Multi-Functional Distributed Computing

Derya Malak*, Mohammad Reza Deylam Salehi*, Berksan Serbetci†, and Petros Elia*
* Communication Systems Department, EURECOM, Biot Sophia Antipolis, FRANCE
*{malak, deylam, elia}@eurecom.fr, † fberks@gmail.com

***Abstract*—The work here studies the communication cost for a multi-server, multi-task distributed computation framework, and does so for a broad class of functions and data statistics. Considering the framework where a user seeks the computation of multiple complex (conceivably non-linear) tasks from a set of distributed servers, we establish communication cost upper bounds for a variety of data statistics, function classes and data placements across the servers. To do so, we proceed to apply, for the first time here, Körner's characteristic graph approach — which is known to capture the structural properties of data and functions — to the promising framework of multi-server multi-task distributed computing. Going beyond the general expressions, and in order to offer clearer insight, we also consider the well-known scenario of cyclic dataset placement and linearly separable functions over the binary field, in which case our approach exhibits considerable gains over the state of the art. Similar gains are identified for multi-linear functions.**

***Index Terms*—Distributed computation; linearly separable functions; nonlinear functions; functional compression; characteristic graph entropy; skewed statistics; and data correlations.**

I. Introduction

As computing requirements become increasingly challenging, distributed computing models have also evolved to be increasingly complex. One such recent model is the multi-server multi-function distributed computing model that consists of a master node, a set of distributed servers, and a user demanding the computation of multiple functions. The master contains the set of all datasets and allocates them to the servers which are then responsible for computing a set of specific subfunctions of datasets. This setting was recently studied by Wan *et al.* in [1] for the class of linearly separable functions, which nicely captures a wide range of real-world tasks [2] such as convolution, the discrete Fourier transform, and a variety of other cases as well. This same work bounded the communication cost, employing linear encoding and linear decoding that leverage the structure of requests.

At the same time though there is a growing need to consider more general classes of functions, including nonlinear functions such as is often the case with subfunctions that produce intermediate values in MapReduce operations [2], or that relate to quantization, classification, and optimization [3]. Intense interest can also be identified in the aforementioned problem of distributed matrix multiplication, which has been explored in a plethora of works including [4]–[7]. In addition to matrix multiplication, other important non-linear function classes include sparse polynomial multiplication [8], as well as permutation invariant [9] and nomographic functions [10].

In this paper, leveraging fundamental principles from functional compression, we study a general multi-server multi-function distributed computing framework composed of a single user requesting a set of functions, computed with the assistance of distributed servers that have partial access to datasets. To achieve our goal, we will draw from the powerful literature, see e.g., [11]–[15], on *Characteristic graphs*, introduced by Körner for source coding [16], in our distributed framework, and establish upper bounds on the achievable sum-rates reflecting the setting's communication requirements.

By extending, for the first time here, Körner's characteristic graph framework [16] to the new multi-server multi-function setting, we are able to reflect the nature of the functions and data statistics, in order to allow each server to build a codebook of encoding functions that determine the transmitted information. Each server, using its own codebook, can transmit a set of functions of the subfunctions of the data available in its storage, and to then provide the user with sufficient information for evaluating the demanded functions. The codebooks allow for a substantial reduction in the communication load.

The employed approach allows us to account for general dataset statistics, correlations, dataset placement, and function classes (see Theorem 2), thus yielding gains over the state of art [1], [17], as showcased for the case of linearly separable functions (see Proposition 1 — the celebrated result of Wan *et al.* [1]) in the presence of statistically skewed data, with correlations (e.g., see our extended work in [18, Proposition 2] for Boolean functions), as well as for the case of multilinear functions where the gains are particularly prominent, again under statistically skewed data. For this last case of multi-linear functions, we provide an upper bound on the achievable sum-rate (see Proposition 2, exploiting characteristic graphs), under a cyclic placement of data that reside in the binary field. We also provide a generalization of some elements in existing works on linearly separable functions [1], [19].

II. Technical Preliminary

In this section, we detail the notion of characteristic graphs devised by Körner [16] in the context of source compression, and the fundamental limits of functional compression, as demonstrated in by Alon and Orlitsky [12], Orlitsky and Roche [13], Feizi and Médard [20], and our prior work [14], [15].

A. Achievable compression rate for computation

Consider the canonical scenario with two servers storing X_1 and X_2. The user requests a bivariate function $F(X_1, X_2)$,

This research was partially supported by a Huawei France-funded Chair towards Future Wireless Networks, and "PEPR Networks of the Future" of France 2030. Co-funded by the European Union (ERC, SENSIBILITÉ, 101077361, and ERC-PoC, LIGHT, 101101031). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

which may be nonlinear. Let $G_{X_1} = (V_{G_{X_1}}, E_{G_{X_1}})$ be the characteristic graph for server one (similarly for G_{X_2}). Below, we explain how to construct G_{X_1}, where $V_{G_{X_1}}$ and $E_{G_{X_1}}$ are the set of vertices and edges, respectively. This concept extends to multivariate functions $F(X_\Omega)$ for $|\Omega| = N > 2$ [20]. A valid coloring of G_{X_1}, denoted by $c_{G_{X_1}}(X_1)$, assigns colors to vertices, where adjacent vertices have different colors.

Definition 1. *(Characteristic graph entropy [12], [16].) Given a random variable X_1 with characteristic graph $G_{X_1} = (V_{X_1},\ E_{X_1})$ and a distribution on its vertices $V_{G_{X_1}} = \mathcal{X}_1$, the entropy of the characteristic graph is expressed as*

$$H_{G_{X_1}}(X_1) = \min_{X_1 \in U_1 \in S(G_{X_1})} I(X_1; U_1)\ , \tag{1}$$

where $S(G_{X_1})$ is the set of all maximal independent sets (MISs) of G_{X_1}, where an MIS is not a subset of any other independent set, and an independent set is formed by a subset $\{x_1^l \in \mathcal{X}_1\}$ in which no two vertices are adjacent [12]. Notation $X_1 \in U_1 \in S(G_{X_1})$ means that the minimization is over all distributions $P_{U_1,X_1}(u_1, x_1)$ such that $P_{U_1,X_1}(u_1, x_1) > 0$ implies $x_1 \in u_1$, where U_1 is an MIS of G_{x_1}.

Orlitsky and Roche extended Körner's graph entropy to the conditional graph entropy for computing $F(X_1, X_2)$ [13]:

$$H_{G_{X_1}}(X_1 \,|\, X_2) = \min_{\substack{X_1 \in U_1 \in S(G_{X_1}) \\ U_1 - X_1 - X_2}} I(X_1; U_1 \,|\, X_2)\ , \tag{2}$$

where $U_1 - X_1 - X_2$ indicates a Markov chain, and $(x_1^1, x_1^2) \in E_{G_{X_1}}$ if $\exists$ an $x_2^1 \in \mathcal{X}_2$, such that i) $P_{X_1,X_2}(x_1^1, x_2^1) \cdot P_{X_1,X_2}(x_1^2, x_2^1) > 0$, and ii) $F(x_1^1, x_2^1) \neq F(x_1^2, x_2^1)$ [21].

