omega = K. \tag{69}$$

Also, we have

$$\kappa_1 + 2\kappa_2 + \cdots + \Omega\kappa_\Omega \leq M\Omega \tag{70}$$

since there are Ω blocks, each of size at most M.

Now, consider two points $k_1, k_2 \in \mathcal{K}_1$. From the definition of multi-level covering design, for the set $\{k_1, k_2\}$, we have

$$|\,\{\omega : \{k_1, k_2\} \cap \mathcal{B}_\omega \neq \varnothing\}\,| \geq 2.$$

Therefore, k_1 and k_2 must be in two distinct blocks. In general, no two points in $\mathcal{K}_1$ can be present in the same block. Thus, we get

$$\Omega \geq \kappa_1. \tag{71}$$

We now find a lower bound on κ_1 to get a lower bound on Ω. From (69) and (70), we get

$$\begin{aligned} M\Omega &\geq \kappa_1 + 2\kappa_2 + \cdots + \Omega\kappa_\Omega \\ &\geq \kappa_1 + 2(\kappa_2 + \cdots + \kappa_\Omega) \\ &= \kappa_1 + 2(K - \kappa_1) \\ &= 2K - \kappa_1. \end{aligned}$$

By rearranging the inequality, we get

$$\kappa_1 \geq 2K - M\Omega. \tag{72}$$

Finally, by combining (71) and (72), we get the required lower bound on Ω as

$$\Omega \geq \frac{2K}{M+1}.$$

Therefore, we can conclude that

$$R^*(K, L=2, M) = C(K, M, 2) = \frac{2K}{M+1}$$

if $(M+1)|K$. ■

We complete this part with a small corollary that extends some of the results to the case of the real and complex fields.

Corollary 1. *The achievability results of Theorem 1 and Theorem 2, as well as the converse in Theorem 5, remain valid even when our setting considers computations and communication over the real and complex fields.*

Proof. In terms of achievable schemes, the proof is direct after observing that the nullspace-based argument in Lemma 1 (see (8)) – which relies on the notions of linear independence of vectors and the invertibility of the matrix $\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}$ – holds for demand matrices defined over any field. Similarly, to conclude that the converse in Theorem 5 extends to the case of the field of real and complex numbers, it suffices to note that a) over the real or complex numbers it holds that for any L and K, there exists a matrix $\mathbf{D} \in \mathbb{R}^{L \times K}$ such that every $L \times L$ submatrix of $\mathbf{D}$ has full rank L, and b) by noting that the proof of Theorem 5 relies only on the matrix factorization in (63), which is valid even over the real and complex fields. ■

V. Tradeoff Between Rate and Number of Servers

The presented results until now considered the task of minimizing the communication cost R without constraints on the number of servers. This current section places constraints on N, and thus explores the natural tradeoff between servers N and rate R.

Theorem 7. *For the (K, L, M) distributed linearly separable function computation problem, for every $\gamma \leq L$, the rate*

$$R^{(\gamma)}(K, L, M) = \min\left\{K, L\left\lceil\frac{K}{L+M-\gamma}\right\rceil\right\} \tag{73}$$

is achievable with

$$N^{(\gamma)} = \left\lceil\frac{L}{\gamma}\right\rceil\left\lceil\frac{K}{L+M-\gamma}\right\rceil \tag{74}$$

servers.

Proof. The achievability of $R^{(\gamma)}(K,L,M)=K$ is trivial, and therefore we consider the scenario $L\left\lceil\frac{K}{L+M-\gamma}\right\rceil<K$. Similar to the proof of Theorem 1, we first consider the case $K=L+M-\gamma$ for some $\gamma\in[L]$, and show the achievability of $R^{(\gamma)}(K,L,M)=L$ for any given $\mathbf{D}\in\mathbb{F}_q^{L\times K}$ using $N=\left\lceil\frac{L}{\gamma}\right\rceil$ servers. Note that the condition $L\left\lceil\frac{K}{L+M-\gamma}\right\rceil<K$ ensures that $M-\gamma>0$, and consequently we have $\gamma<M$ and $K>L$. Furthermore, without loss of generality, we assume that $\text{rank}_q(\mathbf{D})=L$. If this is not the case, we can replace some $L-\text{rank}_q(\mathbf{D})$ linearly dependent rows of $\mathbf{D}$ with a different set of $L-\text{rank}_q(\mathbf{D})$ vectors, so that the resulting matrix has full row rank. If the user can decode the new functions, the removed functions can still be recovered as linear combinations of the original $\text{rank}_q(\mathbf{D})$ functions that were retained. From the full-row-rank matrix $\mathbf{D}$, we find an invertible submatrix $\mathbf{D}_{\mathcal{L}}$ of size $L\times L$, where $\mathcal{L}=\{i_1,i_2,\ldots,i_L\}$ denotes the indices of columns of $\mathbf{D}$ chosen to form $\mathbf{D}_{\mathcal{L}}$. We now give an explicit construction of the task assignment $\mathcal{M}^\star$, where

$$\mathcal{M}^\star=\{\mathcal{M}_1^\star,\mathcal{M}_2^\star,\ldots,\mathcal{M}^\star_{\lceil\frac{L}{\gamma}\rceil}\}.$$

