

# Multi-Server Multi-Function Distributed Computation

Derya Malak <sup>\*</sup>, Mohammad Reza Deylam Salehi , Berksan Serbetci <sup>†</sup> and Petros Elia 

Communication Systems Department, EURECOM, Sophia Antipolis, 06140 Biot, France; deylam@eurecom.fr (M.R.D.S.); fberks@gmail.com (B.S.); elia@eurecom.fr (P.E.)

\* Correspondence: derya.malak@eurecom.fr

<sup>†</sup> This work was conducted when B. Serbetci was a Postdoctoral Researcher at EURECOM.

**Abstract:** The work here studies the communication cost for a multi-server multi-task distributed computation framework, as well as 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 the 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 the case of multi-linear functions.

**Keywords:** distributed computation; linearly separable functions; non-linear functions; functional compression; characteristic graph entropy; multi-server; multi-function; skewed statistics; data correlations



**Citation:** Malak, D.; Deylam Salehi, M.R.; Serbetci, B.; Elia, P. Multi-Server Multi-Function Distributed Computation. *Entropy* **2024**, *26*, 448. <https://doi.org/10.3390/e26060448>

Academic Editors: Chintha Tellambura and Jun Chen

Received: 31 March 2024

Revised: 14 May 2024

Accepted: 23 May 2024

Published: 26 May 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Distributed computing plays an increasingly significant role in accelerating the execution of computationally challenging and complex computational tasks. This growth in influence is rooted in the innate capability of distributed computing to parallelize computational loads across multiple servers. This same parallelization renders distributed computing as an indispensable tool for addressing a wide array of complex computational challenges, spanning scientific simulations, and extracting various spatial data distributions [1], data-intensive analyses for cloud computing [2], and machine learning [3], as well as applications in various other fields such as computational fluid dynamics [4], high-quality graphics for movie and game rendering [5], and a variety of medical applications [6], to name just a few. In the center of this ever-increasing presence of parallelized computing stand modern parallel processing techniques, such as MapReduce [7–9] and Spark [10,11].

However, for distributed computing to achieve the desirable parallelization effect, there is an undeniable need for massive information exchange to and from the various network nodes. Reducing this communication load is essential for scalability [12–15] in various topologies [16–18]. Central to the effort to reduce communication costs stand coding techniques such as those found in [19–36], including gradient coding [21] and different variants of coded distributed computing that nicely yield gains in reliability, scalability, computation speed, and cost-effectiveness [24]. Similar communication-load aspects are often addressed via polynomial codes [37], which can mitigate stragglers and enhance the recovery threshold, while MatDot codes, devised in [31,38] for secure distributed matrix multiplication, can decrease the number of transmissions for distributed matrix multiplication. This same emphasis on reducing communication costs is even more prominent in works like [31,34,35,38–46], which again, focus on distributed matrix multiplication.

For example, focusing on a cyclic dataset placement model, the work in [39] provided useful achievability results, while the authors of [35] have characterized achievability and converse bounds for secure distributed matrix multiplication. Furthermore, the work in [34] found creative methods to exploit the correlation between the entries of the matrix product in order to reduce the cost of communication.

### *1.1. The Multi-Server Multi-Function Distributed Computing Setting and the Need for Accounting for General Non-Linear Functions*

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 for the datasets. This multi-server multi-function setting was recently studied by Wan et al. in [39] for the class of linearly separable functions, which nicely captures a wide range of real-world tasks [7] such as convolution [41], the discrete Fourier transform [47], 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, however, there is growing need to consider more general classes of functions, including non-linear functions, such as is often the case with subfunctions that produce intermediate values in MapReduce operations [7] or that relate to quantization [48], classification [49], and optimization [50]. Intense interest can also be identified in the aforementioned problem of distributed matrix multiplication, which has been explored in a plethora of works, which include [35,42,45,51–53], with a diverse focus that entails secrecy [45,51,53], as well as precision and stragglers [14,35,42,52], to name a few. In addition to matrix multiplication, other important non-linear function classes include sparse polynomial multiplication [54], permutation invariant functions [55]—which often appear in multi-agent settings and have applications in learning, combinatorics, and graph neural networks—as well as nomographic functions [56,57], which can appear in the context of sensor networks and which have strong connections with interference exploitation and lattice codes, as nicely revealed in [56,57].

Our own work here is indeed motivated by this emerging need for distributed computing of non-linear functions, and our goal is to now consider general functions in the context of the multi-server multi-function distributed computing framework while also capturing dataset statistics and correlations and while exploiting the structural properties of the (possibly non-linear) functions requested by the user. For this purpose, we go beyond the linear coding approaches in [39,58,59] and devise demand-based encoding–decoding solutions. Furthermore, we adopt—in the context of the multi-server multi-function framework—the powerful tools from characteristic graphs that are specifically geared toward capturing both the statistical structure of the data as well as the properties of functions beyond the linear case. To help the reader better understand our motivation and contribution, we proceed with a brief discussion on data structure and characteristic graphs.

### *1.2. Data Correlation and Structure*

Crucial in reducing the communication bottleneck of distributed computing is an ability to capture the structure that appears in modern datasets. Indeed, even before computing considerations come into play, capturing the general structure of the data has been crucial in reducing the communication load in various scenarios such as those in the seminal work by Slepian–Wolf [60] and Cover [61]. Similarly, when function computation is introduced, data structure can be a key component. In the context of computing, we have seen the seminal work by Körner and Marton [62], which focused on efficient compression of the modulo 2 sum of two statistically dependent sources, while Lalitha et al. [63] explored linear combinations of multiple statistically dependent sources. Furthermore, for general

bivariate functions of correlated sources, when one of the sources is available as side information, the work of Yamamoto [64] generalized the pioneering work of Wyner and Ziv [65] to provide a rate-distortion characterization for the function computation setting.

It is the case, however, that when the computational model becomes more involved—as is the case in our multi-server multi-function scenario here—the data may often be treated as unstructured and independent [39,58,66–68]. This naturally allows for crucial analytical tractability, but it may often ignore the potential benefits of accounting for statistical skews and correlations in data when aiming to reduce communication costs in distributed computing. Furthermore, this comes at a time when more and more function computation settings—such as in medical imaging analysis [69], data fusion, and group inferences [70], as well as predictive modeling for artificial intelligence [71]—entail datasets with prominent dependencies and correlations. While various works, such as those by Körner–Marton [62], Han–Kobayashi [72], Yamamoto [64], Alon–Orlitsky [73], and Orlitsky–Roche [74], provide crucial breakthroughs in exploiting data structure, to the best of our knowledge, in the context of fully distributed function computation, the structure in functions and data has yet to be considered simultaneously.

### 1.3. Characteristic Graphs

To jointly account for this structure in both data and functions, we draw from the powerful literature on characteristic graphs, introduced by Körner for source coding [75] and used in data compression [62,73,74,76–78], cryptography [79], image processing [80], and bioinformatics [81]. For example, toward understanding the fundamental limits of distributed functional compression, the work in [75] devised the graph entropy approach in order to provide the best possible encoding rate of an information source with vanishing error probability. This same approach, while capturing both function structure and source structure, was presented for the case of one source, and it is not directly applicable to the distributed computing setting. Similarly, the zero-error side information setting in [73] and the lossy encoding setting in [64,74] use Körner’s graph entropy [75] approach to capture both function structure and source structure but were again presented for the case of one source. A similar focus can be found in the works in [73,74,76,77,79]. The same characteristic graph approach nicely used by Feizi and Médard in [82] for the distributed computing setting, for a simple distributed computing framework, and in the absence of considerations for the data structure.

Characteristic graphs, which are used in fully distributed architectures to compress information, can allow us to capture various data statistics and correlations, various data placement arrangements, and various function types. This versatility motivates us to employ characteristic graphs in our multi-server multi-function architecture for distributed computing of non-linear functions.

### 1.4. Contributions

In this paper, leveraging fundamental principles from source and functional compression, as well as graph theory, we study a general multi-server multi-function distributed computing framework composed of a single user requesting a set of functions, which are computed with the assistance of distributed servers that have partial access to the datasets. To achieve our goal, we consider the use of Körner’s characteristic graph framework [75] in our multi-server multi-function setting and proceed to 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 [75] 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 function (or 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, thus yielding gains over the state of the art [39,60], as showcased in our examples for the case of linearly separable functions in the presence of statistically skewed data, as well as for the case of multi-linear 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 Section 4.2) under a cyclic placement of the data that reside in the binary field. We also provide a generalization of some elements in the existing works on linearly separable functions [39,58].

In the end, our work demonstrates the power of using characteristic-graph-based encoding for exploiting the structural properties of functions and data in distributed computing, as well as provides insights into fundamental compression limits, all for the broad scenario of multi-server multi-function distributed computation.

### 1.5. Paper Organization

The rest of this paper is structured as follows. Section 2 describes the system model for the multi-server multi-function architecture, and Section 3 details the main results on the communication cost or sum-rate bounds under the general dataset distributions and correlations, dataset placement models, and general function classes requested by the user over a field of characteristic  $q \geq 2$ , through employing the characteristic graph approach, and contrasts the sum-rate with the relevant prior works, e.g., [39,60]. Finally, we summarize our key results and outline possible future directions in Section 5. We provide a primer for the key definitions and results on characteristic graphs and their fundamental compression limits in Appendix A and give proofs of our main results in Appendix B.

Notation: We denote by  $H(X) = \mathbb{E}[-\log P_X(X)]$  the Shannon entropy of random variable  $X$  drawn from distribution or probability mass function (PMF)  $P_X$ . Let  $P_{X_1, X_2}$  be the joint PMF of two random variables  $X_1$  and  $X_2$ , where  $X_1$  and  $X_2$  are not necessarily independent and identically distributed (i.i.d.), i.e., equivalently, the joint PMF is not in product form. The notation  $X \sim \text{Bern}(\epsilon)$  denotes that  $X$  is Bernoulli distributed with parameter  $\epsilon \in [0, 1]$ . Let  $h(\cdot)$  denote the binary entropy function and  $H_B(B(n, \epsilon))$  denote the entropy of a binomial random variable of size  $n \in \mathbb{N}$ , with  $\epsilon \in [0, 1]$  modeling the success probability of each Boolean-valued outcome. The notation  $X_{\mathcal{S}} = \{X_i : i \in \mathcal{S}\}$  denotes a subset of servers with indices  $i \in \mathcal{S}$  for  $\mathcal{S} \subseteq \Omega$ . The notation  $\mathcal{S}^c = \Omega \setminus \mathcal{S}$  denotes the complement of  $\mathcal{S}$ . We denote the probability of an event  $A$  by  $\mathbb{P}(A)$ . The notation  $1_{x \in A}$  denotes the indicator function, which takes the value 1 if  $x \in A$  and 0 otherwise. The notation  $G_{X_i}$  denotes the characteristic graph that server  $i \in \Omega$  builds for computing  $F(X_{\Omega})$ . The measures  $H_{G_X}(X)$  and  $H_{G_X}(X | Y)$  denote the entropy of characteristic graph  $G_X$  and the conditional graph entropy for random variable  $X$  given  $Y$ , respectively. The notation  $\mathcal{T}(N, K, K_c, M, N_r)$  shows the topology of the distributed system. We note that  $\mathcal{Z}_i$  denotes the indices of datasets stored in  $i \in \Omega$ , and the notation  $K_n(\mathcal{S}) = |\mathcal{Z}_{\mathcal{S}}| = |\bigcup_{i \in \mathcal{S}} \mathcal{Z}_i|$  represents the cardinality of the datasets in the union of the sets in  $\mathcal{S}$  for a given subset  $\mathcal{S} \subseteq \Omega$  of servers. We also note that  $[N] = \{1, 2, \dots, N\}$ ,  $N \in \mathbb{Z}^+$ , and  $[a : b] = \{a, a + 1, \dots, b\}$  for  $a, b \in \mathbb{Z}^+$  such that  $a < b$ . We use the convention  $\text{mod } \{b, a\} = a$  if  $a$  divides  $b$ . We provide the notation in Table 1.

Table 1. Notation.

Distributed-Computation-System-Related Definitions	Symbols
Number of distributed servers; set of distributed servers; capacity of a server	$N; \Omega; M$
Set of datasets; dataset catalog size	$\{D_k\}_{k \in [K]}; K =  \mathcal{K} $
Subfunction $k \in \mathcal{Z}_i \subseteq [K]$	$W_k = h_k(D_k)$
The number of symbols in each $W_k$ ; blocklength	$L; n$
Set of indices of datasets assigned to server $i \in \Omega$ such that $ \mathcal{Z}_i  \leq M$	$\mathcal{Z}_i \subseteq [K]$
Set of subfunctions corresponding to a subset of servers with indices $i \in \mathcal{S}$ for $\mathcal{S} \subseteq \Omega$	$X_{\mathcal{S}} = \{X_i : i \in \mathcal{S}\}$
Recovery threshold	$N_r$
Number of demanded functions by the user	$K_c$
Number of symbols per transmission of server $i \in \Omega$	$T_i$
Topology of the multi-server multi-function distributed computing setting	$\mathcal{T}(N, K, K_c, M, N_r)$
Graph-Theoretic Definitions	Symbols
Characteristic graph that server $i$ builds for computing $F(X_{\Omega})$	$G_{X_i}, i \in \Omega$
Union of characteristic graphs that server $i$ builds for computing $\{F_j(X_{\Omega})\}_{j \in [K_c]}$	$G_{X_i}^{\cup}, i \in \Omega$
Maximal independent set (MIS); set of all MISs of $i \in \Omega$	$U_1; S(G_{X_1})$
A valid coloring of $G_{X_i}$	$c_{G_{X_i}}$
$n$ -th OR power graph; a valid coloring of the $n$ -th OR power graph	$G_{X_i}^n; c_{G_{X_i}^n}(X_i)$
Characteristic graph entropy of $X_i$	$H_{G_{X_i}}(X_i)$
Conditional characteristic graph entropy of $X_i$ such that $i \in \mathcal{S}$ given $X_{\mathcal{S}^c}$	$H_{G_{X_i}}(X_i   X_{\mathcal{S}^c})$

## 2. System Model

This section outlines our multi-server multi-function architecture and details our main technical contributions, namely, the communication cost for the problem of distributed computing of general non-linear functions and the cost for special instances of the computation problem under some simplifying assumptions on the dataset statistics, dataset correlations, placement, and the structures of functions.