Definition 2. *(Chromatic entropy [12].) The chromatic entropy of a graph G_{X_1} is defined as*

$$H^\chi_{G_{X_1}}(X_1) = \min_{c_{G_{X_1}}} H(c_{G_{X_1}}(X_1))\ . \tag{3}$$

Let $G^n_{\mathbf{X}_1} = (V^n_{X_1}, E^n_{X_1})$ be the n-th OR power of a graph G_{X_1} for the source sequence $\mathbf{X}_1$ to compress $F(\mathbf{X}_1,\ \mathbf{X}_2)$. In this OR power graph, $V^n_{X_1} = \mathcal{X}_1^n$ and $(\mathbf{x}_1^1, \mathbf{x}_1^2) \in E^n_{X_1}$, where $\mathbf{x}_1^1 = (x_{11}^1, x_{12}^1, \ldots, x_{1n}^1)$ and similarly for $\mathbf{x}_1^2$, when there exists at least one coordinate $l \in [n]$ such that $(x_{1l}^1, x_{1l}^2) \in E_{X_1}$. A valid coloring of $G^n_{\mathbf{X}_1}$ for computing $F(\mathbf{X}_1,\ \mathbf{X}_2)$ is denoted by $c_{G^n_{\mathbf{X}_1}}(\mathbf{X}_1)$. The encoding function at server one maps $\mathbf{X}_1$ to $c_{G^n_{\mathbf{X}_1}}(\mathbf{X}_1)$, specifying color classes of $\mathbf{X}_1$ where each class forms an independent set inducing the same function outcome. Using Definion 2, the chromatic entropy of $G^n_{\mathbf{X}_1}$ satisfies

$$H^\chi_{G^n_{\mathbf{X}_1}}(\mathbf{X}_1) = \min_{c_{G^n_{\mathbf{X}_1}}} H(c_{G^n_{\mathbf{X}_1}}(\mathbf{X}_1))\ . \tag{4}$$

In [16], Körner has shown the relation between the chromatic and graph entropies, which we detail next.

Theorem 1. *The characteristic graph entropy and the chromatic entropy of $G^n_{\mathbf{X}_1}$ satisfy the following relation [16]:*

$$H_{G_{X_1}}(X_1) = \lim_{n\to\infty} \frac{1}{n} H^\chi_{G^n_{\mathbf{X}_1}}(\mathbf{X}_1)\ . \tag{5}$$

Similarly, from (4) and (5), the conditional graph entropy of X_1 given X_2 optimizing over $\{c_{G^n_{\mathbf{X}_i}}\}_{i\in\{1,2\}}$ is given as

$$H_{G_{X_1}}(X_1 \,|\, X_2) = \lim_{n\to\infty} \min_{\{c_{G^n_{\mathbf{X}_i}}\}} \frac{1}{n} H(c_{G^n_{\mathbf{X}_1}}(\mathbf{X}_1) \,|\, c_{G^n_{\mathbf{X}_2}}(\mathbf{X}_2))\ .$$

B. An encoding framework for multi-function computing

The user demands a set of functions $\{F_j(X_\Omega)\}_{j\in[K_c]} \in \mathbb{R}^{K_c}$ that are possibly nonlinear in the subfunctions. In the case of $K_c > 1$ functions, let $G_{X_i,j} = (V_{X_i}, E_{X_i,j})$ be the characteristic graph that server $i \in \Omega$ builds for computing function $j \in [K_c]$. The graphs $\{G_{X_i,j}\}_{j\in[K_c]}$ are on the same vertex set. Note that $G_{X_i,j}$ for function $j \in [K_c]$ is devised independently from $G_{X_i,j'}$ for any $j' \neq j$. Hence, we can devise a *multi-functional characteristic graph*, which is indeed a union of individual graphs to simultaneously compute a set of functions, as defined in [20, Definition 45]. To that end, server $i \in \Omega$ creates a union of graphs on the same set of vertices V_{X_i} with a set of edges $E^\cup_{X_i}$, which satisfy

$$G^\cup_{X_i} = \bigcup_{j\in[K_c]} G_{X_i,j} = (V_{X_i}, E^\cup_{X_i})\ ,\ E^\cup_{X_i} = \bigcup_{j\in[K_c]} E_{X_i,j}\ . \tag{6}$$

In other words, we need to distinguish x_i^1 and x_i^2 of server X_i if there exists at least one function $F_j(x_\Omega)$, $j \in [K_c]$ out of K_c functions such that $F_j(x_i^1, x^1_{\Omega\setminus i}) \neq F_j(x_i^2, x^1_{\Omega\setminus i})$, for some $P_{X_\Omega}(x_i^1, x^1_{\Omega\setminus i}) \cdot P_{X_\Omega}(x_i^2, x^1_{\Omega\setminus i}) > 0$ given $x^1_{\Omega\setminus i} \in X_{\Omega\setminus i}$. The server then compresses the union $G^\cup_{X_i}$ by exploiting (4) and (5). Hence, exploiting $G^\cup_{X_i}$, we can attain the achievable rate of distributed lossless functional compression.

III. System Model

In the multi-server, multi-function distributed computation framework, the master has access to the set of all datasets, and distributes the datasets across the servers. The total number of servers is N, and each has a capacity of M. Communication from the master to the servers is allowed, whereas the servers are distributed and cannot collaborate. The user requests K_c functions that could be nonlinear. Given the dataset assignment to the servers, any subset of N_r servers is sufficient to compute the functions requested. We denote by $\mathcal{T}(N, K, K_c, M, N_r)$ the topology for the described multi-server multi-function distributed computing setting, which we detail in the following.

1) **Datasets, subfunctions, and placement:** There are K datasets in total, each denoted by D_k, $k \in \mathcal{K}$, where $K = |\mathcal{K}|$. Each distributed server $i \in \Omega = [N]$ with a capacity of M is assigned a subset of datasets with indices $\mathcal{Z}_i \subseteq \mathcal{K}$ such that $|\mathcal{Z}_i| = M$, where the assignments possibly overlap.

Each server $i \in \Omega$ computes a set of subfunctions $\{W_k = h_k(D_k),\ k \in \mathcal{Z}_i \subseteq \mathcal{K}\}$. Datasets $\{D_k\}_{k\in\mathcal{K}}$ could be dependent across $\mathcal{K}$, so could $\{W_k\}_{k\in\mathcal{K}}$. By exploiting the temporal and spatial variation or data dependence, it is possible to decrease the communication cost. We denote the number of symbols in each W_k by L, which equals the blocklength n. Let $X_i = \{W_k\}_{k\in\mathcal{Z}_i} = W_{\mathcal{Z}_i} = \{h_k(D_k)\}_{k\in\mathcal{Z}_i} \in \mathbb{F}_q^{|\mathcal{Z}_i|\times 1}$ denote the set of subfunctions of i-th server, where $\mathbb{F}_q$ is a finite field of characteristic q, $\mathcal{X}_i$ be the alphabet of X_i, and $X_\Omega = (X_1, X_2, \ldots, X_N)$ be the set of subfunctions of all servers.

We denote by boldface letters $\mathbf{W}_k = W_{k1}, W_{k2}, \ldots, W_{kn}$ and $\mathbf{X}_i = X_{i1}, X_{i2}, \ldots, X_{in} \in \mathbb{F}_q^{|\mathcal{Z}_i| \times n}$, the length n sequences of W_k, and of $X_i = W_{\mathcal{Z}_i}$ assigned to server $i \in \Omega$.

2) **Cyclic dataset placement model, computation capacity, and recovery threshold:** We assume that the total number of datasets K is divisible by the number of servers N, i.e., $\frac{K}{N} \doteq \Delta \in \mathbb{Z}^+$. The dataset placement on N distributed servers is conducted in a circular or cyclic manner, in the amount of Δ circular shifts between two consecutive servers, where the shifts are to the right and the final entries are moved to the first positions, if necessary. As a result of cyclic placement, any subset of N_r servers covers the set of all datasets to compute the requested functions from the user. Given $N_r \in [N]$, each server has a storage size or computation cost of $|\mathcal{Z}_i| = M = \Delta(N - N_r + 1)$, and the amount of dataset overlap between the consecutive servers is $\Delta(N - N_r)$. Hence, the set of indices assigned to server $i \in \Omega$ is given as follows:

$$\mathcal{Z}_i = \bigcup_{r=0}^{\Delta-1} \{ \mod \{i, N\} + rN, \quad \mod \{i+1, N\} + rN, \ldots, \quad \mod \{i + N - N_r, N\} + rN \}, \quad (7)$$

where $X_i = W_{\mathcal{Z}_i}$, $i \in \Omega$. As a result of (7), the cardinality of the datasets assigned to each server meets the storage capacity constraint M with equality, i.e., $|\mathcal{Z}_i| = M$, for all $i \in \Omega$.

3) **User demands and structure of the computation:** We address the problem of distributed lossless compression of a set general multi-variable functions $F_j(X_\Omega) : \mathcal{X}_1 \times \mathcal{X}_2 \cdots \times \mathcal{X}_N \to \mathbb{F}_q$, $j \in [K_c]$, requested by the user from the set of servers, where $K_c \geq 1$, and the functions are known by the servers and the user. More specifically, the user, from a subset of distributed servers, aims to compute in a lossless manner the following length n sequence as n tends to infinity:

$$F_j(\mathbf{X}_\Omega) = \{F_j(X_{1l}, X_{2l}, \ldots, X_{Nl})\}_{l=1}^n, \quad j \in [K_c] \ , \quad (8)$$

where $F_j(X_{1l}, X_{2l}, \ldots, X_{Nl})$ is the function outcome for the l-th realization $l \in [n]$, given the length n sequence. The representation in (8) is the most general form of a (conceivably non-linear) multi-variate function, which encompasses the special cases of separable and linearly separable functions.

The user seeks to compute functions that are separable to each dataset. Each demanded function $f_j(\cdot) \in \mathbb{R}$, $j \in [K_c]$ is a function of subfunctions $\{W_k\}_{k \in \mathcal{K}}$ such that $W_k = h_k(D_k) \in \mathbb{F}_q$, where h_k is a general (could be linear or nonlinear) function of dataset D_k. Hence, using $X_i = W_{\mathcal{Z}_i} = \{h_k(D_k)\}_{k \in \mathcal{Z}_i}$, each demanded function $j \in [K_c]$ can be written as

$$\begin{aligned} &f_j(W_\mathcal{K}) = f_j(h_1(D_1), \ldots, h_K(D_K)) \\ &= F_j(\{h_k(D_k)\}_{k \in \mathcal{Z}_1}, \ldots, \{h_k(D_k)\}_{k \in \mathcal{Z}_N}) = F_j(X_\Omega) \ . \end{aligned}$$

In the special case of linearly separable functions [1], the demanded functions take the form:

$$\{F_j(X_\Omega)\}_{j \in [K_c]} = \begin{bmatrix} F_1 & F_2 & \ldots & F_{K_c} \end{bmatrix}^\intercal = \mathbf{\Gamma W} \ , \quad (9)$$

where $\mathbf{W} = \begin{bmatrix} W_1 & W_2 & \ldots & W_K \end{bmatrix}^\intercal \in \mathbb{F}_q^{K \times 1}$ is the subfunction vector, and the coefficient matrix $\mathbf{\Gamma} = \{\gamma_{jk}\} \in \mathbb{F}_q^{K_c \times K}$ is known to the master node, servers, and the user. In other words, $\{F_j(X_\Omega)\}_{j \in [K_c]}$ is a set of linear maps from the subfunctions $\{W_k\}_k$, where $F_j(X_\Omega) = \sum_{k \in \mathcal{K}} \gamma_{jk} \cdot W_k$. Demanded functions are not restricted to be linearly separable, i.e., it may hold that $\{F_j(X_\Omega)\}_{j \in [K_c]} \neq \mathbf{\Gamma W}$.

4) **Communication cost for computing:** To compute $\{F_j(\mathbf{X}_\Omega)\}_{j \in [K_c]}$, each server $i \in \Omega$ constructs a characteristic graph, denoted by G_{X_i}, for compressing X_i. More specifically, for asymptotic lossless computation of the demanded functions, the server builds the n-th OR power $G_{\mathbf{X}_i}^n$ of G_{X_i} for compressing $\mathbf{X}_i$ to determine the transmitted information. The minimal possible code rate achievable to distinguish the edges of $G_{\mathbf{X}_i}^n$ as $n \to \infty$, is given the *Characteristic graph entropy*, $H_{G_{X_i}}(X_i)$. In this work, we solely focus on characterizing the total communication cost from all servers to the user, i.e., the achievable sum-rate, excluding the costs of master-server communication and computations at the servers/user.

Each $i \in \Omega$ builds a mapping from $\mathbf{X}_i$ to a valid coloring of $G_{\mathbf{X}_i}^n$, denoted by $c_{G_{\mathbf{X}_i}^n}(\mathbf{X}_i)$. The coloring $c_{G_{\mathbf{X}_i}^n}(\mathbf{X}_i)$ specifies the color classes of $\mathbf{X}_i$ that form independent sets to distinguish the demanded function outcomes. Given an encoding function g_i that models the transmission of server $i \in \Omega$ for computing $\{F_j(\mathbf{X}_\Omega)\}_{j \in [K_c]}$, we denote by $\mathbf{Z}_i = g_i(\mathbf{X}_i) = e_{X_i}(c_{G_{\mathbf{X}_i}^n}(\mathbf{X}_i))$ the color encoding performed by server $i \in \Omega$ for $\mathbf{X}_i$. Hence, the communication rate of server $i \in \Omega$, for a sufficiently large blocklength n, where T_i is the length for the color encoding performed at $i \in \Omega$, is

$$R_i = \frac{T_i}{L} = \frac{H(e_{X_i}(c_{G_{\mathbf{X}_i}^n}(\mathbf{X}_i)))}{n} \geq H_{G_{X_i}}(X_i) \ , \ i \in \Omega \ , \quad (10)$$

where the inequality follows from exploiting the achievability of $H_{G_{X_i}}(X_i) = \lim_{n \to \infty} \frac{1}{n} H^\chi_{G_{\mathbf{X}_i}^n}(\mathbf{X}_i)$, where $H^\chi_{G_{\mathbf{X}_i}^n}(\mathbf{X}_i)$ is the *chromatic entropy* of the graph $G_{\mathbf{X}_i}^n$ [12], [16].

Using the characteristic graph-based bound in (10), an achievable sum-rate for asymptotic lossless computation is

$$R_{\text{ach}} = \sum_{i \in \Omega} R_i \leq \sum_{i \in \Omega} H_{G_{X_i}}(X_i) \ .$$

We next provide our main results in Section IV.

IV. Main results

We now analyze the multi-server multi-function distributed computing framework using the characteristic graph-based approach from [16]. Unlike previous research, our method is general and accounts for (i) general input statistics or dataset distributions, including skewed data, (ii) correlations across datasets, (iii) any dataset placement model across servers, beyond the cyclic [1] or the Maddah-Ali and Niesen [22] placements, and (iv) general function classes requested by the user, beyond particular functions (see e.g., [1], [23]).

We next present Theorem 2, which addresses the achievable communication cost for the multi-server, multi-function topology. Theorem 2 applies to all input statistics, any dataset correlation model, and for distributed computing of all function classes requested by the user, regardless of data assignment

in server caches. The key element is Körner's characteristic graph [16].

Theorem 2. (Achievable sum-rate via characteristic graphs for general functions and distributions.) *For a given $\mathcal{T}(N,K,K_c,M,N_r)$, under general placement of datasets, general functions $\{f_j(W_\mathcal{K})\}_{j\in[K_c]}$ requested by the user, and general jointly distributed dataset models, including non-uniform inputs and allowing correlations across datasets, the achievable communication rate is upper bounded as follows:*

$$R_{\text{ach}} \leq \sum_{i=1}^{N_r} \min_{Z_i=g_i(X_i)\,:\,g_i\in\mathcal{C}_i} H_{G^{\cup}_{X_i}}(X_i)\ , \tag{11}$$

- *$G^{\cup}_{X_i}=\bigcup_{j\in[K_c]} G_{X_i,j}$ is the union characteristic graph that server $i\in\Omega$ builds for computing $\{f_j(W_\mathcal{K})\}_{j\in[K_c]}$,*
- *$\mathcal{C}_i \ni g_i$ denotes a codebook of functions of that server $i\in\Omega$ uses for computing $\{f_j(W_\mathcal{K})\}_{j\in[K_c]}$,*
- *each W_k, $k\in\mathcal{K}$ is defined over a q-ary field $\mathbb{F}_q$ such that the characteristic is at least 2, and*
- *$Z_i=g_i(X_i)$ s.t., g_i denotes the transmitted information.*

Proof. Consider the general topology, $\mathcal{T}(N,K,K_c,M,N_r)$, under general placement of datasets, and for a set of K_c general functions $\{f_j(W_\mathcal{K})\}_{j\in[K_c]}$ requested by the user, and under general jointly distributed dataset models, including non-uniform inputs and allowing correlations across datasets.

We note that server $i\in\Omega$ builds a *union characteristic graph* for distributed lossless computing $\{f_j(W_\mathcal{K})\}_{j\in[K_c]}$, which we denote by $G^{\cup}_{X_i}=(V_{X_i},E_{X_i})=\bigcup_{j\in[K_c]}G_{X_i,j}$, and is detailed in [20]. In the description of $G^{\cup}_{X_i}$, the set V_{X_i} is the support set of X_i, i.e., $V_{X_i}=\mathcal{X}_i$, and E_{X_i} is the union of edges, i.e., $E_{X_i}=\bigcup_{j\in[K_c]}E_{X_i,j}$, where $E_{X_i,j}$ denotes the set of edges in the characteristic graph $G_{X_i,j}$ that the server builds for distributed lossless computing $f_j(W_\mathcal{K})$, $j\in[K_c]$.

To compute the set of demanded functions $\{f_j(W_\mathcal{K})\}_{j\in[K_c]}$, we assume that server $i\in\Omega$ can use a codebook of functions denoted by $\mathcal{C}_i$ such that $\mathcal{C}_i\ni g_i$, where the user can compute its demanded functions using the set of transmitted information $\{g_i(X_i)\}_{i\in\mathcal{S}}$ provided from any set of $|\mathcal{S}|=N_r$ servers. More specifically, server $i\in\Omega$ chooses a function $g_i\in\mathcal{C}_i$ to encode X_i. Note that g_i represents the mapping from X_i to a valid coloring $c_{G_{X_i}}(X_i)$. We denote by $\mathbf{Z}_i=g_i(\mathbf{X}_i)=e_{X_i}(c_{G^n_{\mathbf{X}_i}}(\mathbf{X}_i))$ the color encoding performed by server $i\in\Omega$ for the length n realization $\mathbf{X}_i$. For convenience, as a result of this encoding, we denote the transmitted information from the server by $Z_i=g_i(X_i), i\in\Omega$. Combining the notions of the union graph defined in (6) and the transmission Z_i from $i\in\Omega$, the rate R_i needed from server i for meeting the user demand is upper bounded by the cost of the best encoding that minimizes the transmission rate of information from server i:

$$R_i \geq \min_{Z_i=g_i(X_i)\,:\,g_i\in\mathcal{C}_i} H_{G^{\cup}_{X_i}}(X_i)\ , \tag{12}$$

where equality is achievable in (12) adapting the relation $H_{G_{X_i}}(X_i)=\lim_{n\to\infty}\frac{1}{n}H_{G^{\chi}_{\mathbf{X}_i}}(\mathbf{X}_i)$ to union graphs. Because the user can recover the desired functions using any N_r servers, the achievable sum-rate is upper bounded by (11). □

Theorem 2 provides a general upper bound on the sum-rate for computing functions for general dataset statistics and correlations, and the placement model, and allows any function type, over a field of characteristic $q\geq 2$. In (11), the codebook $\mathcal{C}_i$ determines the structure of the union characteristic graph $G^{\cup}_{X_i}$, which, in turn, determines the distribution of Z_i. Therefore, the tightness of the rate upper bound relies essentially on the codebook selection. Since (11) is not analytically tractable, we will focus on specific instances of Theorem 2, to gain insights into how input statistics, dataset correlations, and special function classes affect the total communication cost.

We next show that the achievable sum rate for the *distributed linearly separable computation* framework given in [1, Theorem 2] is embedded by our characterization in Theorem 2.

Proposition 1. (Achievable sum-rate for linearly separable functions and i.i.d. subfunctions over $\mathbb{F}_q$.) *For a given $\mathcal{T}(N,K,K_c,M,N_r)$, under the cyclic placement of datasets, where $\frac{K}{N}=\Delta\in\mathbb{Z}^+$, and for a set of K_c linearly separable functions, given as in (9), requested by the user, and given i.i.d. uniformly distributed subfunctions over $\mathbb{F}_q$ where $q\geq 2$, the achievable communication rate satisfies*

$$R_{\text{ach}} \leq \begin{cases} \min\{K_c,\Delta\}N_r\ , & 1\leq K_c\leq \Delta N r\ , \\ \min\{K_c,K\}\ , & \Delta N r < K_c\ . \end{cases} \tag{13}$$

Proof. We here restrict the demand to be linearly separable in $W_\mathcal{K}$, given as in (9). Given N_r, it holds that

$$R_{\text{ach}} \leq \sum_{i=1}^{N_r} \min_{Z_i\,:\,g_i\in\mathcal{C}_i}\ \min_{X_i\in U_i\in S(G^{\cup}_{X_i})} I(X_i;U_i) \tag{14}$$

$$= \sum_{i=1}^{N_r} \Big[H(W^{(i-1)\Delta+M}_{(i-1)\Delta+1}) - H\Big(W^{(i-1)\Delta+M}_{(i-1)\Delta+1}\ \Big|\ Z_i\Big)\Big] \tag{15}$$

$$= \sum_{i=1}^{N_r} \Big[M-\Big(M-H(Z_i)\Big)\Big] = \sum_{i=1}^{N_r} H(Z_i)\ , \tag{16}$$

where (14) follows from (11) and using $H_{G^{\cup}_{X_i}}(X_i)=\min_{X_i\in U_i\in S(G^{\cup}_{X_i})} I(X_i;U_i)$ [13]. Furthermore, if the codebook $\mathcal{C}_i$ is restricted to linear combinations of $W_\mathcal{K}$, we then have

$$Z_i=g_i(X_i)=\Big\{\sum_{k=(i-1)\Delta+1}^{(i-1)\Delta+M}\alpha_k^{(l)}W_k\ ,\ l\in[K_c]\Big\}\ . \tag{17}$$

Server i builds a union graph $G^{\cup}_{X_i}$ for $\{f_j(W_\mathcal{K}),\ j\in[K_c]\}$. Each independent set $U_i\in S(G^{\cup}_{X_i})$ of $G^{\cup}_{X_i}$, with $S(G^{\cup}_{X_i})$ denoting the set of MISs of X_i, is captured by linear functions of $\{W_k\}_{k\in[(i-1)\Delta+1:(i-1)\Delta+M]}$, determined by (17). Hence, the user can recover the K_c functions by linearly combining the transmissions of the N_r servers:

$$f_j(W_\mathcal{K})=\sum_{i=1}^{N_r}\beta_{ji}Z_i=\sum_{k=1}^{K}\gamma_{jk}W_k\ ,\ j\in[K_c]\ . \tag{18}$$

In (15), we use $I(X_i;U_i) = H(X_i) - H(X_i \,|\, U_i)$, where given $i \in [N_r]$ and $\Delta = \frac{K}{N}$, it holds under cyclic placement that

$$X_i = W_{(i-1)\Delta+1}^{(i-1)\Delta+M} = W_{(i-1)\Delta+1}, \ldots, W_{(i-1)\Delta+M} \ ,$$

and $\alpha_k^{(l)}$ are the coefficients for computing function $l \in [K_c]$. In (16), we used that W_k is uniform over $\mathbb{F}_q$ and i.i.d. across $k \in \mathcal{K}$, to rewrite the conditional entropy expression as

$$H\Big(W_{(i-1)\Delta+1}^{(i-1)\Delta+M} \,\Big|\, Z_i\Big) \overset{(a)}{=} H\Big(W_{(i-1)\Delta+1}^{(i-1)\Delta+M}\Big) - H(Z_i) \ ,$$

where (a) follows from that Z_i is a function of $W_{(i-1)\Delta+1}^{(i-1)\Delta+M}$. For a given $l \in [K_c]$, the relation $\sum_{k=(i-1)\Delta+1}^{(i-1)\Delta+M} \alpha_k^{(l)} W_k$ ensures that G_{X_i} has q independent sets where each such set U_i contains q^{M-1} different values of X_i. Exploiting that W_k is i.i.d. and uniform over $\mathbb{F}_q$, each element of Z_i is uniform over $\mathbb{F}_q$. Hence, the achievable sum-rate is upper bounded by

$$\sum_{i=1}^{N_r} \min_{Z_i \,:\, g_i \in \mathcal{C}_i} H_{G_{X_i}^{\cup}}(X_i) \leq K_c N_r \ . \tag{19}$$

Exploiting the cyclic placement model, we can tighten the bound in (19). Note that server $i = 1$ can help recover M subfunctions, and each of servers $i \in [2 : N_r]$ can help recover an additional Δ subfunctions (at most, i.e., Δ transmissions needed to recover Δ subfunctions). Hence, the set of servers $[N_r]$ suffices to provide $M + (Nr - 1)\Delta = N\Delta = K$ subfunctions and reconstruct any desired function of $W_{\mathcal{K}}$. Due to cyclic placement, each W_k is stored in exactly $N - N_r + 1$ servers. Consider the following four scenarios:

(i) When $1 \leq K_c < \Delta$, each server transmits K_c linearly independent combinations of their subfunctions. This leads to resolving $K_c N_r$ linear combinations of K subfunctions from N_r servers that are sufficient to derive the demanded K_c linear functions. Because $K_c N_r < \Delta N_r$, there are $K - K_c N_r > \Delta(N - N_r) = M - \Delta$ unresolved linear combinations of $W_{\mathcal{K}}$.

(ii) When $\Delta \leq K_c \leq \Delta N_r$, it is sufficient for each server to transmit at most Δ linearly independent combinations of their subfunctions. This leads to resolving ΔN_r linear combinations of K subfunctions and $\Delta(N - N_r) = M - \Delta$ unresolved linear combinations of K subfunctions.

(iii) When $\Delta N_r < K_c \leq K$, each server needs to transmit at a rate $\frac{K_c}{N_r}$ where $\frac{K_c}{N_r} > \Delta$ and $\frac{K_c}{N_r} \leq \frac{K}{N_r} = \Delta\big(\frac{N_r + N - N_r}{N_r}\big) = \Delta + \Delta\big(\frac{N - N_r}{N_r}\big)$, which gives the number of linearly independent combinations needed to meet the demand. This yields a sum-rate of K_c. The subset of servers may need to provide up to an additional $\Delta(N - N_r)$ linear combinations, and $\Delta\big(\frac{N - N_r}{N_r}\big)$ is the maximum number of additional linear combinations per server, i.e., the required number of combinations when $K_c = K$.

(iv) When $K < K_c$, it is easy to note that any K linearly independent transmission, given as in (18), suffices to recover $W_{\mathcal{K}}$. Hence, the sum-rate K is achievable.

From (i)-(iv), we obtain the upper bound on the achievable sum-rate, given by (13), which matches the communication cost in [1, Theorem 2]. The i.i.d. distribution assumption for W_k, ensures that this result holds for any $q \geq 2$. □

Theorem 2 leads to Proposition 1 when three conditions hold: (i) the dataset placement across servers is cyclic following the rule in (7), (ii) the subfunctions $W_{\mathcal{K}}$ are i.i.d. and uniform over $\mathbb{F}_q$, and (iii) the codebook $\mathcal{C}_i$ is restricted to linear combinations of $W_{\mathcal{K}}$, which yields that the independent sets of $G_{X_i}^{\cup}$ satisfy a set of linear constraints in $\{W_k\}_{k \in \mathcal{Z}_i}$. The linear encoding and decoding approach for computing linearly separable functions, proposed by Wan *et al.* in [1, Theorem 2], is valid over a finite field of characteristic $q > 3$. However, our Proposition 1, through a graph-based approach and allowing for $q \geq 2$, generalizes [1, Theorem 2] to larger input alphabets.

To highlight the merits of the characteristic graph-based compression in *capturing dataset correlations* within the distributed computation framework, in the extended manuscript [18], we consider a special case of Theorem 2, where the user requests K_c arbitrary Boolean functions regardless of the data assignment. As shown in [18, Proposition 2], leveraging dataset skew, correlations within and across servers, and the structural properties of functions can substantially reduce the overall communication cost compared to the scenario in Proposition 1 for i.i.d. and uniformly distributed subfunctions (see also our Section V).

Despite the existing efforts, see e.g., [11]–[13], [24], to the best of our knowledge, for the given multi-server, multi-function distributed computing scenario, there is still no general encoding framework for computing general nonlinear functions. To that end, we exploit Theorem 2 to determine an upper bound on the achievable sum-rate R_{ach} for distributed computing of a multilinear function in the form of

$$f(W_{\mathcal{K}}) = \prod_{k \in \mathcal{K}} W_k \ , \tag{20}$$

which finds applications in distributed machine learning, sensor networks, and distributed optimization [25].

Drawing on the utility of characteristic graphs in capturing the structures of data and functions, input statistics and correlations, and the general result in Theorem 2, our next result, Proposition 2, exhibits a new upper bound on the achievable sum-rate for distributed computing of *multilinear functions*.

Proposition 2. (Achievable sum-rate for multilinear functions and i.i.d. subfunctions over $\mathbb{F}_2$.) *For a given $\mathcal{T}(N, K, K_c, M, N_r)$, under the cyclic placement of datasets, where $\frac{K}{N} = \Delta \in \mathbb{Z}^+$, and for computing the multilinear function ($K_c = 1$), given as in (20), requested by the user, and given i.i.d. distributed subfunctions $W_k \sim \text{Bern}(\epsilon)$, $k \in \mathcal{K}$, for some $\epsilon \in (0, 1)$, the achievable communication rate is upper bounded as follows:*

$$R_{\text{ach}} \leq \tfrac{1 - (\epsilon_M)^{N^*}}{1 - \epsilon_M} h(\epsilon_M) + (\epsilon_M)^{N^*} h\big(\epsilon_{\xi_N}\big) \cdot 1_{\Delta_N > 0} \ , \tag{21}$$

where

- *$\epsilon_M = \epsilon^M$ is the probability that the product of M subfunctions, with $W_k \sim \text{Bern}(\epsilon)$ being i.i.d. across $k \in \mathcal{K}$, taking the value one, i.e., $\mathbb{P}\Big(\prod_{k \in \mathcal{S} \,:\, |\mathcal{S}| = M} W_k\Big) = \epsilon_M$,*

- $N^* = \left\lfloor \frac{N}{N-N_r+1} \right\rfloor$ *denotes the minimum number of servers needed to compute* $f(W_{\mathcal{K}})$ *in (20), where each one computes a disjoint product of* M *subfunctions, and*
- $\Delta_N = N - N^* \cdot (N - N_r + 1)$ *represents if an additional server is needed, and if* $\Delta_N \geq 1$*, then* ξ_N *denotes the number of subfunctions to be computed by the additional server, and similarly,* $\mathbb{P}\Big(\prod_{k \in \mathcal{S} : |\mathcal{S}| = \xi_N} W_k \Big) = \epsilon_{\xi_N}$.

Proof. Recall that $W_k \sim \text{Bern}(\epsilon)$ are i.i.d. across $k \in \mathcal{K}$, and each server has a capacity $M = \Delta(N - N_r + 1)$. This means that given the number of datasets K, each server can compute the product of $\Delta(N-N_r+1)$ subfunctions and hence, the minimum number of servers to evaluate the multilinear function $f(W_{\mathcal{K}}) = \prod_{k \in \mathcal{K}} W_k$ is $N^* = \left\lfloor \frac{N}{N-N_r+1} \right\rfloor$ such that given its capacity $M = |\mathcal{Z}_i|$, each server can compute the product of a disjoint set of M subfunctions, i.e., $\prod_{k \in \mathcal{Z}_i} W_k$, which operates at a rate of $R_i \geq h(\epsilon_M)$, $i \in \Omega$. Exploiting the characteristic graph approach, we build $G_{X_1} = (V_{X_1}, E_{X_1})$ for X_1, with respect to $X_\Omega \backslash X_1 = X_2, \ldots, X_N$ and $f(W_{\mathcal{K}})$, and similarly for other servers, to characterize the sum-rate for the computation by evaluating the entropy of each graph.

To evaluate the first term in (21), we choose a total of N^* servers with a disjoint set of subfunctions. We denote the selected set of servers by $\mathcal{N}^* \subseteq \Omega$, and the collective computation rate of these N^* servers, as a function of the conditional graph entropies of these servers, becomes

$$\begin{aligned} \sum_{i \in \mathcal{N}^*} R_i &\overset{(a)}{\leq} H_{G_{X_{i_1}}}(X_{i_1}) + H_{G_{X_{i_2}}}(X_{i_2} \,|\, Z_{i_1}) + \ldots \\ &\quad + H_{G_{X_{i_{N^*}}}}(X_{i_{N^*}} \,|\, Z_{i_1}, Z_{i_2}, \ldots, Z_{i_{N^*-1}}) \\ &\overset{(b)}{=} h(\epsilon_M) + \epsilon_M h(\epsilon_M) + (\epsilon_M)^2 h(\epsilon_M) + \ldots \\ &\quad + (\epsilon_M)^{N^*-1} h(\epsilon_M) \overset{(c)}{=} \tfrac{1-(\epsilon_M)^{N^*}}{1-\epsilon_M} h(\epsilon_M) \ , \end{aligned} \tag{22}$$

where (a) follows from assuming $\mathcal{S} = \{i_1, i_2, \cdots, i_{N^*}\}$ with no loss of generality, and (b) from that the rate of server $i_l \in \mathcal{S}$ is positive only when $\prod_{i \in [i_{l-1}]} \prod_{k \in \mathcal{Z}_i} W_k = 1$, which is true with probability $(\epsilon_M)^{l-1}$. Finally, (c) follows from employing the geometric series, i.e., $\sum_{l=0}^{N^*-1} (\epsilon_M)^l = \frac{1-(\epsilon_M)^{N^*}}{1-\epsilon_M}$.

In the case of $\Delta_N = N - N^* \cdot (N - N_r + 1) > 0$, the product of K subfunctions cannot be determined by N^* servers and we need an additional (N^*+1)-th server $i^* \in \Omega$ to aid the computation and determine the outcome of $f(W_{\mathcal{K}})$ by computing the product of the remaining ξ_N subfunctions. In other words, if $\Delta_N > 0$ and $\prod_{i \in \mathcal{S}} \prod_{k \in \mathcal{Z}_i} W_k = 1$, the (N^*+1)-th server determines the outcome of $f(W_{\mathcal{K}})$ by computing the product of subfunctions $W_k \sim \text{Bern}(\epsilon)$, $k \in \big[N - \xi_N + 1 : N\big]$, that cannot be captured by the previous N^* servers. Hence, the additional rate, given by the second term in (21), is given by the product of the term

$$(\epsilon_M)^{N^*} = \mathbb{P}\Big(\prod_{i \in \mathcal{S}} \prod_{k \in \mathcal{Z}_i} W_k = 1 \Big) \ ,$$

with $1_{\Delta_N > 0}$, and $h\big(\epsilon_{\xi_N}\big)$. Combining this rate term with (22), we prove the statement of the proposition. □

We next detail several numerical examples to showcase the achievable gains in the total communication cost.

V. Numerical Evaluations

To gain insight into our analytical results and demonstrate the savings in the total communication cost, we next provide numerical examples for computing linearly separable functions (Sections V-A-V-B), and multilinear functions (Section V-C).

To that end, η_{lin} denotes the gain of the sum-rate for the graph entropy approach in [18, Eq. (9)] over the sum-rate derived by Wan *et al.* in [1], and rederived in (13). Furthermore, η_{SW} is the gain of the sum-rate for the graph entropy approach in (21) over the sum-rate of Slepian-Wolf [17]. To capture general statistics, i.e., dataset skewness and correlations, and make a fair comparison, we adapt the transmission model in [1] via modifying the i.i.d. dataset assumption.

In the following two subsections, we devise scenarios for distributed computing of a class of linearly separable functions given in (9) for general topologies, with general N, $K \geq 2$, M, N_r, K_c, over $\mathbb{F}_2$, where each demanded function takes the form $f_j(W_{\mathcal{K}}) = \sum_{k \in \mathcal{K}} \gamma_{jk} W_k \mod 2$, $j \in [K_c]$, under a specific correlation model across subfunctions, with the correlation coefficient denoted by ρ. Furthermore, we assume for $K_c > 1$ that $\mathbf{\Gamma} = \{\gamma_{jk}\} \in \mathbb{F}_2^{K_c \times K}$ is full rank. We assume an identical skew parameter $\epsilon \in [0, 1]$ for each subfunction, where $W_k \sim$Bern(ϵ), $k \in \mathcal{K}$, use cyclic placement as in (7), and incorporate the correlation between the subfunctions.

In [18], we provided an example for $K_c = 1$ for different topologies by capturing the correlations across the subfunctions and the servers as well as the structure of the linearly separable function while compressing, using the characteristic graph approach. We observed that the gain, η_{lin}, as a function of N, K, and N_r, grows faster compared to models not capturing correlations and the function structure, until η_{lin} reaches the maximum level attributed to full correlation.

A. For $K_c = 2$ (uncorrelated or correlated subfunctions).

To gain insights into the behavior of η_{lin}, we consider a simplified distributed computation model with $K = N = 3$, $N_r = 2$, where the subfunctions W_1, W_2, W_3 are stored in X_1, X_2 and X_3 in a cyclic manner, with $h(W_k) = \epsilon$, $k \in [3]$, and $K_c = 2$ with $f_1(W_{\mathcal{K}}) = W_2$, and $f_2(W_{\mathcal{K}}) = W_2 + W_3$.

For $N_r = 2$, using the characteristic graph approach for individual servers, the achievable sum-rate $R_{ach}(G)$ is the minimum total communication cost over transmission orderings. This involves compressing G_{X_i} by server i without side information and the conditional entropy of G_{X_j} for $j \in \Omega \backslash i$, using Z_i from server i as side information.

$$\begin{aligned} R_{ach}(G) = \min\{ & H_{G_{X_1}}(X_1) + H_{G_{X_2}}(X_2 \,|\, Z_1), \\ & H_{G_{X_2}}(X_2) + H_{G_{X_1}}(X_1 \,|\, Z_2) \} \ . \end{aligned} \tag{23}$$

In (23), the first term captures $G_{X_1} = (V_{X_1}, E_{X_1})$ where $V_{X_1} = \{0, 1\}^2$ is built based on using the support of W_1 and W_2, and the edges E_{X_1} are built based on distinguishing different W_2 values, hence requiring 2 colors. Similarly, server two constructs $G_{X_2} = (V_{X_2}, E_{X_2})$ given Z_1, where $V_{X_2} =$

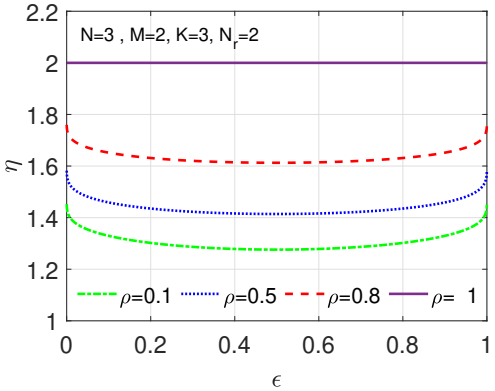

Fig. 1: η_{lin} versus ϵ for $K_c = 2$, $N_r = 2$, using the PMF in (25).

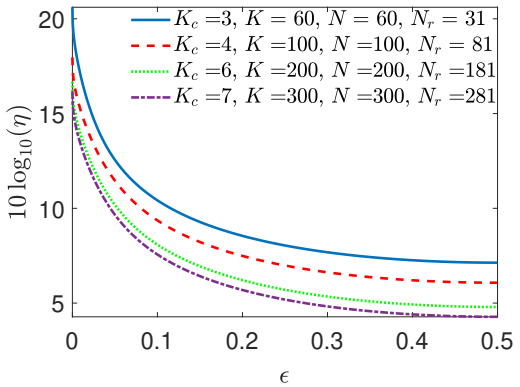

Fig. 2: $10\log_{10}(\eta_{lin})$ versus ϵ under $\rho = 0$, for $K_c > 2$.

$\{0,1\}^2$ using the support of W_2 and W_3, and Z_1 determines f_1, and hence, to compute f_2, the set E_{X_2} contains the pairs of V_{X_2} with different W_3 values, requiring 2 distinct colors. As a result, the first term yields a sum-rate of $h(\epsilon) + h(\epsilon) = 2h(\epsilon)$. Similarly, the second term of (23) captures $G_{X_2} = (V_{X_2}, E_{X_2})$, where server two builds G_{X_2} using the support of W_2, and W_3, and is a complete graph, requiring 4 different colors. Given Z_2, both f_1 and f_2 are deterministic. Hence, given Z_2, G_{X_1} has no edges, i.e., $H_{G_{X_1}}(X_1 \,|\, Z_2) = 0$. As a result, the second term of (23) yields the same sum-rate of $2h(\epsilon) + 0 = 2h(\epsilon)$. Hence, the minimum required rate is $R_{ach}(G) = 2h(\epsilon)$, with the second term achieving a lower recovery threshold of $N_r = 1$ versus the first term ($N_r = 2$).

Alternatively, in the linearly separable approach [1], N_r servers transmit the requested function of the datasets stored in their caches. For distributed computing of f_1 and f_2, servers one and two transmit at rates $H(W_2) = h(\epsilon)$, which suffices for computing f_1, and $H(W_2 + W_3)$, for function f_2, respectively. As a result, the achievable communication cost is given by $R_{ach}(lin) = h(\epsilon) + h(W_2 + W_3)$.

Under this setting, for $\rho = 0$, we present below the gain for computing f_1 and f_2 as a function of $\epsilon \in [0, 1]$:

$$\eta_{lin}(\epsilon) = \frac{h(\epsilon) + h(2\epsilon(1-e))}{2h(\epsilon)} \begin{cases} = 1, & \epsilon = \frac{1}{2} \ , \\ > 1, & \epsilon \neq \frac{1}{2} \ , \end{cases} \tag{24}$$

where $h(2\epsilon(1-\epsilon)) \geq h(\epsilon)$ due to the concavity of $h(\cdot)$, and η_{lin} approaches 1.5 as $\epsilon \to \{0, 1\}$.

We incorporate the correlation model from [26] for each W_k, identically distributed as $W_k \sim \text{Bern}(\epsilon)$, with correlation ρ across any two subfunctions, yielding the following probability mass function (PMF) for $f(W_{\mathcal{K}})$:

$$\mathbb{P}(f(W_K) = y) = \binom{K}{y} \epsilon^y (1-\epsilon)^{K-y} (1-\rho) \cdot 1_{y \in A_1} + \epsilon^{\frac{y}{K}} (1-\epsilon)^{\frac{K-y}{K}} \rho \cdot 1_{y \in A_2} \ , \ y \in \{0, \cdots, K\} \ , \tag{25}$$

where $1_{y \in A_1}$ and $1_{y \in A_2}$ are indicator functions, where $A_1 = \{0, 1, \cdots, K\}$ and $A_2 = \{0, K\}$.

We next use the joint PMF in (25), where we observe that $f_2 \sim ((1-\epsilon)^2(1-\rho)+(1-\epsilon)\rho,\ 2\epsilon(1-\epsilon)(1-\rho),\ \epsilon^2(1-\rho)+\epsilon\rho)$. This PMF results in

$$\eta_{lin} = \frac{h(\epsilon) + H(f_2)}{h(\epsilon) + (1-\epsilon)h(\zeta_1) + \epsilon h(\zeta_2)} \ , \tag{26}$$

where $\zeta_1 = (1-\epsilon)(1-\rho) + \rho$, and $\zeta_2 = (1-\epsilon)(1-\rho)$. For this model, we illustrate η_{lin} versus ϵ in Figure 1, for different ρ values. Evaluating (26), the peak achievable gain is attained when $\rho = 1$ at $f_2 \sim ((1-\epsilon),\ 0,\ \epsilon)$, yielding $H(W_2 + W_3) = h(\epsilon)$ and $H(W_3 \,|\, W_2) = (1-\epsilon)h(\rho) = 0$, and hence, $\eta_{lin} = N_r = 2$, as shown by the purple (solid) curve. On the other hand, for $\rho = 0$, we observe that $f_2 \sim ((1-\epsilon)^2,\ 2\epsilon(1-\epsilon),\ \epsilon^2)$, yielding $H(W_2+W_3) = h((1-\epsilon)^2,\ 2\epsilon(1-\epsilon),\ \epsilon^2) = h(2\epsilon(1-\epsilon)) + ((1-\epsilon)^2 + \epsilon^2)h\left(\frac{\epsilon^2}{\epsilon^2+(1-\epsilon)^2}\right)$ and $H(W_3 \,|\, W_2) = (1-\epsilon)h(\epsilon) + \epsilon h(\epsilon) = h(\epsilon)$, and it can be shown that $\eta_{lin} \geq 1.25$.

B. For $K_c \in [N_r]$, where the subfunctions are uncorrelated.

Here, we generalize the example in Section V-A assuming that $K = N$, and $\rho = 0$. For the linearly separable model in (9), exploiting [18, Proposition 2], which is an application of Theorem 2 to subfunctions over $\mathbb{F}_2$, the sum-rate for distributed lossless computing of $f(W_{\mathcal{K}})$ is derived as

$$\sum_{i \in \Omega} R_i \leq N^* \cdot h(\epsilon_M) + 1_{\Delta_N > 0} \cdot h(\epsilon_{\xi_N}) \ . \tag{27}$$

The communication cost for general K_c is expressed as:

$$R_{ach}(lin) = N_r \cdot h(\epsilon_M), \tag{28}$$

which as $\epsilon \to \{0,\ 1\}$ yields that $h(\epsilon_M) \to 0$. Subsequently, the achievable communication cost for the characteristic graph model is calculated as

$$R_{ach}(G) = K_c \cdot N^* \cdot h(\epsilon) \ . \tag{29}$$

To understand the behavior of $\eta_{lin} = \frac{N_r}{K_c N^*} \cdot \frac{h(\epsilon_M)}{h(\epsilon)}$, for a given value of $\frac{N_r}{K_c N^*}$, we need to examine the dynamic component $\frac{h(\epsilon_M)}{h(\epsilon)}$. Exploiting Schur concavity of the binary entropy function, i.e., $h(\mathbb{E}[X]) \geq \mathbb{E}[h(X)]$, we can establish an upper bound on this component, in the following manner, as $\epsilon \to \{0,\ 1\}$, where the inequality between the left and right-hand sides becomes loose as M increases:

$$\lim_{\epsilon \to \{0,\ 1\}} \frac{h(\epsilon_M)}{h(\epsilon)} \leq M \ , \quad M \in \mathbb{Z}^+ \ , \tag{30}$$

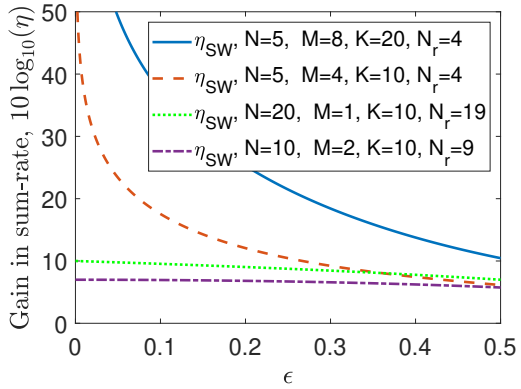

Fig. 3: Gain $10\log_{10}(\eta_{SW})$ versus ϵ for $K_c = 1$, $\rho = 0$, $N_r = N-1$, for computing (20) in Section V-C.

and as $\epsilon \to \{0, 1\}$, η_{lin} approximates the product of $\frac{N_r}{K_c \cdot N^*}$ and M. We illustrate the upper bound on η_{lin} in Figure 2, and demonstrate the η_{lin} behavior for various $K = N$, M, and K_c and for $\rho = 0$ across various topologies for $\epsilon \leq 1/2$. Due to the symmetry of η_{lin} inherited from the entropy function, we only plot for $\epsilon \leq 1/2$. The multiplicative coefficient $\frac{N_r}{K_c N^*}$ of η_{lin} determines the growth, which is depicted by the curves.

For a given topology, for $\rho = 0$, using (30), η_{lin} exponentially grows with decreasing $\epsilon \in [0, 1/2]$ (or increasing $\epsilon \in [1/2, 1]$ observing that the upper bound for η_{lin} is symmetric around $\epsilon = 1/2$), and a $100\times$ reduction in the total communication cost is possible as ϵ approaches $\{0, 1\}$ as shown in Figure 2 by the blue (solid) curve. The gain over [1, Theorem 2], η_{lin}, for a given topology, changes proportionally to $\frac{N_r}{K_c N^*}$. The gain over [17], η_{SW}, for $\rho = 0$ linearly scales with $\frac{K}{K_c N^*}$. For instance, the gain for the blue (solid) curve in Figure 2 is $\eta_{SW} = 10$. In general, other functions in $\mathbb{F}_2$, such as bitwise AND and the multilinear function (see e.g., Proposition 2), are more skewed than linearly separable functions. Therefore, the cost in (29) can serve as an upper bound for the communication costs of those functions.

C. Distributed Computing of K-Multi-Linear Functions

Proposition 2 illustrates how the characteristic graph approach helps decrease the communication cost for distributed computing of multilinear functions, given as in (20), compared to recovering the local computations $\prod_{k \in \mathcal{S}: |\mathcal{S}|=M} W_k$ using [17]. We next showcase the achievable gains η_{SW} of Proposition 2 (where the functions are over $\mathbb{F}_2$).

We study the behaviors of η_{SW} versus the skewness parameter ϵ for computing the multilinear function given in (20) for i.i.d. uniform $W_k \sim \text{Bern}(\epsilon)$, $\epsilon \in [0, 1/2]$ across $k \in \mathcal{K}$, and for a given $\mathcal{T}(N, K, K_c, M, N_r)$ with parameters N, K, $M = \Delta(N - N_r + 1)$, such that $N_r = N - 1$, $K_c = 1$, $\rho = 0$, and the number of replicates per dataset is $\frac{MN}{K} = 2$. We use Proposition 2 to determine the sum-rate upper bound, and illustrate the gains $10\log_{10}(\eta_{SW})$ versus ϵ in Figure 3. From Figure 3, we observe that the sum-rate gain of the graph entropy-based approach versus the fully distributed approach of [17], η_{SW}, could reach up to more than 10-fold gain in compression rate for uniform and up to 10^6-fold for skewed data. The results on η_{SW} showcase that our proposed scheme can guarantee an exponential rate reduction over [17] as a function of decreasing ϵ. Furthermore, the sum-rate gains scale linearly with the cache size M, which scales with K given $N_r = N - 1$. Note that η_{SW} diminishes with increasing N when M and Δ are kept fixed. For additional multi-linear function requests when $M \ll K$, a fixed total cache size MN and, consequently, a fixed K are discussed in [18].

ACKNOWLEDGMENT

The authors thank Prof. Kai Wan at HUST Wuhan, China.

REFERENCES

[1] K. Wan *et al.*, "Distributed linearly separable computation," *IEEE Trans. Inf. Theory*, vol. 68, no. 2, pp. 1259–1278, Nov. 2021.
[2] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.
[3] A. Mitra *et al.*, "Distributed inference with sparse and quantized communication," *IEEE Trans. Signal Process.*, vol. 69, pp. 3906–21, Jun. 2021.
[4] Z. Jia and S. A. Jafar, "On the capacity of secure distributed batch matrix multiplication," *IEEE Trans. Inf. Theory*, vol. 67, pp. 20–37, Sep. 2021.
[5] W.-T. Chang and R. Tandon, "On the capacity of secure distributed matrix multiplication," in *Proc., IEEE Global Commun. Conf.*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6.
[6] R. G. D'Oliveira *et al.*, "Notes on communication and computation in secure distributed matrix multiplication," in *Proc., IEEE Conf. Commun. and Netw. Secur.*, virtual, Jun. 2020, pp. 1–6.
[7] Q. Yu *et al.*, "Straggler mitigation in distributed matrix multiplication: Fundamental limits and optimal coding," *IEEE Trans. Inf. Theory.*, vol. 66, no. 3, pp. 1920–1933, Jan. 2020.
[8] M. Monagan and R. Pearce, "Parallel sparse polynomial multiplication using heaps," in *Proc. Int. Symp. Symb. Algebr.*, New York, NY, USA, Jun. 2009, pp. 263–270.
[9] C. D. Hsu *et al.*, "Scalable reinforcement learning policies for multi-agent control," in *Proc., IEEE Int. Conf. Intell. Robots Syst.*, Prague, Czech, Sep. 2021, pp. 4785–4791.
[10] M. Goldenbaum *et al.*, "Nomographic functions: Efficient computation in clustered gaussian sensor networks," *IEEE Trans. Wirel. Commun.*, vol. 14, no. 4, pp. 2093–2105, Dec. 2014.
[11] J. Korner and K. Marton, "How to encode the modulo-two sum of binary sources (corresp.)," *IEEE Trans. Inf. Theory*, vol. 25, no. 2, pp. 219–221, Mar. 1979.
[12] N. Alon and A. Orlitsky, "Source coding and graph entropies," *IEEE Trans. Inf. Theory*, vol. 42, no. 5, pp. 1329–39, Sep. 1996.
[13] A. Orlitsky and J. R. Roche, "Coding for computing," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 903–17, Mar. 2001.
[14] D. Malak, "Weighted graph coloring for quantized computing," in *Proc., IEEE Int. Symp. Inf. Theory*, Taipei, Taiwan, Jun. 2023, pp. 2290–2295.
[15] M. R. Deylam Salehi and D. Malak, "An achievable low complexity encoding scheme for coloring cyclic graphs," in *Proc., Annu. Allerton Conf. Commun. Control Comput.*, Monticello, IL, USA, Sep. 2023.
[16] J. Körner, "Coding of an information source having ambiguous alphabet and the entropy of graphs," in *Proc., Prague Conf. Inf. Theory*, Prague, Czech, Sep. 1973.
[17] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.
[18] D. Malak *et al.*, "Multi-server multi-function distributed computation," *Entropy*, vol. 26, no. 6, p. 448, Jun. 2024.
[19] W. Huang *et al.*, "Fundamental limits of distributed linearly separable computation under cyclic assignment," in *Proc., IEEE Int. Symp. Inf. Theory*, Taipei, Taiwan, Jun. 2023.
[20] S. Feizi and M. Médard, "On network functional compression," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5387–401, Sep. 2014.
[21] H. Witsenhausen, "The zero-error side information problem and chromatic numbers (corresp.)." *IEEE Trans. Inf. Theory*, vol. 22, no. 5, pp. 592–593, Sep. 1976.
[22] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–67, May 2014.
[23] A. Khalesi and P. Elia, "Multi-user linearly-separable distributed computing," *IEEE. Trans. Inf. Theory*, vol. 69, pp. 6314–39, Jun. 2023.
[24] T. Han and K. Kobayashi, "A dichotomy of functions F(X, Y) of correlated sources (X, Y)," *IEEE Trans. Inf. Theory*, vol. 33, no. 1, pp. 69–76, Jan. 1987.
[25] G. Kaur, "A comparison of two hybrid ensemble techniques for network anomaly detection in spark distributed environment," *J. Inform. Secur. Appl.*, vol. 55, no. 102601, Dec. 2020.
[26] C. A. R. Diniz *et al.*, "Bayesian analysis of a correlated binomial model," *Braz. J. Probab. Stat.*, vol. 24, no. 1, p. 68–77, Mar. 2010.