For every $\theta\in\left[\left\lceil\frac{L}{\gamma}\right\rceil-1\right]$, we define

$$\mathcal{M}_\theta^\star=\mathcal{M}^\star_{\mathcal{L},\theta}\bigcup([K]\setminus\mathcal{L}) \tag{75}$$

and

$$\mathcal{M}^\star_{\lceil\frac{L}{\gamma}\rceil}=\mathcal{M}^\star_{\mathcal{L},\lceil\frac{L}{\gamma}\rceil}\bigcup([K]\setminus\mathcal{L}) \tag{76}$$

where

$$\mathcal{M}^\star_{\mathcal{L},\theta}=\left\{i_{(\theta-1)\gamma+1},i_{(\theta-1)\gamma+2},\ldots,i_{\theta\gamma}\right\}$$

and

$$\mathcal{M}^\star_{\mathcal{L},\lceil\frac{L}{\gamma}\rceil}=\left\{i_{(\lceil\frac{L}{\gamma}\rceil-1)\gamma+1},i_{(\lceil\frac{L}{\gamma}\rceil-1)\gamma+2},\ldots,i_L\right\}.$$

Note that, $|\mathcal{M}_\theta^\star|=\gamma+K-L=M$, for every $\theta\in\left[\lceil\frac{L}{\gamma}\rceil-1\right]$, and $|\mathcal{M}^\star_{\lceil\frac{L}{\gamma}\rceil}|=L-(\lceil\frac{L}{\gamma}\rceil-1)\gamma+K-L\le M$. Notice that the collection of sets $\{\mathcal{M}^\star_{\mathcal{L},\theta}:\theta\in\left[\left\lceil\frac{L}{\gamma}\right\rceil\right]\}$ is a partition of $\mathcal{L}$.

To now prove the achievability of $R^{(\gamma)}(K,L,M)=L$, it is sufficient to show that $\text{rank}_q(\mathbf{N}_{\mathbf{D},\mathcal{M}^\star})=L$, where $\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}$ is defined as per (7). Suppose the rank of $\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}$ is less than L, then there exists a row ℓ' (where $i_{\ell'}\in\mathcal{M}^\star_{\mathcal{L},\theta}$ for some $\theta\in\left[\left\lceil\frac{L}{\gamma}\right\rceil\right]$) of $\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}$

$$\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell',:)=\sum_{\ell=1,i_\ell\notin\mathcal{M}^\star_{\mathcal{L},\theta}}^{L}\alpha_\ell\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell,:) \tag{77}$$

where not all α_ℓ-s are zeros. Note that, for every ℓ, ℓ' such that $i_\ell, i_{\ell'} \in \mathcal{M}^\star_{\mathcal{L},\theta}$ for any θ, the rows $\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell,:)$ and $\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell',:)$ are linearly independent by definition. Now, on the one hand, we have

$$\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell',:)\mathbf{D}_{\mathcal{L}} = \sum_{\ell: i_\ell \in \mathcal{M}^\star_{\mathcal{L},\theta}} \beta_\ell \mathbf{e}_\ell^\mathsf{T} \tag{78}$$

where β_ℓ is non-zero for at least one ℓ such that $i_\ell \in \mathcal{M}^\star_{\mathcal{L},\theta}$. It follows from the fact that $\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell',:)$ is a vector in the left nullspace of $\mathbf{D}_{(\mathcal{M}^\star_\theta)^c} = \mathbf{D}_{\mathcal{L}\setminus\mathcal{M}^\star_{\mathcal{L},\theta}}$. On the other hand, from (77), we have

$$\begin{aligned}\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell',:)\mathbf{D}_{\mathcal{L}} &= \left(\sum_{\ell=1, i_\ell \notin \mathcal{M}^\star_{\mathcal{L},\theta}}^{L} \alpha_\ell \mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell,:)\right)\mathbf{D}_{\mathcal{L}} \\ &= \sum_{\ell=1, i_\ell \notin \mathcal{M}^\star_{\mathcal{L},\theta}}^{L} \alpha_\ell \Big(\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell,:)\mathbf{D}_{\mathcal{L}}\Big) \\ &= \sum_{\ell=1, i_\ell \notin \mathcal{M}^\star_{\mathcal{L},\theta}}^{L} \omega_\ell \mathbf{e}_\ell^\mathsf{T} \end{aligned} \tag{79}$$

where $\omega_\ell = \alpha_\ell\beta_\ell$ are all not zeros. Clearly, (78) and (79) contradict. Therefore, $\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}$ cannot have rank less than L. In other words, $\text{rank}_q(\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}) = L$, and therefore, for any given K, L, and M with $K = L + M - \gamma$, and for any $\mathbf{D} \in \mathbb{F}_q^{L\times K}$, the rate

$$R^{(\gamma)}(K, L, M) = L$$

is achievable with $N = \left\lceil \frac{L}{\gamma} \right\rceil$.

For a general K, we partition the demand matrix $\mathbf{D} \in \mathbb{F}_q^{L\times K}$ into $\lceil K/K' \rceil$ sub-demand matrices, where $K' = L + M - \gamma$. That is

$$\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_{\lceil K/K' \rceil}]$$

where $\mathbf{D}_\nu \in \mathbb{F}_q^{L\times K'}$ for all $\nu \in [\lceil K/K' \rceil - 1]$, and $\mathbf{D}_{\lceil K/K' \rceil} \in \mathbb{F}_q^{L\times(K - K'(\lceil K/K' \rceil - 1))}$. Then for every $\ell \in [L]$, as in (22), we get

$$F_\ell(\mathcal{W}) = \sum_{\nu=1}^{\lceil K/K' \rceil} F_\ell^{(\nu)}(.) \tag{80}$$

where $F_\ell^{(\nu)}(.)$ are linearly separable functions of subsets of datasets (see (24)). In addition, the supports of these functions (the domain of the functions) for different values of ν are disjoint. Therefore, using the scheme described for the case $K' = L + M - \gamma$ in the first part of this proof, the user can retrieve $F_\ell^{(\nu)}$, for every $\ell \in [L]$, with a communication cost L and with a set of $\lceil L/\gamma \rceil$ servers. This is true for every $\nu \in [\lceil K/K' \rceil]$. Consequently, using (80), the user can retrieve the demanded functions $\{F_\ell(\mathcal{W}) : \ell \in [L]\}$. Therefore, for every $\gamma \in [L]$, the rate

$$R^{(\gamma)}(K, L, M) = L\left\lceil \frac{K}{K'} \right\rceil = L\left\lceil \frac{K}{L + M - \gamma} \right\rceil \tag{81}$$

is achievable with $N^{(\gamma)} = \left\lceil \frac{L}{\gamma} \right\rceil \left\lceil \frac{K}{K'} \right\rceil$ servers. This completes the proof of Theorem 7. ∎

Remark 8. *If $\gamma = L$, the task assignment corresponds to a configuration in which no subfunction is computed on more than one server. This assignment strategy can be applied independently of the demand matrix and is often referred to as 'disjoint placement' [22]. From* (73)*, the rate corresponding to the disjoint task assignment is*

$$R_{\textit{disjoint}}(K, L, M) = R^{(\gamma=L)}(K, L, M) = \min\left\{K, L\left\lceil \frac{K}{M} \right\rceil\right\}.$$

The work [31], which considers a multi-user setting with server-to-user connectivity where each user makes a single request, adopts a 'disjoint support' assumption. This assumption differs from the disjoint task assignment. However, if the number of users is set equal to L and full connectivity is allowed, the model in [31] reduces to the setting considered in this work. Under these conditions, the 'disjoint support' assumption in [31] coincides with the disjoint task assignment. It is also worth noting that

$$\begin{aligned}
\frac{R_{\textit{disjoint}}(K, L, M)}{R_1(K, L, M)} &= \frac{R^{(\gamma=L)}}{R^{(\gamma=1)}} \leq \frac{\min\left\{K, L\left\lceil \frac{K}{M} \right\rceil\right\}}{\min\left\{K, L\left\lceil \frac{K}{L+M-1} \right\rceil\right\}} \\
&\leq \frac{\min\left\{K, L\left\lceil \frac{K}{M} \right\rceil\right\}}{L\left\lceil \frac{K}{L+M-1} \right\rceil} \leq \begin{cases} \frac{L\left\lceil \frac{K}{M} \right\rceil}{\frac{LK}{L+M-1}} & \textit{if } L < M \\ \frac{K}{\frac{LK}{L+M-1}} & \textit{if } L \geq M \end{cases} \\
&\leq \begin{cases} \frac{\frac{K}{M}+1}{\frac{K}{L+M-1}} & \textit{if } L < M \\ \frac{L+M-1}{L} & \textit{if } L \geq M \end{cases} \\
&\leq \begin{cases} \frac{L+M-1}{M} + \frac{L+M-1}{K} & \textit{if } L < M \\ \frac{L+M-1}{L} & \textit{if } L \geq M \end{cases} \\
&\leq \begin{cases} 2 + 1 & \textit{if } L < M \\ 2 & \textit{if } L \geq M \end{cases} \\
&\leq 3.
\end{aligned}$$

Directly from the above, we can also conclude that the entire achievable rate region characterized in Theorem 7 lies within a constant factor of 9 of our novel lower bound presented in Theorem 3.

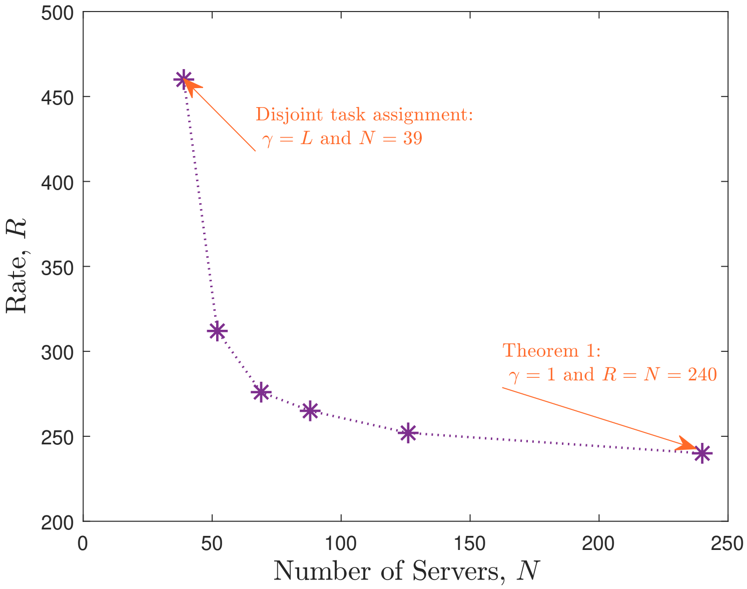


Fig. 2: Rate, R vs. number of servers, N for $K = 460, L = 12, M = 12$.

Remark 9. *Let $\gamma_1 < \gamma_2 \leq L$ such that the conditions $\gamma_1 | L$ and $(L + M - \gamma_1)|K$, and $\gamma_2|L$ and $(L + M - \gamma_2)|K$ hold. Then, from Theorem 7, we get*

$$\frac{R^{(\gamma_2)}}{R^{(\gamma_1)}} = \frac{L + M - \gamma_1}{L + M - \gamma_2}$$

and

$$\frac{N^{(\gamma_2)}}{N^{(\gamma_1)}} = \frac{\gamma_1(L + M - \gamma_1)}{\gamma_2(L + M - \gamma_2)}.$$

Note that the penalty in rate is minimal, while we achieve a significant reduction in the required number of servers with no increase in per-server computation cost, especially when γ_1 and γ_2 are small.

The tradeoff between rate and number of servers for the $(K = 460, L = 12, M = 12)$ distributed linearly separable function computation problem is shown in Figure 2. Note that to achieve a rate $R = 240$, the system requires $N = 240$ servers. However, corresponding to $\gamma = 2$, a rate $R = 252$ is achievable with $N = 126$ servers. In other words, we could achieve almost the same rate with nearly half the number of servers compared to the previous case. Similarly, the (N, R) pairs $(88, 264), (69, 276), (52, 312), (39, 460)$ are also achievable using Theorem 7. Note that we have not included (N, R) pairs corresponding to all possible values of γ between 1 and L in Figure 2, since the ceiling functions in the expressions for R and N can make the curve non-decreasing for certain choices of γ.

Suppose that the pairs $(N^{(\gamma_1)}, R^{(\gamma_1)})$ and $(N^{(\gamma_2)}, R^{(\gamma_2)})$ are achievable in the $N - R$ curve. Then the pair $(\delta N^{(\gamma_1)} + (1 - \delta)N^{(\gamma_2)}, \delta R^{(\gamma_1)} + (1 - \delta)R^{(\gamma_2)})$, $\delta \in [0, 1]$, is also achievable when certain divisibility conditions are satisfied. Let γ_1 and γ_2 be such that the conditions $\gamma_1 | L$ and $(L + M - \gamma_1)|K$, and $\gamma_2 | L$ and $(L + M - \gamma_2)|K$ hold. Then, using Theorem 7, the pairs $(N^{(\gamma_1)}, R^{(\gamma_1)})$ and $(N^{(\gamma_2)}, R^{(\gamma_2)})$ are achievable, where $N^{(\gamma_1)} = LK/(\gamma_1(L + M - \gamma_1))$, $R^{(\gamma_1)} = LK/(L+M-\gamma_1)$, $N^{(\gamma_2)} = LK/(\gamma_2(L+M-\gamma_2))$, and $R^{(\gamma_2)} = LK/(L+M-\gamma_2)$. Further, assume that $\delta \in [0, 1]$ is such that the conditions $(L + M - \gamma_1)|\delta K$ and $(L + M - \gamma_2)|(1 - \delta)K$ hold. We now show the achievability of the pair (N, R), where $N = \delta N^{(\gamma_1)} + (1 - \delta)N^{(\gamma_2)}$ and $R = \delta R^{(\gamma_1)} + (1 - \delta)R^{(\gamma_2)}$. Let $\mathbf{D} \in \mathbb{F}_q^{L \times K}$ be the demand matrix. Now, partition $\mathbf{D}$ column-wise into two sub-demand matrices $\mathbf{D}_1 \in \mathbb{F}_q^{L \times \delta K}$ and $\mathbf{D}_2 \in \mathbb{F}_q^{L \times (1-\delta)K}$. Now, design task assignment and server transmissions using Theorem 7 on $\mathbf{D}_1$ and $\mathbf{D}_2$ with $\gamma = \gamma_1$ and $\gamma = \gamma_2$, respectively. Corresponding to $\mathbf{D}_1$, the rate $L(\delta K)/(L+M-\gamma_1)$ is achievable using $L(\delta K)/(\gamma_1(L+M-\gamma_1))$ servers. Similarly, the rate $L((1 - \delta)K)/(L + M - \gamma_2)$ is achievable for the sub-demand matrix $\mathbf{D}_2$ with $L((1 - \delta)K)/(\gamma_2(L + M - \gamma_2))$ servers. The user can decode the demanded functions as they are the sums of the corresponding sub-demand functions obtained from $\mathbf{D}_1$ and $\mathbf{D}_2$. Therefore, the rate

$$\begin{aligned} R &= \frac{L\delta K}{L + M - \gamma_1} + \frac{L(1 - \delta)K}{L + M - \gamma_2} \\ &= \delta R^{(\gamma_1)} + (1 - \delta)R^{(\gamma_2)} \end{aligned}$$

is achievable using

$$\begin{aligned} N &= \frac{L\delta K}{\gamma_1(L + M - \gamma_1)} + \frac{L(1 - \delta)K}{\gamma_2(L + M - \gamma_2)} \\ &= \delta N^{(\gamma_1)} + (1 - \delta)N^{(\gamma_2)} \end{aligned}$$

servers.

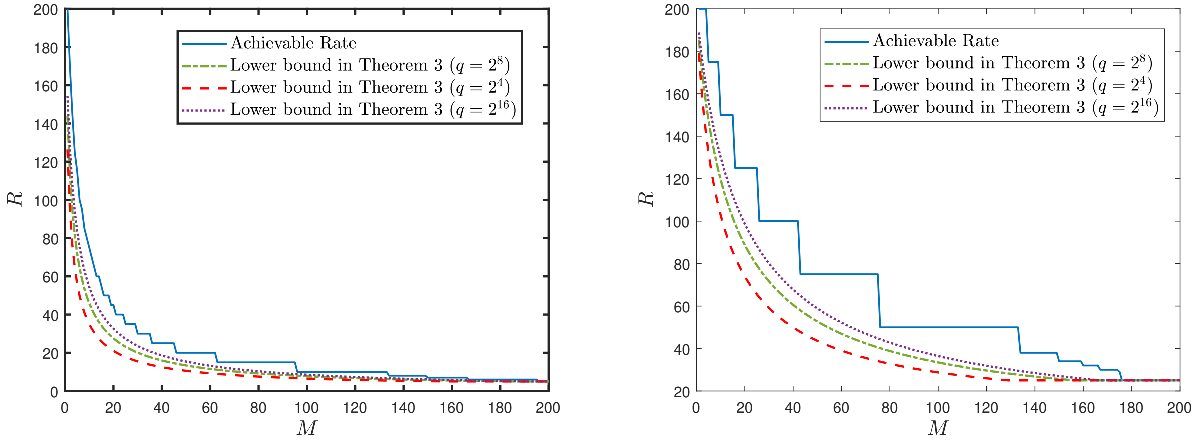
VI. Conclusions

In this work, we studied the problem of distributed computation of linearly separable functions and, under the assumption of linear encoding/decoding without subpacketization, characterized the optimal computation vs. communication-rate tradeoff to within a small constant factor across broad regimes. To establish the achievability results and tight converse bounds, we introduced novel design and analysis tools of broader relevance to coded distributed computing. On the achievability side, we introduced a nullspace-based design principle that simultaneously guides task assignment and server transmissions, ensuring exact decodability and attaining the global optimum when $K \leq L + M - 1$. For $K > L + M - 1$, we proposed two additional schemes; A first scheme based on column-wise partitioning that applies broadly and remains within a constant factor of optimal, and another, effective for $M \geq K/2$, that enjoys task assignments, at the servers, that are independent of the demand matrix. On the converse side, we reformulated the problem as a sparse matrix factorization to derive new information-theoretic lower bounds. These bounds not only certify the near-optimality of our schemes but also reveal a structural connection to combinatorial design theory, with the general covering number emerging as a fundamental limit. We further showed that communication efficiency degrades gracefully under server reduction, allowing comparable performance with fewer active servers.

Figures 3a and 3b illustrate the natural tradeoff between rate and computation cost M in a (K, L, M) distributed linearly separable function computing system. The rate is non-increasing in M, with extremes $R = K$ at $M = 1$ and $R = L$ at $M = K$. The figures compare the achievable rates from Theorems 1 and 2 with the lower bound of Theorem 3 for various field sizes q, where discontinuities from the ceiling function in (6) are most visible when L is close to K and the gap between the achievable rate and the converse diminishes as q grows.

Similarly, Figures 4a and 4b illustrate the tradeoff between rate and the number of requested functions L in (K, L, M) distributed linearly separable function computation for $(K = 200, M = 5)$ and $(K = 200, M = 15)$. For fixed K and M, the rate is generally non-decreasing in L, though the ceiling function in (6) can introduce flat, non-monotonic segments. These flat regions appear in the achievable rate curves, while for large L the behavior becomes linear since Scheme 1 achieves $R_1(K, L, M) = L$ whenever $L \geq K - M + 1$.

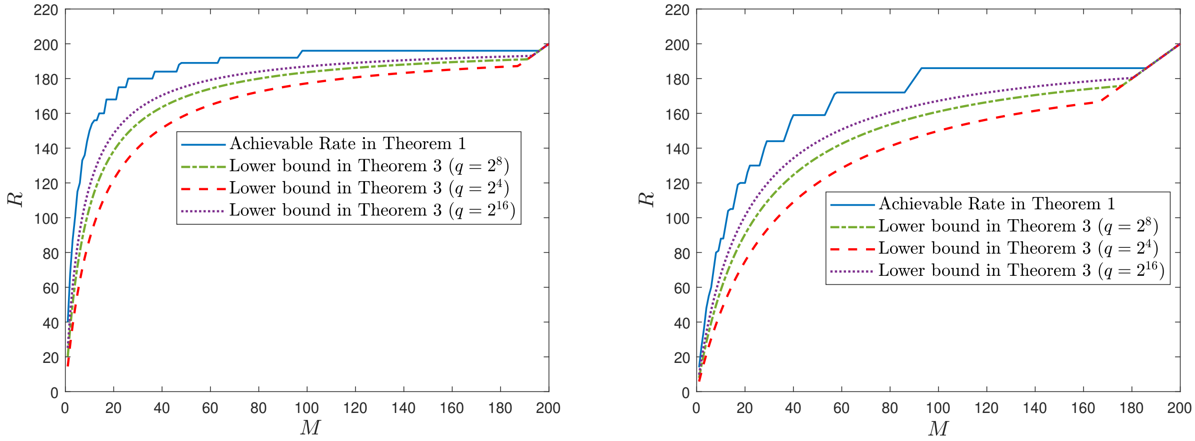
Several research problems remain open. A first direction is to address system-level constraints such as stragglers and heterogeneity across servers, where varying computational or communi-



(a) $(K = 200, L = 5, M)$ distributed linearly separable function computation problem.

(b) $(K = 200, L = 25, M)$ distributed linearly separable function computation problem.

Fig. 3: Rate R vs computation cost M for a fixed number of requests L.



(a) $(K = 200, L, M = 5)$ distributed linearly separable function computation problem.

(b) $(K = 200, L, M = 15)$ distributed linearly separable function computation problem.

Fig. 4: Rate R vs number of requests L for a fixed computation cost M.

cation capabilities may fundamentally reshape the optimal assignments and achievable tradeoffs. Another is to exploit data correlations, which, unlike the independent setting studied here, can be leveraged to reduce communication load and enable more efficient coding strategies. Beyond these extensions, our framework can serve as a foundation for advancing AI-driven applications and modern distributed machine learning tasks.

REFERENCES

[1] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, p. 107–113, Jan. 2008.

[2] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proc. USENIX Conf. Hot Topics in Cloud Comput.*, ser. HotCloud'10, 2010, p. 10.

[3] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "A unified coding framework for distributed computing with straggling servers," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2016, pp. 1–6.

[4] E. Ozfatura, S. Ulukus, and D. Gündüz, "Straggler-aware distributed learning: Communication–computation latency trade-off," *Entropy*, vol. 22, no. 5, 2020. [Online]. Available: https://www.mdpi.com/1099-4300/22/5/544

[5] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1514–1529, 2018.

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

[7] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Straggler mitigation in distributed matrix multiplication: Fundamental limits and optimal coding," *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1920–1933, 2020.

[8] 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 Process. Mag.*, vol. 37, no. 3, pp. 136–145, 2020.

[9] M. Xhemrishi, R. Bitar, and A. Wachter-Zeh, "Distributed matrix-vector multiplication with sparsity and privacy guarantees," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2022, pp. 1028–1033.

[10] K. Wan, H. Sun, M. Ji, and G. Caire, "On secure distributed linearly separable computation," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 3, pp. 912–926, 2022.

[11] Q. Yu and A. S. Avestimehr, "Coded computing for resilient, secure, and privacy-preserving distributed matrix multiplication," *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 59–72, 2021.

[12] X. Yuan and H. Sun, "Vector linear secure aggregation," 2025. [Online]. Available: https://arxiv.org/abs/2502.09817

[13] 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, 2018.

[14] Q. Yan, S. Yang, and M. Wigger, "Storage-computation-communication tradeoff in distributed computing: Fundamental limits and complexity," *IEEE Trans. Inf. Theory*, vol. 68, no. 8, pp. 5496–5512, 2022.

[15] E. Parrinello, E. Lampiris, and P. Elia, "Coded distributed computing with node cooperation substantially increases speedup factors," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2018, pp. 1291–1295.

[16] F. Brunero and P. Elia, "Multi-access distributed computing," *IEEE Trans. Inf. Theory*, vol. 70, no. 5, pp. 3385–3398, 2024.

[17] E. Peter, K. K. K. Namboodiri, and B. S. Rajan, "Wireless MapReduce arrays for coded distributed computing," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2024, pp. 163–168.

[18] Y. Wang and Y. Wu, "Coded distributed computing with pre-set data placement and output functions assignment," *IEEE Trans. Inf. Theory*, vol. 71, no. 3, pp. 2195–2217, 2025.

[19] N. Woolsey, R.-R. Chen, and M. Ji, "A combinatorial design for cascaded coded distributed computing on general networks," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 5686–5700, 2021.

[20] M. Ye and E. Abbe, "Communication-computation efficient gradient coding," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 5610–5619.

[21] A. Reisizadeh, S. Prakash, R. Pedarsani, and A. S. Avestimehr, “Tree gradient coding,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2019, pp. 2808–2812.

[22] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, “Gradient coding: Avoiding stragglers in distributed learning,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 3368–3376.

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

[24] K. Lee, C. Suh, and K. Ramchandran, “High-dimensional coded matrix multiplication,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2017, pp. 2418–2422.

[25] K. Wan, H. Sun, M. Ji, and G. Caire, “On the tradeoff between computation and communication costs for distributed linearly separable computation,” *IEEE Trans. Commun.*, vol. 69, no. 11, pp. 7390–7405, 2021.

[26] H. L. Van Trees, *Optimum array processing: Part IV of detection, estimation, and modulation theory*. John Wiley & Sons, 2004.

[27] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[28] S. Nath, P. B. Gibbons, S. Seshan, and Z. R. Anderson, “Synopsis diffusion for robust aggregation in sensor networks,” in *Proc. Int. Conf. Embedded Networked Sensor Syst.*, 2004, p. 250–262.

[29] K. Wan, H. Sun, M. Ji, and G. Caire, “Distributed linearly separable computation,” *IEEE Trans. Inf. Theory*, vol. 68, no. 2, pp. 1259–1278, 2022.

[30] A. Khalesi and P. Elia, “Multi-user linearly-separable distributed computing,” *IEEE Trans. Inf. Theory*, vol. 69, no. 10, pp. 6314–6339, 2023.

[31] ——, “Tessellated distributed computing,” *IEEE Trans. Inf. Theory*, vol. 71, no. 6, pp. 4754–4784, 2025.

[32] J. Korner and K. Marton, “How to encode the modulo-two sum of binary sources (corresp.),” *IEEE Trans. Inf. Theory*, vol. 25, no. 2, pp. 219–221, 1979.

[33] D. Slepian and J. Wolf, “Noiseless coding of correlated information sources,” *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[34] T. Han and K. Kobayashi, “A dichotomy of functions $f(x, y)$ of correlated sources (x, y),” *IEEE Trans. Inf. Theory*, vol. 33, no. 1, pp. 69–76, 1987.

[35] A. B. Wagner, “On distributed compression of linear functions,” *IEEE Trans. Inf. Theory*, vol. 57, no. 1, pp. 79–94, 2011.

[36] V. Lalitha, N. Prakash, K. Vinodh, P. V. Kumar, and S. S. Pradhan, “Linear coding schemes for the distributed computation of subspaces,” *IEEE J. Sel. Areas Commun.*, vol. 31, no. 4, pp. 678–690, 2013.

[37] R. Appuswamy, M. Franceschetti, N. Karamchandani, and K. Zeger, “Network coding for computing: Cut-set bounds,” *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 1015–1030, 2011.

[38] X. Guang, R. W. Yeung, S. Yang, and C. Li, “Improved upper bound on the network function computing capacity,” *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3790–3811, 2019.

[39] X. Guang and R. Zhang, “Zero-error distributed compression of binary arithmetic sum,” *IEEE Trans. Inf. Theory*, vol. 70, no. 5, pp. 3100–3117, 2024.

[40] S. Feizi and M. Médard, “On network functional compression,” *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5387–5401, 2014.

[41] D. Malak, M. R. Deylam Salehi, B. Serbetci, and P. Elia, “Multi-server multi-function distributed computation,” *Entropy*, vol. 26, no. 6, 2024.

[42] S. H. Lim, C. Feng, A. Pastore, B. Nazer, and M. Gastpar, “Towards an algebraic network information theory: Distributed lossy computation of linear functions,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2019, pp. 1827–1831.

[43] X. Guang, X. Sun, and R. Zhang, "Distributed source coding for compressing vector-linear functions," 2025. [Online]. Available: https://arxiv.org/abs/2508.02996

[44] H. Chen, M. Cheng, and Y. Wu, "On the fundamental limits of decentralized linearly separable computation under cyclic assignment," *IEEE Trans. Commun.*, pp. 1–1, 2025.

[45] Y. Bi, M. Wigger, and Y. Wu, "Normalized delivery time of wireless MapReduce," *IEEE Trans. Inf. Theory*, vol. 70, no. 10, pp. 7005–7022, 2024.

[46] K. Wan, H. Sun, M. Ji, D. Tuninetti, and G. Caire, "On the optimal load-memory tradeoff of cache-aided scalar linear function retrieval," *IEEE Trans. Inf. Theory*, vol. 67, no. 6, pp. 4001–4018, 2021.

[47] Y. Yao and S. A. Jafar, "The capacity of 3 user linear computation broadcast," *IEEE Trans. Inf. Theory*, vol. 70, no. 6, pp. 4414–4438, 2024.

[48] Y. Ma and D. Tuninetti, "An achievable scheme for the $K-$user linear computation broadcast channel," 2025. [Online]. Available: https://arxiv.org/abs/2501.12322

[49] N. Raviv, I. Tamo, R. Tandon, and A. G. Dimakis, "Gradient coding from cyclic MDS codes and expander graphs," *IEEE Trans. Inf. Theory*, vol. 66, no. 12, pp. 7475–7489, 2020.

[50] S. Li, S. M. M. Kalan, A. S. Avestimehr, and M. Soltanolkotabi, "Near-optimal straggler mitigation for distributed gradient methods," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. Workshops (IPDPSW)*, 2018, pp. 857–866.

[51] A. Gholami, T. Jahani-Nezhad, K. Wan, and G. Caire, "Optimal communication-computation trade-off in hierarchical gradient coding," 2025. [Online]. Available: https://arxiv.org/abs/2502.18251

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

[53] S. Wang, J. Liu, N. Shroff, and P. Yang, "Computation efficient coded linear transform," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2019, pp. 577–585.

[54] 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 *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2019, pp. 1215–1225.

[55] Z. Jia and S. A. Jafar, "On the capacity of secure distributed batch matrix multiplication," *IEEE Trans. Inf. Theory*, vol. 67, no. 11, pp. 7420–7437, 2021.

[56] W.-T. Chang and R. Tandon, "On the capacity of secure distributed matrix multiplication," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2018, pp. 1–6.

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

[58] R. G. L. D'Oliveira, S. El Rouayheb, and D. Karpuk, "GASP codes for secure distributed matrix multiplication," *IEEE Trans. Inf. Theory*, vol. 66, no. 7, pp. 4038–4050, 2020.

[59] O. Makkonen and C. Hollanti, "Analog secure distributed matrix multiplication over complex numbers," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2022, pp. 1211–1216.

[60] C. Hofmeister, R. Bitar, and A. Wachter-Zeh, "CAT and DOG: Improved codes for private distributed matrix multiplication," 2025. [Online]. Available: https://arxiv.org/abs/2501.12371

[61] F. Montecalvo, "Some constructions of general covering designs," *Electron. J. Combinatorics*, vol. 19, no. 3, pp. 1–16, 2012.

[62] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes*. Elsevier, 1977.

[63] J. Fulman, "Random matrix theory over finite fields," *Bulletin of the American Mathematical Society*, vol. 39, no. 1, pp. 51–85, Oct. 2001.

[64] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[65] P. Crescenzi, F. Montecalvo, and G. Rossi, “Optimal covering designs: complexity results and new bounds,” *Discrete Appl. Math.*, vol. 144, no. 3, pp. 281–290, 2004.

[66] C. J. Colbourn and J. H. Dinitz, *Handbook of Combinatorial Designs, Second Edition (Discrete Mathematics and Its Applications).* Chapman & Hall/CRC, 2006.

[67] F. Montecalvo, “Asymptotic bounds for general covering designs,” *J. Combinatorial Des.*, vol. 23, no. 1, pp. 18–44, 2014.