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 server 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 non-linear. 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.

### 2.1. Datasets, Subfunctions, and Placement into Distributed Servers

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

Each server computes a set of subfunctions  $W_k = h_k(D_k)$  for  $k \in \mathcal{Z}_i \subseteq [K], i \in \Omega$ . Datasets  $\{D_k\}_{k \in [K]}$  could be dependent (We note that by exploiting the temporal and spatial variation or dependence of data, it is possible to decrease the communication cost.) across  $\mathcal{K}$ , so could  $\{W_k\}_{k \in [K]}$ . We denote the number of symbols in each  $W_k$  by  $L$ , which equals the blocklength  $n$ . Let  $X_i = \{W_k\}_{k \in \mathcal{Z}_i} = W_{\mathcal{Z}_i} = \{h_k(D_k)\}_{k \in \mathcal{Z}_i}$  denote the set of subfunctions of the  $i$ -th server,  $\mathcal{X}_i$  be the alphabet of  $X_i$ , and  $X_{\Omega} = (X_1, X_2, \dots, X_N)$  be the set of subfunctions of all servers. We denote with  $\mathbf{W}_k = W_{k1}, W_{k2}, \dots, W_{kn}$  and  $\mathbf{X}_i = X_{i1}, X_{i2}, \dots, X_{in} \in \mathbb{F}_q^{|\mathcal{Z}_i| \times n}$ , the length  $n$  sequences of subfunction  $W_k$ , and of  $W_{\mathcal{Z}_i}$  assigned to server  $i \in \Omega$ .

### 2.2. 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 number of  $\Delta$  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:

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

where  $X_i = W_{\mathcal{Z}_i}$ ,  $i \in \Omega$ . As a result of (1), 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  $i \in \Omega$ .

### 2.3. User Demands and Structure of the Computation

We address the problem of distributed lossless compression of a set of general multi-variable functions  $F_j(X_\Omega) : \mathcal{X}_1 \times \mathcal{X}_2 \cdots \times \mathcal{X}_N \rightarrow \mathbb{F}_q$ ,  $j \in [K_c]$ , 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:

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

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

In this work, 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 function (could be linear or non-linear) of dataset  $D_k$ . Hence, using the relation  $X_i = W_{\mathcal{Z}_i} = \{h_k(D_k)\}_{k \in \mathcal{Z}_i}$ , each demanded function  $j \in [K_c]$  can be written in the following form:

$$f_j(W_{\mathcal{K}}) = f_j(h_1(D_1), \dots, h_K(D_K)) = F_j(\{h_k(D_k)\}_{k \in \mathcal{Z}_1}, \dots, \{h_k(D_k)\}_{k \in \mathcal{Z}_N}) = F_j(X_\Omega). \quad (3)$$

In the special case of linearly separable functions (Special instances of the linearly separable representation of subfunctions  $\{W_k\}_k$  given in (4) are linear functions of the datasets  $\{D_k\}$  and are denoted by  $F_j = \sum_k \gamma_{jk} D_k$ .) [39], the demanded functions take the form:

$$\{F_j(X_\Omega)\}_{j \in [K_c]} = [F_1 \ F_2 \ \dots \ F_{K_c}]^T = \mathbf{\Gamma} \mathbf{W}, \quad (4)$$

where  $\mathbf{W} = [W_1 \ W_2 \ \dots \ W_K]^T \in \mathbb{F}_q^{K \times 1}$  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  $F_j(X_\Omega) = \sum_{k \in [K]} \gamma_{jk} \cdot W_k$ . We do not restrict  $\{F_j(X_\Omega)\}_{j \in [K_c]}$  to linearly separable functions, i.e., it may hold that  $\{F_j(X_\Omega)\}_{j \in [K_c]} \neq \mathbf{\Gamma} \mathbf{W}$ .

### 2.4. Communication Cost for the Characteristic-Graph-Based Computing Approach

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  $G_{X_i}^n$  of  $G_{X_i}$  for compressing  $X_i$  to determine the transmitted information. The minimal possible code rate achievable to distinguish the edges of  $G_{X_i}^n$  as  $n \rightarrow \infty$  is given by the Characteristic graph entropy,  $H_{G_{X_i}}(X_i)$ . For a primer on key graph-theoretic concepts, characteristic-graph-related defini-

tions, and the fundamental compression limits of characteristic graphs, we refer the reader to [76,79,82]. In this work, we solely focus on the characterization of the total communication cost from all servers to the user, i.e., the achievable sum-rate, without accounting for the costs of communication between the master and the servers and of computations performed at the servers and the 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

$$R_i = \frac{T_i}{L} = \frac{H(e_{X_i}(c_{G_{\mathbf{X}_i}^n}(\mathbf{X}_i)))}{n} \geq H_{G_{X_i}}(X_i), \quad i \in \Omega, \tag{5}$$

where the inequality follows from exploiting the achievability of  $H_{G_{X_i}}(X_i) = \lim_{n \rightarrow \infty} \frac{1}{n} H_{G_{\mathbf{X}_i}^n}^{\chi}(\mathbf{X}_i)$ , where  $H_{G_{\mathbf{X}_i}^n}^{\chi}(\mathbf{X}_i)$  is the chromatic entropy of the graph  $G_{\mathbf{X}_i}^n$  [73,75]. We refer the reader to Appendix A.2 for a detailed description of the notions of chromatic and graph entropies (cf. (A9) and (A10), respectively).

For the multi-server multi-function distributed setup, using the characteristic-graph-based fundamental limit in (5), an achievable sum-rate for asymptotic lossless computation is

$$R_{\text{ach}} = \sum_{i \in \Omega} R_i \leq \sum_{i \in \Omega} H_{G_{X_i}}(X_i). \tag{6}$$

We next provide our main results in Section 3.

### 3. Main Results

In this section, we analyze the multi-server multi-function distributed computing framework exploiting the characteristic-graph-based approach in [75]. In contrast to the previous research attempts in this direction, our solution method is general, and it captures (i) general input statistics or dataset distributions or the skew in data instead of assuming uniform distributions, (ii) correlations across datasets, (iii) any dataset placement model across servers beyond the cyclic [39] or the Maddah–Ali and Niesen [83] placements, and (iv) general function classes requested by the user, instead of focusing on a particular function type (see, e.g., [39,67,84]).

Subsequently, we delve into specific function computation scenarios. First, we present our main result (Theorem 1), which is the most general form that captures (i)–(iv). We then demonstrate (in Proposition 1) that the celebrated result of Wan et al. [Theorem 2] [39] can be obtained as a special case of Theorem 1, given that (i) the datasets are i.i.d. and uniform over  $q$ -ary fields, (ii) the placement of datasets across servers is cyclic, and (iii) the demanded functions are linearly separable, given as in (4). Under a correlated and identically distributed Bernoulli dataset model with a skewness parameter  $\epsilon \in (0, 1)$  for datasets, we next present in Proposition 2 the achievable sum rate for computing Boolean functions. Finally, in Proposition 3, we analyze our characteristic-graph-based approach for evaluating multi-linear functions, a pertinent class of non-linear functions, under the assumption of cyclic placement and i.i.d. Bernoulli-distributed datasets with parameter  $\epsilon$  and derive an upper bound on the sum rate needed. To gain insight into our analytical results and demonstrate the savings in the total communication cost, we provide some numerical examples.

We next present our main theorem (Theorem 1), on the achievable communication cost for the multi-server multi-function topology, which holds for all input statistics under any correlation model across datasets and for the distributed computing of all function classes

requested by the user, regardless of the data assignment over the servers' caches. The key to capturing the structure of general functions in Theorem 1 is the utilization of a characteristic-graph-based compression technique, as proposed by Körner in [75] (For a more detailed description of characteristic graphs and their entropies, see Appendix A.2.).

**Theorem 1** (Achievable sum-rate using the characteristic graph approach for general functions and distributions). *In the multi-server multi-function distributed computation model, denoted by  $\mathcal{T}(N, K, K_c, M, N_r)$ , 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, the characteristic-graph-based compression yields the following upper bound on the achievable communication rate:*

$$R_{\text{ach}} \leq \sum_{i=1}^{N_r} \min_{Z_i = g_i(X_i) : g_i \in \mathcal{C}_i} H_{G_{X_i}^{\cup}}(X_i), \tag{7}$$

where

- $G_{X_i}^{\cup} = \bigcup_{j \in [K_c]} G_{X_i, j}$  is the union characteristic graph (We refer the reader to (A12) (Appendix A.2) for the definition of a union of characteristic graphs.) 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 that server  $i \in \Omega$  uses for computing  $\{f_j(W_{\mathcal{K}})\}_{j \in [K_c]}$ ,
- each subfunction  $W_k, k \in \mathcal{K}$  is defined over a  $q$ -ary field such that the characteristic is at least 2, and
- $Z_i = g_i(X_i)$  such that  $g_i \in \mathcal{C}_i$  denotes the transmitted information from server  $i \in \Omega$ .

**Proof.** See Appendix B.1.  $\square$

Theorem 1 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  $q \geq 2$ . We note that in (7), the codebook  $\mathcal{C}_i$  determines the structure of the union characteristic graph  $G_{X_i}^{\cup}$ , which, in turn, determines the distribution of  $Z_i$ . Therefore, the tightness of the rate upper bound relies essentially on the codebook selection. We also note that it is possible to analyze the computational complexity of building a characteristic graph and computing the bound in (7) via evaluating the complexity of the transmissions  $Z_i$  determined by  $\{f_j(W_{\mathcal{K}})\}_{j \in [K_c]}$  for a given  $i \in \Omega$ . However, the current manuscript focuses primarily on the cost of communication, and we leave the computational complexity analysis to future work. Because (7) is not analytically tractable, in the following, we focus on special instances of Theorem 1 to gain insights into the effects of input statistics, dataset correlations, and special function classes in determining the total communication cost.

We next demonstrate that the achievable communication cost for the special scenario of the distributed linearly separable computation framework given in [Theorem 2] [39] is embedded by the characterization provided in Theorem 1. We next showcase the achievable sum rate result for linearly separable functions.

**Proposition 1** (Achievable sum-rate using the characteristic graph approach for linearly separable functions and i.i.d. subfunctions over  $\mathbb{F}_q$ ). *In the multi-server multi-function distributed computation model, denoted by  $\mathcal{T}(N, K, K_c, M, N_r)$ , 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 (4), requested by the user, and given i.i.d. uniformly distributed subfunctions over a field of characteristic*

$q \geq 2$ , the characteristic-graph-based compression yields the following bound on the achievable communication rate:

$$R_{\text{ach}} \leq \begin{cases} \min\{K_c, \Delta\} N_r, & 1 \leq K_c \leq \Delta N_r, \\ \min\{K_c, K\}, & \Delta N_r < K_c. \end{cases} \quad (8)$$

**Proof.** See Appendix B.2.  $\square$

We note that Theorem 1 results in Proposition 1 when three conditions hold: (i) the dataset placement across servers is cyclic following the rule in (1), (ii) the subfunctions  $W_{\mathcal{K}}$  are i.i.d. and uniform over  $\mathbb{F}_q$  (see (A21) in Appendix B.2), and (iii) the codebook  $\mathcal{C}_i$  is restricted to linear combinations of subfunctions  $W_{\mathcal{K}}$ , which yields that the independent sets of  $G_{X_i}^{\cup}$  satisfy a set of linear constraints (We detail these linear constraints in Appendix B.2, where the set of linear equations given in (A22) is used to simplify the entropy  $H_{G_{X_i}^{\cup}}(X_i)$  of the union characteristic graph  $G_{X_i}^{\cup}$  via the expression given in (A20) for evaluating the upper bound given in (A18) on the achievable sum rate for computing the desired functions via exploiting the entropies of the union characteristic graphs for each of the  $N_r$  servers, given the recovery threshold  $N_r$ .) in the variables  $\{W_k\}_{k \in \mathcal{Z}_i}$ . Note that the linear encoding and decoding approach for computing linearly separable functions, proposed by Wan et al. in [Theorem 2] [39], is valid over a field of characteristic  $q > 3$ . However, in Proposition 1, the characteristic of  $\mathbb{F}_q$  is at least 2, i.e.,  $q \geq 2$ , generalizing [Theorem 2] [39] to larger input alphabets.

Next, we aim to demonstrate the merits of the characteristic-graph-based compression in capturing dataset correlations within the multi-server multi-function distributed computation framework. More specifically, we restrict the general input statistics in Theorem 1 such that the datasets are correlated and identically distributed, where each subfunction follows a Bernoulli distribution with the same parameter  $\epsilon$ , i.e.,  $W_k \sim \text{Bern}(\epsilon)$ , with  $\epsilon \in (0, 1)$ , and the user demands  $K_c$  arbitrary Boolean functions regardless of the data assignment. Similarly to Theorem 1, the following proposition (Proposition 2) holds for general function types (Boolean) regardless of the data assignment.

**Proposition 2** (Achievable sum-rate using the characteristic graph approach for general functions and identically distributed subfunctions over  $\mathbb{F}_2$ ). *In the multi-server multi-function distributed computing setting, denoted by  $\mathcal{T}(N, K, K_c, M, N_r)$ , under the general placement of datasets, and for a set of  $K_c$  Boolean functions  $\{f_j(W_{\mathcal{K}})\}_{j \in [K_c]}$  requested by the user, and given identically distributed and correlated subfunctions with  $W_k \sim \text{Bern}(\epsilon)$ ,  $k \in [K]$ , where  $\epsilon \in (0, 1)$ , the characteristic-graph-based compression yields the following bound on the achievable communication rate:*

$$R_{\text{ach}} \leq \sum_{i=1}^{N_r} \min_{Z_i = g_i(X_i) : g_i \in \mathcal{C}_i} h(Z_i), \quad (9)$$

where

- $\mathcal{C}_i \ni g_i : \{0, 1\}^M \rightarrow \{0, 1\}$  denotes a codebook of Boolean functions that server  $i \in \Omega$  uses,
- $Z_i = g_i(X_i)$  such that  $g_i \in \mathcal{C}_i$  denotes the transmitted information from server  $i \in \Omega$ ,
- $G_{X_i}^{\cup}$  has two maximal independent sets (MISs), namely,  $s_0(G_{X_i}^{\cup})$  and  $s_1(G_{X_i}^{\cup})$ , yielding  $Z_i = 0$  and  $Z_i = 1$ , respectively, and
- the probability that  $W_{\mathcal{Z}_i}$  yields the function value  $Z_i = 1$  is given as

$$\mathbb{P}(Z_i = 1) = \mathbb{P}(W_{\mathcal{Z}_i} \in s_1(G_{X_i}^{\cup})), \quad i \in \Omega. \quad (10)$$

**Proof.** See Appendix B.3.  $\square$

While, admittedly, the above approach (Proposition 2) may not directly offer sufficient insight, it does employ the new machinery to offer a generality that allows us to plug in any set of parameters to determine the achievable performance.

Contrasting Propositions 1–2, which give the total communication costs for computing linearly separable and Boolean functions, respectively, over  $\mathbb{F}_2$ , Proposition 2, by exploiting the skew and correlations of datasets indexed by  $\mathcal{Z}_i$ , as well as the functions’ structures via the MISs  $s_0(G_{X_i}^U)$  and  $s_1(G_{X_i}^U)$  of server  $i \in \Omega$ , demonstrates that harnessing the correlation across the datasets can indeed reduce the total communication cost versus the setting in Proposition 1, devised with the assumption of i.i.d. and uniformly distributed subfunctions.

The prior works have focused on devising distributed computation frameworks and exploring their communication costs for specific function classes. For instance, in [62], Körner and Marton have restricted the computation to be the binary sum function, and in [72], Han and Kobayashi have classified functions into two categories depending on whether they can be computed at a sum rate that is lower than that of [60]. Furthermore, the computation problem has been studied for specific topologies, e.g., the side information setting in [73,74]. Despite the existing efforts, see, e.g., [62,72–74], to the best of our knowledge, for the given multi-server multi-function distributed computing scenario, there is still no general framework for determining the fundamental limits of the total communication cost for computing general non-linear functions. Indeed, for this setting, the most pertinent existing work that applies to general non-linear functions and provides an upper bound on the achievable sum rate is that of Slepian–Wolf [60]. On the other hand, the upper bound on the achievable computation scheme presented in Theorem 1 can provide savings in the communication cost over [60] for functions including linearly separable functions and beyond. To that end, we exploit Theorem 1 to determine an upper bound on the achievable sum-rate for distributed computing of a multi-linear function in the form of

$$f(W_{\mathcal{K}}) = \prod_{k \in [K]} W_k. \tag{11}$$

Note that (11) is used in various scenarios, including distributed machine learning, e.g., to reduce variance in noisy datasets via ensemble learning [85] or weighted averaging [86], sensor network applications to aggregate readings for improved data analysis [87], as well as distributed optimization and financial modeling, where these functions play pivotal roles in establishing global objectives and managing risk and return [88,89].

Drawing on the utility of characteristic graphs in capturing the structures of data and functions, as well as input statistics and correlations, and the general result in Theorem 1, our next result, Proposition 3, demonstrates a new upper bound on the achievable sum rate for computing multi-linear functions within the framework of multi-server and multi-function distributed computing via exploiting conditional graph entropies.

**Proposition 3** (Achievable sum-rate using the characteristic graph approach for multi-linear functions and i.i.d. subfunctions over  $\mathbb{F}_2$ ). *In a multi-server multi-function distributed computing setting, denoted by  $\mathcal{T}(N, K, K_c, M, N_r)$ , under the cyclic placement of datasets, where  $\frac{K}{N} = \Delta \in \mathbb{Z}^+$ , and for computing the multi-linear function ( $K_c = 1$ ), given as in (11), requested by the user, and given i.i.d. uniformly distributed subfunctions  $W_k \sim \text{Bern}(\epsilon)$ ,  $k \in [K]$ , for some  $\epsilon \in (0, 1)$ , the characteristic-graph-based compression yields the following bound on the achievable communication rate:*

$$R_{\text{ach}} \leq \frac{1 - (\epsilon_M)^{N^*}}{1 - \epsilon_M} \cdot h(\epsilon_M) + (\epsilon_M)^{N^*} \cdot \mathbf{1}_{\Delta_N > 0} \cdot h(\epsilon_{\xi_N}), \tag{12}$$

where

- $\epsilon_M = \epsilon^M$  denotes the probability that the product of  $M$  subfunctions, with  $W_k \sim \text{Bern}(\epsilon)$  being i.i.d. across  $k \in [K]$ , take the value one, i.e.,  $\mathbb{P}\left(\prod_{k \in \mathcal{S}: |\mathcal{S}|=M} W_k\right) = \epsilon_M$ ,
- the variable  $N^* = \left\lfloor \frac{N}{N - N_r + 1} \right\rfloor$  denotes the minimum number of servers needed to compute  $f(W_{\mathcal{K}})$ , given as in (11), where each of these servers computes a disjoint product of  $M$  subfunctions, and

- the variable  $\Delta_N = N - N^* \cdot (N - N_r + 1)$  represents whether an additional server is needed to aid the computation, and if  $\Delta_N \geq 1$ , then  $\zeta_N$  denotes the number of subfunctions to be computed by the additional server, and similarly to the above,  $\mathbb{P}\left(\prod_{k \in \mathcal{S}: |\mathcal{S}|=\zeta_N} W_k\right) = \epsilon^{\zeta_N}$ .

**Proof.** See Appendix B.4.  $\square$

We next detail two numerical examples (Sections 4.1 and 4.2) to showcase the achievable gains in the total communication cost for Proposition 2 and Proposition 3, respectively.

#### 4. Numerical Evaluations to Demonstrate the Achievable Gains

Given  $\mathcal{T}(N, K, K_c, M, N_r)$ , to gain insight into our analytical results and demonstrate the savings in the total communication cost, we provide some numerical examples. To demonstrate Proposition 2, in Section 4.1, we focus on computing linearly separable functions, and in Section 4.2 (cf. Proposition 3), we focus on multi-linear functions, respectively.

To that end, to characterize the performance of our characteristic-graph-based approach for linearly separable functions, we denote by  $\eta_{lin}$  the gain of the sum-rate for the characteristic-graph-based approach given in (9) over the sum-rate of the distributed scheme of Wan et al. in [39], given in (8), and by  $\eta_{SW}$  the gain of the sum-rate in (9) over the sum-rate of the fully distributed approach of Slepian–Wolf [60]. To capture general statistics, i.e., dataset skewness and correlations, and make a fair comparison, we adapt the transmission model of Wan et al. in [39] via modifying the i.i.d. dataset assumption.

We next study an example scenario (Section 4.1) for computing a class of linearly separable functions (4) over  $\mathbb{F}_2$ , where each of the demanded functions takes the form  $f_j(W_K) = \sum_{k \in [K]} \gamma_{jk} W_k \pmod 2, j \in [K_c]$  under a specific correlation model across subfunctions. More specifically, when the subfunctions  $W_k \sim \text{Bern}(\epsilon)$  are identically distributed and correlated across  $k \in [K]$ , and  $\Delta \in \mathbb{Z}^+$ , we model the correlation across datasets (a) exploiting the joint PMF model in [Theorem 1] [90] and (b) for a joint PMF described in Table 2. Furthermore, we assume for  $K_c > 1$  that  $\Gamma = \{\gamma_{jk}\} \in \mathbb{F}_2^{K_c \times K}$  is full rank. For the proposed setting, we next demonstrate the achievable gains  $\eta_{lin}$  of our proposed technique versus  $\epsilon$  for computing (4) as a function of skew,  $\epsilon$ , and correlation,  $\rho$ , of datasets,  $K_c \in [N_r] < K$ , and other system parameters and showcase the results via Figures 1, 3–5.

**Table 2.** Joint PMF  $P_{W_2, W_3}$  of  $W_2$  and  $W_3$  with a crossover parameter  $p$ , in Section 4.1 (Scenario II).

$P_{W_2, W_3}(W_2, W_3)$	$W_2 = 0$	$W_2 = 1$
$W_3 = 0$	$(1 - \epsilon)(1 - p')$	$\epsilon p$
$W_3 = 1$	$\epsilon p$	$\epsilon(1 - p)$

##### 4.1. Example Case: Distributed Computing of Linearly Separable Functions over $\mathbb{F}_2$

We consider the computation of the linearly separable functions given in (4) for general topologies, with general  $N, K, M, N_r, K_c$ , over  $\mathbb{F}_2$ , with an identical skew parameter  $\epsilon \in [0, 1]$  for each subfunction, where  $W_k \sim \text{Bern}(\epsilon), k \in [K]$ , using cyclic placement as in (1) and incorporating the correlation between the subfunctions, with the correlation coefficient denoted by  $\rho$ . We consider three scenarios, as described next:

**Scenario I. The number of demanded functions is  $K_c = 1$ , where the subfunctions could be uncorrelated or correlated.**

This scenario is similar to the setting in [39], although different from [39], which is valid over a field of characteristic  $q > 3$ , we consider  $\mathbb{F}_2$ , and in the case of correlations, i.e., when  $\rho > 0$ , we capture the correlations across the transmissions (evaluated from subfunctions of datasets) from distributed servers, as detailed earlier in Section 3. We first assume that the subfunctions are not correlated, i.e.,  $\rho = 0$ , and evaluate  $\eta_{lin}$  for  $f(W_K) = \sum_{k \in [K]} W_k$

mod 2. The parameter of  $f(W_{\mathcal{K}})$ , i.e., the probability that  $f(W_{\mathcal{K}})$  takes the value 1 can be computed using the recursive relation

$$\begin{aligned} \mathbb{P}\left(\sum_{k \in \mathcal{S}: |\mathcal{S}|=l \leq K} W_k \bmod 2 = 1\right) &= \sum_{k \in \mathcal{S}: |\mathcal{S}|=l \leq K, k \text{ odd}} \mathbb{P}(B(K, \epsilon)) = k \\ &= (1 - \epsilon_{l-1}) \cdot \epsilon + \epsilon_{l-1} \cdot (1 - \epsilon) \\ &\doteq \epsilon_l, \quad 1 < l \leq K, \end{aligned} \tag{13}$$

where  $B(K, \epsilon)$  is the binomial PMF, and  $\epsilon_l$  is the probability of the modulo 2 sum of any  $1 < l \leq K$  subfunctions taking the value one, with  $W_k \sim \text{Bern}(\epsilon)$  being i.i.d. across  $k \in \mathcal{S}$ , with the convention  $\epsilon_1 = \epsilon$ .

Given  $N_r$ , we denote by  $N^* = \lfloor \frac{N}{N-N_r+1} \rfloor$  the minimum number of servers, corresponding to the subset  $\mathcal{N}^* \subseteq \Omega$ , needed to compute  $f(W_{\mathcal{K}})$ , where each server, with a cache size of  $M$ , computes a sum of  $M$  subfunctions, where across these  $N^*$  servers, the sets of subfunctions are disjoint. Hence,  $\mathbb{P}\left(\sum_{k \in \mathcal{S}: |\mathcal{S}|=M} W_k\right) = \epsilon_M$ . Furthermore, the variable  $\Delta_N = N - N^* \cdot (N - N_r + 1)$  represents whether additional servers in addition to  $N^*$  servers are needed to aid the computation, and if  $\Delta_N \geq 1$ , then  $\Delta \cdot \Delta_N \doteq \zeta_N$  denotes the number of subfunctions to be computed by the set of additional servers, namely,  $\mathcal{I}^* \in \Omega$ , and similarly to the above,  $\mathbb{P}\left(\sum_{k \in \mathcal{S}: |\mathcal{S}|=\zeta_N} W_k\right) = \epsilon_{\zeta_N}$ , which is obtained by evaluating  $\epsilon_l$  at  $l = \zeta_N$ .

Adapting (8) for  $\mathbb{F}_2$ , we obtain the total communication cost  $R_{ach}(lin)$  for computing the linearly separable function  $f(W_{\mathcal{K}}) = \sum_{k \in [\mathcal{K}]} W_k \bmod 2$  as

$$R_{ach}(lin) = \sum_{i=1}^{N_r} h\left(\sum_{k \in \mathcal{Z}_i} W_k\right) = N_r \cdot h(\epsilon_M). \tag{14}$$

Using Proposition 2 and (13), we derive the sum rate for distributed lossless computing of  $f(W_{\mathcal{K}})$  as

$$\sum_{i \in \Omega} R_i \leq N^* \cdot h(\epsilon_M) + 1_{\Delta_N > 0} \cdot h(\epsilon_{\zeta_N}), \tag{15}$$

where the indicator function  $1_{\Delta_N > 0}$  captures the rate contribution from the additional server, if any. Using (15), the gain  $\eta_{lin}$  over the linearly separable solution of [39] is presented as

$$\eta_{lin} = \frac{N_r \cdot h(\epsilon_M)}{N^* \cdot h(\epsilon_M) + 1_{\Delta_N > 0} \cdot h(\epsilon_{\zeta_N})}, \tag{16}$$

where  $h(\epsilon_{\zeta_N})$  represents the rate needed from the set of additional servers  $\mathcal{I}^* \in \Omega$ , aiding the computation through communicating the sum of the remaining subfunctions in the set  $\mathcal{C} \subseteq \mathcal{Z}_{\mathcal{I}^*}$ , where the summation for these remaining functions in  $\mathcal{C} \subseteq \mathcal{Z}_{\mathcal{I}^*}$  is denoted as  $\sum_{k \in \mathcal{C} \subseteq \mathcal{Z}_{\mathcal{I}^*}: k \notin \cup_{i \in \mathcal{N}^*} \mathcal{Z}_i, |\mathcal{C}|=\zeta_N} W_k$ , which cannot be captured by the set  $\mathcal{N}^*$ .

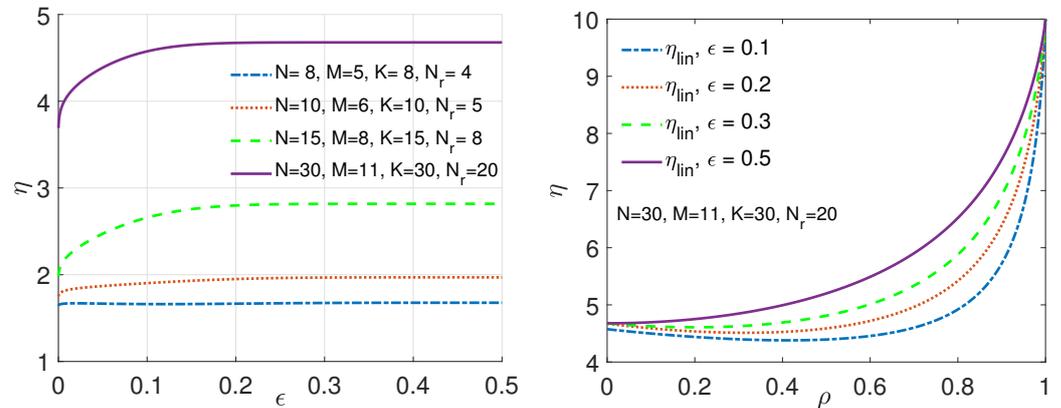
Given  $K_c = 1$  for the given modulo 2 sum function, we next incorporate the correlation model in [90] for each  $W_k$ , identically distributed with  $W_k \sim \text{Bern}(\epsilon)$ , and correlation  $\rho$  across any two subfunctions. The formulation in [90] yields the following PMF for  $f(W_{\mathcal{K}})$ :

$$\begin{aligned} \mathbb{P}(f(W_{\mathcal{K}}) = y) &= \binom{K}{y} \epsilon^y (1 - \epsilon)^{K-y} (1 - \rho) \cdot 1_{y \in A_1} \\ &+ \epsilon^{\frac{y}{K}} (1 - \epsilon)^{\frac{K-y}{K}} \rho \cdot 1_{y \in A_2}, \quad y \in \{0, \dots, K\}, \end{aligned} \tag{17}$$

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

We depict the behavior of our gain,  $\eta_{lin}$ , using the same topology  $\mathcal{T}(N, K, K_c, M, N_r)$  as in [39], with different system parameters  $(N, K, M, N_r)$ , under  $\rho = 0$  in Figure 1-(Left).

As we increase both  $N$  and  $K$ , along with the number of active servers,  $N_r$ , the gain,  $\eta_{lin}$ , of the characteristic graph approach increases. This stems from the characteristic graph approach to compute functions  $f(W_K)$  of  $W_K$  using  $N^*$  servers. From Figure 1-(Right), it is evident that by capturing correlations between the subfunctions, hence, across the servers' caches,  $\eta_{lin}$  grows more rapidly until it reaches the maximum of (16), corresponding to  $\eta_{lin} = \frac{N_r}{N^*} = 10$ , attributed to full correlation (see, Figure 1-(Right)).



**Figure 1.** The gain  $\eta_{lin}$  of the characteristic graph approach for  $K_c = 1$  in Section 4.1 (Scenario I). (Left)  $\rho = 0$  for various distributed topologies. (Right) The correlation model given as (17) for  $\mathcal{T}(30, 30, 1, 11, 20)$  with different  $\epsilon$  values.

What we can also see is that for  $\rho = 0$ , the gain rises with the increase in  $\epsilon$  and linearly grows with  $\frac{N_r}{N^*}$ . As  $\rho$  increases, reaching its maximum at  $\rho = 1$ , the gain is maximized, yielding the minimum communication cost that can be achieved with our technique. Here, the gain  $\eta_{lin}$  is dictated by the topology and is given as  $\eta_{lin} = \frac{N_r}{N^*}$ . This linear relation shows that this specific topology can provide a very substantial reduction in the total communication cost, as  $\rho$  goes to 1, over the state of the art [39], as shown in Figure 1-(Right) via the purple (solid) curve. Furthermore, one can draw a comparison between the characteristic graph approach and the approach in [60]. Here, we represent the gain as  $\eta_{SW}$ . It is noteworthy that the sum-rate of all servers using the coding approach of Slepian–Wolf [60] is  $R_{ach}(SW) = H(W_K)$ . With  $\rho = 0$ , this expression simplifies to  $R_{ach}(SW) = K \cdot H(W_k)$ , resulting again in a substantial reduction in the communication cost, as we see from  $R_{ach}(lin)$  in (14) for the same topology of the purple (solid) curve as shown in Figure 1-(Right).

**Scenario II. The number of demanded functions is  $K_c = 2$ , where the subfunctions could be uncorrelated or correlated.**

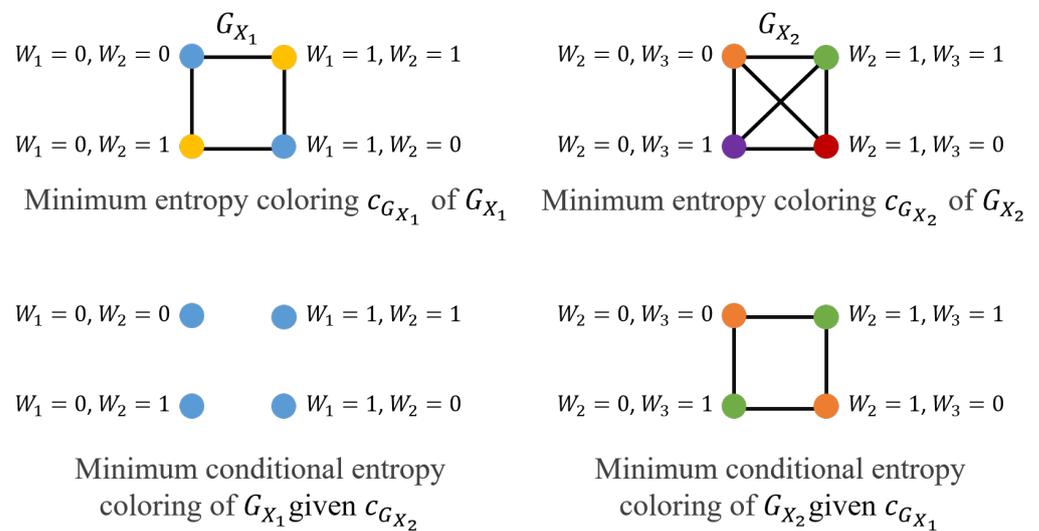
To gain insights into the behavior of  $\eta_{lin}$ , we consider an example distributed computation model with  $K = N = 3$ ,  $N_r = 2$ , where the subfunctions  $W_1, W_2, W_3$  are assigned to  $X_1, X_2$ , and  $X_3$  in a cyclic manner, with  $h(W_k) = \epsilon, k \in [3]$ , and  $K_c = 2$  with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

Given  $N_r = 2$ , using the characteristic graph approach for individual servers, an achievable compression scheme, for a given ordering  $i$  and  $j$  of server transmissions, relies on first compression of the characteristic graph  $G_{X_i}$  constructed by server  $i \in \Omega$  that has no side information and then the conditional rate needed for compressing the colors of  $G_{X_j}$  for any other server  $j \in \Omega \setminus i$  via incorporating the side information  $Z_i = g_i(X_i)$  obtained from server  $i \in \Omega$ . Thus, contrasting the total communication cost associated with the possible orderings, the minimum total communication cost  $R_{ach}(G)$  can be determined (We can generalize (18) to  $N_r > 2$ , where, for a given ordering of server transmissions, any consecutive server that transmits sees all previous transmissions as side information

and the best ordering that has the minimum total communication cost, i.e.,  $R_{ach}(G)$ . The achievable sum rate here takes the form

$$R_{ach}(G) = \min\{H_{G_{X_1}}(X_1) + H_{G_{X_2}}(X_2 | Z_1), \quad H_{G_{X_2}}(X_2) + H_{G_{X_1}}(X_1 | Z_2)\}. \quad (18)$$

Focusing on the characteristic graph approach, we illustrate how each server builds its union characteristic graph for simultaneously computing  $f_1$  and  $f_2$  according to (A12) (as detailed in Appendix A.2.1), in Figure 2. In (18), the first term corresponds to  $G_{X_1} = (V_{X_1}, E_{X_1})$ , where  $V_{X_1} = \{0, 1\}^2$  is built using the support of  $W_1$  and  $W_2$ , and the edges  $E_{X_1}$  are built based on the rule that  $(x_1^1, x_1^2) \in E_{X_1}$  if  $F(x_1^1, x_2) \neq F(x_1^2, x_2)$  for some  $x_2 \in V_{X_2}$ , which, as we see here, requires two colors. Similarly, server 2 constructs  $G_{X_2} = (V_{X_2}, E_{X_2})$  given  $Z_1$ , where  $V_{X_2} = \{0, 1\}^2$  using the support of  $W_2$  and  $W_3$ , and where  $Z_1$  determines  $f_1 = W_2$ , and hence, to compute  $f_2 = W_2 + W_3$  given  $f_1 = W_2$ , any two vertices taking values (Here,  $x_2^1 = (w_2^1, w_3^1)$  and  $x_2^2 = (w_2^2, w_3^2)$  represent two different realizations of the pair of subfunctions  $W_2$  and  $W_3$ .)  $x_2^1 = (w_2^1, w_3^1) \in V_{X_2}$  and  $x_2^2 = (w_2^2, w_3^2) \in V_{X_2}$  are connected if  $w_3^1 \neq w_3^2$ . Hence, we require two distinct colors for  $G_{X_2}$ . As a result, the first term yields a sum rate of  $h(\epsilon) + h(\epsilon) = 2h(\epsilon)$ . Similarly, the second term of (18) captures the impact of  $G_{X_2} = (V_{X_2}, E_{X_2})$ , where server 2 builds  $G_{X_2}$  using the support of  $W_2$  and  $W_3$ , and  $G_{X_2}$  is a complete graph to distinguish all possible binary pairs to compute  $f_1$  and  $f_2$ , 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, which means that  $H_{G_{X_1}}(X_1 | Z_2) = 0$ . As a result, the ordering of server transmission given by the second term of (18) yields the same sum rate of  $2h(\epsilon) + 0 = 2h(\epsilon)$ . For this setting, the minimum required rate is  $R_{ach}(G) = 2h(\epsilon)$ , and the configuration captured by the second term provides a lower recovery threshold of  $N_r = 1$  versus  $N_r = 2$  for the configurations of server transmissions given by the first term (18). The different  $N_r$  achieved by these two configurations is also captured by Figure 2.



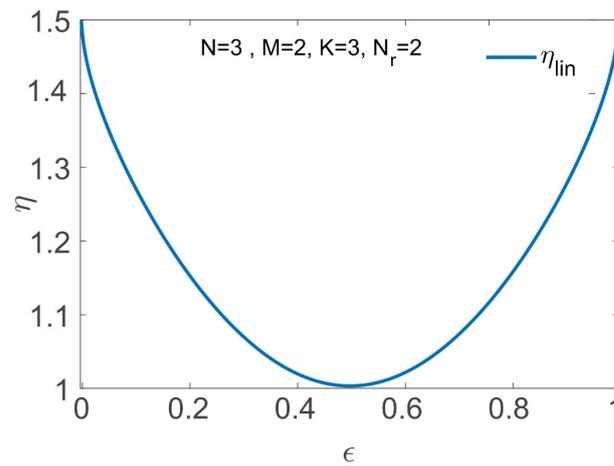
**Figure 2.** Colorings of graphs in Section 4.1 (Scenario II). (Top Left–Right) Characteristic graphs  $G_{X_1}$  and  $G_{X_2}$ , respectively. (Bottom Left–Right) The minimum conditional entropy colorings of  $G_{X_1}$  given  $c_{G_{X_2}}$  and  $G_{X_2}$  given  $c_{G_{X_1}}$ , respectively.

Alternatively, in the linearly separable approach [39],  $N_r$  servers transmit the requested function of the datasets stored in their caches. For distributed computing of  $f_1$  and  $f_2$ , servers 1 and 2 transmit at rate  $H(W_2) = h(\epsilon)$ , for computing  $f_1$ , and at rate  $H(W_2 + W_3)$ , for function  $f_2$ . As a result, the achievable communication cost is given by  $R_{ach}(lin) = h(\epsilon) + h(W_2 + W_3)$ . Here, for a fair comparison, we update the model studied in [39] to capture the correlation within each server without accounting for the correlation across the servers.

Under this setting, for  $\rho = 0$ , we see that the gain  $\eta_{lin}$  of the characteristic graph approach over the linearly separable solution of [39] for computing  $f_1$  and  $f_2$  as a function of  $\epsilon \in [0, 1]$  takes the form

$$\eta_{lin}(\epsilon) = \frac{h(\epsilon) + h(2\epsilon(1 - \epsilon))}{2h(\epsilon)} \begin{cases} = 1 & , \epsilon = \{\frac{1}{2}\}, \\ > 1 & , \epsilon \in [0, 1] \setminus \{\frac{1}{2}\}, \end{cases} \tag{19}$$

where  $\eta_{lin}(\epsilon) > 1$  for  $\epsilon \neq \frac{1}{2}$  follows from the concavity of  $h(\cdot)$ , which yields the inequality  $h(2\epsilon(1 - \epsilon)) \geq h(\epsilon)$ . Furthermore,  $\eta_{lin}$  approaches 1.5 as  $\epsilon \rightarrow \{0, 1\}$  (see Figure 3).



**Figure 3.**  $\eta_{lin}$  in (19) versus  $\epsilon$ , for distributed computing of  $f_1 = W_2$  and  $f_2 = W_2 + W_3$ , where  $K_c = 2$ ,  $N_r = 2$ , with  $\rho = 0$ , in Section 4.1 (Scenario II).

We next examine the setting where the correlation coefficient  $\rho$  is nonzero, using the joint PMF  $P_{W_2, W_3}$ , as depicted in Table 2, of the required subfunctions ( $W_2$  and  $W_3$ ) in computing  $f_1$  and  $f_2$ . This PMF describes the joint PMF corresponding to a binary non-symmetric channel model, where the correlation coefficient between  $W_2$  and  $W_3$  is  $\rho = \frac{1-p}{1-\epsilon}$ , and where  $p' = \frac{\epsilon p}{1-\epsilon}$ . Thus, our gain here compared to the linearly separable encoding and decoding approach of [39] is given as

$$\eta_{lin} = \frac{H(W_2) + H(W_2 + W_3)}{H(W_2, W_3)} = \frac{h(\epsilon) + h(2\epsilon p)}{h(\epsilon) + (1 - \epsilon)h(\frac{\epsilon p}{1-\epsilon}) + \epsilon h(p)}. \tag{20}$$

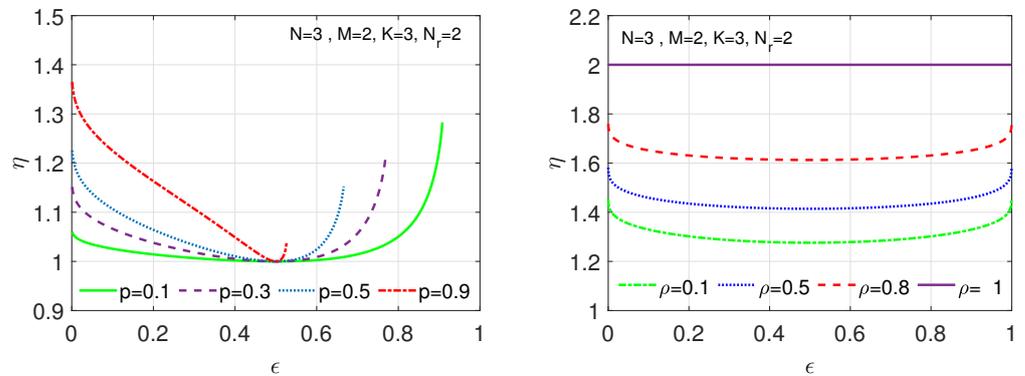
We consider now the correlation model in Table 2, where coefficient  $\rho$  rises in  $\epsilon$  for a fixed  $p$ . In Figure 4-(Left), we illustrate the behavior of  $\eta_{lin}$ , given by (20), for computing  $f_1$  and  $f_2$  for  $N_r = 2$  as a function of  $p$  and  $\epsilon$ , where for this setting, the correlation coefficient  $\rho$  is a decreasing function of  $p$  and an increasing function of  $\epsilon$ . We observe from (20) that the gain  $\eta_{lin}$  satisfies  $\eta_{lin} \geq 1$  for all  $\epsilon \in [0, 1]$ , which monotonically increases in  $p$ —and hence monotonically decreases in  $\rho$  due to the relation  $\rho = \frac{1-p}{1-\epsilon}$ —as a function of the deviation of  $\epsilon$  from  $1/2$ . For  $\epsilon \in (0.5, 1]$ ,  $\eta_{lin}$  increases in  $\epsilon$ . For example, for  $p = 0.1$  then  $\eta_{lin}(1) = 1.28$ , as depicted by the green (solid) curve. Similarly, given  $\epsilon \in [0, 0.5)$ , decreasing  $\epsilon$  results in  $\eta_{lin}$  to exhibit a rising trend, e.g., for  $p = 0.9$  then  $\eta_{lin}(0) = 1.36$ , as shown by the red (dash-dotted) curve. As  $p$  approaches one,  $\eta_{lin}$  goes to 1.5 as  $\epsilon$  tends to zero, which can be derived from (20). We here note that the gains are generally smaller than in the previous set of comparisons, as shown in Figure 3.

More generally, given a user request consisting of  $K_c = 2$  linearly separable functions (i.e., satisfying (4)), and after considering (20) beyond  $N_r = 2$ , we see that  $\eta_{lin}$  is at most  $N_r$  as  $\rho$  approaches one. We next use the joint PMF model used in obtaining (17), where we

observe that  $f_2 \sim ((1 - \epsilon)^2(1 - \rho) + (1 - \epsilon)\rho, 2\epsilon(1 - \epsilon)(1 - \rho), \epsilon^2(1 - \rho) + \epsilon\rho)$ , to see that the gain takes the form

$$\eta_{lin} = \frac{h(\epsilon) + H(f_2)}{h(\epsilon) + (1 - \epsilon)h(\zeta_1) + \epsilon h(\zeta_2)}, \tag{21}$$

where  $\zeta_1 = (1 - \epsilon)(1 - \rho) + \rho$ , and  $\zeta_2 = (1 - \epsilon)(1 - \rho)$ . For this model, we illustrate  $\eta_{lin}$  versus  $\epsilon$  in Figure 4-(Right) for different  $\rho$  values. Evaluating (21), 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, a gain  $\eta_{lin} = N_r = 2$ , 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 hence, it can be shown that the gain is lower bounded as  $\eta_{lin} \geq 1.25$ .



**Figure 4.**  $\eta_{lin}$  versus  $\epsilon$ , for distributed computing of  $f_1 = W_2$  and  $f_2 = W_2 + W_3$ , where  $K_c = 2$ ,  $N_r = 2$ , in Section 4.1, using different joint PMF models for  $P_{W_2, W_3}$  (Scenario II). (Left)  $\eta_{lin}$  in (20) for the joint PMF in Table 2 for different values of  $p$ . (Right)  $\eta_{lin}$  for the joint PMF in (17) for different values of  $\rho$ .

**Scenario III. The number of demanded functions is  $K_c \in [N_r]$ , and the number of datasets is equal to the number of servers, i.e.,  $K = N$ , where the subfunctions are uncorrelated.**

We now provide an achievable rate comparison between the approach in [39] and our graph-based approach, as summarized by our Proposition 1, which generalizes the result in [Theorem 2] [39] to finite fields with characteristics  $q \geq 2$ , for the case of  $\rho = 0$ .

Here, to capture dataset skewness and make a fair comparison, we adapt the transmission model of Wan et al. in [39] via modifying the i.i.d. dataset assumption and taking into account the skewness incurred within each server in determining the local computations  $\sum_{k \in \mathcal{S}: |S|=M} W_k$  at each server.

For the linearly separable model in (4), adapted to account for our setting, exploiting the summation  $\sum_{k \in \mathcal{Z}_i} W_k$ , and  $\epsilon_M$  given in (15), the communication cost for a general number of  $K_c$  with  $\rho = 0$  is expressed as

$$R_{ach}(lin) = N_r \cdot h(\epsilon_M). \tag{22}$$

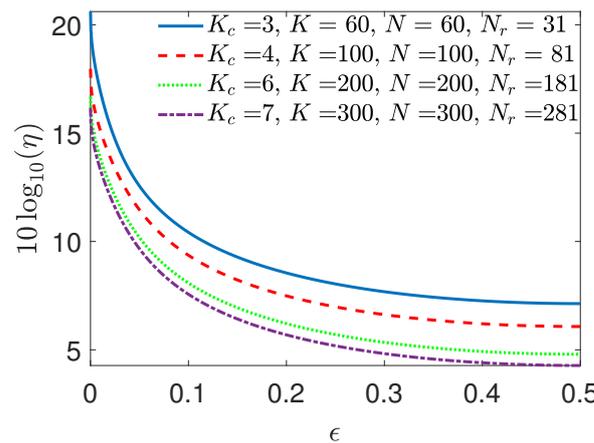
In (22), as  $\epsilon$  approaches 0 or 1, then  $h(\epsilon_M) \rightarrow 0$ . Subsequently, the achievable communication cost for the characteristic graph model can be determined as

$$R_{ach}(G) = K_c \cdot N^* \cdot h(\epsilon). \tag{23}$$

To understand the behavior of  $\eta_{lin} = \frac{N_r}{K_c N^*} \cdot \frac{h(\epsilon_M)}{h(\epsilon)}$ , knowing that  $\frac{N_r}{K_c N^*}$  is a fixed parameter, we need to examine the dynamic component  $\frac{h(\epsilon_M)}{h(\epsilon)}$ . Exploiting Schur concavity (A real-valued function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is Schur concave if  $f(x_1, x_2, \dots, x_n) \leq f(y_1, y_2, \dots, y_n)$  holds whenever  $(x_1, x_2, \dots, x_n)$  majorizes  $(y_1, y_2, \dots, y_n)$ , i.e.,  $\sum_{i=1}^k x_i \geq \sum_{i=1}^k y_i$ , for all  $k \in [n]$  [91].) for the binary entropy function, which tells us that  $h(\mathbb{E}[X]) \geq \mathbb{E}[h(X)]$ , we can see that as  $\epsilon$  approaches 0 or 1, then

$$\lim_{\epsilon \rightarrow \{0,1\}} \frac{h(\epsilon_M)}{h(\epsilon)} \leq M, \quad M \in \mathbb{Z}^+, \tag{24}$$

where the inequality between the left- and right-hand sides becomes loose as a function of  $M$ . As a result, as  $\epsilon$  approaches 0 or 1, then  $\eta_{lin} \approx M \cdot \frac{N_r}{K_c N^*}$ , which follows from exploiting (22), (23) and the achievability of the upper bound in (24). We illustrate the upper bound on  $\eta_{lin}$  in Figure 5 and demonstrate the  $\eta_{lin}$  behavior for  $K_c$  demanded functions across various topologies with circular dataset placement, namely, for various  $K = N$ , i.e., when the amount of circular shift between two consecutive servers is  $\Delta = \frac{K}{N} = 1$  and the cache size is  $M = N - N_r + 1$ , and for  $\rho = 0$  and  $\epsilon \leq 1/2$ . We focus only on plotting  $\eta_{lin}$  for  $\epsilon \leq 1/2$ , accounting for the symmetry of the entropy function. Therefore, we only plot for  $\epsilon \leq 1/2$ . The multiplicative coefficient  $\frac{N_r}{K_c N^*}$  of  $\eta_{lin}$  determines the growth, which is depicted by the curves.



**Figure 5.**  $\eta_{lin}$  in a logarithmic scale versus  $\epsilon$  for  $K_c$  demanded functions for various values of  $K_c$ , with  $\rho = 0$  for different topologies, as detailed in Section 4.1 (Scenario III).

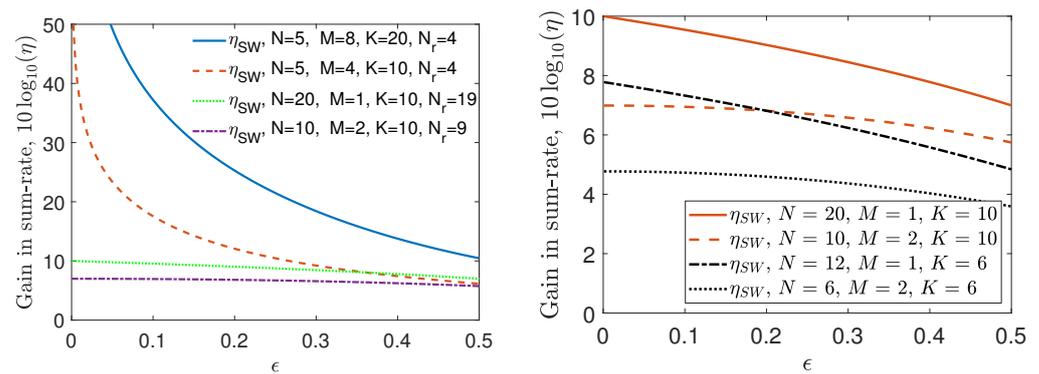
Thus, we see that for a given topology  $\mathcal{T}(N, K, K_c, M, N_r)$  with  $K_c$  demanded functions, for  $\rho = 0$ , using (24), we see that  $\eta_{lin}$  exponentially grows with term  $1 - \epsilon$  for  $\epsilon \in [0, 1/2]$  (Here, we note that the behavior of  $\eta_{lin}$  is symmetric around  $\epsilon = 1/2$ .), and very substantial reduction in the total communication cost is possible as  $\epsilon$  approaches  $\{0, 1\}$ , as shown in Figure 5 by the blue (solid) curve. The gain over [Theorem 2] [39],  $\eta_{lin}$ , for a given topology, changes proportionally to  $\frac{N_r}{K_c N^*}$ . The gain over [60],  $\eta_{SW}$ , for  $\rho = 0$  linearly scales (Incorporating the dataset skew to Proposition 1 ([Theorem 2] [39]),  $R_{ach}(lin)$  is simplified to (22), which from (24) can linearly grow in  $M = N - N_r + 1$  at high skew, explaining the inferior performance of Proposition 1 over [60] as a function of the skew.) with  $\frac{K}{K_c N^*}$ . For instance, the gain for the blue (solid) curve in Figure 5 is  $\eta_{SW} = 10$ .

In general, other functions in  $\mathbb{F}_2$ , such as bitwise AND and the multi-linear function (see, e.g., Proposition 3) are more skewed and have lower entropies than linearly separable functions and, hence, are easier to compute. Therefore, the cost given in (23) can serve as an upper bound for the communication costs of those more skewed functions in  $\mathbb{F}_2$ .

We have here provided insights into the achievable gains in communication cost for several scenarios. We leave the study of  $\eta_{lin}$  for more general topologies  $\mathcal{T}(N, K, K_c, M, N_r)$

and correlation models beyond (17) devised for linearly separable functions, and beyond the joint PMF model in Table 2, as future work.

Proposition 3 illustrates the power of the characteristic graph approach in decreasing the communication cost for distributed computing of multi-linear functions, given as in (11), compared to recovering the local computations  $\prod_{k \in \mathcal{S}: |\mathcal{S}|=M} W_k$  using [60]. We denote by  $\eta_{SW}$  the gain of the sum-rate for the graph entropy-based approach given in (12)—using the conditional entropy-based sum-rate expression in (A30)—over the sum-rate of the fully distributed scheme of Slepian–Wolf [60] for computing (11). For the proposed setting, we next showcase the achievable gains  $\eta_{SW}$  of Proposition 3 via an example and showcase the results via Figure 6.



**Figure 6.** Gain  $10 \log_{10}(\eta_{SW})$  versus  $\epsilon$  for computing (11), where  $K_c = 1, \rho = 0, N_r = N - 1$ . (Left) The set of parameters  $N, K$ , and  $M$  are indicated for each configuration. (Right)  $10 \log_{10}(\eta_{SW})$  versus  $\epsilon$  to observe the effect of  $N$  for a fixed total cache size  $MN$  and fixed  $K$ .

#### 4.2. Distributed Computation of $K$ -Multi-Linear Functions over $\mathbb{F}_2$

We study the behaviors of  $\eta_{SW}$  versus the skewness parameter  $\epsilon$  for computing the multi-linear function given in (11) for i.i.d. uniform  $W_k \sim \text{Bern}(\epsilon), \epsilon \in [0, 1/2]$  across  $k \in [K]$ , 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 3 to determine the sum-rate upper bound and illustrate the gains  $10 \log_{10}(\eta_{SW})$  in decibels versus  $\epsilon$  in Figure 6.

From the numerical results in Figure 6 (Left), we observe that the sum-rate gain of the graph entropy-based approach versus the fully distributed approach of [60],  $\eta_{SW}$ , 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 for  $\eta_{SW}$  showcase that our proposed scheme can guarantee an exponential rate reduction over [60] as a function of decreasing  $\epsilon$ . Furthermore, the sum-rate gains scale linearly with the cache size  $M$ , which scales with  $K$  given  $N_r = N - 1$ . Note that  $\eta_{SW}$  diminishes with increasing  $N$  when  $M$  and  $\Delta$  are kept fixed. In Figure 6 (Right), for  $M \ll K$ , a fixed total cache size  $MN$ , and hence, fixed  $K$ , the gain  $\eta_{SW}$  for large  $N$  and small  $M$  is higher versus small  $N$  and large  $M$ , demonstrating the power of the graph-based approach as the topology becomes more and more distributed.

### 5. Conclusions

In this paper, we devised a distributed computation framework for general function classes in multi-server multi-function, single-user topologies. Specifically, we analyzed the upper bounds for the communication cost for computing in such topologies, exploiting Körner’s characteristic graph entropy, by incorporating the structures in the dataset and functions, as well as the dataset correlations. To showcase the achievable gains of our framework and perceive the roles of dataset statistics, correlations, and function classes, we performed several experiments under cyclic dataset placement over a field of characteristic two. Our numerical evaluations for distributed computing of linearly separable functions, as demonstrated in Section 4.1 via three scenarios, indicate that by incorporating dataset

correlations and skew, it is possible to achieve a very substantial reduction in the total communication cost over the state of the art. Similarly, for distributed computing of multi-linear functions, in Section 4.2, we demonstrate a very substantial reduction in the total communication cost versus the state of the art. Our main results (Theorem 1 and Propositions 1–3) and observations through the examples help us gain insights into reducing the communication cost of distributed computation by taking into account the structures of datasets (skew and correlations) and functions (characteristic graphs).

The potential directions include providing a tighter achievability result for Theorem 1 and devising a converse bound on the sum-rate. They involve conducting experiments under the scheme of the coded scheme of Maddah–Ali and Niesen detailed in [83] in order to capture the finer-grained granularity of placement that can help tighten the achievable rates. They also involve, beyond the special cases detailed in Propositions 1–3, exploring the achievable gains for a broader set of distributed computation scenarios, e.g., over-the-air computing, cluster computing, coded computing, distributed gradient descent, or more generally, distributed optimization and learning and goal-oriented and semantic communication frameworks, which can be reinforced by compression by capturing the skewness, correlations, and placement of datasets, the structures of functions, and topology.

**Author Contributions:** Conceptualization, D.M. and P.E.; methodology, P.E. and D.M.; software, D.M. and M.R.D.S.; validation, D.M., M.R.D.S. and B.S.; formal analysis, D.M.; investigation, D.M. and M.R.D.S.; resources, D.M.; data curation (not applicable); writing—original draft preparation, D.M. and M.R.D.S.; writing—review and editing, D.M., B.S. and M.R.D.S.; visualization, D.M. and M.R.D.S.; supervision, D.M. and P.E.; project administration, D.M. and P.E.; funding acquisition, D.M. and P.E. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was partially supported by a Huawei France-funded Chair towards Future Wireless Networks and supported by the program “PEPR Networks of the Future” of France 2030. Co-funded by the European Union (ERC, SENSIBILITÉ, 101077361, and ERC-PoC, LIGHT, 101101031). The 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.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data is contained within the article.

**Acknowledgments:** The authors thank Kai Wan at the Huazhong University of Science and Technology, Wuhan, China for interesting discussions.

**Conflicts of Interest:** The authors declare a conflict of interest with MIT, Northeastern, UT Austin, and Inria Paris research center due to academic relationships. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## Abbreviations

The following abbreviations are used in this manuscript:

ach	Achievable
Bern	Bernoulli
G	Graph
i.i.d.	Independent and identically distributed
lin	Linearly separable encoding
MIS	Maximal independent set
PMF	Probability mass function
SW	Slepian–Wolf encoding

### Appendix A. Technical Preliminary

Here, we detail the notion of characteristic graphs and their entropy in the context of source compression. We recall that the below proofs use the notation given in Section 1.5.

#### Appendix A.1. Distributed Source Compression, and Communication Cost

Given statistically dependent, finite-alphabet, i.i.d. random sequences  $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N$  where  $\mathbf{X}_i \in \mathbb{F}_q^{|\mathcal{Z}_i| \times n}$  for  $i \in \Omega$ , the Slepian–Wolf theorem gives a theoretical lower bound for the lossless coding rate of distributed servers in the limit as  $n$  goes to infinity. Denoting by  $R_i$  the encoding rate of server  $i \in \Omega$ , the sum-rate (or communication cost) for distributed source compression is given by

$$\sum_{i \in \mathcal{S}} R_i \geq H(X_{\mathcal{S}} | X_{\mathcal{S}^c}), \quad \mathcal{S} \subseteq \Omega, \tag{A1}$$

where  $\mathcal{S}$  denotes the indices of a subset of servers,  $\mathcal{S}^c = \Omega \setminus \mathcal{S}$  its complement, and  $X_{\mathcal{S}} = \{X_i, i \in \mathcal{S}\}$ .

We recall that in the case of distributed source compression, given by the coding theorem of Slepian–Wolf [60], the encoder mappings specify the bin indices for the server sequences  $\mathbf{X}_i$ . The bin index is such that every bin of each  $n$ -vector  $\mathbf{X}_i$  of server  $i \in \Omega$  is randomly drawn under the uniform distribution across the set  $\{0, 1, \dots, 2^{nR_i} - 1\}$  of  $2^{nR_i}$  bins. The transmission of server  $i \in \Omega$  is  $e_{X_i}(\mathbf{X}_i)$ , where  $e_{X_i} : \mathbf{X}_i \rightarrow \{0, 1, \dots, 2^{nR_i} - 1\}$  is the encoding function of  $i \in \Omega$  onto bins. The total number of symbols in  $e_{X_i}(\mathbf{X}_i)$  is  $T_i = H(e_{X_i}(\mathbf{X}_i))$ . This value corresponds to the aggregate number of symbols in the transmitted subfunctions from the server. Hence, the communication cost (rate) of  $i \in \Omega$  for a sufficiently large  $n$  satisfies

$$R_i = \frac{T_i}{L} = \frac{H(e_{X_i}(\mathbf{X}_i))}{n} \geq H(X_i), \tag{A2}$$

where the cost can be further reduced via a more efficient mapping  $e_{X_i}(\mathbf{X}_i)$  if  $W_k, k \in [K]$  are correlated.

#### Appendix A.2. Characteristic Graphs, Distributed Functional Compression, and Communication Cost

In this section, we provide a summary of key graph-theoretic points devised by Körner [75] and studied by Alon and Orlitsky [73] and Orlitsky and Roche [74] to understand the fundamental limits of distributed computation.

Let us consider the canonical scenario with two servers, storing  $X_1$  and  $X_2$ , respectively. The user requests a bivariate function  $F(X_1, X_2)$  that could be linearly separable or in general non-linear. Associated with the source pair  $(X_1, X_2)$  is a characteristic graph  $G$ , as defined by Witsenhausen [92]. We denote by  $G_{X_1} = (V_{G_{X_1}}, E_{G_{X_1}})$  the characteristic graph that server one builds (server two similarly builds  $G_{X_2}$ ) for computing (We detail the compression problem for the simultaneous computation of a set of requested functions in Appendix A.2.1.)  $F(X_1, X_2)$ , determined as a function of  $X_1, X_2$ , and  $F$ , where  $V_{G_{X_1}} = \mathcal{X}_1$  and an edge  $(x_1^1, x_1^2) \in E_{G_{X_1}}$  if and only if there exists a  $x_2^1 \in \mathcal{X}_2$  such that  $P_{X_1, X_2}(x_1^1, x_2^1) \cdot P_{X_1, X_2}(x_1^2, x_2^1) > 0$  and  $F(x_1^1, x_2^1) \neq F(x_1^2, x_2^1)$ . Note that the idea of building  $G_{X_1}$  can also be generalized to multivariate functions,  $F(X_{\Omega})$  where  $\Omega = [N]$  for  $N > 2$  [82]. In this paper, we only consider vertex colorings. A valid coloring of a graph  $G_{X_1}$  is such that each vertex of  $G_{X_1}$  is assigned a color (code) such that adjacent vertices receive disjoint colors (codes). Vertices that are not connected can be assigned to the same or different colors. The chromatic number  $\chi(G_{X_1})$  of a graph  $G_{X_1}$  is the minimum number of colors needed to have a valid coloring of  $G_{X_1}$  [76,77,79].

**Definition A1** (Characteristic graph entropy [73,75]). Given a random variable  $X_1$  with characteristic graph  $G_{X_1} = (V_{X_1}, E_{X_1})$  for computing function  $f(X_1, X_2)$ , the entropy of the characteristic graph is expressed as

$$H_{G_{X_1}}(X_1) = \min_{X_1 \in U_1 \in S(G_{X_1})} I(X_1; U_1), \tag{A3}$$

where  $S(G_{X_1})$  is the set of all MISs of  $G_{X_1}$ , where an MIS is not a subset of any other independent set, where an independent set of a graph is a set of its vertices in which no two vertices are adjacent [93]. 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  $G_{X_1}$ .

Similarly, the conditional graph entropy for  $X_1$  with characteristic graph  $G_{X_1}$  for computing  $f(X_1, X_2)$ , given  $X_2$  as side information, is defined in [74] using the notation  $U_1 - X_1 - X_2$  that indicates a Markov chain:

$$H_{G_{X_1}}(X_1 | X_2) = \min_{\substack{X_1 \in U_1 \in S(G_{X_1}) \\ U_1 - X_1 - X_2}} I(X_1; U_1 | X_2). \tag{A4}$$

The Markov chain relation in (A4) implies that  $H_{G_{X_1}}(X_1 | X_2) \leq H_{G_{X_1}}(X_1)$  [Ch. 2] [94]. In (A4), the goal is to determine the equivalence classes  $U_1$  of  $x_1^i \in X_1$  that have the same function outcome  $\forall x_2^j \in \mathcal{X}_2$  such that  $P_{X_1, X_2}(x_1^i, x_2^j) > 0$ . We next consider an example to clarify the distinction between characteristic graph entropy,  $H_{G_{X_1}}(X_1)$ , and entropy of a conditional characteristic graph, or conditional graph entropy,  $H_{G_{X_1}}(X_1 | X_2)$ .

**Example A1** (Characteristic graph entropy of ternary random variables [Examples 1–2] [74]). In this example, we first investigate the characteristic graph entropy,  $H_{G_{X_1}}(X_1)$ , and the conditional graph entropy,  $H_{G_{X_1}}(X_1 | X_2)$ .

- Let  $P_{X_1}$  be a uniform PMF over the set  $\{1, 2, 3\}$ . Assume that  $G_X$  has only one edge, i.e.,  $E_{X_1} = \{(1, 3)\}$ . Hence, the set of MISs is given as  $S(G_{X_1}) = \{\{1, 2\}, \{2, 3\}\}$ . To determine the entropy of a characteristic graph, i.e.,  $H_{G_{X_1}}(X_1)$ , from (A3), our objective is to minimize  $I(X_1; U_1)$ , which is a convex function of  $\mathbb{P}(U_1 | X_1)$ . Hence,  $I(X_1; U_1)$  is minimized when the conditional distribution of  $\mathbb{P}(U_1 | X_1)$  is selected as  $\mathbb{P}(U_1 = \{1, 2\} | X_1 = 1) = 1$ ,  $\mathbb{P}(U_1 = \{2, 3\} | X_1 = 3) = 1$ , and  $\mathbb{P}(U_1 = \{1, 2\} | X_1 = 2) = \mathbb{P}(U_1 = \{2, 3\} | X_1 = 2) = 1/2$ . As a result of this PMF, we have

$$H_{G_{X_1}}(X_1) = H(U_1) - H(U_1 | X_1) = 1 - \frac{1}{3} = \frac{2}{3}. \tag{A5}$$

- Let  $P_{X_1, X_2}$  be a uniform PMF over the set  $\{(x_1, x_2) : x_1, x_2 \in \{1, 2, 3\}, x_1 \neq x_2\}$  and  $E_{X_1} = \{(1, 3)\}$ . Note that  $H(X_1 | X_2) = 1$  given the joint PMF. To determine the conditional characteristic graph entropy, i.e.,  $H_{G_{X_1}}(X_1 | X_2)$ , using (A4), our objective is to minimize  $I(X_1; U_1 | X_2)$ , which is convex in  $\mathbb{P}(U_1 | X_1)$ . Hence,  $I(X_1; U_1 | X_2)$  is minimized when  $\mathbb{P}(U_1 | X_1)$  is selected as  $\mathbb{P}(U_1 = \{1, 2\} | X_1 = 1) = \mathbb{P}(U_1 = \{2, 3\} | X_1 = 3) = 1$ , and  $\mathbb{P}(U_1 = \{1, 2\} | X_1 = 2) = \mathbb{P}(U_1 = \{2, 3\} | X_1 = 2) = 1/2$ . Hence, we obtain

$$\begin{aligned} H(U_1 | X_2) &= \frac{1}{3}H(U_1 | X_1 \in \{2, 3\}) + \frac{1}{3}H(U_1 | X_1 \in \{1, 3\}) + \frac{1}{3}H(U_1 | X_1 \in \{1, 2\}) \\ &= \frac{1}{3}h\left(\frac{1}{4}\right) + \frac{1}{3} + \frac{1}{3}h\left(\frac{1}{4}\right), \end{aligned} \tag{A6}$$

which yields, using  $U_1 - X_1 - X_2$ , that

$$\begin{aligned} H_{G_{X_1}}(X_1 | X_2) &= H(U_1 | X_2) - H(U_1 | X_1, X_2) = H(U_1 | X_2) - H(U_1 | X_1) \\ &= \left[ \frac{1}{3}h\left(\frac{1}{4}\right) + \frac{1}{3} + \frac{1}{3}h\left(\frac{1}{4}\right) \right] - \frac{1}{3} = \frac{2}{3}h\left(\frac{1}{4}\right). \end{aligned} \tag{A7}$$

**Definition A2** (Chromatic entropy [73]). *The chromatic entropy of a graph  $G_{X_1}$  is defined as*

$$H_{G_{X_1}}^\chi(X_1) = \min_{c_{G_{X_1}}} H(c_{G_{X_1}}(X_1)), \tag{A8}$$

where the minimization is over the set of colorings such that  $c_{G_{X_1}}$  is a valid coloring of  $G_{X_1}$ .

Let  $G_{X_1}^n = (V_{X_1}^n, E_{X_1}^n)$  be the  $n$ -th OR power of a graph  $G_{X_1}$  for the source sequence  $X_1$  to compress  $F(X_1, X_2)$ . In this OR power graph,  $V_{X_1}^n = \mathcal{X}_1^n$  and  $(x_1^1, x_1^2) \in E_{X_1}^n$ , where  $x_1^1 = (x_{11}^1, x_{12}^1, \dots, x_{1n}^1)$  and similarly for  $x_1^2$ , when there exists at least one coordinate  $l \in [n]$  such that  $(x_{1l}^1, x_{1l}^2) \in E_{X_1}$ . We denote a coloring of  $G_{X_1}^n$  by  $c_{G_{X_1}^n}(X_1)$ . The encoding function at server one is a mapping from  $X_1$  to the colors  $c_{G_{X_1}^n}(X_1)$  of the characteristic graph  $G_{X_1}^n$  for computing  $F(X_1, X_2)$ . In other words,  $c_{G_{X_1}^n}(X_1)$  specifies the color classes of  $X_1$  such that each color class forms an independent set that induces the same function outcome.

Using Definition A2, we can determine the chromatic entropy of graph  $G_{X_1}^n$  as

$$H_{G_{X_1}^n}^\chi(X_1) = \min_{c_{G_{X_1}^n}} H(c_{G_{X_1}^n}(X_1)). \tag{A9}$$

In [75], Körner has shown the relation between the chromatic and graph entropies, which we detail next.

**Theorem A1** (Chromatic entropy versus graph entropy [75]). *The following relation holds between the characteristic graph entropy and the chromatic entropy of graph  $G_{X_1}^n$  in the limit of large  $n$ :*

$$H_{G_{X_1}}^\chi(X_1) = \lim_{n \rightarrow \infty} \frac{1}{n} H_{G_{X_1}^n}^\chi(X_1). \tag{A10}$$

Similarly, from (A9) and (A10), the conditional graph entropy of  $X_1$  given  $X_2$  is given as

$$H_{G_{X_1}}^\chi(X_1 | X_2) = \lim_{n \rightarrow \infty} \min_{c_{G_{X_1}^n}, c_{G_{X_2}^n}} \frac{1}{n} H(c_{G_{X_1}^n}(X_1) | c_{G_{X_2}^n}(X_2)). \tag{A11}$$

### Appendix A.2.1. A Characteristic-Graph-Based Encoding Framework for Simultaneously Computing a Set of Functions

The user demands a set of functions  $\{F_j(X_\Omega)\}_{j \in [K_c]} \in \mathbb{R}^{K_c}$  that are possibly non-linear in the subfunctions. In our proposed framework, for the distributed computing of these functions, we leverage characteristic graphs that can capture the structure of subfunctions. To determine the achievable rate of distributed lossless functional compression, we determine the colorings of these graphs and evaluate the entropy of such colorings. 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.

Union graphs for simultaneously computing a set of functions with side information have been considered in [82], using multi-functional characteristic graphs. A multi-functional characteristic graph is an OR function of individual characteristic graphs for different functions [Definition 45] [82]. 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  $E_{X_i}^\cup$ , which satisfies

$$G_{X_i}^\cup = \bigcup_{j \in [K_c]} G_{X_{i,j}} = (V_{X_i}, E_{X_i}^\cup), \quad E_{X_i}^\cup = \bigcup_{j \in [K_c]} E_{X_{i,j}}. \tag{A12}$$

In other words, we need to distinguish the outcomes  $x_i^1$  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  $F_j(x_i^1, x_{\Omega \setminus i}^1) \neq F_j(x_i^2, x_{\Omega \setminus i}^1)$ , for some  $P_{X_\Omega}(x_i^1, x_{\Omega \setminus i}^1) \cdot P_{X_\Omega}(x_i^2, x_{\Omega \setminus i}^1) > 0$  given  $x_{\Omega \setminus i}^1 \in X_{\Omega \setminus i}$ . The server then compresses the union  $G_{X_i}^\cup$  by exploiting (A9) and (A10).

In the special case when the number of demanded functions  $K_c$  is large (or tends to infinity), such that the union of all subspaces spanned by the independent sets of each  $G_{X_i,j}$ ,  $j \in [K_c]$  is the same as the subspace spanned by  $X_i$ , MISs of  $G_{X_i}^\cup$  in (A12) for server  $i \in \Omega$  become singletons, rendering  $G_{X_i}^\cup$  a complete graph. In this case, the problem boils down to the paradigm of distributed source compression (see Appendix A.1).

### Appendix A.2.2. Distributed Functional Compression

The fundamental limit of functional compression has been given by Körner [75]. Given  $\mathbf{X}_i \in \mathbb{F}_q^{|\mathcal{Z}_i| \times n}$  for server  $i \in \Omega$ , the encoding function  $e_{X_i}$  specifies MISs given by the valid colorings  $c_{G_{X_i}^n}(\mathbf{X}_i)$ . Let the number of symbols in  $\mathbf{Z}_i = g_i(\mathbf{X}_i) = e_{X_i}(c_{G_{X_i}^n}(\mathbf{X}_i))$  be  $T_i$  for server  $i \in \Omega$ . Hence, the communication cost of server  $i$ , as  $n \rightarrow \infty$  is given by (5).

Defining  $G_{X_S} = [G_{X_i}]_{i \in S}$  for a given subset  $S \subseteq \Omega$  chosen to guarantee distributed computation of  $F(X_\Omega)$ , i.e.,  $|S| \geq N_r$ , the sum-rate of servers for distributed lossless functional compression for computing  $F(\mathbf{X}_\Omega) = \{F(X_{1l}, X_{2l}, \dots, X_{Nl})\}_{l=1}^n$  equals

$$R_{\text{ach}} = \sum_{i \in S} R_i \geq H_{G_{X_S}}(X_S | Z_{S^c}), \quad S \subseteq \Omega, \tag{A13}$$

where  $H_{G_{X_S}}(X_S)$  is the joint graph entropy of  $S \subseteq \Omega$ , and it is defined as [Definition 30] [82]:

$$H_{G_{X_S}}(X_S) = \lim_{n \rightarrow \infty} \min_{\{c_{G_{X_i}^n}\}_{i \in S}} \frac{1}{n} H(c_{G_{X_i}^n}(\mathbf{X}_i), i \in S), \tag{A14}$$

where  $c_{G_{X_i}^n}(\mathbf{X}_i)$  is the coloring of the  $n$ -th power graph  $G_{X_i}^n$  that  $i \in \Omega$  builds for computing  $f(\mathbf{X}_\Omega)$  [82].

Similarly, exploiting [Definition 31] [82], the conditional graph entropy of the servers is given as

$$H_{G_{X_S}}(X_S | Z_{S^c}) = \lim_{n \rightarrow \infty} \min_{\{c_{G_{X_i}^n}\}_{i \in \Omega}} \frac{1}{n} H(c_{G_{X_i}^n}(\mathbf{X}_i), i \in S | e_{X_i}(c_{G_{X_i}^n}(\mathbf{X}_i)), i \in S^c). \tag{A15}$$

Using (A12), we jointly capture the structures of the set of demanded functions. Hence, this enables us to provide a refined communication cost model in (5) versus the characterizations as a function of  $K_c$ , see, e.g., [39,58,68].

## Appendix B. Proofs of Main Results

### Appendix B.1. Proof of Theorem 1

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 characteristic graph (The characteristic-graph-based approach is valid provided that each subfunction  $W_k$ ,  $k \in \mathcal{K}$  contained in  $X_i = W_{\mathcal{Z}_i}$  is defined over a  $q$ -ary field such that  $q \geq 2$ , to ensure that the union graph  $G_{X_i}^\cup$ ,  $i \in \Omega$  (or  $G_{X_i,j}$ ,  $j \in [K_c]$  each) has more than one vertex.)  $G_{X_i,j}$  for distributed lossless computing of  $f_j(W_{\mathcal{K}})$ ,  $j \in [K_c]$ . Similarly, server  $i \in \Omega$  builds a union characteristic graph for computing  $\{f_j(W_{\mathcal{K}})\}_{j \in [K_c]}$ . We denote by  $G_{X_i}^\cup = (V_{X_i}, E_{X_i}) = \cup_{j \in [K_c]} G_{X_i,j}$  the *union characteristic graph*,

given as in (A12). In the description of  $G_{X_i}^{\cup}$ , 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  $G_{X_{i,j}}$ , which is the characteristic graph the server builds for distributed lossless computing  $f_j(W_{\mathcal{K}})$  for a given function  $j \in [K_c]$ .

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, in the context of encoding characteristic graphs, the mapping from  $X_i$  to a valid coloring  $c_{G_{X_i}}(X_i)$ . We denote by  $Z_i = g_i(\mathbf{X}_i) = e_{X_i}(c_{G_{X_i}}(\mathbf{X}_i))$  the color encoding performed by server  $i \in \Omega$  for the length  $n$  realization of  $X_i$ , denoted by  $\mathbf{X}_i$ . For convenience, we use the following shorthand notation to represent the transmitted information from the server:

$$Z_i = g_i(X_i), \quad i \in \Omega. \tag{A16}$$

Combining the notions of the union graph in (A12) and the encodings of the individual servers given in (A16), the rate  $R_i$  needed from server  $i \in \Omega$  for meeting the user demand is upper bounded by the cost of the best encoding, which minimizes the rate of information transmission from the respective server. Equivalently,

$$R_i \geq \min_{Z_i = g_i(X_i) : g_i \in \mathcal{C}_i} H_{G_{X_i}^{\cup}}(X_i), \tag{A17}$$

where equality is achievable in (A17). Because the user can recover the desired functions using any set of  $N_r$  servers, the achievable sum rate is upper bounded by

$$R_{\text{ach}} \leq \sum_{i=1}^{N_r} \min_{Z_i = g_i(X_i) : g_i \in \mathcal{C}_i} H_{G_{X_i}^{\cup}}(X_i). \tag{A18}$$

Appendix B.2. Proof of Proposition 1

For the multi-server multi-function distributed computing architecture, this proposition restricts the demand to be a set of linearly separable functions, given as in (4). Given the recovery threshold  $N_r$ , it holds that

$$R_{\text{ach}} \leq \sum_{i=1}^{N_r} \min_{Z_i = g_i(X_i) : g_i \in \mathcal{C}_i} H_{G_{X_i}^{\cup}}(X_i) = \sum_{i=1}^{N_r} \min_{Z_i : g_i \in \mathcal{C}_i} \min_{X_i \in \mathcal{U}_i \in \mathcal{S}(G_{X_i}^{\cup})} I(X_i; U_i) \tag{A19}$$

$$= \sum_{i=1}^{N_r} \left[ H(W_{(i-1)\Delta+1}^{(i-1)\Delta+M}) - H(W_{(i-1)\Delta+1}^{(i-1)\Delta+M} \mid Z_i) \right] \tag{A20}$$

$$= \sum_{i=1}^{N_r} \left[ M - (M - H(Z_i)) \right] = \sum_{i=1}^{N_r} H(Z_i), \tag{A21}$$

where in (A19), we used the identity  $H_{G_{X_i}^{\cup}}(X_i) = \min_{X_i \in \mathcal{U}_i \in \mathcal{S}(G_{X_i}^{\cup})} I(X_i; U_i)$ . Furthermore, if the codebook  $\mathcal{C}_i$  is restricted to linear combinations of subfunctions,  $Z_i$  is given by the following set of linear equations:

$$Z_i = g_i(X_i) = \left\{ \sum_{k=(i-1)\Delta+1}^{(i-1)\Delta+M} \alpha_k^{(l)} W_k, l \in [K_c] \right\}. \tag{A22}$$

In other words,  $Z_i, i \in [N_r]$ , is a vector-valued function. Note that each server contributes to determining the set of linearly separable functions  $\{f_j(W_{\mathcal{K}}), j \in [K_c]\}$  of datasets, given as in (4), in a distributed manner. Hence, each independent set  $\mathcal{U}_i \in \mathcal{S}(G_{X_i}^{\cup})$ ,

with  $S(G_{X_i}^{\cup})$  denoting the set of MISs of  $X_i$ , of  $X_i$  is captured by the linear functions of  $\{W_k\}_{k \in [(i-1)\Delta+1:(i-1)\Delta+M]}$ , i.e., each  $U_i \in S(G_{X_i}^{\cup})$  is determined by (A22). Hence, the user can recover the requested functions by linearly combining the transmissions of the  $N_r$  servers:

$$f_j(W_{\mathcal{K}}) = \sum_{i=1}^{N_r} \beta_{ji} Z_i = \sum_{i=1}^{N_r} \beta_{ji} g_i(X_i) = \sum_{k=1}^K \gamma_{jk} W_k, \quad j \in [K_c]. \tag{A23}$$

In (A20), we use the definition of mutual information,  $I(X_i; U_i) = H(X_i) - H(X_i | U_i)$ , where given  $i \in [N_r]$  and  $\Delta = \frac{K}{N}$ , it holds under cyclic placement that

$$X_i = W_{(i-1)\Delta+1}^{(i-1)\Delta+M} = W_{(i-1)\Delta+1}, W_{(i-1)\Delta+2}, \dots, W_{(i-1)\Delta+M}, \tag{A24}$$

and  $\alpha_k^{(l)}$  are the coefficients for computing function  $l \in [K_c]$ . In (A21), we used the fact that  $W_k$  is uniform over  $\mathbb{F}_q$  and i.i.d. across  $k \in [K]$  and rewrote the conditional entropy expression such that

$$H\left(W_{(i-1)\Delta+1}^{(i-1)\Delta+M} \mid Z_i\right) = H\left(W_{(i-1)\Delta+1}^{(i-1)\Delta+M}, Z_i\right) - H(Z_i) \stackrel{(a)}{=} H\left(W_{(i-1)\Delta+1}^{(i-1)\Delta+M}\right) - H(Z_i), \tag{A25}$$

where (a) follows from the fact that  $Z_i$  is a function of  $W_{(i-1)\Delta+1}^{(i-1)\Delta+M}$ . For a given  $l \in [K_c]$  and field size  $q$ , 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 the fact 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

$$\sum_{i=1}^{N_r} \min_{Z_i: g_i \in \mathcal{C}_i} H_{G_{X_i}^{\cup}}(X_i) \leq K_c N_r. \tag{A26}$$

Exploiting the cyclic placement model, we can tighten the bound in (A26). Note that server  $i = 1$  can help recover  $M$  subfunctions (at most, i.e.,  $M$  transmissions needed to recover  $M$  subfunctions), and each of the servers  $i \in [2 : N_r]$  can help recover an additional  $\Delta$  subfunctions (at most, i.e.,  $\Delta$  transmissions are needed to recover  $\Delta$  subfunctions). Hence, the set of servers  $[N_r]$  suffices to provide  $M + (N_r - 1)\Delta = N\Delta = K$  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. Now, let us consider the following four scenarios:

- (i) When  $1 \leq K_c < \Delta$ , it is sufficient for each server to transmit  $K_c$  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  $K$  subfunctions.
- (ii) When  $\Delta \leq K_c \leq \Delta N_r$ , it is sufficient for each server to transmit at most  $\Delta$  linearly independent combinations of their subfunctions. This leads to resolving  $\Delta N_r$  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  $\frac{K_c}{N_r} \leq \frac{K}{N_r} = \Delta \left(\frac{N_r + N - N_r}{N_r}\right) = \Delta + \Delta \left(\frac{N - N_r}{N_r}\right)$ , 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  $\Delta \left(\frac{N - N_r}{N_r}\right)$  defines 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 since any  $K$  linearly independent equation in (A23) suffices to recover  $W_{\mathcal{K}}$ , the sum-rate  $K$  is achievable.

From (i)–(iv), we obtain the following upper bound on the achievable sum-rate:

$$\sum_{i=1}^{N_r} \min_{Z_i: g_i \in \mathcal{C}_i} H_{G_{X_i}^U}(X_i) = \begin{cases} K_c N_r, & 1 \leq K_c < \Delta, \\ \Delta N_r, & \Delta \leq K_c \leq \Delta N_r, \\ K_c, & \Delta N_r < K_c \leq K, \\ K, & K < K_c, \end{cases} \quad (\text{A27})$$

where it is easy to note that (A27) matches the communication cost in [Theorem 2] [39]. The i.i.d. distribution assumption for  $W_k$  ensures that this result holds for any  $q \geq 2$ .

*Appendix B.3. Proof of Proposition 2*

Similarly as in Theorem 1, we let  $G_{X_i}^U = \bigcup_{j \in [K_c]} G_{X_i, j}$  denote the *union characteristic graph* that server  $i \in \Omega$  builds for computing  $\{f_j(W_{\mathcal{K}})\}_{j \in [K_c]}$ . Note that given  $W_{\mathcal{Z}_i} = W_{1+(i-1)\Delta}, W_{1+(i-1)\Delta+1}, \dots, W_{M+(i-1)\Delta}$ , the support set of server  $i \in \Omega$  has a cardinality of  $\mathcal{X}_i = 2^M$ . Because the user demand is a collection of Boolean functions, in this scenario, each server  $i \in \Omega$  builds a graph with two independent sets at most, denoted by  $s_0(G_{X_i}^U)$  and  $s_1(G_{X_i}^U)$ , yielding the function values  $Z_i = 0$  and  $Z_i = 1$ , respectively.

Given the recovery threshold  $N_r$ , any subset  $\mathcal{S}$  of servers with  $|\mathcal{S}| = N_r$  stores the set  $\mathcal{K}$ , which is sufficient to compute the demanded functions. Given server  $i \in \Omega$ , consider the set of all  $w_{\mathcal{Z}_i} \in W_{\mathcal{Z}_i}$ , which satisfies

$$f(w_{\mathcal{Z}_i}, w_{\mathcal{Z}_S \setminus \mathcal{Z}_i}) = 1, \quad \forall w_{\mathcal{Z}_S \setminus \mathcal{Z}_i} \in \{0, 1\}^{|\mathcal{K} \setminus \mathcal{Z}_i|}, \quad (\text{A28})$$

where notation  $w_{\mathcal{Z}_S \setminus \mathcal{Z}_i}$  denotes the dataset values for the set of datasets stored in the subset of servers  $\mathcal{S} \setminus i$ . Note, in general, that  $K_n(\mathcal{S}) = |\mathcal{Z}_S| = |\bigcup_{i \in \mathcal{S}} \mathcal{Z}_i|$ . In the case of cyclic placement based on (1), out of the set of all datasets  $\mathcal{K}$ , there are  $\Delta$  datasets that belong exclusively to server  $i \in \Omega$ . In this case,  $|\mathcal{K} \setminus \mathcal{Z}_i| = K - \Delta$ .

Note that (A28) captures the independent set  $s_1(G_{X_i}^U) \ni w_{\mathcal{Z}_i}$ . Equivalently, the set of dataset values  $W_{\mathcal{Z}_i}$  that lands in  $s_1(G_{X_i}^U)$  of  $G_{X_i}^U$  yields  $Z_i = 1$ . The transmitted information takes the value  $Z_i = 1$  with a probability

$$\mathbb{P}(Z_i = 1) = \mathbb{P}(W_{\mathcal{Z}_i} \in s_1(G_{X_i}^U)), \quad i \in \Omega, \quad (\text{A29})$$

using which the upper bound on the achievable sum rate can be determined.

*Appendix B.4. Proof of Proposition 3*

Recall that  $W_k \sim \text{Bern}(\epsilon)$  are i.i.d. across  $k \in [K]$ , 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 multi-linear function  $f(W_{\mathcal{K}}) = \prod_{k \in [K]} W_k$  is  $N^* = \lfloor \frac{N}{N - N_r + 1} \rfloor$  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 variables  $X_{\Omega} \setminus X_1 = X_2, \dots, X_N$  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 (12), 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

$$\begin{aligned}
\sum_{i \in \mathcal{N}^*} R_i &\stackrel{(a)}{\leq} H_{G_{X_{i_1}}} (X_{i_1}) + H_{G_{X_{i_2}}} (X_{i_2} | Z_{i_1}) + \cdots + H_{G_{X_{i_{N^*}}}} (X_{i_{N^*}} | Z_{i_1}, Z_{i_2}, \dots, Z_{i_{N^*-1}}) \\
&\stackrel{(b)}{=} h(\epsilon_M) + \epsilon_M h(\epsilon_M) + (\epsilon_M)^2 h(\epsilon_M) + \cdots + (\epsilon_M)^{N^*-1} h(\epsilon_M) \\
&\stackrel{(c)}{=} \frac{1 - (\epsilon_M)^{N^*}}{1 - \epsilon_M} \cdot h(\epsilon_M),
\end{aligned} \tag{A30}$$

where (a) follows from assuming  $\mathcal{S} = \{i_1, i_2, \dots, i_{N^*}\}$  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 sum of the terms in the geometric series, i.e.,  $\sum_{l=0}^{N^*-1} (\epsilon_M)^l = \frac{1 - (\epsilon_M)^{N^*}}{1 - \epsilon_M}$  (While Proposition 3 uses the conditional graph entropies, the statements of Theorem 1 and Proposition 1, and Proposition 2 do not take into account the notion of conditional graph entropies. However, as indicated in Section 4.1 for computing linearly separable functions, and in Section 4.2 for computing multi-linear functions, respectively, we used the conditional entropy-based sum rate in (A30) to evaluate and illustrate the achievable gains over [39,60].).

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 additional servers  $\mathcal{I}^* \in \Omega$  to aid the computation and determine the outcome of  $f(W_K)$  by computing the product of the remaining  $\zeta_N$  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  $f(W_K)$  by computing the product of subfunctions  $W_k \sim \text{Bern}(\epsilon)$ ,  $k \in [N - \zeta_N + 1 : N]$  that cannot be captured by the previous  $N^*$  servers. Hence, the additional rate, given by the second term in (12), is given by the product of the term

$$(\epsilon_M)^{N^*} = \mathbb{P}\left(\prod_{i \in \mathcal{S}} \prod_{k \in \mathcal{Z}_i} W_k = 1\right), \tag{A31}$$

with  $1_{\Delta_N > 0}$ , and  $h(\epsilon_{\zeta_N})$ . Combining this rate term with (A30), we prove the statement of the proposition.

## References

1. Yang, C.; Wu, H.; Huang, Q.; Li, Z.; Li, J. Using spatial principles to optimize distributed computing for enabling the physical science discoveries. *Proc. Natl. Acad. Sci. USA* **2011**, *108*, 5498–5503. [\[CrossRef\]](#)
2. Shamsi, J.; Khojaye, M.A.; Qasmi, M.A. Data-intensive cloud computing: Requirements, expectations, challenges, and solutions. *J. Grid Comput.* **2013**, *11*, 281–310. [\[CrossRef\]](#)
3. Yang, H.; Ding, T.; Yuan, X. Federated Learning With Lossy Distributed Source Coding: Analysis and Optimization. *IEEE Trans. Commun.* **2023**, *71*, 4561–4576. [\[CrossRef\]](#)
4. Gan, G. Evaluation of room air distribution systems using computational fluid dynamics. *Energy Build.* **1995**, *23*, 83–93. [\[CrossRef\]](#)
5. Gao, Y.; Wang, L.; Zhou, J. Cost-efficient and quality of experience-aware provisioning of virtual machines for multiplayer cloud gaming in geographically distributed data centers. *IEEE Access* **2019**, *7*, 142574–142585. [\[CrossRef\]](#)
6. Lushbough, C.; Brendel, V. An overview of the BioExtract Server: A distributed, Web-based system for genomic analysis. In *Advances in Computational Biology*; Springer: New York, NY, USA, 2010; pp. 361–369.
7. Dean, J.; Ghemawat, S. MapReduce: Simplified data processing on large clusters. *Commun. ACM* **2008**, *51*, 107–113. [\[CrossRef\]](#)
8. Grolinger, K.; Hayes, M.; Higashino, W.A.; L'Heureux, A.; Allison, D.S.; Capretz, M.A. Challenges for MapReduce in Big Data. In Proceedings of the IEEE World Congress Services, Anchorage, AK, USA, 27 June–2 July 2014; pp. 182–189.
9. Al-Khasawneh, M.A.; Shamsuddin, S.M.; Hasan, S.; Bakar, A.A. MapReduce a Comprehensive Review. In Proceedings of the International Conference on Smart Computing and Electronic Enterprise (ICSCEE), Shah Alam, Malaysia, 11–12 July 2018; pp. 1–6.
10. Zaharia, M.; Chowdhury, M.; Franklin, M.J.; Shenker, S.; Stoica, I. Spark: Cluster computing with working sets. In Proceedings of the USENIX Workshop on Hot Topics in Cloud Computing, Boston, MA, USA, 22 June 2010.
11. Khumoyun, A.; Cui, Y.; Hanku, L. