EURECOM
Campus SophiaTech
CS 50193
06904 Sophia Antipolis cedex FRANCE

Research Report RR-22-348

Multi-user Linearly Separable Computation Sparse Factorization Meets Coding Theory

Friday 20th May, 2022 (10:05)

Ali Khalesi, Petros Elia

Tel : (+33) 4 93 00 81 00
Fax : (+33) 4 93 00 82 00
Email : {Ali.Khalesi, Petros.Elia}@eurecom.fr

[1] EURECOM's research is partially supported by its industrial members: BMW Group Research and Technology, IABG, Orange, SAP, NortonLifeLock

Abstract

In this work, we explore the problem of multi-user linearly separable computation, where N servers help compute the desired functions (jobs) of K users, and each desired function can be written as a linear combination of up to L (generally non-linear) subtasks (or sub-functions). Each server computes some of the subtasks, and communicates a linear combination of its computed outputs to a fraction of the users, where then each user linearly combines its received data in order to recover its desired function. We explore the computation and communication relationship between how many subtasks each server computes vs. how much data each user receives. For a matrix $\mathbf{F}$ representing the linear coefficients of the set of requested functions, our problem becomes equivalent to the open problem of matrix factorization $\mathbf{F} = \mathbf{D}\mathbf{E}$ over finite fields, where a sparse decoding matrix $\mathbf{D}$ and encoding matrix $\mathbf{E}$ imply reduced communication and computation costs respectively. This paper establishes a novel relationship between our problem, matrix factorization, syndrome decoding and covering codes. To reduce the computation cost, the above $\mathbf{D}$ is drawn from a here-introduced class of so-called partial-covering codes, whose study here yields the computation cost bounds that we present. To then reduce the communication cost, these coding-theoretic properties are explored in the regime of codes that have low-density parity check matrices. The work reveals — first for the commonly used one-shot scenario — that in the limit of large N, the optimal computation cost per server scales as a parameter $\gamma = \rho \in (H_q^{-1}(\frac{\log_q(L)}{N}), H_q^{-1}(K/N))$ — where H_q is q-ary entropy function — and that this can be achieved with communication cost that scales as $O(\sqrt{\log_q(N)})$. This in turn reveals the role of the computational rate $\log_q(L)/N$, showing that this rate cannot exceed what one might call the computational capacity $H_q(\gamma)$ of the system. We show that our coded approach yields unbounded gains over the uncoded scenario. In the end, we also explore the multi-shot scenario, for which we derive bounds on the computational cost.

Multi-user Linearly Separable Computation
Sparse Factorization Meets Coding Theory

Ali Khalesi and Petros Elia

Abstract

In this work, we explore the problem of multi-user linearly separable computation, where N servers help compute the desired functions (jobs) of K users, and each desired function can be written as a linear combination of up to L (generally non-linear) subtasks (or sub-functions). Each server computes some of the subtasks, and communicates a linear combination of its computed outputs to a fraction of the users, where then each user linearly combines its received data in order to recover its desired function. We explore the computation and communication relationship between how many subtasks each server computes vs. how much data each user receives.

For a matrix $\mathbf{F}$ representing the linear coefficients of the set of requested functions, our problem becomes equivalent to the open problem of matrix factorization $\mathbf{F} = \mathbf{D}\mathbf{E}$ over finite fields, where a sparse decoding matrix $\mathbf{D}$ and encoding matrix $\mathbf{E}$ imply reduced communication and computation costs respectively. This paper establishes a novel relationship between our problem, matrix factorization, syndrome decoding and covering codes. To reduce the computation cost, the above $\mathbf{D}$ is drawn from a here-introduced class of so-called partial-covering codes, whose study here yields the computation cost bounds that we present. To then reduce the communication cost, these coding-theoretic properties are explored in the regime of codes that have low-density parity check matrices. The work reveals — first for the commonly used one-shot scenario — that in the limit of large N, the optimal computation cost per server scales as a parameter $\gamma = \rho \in (H_q^{-1}(\frac{\log_q(L)}{N}), H_q^{-1}(K/N))$ — where H_q is q-ary entropy function — and that this can be achieved with communication cost that scales as $O(\sqrt{\log_q(N)})$. This in turn reveals the role of the computational rate $\log_q(L)/N$, showing that this rate cannot exceed what one might call the computational capacity $H_q(\gamma)$ of the system. We show that our coded approach yields unbounded gains over the uncoded scenario. In the end, we also explore the multi-shot scenario, for which we derive bounds on the computational cost.

Keywords

Distributed computation, Linearly separable function, Coding theory, Sparse matrix factorization.

I. Introduction

There is an ever-increasing need for distributed computing systems that can speed up processing of non-linear and computationally hard functions. The main goal of distributed computing is to utilize parallel processing techniques to offload computations to a group of distributed servers so that the computation time is reduced. This parallelization relates to various frameworks — such as MapReduce [1] and Spark [2] — and it entails several challenges that involve accuracy [3]–[6], scalability [7]–[11], privacy and security [12]–[24], as well as latency and straggler mitigation [25]–[32]. For a detailed survey of some of these efforts, the reader is referred to [33], [34]. A crucial ingredient in distributed computing involves the communication complexity which refers to the amount of communication required to solve a computational problem when the desired task is distributed among two or more parties [35]. This celebrated computation-vs-communication relationship has been studied in a variety of different forms and scenarios [27], [36]–[46] for various types of problems.

a) Preliminary description of setting: The same relationship between computation and communication costs, is the topic of interest in our work here for a very broad and practical setting of multi-user, multiserver computation of linearly-separable functions. Such functions appear in several classes of problems such as for example in training large-scale machine learning algorithms and deep neural networks with massive data [36], where computation cost is crucial [47], [48].

In particular, our setting here considers a master node that manages N server nodes that must contribute in a distributed manner to the computation of the desired function by each of the K different users.

Under the linearly-separable assumption (cf. [49]), we consider that user $k \in \{1, 2, \ldots, K\}$ demands a function $F_k(D_1, D_2, \ldots, D_L)$ that takes as input L datasets $D_1, D_2, \ldots, D_L$, and

each such requested function takes the basic form

$$F_k(D_1, \ldots, D_L) = \sum_{\ell=1}^{L} f_{k,\ell} f_\ell(D_\ell) = \sum_{\ell=1}^{L} f_{k,\ell} W_\ell \tag{1}$$

where in the above, $W_\ell = f(D_\ell)$ denotes the computed output when the input is D_ℓ, and where $f_{k,\ell}$ are the combining coefficients which belong, together with the entries of W_ℓ, in some finite field. Upon notification of the users' requests — which are jointly described by the $K \times L$ matrix $\mathbf{F}$ that contains the different coefficients $f_{k,\ell}$ — the master instructs the servers to compute some of the functions $f(D_\ell)$ for a group of datasets. Each server may compute a different number of functions, and the more the functions, the more the computational cost. Upon completing their computations, each server communicates linear combinations of its locally computed outputs (files) to carefully selected subsets of users. Each user can then only linearly combine what it receives by all the servers that have transmitted to it, and the goal is for each user to recover its desired function. The problem is completed when every user k retrieves their desired $f_k(D_1, \ldots, D_L)$.

We note that there is a clear differentiation between the server nodes which are asked to compute hard (generally non-linear) functions, and the users that can only linearly combine their received outputs. Generating the so-called output file $W_\ell = f(D_\ell), \ell \in \{1, 2, \ldots, L\}$ can be the result of a computationally intensive task that may for example relate to training a deep learning model on a dataset, or it can relate to the distributed gradient coding problem [25], [50]–[52], the distributed linear-transform computation problem [38], [53], or even the distributed matrix multiplication and the distributed multivariate polynomial computation problems [26], [29]–[31], [41], [54]–[58].

b) Brief summary of the basic ingredients of the problem: Our setting brings to the fore the following crucial questions.

- How many and which functions $f(D_i)$ must each server compute?
  - This defines the computation cost per server: the more the functions that each server must compute, the higher the complexity (computational cost) at that server. The extreme centralized scenario would imply a maximal computational delay, as it would imply that the one active server would need to compute all L sub-functions. This same centralized setting though would imply minimal communication cost, equal to (as we can see) one shot per user. The other extreme scenario would imply a minimal computation cost of L/N sub-functions/jobs per server, but a maximal communication cost of N shots received per user.
- What linear combinations of its computed outputs must each server generate?
  - These linear combination coefficients define an $N \times L$ matrix $\mathbf{E}$ that describes the encoding done at the different servers. This matrix must be designed as a function of the jobs which are described by the $K \times L$ matrix $\mathbf{F}$.
  - The number of non-zero elements in $\mathbf{E}$ reflects the computation cost on the collective of servers.
- How many such linear combinations (of locally computed outputs) must each server communicate, and to how many users?
  - This defines the communication cost. The more data each user gets, the higher the cost.
- How must each user combine (linearly decode) the computed outputs arriving from the servers?
  - This step is determined by a $K \times N$ decoding matrix $\mathbf{D}$ which must be carefully designed. The number of non-zero elements of $\mathbf{D}$ reflects our communication cost. If for example the kth row of $\mathbf{D}$ has many non-zero elements, then the kth user must receive data from many servers.
- How sparse can $\mathbf{D}$ and $\mathbf{E}$ be so that each user recovers their desired function?

  - This defines the overall costs in computation and communication. As one might expect, the larger the number L of possible subtasks/datasets, the higher the worst-case costs.

To answer these questions, we take a novel approach that employs coding theory. The general idea behind our approach is described as follows.

c) Brief summary of the new connection to sparse matrix factorization and coding theory:

- *Connection with sparse matrix factorization:* First, when exploring our distributed computing problem, one can see that the feasibility conditions that ensure that each user recovers its desired function, constitute in fact a (preferably sparse) matrix factorization problem of the form

$$\mathbf{D}\mathbf{E} = \mathbf{F} \tag{2}$$

  where the problem is over some q-sized finite field $\mathbb{F}$, and where any potential sparsity of $\mathbf{D}$ and $\mathbf{E}$ translates to savings in communication and computation costs respectively.
- *Connection to coding theory and syndrome decoding:* To then resolve this problem in a manner that yields non-trivial sparse factors, we notice that — if for example, we were to fix the above matrix $\mathbf{D}$, and associate this to the parity-check matrix of some linear code — then for each column $\mathbf{E}_l$ of $\mathbf{E}$ and associated column $\mathbf{F}_l$ of $\mathbf{F}$, the corresponding equation $\mathbf{D} \cdot \mathbf{E}_l = \mathbf{F}_l$ would tells us that the desired sparse $\mathbf{E}_l$ can be the lowest-weight coset leader whose syndrome is equal to $\mathbf{F}_l$. Hence the columns of $\mathbf{E}$ are associated to error vectors, the columns of $\mathbf{F}$ to the corresponding syndromes, and $\mathbf{D}$ is assigned the role of a parity check matrix, and the question is of which code?
- *Connection to covering codes and the new class of partially covering codes:* The above connection with syndromes in turn brings about the concept of covering codes that refer to codes with good covering properties, which in turn entail low weight $\mathbf{E}_l$, which is what we need. In coding theory though — where any error vector is possible — such covering codes consider a full space of possible syndromes, where any appropriately-dimensioned vector can indeed be a syndrome. To account for the fact though that $\mathbf{F}$ corresponds to a *restricted* set of syndromes, we here extend the theory of covering codes to the new class of *partial covering codes*, the analysis of which is an interesting and non-trivial coding-theoretic contribution of this work.
- *Connection with codes having low-density parity-check matrices:* The above effort is concluded when the aforementioned exploration of covering and partial covering codes (which yielded a sparse $\mathbf{E}$), is extended to involve analysis of codes with a sparse $\mathbf{D}$.
- *Extending the one-shot scenario:* Our framework allows us to address but also extend the one-shot scenario which is the scenario of choice in various works (see for example [49]) and which asks that each server can send only one linear combination to one set of users. In addition to this model, we

here also consider the practical and realistic scenario where, for the same fixed subset of tasks/files $\{f(D_\ell)\}$ computed locally at each server, the server can communicate linear combinations to various sets of users.

d) Highlights of contributions: Our focus is on establishing the normalized computation[1] cost $\gamma = \frac{1}{N}\max_{l\in\{1,\dots,L\}} \omega(\mathbf{E}(:,l))$, and the normalized communication cost $\delta = \omega(\mathbf{D})/KN$. In our setting, $\gamma \in [0,1]$ represents the maximum fraction of all servers that must compute any subfunction, while $\delta \in [0,1]$ represents the average fraction of servers each user gets data from. Hence in our setting, $\Delta = \delta N$ represents the average number of 'symbols' received by each user.

We do so first for the one-shot case. We proceed to highlight some of the derived results. The exact rigorous expressions can be found in the following chapters.

- Theorem 1 makes the connection between coding theory and our distributed computing problem, by showing that a (γ,δ)-feasible distributed computing scheme exists if and only if the decoding matrix $\mathbf{D}$ is the parity check matrix of an N-length code $\mathcal{C} \subset \mathbb{F}^N$ over a field $\mathbb{F}$ where this code has minimum normalized distance from each vector $\{\mathbf{x} \in \mathbb{F}^N | \mathbf{D}\mathbf{x} = \mathbf{F}(:,l), l \in \{1,\dots,L\}\}$ that is at most γN. This brings to the fore the concept of covering and partial covering codes, where covering codes are codes that guarantee a minimum distance to each vector of the entire vector space, while partial covering codes must guarantee a minimum distance to only a specific subset of the entire space. Establishing the properties of such partial covering codes is key to our problem.
- Theorem 2 shows that in the limit of large N, the optimal computation cost per server is in the range $\gamma \in (H_q^{-1}(\frac{\log_q(L)}{N}), H_q^{-1}(K/N))$, where H_q is the entropy function over our field of size q. This theorem reveals the role of what one might refer to as the *functional rate* $R_f = \log_q(L)/N$. The higher this rate, the more 'involved' is the space of functions we are asked to compute over. In this sense — given that $\frac{log_q(L)}{N} \leq H_q(\gamma)$ — the expression $H_q(\gamma)$ plays the role of an upper bound on what one might call the *functional capacity* of the system.
- Then by extending the famous covering codes Theorem of Blinovskii from [59], we extend our bounds on partial covering codes to the setting of codes with low density parity check matrices, revealing that the aforementioned complexity γ can be achieved with communication cost that scales as $\Delta = O(\sqrt{\log_q(N)})$. This latter cost is unboundedly better than the uncoded approach of resource-sharing between the two extreme regimes discussed previously in the introduction (See Figure 4 in Section IV-D).

[1] Both communication and computation costs will be defined in more detail later on. Also, in the following, $\omega()$ represents the well known Hamming weight of the argument vector or matrix.

- We also consider the multi-shot scenario where, for the same fixed subset of tasks/files $\{f(D_\ell)\}$ computed locally at each server, the server can communicate linear combinations to various sets of users. For this setting, Theorem 4 reveals a range of parameters for which the multi-shot approach provides computational savings over the single-shot scenario.

A. Paper Organization

The rest of the paper is organized as follows. Section II introduces the Multi-user linearly separable system model. Section III formulates our problem, focusing on the single-shot scenario, for which Section IV presents the main results. This latter section first makes the connection to coding theory, and then presents the converse and achievability on computation cost, as well as the bound on the communication cost. Section IV-D offers some insights including a discussion on the gains due to coding. Subsequently, in Section V, we present our proposed achievable multi-shot scheme and the corresponding results, and finally we conclude in Section VI.

Notations: We define $[n] \triangleq \{1, 2, \ldots, n\}$. For matrices $\mathbf{A}$ and $\mathbf{B}$, $[\mathbf{A}, \mathbf{B}]$ indicates the horizontal concatenation of two matrices. For any matrix $\mathbf{X} \in \mathbb{F}^{m \times n}$, then $\mathbf{X}(i, j),\ i \in [m],\ j \in [n]$ represents the entry in the ith row and jth column, while $\mathbf{X}(i, :),\ i \in [m]$, represents the ith row, and $\mathbf{X}(:, j),\ j \in [n]$ represents the jth column of $\mathbf{X}$. For two index sets $\mathcal{I} \subset [m], \mathcal{J} \in [n]$, then $\mathbf{X}(\mathcal{I}, \mathcal{J})$ represents the sub-matrix comprised of the rows in $\mathcal{I}$ and columns in $\mathcal{J}$. We will use $\omega(\mathbf{X})$ to represent the number of nonzero elements of some matrix (or vector) $\mathbf{X}$. We denote the finite field $\mathbf{GF}(q)$ as $\mathbb{F}$. For any code $\mathcal{C} \subseteq \mathbb{F}^n$ and any vector $\mathbf{x} \in \mathbb{F}^n$, we use $d(\mathbf{x}, \mathcal{C})$ to represent the hamming distance of $\mathbf{x}$ to the nearest codeword in $\mathcal{C}$. We will dedicate the use of the letter ρ when referring to normalized covering radii, and we will often use $\rho(\mathcal{C})$ to indicate the normalized covering radius of a specific code $\mathcal{C} \in \mathbb{F}^n$. We will often use the notation $\mathcal{C}_{\mathbf{H}}$ to refer to a code whose parity check matrix is $\mathbf{H}$, and similarly, we will use notation $\mathbf{H}_{\mathcal{C}}$ to refer to a matrix that is the parity-check matrix of a specific linear code $\mathcal{C}$. For some $k \leq n,\ k, n \in \mathbb{N}$, we will also often use the notation $\mathcal{C}(k, n)$ to emphasise that a linear code has message length k to codeword length n. For any two codes $\mathcal{C}_1$ and $\mathcal{C}_2$, we will use $[\mathcal{C}_1, \mathcal{C}_2]$ to represent the code resulting from direct product of two codes. For some vector $\mathbf{x} \in \mathbb{F}^n$, we will use $\mathcal{C}_2 =< \mathbf{x}, \mathcal{C}_1 >$ to represents a code whose span is the union of $\mathbf{x}$ with the span of a code $\mathcal{C}_1$. Furthermore $V_q(n, \rho)$ will represent the volume of a Hamming ball in $\mathbb{F}^n$ of radius ρn. For $0 \leq x \leq 1 - \frac{1}{q}, x \in \mathbb{R}$, to represent the q-ary entropy function we will use $H_q(x) \triangleq x \log_q(q-1) - x \log_q(x) - (1-x) \log_q(1-x)$, while when $q = 2$ we will use the simplified $H(x)$. We will use $\sup(\mathbf{x}^\intercal)$ to represent the support of some vector $\mathbf{x}^\intercal \in \mathbb{F}^n$, representing the set of indices of non-zero elements. We will also use the notation $\epsilon(n)$ to represent an expression which, in the large n setting, goes to zero.

II. System Model

We consider the multi-user linearly-separable distributed computation setting (cf. Fig. 1), which consists of K users, N active (non-idle) servers, and a master node that coordinates servers and users. The main two characteristics of this setting is that the tasks performed at the servers, substantially outweigh in computational complexity the linear operations performed at the different users, and also that the cost of having the servers communicate to the users is indeed non-trivial. We consider the setting where each server can use T consecutive 'shots' to communicate different messages to different subsets of users, where in particular, during shot (time-slot) $t \in [T]$, server n communicates to some arbitrary user set $\mathcal{T}_{n,t} \subset [K]$.

In our setting, each user asks for a (generally non-linear) function from a space of linearly separable functions, where each such function takes as input several datasets. Each function can be decomposed into a different linear combination of individual (again generally non-linear, and computationally hard) sub-functions $f_\ell(D_\ell)$ that each take a single dataset D_ℓ as input (by definition of what linearly-separable means). Thus each user $k \in [K]$ demands a function $F_k(D_1, \ldots, D_L)$ of L independent datasets D_l, $l \in [L]$, where this function takes the general linearly-separable form

$$F_k(D_1, D_2, \ldots, D_L) \triangleq f_{k,1} f_1(D_1) + f_{k,2} f_2(D_2) + \ldots + f_{k,L} f_L(D_L), \;\; k \in [K] \tag{3}$$

$$= f_{k,1} W_1 + f_{k,2} W_2 + \ldots + f_{k,L} W_L, \;\; k \in [K] \tag{4}$$

where in the above, $W_\ell = f_\ell(D_\ell) \in \mathbb{F}$, $l \in [L]$ is a so-called 'file' output, and $f_{k,\ell} \in \mathbb{F}$, $k \in [K], \ell \in [L]$ are the linear combination coefficients.

A. Phases

The model involves three phases, with the first being the *demand phase*, then the *assignment and computation phase* and then the *transmission and decoding phase*. In the demand phase, each user $k \in [K]$ sends the information of its desired function $F_k(.)$ to the master node, who then deduces the linear decomposition of this function according to (4). Then based on these K desired functions, during the assignment and computation phase, the master assigns some of the datasets to each server, who then proceeds to calculate the corresponding files $W_\ell = f_\ell(D_\ell)$ for their locally available datasets. Based on this assignment, each dataset D_ℓ will be placed at all the servers in some chosen set $\mathcal{W}_\ell$.

During the transmission phase, each server $n \in [N]$ transmits T shots during time slots $t = 1, 2, \ldots, T$, where each transmission is in the form of a linear combination of the locally available output files at the

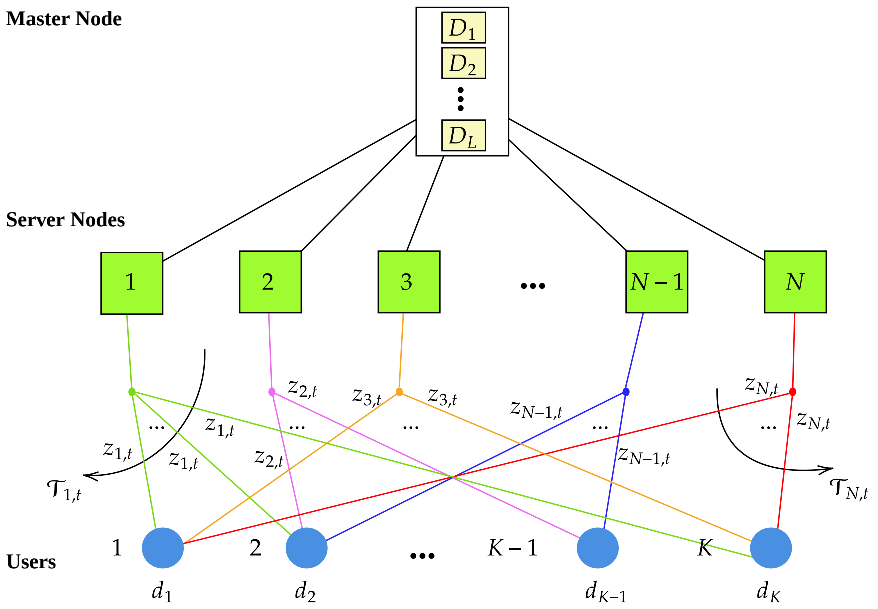

Fig. 1. The K-user, N-server Linearly Separable Computation setting. After each user informs the master of its desired function $F_k(.)$, each component subfunction $W_\ell = f_\ell(D_\ell)$ is evaluated at each server in $\mathcal{W}_\ell$. During time slot t, each server n transmits a linear combination $z_{n,t}$ (of the locally available files) to all users in $\mathcal{T}_{n,t}$. This combination is defined by the coefficients $e_{n,\ell,t}$. Finally, to decode, each user $k \in [K]$ linearly combines (based on decoding vectors $\mathbf{d}_k$) all the received signals from all the slots and all the servers it is connected to. Decoding must produce for each user its desired function $F_k(D_1, \ldots, D_L)$.

server, and where each such value is destined for some subset of users $\mathcal{T}_{n,t}$. In particular, during time slot t, each server n transmits

$$z_{n,t} \triangleq \sum_{\ell \in [L]} e_{n,l,t} W_l, \;\; n \in [N], t \in [T] \tag{5}$$

where $e_{n,l,t} \in \mathbb{F}$ are the so-called encoding coefficients determined by the master. Finally during the decoding part, each user k linearly combines the received signals as follows

$$F'_k \triangleq \sum_{n \in [N], t \in [T]} d_{k,n,t} z_{n,t} \tag{6}$$

for some decoding coefficients $d_{k,n,t} \in \mathbb{F}, n \in [N], t \in [T]$ determined again by the master node. Naturally $d_{k,n,t} = 0, \forall k \notin \mathcal{T}_{n,t}$. Decoding is successful when $F'_k = F_k$ for all $k \in [K]$.

B. Computation and Communication Costs

Remembering that $|\mathcal{W}_\ell|$ indicates the number of servers that compute a subfunction $W_\ell = f_\ell(D_\ell)$, $\ell \in [L]$, our *normalized computational cost* metric takes the form

$$\gamma \triangleq \frac{\max\limits_{l \in [L]} |\mathcal{W}_\ell|}{N} \tag{7}$$

to represent the maximum fraction of all servers that must compute any subfunction.

We also formally define the *normalized communication cost* as

$$\delta \triangleq \frac{\sum_{t=1}^{T} \sum_{n=1}^{N} |\mathcal{T}_{n,t}|}{KN} \tag{8}$$

to represent the average fraction of servers that each user gets data from. Hence in our setting,

$$\Delta \triangleq \delta N \tag{9}$$

represents the average number of transmitted 'symbols' received by each user. We wish to provide schemes that correctly compute the desired functions, at reduced computation and communication costs.

III. Problem Formulation: One-Shot Setting

In this single shot setting with $T = 1$, we will remove the use of the index t. Thus the transmitted value from (5) will take the form

$$z_n = \sum_{\ell \in [L]} e_{n,l} W_l, \;\; n \in [N] \tag{10}$$

where $e_{n,l} \in \mathbb{F}$ will be the corresponding encoding coefficients, and where each such transmitted value at server n will now be destined for the users in set $\mathcal{T}_n$. Similarly the decoding value at each user k (cf. (6)) will take the form $F'_k \triangleq \sum_{n \in [N]} d_{k,n} z_n$ for $d_{k,n}, n \in [N]$ being the decoding coefficients. The desired functions $F_k(.)$ (cf. (4)), their linear decomposition coefficients $f_{k,\ell}$ (cf. (4)), and the decoded functions $F'_k(.)$ (6) remain the same. With the above in place, we will use

$$\mathbf{f} \triangleq [F_1, F_2, \ldots, F_K]^\intercal \tag{11}$$

$$\mathbf{f}_k \triangleq [f_{k,1}, f_{k,2}, \ldots, f_{k,L}]^\intercal \; k \in [K], \tag{12}$$

$$\mathbf{w} \triangleq [W_1, W_2, \ldots, W_L]^\intercal \tag{13}$$

where $\mathbf{f}$ represents the vector of the output demanded functions (cf. (4)), $\mathbf{f}_k$ the vector of function coefficients for user k (cf. (4)), and $\mathbf{w}$ the vector of output files. We also have

$$\mathbf{e}_n \triangleq [e_{n,1}, e_{n,2}, \ldots, e_{n,L}]^\intercal, \; n \in [N] \tag{14}$$

$$\mathbf{z} \triangleq [z_1, z_2, \ldots, z_N]^\intercal \tag{15}$$

respectively representing the encoding vector at server n and the overall transmitted vector across all the servers (cf. (10)). Furthermore we have

$$\mathbf{d}_k \triangleq [d_{k,1}, d_{k,2}, \ldots, d_{k,N}]^\intercal, \; k \in [K] \tag{16}$$

$$\mathbf{f}' \triangleq [F'_1, F'_2, \ldots, F'_K]^\intercal \tag{17}$$

respectively representing the decoding vector at user k, and the vector of the decoded functions across all the users. Furthermore we have

$$\mathbf{F} \triangleq [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_K]^\intercal \in \mathbb{F}^{K\times L} \tag{18}$$

$$\mathbf{E} \triangleq [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N]^\intercal \in \mathbb{F}^{N\times L} \tag{19}$$

$$\mathbf{D} \triangleq [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K]^\intercal \in \mathbb{F}^{K\times N} \tag{20}$$

where $\mathbf{F}$ represents the $K \times L$ matrix of all function coefficients across all the users, where $\mathbf{E}$ represents the $N \times L$ *encoding matrix* across all the servers, and where $\mathbf{D}$ represents the $K \times N$ *decoding matrix* across all the users.

Directly from (4), we have that

$$\mathbf{f} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_K]^\intercal \mathbf{w} \tag{21}$$

and from (5) we have the overall transmitted vector taking the form

$$\mathbf{z} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N]^\intercal \mathbf{w} = \mathbf{E}\mathbf{w}. \tag{22}$$

Furthermore, directly from (6) we have that

$$F'_k = \mathbf{d}_k^T \mathbf{z} \tag{23}$$

and thus we have

$$\mathbf{f}' = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K]^\intercal \mathbf{z} = \mathbf{D}\mathbf{z}. \tag{24}$$

Recall that we must guarantee that

$$\mathbf{f}' = \mathbf{f}. \tag{25}$$

After substituting (21), (22) and (24) into (25), we see that the above feasibility condition in (25) is satisfied if

$$\mathbf{D}\mathbf{E}\mathbf{w} = \mathbf{F}\mathbf{w}. \tag{26}$$

Given that naturally, the server has not computed the output files in $\mathbf{w}$, and given that we wish (26) to hold for all $\mathbf{w}$, we can conclude that for feasibility to hold, we must guarantee

$$\mathbf{D}\mathbf{E} = \mathbf{F}. \tag{27}$$

At this point, since $\mathcal{W}_\ell = \sup(\mathbf{E}(:, \{l\})^\intercal)$, and since $|\mathcal{W}_\ell| = \omega(\mathbf{E}(:, \{l\}))$, we have that

$$\max_{l\in[L]} \omega(\mathbf{E}(:, l)) = \max_{\ell\in[L]} |\mathcal{W}_\ell| \tag{28}$$

which simply tells us that our computational cost γ from (7) takes the form

$$\gamma = \frac{1}{N}\max_{l\in[L]} \omega(\mathbf{E}(:,l)). \tag{29}$$

Similarly, directly from (6) and (9), we see that

$$\delta = \frac{\omega(\mathbf{D})}{KN} \tag{30}$$

which simply says that

$$\Delta = \frac{\omega(\mathbf{D})}{K}. \tag{31}$$

It is now clear that decomposing $\mathbf{F}$ into the product of two relatively sparse matrices $\mathbf{D}$ and $\mathbf{E}$, implies reduced communication and computation costs respectively.

We here provide a simple example to help clarify the setting and the notations.

A. Simple Example

As described in Figure 2, we consider the example of a system with a master node, $N = 8$ servers, $K = 4$ users, $L = 6$ datasets, and a field of size $q = 7$.

Let us assume that the users ask the following functions:

$$F_1 = 2f_1(D_1) + 4f_2(D_2) + 4f_3(D_3) + 5f_4(D_4) + 5f_5(D_5) = \mathbf{f}_1^{\intercal}\mathbf{w}, \tag{32}$$

$$F_2 = 3f_1(D_1) + 4f_2(D_2) + 5f_3(D_3) + 2f_4(D_4) + 6f_5(D_5) + 6f_6(D_6) = \mathbf{f}_2^{\intercal}\mathbf{w}, \tag{33}$$

$$F_3 = 2f_1(D_1) + 4f_2(D_2) + 6f_3(D_3) + 5f_4(D_4) + 2f_5(D_5) = \mathbf{f}_3^{\intercal}\mathbf{w}, \tag{34}$$

$$F_4 = 3f_1(D_1) + 5f_2(D_2) + 2f_4(D_4) + 3f_5(D_5) + f_6(D_6) = \mathbf{f}_4^{\intercal}\mathbf{w} \tag{35}$$

where $F_k, \mathbf{f}_k,\ k \in [4]$ and $\mathbf{w}$ are respectively defined in (4), (13) and (12). Consequently from (18), our demand matrix takes the form

$$\mathbf{F} = \begin{bmatrix} 2 & 4 & 4 & 5 & 5 & 0 \\ 3 & 4 & 5 & 2 & 6 & 6 \\ 2 & 4 & 6 & 5 & 2 & 0 \\ 3 & 5 & 0 & 2 & 3 & 1 \end{bmatrix}. \tag{36}$$

In the assignment phase, the master allocates $D_1, D_2, \ldots, D_6$ to the 8 servers according to

$$\mathcal{W}_1 = \{1,2,3,5,8\},\ \mathcal{W}_2 = \{1,2,3,4,6,7\},\ \mathcal{W}_3 = \{1,2,3\},\ \mathcal{W}_4 = \{1,4,5,7\} \tag{37}$$

$$\mathcal{W}_5 = \{1,2,4,5,6,8\},\ \mathcal{W}_6 = \{3,4,5,6,7,8\} \tag{38}$$

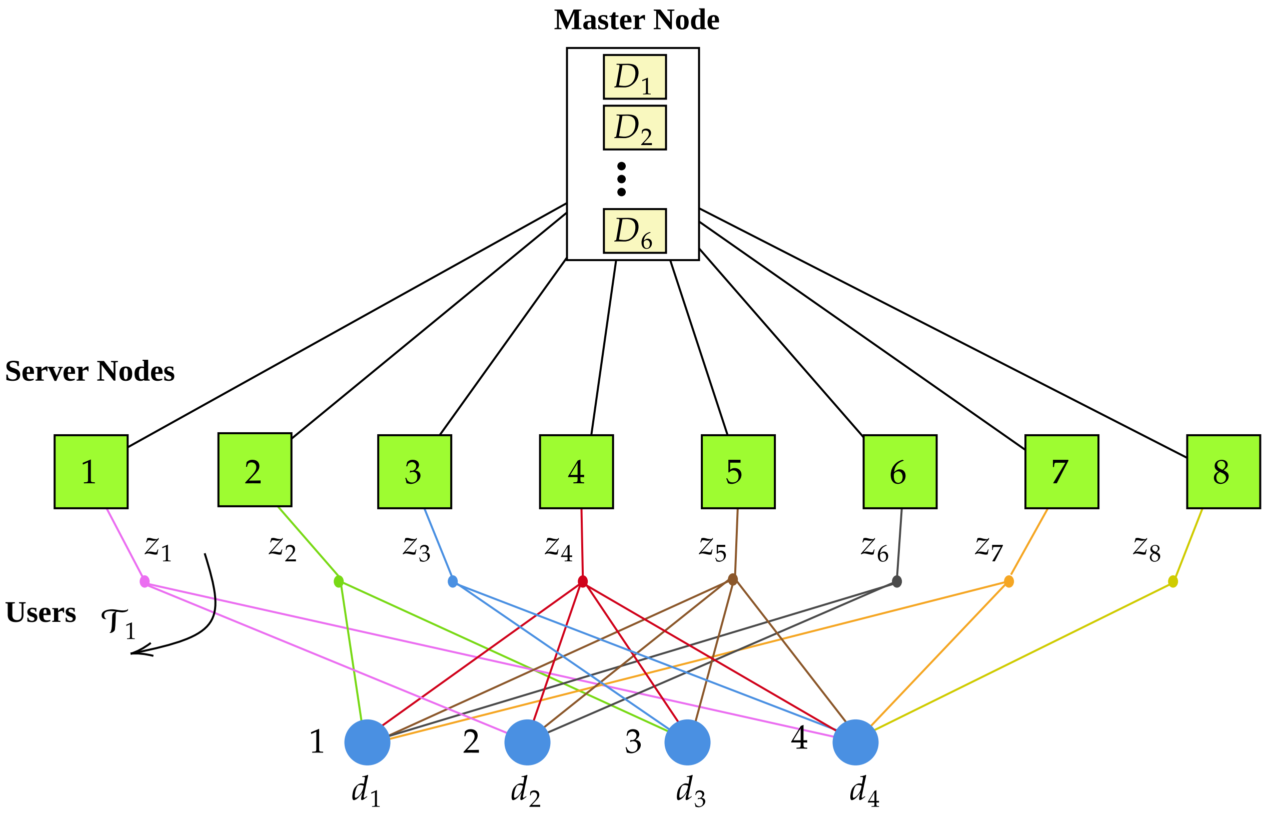

Fig. 2. Multi-user linearly separable setting with 8 servers, 4 users and 6 datasets.

so that for example dataset 3 resides at servers $\{1, 2, 3\}$, or equivalently, server 2 is assigned datasets D_1, D_2, D_3, D_5 and thus has to compute $W_1 = f(D_1), W_2 = f(D_2), W_3 = f(D_3), W_5 = f(D_5)$. A quick inspection shows that the normalized computation cost (cf. (7)) is equal to

$$\gamma = \frac{\max_{l \in [6]} |\mathcal{W}_\ell|}{8} = 6/8. \tag{39}$$

After computing their designated output files, each server n transmits z_n as follows

$$z_1 = 2W_1 + 6W_2 + 3W_3 + W_4 + 2W_5, \quad z_2 = 4W_1 + 5W_2 + 2W_3 + 3W_5, \tag{40}$$

$$z_3 = W_1 + 2W_2 + W_3 + 2W_6, \quad z_4 = W_2 + 2W_4 + 4W_5 + W_6, \tag{41}$$

$$z_5 = 2W_1 + W_4 + 3W_5 + 2W_6, \quad z_6 = 2W_2 + 5W_5 + 3W_6 \tag{42}$$

$$z_7 = W_2 + 2W_4 + 4W_6, \quad z_8 = 2W_1 + 4W_5 + 5W_6 \tag{43}$$

corresponding to an encoding matrix (cf. (22)) of the form

$$\mathbf{E} = \begin{bmatrix} 2 & 6 & 3 & 1 & 2 & 0 \\ 4 & 5 & 2 & 0 & 3 & 0 \\ 1 & 2 & 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 2 & 4 & 1 \\ 2 & 0 & 0 & 1 & 3 & 2 \\ 0 & 2 & 0 & 0 & 5 & 3 \\ 0 & 1 & 0 & 2 & 0 & 4 \\ 2 & 0 & 0 & 0 & 4 & 5 \end{bmatrix}. \tag{44}$$

We can quickly verify (cf. (39)) that indeed $\max\limits_{l\in[6]} \omega(\mathbf{E}(:,l))/8 = 6/8 = \gamma$.

Subsequently, the master asks each server n to send its generated z_n to its designated receiving users, such that for each server, these user sets are:

$$\mathcal{T}_1 = \{2,4\},\ \mathcal{T}_2 = \{1,3\},\ \mathcal{T}_3 = \{3\},\ \mathcal{T}_4 = \{1,2,3,4\}, \tag{45}$$

$$\mathcal{T}_5 = \{1,2,3,4\},\ \mathcal{T}_6 = \{1,2\}, \mathcal{T}_7 = \{1,4\}, \mathcal{T}_8 = \{4\}, \tag{46}$$

where for example server 2 will transmit z_2 to users 1 and 3. A quick inspection also shows that users 1 and 4 receive 5 different symbols, whereas users 2 and 3 receive 4 symbols each. This corresponds to a normalized communication cost (cf. (9)) equal to

$$\delta = \frac{\sum_{n=1}^{8} |\mathcal{T}_n|}{4\cdot 8} = (5+4+4+6)/32 = 19/32 \tag{47}$$

corresponding to an average of $\Delta = \frac{19}{4}$ symbols received per user.

To decode, each user $k \in [4]$ computes the linear combination F'_k as

$$\begin{aligned} F'_1 &= 2z_2 + 3z_4 + 4z_5 + 2z_6 + z_7, & F'_2 &= 4z_1 + 2z_4 + z_5 + 3z_6 \\ F'_3 &= 4z_2 + 5z_3 + 2z_4 + z_5, & F'_4 &= 4z_1 + 2z_3 + z_4 + 2z_5 + 4z_7 + 5z_8 \end{aligned} \tag{48}$$

corresponding to a decoding matrix of the form

$$\mathbf{D} = \begin{bmatrix} 0 & 2 & 0 & 3 & 4 & 2 & 1 & 0 \\ 4 & 0 & 0 & 2 & 1 & 3 & 0 & 0 \\ 0 & 4 & 5 & 2 & 1 & 0 & 0 & 0 \\ 4 & 0 & 2 & 1 & 2 & 0 & 4 & 5 \end{bmatrix}. \tag{49}$$

A quick verification[2] reveals the correctness of decoding and that indeed $F'_k = F_k$ for all $k = 1,2,3,4$. For example, for the first user, we see that $F'_1 = 2z_2+3z_4+4z_5+2z_6+z_7 = 2(4W_1+5W_2+2W_3+3W_5)+ 3(W_2+2W_4+4W_5+W_6)+4(2W_1+W_4+3W_5+2W_6)+2(2W_2+5W_5+3W_6)+(W_2+2W_4+4W_6) =$

[2] Let us recall that each decoded symbol takes the form $F'_k = \mathbf{d}_k^{\intercal}\mathbf{z}$ where $\mathbf{d}_k^{\intercal}$ is the kth row of $\mathbf{D}$, and where $\mathbf{z} = [z_1\ z_2\ \cdots\ z_N]^T$.

$2W_1 + 4W_2 + 4W_3 + 5W_4 + 5W_5 + 0W_6$ which indeed matches F_1. In this example, each user recovers their desired function, with a corresponding normalized computational cost $\gamma = 3/4$ and a normalized communication cost $\delta = 19/32$. This has just been an example to illustrate the setting. The effort to find a solution with reduced computation and communication costs, follows in the section below.

IV. Computation-vs-Communication: The One-Shot Setting

In this section we present the results for the one-shot setting. We first rigorously establish the bridge between our problem, coding theory, covering and partial covering codes. The main results — focusing first on the computational aspects — are presented in Section IV-B which derives bounds on the optimal computational cost in the large N setting. With these results in place, the subsequent Section IV-C extends our consideration to the communication cost as well. Finally, Section IV-D offers some intuition on the results of this current section.

We briefly recall (cf. [60]) that an n-length code $\mathcal{C} \subset \mathbb{F}^n$ is called a ρ-covering code if it satisfies

$$d(\mathbf{x}, \mathcal{C}) \leq \rho n, \ \ \forall \mathbf{x} \in \mathbb{F}^n \tag{50}$$

for some $\rho \in (0,1)$ which is referred to as the normalized covering radius.

A. Establishing a relationship to coding theory

We will first seek to decompose $\mathbf{F}$ into $\mathbf{F} = \mathbf{D}\mathbf{E}$ under a constrained computation cost, i.e., under a sparsity constraint on $\mathbf{E}$. For $\mathbf{E}_l \triangleq \mathbf{E}(:,l)$ and $\mathbf{F}_l \triangleq \mathbf{F}(:,l)$ denoting the lth column of $\mathbf{D}$ and $\mathbf{E}$ respectively, we can rewrite our decomposition as

$$\mathbf{D}\mathbf{E}_l = \mathbf{F}_l, \quad l \in [L]. \tag{51}$$

If we viewed $\mathbf{D} \in \mathbb{F}^{K \times N}$ as a parity check matrix $\mathbf{H}_{\mathcal{C}} = \mathbf{D}$ of a code $\mathcal{C} \subset \mathbb{F}^N$, then we could view $\mathbf{E}_l \in \mathbb{F}^N$ as an arbitrary error pattern, and $\mathbf{F}_l \in \mathbb{F}^K$ as the corresponding syndrome. Since we wish to sparsify $\mathbf{E}_l$, we are interested in $\mathbf{E}_l$ being the minimum-weight coset leader whose syndrome is $\mathbf{F}_l$. This is simply the output of the minimum-distance syndrome decoder[3]. To get a first handle on the weights of $\mathbf{E}_l$, we can refer to the theory of covering codes which bounds the weights of coset leaders, where these weights are bounded by the code's covering radius $\rho(\mathcal{C})N$, for some normalized radius $\rho(\mathcal{C}) \in (0,1)$. This covering radius $\rho(\mathcal{C})N$ upper bounds the weights of the coset leaders. [4] Hence the covering radius upper bounds our computational cost.

[3] Naturally our viewing $\mathbf{D}$ as a parity check matrix, does not limit the scope of options in choosing $\mathbf{D}$. Similarly, associating $\mathbf{E}_l$ the role of an error pattern, or a minimum-weight coset leader, is again not a limiting association.

[4] Let us recall (cf. [61]) that the (preferred) coset leaders are the minimum-weight vectors in each row of the standard array.

To capture some of the coding-theoretic properties, we will transition to the traditional coding-theoretic notation which speaks of an n-length code $\mathcal{C}$ of rate k/n, where for us $n = N$ and $k = N - K$. The parity check matrix $\mathbf{H}_{\mathcal{C}} \in \mathbb{F}^{(n-k)\times n}$ will generally be associated to our decoding matrix $\mathbf{D} \in \mathbb{F}^{K\times N}$, the received (or error) vectors $\mathbf{x} \in \mathbb{F}^{n}$ will be associated to the encoding vectors $\mathbf{E}_l \in \mathbb{F}^{N}$, and its syndrome $\mathbf{s}_{\mathbf{x}} \in \mathbb{F}^{n-k}$ (or just $\mathbf{s}$ depending on the occasion) will be associated to $\mathbf{F}_l \in \mathbb{F}^{K}$. When we write $\mathcal{C}_{\mathbf{D}}$ (or $\mathcal{C}_{\mathbf{H}}$) we will refer to the code whose parity check matrix is $\mathbf{D}$ (or $\mathbf{H}$).

As a first step, we extend the concept of covering codes to the following class.

Definition 1. For some $\rho \in (0, 1]$, we say that a set $\mathcal{X} \subseteq \mathbb{F}^{n}$ is ρ-covered by a code $\mathcal{C} \subseteq \mathbb{F}^{n}$ iff

$$d(\mathbf{x}, \mathcal{C}) \leq \rho n, \ \ \forall \mathbf{x} \in \mathcal{X} \tag{52}$$

in which case, we say that $\mathcal{C}$ is a $(\rho, \mathcal{X})$-partial covering code.

Naturally when $\mathcal{X} = \mathbb{F}^{n}$, such a partial covering code is simply the traditional covering code. We are now able to link partial covering codes to our distributed computing problem.

Theorem 1. *A solution to the multi-user linearly separable problem* $\mathbf{DE} = \mathbf{F}$ *with normalized computational cost* γ *exists if and only if* $\mathbf{D}$ *is the parity check matrix to a* $(\gamma, \mathcal{X})$*-partial covering code* $\mathcal{C}_{\mathbf{D}}$ *for some existing set, where* $\mathcal{X}_{\mathbf{F},\mathbf{D}}$ *is defined as,*

$$\mathcal{X} \supset \mathcal{X}_{\mathbf{F},\mathbf{D}} \triangleq \{\mathbf{x} \in \mathbb{F}^{N} | \mathbf{D}\mathbf{x} = \mathbf{F}(:, l), \ \textit{for some } l \in [L]\}. \tag{53}$$

With such $\mathbf{D}$ *in place, each* $\mathbf{E}(:, l)$ *is the output of the minimum-distance syndrome decoder of* $\mathcal{C}_{\mathbf{D}}$ *for syndrome* $\mathbf{F}(:, l)$*.*

Proof. To first prove that the complexity constraint indeed requires $\mathbf{D}$ to correspond to a partial covering code that covers $\mathcal{X}$, let us assume that $\mathbf{D}$ does not have this property, and that there exists an $\mathbf{x} \in \mathcal{X}$ such that $d(\mathbf{x}, \mathcal{C}_{\mathbf{D}}) > \rho n$. Let $\mathbf{c}_{\min}$ be the closest codeword to $\mathbf{x}$ in the sense that $d(\mathbf{x}, \mathbf{c}_{\min}) = d(\mathbf{x}, \mathcal{C}_{\mathbf{D}})$. Now let $\mathbf{e}_{\min} = \mathbf{x} - \mathbf{c}_{\min}$ and note, directly from the above assumption, that $\omega(\mathbf{e}_{\min}) > \rho n$. Naturally $\mathbf{D}\mathbf{x} = \mathbf{D}(\mathbf{e}_{\min} + \mathbf{c}_{min}) = \mathbf{D}\mathbf{e}_{\min}$ by virtue of the fact that $\mathbf{D}$ is the parity check matrix of $\mathcal{C}_{\mathbf{D}}$. Since $\mathbf{x} \in \mathcal{X}$, we know that $\exists \, l \in [L]$ such that $\mathbf{D}\mathbf{x} = \mathbf{F}(:, l)$ which directly means that $\exists \, l \in [L]$ such that $\mathbf{D}\mathbf{e}_{\min} = \mathbf{F}(:, l)$. This $\mathbf{e}_{\min}$ is the coset leader associated to syndrome $\mathbf{F}(:, l)$.

Since though $\mathbf{DE} = \mathbf{F}$, we also have that $\mathbf{DE}(:, l) = \mathbf{F}(:, l)$. Since $\mathbf{E}(:, l)$ and $\mathbf{e}_{\min}$ are in the same coset (of the same syndrome $\mathbf{F}(:, l)$), and since $\mathbf{e}_{\min}$ is the minimum-weight coset leader, we can conclude that $\omega(\mathbf{E}(:, l)) \geq \mathbf{e}_{\min}$. Thus the assumption that $\omega(\mathbf{e}_{\min}) > \rho n$ implies that $\omega(\mathbf{E}(:, l)) > \rho n$ which contradicts the complexity requirement that $\omega(\mathbf{E}(:, l)) \leq \rho n$ from (29). Thus if $\mathbf{D}$ does not correspond to a partial covering code that covers $\mathcal{X}_{\mathbf{F},\mathbf{D}}$, the complexity constraint is violated.

On the other hand, recalling that $\mathcal{C}_{\mathbf{D}}$ is a partial covering code for $\mathcal{X}$, means that for any $\mathbf{x} \in \mathcal{X}$ then $d(\mathbf{x}, \mathcal{C}_{\mathbf{D}}) \leq \rho n$. For the same $\mathbf{x} \in \mathcal{X}$, let $\mathbf{c}_{\min}$ be again its closest codeword, and let $\mathbf{e}_{\min} = \mathbf{x} - \mathbf{c}_{\min}$, where again by definition of the partial covering code, $\omega(\mathbf{e}_{\min}) \leq \rho n$. Since, like before, $\mathbf{D}\mathbf{e}_{\min} = \mathbf{F}(:, l)$ for some $l \in [L]$, then we simply set $\mathbf{E}(:, l) = \mathbf{e}_{\min}$ whose weight is indeed sufficiently low to guarantee the complexity constraint. We recall that for each $\mathbf{F}(:, l)$, this coset leader $\mathbf{E}(:, l) = \mathbf{e}_{\min}$ can be found using the minimum-distance syndrome decoder.

□

Now that we have established the connection with partial covering codes, we present the fundamental results on Subsection IV-B.

B. Bounds on The Optimal Computation Cost

The following result bounds the optimal computational cost of any solution of the multi-user linearly-separable computation.

Theorem 2. *For the distributed linearly separable problem with K users, N servers and any number of L subfunctions, the optimal computation cost is bounded as*

$$\gamma \in (H_q^{-1}(\frac{\log_q(L)}{N}), H_q^{-1}(\frac{K}{N})). \tag{54}$$

Proof. The proof of the converse (lower bound in (54)) employs sphere-covering arguments for a partial covering code, and can be found in Appendix A. The proof of achievability (upper bound in (54)) results from covering and partial covering-code arguments, and can be found in Appendix B.

□

Remark 1. The above metric γ captures the degree of sparsity of the encoding matrix $\mathbf{E}$, and thus describes the maximum fraction of all servers that must compute any subfunction. The theorem reveals that the optimal worst-case computational load, in units of subfunctions computed per server, is lower-bounded by $LH_q^{-1}(\log_q(L)/N)$ and upper-bounded by $LH_q^{-1}(K/N)$. The two bounds meet when $L = q^K$.

Theorem 2 suggests a range of computational costs. In the next corollary, we will describe the conditions under which a reduced normalized computational cost, inside this range, can be achieved. This reduced cost will relate to (our ability to choose) a set $\mathcal{X} \subset \mathbb{F}^N$. As we will see, a smaller $\mathcal{X}$ will imply a smaller γ. To understand the connection between our problem and this set $\mathcal{X}$, and thus to better understand the following theorem whose proof will be fully presented in Appendix C, we provide the following sketch of some crucial elements in the proof of Theorem 2. In particular, we will here sketch an algorithm that iterates in order to converge to the aforementioned $\mathcal{X}$, to the corresponding decoding matrix $\mathbf{D}$, and then to the corresponding normalized complexity γ. Before describing the algorithm, it is worth noting that a crucial ingredient can be found in Lemma 4 (see Appendix D), which modifies the approach in [62] to design — for any set $\mathcal{X}' \in \mathbb{F}^N$ — a $(\rho, \mathcal{X}')$-partial covering code for some $\rho = H_q^{-1}(\frac{K}{N} - (1 - \frac{\log_q(|\mathcal{X}'|)}{N})$.

With this in place, the algorithm starts by picking an initial set $\mathcal{X}_0 \in \mathbb{F}^N, |\mathcal{X}_0| = Lq^{N-K}$, and then applies Lemma 4 to construct a $(\rho_0, \mathcal{X}_0)$-partial covering code, $\mathcal{C}_0$, where $\rho_0 = H_q^{-1}(\frac{K}{N} - (1 - \frac{\log_q(|\mathcal{X}_0|)}{N})$. With this code in place, we create — as a function of $\mathcal{C}_0$ — the set $\mathcal{X}_{\mathbf{F},\mathbf{D},0}$ as defined in (53) where $\mathbf{D} = \mathbf{H}_{\mathcal{C}_0}$, and then we check if $\mathcal{X}_0 \supseteq \mathcal{X}_{\mathbf{F},\mathbf{D},0}$. If so, then the algorithm terminates, else we go to the next iteration which starts by picking a new larger set $\mathcal{X}_1 \in \mathbb{F}^N, |\mathcal{X}_1| = Lq^{N-K} + 1$, then uses Lemma 4 to create a new $(\rho_1, \mathcal{X}_1)$-partial covering code for $\rho_1 = H_q^{-1}(\frac{K}{N} - (1 - \frac{\log_q(|\mathcal{X}_1|)}{N})$, and then compares if $\mathcal{X}_1 \supseteq \mathcal{X}_{\mathbf{F},\mathbf{D},1}$. This procedure terminates during some round m where this terminating round is the first round for which the chosen set $\mathcal{X}_m$ (now of cardinality $|\mathcal{X}_m| = Lq^{N-K} + m$) and the corresponding $(\rho_m, \mathcal{X}_m)$-partial covering code with $\rho_m = H_q^{-1}(\frac{K}{N} - (1 - \frac{\log_q(|\mathcal{X}_m|)}{N})$, yield $\mathcal{X}_m \supseteq \mathcal{X}_{\mathbf{F},\mathbf{D},m}$.

In the following Corollary, the mentioned $\mathcal{X}$ refers to the terminating[5] $\mathcal{X}_m$, and the decoding matrix $\mathbf{D}$ will be the parity-check matrix of the aforementioned $(\rho_m, \mathcal{X}_m)$-partial covering code that covers the terminating $\mathcal{X} = \mathcal{X}_m$, while the normalized computation cost in the theorem will take the form $\gamma = \rho = \rho_m$.

With the above in place, the following Theorem speaks of a set $\mathcal{X}$ that is ρN-covered by a code $\mathcal{C}_{\mathbf{D}}$ that generates — as described in (53) — its set $\mathcal{X}_{\mathbf{F},\mathbf{D}}$.

Corollary 1. If there exists a set $\mathcal{X}$ that is ρN covered by a code $\mathcal{C}_{\mathbf{D}}$ such that $\mathcal{X} \supseteq \mathcal{X}_{\mathbf{F},\mathbf{D}}$, then the computation cost $\rho \leq H_q^{-1}(\frac{K}{N} - (1 - \frac{\log_q(|\mathcal{X}|)}{N}) + \epsilon(N))$ is achievable. If $\mathcal{X} = \mathcal{X}_{\mathbf{F},\mathbf{D}}$ then the computation cost becomes $H_q^{-1}(\frac{\log_q(L)}{N})$ and thus becomes the optimal.

Proof. The proof can be found in Appendix C. □

As suggested before, one can imagine that covering a smaller $\mathcal{X}$ could imply the existence of a smaller covering radius, which in our case implies a smaller normalized computational cost.

C. Single Shot Schemes with sub-Optimal Computation and Communication Cost

The following Theorem 3 combines computation and communication considerations. Theorem 3 builds on Theorem 1, where now we consider that any chosen decoding matrix $\mathbf{D}$ will automatically yield a

[5] Note that in the worst case this termination will happen when $\mathcal{X}_m = \mathbb{F}^N$, in which case the output code will be a covering code.

communication cost $\Delta = \frac{\omega(\mathbf{H})}{K}$.

Theorem 3. *For the distributed linearly separable problem with K users, N servers and L subfunctions, the optimal computation cost is bounded as*

$$\gamma \in (H_q^{-1}(\frac{\log_q(L)}{N}), H_q^{-1}(\frac{K}{N})). \tag{55}$$

and for any achievable computational cost $\gamma \leq \min\{\frac{\sqrt{5}-1}{2}, 1-\frac{1}{q}\}$, then the corresponding achievable communication cost takes the form

$$\Delta = O(\sqrt{\log_q(N)}). \tag{56}$$

Proof. The proof can be found in Appendix E. □

We here offer a quick sketch of the proof of the above theorem. The proof first employs a modified version of the famous result by Blinovskii in [59] which proved that, as n goes to infinity, almost all random linear codes $\mathcal{C}(k,n)$ are covering codes, as long as the normalized covering radius satisfies $\rho \geq H_q^{-1}(\frac{n-k}{n})$. This modification of Blinovskii's theorem is presented in Theorem 5, whose proof if found in Appendix E. With this modification in place, we proved that almost all $(k,n))$ random linear codes with

$$\rho = H_q^{-1}(\frac{\log_q(|\mathcal{X}|)-k}{n}) \tag{57}$$

are $(\rho,\mathcal{X})$-partial covering codes, each for some set $\mathcal{X} \in \mathbb{F}^n$. This is again in Theorem 5. With this theorem in place, we then employ a concatenation argument (this can be found in the proof of Theorem 3 in Appendix E), we were be able to build a sparse parity-check matrix $\mathbf{H}$ of a partial covering code. Then going back to Theorem 1, completes the proof of Theorem 3. Having the covering code helps bound the computational cost, and having this code being sparse, helps bound the communication cost.

To shed some more light on the effort to prove that sparse parity check codes can indeed offer reduced computational costs, we had to show that sparse codes can indeed offer good partial-covering properties. To do that, we followed some of the steps described below. In particular, we designed an algorithm that begins with constructing a sparse parity check code that can cover, for a given radius ρ_0, a minimum necessary cardinality set $\mathcal{X}_0$ where this minimum cardinality of $|\mathcal{X}_0| = Lq^{N-K}$ is imposed on us by $\mathbf{F}$. The parity-check matrix of this first code is $\mathbf{H}_0$. Then following the steps in the proof of Theorem 1, we set $\mathbf{D} = \mathbf{H}_0$ and check if $\mathcal{X}_0 \supseteq \mathcal{X}_{\mathbf{F},\mathbf{D}}$ holds. If it indeed holds, the algorithm outputs $\mathbf{D}$ and $\mathcal{X}_0$, and the corresponding complexity is $\gamma = \rho_0$, where this ρ value is derived from (57) by setting $\mathcal{X} = \mathcal{X}_0$. Otherwise the algorithm constructs another partial sparse covering code with a new parity check matrix $\mathbf{H}_1$, now covering a set $\mathcal{X}_1$ with cardinality $|\mathcal{X}_1| = Lq^{N-K}+1$, and then checks again the same inclusion condition as above. The procedure continues until it terminates, with some covered set $\mathcal{X}_m$ of cardinality $|\mathcal{X}_m| = Lq^{N-K} + m$. As before, reaching $\mathcal{X}_m = \mathbb{F}^N$ will terminate the algorithm (if it has not terminated before that). In the proposition below, the set $\mathcal{X}$ is exactly our terminating set $\mathcal{X}_m$ referred to above.

Proposition 1. *Focusing on the achievable scheme proposed in Theorem 3 and its corresponding* $\mathbf{D}$ *that was designed as a function of* $\mathbf{F}$ *then if there exists a subset* $\mathcal{X} \supseteq \mathcal{X}_{\mathbf{F},\mathbf{D}}, \mathcal{X} \subseteq \mathbb{F}^N$ *that is* ρN*-covered by* $\mathcal{C}_{\mathbf{D}}$*, then the computation cost* $\rho = H_q^{-1}(\frac{K}{N} - (1 - \frac{\log_q(|\mathcal{X}|)}{N}))$ *is achievable. If* $\mathcal{X} = \mathcal{X}_{\mathbf{F},\mathbf{D}}$ *then the computation cost converges to the optimal* $H_q^{-1}(\frac{\log_q(L)}{N})$*. The above remains in place for any* $\mathbf{D}$ *which yields communication cost no less than* $\Delta = O(\sqrt{\log_q(N)})$*.*

Proof. The proof can be found in Appendix G. □

Note that such like Corollary 1, covering a smaller $\mathcal{X}$ could imply the existence of a smaller covering radius, which in our case implies a smaller normalized computational cost.

D. Discussion and Comparison

Looking at Theorem 3, we see that the optimal computation cost lies in the region $\gamma \in (H_q^{-1}(\frac{\log_q(L)}{N}), H_q^{-1}(\frac{K}{N}))$ and that this can be achieved with communication cost of the form $\Delta = O(\sqrt{\log_q(N)})$ corresponding to $\delta = O(\sqrt{\log_q(N)})/N$. To get a better understanding of the improvements that come from our coded approach, let us compare it to the uncoded single-shot case. Let us look at Figure 3, and the two points labeled point 1 and point 2. Point 1 corresponds to $(\gamma = 1/N, \delta = 1)$ and it can be achieved by having (because of the single-shot assumption) $N(q-1) = L$ datasets, each having to compute a single subfunction, which in turn implies that each server has to be connected to all the users. This example corresponds to the case where $\mathbf{D} = \mathbf{F} \in \mathbb{F}^{K \times N(q-1)}$ and $\mathbf{E} = \mathbf{I}_{N \times N}$, an identity matrix. Note that this example reaches is optimal since $L = N(q-1) = \binom{N}{1}(q-1) \simeq q^{NH_q(1/N)} = q^{NH_q(\gamma)}$ where N is big enough. $\delta = 1$ is the case where $\mathbf{F}$ contains no zero element. Note that point 1 is representation of all such examples that achieves our established converse in Theorem 2 with maximum communication cost thus they are optimal in terms of normalized computation cost.

On the other hand, point 2 corresponds to $(\gamma = 1, \delta = 1/K)$ and it can be achieved by activating K servers and then (again because of the single-shot assumption) asking each server to compute all L subfunctions, and asking each server to transmit a single message to a single user. The line connecting the two points (by employing time-sharing) describes the optimal performance under the one-shot uncoded assumption.

Then point 3 is a guaranteed achievable point $(\gamma = H_q^{-1}(\frac{K}{N}), \delta = O(\sqrt{\log_q(N)})/N)$ from our approach, while point 4 with $(\gamma = H_q^{-1}(\frac{\log_q(L)}{N}), \delta = O(\sqrt{\log_q(N)})/N)$ is conditionally achievable

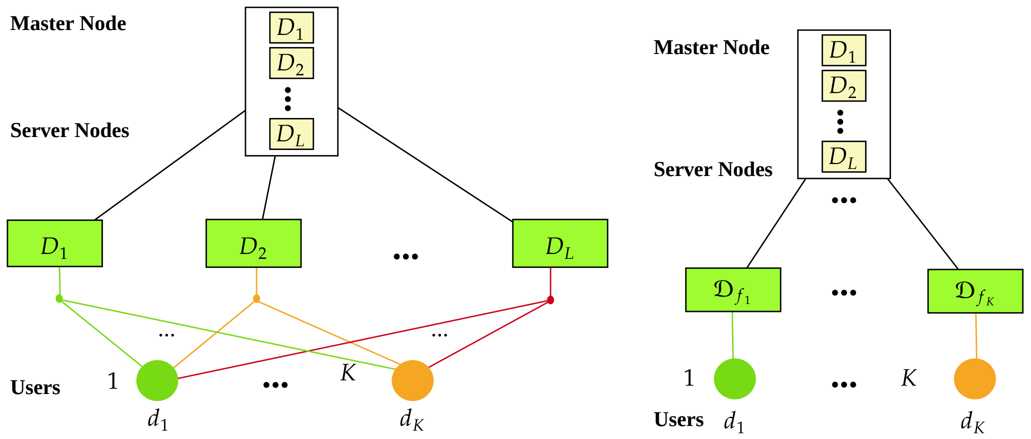

Fig. 3. (Left): Uncoded scheme for Point 1 corresponding to $(\gamma = 1/N, \delta = 1)$. Each of the $N(q-1) = L$ servers, computes one subfunction, but sends to all K users. (Right): Uncoded scheme for Point 2 corresponding to $(\gamma = 1, \delta = 1/K)$. Activate K servers, each computes L subfunctions, and each transmits to a single user.

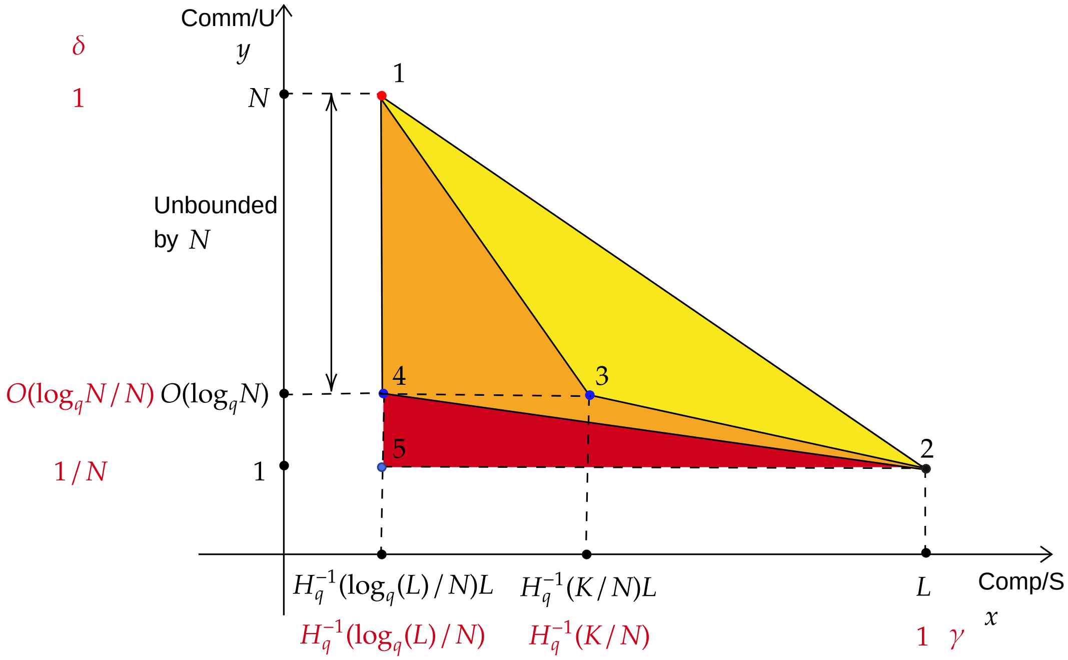

Fig. 4. The plot summarizes the results of Theorem 3, discussed in subsection IV-D. Note that K/N and $\log_q(L)/N$ are fixed while N approaches to infinity so that Theorem 3 holds.

(see Theorem 1). No point left of point 4 can be achieved, and the triangle between points $5, 2, 4$ could be achievable under techniques that further reduce the sparsity of $\mathbf{D}$. The details of the above discussion is presented in Figure 4.

V. Achievable Results for The Multi-Shot Setting ($T > 1$)

In this Section we present our achievable results for the multi-shot setting where NT, the number of servers multiplied by the number of shots, goes to infinity while $K/NT, \frac{\log_q(L)}{NT}$ are fixed. The investigation of such setting is motivated by a practical perspective, where each server is able to send T separate different linear combination of files. To save on the computation resource of each server. The method we use in this Section is similar to the Section IV. To this end, first in the Subsection V-A, we are formalizing the problem such like Section III, next in the Subsection V-B, we establish the same relationship of the multi-shot problem to the coding theory, in particular we show that how the achievable schemes of Theorem 2 for the one-shot setting helps us to get an achievable scheme for the multi-shot setting which bounds the normalized computation. In the Subsection V-C, we present the total results and finally in Subsection IV-C, we compare our results to the previous known schemes and the one shot setting.

A. Problem Formulation: Multi-Shot Setting

The general multi-shot formulation has been described in this section, where we recall that the main difference is addition of t, the time slot or shot to some of the parameters. We define the following variables such as what has been done in Section III, for the one-shot system model.

$$\mathbf{f} \triangleq [F_1, F_2, \ldots, F_K]]^\intercal \tag{58}$$

$$\mathbf{f}_k \triangleq [f_{k,1}, f_{k,2}, \ldots, f_{k,L}]^\intercal,\ k \in [K], \tag{59}$$

$$\mathbf{w} \triangleq [W_1, W_2, \ldots, W_L]^\intercal, \tag{60}$$

$$\tag{61}$$

In the next definitions, we see that the index of t is included to show that for which shot or slot, the encoding coefficients and transmit symbols is produced.

$$\mathbf{e}_{n,t} \triangleq [e_{n,1,t}, e_{n,2,t}, \ldots, e_{n,L,t}]^\intercal,\ n \in [N],\ t \in [T] \tag{62}$$

$$\mathbf{z}_t \triangleq [z_{1,t}, z_{2,t}, \ldots, z_{N,t}]^\intercal,\ t \in [T] \tag{63}$$

$$\mathbf{z} \triangleq [\mathbf{z}_1^\intercal, \mathbf{z}_2^\intercal, \ldots, \mathbf{z}_T^\intercal]^\intercal, \tag{64}$$

Also for decoding the index t is included so that the decoding coefficients for each received signal during tth shot be differentiated. Note that the decoded message of each user is produced by a linear combination of all received signal during all T shots.

$$\mathbf{d}_{k,t} \triangleq [d_{k,1,t}, d_{k,2,t}, \dots, d_{k,N,t}]^\intercal,\ k \in [K], t \in [T] \tag{65}$$

$$\mathbf{d}_{k} \triangleq [\mathbf{d}_{k,1}^\intercal, \mathbf{d}_{k,2}^\intercal, \dots, \mathbf{d}_{k,T}^\intercal]^\intercal,\ k \in [K] \tag{66}$$

$$\mathbf{f}' \triangleq [F'_1, F'_2, \dots, F'_K]^\intercal \tag{67}$$

$$\mathbf{F} \triangleq [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_K]^\intercal \tag{68}$$

$$\mathbf{E}_t \triangleq [\mathbf{e}_{1,t}, \mathbf{e}_{2,t}, \dots, \mathbf{e}_{N,t}]^\intercal,\ t \in [T] \tag{69}$$

Again from (4), it can be inferred

$$\mathbf{f} = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_K]^\intercal \mathbf{w} \tag{70}$$

Also from (5) where $\mathbf{e}_{n,t}$ consists of encoding coefficients of server n and t shot, we conclude that

$$\mathbf{z}_t = \mathbf{E}_t \mathbf{w} = [\mathbf{e}_{1,t}, \mathbf{e}_{2,t}, \dots, \mathbf{e}_{N,t}]^\intercal \mathbf{w} \tag{71}$$

which indicates encoded files to be sent in the t-th shot of the transmission phase. Cumulating all shots we have

$$\mathbf{z} = \mathbf{E}\mathbf{w} \tag{72}$$

where $\mathbf{E}$ is similarly called *Encoding Matrix* and is defined as follows

$$\mathbf{E} \triangleq [\mathbf{E}_1^\intercal, \mathbf{E}_2^\intercal, \dots, \mathbf{E}_T^\intercal]^\intercal \in \mathbb{F}^{NT \times L}. \tag{73}$$

From (6) where decoding coefficients $d_{k,n,t}$ for all servers $n \in [N]$ and shots $t \in [T]$ is included $\mathbf{d}_k$, we have for each user

$$F'_k = \mathbf{d}_k^T \mathbf{z} \tag{74}$$

where writing that for all users we have

$$\mathbf{f}' = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_K]^\intercal \mathbf{z}. \tag{75}$$

We also define a Decoding Matrix as well

$$\mathbf{D} \triangleq [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_K]^\intercal \in \mathbb{F}^{K \times NT}. \tag{76}$$

Therefore the scheme is correct if and only if for any $D_l, l \in [L]$, we have

$$\mathbf{f} = \mathbf{f}'. \tag{77}$$

By substituting (70), (71), (75) into (77) and make sure that it holds for all possible $W_l, l \in [L]$, we conclude that if and only if

$$\mathbf{D}\mathbf{E} = \mathbf{F}, \tag{78}$$

the scheme is can successfully deliver the desired functions of each user for all possible produced files $\mathbf{w} \in \mathbb{F}^{L\times 1}$. The reasoning here is also similar to the derivation of (27).

Cost Formulation: According to the system model, the master node has to allocate the minimum number of necessary datasets, so that each server $n \in [N]$ contains all the datasets in $\cup_{t=1}^{T} \sup(\mathbf{e}_{n,t})$, therefore we have for all feasible schemes satisfying (27),

$$\cup_{t=1}^{T} \sup(\mathbf{E}([(t-1)N+1 : tN], \{l\})^\intercal) = \mathcal{W}_\ell, \forall \ell \in [L]. \forall t \in [T]. \tag{79}$$

where $\mathbf{E}$ is defined in (73). Also note that

$$\omega(\mathbf{E}(:,l)) = \sum_{t=1}^{T} |\sup(\mathbf{E}([(t-1)N+1, tN], \{l\})^\intercal)| \geq |\mathcal{W}_\ell|,\ l \in [L] \tag{80}$$

where the non-equality is resulted from the union bound on the cardinality of sets having (79). Then we see that,

$$\max_{l\in[L]} \omega(\mathbf{E}(:,l)) \geq \max_{\ell\in[L]} |\mathcal{W}_\ell| \tag{81}$$

Note that the equality results where $T = 1$, the one-shot setting case which has been extensively investigated in Section IV. In the subsequent sections we use (81) to characterize the computation cost. It means that the non-zero elements of $\mathbf{E}(:,l)$ normalized by N active servers, is just an upper bound on the fraction of servers working on any particular sub-function.

From (6) and (9), we again see that

$$\omega(\mathbf{D}) = \Delta K. \tag{82}$$

B. Establishing a Relationship to the coding theory

We use the same similarities established between (51) and syndrome decoder described in Subsection IV-A, except that here $\mathbf{E} \in \mathbb{F}^{K\times NT}$ and $\mathbf{D} \in \mathbb{F}^{NT\times L}$ therefore the similar linear code $\mathcal{C}(n,k)$ has to have the dimensions $NT = n, K = n-k$ while $K \leq NT$. Note that from (7), (73) and (81), we have also $\gamma \leq \max\limits_{l\in[L]} \omega(\mathbf{E}(:,l))/N$, which implies that the number of non-zero elements of each column over the number of servers upper bounds γ. Also note that the total number of possible different columns of $\mathbf{F} \in \mathbb{F}^{K\times L}$ remains the same *i.e.* $L \leq q^K$.

Now consider $\mathcal{C}_\mathbf{D}$ to be a ρ-covering code then the same relationship where $\mathbf{F}$ consists of all possible different vectors in $\mathbb{F}^K$ *i.e.* $L = q^K$ remains except that here $\gamma \leq \rho T$, since the normalized covering radius here is normalized on NT, while the cost follows $\gamma \leq \max_{l\in[L]} \omega(\mathbf{E}(:,l))/N,$. In other words here ρT is an upper-bound on the computation cost while in the previous section the similar relationship held with equality.

Therefore we can see that Theorem 1 where acts as a bridge to connect our system model and the results in the coding theory in the previous section, results to a guarantee $\gamma \leq \rho T$, in particular it just guarantees an upper bound on the cost. Note that the communication cost follows (82) relation.

C. Achievable Results for The Multi-Shot Setting

Considering the difference mentioned in the previous Subsection, with the same approach for the achievability results of Theorems 2, we have the following result

Theorem 4. *For the distributed linearly separable problem with K users, N servers and any number of L sub-functions the optimal computation cost γ is bounded by*

$$\gamma \leq TH_q^{-1}(\frac{K}{NT}). \tag{83}$$

Proof. □

D. Discussion and Comparison

Such like the Discussion and Comparison results of Section IV, we summarized our achievable scheme results of Theorem 4 in this subsection compared to the previous section's results in terms of computation cost. To analyse the results, we have the following observations:

1) Asymptotic analysis of the computation cost per server with respect to number of shots: It is interesting to see that when T approaches infinity the computation cost per server approaches to zero, in particular

Lemma 1. We claim that,

$$\lim_{T\to\infty} TH_q^{-1}(c/T) = 0, \tag{84}$$

where c is fixed.

Proof. The proof can be found in appendix N. □

Note that from Theorem 4, we see the computation cost per server is bounded by $LTH_q^{-1}(\frac{K}{NT})$. Concluding from Lemma 1, we see that as the number of shots or T grows to infinity while $K \leq N$ and K/N is fixed the normalized computation cost approaches to zero. This observation shows us that comparing the computation cost of the one-shot scheme with the same number of users K and servers

N, if T is large enough then the normalized computation cost can be arbitrarily be close to zero. This phenomena shows the advantage of using multiple shots with respect to the normalized computation cost. To analyze the non-asymptotic effect of an increase in T, the number of shots we begin with the following lemma,

Lemma 2. Let $f = TH_q^{-1}(c/T), 0 \leq c/T \leq 1-1/q$, then the derivative of f with respect to T satisfies,

$$\frac{\partial f}{\partial T} = \frac{H_q(f/T)}{\log_q\left(\frac{f/T}{1-f/T}(q-1)\right)} + f/T, \tag{85}$$

Proof. Appendix O contains the proof. □

From Lemma 2 and observing that $\frac{\partial f}{\partial T} \leq 0$ where $0 \leq H_q^{-1}(K/NT) = f/T \leq 1/q$, we conclude that since $0 \leq H_q^{-1}(K/NT) = f/T \leq 1/2 = 1-1/q$ when $q = 2$, increasing the number of shots while K, N are large enough would results a strict monotonic decrease of the normalized computation cost as described in Theorem 4, thus the bound on the computation cost per server decreases monotonically since $L\gamma$ is monotonically decreasing. On the other hand when $q > 2$, we see that if $T \geq T_0$, where $H_q(K/NT_0) = 1/q$, the same monotonic decrease in the number of shots will happen.

VI. Conclusion

In this paper, we introduced a new multi-user distributed computation system model based on [49], [63] which is an extension to many distributed computing applications especially in distributed machine learning. We established a novel coding theoretic view on the feasibility condition of the system model and build a bridge between covering codes and sparse matrix factorization problem in the finite fields.

Establishing the relationship between our problem and covering codes necessitates us to generalize definition of covering codes which result to introduction of new types of codes called partial covering codes. We also showed the interesting connection with this new class of codes which has led to the modification of well-studied results on the achievability and converse of sphere-covering theorems. We also built our conditional optimal achievable result for the one-shot setting.

discussion about the metric

As we studied the multi-shot setting, we had understood that when the number of shots T goes to infinity while the number of users K and servers N are fixed and NT is large enough, the normalized computation cost approaches zero.

This paper also provides a novel look into sparse matrix factorization problem in finite fields and the new introduced partial covering codes which may interest other researchers. Also analysis the problem, a similar connection to the transposed version of (27), $\mathbf{E}^\intercal\mathbf{D}^\intercal = \mathbf{F}^\intercal$, would be an interesting topic for future researches.

In the introduction we have mentioned some applications similar to applications mentioned in [49] for the linearly separable problem. To further motivate our problem we can mention a hierarchical or tree-like scenario introduced in [42], [64] which was intended to solve bandwidth limitation and stragglers' delay simultaneously in the gradient coding setting [26]. In the hierarchical setting used for the distributed gradient coding problem there are some users[6] connected each to a group of servers where instead of conventional topology the aggregation of the sub-gradients executes hierarchically. In particular, each user computes a linearly separable function with its received messages and sends it to the Aggregator which it finally computes the gradient. Our system model can be regarded as an extension to the hierarchical system model in [42], [64], since the users can be connected arbitrarily to a any subset of servers and also the servers can send messages not only in one shot but in multiple shots through a broadcast parallel channel.

To compare or results to the existing results, we know that in [49], the setting where there is only one user requesting multiple linearly separable function has been studied. On that setting, the cyclic assignment of datasets to the servers has been utilized to mitigate the effect of stragglers. The cyclic assignment makes that scheme to be independent of the task functions. Because of that reason in [49] the coefficients of the functions are assumed to be distributed uniformly and *i.i.d* and the decodability of the received data is probabilistic, while in our system model the master node is totally aware of the requested functions.

It is necessary to mention that [63] and [65] are an extension to [49], where in [63], communication-computation trade-off in terms of straggler nodes has been studied for the same task function blind setting and in [65] the secure version of the same setting in [49] has been investigated.

Also note that the definition of linear separability in (4) absorbs the cases where F_k itself is a linear combination of some linearly separable functions in particular it can be also be formed as,

$$F_k = \sum_{i=1}^{M} F_k^{(i)}, \tag{86}$$

which $F_k^{(i)}$ are linearly separable functions.

[6] In [42] are named the master nodes

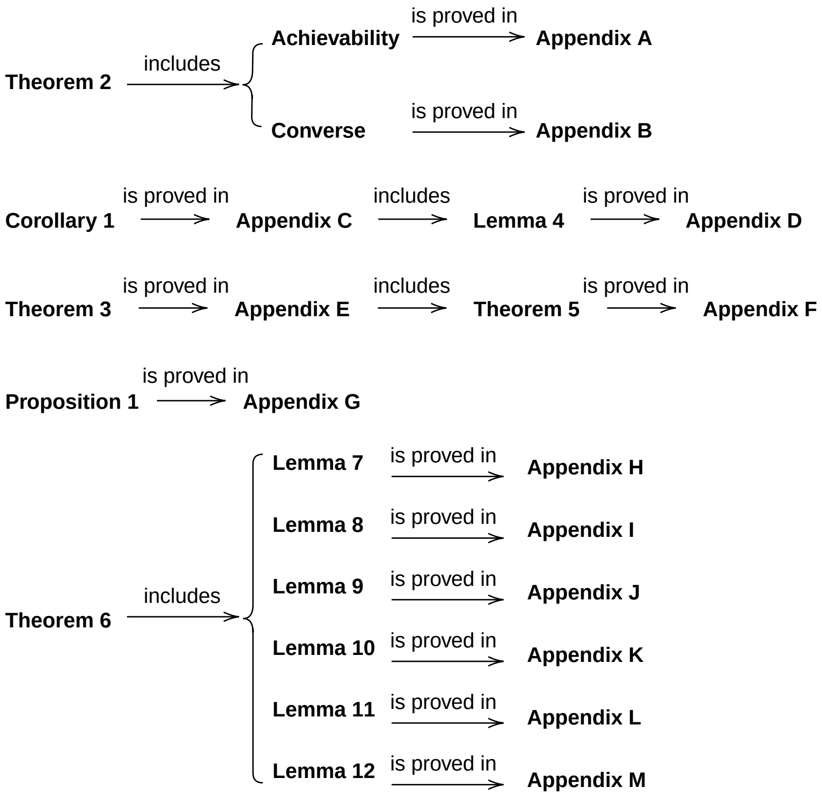

Fig. 5. Map of the Appendices and Theorems.

Appendix A
Proof of The Converse of Theorem 2

Proof. For the lower bound of (54), we modify the sphere-covering bound for the partial covering codes in the following Lemma,

Lemma 3. For a set $\mathcal{X}$ that satisfies $\mathcal{X} \subseteq \mathbb{F}_q^n,\ |\mathcal{X}| = q^k L,\ k \in \mathbb{N}$, an existing $(\rho, \mathcal{X})$-partial covering code $\mathcal{C}(k,n)$ has to satisfy

$$\log_q(L) \leq \log_q(V_q(n,\rho)). \tag{87}$$

Proof. **(Proof of Lemma 3)** Since the number of codewords is q^k, the maximum number of points they can ρn-cover is $q^k V_q(n,\rho)$, therefore we have,

$$Lq^k \leq V_q(n,\rho) q^k, \tag{88}$$

taking $\log_q$ from both sides (87) results. □

Now consider $\mathcal{X}$ in Theorem 1 to be the case for Lemma 3, note that if $|\mathcal{X}| = Lq^k$ then $\mathcal{X} = \mathcal{X}_F$. Then by substituting $N = n, K = n-k$, we see that $\log_q(L) \leq \log_q(V_q(N,\rho))$. By using the estimate $q^{NH_q(\rho)-o(N)} \leq V_q(N,\rho) \leq q^{NH_q(\rho)}$, we can conclude that $\log_q(L) \leq NH_q(\rho)$ and the claim $H_q^{-1}(\frac{\log_q(L)}{N}) \leq \rho$ results. □

Appendix B
Proof of The Achievability of Theorem 2

Referring to [62], there exist at least a $\mathcal{C}_{\mathcal{X}}(k,n)$, ρ-covering code satisfying

$$n - k \geq \log_q(V_q(n,\rho)) - 2\log_2(n) + \log_q(n) - O(1). \tag{89}$$

Applying Theorem 1 we see that, $\mathbf{D} = \mathbf{H}_{\mathcal{C}}, N = n, K = n-k$ and $\mathcal{X} = \mathbb{F}^n$, therefore there has to exist a feasible scheme for the multi-user linearly separable problem with computational cost ρ that satisfies,

$$K/N \geq \log_q(V_q(N,\rho))/N - 2\log_2(N)/N + \log_q(N)/N - O(1)/N. \tag{90}$$

Combining with the estimate $q^{NH_q(\rho)-o(N)} \leq V_q(N,\rho) \leq q^{NH_q(\rho)}$, we conclude that

$$K/N \geq H_q(\rho) - \epsilon(N), \tag{91}$$

where $\epsilon(N)$ is a term that approaches to zero as N increases. In other formulation it is simply can be derived that $\rho \leq H_q^{-1}(K/N + \epsilon(N))$, where for large enough N the claim results.

Appendix C
Proof of Corollary 1

Here we first prove the existence of some partial covering linear code for any $\mathcal{B}_q(0,\rho) \subseteq \mathcal{X} \subseteq \mathbb{F}^n$, in particular we prove the existence of $(\rho, \mathcal{X})$-partial covering linear code $\mathcal{C}$ where $\mathcal{X} \subseteq \mathbb{F}^n$ has two condition, first $\mathcal{B}_q(0,\rho) \subseteq \mathcal{X}$ and secondly $|\mathcal{X}| = q^k L$. This has been done in the following Lemma, via a linear greedy algorithm.[7]

[7] If in some sub-optimal case, there exist a row in $\mathbf{E}$ where all of its entries is zero, it will correspond to the case where there will be a server without any workload. In this case the mater node slightly modifies the scheme. It will replicate an arbitrary server pattern without containing $l = \text{argmax}\,\omega(\mathbf{E}(:,\mathbf{l}))$, and sends exactly the same linear combinations of that server except the ones containing $l = \text{argmax}\,\omega(\mathbf{E}(:,\mathbf{l}))$. On this case the initial server would stop sending the same linear combinations to the users that it had already sends except the ones that contains $l = \text{argmax}\,\omega(\mathbf{E}(:,\mathbf{l}))$. In other words it share the computation and communication workload with the already idle server. This modified achievable result dose better in terms of the normalized computation cost defined in (7) then the proposed scheme since while the nominator is the same and then denominator has been increases.

Lemma 4. Consider a (k, n) code $\mathcal{C}$ that is a $(\rho, \mathcal{X})$-partial covering code for a set $\mathcal{X} \subseteq \mathbb{F}_q^n$ that includes $\mathcal{B}_q(0, \rho) \subseteq \mathcal{X}$ and which has size $|\mathcal{X}| = L'q^k$. Such code exists for some L' that satisfies

$$\log_q(L') \geq \log_q(V_q(n, \rho)) - 2\log_2(n) + \log_q(n) - O(1), ^{8} \tag{92}$$

Proof. The proof is included in Appendix D. □

Now let's define $\mathcal{A}_m \triangleq \{|\mathcal{X}| = m | \mathcal{B}_q(0, \rho) \subseteq \mathcal{X} \subseteq \mathbb{F}^n\}$ to be the family of all subsets of $\mathbb{F}^n$ with cardinality m. Now the proposed scheme follows the following algorithm,

1) Assign $m = Lq^{N-K}$.
2) For each $\mathcal{X}$ in $\mathcal{A}_m$ find a $(\rho, \mathcal{X})$- partial covering code $\mathcal{C}_\mathcal{X}$ via Lemma 4.
3) For each $\mathcal{X}$ in $\mathcal{A}_m$, choose $\mathbf{D} = \mathbf{H}_{\mathcal{C}_\mathcal{X}}$ and $\mathcal{X}_{\mathbf{F},\mathbf{D}} = \{\mathbf{x} \in \mathbb{F}^N | \mathbf{D}\mathbf{x} = \mathbf{F}(:, l), \text{ for some } l \in [L]\}$.
4) For each $\mathcal{X}$ in $\mathcal{A}_m$, if $\mathcal{X} \subseteq \mathcal{X}_{\mathbf{F},\mathbf{D}}$, output D and $\mathcal{X}$.
5) If $\mathcal{X} \nsubseteq \mathcal{X}_{\mathbf{F},\mathbf{D}}$, then increase m by one and start again from step 2.

Suppose that the scheme terminates and outputs $\mathcal{D}$ and $\mathcal{X}$ at the fourth step while $m \neq q^N$. By Lemma 4 it has been guaranteed that $\mathcal{C}_\mathcal{X}$ satisfies,

$$\log_q(|\mathcal{X}|q^{-k}) \geq \log_q(V_q(n, \rho)) - 2\log_2(n) + \log_q(n) - O(1), \tag{93}$$

where combining by Theorem 1 while $N = n, K = n - k$ and $\mathcal{X} \supseteq \mathcal{X}_{\mathbf{F},\mathbf{D}}$ is ρn-covered by $\mathcal{C}_\mathcal{D} = \mathcal{C}_\mathcal{X}$, we can conclude that there has to exist a multi-user linearly separable feasible scheme with ρ computation cost satisfying,

$$\frac{\log_q(|\mathcal{X}|) - (N-K)}{N} \geq \log_q(V_q(N, \rho))/N - 2\log_2(N)/N + \log_q(N)/N - O(1)/N, \tag{94}$$

combining with the estimate $q^{NH_q(\rho) - o(N)} \leq V_q(N, \rho) \leq q^{NH_q(\rho)}$, we conclude that

$$(\frac{K}{N} - 1 + \frac{\log_q(|\mathcal{X}|)}{N}) \geq H_q(\rho) - \epsilon(N), \tag{95}$$

where $\epsilon(N)$ is a term that approaches to zero as N increases. In other formulation it is simply can be derived that the feasible scheme has the computation cost

$$\rho \leq H_q^{-1}(K/N - 1 + \log_q(|\mathcal{X}|/N) + \epsilon(N)), \tag{96}$$

where for large enough N the claim results. Note that when $\mathcal{X} = \mathcal{X}_{\mathbf{F},\mathbf{D}}$ be the output then $|\mathcal{X}| = Lq^{N-K}$ and by substituting in (96), we see that the computation cost $\rho = H_q^{-1}(\log_q(L)/N + \epsilon(N))$. We see that the algorithm finally terminates since at last $m = q^N > Lq^{N-K}$ and $\mathcal{X} = \mathbb{F}^N$. Note that on that case the chosen code $\mathcal{C}_\mathcal{X}(k, n)$ is a normal ρ-covering code used in in Appendix B, for the proof of Theorem 2.

Appendix D
Proof of Lemma 4

Proof. We here start by employing the recursive construction approach of Cohen and Frankl in [62]. This recursive approach builds an $(n, j+1)$ code $\mathcal{C}_{j+1}$ from a previous (n, j) code $\mathcal{C}_j$, by carefully adding a vector $\mathbf{x}$ in the basis of $\mathcal{C}_j$. Our aim will be to recursively construct ever bigger codes that cover an ever increasing portion of our set $\mathcal{X}$.

Let us start by setting $\mathcal{C}_0 = \{\mathbf{0}\}$, and recall that

$$L = q^{n-k'}, \ k' > k. \tag{97}$$

Let $Q(\mathcal{C})$ denote the set of points in $\mathcal{X}$ that are not ρn-covered by $\mathcal{C}$, and let

$$q(\mathcal{C}) \triangleq \frac{|Q(\mathcal{C})|}{q^{n+k-k'}} \tag{98}$$

where naturally

$$|Q(\mathcal{C}_0)| = q^{n+k-k'} - V_q(n, \rho) \tag{99}$$

and

$$q(\mathcal{C}_0) = 1 - V_q(n, \rho) q^{-(n+k-k')}. \tag{100}$$

We also need the following lemma from [62].

Lemma 5 ([62])**.** Let $\mathcal{Y} \subseteq \mathbb{F}^n, \mathcal{Z} \subset \mathbb{F}^n$, and consider $\mathcal{Y} + \mathbf{x} = \{\mathbf{y} + \mathbf{x} : \mathbf{y} \in \mathcal{Y}\}$ for some $\mathbf{x} \in \mathbb{F}$. Then

$$\mathbb{E}(|(\mathcal{Y} + \mathbf{x}) \cap \mathcal{Z}|) = q^{-n} |\mathcal{Y}||\mathcal{Z}| \tag{101}$$

where the average is taken, with uniform probability, over all $\mathbf{x} \in \mathbb{F}^n$.

Now, we develop the proof in two sections:

1) **Binary Case:** The proof for $q = 2$ where $k = k'$ (corresponding only to the singular case of maximal $L = 2^K$) has been presented in [66] and [62] in two different ways. We will modify the latter approach to establish our claim for any $k' \geq k$ (which will allow us to handle any possible L). First let us easily deduce from Lemma 5 that there exists an $\mathbf{x} \in \mathbb{F}^n$ for which $|(\mathcal{Y} + \mathbf{x}) \cap \mathcal{Z}| \leq \frac{|\mathcal{Y}||\mathcal{Z}|}{q^n}$. Now let us set $\mathcal{Y} = \mathcal{Z} = Q(\mathcal{C}_j)$, and let us append a vector $\mathbf{x}$ to the generator matrix of $\mathcal{C}_j$ to create $\mathcal{C}_{j+1}$, where $\mathbf{x}$ is chosen to minimize $|\mathcal{Q}(\mathcal{C}_{j+1})|$. Now we can directly verify that

$$|\mathcal{Q}(\mathcal{C}_{j+1})| = |\mathcal{Q}(\mathcal{C}_j) \cap \mathcal{Q}(\mathcal{C}_j + \mathbf{x})| = |\mathcal{Q}(\mathcal{C}_j) \cap (\mathcal{Q}(\mathcal{C}_j) + \mathbf{x})| \leq |\mathcal{Q}(\mathcal{C}_j)|^2 / 2^n \tag{102}$$

which implies that

$$q(\mathcal{C}_{j+1}) \leq q(\mathcal{C}_j)^2 2^{k-k'} \leq q(\mathcal{C}_j)^2, \tag{103}$$

where the latter inequality holds because $k' \geq k$. Combining (100) and (103), gives

$$q(\mathcal{C}_k) \leq q(\mathcal{C}_0)^{2^k} \leq (1 - V_q(n,\rho)2^{-(n-k'+k)})^{2^k}, \tag{104}$$

where the latter inequality again holds due to the fact that $k' \geq k$. Now let us continue this recursion until k is such that

$$2^k = \lceil (n-k'+k)2^{(n-k'+k)} \ln(2)/V_2(n,\rho) \rceil, \tag{105}$$

in which case — given that $(1-\frac{1}{x})^x \leq e^{-1}$, $\forall x \geq 1$ — we get that

$$q(\mathcal{C}_k) < 2^{-(n+k-k')} \tag{106}$$

which automatically yields that $Q(\mathcal{C}_k) = 0$. This, again with the choice of k in (105), tells us that for a set $\mathcal{X}$ that satisfies $\mathcal{B}_q(0,\rho) \subseteq \mathcal{X} \subseteq \mathbb{F}_q^n$, $|\mathcal{X}| = Lq^k$, then indeed there exist a $(\rho, \mathcal{X})$-partial covering code $\mathcal{C}(n,k)$ satisfying

$$0 \leq \log_q(L/V_q(n,\rho)) + 2\log_2(\log_q(|\mathcal{X}|)) - \log_q(\log_q(|\mathcal{X}|)) + O(1). \tag{107}$$

This can be considered as a tighter version of our Lemma 4. After a few very basic algebraic manipulations, and after setting $n = k$, we get the proof of Lemma 4 — for the binary case of $q = 2$ — in its current form.

2) **Non-Binary Case:** Considering first an arbitrary $\mathcal{Z} \subset \mathbb{F}^n$, we have that

$$\mathbb{E}(1 - (q^{-n+k'-k}|(\mathcal{Z}+\mathbf{x}) \cup \mathcal{Z}|)) = \mathbb{E}(1 - q^{-n+k'-k}((|(\mathcal{Z}+\mathbf{x})| + |\mathcal{Z}|) - |(\mathcal{Z}+\mathbf{x}) \cap \mathcal{Z}|)) \tag{108}$$

$$= \mathbb{E}(1 - 2q^{-n+k'-k}|\mathcal{Z}| + q^{-n+k'-k}|(\mathcal{Z}+\mathbf{x}) \cap \mathcal{Z}|) \tag{109}$$

$$\overset{(a)}{=} 1 - 2q^{-n+k'-k}|\mathcal{Z}| + q^{-2n+k'-k}|\mathcal{Z}|^2 \tag{110}$$

$$\overset{(b)}{\leq} 1 - 2q^{-(n-k'+k)}|\mathcal{Z}| + q^{-2(n-k'+k)}|\mathcal{Z}|^2 \tag{111}$$

$$= (1 - \frac{|\mathcal{Z}|}{q^{(n-k'+k)}})^2, \tag{112}$$

where (a) is directly from Lemma 5, and where (b) holds since $k' \geq k$. Similarly to the binary case, we begin with $\mathcal{C}_0 = \{\mathbf{0}\}$, and again recursively extend as

$$\mathcal{C}_j =< \mathcal{C}_{j-1}; \mathbf{x} >, \tag{113}$$

where $\mathbf{x}$ is chosen so that $|\mathcal{Z}|$ is maximized. We do so, after again having set $\mathcal{Z} = Q(\mathcal{C}_j)$.

At this point, from (112) we have that

$$q(\mathcal{C}_{j+1}) \leq q(\mathcal{C}_j)^2. \tag{114}$$

We now consider the following lemma from [62, Lemma 2].

Lemma 6. ([62, Lemma 2]) For $\mathcal{Z} \subseteq \mathcal{X}$, where $|\mathcal{Z}|q^{-(n-k'+k)} = \epsilon < (q(n-k'+k))^{-1}$, then

$$\mathbb{E}(1 - q^{-(n-k'+k)}| \cup_{\alpha \in \mathbb{F}_q} \mathcal{Z} + \alpha \mathbf{x}|) \leq (1-\epsilon)^{q(1-(2(n-k'+k))^{-1})}. \tag{115}$$

Continuing from $\mathcal{Z} = \mathcal{X} \cap (\cup_{\mathbf{c} \in \mathcal{C}_{j-1}} \mathcal{B}_q(\mathbf{c}, \rho))$, where

$$|\mathcal{Z}| < \frac{1}{n} q^{(n-k'+k-1)}, \qquad q(\mathcal{C}_{j+1}) \leq q(\mathcal{C}_j)^{q(1-(2(n-k'+k)^{-1}))}$$

we have that

$$q(\mathcal{C}_{j+1}) \leq (1 - q^{n-k'+k} V_q(n,\rho))^{(q(1-(2(n-k'+k))^{-1}))^j} \leq (1 - q^{n+k-k'} V_q(n,\rho))^{e^{-0.5} q^j}, \tag{116}$$

since $(1-(2(n-k'+k))^{-1}) \geq (1-(2(n-k'+k))^{-1})^{n-k'+k-1} \geq e^{-0.5}$. For

$$j_1 \triangleq \arg\min_j \{q(\mathcal{C}_j) \leq 1 - (q(n+k-k'))^{-1}\} \tag{117}$$

we see that

$$j_1 \leq n - \log_q(q^{k'-k} V_q(n,\rho)) - \log_q(n+k-k') + O(1) \tag{118}$$

where the inequality holds by first observing that Lemma 6 yields

$$1 - (q(n-k'+k))^{-1} \leq q(\mathcal{C}_{j-1}) \leq (1 - q^{(n-k'+k)V_q(n,\rho)})^{q^{j_1-1} e^{-1/2}}, \tag{119}$$

and then by comparing the upper and lower bounds in (119).

We have a (n, j_1) code $\mathcal{C}$ and (114). We are now looking for the minimum number j_2 of generators $\mathbf{x}$ that has to be appended to (the generator of) $\mathcal{C}$ in order to get a (n, j_1+j_2) code with $q(\mathcal{C}_{j_1+j_2}) \leq q^{-(n-k'+k)}$. We note that $q(\mathcal{C}_{j_1}) \leq 1-(q(n-k'+k))^{-1}$, so by (119) we only need to ensure that $(1-(q(n-k'+k))^{-1})^{2^{j_2}} \leq q^{-(n-k'+k)}$ which can be achieved by using

$$j_2 = 2\log_2(n-k'+k) + O(1). \tag{120}$$

Hence for $k \leq j_1 + j_2$, there indeed exist (n,k) codes with normalized covering radius no bigger than ρ. Applying (118),(120),(97) and the fact that $|\mathcal{X}| = Lq^k$, proves (92) and the entire Lemma 4.

□

Appendix E
Proof of Theorem 3

Proof. For the converse about the normalized computation cost, here the same converse argument holds as of Theorem 2.

We start with Theorem 5 which is an extension to the famous Theorem of Blinovskii in [59], where he had proved that almost all linear codes satisfies sphere-covering bound. To articulate our theorem we first begin with Definition 2.

Definition 2. Let $\rho \in (0, 1 - \frac{1}{q}]$ and let $\tau \in (0, 1]$. A code $\mathcal{C} \subseteq \mathbb{F}^n$ is set to be a (ρ, τ)-partial covering code if there exists a set $\mathcal{X} \subseteq \mathbb{F}^n$ with $\frac{1}{n}\log_q(|\mathcal{X}|) = 1 - \tau$ that is ρ-covered by $\mathcal{C}$.

Theorem 5. *Let $\rho \in (0, 1 - \frac{1}{q}]$ and let $\tau \in (0, 1 - H_q(\rho)]$. Let $\mathcal{C}_{k,n}$ be the ensemble of all linear codes generated by all possible $k \times n$ matrices in $\mathbb{F}^{k\times n}$. Then there exist an infinite sequence k_n that satisfies*

$$\frac{k_n}{n} \leq 1 - \tau - H_q(\rho) + O(n^{-1}\log_q(n)) \tag{121}$$

such that the fraction of codes $\mathcal{C}_n \in \mathcal{C}_{k_n,n}$ that are (ρ, τ)-partial covering, tends to 1 as n grows to infinity. Thus in the limit of large n, almost all codes of rate less than $1 - \tau - H(\rho)$ will be (ρ, τ)-partial covering.

Proof. The proof can be found on the Appendix F. □

Now the proposed scheme follows the following steps,

1) Assign $m = L$.
2) Let $\tau = \frac{K - \log_q(m)}{N}$.
3) Now consider $g(n)$ to be the fraction of codes that are (ρ, τ)-partial covering in $\mathcal{C}_{k_n,n}$ for the claimed sequence k_n of Theorem 5. Now let's m_n represent the lower bound on the number of (ρ, τ)- partial covering in the ensemble $\mathcal{C}_{k_n,n}$ as,

$$m_n \triangleq g(n) q^{k_n n}, \tag{122}$$

and put all of them in to set $\mathcal{B} \triangleq \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{m_n}\}$. Now let

$$\mathbf{D}_n \triangleq \begin{bmatrix} \mathbf{H}_{\mathcal{C}_1} & & & \\ & \mathbf{H}_{\mathcal{C}_2} & & \\ & & \ddots & \\ & & & \mathbf{H}_{\mathcal{C}_{m_n}} \end{bmatrix}, \tag{123}$$

and let $K = m_n(n - k_n)$ and $N = m_n n$. We observe that $\mathcal{C}_{\mathcal{W}_n} = [\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{m_n}]$ and let

$$\mathcal{X}_{\mathbf{F},\mathbf{D}} \triangleq \{\mathbf{x} \in \mathbb{F}^N | \mathbf{D}\mathbf{x} = \mathbf{F}(:, l), for\ some\ l \in [L]\}. \tag{124}$$

Also consider a set defined as

$$\mathcal{X} \triangleq \{\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{m_n}] | \mathbf{x}_i \in \mathcal{X}_i\} \tag{125}$$

where $\mathcal{X}_i, i \in [m_n]$ is the set of all the points that are ρn-covered by $\mathcal{C}_i$. Note that

$$|\mathcal{X}_i| \geq q^{n(1-\tau)}, \forall i \in [m_n], \tag{126}$$

because of Definition 2. Since

$$\forall \mathbf{x} \in \mathcal{X} : \ \mathbf{d}(\mathbf{x}, \mathcal{C})/N = \sum_{i=1}^{m_n} \mathbf{d}(\mathbf{x}_i, \mathcal{C}_i)/N \leq \sum_{i=1}^{m_n} \frac{\rho n}{m_n n} = \sum_{i=1}^{m_n} \rho \frac{1}{m_n} = \rho. \tag{127}$$

$\mathcal{C}_{\mathbf{D}_n}$ is also a $(\rho, \mathcal{X})$-partial covering code. Now if $\mathcal{X} \not\supseteq \mathcal{X}_{\mathbf{F},\mathbf{D}}$, then m has to be increased by one and the procedure has to be started from step 1.

4) Making sure $\mathcal{X} \supseteq \mathcal{X}_{\mathbf{F},\mathbf{D}}$, Let's define $k'_n \triangleq n - k_n$. From (121) we know that

$$\frac{k'_n}{n} \geq \tau + H_q(\rho) - O(n^{-1} \log_q(n)). \tag{128}$$

Now let $R \triangleq \frac{K}{N}$, the dimensions of $\mathbf{D}$, we see that also $R = \frac{k'_n}{n}$ since $K = k'_n m_n, N = n m_n$. Thus the rate of the resulted feasible scheme can be written as

$$K/N = R = H_q(\rho) + \tau - \epsilon(N). \tag{129}$$

Note that as n goes to infinity N also approaches to infinity, while the term $O(n^{-1} \log_q(n))$ can be made arbitrarily small.

In other words it can be inferred that

$$\rho = H_q^{-1}(\frac{\log_q(m)}{N} + \epsilon(N)). \tag{130}$$

Also we have,

$$\frac{\omega(\mathbf{D}_n)}{K} \overset{(a)}{\leq} \frac{m_n n k'_n}{m_n k'_n} = n, \tag{131}$$

Equation (a) holds since $\omega(\mathbf{D}_n) = m_n k_n n$ is the maximum number of nonzero elements that $\mathbf{D}$ can have.

Note that, taking logarithm from both side of $N = m_n n$, regarding (122), $k_n = (1 - R)n$ we have

$$\log_q(n) + n^2(1 - R) + \log_q(g(n)) = \log_q(N), \tag{132}$$

therefore we have $n^2(1-R) \le \log_q(N)$ and $n \le \sqrt{\frac{\log_q(N)}{(1-R)}}$, combining with (131) and Theorem 1, we have that

$$\Delta \le \sqrt{\frac{\log_q(N)}{(1-R)}}, \tag{133}$$

where as mentioned before R is constant.

We claim that the above scheme terminates, since if m get increased until $m = q^K$ then $\tau = 0$. We see that $\mathcal{X}_i = \mathbb{F}^n$ since $|\mathcal{X}_i| \ge q^n$ by applying Definition 2 and Theorem 5. Therefore by (125), $\mathcal{X} = \mathbb{F}^N = \mathbb{F}^{m_n n}$, which means that $\mathcal{C}_{\mathcal{D}_n}(N, N-K)$ is a ρ-covering code and $\mathcal{X} \supseteq \mathcal{X}_{\mathbf{F},\mathbf{D}}$, finally the scheme would terminates at Step 4 with computation cost $\rho = H_q^{-1}(\frac{K}{N} + \epsilon(N))$ from (130) and the communication cost as (133).

□

Appendix F

Definition 3. Let $\rho \in (0, 1-\frac{1}{q}]$. We say that a set $\mathcal{X} \subseteq \mathbb{F}^n$ is ρ-covered by a code $\mathcal{C} \subseteq \mathbb{F}^n$ iff

$$d(\mathbf{x}, \mathcal{C}) \le \rho n, \ \ \forall \mathbf{x} \in \mathcal{X}. \tag{134}$$

In such case, we say that $\mathcal{C}$ is a $(\rho, \mathcal{X})$-partial covering code.

Definition 4. Let $\rho \in (0, \min\{1-\frac{1}{q}, \frac{\sqrt{5}-1}{2}\}]$ and let $\tau \in (0,1]$. A code $\mathcal{C} \subseteq \mathbb{F}^n$ is set to be a (ρ,τ)-partial covering code if there exists a set $\mathcal{X} \subseteq \mathbb{F}^n$ with $\frac{1}{n}\log_q(|\mathcal{X}|) = 1-\tau$ that is ρ-covered by $\mathcal{C}$.

Theorem 6. *Let $\rho \in (0, \min\{1-\frac{1}{q}, \frac{\sqrt{5}-1}{2}\}]$ and let $\tau \in (0, 1-H(\rho)]$. Let $\mathcal{C}_{k,n}$ be the ensemble of all linear codes generated by all possible $k \times n$ matrices from $\mathbb{F}^{k\times n}$. Then there exist an infinite sequence k_n that satisfies*

$$\frac{k_n}{n} \le 1-\tau-H(\rho)+O(n^{-1}, \log_q(n)), \tag{135}$$

such that the fraction of codes $\mathcal{C}_n \in \mathcal{C}_{k_n,n}$ that are (ρ,τ)-partial covering, tends to 1 as n grows to infinity. Thus in the limit of large n, almost all codes of rate less than $1-\tau-H(\rho)$ will be (ρ,τ)-partial covering.

Informal Proof:

First, we prove that with a consistent enumeration of codewords, each nonzero point in $\mathbb{F}^n$ has the same chance to be a codeword of a certain index.

Second, we introduce a random subset named Covered Set $\mathcal{X}_{\mathcal{C}}$ of size $2^{n(1-\tau)}$. Based on a determined $\mathcal{C} \in \mathcal{C}_{k^*,n}$, we pick $\mathcal{X}_C$ such that all the codewords of $\mathcal{C}$ to be inside it. We see that every point in $\mathbb{F}^n \backslash \mathcal{B}(\mathbf{0}, \rho)$ has the same chance to be in that subset.

Since we are interested to study the ρ-coverage of points inside $\mathcal{X}_{\mathcal{C}}$, by utilizing a conditional probability, we are able to explicitly derive $\mathbb{P}(\mathbf{c}_i = \mathbf{x}|\mathbf{x} \in \mathcal{X}_{\mathcal{C}})$.

Third, Because $\mathcal{B}(\mathbf{0}, \rho)$ is covered and also is a subset of covered set we focus our effort on the coverage of $\mathcal{X}_{\mathcal{C}} \backslash \mathcal{B}(\mathbf{0}, \rho)$.

Fourth, We prove that if the size of the codes in the ensemble, be properly chosen the conditional average on the number of codewords that covers each point in $\mathbf{x} \in \mathcal{X}_{\mathcal{C}}$ where $\mathcal{X}_{\mathcal{C}}$ is assumed to be determined is above $n^{\alpha}, \alpha > 1$, therefore almost all codes are almost (ρ, τ)-partial covering.

Finally, utilizing a linear greedy algorithm and successive appending of $\lfloor \log_q n(1-\tau) \rfloor$ to these almost (ρ, τ)-partial covering codes, we convert them into complete (ρ, τ)-partial covering codes without any essential decrease of their proportion.

Formal Proof:

Denote $k^* \triangleq k - \lceil \log_q (n(1-\tau)) \rceil$. Consider $\mathcal{C}_{k^*,n}$ to be the ensemble of linear codes defined by $k^* \times n$ generator matrices with elements chosen randomly and independently with probability $\frac{1}{q}$ from $\mathbb{F}_q$. Any non-void linear combination of rows of the generator matrix gives all possible q^n vectors. The zero codeword corresponds to the void linear combination of the rows and is present in all codes. Assume some consistent enumeration of the codewords in these codes *i.e.* the word with the same index are given by the same linear combination of vectors from the generator matrix. By convention the first codeword in all codes is the zero word.

Lemma 7. We claim that,

$$\mathbb{P}(\mathbf{c}_i = \mathbf{x}) = q^{-n}, \tag{136}$$

where $i \neq 1, \mathbf{x} \neq 0$ and the probability is over all $\mathcal{C} \in \mathcal{C}_{k^*,n}$.

Proof. The proof is presented in Appendix H. □

Consider that n is large enough so that $q^{n(1-\tau)} \geq V_q(n, \rho) + q^{k^*}$ holds. Let the Covered Set $\mathcal{X}_{\mathcal{C}}$ be chosen such that

$$|\mathcal{X}_{\mathcal{C}}| = q^{n(1-\tau)}, \tag{137}$$

$$\mathcal{C} \cup \mathcal{B}(\mathbf{0}, \rho) \subseteq \mathcal{X}_C \subseteq \mathbb{F}^n, \tag{138}$$

holds where $\mathcal{C} \in \mathcal{C}_{k^*,n}$ is a random linear code. In fact $\mathcal{X}_{\mathcal{C}}$ elements first is chosen from $\mathcal{C}$ and $\mathcal{B}(\mathbf{0}, \rho)$ elements then each element of $\mathcal{X}_{\mathcal{C}} \backslash \mathcal{C} \cup \mathcal{B}(0, \rho)$ is chosen uniformly, independently at random from $\mathbb{F}^n \backslash \mathcal{C} \cup \mathcal{B}(0, \rho)$ note that $\mathcal{X}_{\mathcal{C}}$ is dependent to $\mathcal{C}$. Through Lemma 8 we get the probability of each point to be inside the covered set $\mathcal{X}_{\mathcal{C}}$.

Lemma 8. We claim that,

$$\mathbb{P}(\mathbf{x} \in \mathcal{X}_{\mathbf{C}}) = \begin{cases} 1 & \omega(\mathbf{x}) \leq \rho n \\ \frac{q^{n(1-\tau)} - V(\rho,n)}{q^n - V(\rho,n)} & \omega(\mathbf{x}) > \rho n \end{cases}, \tag{139}$$

Proof. The proof is presented in Appendix I. □

Now utilizing Lemma 8, for all $\mathbf{x} \in \mathcal{X}_{\mathcal{C}}$ we derive the probability that a codeword be a point where it is assumed that the point is in the subset $\mathcal{X}_{\mathcal{C}}$. In particular

Lemma 9. we claim that

$$\mathbb{P}(\mathbf{c}_i = \mathbf{x} | \mathbf{x} \in \mathcal{X}_{\mathcal{C}}) = \begin{cases} 1 & i = 1, \mathbf{x} = 0 \\ q^{-(n)} & i \in [2 : K^*], \mathbf{x} \neq 0, \omega(\mathbf{x}) \leq \rho n \\ q^{-n(1-\tau)} \frac{1 - q^{-n} V(\rho,n)}{1 - q^{-(n-n\tau)} V(\rho,n)} = q^{-(n-n\tau)} \delta(n) & i \in [2 : K^*] \, and \, \omega(\mathbf{x}) > \rho n, \end{cases} \tag{140}$$

where $\delta(n) > 1$ is a statement that goes to 1 as n approaches to infinity.

Proof. The proof has been presented in Appendix J. □

We know that any $\mathbf{x} \in \mathcal{B}(0, \rho)$ is covered and are in $\mathcal{X}_{\mathcal{C}}$, for all $\mathcal{C} \in \mathcal{C}_{k^*,n}$. Since $\mathbf{0}$ is a codeword present in all linear codes and $\mathcal{X}_{\mathcal{C}}$ always satisfies (139) from now on we focus on the ρ-coverage of $\mathcal{X}_{\mathcal{C}} \backslash \mathcal{B}(\mathbf{0}, \rho)$. In this direction, we define

$$\mathcal{X}'_{\mathcal{C}} \triangleq \mathcal{X} \backslash \mathcal{B}(\mathbf{0}, \rho). \tag{141}$$

For every $\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}$ define the random variable $\eta_{\mathbf{x},i}$ by $\eta_{\mathbf{x},i} = 1$ if i-th codeword ρn-covers $\mathbf{x}$ otherwise $\eta_{\mathbf{x},i} = 0$.

$$\eta_{\mathbf{x}} \triangleq \sum_{i=1}^{2^{k^*}} \eta_{\mathbf{x},i}. \tag{142}$$

We see that $\eta_{\mathbf{x}}$ is the number of codewords covers a $\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}$. Now by the following Lemma we derive an average for this random variable.

Lemma 10. By averaging over all $\mathcal{C} \in \mathcal{C}_{K^*,n}$ we have

$$\mathbb{E}(\eta_{\mathbf{x}} | \mathbf{x} \in \mathcal{X}'_{\mathcal{C}}) = |\{0\} \cap \mathcal{B}(\mathbf{x}, \rho)| \times 1 \tag{143}$$

$$+ (q^{k^*} - 1)[|(\mathcal{B}(0, \rho) \backslash \{0\}) \cap \mathcal{B}(\mathbf{x}, \rho)| q^{-n} \tag{144}$$

$$+ |\mathcal{B}(\mathbf{x}, \rho) \backslash \mathcal{B}(0, \rho)| q^{-n(1-\tau)} \delta(n)]. \tag{145}$$

Proof. The proof is in the Appendix K. □

Lemma 11. We claim that,

$$\frac{Var(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}})}{\mathbb{E}(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}})q^2} \leq 1. \tag{146}$$

Proof. The proof is presented in Appendix L. □

By Chebyshev's inequality, for all $\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}$, from Lemma 11 we know that,

$$\mathbb{P}(|\eta_{\mathbf{x}} - \mathbb{E}(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}})| > q^{\epsilon+1}\sqrt{\mathbb{E}(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}})}\Big|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}) < \frac{Var(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}})}{q^{2\epsilon+2}\mathbb{E}(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}})} \leq q^{-2\epsilon}. \tag{147}$$

To elaborate more on the meaning of (147), Suppose that $\eta_{\mathbf{x}} \leq \mathbb{E}(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}})$ then the argument inside the probability argument will be,

$$\mathbb{E}(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}) - \eta_{\mathbf{x}} > q^{\epsilon+1}\sqrt{\mathbb{E}(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}})}, \tag{148}$$

which is simply,

$$\mathbb{E}(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}) - q^{\epsilon+1}\sqrt{\mathbb{E}(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}})} > \eta_{\mathbf{x}}. \tag{149}$$

For the chebyshev inequality to be useful in the next steps, we have to make sure that,

$$\beta(\epsilon) \triangleq \mathbb{E}(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}) - q^{\epsilon+1}\sqrt{\mathbb{E}(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}})} > 0, \tag{150}$$

therefore we should have

$$\mathbb{E}(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}) > q^{2\epsilon+2}. \tag{151}$$

In this regard, by noting Lemma 10 we have to prove that,

$$|\{0\} \cap \mathcal{B}(\mathbf{x}, \rho)| \times 1 + (q^{k^*} - 1)[|(\mathcal{B}(0, \rho)\backslash\{0\}) \cap \mathcal{B}(\mathbf{x}, \rho)|q^{-n} \tag{152}$$

$$+ |\mathcal{B}(\mathbf{x}, \rho)\backslash\mathcal{B}(0, \rho)|q^{-n(1-\tau)}\delta(n)] > q^{2\epsilon+2}. \tag{153}$$

By choosing k^* and ϵ some reasonable parameters.

Also we have to make sure that for sufficiently large n there exist an $\alpha > 1$ such that,

$$\mathbb{E}(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}) - q^{\epsilon+1}\sqrt{\mathbb{E}(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}})} = n^{\alpha}. \tag{154}$$

This has to also happen by properly choosing k^* and ϵ.

Lemma 12. We claim that for every $\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}$,

$$|\mathcal{B}(\mathbf{x}, \rho)\backslash\mathcal{B}(0, \rho)| > q^{nH_q(\rho)-o(n)}. \tag{155}$$

where $0 < \rho \leq \min\{1 - 1/q, \frac{\sqrt{5}-1}{2}\}$.

Proof. The proof is in the Appendix M. □

Noting Lemma 10 and Lemma 12, we have that

$$E(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}) > (q^{k^*} - 1)q^{nH_q(\rho)-o(n)-n(1-\tau)}\delta(n). \tag{156}$$

Therefore as mentioned above to assure (150), (151) we have to choose k^*, ϵ such that

$$(q^{k^*} - 1)q^{nH(\rho)-o(n)-n(1-\tau)}\delta(n) \geq q^{2\epsilon+1}, \tag{157}$$

holds for a large enough n.

If all the steps above be true then we have for all $\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}$ that

$$\mathbb{P}(\eta_{\mathbf{x}} < n^{\alpha}\big|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}) < q^{-2\epsilon}. \tag{158}$$

Now we utilise the same technique used in [59], we call a point in $\mathcal{X}'_{\mathcal{C}}$ partial-remote whenever it is ρn-covered by fewer than $n^{\alpha}, \alpha > 1$ codewords.

Let Q_0 stand for the set of partial remote points in $\mathcal{X}'_{\mathcal{C}}$ and define its normalized value as

$$q_0 \triangleq \frac{|\mathcal{Q}_0|}{q^{n(1-\tau)} - V_q(n, \rho)}. \tag{159}$$

Noting (158) the following statement results

$$\sum_{\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}} \mathbb{P}(\eta_{\mathbf{x}} < n^{\alpha}\big|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}) \leq (q^{n(1-\tau)} - V_q(n, \rho))q^{-2\epsilon}, \tag{160}$$

where the summation is over all the elements of a determined $\mathcal{X}'_{\mathcal{C}}$. Now also we know that,

$$\sum_{\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}} \mathbb{P}(\eta_{\mathbf{x}} < n^{\alpha}\big|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}) \overset{(a)}{=} \sum_{\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}} \mathbb{E}[\mathbb{1}(\eta_{\mathbf{x}} < n^{\alpha}\big|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}})] \tag{161}$$

$$\overset{(b)}{=} \mathbb{E}[\sum_{\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}} \mathbb{1}(\eta_{\mathbf{x}} < n^{\alpha}\big|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}})] \tag{162}$$

$$\overset{(c)}{=} \mathbb{E}[|\mathcal{Q}_0|], \tag{163}$$

where (a) results from the definition of the probability and $\mathbb{E}$ is an average over all codes in $\mathcal{C}_{k^*,n}$ and (b) is simply result of commutative property of $\mathbb{E}$ and $\sum$, finally (c) is the result of the definition of $\mathcal{Q}_0$.

Now combining (159),(160) and (163) we have,

$$\mathbb{E}(q_0) \leq q^{-2\epsilon}, \tag{164}$$

So far we have derived that there exist an upper bound on the normalized average number of partial-remote points $\mathbb{E}(q_0)$ of partial remote points in $\mathcal{X}'_{\mathcal{C}}$ for codes in an existing $\mathcal{C}_{k^*,n}$

$$\mathbb{E}(q_{\mathbf{0}}) < q^{-2\epsilon}. \tag{165}$$

Using Markov's inequality, we estimate the deviation of the normalized number of remote points from the mean,

$$\mathbb{P}(q_0 > 2^{\epsilon}\mathbb{E}(q_0)) < 2^{-\epsilon} \tag{166}$$

Thus the inequality,

$$q_0 < 2^{-\epsilon}, \tag{167}$$

holds for a proportion greater than $1 - 2^{-\epsilon}$ of all codes,

Now we apply the procedure of successive appending cosets to an initial code $\mathcal{C}' \in \mathcal{C}_{k^*,n}$ satisfying (166).

An argument similar to the derivation of $q(\mathcal{C}_{i+1}) \leq q(\mathcal{C}_i)^2$, [9] shows that the average normalized q_1 over $\mathbf{x} \in \mathbb{F}^n$ of remote points for $\mathcal{C}' \cup < \mathcal{C}'; \mathbf{x} >$ where $\mathbf{x}$ (has been added optimally to cover $\mathcal{X}_{\mathcal{C}}$ for each code $\mathcal{C}$ satisfying (166)) satisfies the inequality

$$\mathbb{E}(q_1) \leq q_0^2. \tag{168}$$

From Markov's inequality, we get

$$\mathbb{P}(q_1 > q^{\lambda}\mathbb{E}(q_1)) < q^{-\lambda}, \tag{169}$$

thus the proportion of codes which satisfies

$$q_1 < q^{\lambda - 2\epsilon}, \tag{170}$$

is at least $1 - q^{-\lambda}$.

Applying the same procedure to all of the codes satisfying (170) we conclude that

$$q_1 < q^{\lambda - 2\epsilon} \tag{171}$$

for a proportion at least $(1 - q^{-\epsilon})(1 - q^{-\lambda})$ of codes in $\mathcal{C}_{k^*+1,n}$.

Continuing the procedure we get,

$$q_i < q^{2^i(\lambda - \epsilon) - \lambda}, \tag{172}$$

for a proportion at least $(1 - q^{-\epsilon})(1 - q^{-\lambda})^i$ of the code in $\mathcal{C}_{k^*+i,n}$, we stop at the step m such that

$$q_m < q^{-n(1-\tau)}. \tag{173}$$

choose $m = \lceil \log_2(n - \tau) \rceil$ to satisfy (173) it is sufficient to choose $\lambda = \epsilon - 1(173)$.

[9] Which had been used by Cohen in [60] and extended in (114) at the appendix D

Thus for a proportion of codes from $\mathcal{C}_{k,n}$ at least to

$$(1-q^{-\epsilon})(1-q^{-\epsilon+1})^{\lceil \log_2 n(1-\tau)\rceil}, \tag{174}$$

we have $q_m < q^{-n(1-\tau)}$ thus [10] every $\mathbf{x} \in \mathcal{X}$ is ρn-covered by at least, n^{α} codewords by choosing $\epsilon = 2\log_q \log_2(n(1-\tau))$ plugging the value in (174), assuming $\mathbb{E}(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}) \geq (n(1-\tau))^{\alpha}$, $\alpha > 1$ by (156) and noticing that $\beta(\epsilon) > 0$ guaranteeing an ρn covering, we understand that the claim results, in particular we see that $\beta(\epsilon)$ is positive and $\mathbb{E}(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}) \geq (n(1-\tau))^{\alpha}$ for $\alpha > 1$. To this happen we know that,

$$\mathbb{E}(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}) \geq 2^{k^*}2^{nH(\rho)-o(n)}2^{-n(1-\tau)}, \tag{175}$$

so we make the RHS of the above equation to be $(n-n\tau)^{\alpha}$, in other words we let n to be large enough so that we have,

$$\exists\ \alpha > 1 : (n-n\tau)^{\alpha} = 2^{k^*}2^{nH(\rho)-o(n)}2^{-(n-n\tau)}, \tag{176}$$

taking logarithm from the both sides and dividing by n, we have

$$\frac{k^*}{n} = 1-\tau-H(\rho)+\frac{\alpha\log_2(n-n\tau)+o(n)}{n}. \tag{177}$$

Now by substituting this into $\beta(\epsilon)$ argument we can also guarantee that $\beta(\epsilon)$ is positive.

After successive appending points to an initial code in $\mathcal{C}_{k^*,n}$, we have

$$\frac{k^*+\log_2(n-n\tau)}{n} = 1-\tau-H(\rho)+\frac{(\alpha+1)\log_2(n-n\tau)+o(n)}{n}. \tag{178}$$

Therefore since

$$\frac{(\alpha+1)\log_2(n-n\tau)+o(n)}{n} \in O(n^{-1}log_2(n-n\tau)), \tag{179}$$

the claim of the theorem results when we consider,

$$k_n = k^* + \log_2(n(1-\tau)).$$

Appendix G
Proof of Proposition 1

Referring to the Proof of Theorem 3 on Appendix E. Suppose $L \leq m < q^K$ and $\mathcal{X} \supseteq \mathcal{X}_{\mathbf{F},\mathbf{D}}$. From (125) we see that

$$|\mathcal{X}| \overset{(a)}{=} \Pi_{i=1}^{m_n}|\mathcal{X}_i| \tag{180}$$

[10] It makes $|\mathcal{Q}_0| = 0$.

$$\overset{(b)}{\geq} q^{nm_n(1-\tau)} \tag{181}$$

$$\overset{(c)}{=} q^{N(1-\tau)}, \tag{182}$$

where (a) comes from the definition of $\mathcal{X}$, equality (b) holds from (126) and (c) is true since $N = nm_n$. From (129) and (182), we conclude that

$$\rho = H_q^{-1}(\frac{K}{N} - \tau + \epsilon(N)) \tag{183}$$

$$\leq H_q^{-1}(\frac{K}{N} - (1 - \frac{\log_q(|\mathcal{X}|)}{N}) + \epsilon(N)), \tag{184}$$

which corresponds to the claim. Note that where $m = L$ from (130), $\rho = H_q^{-1}(\frac{\log_q(L)}{N} + \epsilon(N))$. Also note that the communication cost remains as described in (133).

Appendix H

Proof. Consider index $i \neq 0$ corresponds to $\mathbf{d}_i \in \mathbb{F}^n \backslash \mathbf{0}$ a column vector and $\mathbf{G} \in \mathbb{F}^{k^* \times n}$ be a random generator matrix that its elements are chosen uniformly at random and independently from $\mathbb{F}$. Note that

$$\mathbf{d}_i \mathbf{G} = \sum_{j=1}^{n} d_i(j,1)\mathbf{G}(j,:) = \mathbf{c}_i. \tag{185}$$

As can be seen $\mathbf{c}_i$ is the summation of a number of rows of $\mathbf{G}$, where by definition is a uniformly and random vector from $\mathbb{F}^n$. Therefore the claim (136) results. □

Appendix I

Proof. Since any $\mathbf{x} \in \mathcal{B}(0,\rho)$ is present in $\mathcal{X}_{\mathcal{C}}$, (139) holds for $\omega(\mathbf{x}) \leq \rho n$.

For $\omega(\mathbf{x}) > \rho n, \mathbf{x} \in \mathbb{F}^n$, we have that

$$\mathbb{P}(\mathbf{x} \in \mathcal{X}_{\mathcal{C}}) = \mathbb{P}(\mathbf{x} \in \mathcal{C}) + \mathbb{P}(\mathbf{x} \notin \mathcal{C})\mathbb{P}(\mathbf{x}\, \textit{be choosen randomly} | \mathbf{x} \notin \mathcal{C}), \tag{186}$$

where $\mathcal{C}$ is the code that has been chosen uniformly at random from $\mathcal{C}_{k^*,n}$. Consider $\mathbf{y} \in \mathbb{F}^n, \omega(y) \geq \rho n$ then since we know that $\mathbb{P}(\mathbf{x} \in \mathcal{C}) = \mathbb{P}(\mathbf{y} \in \mathcal{C})$, by noting that $\mathbf{x}$ and $\mathbf{y}$ has the same chance to be a codeword and $\mathbb{P}(\mathbf{x}\, \textit{be choosen randomly} | \mathbf{x} \notin \mathcal{C}) = \mathbb{P}(\mathbf{y}\, \textit{be choosen randomly} | \mathbf{y} \notin \mathcal{C})$, by the method that has been described in step 3, we have for all $\mathbf{x}, \mathbf{y} \in \mathbb{F}^n$ with $\omega(\mathbf{x}), \omega(\mathbf{y}) > \rho n$,

$$\mathbb{P}(\mathbf{x} \in \mathcal{X}_{\mathcal{C}}) = \mathbb{P}(\mathbf{y} \in \mathcal{X}_{\mathcal{C}}). \tag{187}$$

Now consider the following argument,

$$\sum_{\mathbf{y} \in \mathbb{F}^n \backslash \mathcal{B}(0,\rho)} \mathbb{P}(\mathbf{y} \in \mathcal{X}_{\mathcal{C}}) \overset{(a)}{=} \sum_{\mathbf{y} \in \mathbb{F}^n \backslash \mathcal{B}(0,\rho)} \mathbb{E}[\mathbb{1}(\mathbf{y} \in \mathcal{X}_{\mathcal{C}})] \tag{188}$$

$$\overset{(b)}{=} \mathbb{E}[\sum_{\mathbf{y}\in\mathbb{F}^n\backslash\mathcal{B}(0,\rho)} \mathbb{1}(\mathbf{y} \in \mathcal{X}_{\mathcal{C}})] \tag{189}$$

$$\overset{(c)}{=} \mathbb{E}[q^{n(1-\tau)} - V_q(n,\rho)] = q^{n(1-\tau)} - V_q(n,\rho). \tag{190}$$

Equation (a) holds the average $\mathbb{E}$ is over all codes in $\mathcal{C}_{k^*,n}$ and choices of $\mathcal{X}_{\mathcal{C}}$ which is dependant on $\mathcal{C} \in \mathcal{C}_{k^*,n}$, and (b) is true since two independent summation and average over events has been swapped, (c) results since for every occurrence of $\mathcal{C} \in \mathcal{C}_{k^*,n}$, $q^{n(1-\tau)} - V_q(n,\rho)$ elements of $\mathbb{F}\backslash\mathcal{B}(0,\rho)$ is in the set $\mathcal{X}_{\mathcal{C}}$.

Also since (187) is true, we have

$$(q^n - V_q(n,\rho))\mathbb{P}(\mathbf{x} \in \mathcal{X}_{\mathcal{C}}) = \sum_{\mathbf{y}\in\mathbb{F}^n\backslash\mathcal{B}(0,\rho)} \mathbb{P}(\mathbf{y} \in \mathcal{X}_{\mathcal{C}}), \tag{191}$$

then the claim results.

□

Appendix J

Proof. Now note the following argument for any $i \neq 1, \mathbf{x} \neq \mathbf{0}$,

$$\mathbb{P}[\mathbf{c}_i = \mathbf{x}|\mathbf{x} \in \mathcal{X}_{\mathcal{C}}]\mathbb{P}[\mathbf{x} \in \mathcal{X}_{\mathcal{C}}] \tag{192}$$

$$\overset{(a)}{=} \mathbb{P}[[\mathbf{c}_i = \mathbf{x}] \cap [\mathbf{x} \in \mathcal{X}_{\mathcal{C}}]] \tag{193}$$

$$\overset{(b)}{=} \mathbb{P}[[\mathbf{c}_i = \mathbf{x}] \ \cap \ [\mathbf{c}_i \in \mathcal{X}_{\mathcal{C}}]] \tag{194}$$

$$\overset{(c)}{=} \mathbb{P}[[\mathbf{c}_i = \mathbf{x}] \ \cap \ [True]] \tag{195}$$

$$\overset{(d)}{=} \mathbb{P}[\mathbf{c}_i = \mathbf{x}] \tag{196}$$

$$\overset{(e)}{=} q^{-n}, \tag{197}$$

where (a) is derived from the definition of conditional probability [67], (b) is true since occurrence of the intersection of two events mentioned on the LHS is exactly the same as the RHS of the equation since for both of the events, event $\mathbf{x} = \mathbf{c}_i$ must be true, (c) is true since $\mathcal{X}_{\mathcal{C}}$ always contains the codewords of the random code $\mathcal{C} \in \mathcal{C}_{k^*,n}$, (d) is true since simply intersection of an event with an always true event is the event itself and (e) results from (136) at Lemma 7. Summarily the claim results. □

Appendix K

Proof. Consider a hamming ball of ρ around $\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}$,

- Considering (140), we know that if $|\{0\} \cap \mathcal{B}(\mathbf{x},\rho)| = 1$, then with probability one $\mathbf{c}_0$ covers the point but since $\mathbf{x} \in \mathcal{X}'_{\mathcal{C}}$ therefore always $|\{0\} \cap \mathcal{B}(\mathbf{x},\rho)| = 0$.

- From (140), we know the probability that any point $\mathbf{x}'$ in $(\mathcal{B}(0,\rho)\backslash\{0\})\cap\mathcal{B}(\mathbf{x},\rho)$ be a codeword $i, i\neq 0$ is $\mathbb{P}(\mathbf{c}_i=\mathbf{x}'|\mathbf{x}'\in\mathcal{X}'_{\mathcal{C}})$ which is q^{-n}. Since $\eta_{\mathbf{x}}$ by definition consists of all codewords, these points contribute $(q^{k^*}-1)[|(\mathcal{B}(0,\rho)\backslash\{0\})\cap\mathcal{B}(\mathbf{x},\rho)|q^{-n}$ to the average.
- From (140), we know the probability that any point $\mathbf{x}'$ in $\mathcal{B}(\mathbf{x},\rho)\backslash\mathcal{B}(0,\rho)$ be a codeword $i, i\neq 0$ is $\mathbb{P}(\mathbf{c}_i=\mathbf{x}'|\mathbf{x}'\in\mathcal{X}'_{\mathcal{C}})$ which is $q^{-n(1-\tau)}\delta(n)$.

□

Appendix L

Proof. We proof the Lemma in two steps,

1) We define, $\forall i\in[q^{k^*}]:\ \mathbb{E}(\eta_{\mathbf{x},i}|\mathbf{x}\in\mathcal{X}'_{\mathcal{C}})\triangleq\overline{\eta},\mathbb{E}(\eta_{\mathbf{x},i}\eta_{\mathbf{x},j}|\mathbf{x}\in\mathcal{X}'_{\mathcal{C}})\triangleq\overline{\eta\eta}$.

$$Var(\eta_{\mathbf{x}}|\mathbf{x}\in\mathcal{X}_{\mathcal{C}})\leq(q^{k^*}-1)(q-1)\overline{\eta}(1-\frac{q-2}{q-1}\overline{\eta}), \tag{198}$$

since

$$\begin{aligned} Var(\eta_{\mathbf{x}}|\mathbf{x}\in\mathcal{X}_{\mathcal{C}}) &= \mathbb{E}((\sum_{i=1}^{q^{k^*}}\eta_{\mathbf{x},i})^2|\mathbf{x}\in\mathcal{X}'_{\mathcal{C}})-\mathbb{E}^2(\sum_{i=1}^{q^{k^*}}\eta_{\mathbf{x},i}|\mathbf{x}\in\mathcal{X}'_{\mathcal{C}}) && (199)\\ &= \mathbb{E}(\sum_{i=1}^{q^{k^*}}\eta_{\mathbf{x},i}^2+\sum_{p.i=1,i\neq p}^{q^{k^*}}\eta_{\mathbf{x},i}\eta_{\mathbf{x},p}|\mathbf{x}\in\mathcal{X}'_{\mathcal{C}})-\mathbb{E}^2(\sum_{i=1}^{q^{k^*}}\eta_{\mathbf{x},i}|\mathbf{x}\in\mathcal{X}'_{\mathcal{C}}) && (200)\\ &= \sum_{i=1}^{q^{k^*}}\mathbb{E}(\eta_{\mathbf{x},i}|\mathbf{x}\in\mathcal{X}'_{\mathcal{C}})+\sum_{p,i=1,i\neq p}^{q^{k^*}}\mathbb{E}(\eta_{\mathbf{x},i}\eta_{\mathbf{x},p}|\mathbf{x}\in\mathcal{X}'_{\mathcal{C}})-\mathbb{E}^2(\sum_{i=1}^{q^{k^*}}\eta_{\mathbf{x},i}|\mathbf{x}\in\mathcal{X}'_{\mathcal{C}}) && (201)\\ &= q^{k^*}\overline{\eta}+q^{k^*}(q^{k^*}-1)\overline{\eta\eta}-q^{2k^*}\overline{\eta}. && (202)\end{aligned}$$

Now substituting in (198),we modify the claim as follows,

$$q^{k^*}\overline{\eta}+q^{k^*}(q^{k^*}-1)\overline{\eta\eta}-q^{2k^*}\overline{\eta}-(q^{k^*}-1)(q-1)\overline{\eta}(1-\frac{q-2}{q-1}\overline{\eta})\leq 0. \tag{203}$$

To prove (203), we begin from the left hand side,

$$\begin{aligned} & q^{k^*}\overline{\eta}+q^{k^*}(q^{k^*}-1)\overline{\eta\eta}-q^{2k^*}\overline{\eta}-(q^{k^*}-1)(q-1)\overline{\eta}(1-\frac{q-2}{q-1}\overline{\eta}) && (204)\\ & \overset{(a)}{\leq} q^{k^*}\overline{\eta}+q^{k^*}(q^{k^*}-1)\overline{\eta}^2-q^{2k^*}\overline{\eta}-(q^{k^*}-1)(q-1)\overline{\eta}(1-\frac{q-2}{q-1})\overline{\eta} && (205)\\ & \overset{(b)}{\leq} -(q^{k^*+1}-2q^{k^*}-q+1)\overline{\eta}+(q^{k^*+1}-3q^{k^*}-q+2)\overline{\eta}^2 && (206)\\ & \overset{(c)}{\leq} 0. && (207)\end{aligned}$$

Equation (a) holds since $\overline{\eta\eta} \leq \overline{\eta}^2$, (b) is true by simply arranging the equations and finally (c) is true because of two reasons first $0 \leq \overline{\eta} \leq 1$ and second by noticing that $q^{k^*+1} - 2q^{k^*} - q + 1 > q^{k^*+1} - 3q^{k^*} - q + 2$ as far as $q^{k^*} \geq 1$.

2) Now we also have to prove that,

$$\frac{(q^{k^*}-1)(q-1)\overline{\eta}(1-\frac{q-2}{q-1}\overline{\eta})}{\mathbb{E}(\eta_{\mathbf{x}}|\mathbf{x} \in \mathcal{X}'_{\mathcal{C}})q^2} \leq 1. \tag{208}$$

Therefore by beginning from the right side we have,

$$\frac{(q^{k^*}-1)(q-1)\overline{\eta}(1-\frac{q-2}{q-1}\overline{\eta})}{\mathbb{E}(\eta_{\mathbf{x}})q^2} \overset{(a)}{=} \frac{(q^{k^*}-1)(q-1)(1-\frac{q-2}{q-1}\overline{\eta})}{q^{k^*}q^2} \tag{209}$$

$$\overset{(b)}{=} (\frac{q^{k^*}-1}{q^{k^*}}) \times (\frac{q-1}{q^2}) \times (1 - \frac{q-2}{q-1}\overline{\eta}) \tag{210}$$

$$\overset{(c)}{\leq} 1. \tag{211}$$

Equation (a) holds since $\mathbb{E}(\eta_{\mathbf{x}}) = \sum_{i=1}^{q^{k^*}} \mathbb{E}(\eta_{\mathbf{x},i}) = q^{k^*}\overline{\eta}$ and (b) holds simply by rearranging the statement and finally (c) is true since each multiplicative element is is non negative and less than 1.

combining the results of the two steps, namely (198) and (208), (146) results. □

APPENDIX M

Proof. Let's define,

$$\mathcal{I}_{(\omega(\mathbf{x}),\rho)} \triangleq |\mathcal{B}(\mathbf{x},\rho) \cap \mathcal{B}(0,\rho)|. \tag{212}$$

We know that

$$\rho n = \underset{\omega(\mathbf{x}):\mathbf{x}\in\mathcal{X}_{\mathcal{C}}\backslash\mathcal{B}(0,\rho)}{argmax} |\mathcal{I}_{(\omega(\mathbf{x}),\rho)}|, \tag{213}$$

since the distance between $\mathbf{x} \in \mathcal{X}_{\mathcal{C}}\backslash\mathcal{B}(0,\rho)$ and $\mathbf{0}$ is minimum where $\omega(\mathbf{x}) = \rho n$. We know that

$$|\mathcal{B}(\mathbf{x},\rho)\backslash\mathcal{B}(0,\rho)| = Vol(n,\rho) - |\mathcal{I}(\rho,\rho)|. \tag{214}$$

Now Lets focus on the case where $q = 2$ and $0 \leq \rho\frac{1}{2}$, from [68] we have,

$$|\mathcal{I}(\rho,\rho)| = \sum_{i=0}^{\lfloor\frac{n\rho}{2}\rfloor}\sum_{j=0}^{i}\binom{n\rho}{i}\binom{n-n\rho}{j} + \sum_{i=\lfloor\frac{n\rho}{2}\rfloor+1}^{n\rho}\sum_{j=0}^{n\rho-i}\binom{n\rho}{i}\binom{n-n\rho}{j}. \tag{215}$$

Note that here $n\rho \leq n - n\rho$. Also we know that,

$$vol(n,\rho) = \sum_{i=0}^{\rho n}\binom{n}{i} = \sum_{i=0}^{\lfloor\frac{n\rho}{2}\rfloor}\sum_{j=0}^{n\rho-i}\binom{n\rho}{i}\binom{n-n\rho}{j} \tag{216}$$

$$+ \sum_{i=\lfloor\frac{n\rho}{2}\rfloor+1}^{n\rho}\sum_{j=0}^{n\rho-i}\binom{n\rho}{i}\binom{n-n\rho}{j}. \tag{217}$$

Then by substituting (215) and (217) into (214), we conclude that,

$$|\mathcal{B}(\mathbf{x},\rho)\backslash\mathcal{B}(0,\rho)| = \sum_{i=0}^{\lfloor\frac{n\rho}{2}\rfloor}\sum_{j=i+1}^{n\rho-i}\binom{n\rho}{i}\binom{n-n\rho}{j}. \tag{218}$$

since $0<\rho\leq\frac{1}{2}$, we have

$$\binom{n}{n\rho} = \sum_{i=0}^{\lfloor\frac{n\rho}{2}\rfloor}\binom{n\rho}{i}\binom{n-n\rho}{n\rho-i} + \sum_{i=\lfloor\frac{n\rho}{2}\rfloor+1}^{n\rho}\binom{n\rho}{i}\binom{n-n\rho}{n\rho-i} \tag{219}$$

$$= \sum_{i=0}^{\lfloor\frac{n\rho}{2}\rfloor}\binom{n\rho}{i}\binom{n-n\rho}{n\rho-i} + \sum_{i=\lfloor\frac{n\rho}{2}\rfloor+1}^{n\rho}\binom{n\rho}{n\rho-i}\binom{n-n\rho}{n\rho-i} \tag{220}$$

$$= \sum_{i=0}^{\lfloor\frac{n\rho}{2}\rfloor}\binom{n\rho}{i}\binom{n-n\rho}{n\rho-i} + \sum_{i=0}^{n\rho-\lfloor\frac{n\rho}{2}\rfloor-1}\binom{n\rho}{i}\binom{n-n\rho}{i}, \tag{221}$$

Note that,

$$|\mathcal{B}(\mathbf{x},\rho)\backslash\mathcal{B}(0,\rho)| \overset{(a)}{=} \sum_{i=0}^{\lfloor\frac{n\rho}{2}\rfloor}\sum_{j=i+1}^{n\rho-i}\binom{n\rho}{i}\binom{n-n\rho}{j} \tag{222}$$

$$\overset{(b)}{=} \sum_{i=0}^{\lfloor\frac{n\rho}{2}\rfloor}\sum_{j=i+2}^{n\rho-i-1}\binom{n\rho}{i}\binom{n-n\rho}{j} \tag{223}$$

$$+ \sum_{i=0}^{\lfloor\frac{n\rho}{2}\rfloor}\binom{n\rho}{i}\binom{n-n\rho}{n\rho-i}(q-1)^{n\rho} \tag{224}$$

$$+ \sum_{i=0}^{\lfloor\frac{n\rho}{2}\rfloor-\mathbb{1}[\lfloor\frac{n\rho}{2}\rfloor=\frac{n\rho}{2}]}\binom{n\rho}{i}\binom{n-n\rho}{i+1} \tag{225}$$

$$\overset{(c)}{\geq} \sum_{i=0}^{\lfloor\frac{n\rho}{2}\rfloor}\binom{n\rho}{i}\binom{n-n\rho}{n\rho-i} + \sum_{i=0}^{n\rho-\lfloor\frac{n\rho}{2}\rfloor-1}\binom{n\rho}{i}\binom{n-n\rho}{i} \tag{226}$$

$$\overset{(d)}{=} \binom{n}{n\rho} \overset{(e)}{\geq} 2^{nH(\rho)-o(n)}. \tag{227}$$

Equation (a) holds because of (218), (b) results by just expanding the inner summation, (c) is true since the first statement of RHS is non-zero, the second term is also present on the left side of LHS and the third one is true because of Lemma 1, (d) is true because of (221) and finally (e) is the result of strilings non-inequality for binomial statements where $0<\rho\leq\frac{1}{2}$.

Therefore we have

$$|\mathcal{B}(\mathbf{x},\rho)\backslash\mathcal{B}(0,\rho)| \overset{(a)}{=} \sum_{i=0}^{\lfloor\frac{n\rho}{2}\rfloor}\sum_{j=i+1}^{n\rho-i}\binom{n\rho}{i}\binom{n-n\rho}{j} \tag{228}$$

$$\overset{(b)}{=} \sum_{i=0}^{\lfloor\frac{n\rho}{2}\rfloor}\sum_{j=i+2}^{n\rho-i-1}\binom{n\rho}{i}\binom{n-n\rho}{j} \tag{229}$$

$$+\sum_{i=0}^{\lfloor \frac{n\rho}{2}\rfloor}\binom{n\rho}{i}\binom{n-n\rho}{n\rho-i} \tag{230}$$

$$+\sum_{i=0}^{\lfloor \frac{n\rho}{2}\rfloor-\mathbb{1}[\lfloor \frac{n\rho}{2}\rfloor=\frac{n\rho}{2}]}\binom{n\rho}{i}\binom{n-n\rho}{i+1} \tag{231}$$

$$\overset{(c)}{\geq}\sum_{i=0}^{\lfloor \frac{n\rho}{2}\rfloor}\binom{n\rho}{i}\binom{n-n\rho}{n\rho-i}+\sum_{i=0}^{n\rho-\lfloor \frac{n\rho}{2}\rfloor-1}\binom{n\rho}{i}\binom{n-n\rho}{i} \tag{232}$$

$$\overset{(d)}{=}\binom{n}{n\rho}\overset{(e)}{\geq}2^{nH(\rho)-o(n)}. \tag{233}$$

Equation (a) holds because of (218), (b) results by just expanding the inner summation, (c) is true since the first statement of RHS is non-zero, the second term is also present on the left side of LHS and the third one is true because of Lemma 1, (d) is true because of (221) and finally (e) is the result of strilings non-inequality for binomial statements.

Lemma 1. If $0<\rho\leq 1/2$ and $\rho n\in\mathbb{N}$ then,

$$\sum_{i=0}^{\lfloor \frac{n\rho}{2}\rfloor-\mathbb{1}[\lfloor \frac{n\rho}{2}\rfloor=\frac{n\rho}{2}]}\binom{n\rho}{i}\binom{n-n\rho}{i+1}\geq\sum_{i=0}^{n\rho-\lfloor \frac{n\rho}{2}\rfloor-1}\binom{n\rho}{i}\binom{n-n\rho}{i}. \tag{234}$$

We solve the problem by discussing about the following two cases,

- If $2|\rho n$, then $\lfloor\frac{n\rho}{2}\rfloor=\frac{n\rho}{2}$, our claim becomes

$$\sum_{i=0}^{\frac{n\rho}{2}-1}\binom{n\rho}{i}\binom{n-n\rho}{i+1}\geq\sum_{i=0}^{\frac{n\rho}{2}-1}\binom{n\rho}{i}\binom{n-n\rho}{i}. \tag{235}$$

It is true since $0<\rho\leq\frac{1}{2}$ we have $\forall i\in[0:n\rho/2-1]:i+1\leq(n-n\rho)/2$ then

$$\binom{n-n\rho}{i+1}\geq\binom{n-n\rho}{i}. \tag{236}$$

- If $2\nmid\rho n$, then $\lfloor\frac{n\rho}{2}\rfloor=\frac{n\rho}{2}-0.5$ our claim becomes

$$\sum_{i=0}^{\frac{n\rho}{2}-0.5}\binom{n\rho}{i}\binom{n-n\rho}{i+1}\geq\sum_{i=0}^{\frac{n\rho}{2}-0.5}\binom{n\rho}{i}\binom{n-n\rho}{i}. \tag{237}$$

Then we can discuss two cases,

1) If $n\in\mathbb{N}$ is an odd number then $n-np$ is an even number and also since $\rho n\in\mathbb{N}$, $0<\rho<1/2$ and $n\rho\leq\frac{n-1}{2}$ consequently $2(\frac{n\rho}{2}+0.5)\leq n-n\rho$, therefore similar to the previous case we have $\forall i\in[0:n\rho/2-0.5]:i+1\leq(n-n\rho)/2$,

$$\binom{n-n\rho}{i+1}\geq\binom{n-n\rho}{i}, \tag{238}$$

and the claim results.

2) If $n \in \mathbb{N}$ is an even number then $n - np$ is an odd number. Since $0 < \rho \leq 1/2$ therefore similar to the previous case we have $\forall i \in [0 : n\rho/2 - 0.5] : i + 1 \leq (n - n\rho + 1)/2$,
$$\binom{n - n\rho}{i + 1} \geq \binom{n - n\rho}{i}, \tag{239}$$
and the claim results.

Also If $q > 2$, Note the following Lemma,

Lemma 13. We claim that,

$$|\mathcal{I}(\rho, \rho)| = \sum_{i+j=\rho n,\, i \leq j} \binom{n - \rho n}{i} \binom{\rho n}{j} (q - 1)^{\rho n} + g(q), \tag{240}$$

where polynomial $g(q)$ at most $V_q(n, \rho - 1/n)$.

Proof. Let $\mathbf{a} \in \mathbb{F}^n, \omega(\mathbf{a}) = \rho n$ and $\mathbf{b} \in \mathcal{B}(\mathbf{0}, \rho) \cap \mathcal{B}(\mathbf{a}, \rho)$. Now given $\mathbf{a}$, $\mathbf{b}$, their elements can be grouped into three parts,

$$\mathcal{B} \triangleq \{b | \mathbf{a}(i) = 0, \mathbf{b}(i) \neq 0, i \in [n]\}, \tag{241}$$

$$\mathcal{C} \triangleq \{c | \mathbf{a}(i) \neq 0, \mathbf{b}(i) = -\mathbf{b}(i), i \in [n]\}, \tag{242}$$

Let's define,

$$x \triangleq |\mathcal{B}|, \tag{243}$$

$$y \triangleq |\mathcal{C}|, \tag{244}$$

Without loss of generality consider $\mathbf{a}$ to be,

$$\mathbf{a} = [0, \ldots, 0, a_1, a_2, \ldots, a_y, \ldots, a_{\rho n}]. \tag{245}$$

where the elements for $\mathbf{b}$ can be shown as Fig. 1. Since $\omega(\mathbf{b}) \leq \rho n$ and $d(\mathbf{a}, \mathbf{b}) \leq \rho n$, we have,

$$x + y \leq \rho n, \tag{246}$$

$$x \leq y. \tag{247}$$

and for every tuple $(x, y) \in [\rho n]^2$, there are $\binom{n-\rho n}{x}\binom{\rho n}{y}(q-1)^{x+y}$ points in the intersection. Therefore for the case where $x+y = \rho n$, there is $\sum_{i+j=\rho n,\, i \leq j} \binom{n-\rho n}{i}\binom{\rho n}{j}(q-1)^{\rho n}$ points and for the case $x+y < \rho n$, because of (247), there are at most $V_q(n, \rho - 1/n)$ points in the intersection and the claim results. □

Now note that

$$|\mathcal{B}(\mathbf{x}, \rho) \backslash \mathcal{B}(0, \rho)| \overset{(a)}{=} V_q(n, \rho) - |\mathcal{I}(\rho, \rho)| \tag{248}$$

$$\overset{(b)}{=} V_q(n, \rho - 1/n) + \binom{n - \rho n}{\rho n} \tag{249}$$

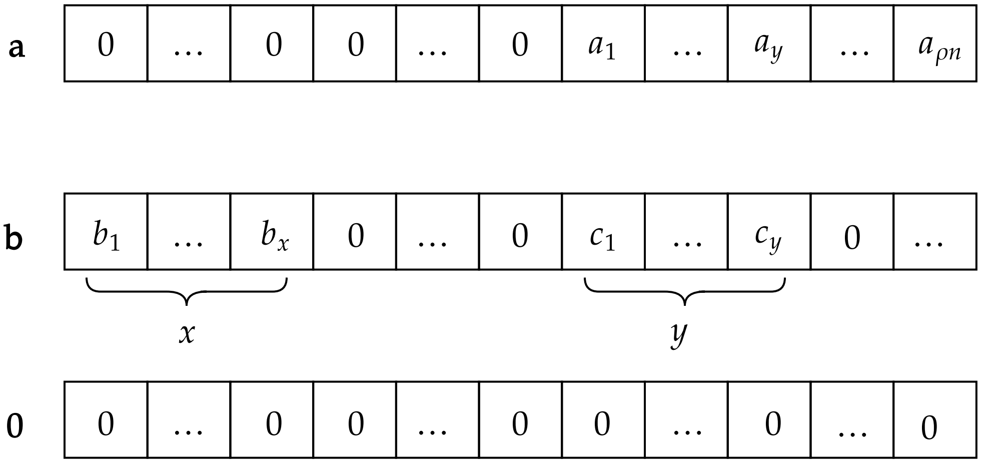
a
0 … 0 0 … 0 a_1 … a_y … $a_{\rho n}$
b
b_1 … b_x 0 … 0 c_1 … c_y 0 …
x
y
0
0 … 0 0 … 0 0 … 0 … 0

Fig. 6. Bounding the intersection for $q > 2$.

$$- \sum_{i+j=\rho n,\, i\leq j} \binom{n-\rho n}{i}\binom{\rho n}{j}(q-1)^{\rho n} - g(q) \tag{250}$$

$$\overset{(c)}{\geq} \sum_{i+j=\rho n,\, i>j} \binom{n-\rho n}{i}\binom{\rho n}{j}(q-1)^{\rho n} \tag{251}$$

$$\overset{(d)}{=} \sum_{j=\min\{0,2\rho n-n\}}^{\lfloor \frac{\rho n}{2}\rfloor -1} \binom{n-\rho n}{\rho n-j}\binom{\rho n}{j}(q-1)^{\rho n}. \tag{252}$$

Now from Stirling's bound we know that,

$$\forall j \in [\min\{0, 2\rho n - n\}, \lfloor \frac{\rho n}{2} \rfloor - 1], \tag{253}$$

$$\binom{n-\rho n}{\rho n-j}\binom{\rho n}{j} \geq \sqrt{\frac{n-\rho n}{8(\rho n-j)(n-2\rho n+j)}}2^{nH((\rho n-j)/(n-\rho n))} \tag{254}$$

$$\times \sqrt{\frac{\rho n}{8(j)(\rho n-j)}}2^{nH(j/\rho n)} \tag{255}$$

$$= 2^{n[H((\rho n-j)/\rho n)+H((\rho n-j)/(n-\rho n))]-o(n)}. \tag{256}$$

Let's define,

$$\kappa \triangleq \frac{\rho n - j}{\rho n}. \tag{257}$$

Now the exponent in (256) would become,

$$n[H(\kappa) + H(\kappa\frac{\rho}{1-\rho})] - o(n). \tag{258}$$

Now suppose that n be large enough so that $n\rho^2 \in \mathbb{N}$ then for the case where $0 < \rho \leq \frac{1}{2}$ set $0 \leq j = n\rho^2 \leq \lfloor \frac{n\rho}{2} \rfloor - 1$, then $\kappa = 1 - \rho$, and for the case where $\frac{1}{2} < \rho \leq -1/2 + \sqrt{5}/2$ set $2\rho n - n \leq j = n\rho(1-\rho) \leq \lfloor \frac{n\rho}{2} \rfloor - 1$, then $\kappa = \rho$. By substituting in (258) and utilizing it in (256), then we have

$$\sum_{j=\min\{0,2\rho n-n\}}^{\lfloor \frac{\rho n}{2} \rfloor -1} \binom{n-\rho n}{\rho n - j}\binom{\rho n}{j}(q-1)^{\rho n} \geq q^{nH_q(\rho)-o(n)}, \tag{259}$$

where $0 < \rho \leq -1/2 + \sqrt{5}/2$. □

Appendix N

We define $h(x) \triangleq -x\log_q(x)$ and we know that,

$$h(x) \leq H_q(x),\ 0 \leq x \leq 1 - 1/q, \tag{260}$$

Therefore

$$H_q^{-1}(x) \leq h^{-1}(x). \tag{261}$$

we know that if $y = x\ln(x)$, then $x = e^{W(y)}$ where $W(.)$ is Lambert function. Note that $h(x) = -\log_q(e)x\ln(x)$, which implies

$$h^{-1}(x) = e^{W(-\ln(q)x)}. \tag{262}$$

Let $c > 0$ be a real number, note that

$$\lim_{T\to\infty} Te^{W(-c/T)} \overset{(a)}{=} \lim_{T\to\infty} \frac{e^{W(-c/T)}}{1/T} \tag{263}$$

$$\overset{(b)}{=} \lim_{T\to\infty} \frac{e^{W(-c/T)}\frac{1}{-c/T+e^{W(-c/T)}}cT^{-2}}{-T^{-2}} \tag{264}$$

$$\overset{(c)}{=} \lim_{T\to\infty} \frac{ce^{W(-c/T)}}{c/T - e^{W(-c/T)}} \tag{265}$$

$$\overset{(d)}{=} \lim_{T\to\infty} \frac{cTe^{W(-c/T)}}{c - Te^{W(-c/T)}} \tag{266}$$

$$\overset{(e)}{=} \frac{\lim_{T\to\infty} cTe^{W(-c/T)}}{\lim_{T\to\infty} c - Te^{W(-c/T)}}, \tag{267}$$

where (a) follows by rearranging the algebraic statements, (b) holds by using L'Hopital's rule, (c) and (d) holds by rearranging and (e) follows from existence assumption of the limit and algebraic limit theorem. Now taking $\lim_{T\to\infty} Te^{W(-c/T)}$ as the unknown parameters of (267), there is no other way but

$$\lim_{T\to\infty} Te^{W(-c/T)} = 0. \tag{268}$$

Now we conclude,

$$\lim_{T\to\infty} TH_q^{-1}((c/T) \overset{(a)}{\leq} \lim_{T\to\infty} Th^{-1}(c/T) \tag{269}$$

$$\overset{(b)}{\leq} \lim_{T\to\infty} Te^{W(-\ln(q)c/T)} \tag{270}$$

$$\overset{(c)}{=} 0, \tag{271}$$

where (a) follows by (261), (b) is the results of (262) and (c) follows by (268). Thus the claim results.

Appendix O

We see that

$$c/T = H_q(f/T). \tag{272}$$

Taking derivative from both sides with respect to T, we have

$$c = \log_q(\frac{f/T}{1-f/T}(q-1))(\frac{\partial f}{\partial T}T - f). \tag{273}$$

Now putting (272) into (273) and rearranging, we get the claim.

References

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

[2] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, “Spark: Cluster computing with working sets,” in *2nd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 10)*, 2010.

[3] T. Jahani-Nezhad and M. A. Maddah-Ali, “Codedsketch: A coding scheme for distributed computation of approximated matrix multiplication,” *IEEE Transactions on Information Theory*, vol. 67, no. 6, pp. 4185–4196, 2021.

[4] J. Wang, Z. Jia, and S. A. Jafar, “Price of precision in coded distributed matrix multiplication: A dimensional analysis,” in *2021 IEEE Information Theory Workshop (ITW)*, pp. 1–6, IEEE, 2021.

[5] E. Ozfatura, S. Ulukus, and D. Gündüz, “Coded distributed computing with partial recovery,” *IEEE Transactions on Information Theory*, 2021.

[6] K. Wan, H. Sun, M. Ji, D. Tuninetti, and G. Caire, “Cache-aided matrix multiplication retrieval,” *IEEE Transactions on Information Theory*, 2022.

[7] S. Li, Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, “A scalable framework for wireless distributed computing,” *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2643–2654, 2017.

[8] F. Haddadpour, M. M. Kamani, M. Mahdavi, and V. Cadambe, “Trading redundancy for communication: Speeding up distributed sgd for non-convex optimization,” in *International Conference on Machine Learning*, pp. 2545–2554, PMLR, 2019.

[9] C.-S. Yang, R. Pedarsani, and A. S. Avestimehr, “Coded computing in unknown environment via online learning,” in *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 185–190, IEEE, 2020.

[10] N. Charalambides, H. Mahdavifar, and A. O. Hero III, “Numerically stable binary coded computations,” *arXiv preprint arXiv:2109.10484*, 2021.

[11] M. Soleymani, H. Mahdavifar, and A. S. Avestimehr, “Analog lagrange coded computing,” *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 1, pp. 283–295, 2021.

[12] H. Sun and S. A. Jafar, “The capacity of private computation,” *IEEE Transactions on Information Theory*, vol. 65, no. 6, pp. 3880–3897, 2018.

[13] M. Soleymani and H. Mahdavifar, “Distributed multi-user secret sharing,” *IEEE Transactions on Information Theory*, vol. 67, no. 1, pp. 164–178, 2020.

[14] A. Khalesi, M. Mirmohseni, and M. A. Maddah-Ali, “The capacity region of distributed multi-user secret sharing,” *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 3, pp. 1057–1071, 2021.

[15] M. Soleymani, H. Mahdavifar, and A. S. Avestimehr, “Privacy-preserving distributed learning in the analog domain,” *arXiv preprint arXiv:2007.08803*, 2020.

[16] M. Soleymani, R. E. Ali, H. Mahdavifar, and A. S. Avestimehr, “List-decodable coded computing: Breaking the adversarial toleration barrier,” *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 3, pp. 867–878, 2021.

[17] R. Bitar, M. Xhemrishi, and A. Wachter-Zeh, “Adaptive private distributed matrix multiplication,” *IEEE Transactions on Information Theory*, 2022.

[18] C. Hofmeister, R. Bitar, M. Xhemrishi, and A. Wachter-Zeh, “Secure private and adaptive matrix multiplication beyond the singleton bound,” *arXiv preprint arXiv:2108.05742*, 2021.

[19] H. Akbari-Nodehi and M. A. Maddah-Ali, “Secure coded multi-party computation for massive matrix operations,” *IEEE Transactions on Information Theory*, vol. 67, no. 4, pp. 2379–2398, 2021.

[20] Z. Jia and S. A. Jafar, “On the capacity of secure distributed batch matrix multiplication,” *IEEE Transactions on Information Theory*, vol. 67, no. 11, pp. 7420–7437, 2021.

[21] Z. Chen, Z. Jia, Z. Wang, and S. A. Jafar, “Gcsa codes with noise alignment for secure coded multi-party batch matrix multiplication,” *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 1, pp. 306–316, 2021.

[22] C.-S. Yang and A. S. Avestimehr, “Coded computing for secure boolean computations,” *IEEE Journal on Selected Areas in Information Theory*, vol. 2, no. 1, pp. 326–337, 2021.

[23] Q. Yu and A. S. Avestimehr, “Coded computing for resilient, secure, and privacy-preserving distributed matrix multiplication,” *IEEE Transactions on Communications*, vol. 69, no. 1, pp. 59–72, 2020.

[24] M. Xhemrishi, R. Bitar, and A. Wachter-Zeh, “Distributed matrix-vector multiplication with sparsity and privacy guarantees,” *arXiv preprint arXiv:2203.01728*, 2022.

[25] N. Raviv, I. Tamo, R. Tandon, and A. G. Dimakis, “Gradient coding from cyclic mds codes and expander graphs,” *IEEE Transactions on Information Theory*, vol. 66, no. 12, pp. 7475–7489, 2020.

[26] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, “Speeding up distributed machine learning using codes,” *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1514–1529, 2017.

[27] M. Egger, R. Bitar, A. Wachter-Zeh, and D. Gündüz, “Efficient distributed machine learning via combinatorial multi-armed bandits,” *arXiv preprint arXiv:2202.08302*, 2022.

[28] K. Wan, H. Sun, M. Ji, and G. Caire, “Distributed linearly separable computation,” *IEEE Transactions on Information Theory*, 2021.

[29] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, “Straggler mitigation in distributed matrix multiplication: Fundamental limits and optimal coding,” *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1920–1933, 2020.

[30] Q. Yu, M. Maddah-Ali, and S. Avestimehr, “Polynomial codes: an optimal design for high-dimensional coded matrix multiplication,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[31] Z. Jia and S. A. Jafar, “Cross subspace alignment codes for coded distributed batch computation,” *IEEE Transactions on Information Theory*, vol. 67, no. 5, pp. 2821–2846, 2021.

[32] A. Behrouzi-Far and E. Soljanin, “Efficient replication for straggler mitigation in distributed computing,” *arXiv preprint arXiv:2006.02318*, 2020.

[33] J. S. Ng, W. Y. B. Lim, N. C. Luong, Z. Xiong, A. Asheralieva, D. Niyato, C. Leung, and C. Miao, “A survey of coded distributed computing,” *arXiv preprint arXiv:2008.09048*, 2020.

[34] S. Li and S. Avestimehr, *Coded Computing: Mitigating Fundamental Bottlenecks in Large-Scale Distributed Computing and Machine Learning*, vol. 17. 2020.

[35] A. C.-C. Yao, “Communication complexity and its applications,” in *International Workshop on Frontiers in Algorithmics*, pp. 2–2, Springer, 2009.

[36] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, “A survey on distributed machine learning,” *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1–33, 2020.

[37] S. Ulukus, S. Avestimehr, M. Gastpar, S. Jafar, R. Tandon, and C. Tian, “Private retrieval, computing and learning: Recent progress and future challenges,” *IEEE Journal on Selected Areas in Communications*, 2022.

[38] S. Wang, J. Liu, N. Shroff, and P. Yang, “Fundamental limits of coded linear transform,” *arXiv preprint arXiv:1804.09791*, 2018.

[39] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, “A fundamental tradeoff between computation and communication in distributed computing,” *IEEE Transactions on Information Theory*, vol. 64, no. 1, pp. 109–128, 2017.

[40] Q. Yu, M. Maddah-Ali, and S. Avestimehr, “Polynomial codes: an optimal design for high-dimensional coded matrix multiplication,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[41] S. Dutta, M. Fahim, F. Haddadpour, H. Jeong, V. Cadambe, and P. Grover, “On the optimal recovery threshold of coded matrix multiplication,” *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 278–301, 2019.

[42] A. Reisizadeh, S. Prakash, R. Pedarsani, and A. S. Avestimehr, “Codedreduce: A fast and robust framework for gradient aggregation in distributed learning,” *IEEE/ACM Transactions on Networking*, 2021.

[43] N. Woolsey, R.-R. Chen, and M. Ji, “A new combinatorial coded design for heterogeneous distributed computing,” *IEEE Transactions on Communications*, vol. 69, no. 9, pp. 5672–5685, 2021.

[44] N. Woolsey, R.-R. Chen, and M. Ji, “Coded elastic computing on machines with heterogeneous storage and computation speed,” *IEEE Transactions on Communications*, vol. 69, no. 5, pp. 2894–2908, 2021.

[45] N. Woolsey, J. Kliewer, R.-R. Chen, and M. Ji, “A practical algorithm design and evaluation for heterogeneous elastic computing with stragglers,” *arXiv preprint arXiv:2107.08496*, 2021.

[46] M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, and H. V. Poor, “Distributed learning in wireless networks: Recent progress and future challenges,” *IEEE Journal on Selected Areas in Communications*, 2021.

[47] M. Zinkevich, M. Weimer, L. Li, and A. Smolcohen1985gooda, “Parallelized stochastic gradient descent,” *Advances in neural information processing systems*, vol. 23, 2010.

[48] T. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman, “Project adam: Building an efficient and scalable deep learning training system,” in *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pp. 571–582, 2014.

[49] K. Wan, H. Sun, M. Ji, and G. Caire, “Distributed linearly separable computation,” *IEEE Transactions on Information Theory*, pp. 1–1, 2021.

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

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

[52] W. Halbawi, N. Azizan, F. Salehi, and B. Hassibi, “Improving distributed gradient descent using reed-solomon codes,” in *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 2027–2031, IEEE, 2018.

[53] S. Dutta, V. Cadambe, and P. Grover, “Short-dot: Computing large linear transforms distributedly using coded short dot products,” *Advances In Neural Information Processing Systems*, vol. 29, 2016.

[54] A. Ramamoorthy, L. Tang, and P. O. Vontobel, “Universally decodable matrices for distributed matrix-vector multiplication,” in *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 1777–1781, IEEE, 2019.

[55] A. B. Das and A. Ramamoorthy, “Distributed matrix-vector multiplication: A convolutional coding approach,” in *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 3022–3026, IEEE, 2019.

[56] F. Haddadpour and V. R. Cadambe, “Codes for distributed finite alphabet matrix-vector multiplication,” in *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 1625–1629, IEEE, 2018.

[57] S. Wang, J. Liu, and N. Shroff, “Coded sparse matrix multiplication,” in *International Conference on Machine Learning*, pp. 5152–5160, PMLR, 2018.

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

[59] V. M. Blinovskii, “Lower asymptotic bound on the number of linear code words in a sphere of given radius in (f_q)^n,” *Problemy Peredachi Informatsii*, vol. 23, no. 2, pp. 50–53, 1987.

[60] G. Cohen, I. Honkala, S. Litsyn, and A. Lobstein, *Covering codes*. Elsevier, 1997.

[61] R. M. Roth, “Introduction to coding theory,” *IET Communications*, vol. 47, 2006.

[62] G. Cohen and P. Frankl, “Good coverings of hamming spaces with spheres,” *Discrete Mathematics*, vol. 56, no. 2-3, pp. 125–131, 1985.

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

[64] A. Reisizadeh, S. Prakash, R. Pedarsani, and A. S. Avestimehr, “Tree gradient coding,” in *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 2808–2812, IEEE, 2019.

[65] K. Wan, H. Sun, M. Ji, and G. Caire, “On secure distributed linearly separable computation,” *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 3, pp. 912–926, 2022.

[66] G. Cohen, “A nonconstructive upper bound on covering radius,” *IEEE Transactions on Information Theory*, vol. 29, no. 3, pp. 352–353, 1983.

[67] A. N. Kolmogoroff, *Foundations of the Theory of Probability*. Chelsea Publishing Company, 1956.

[68] H. E. Danoyan, “On some properties of intersection and union of spheres in hamming metric,” *Mathematical Problems of Computer Science*, vol. 39, pp. 119–124, 2013.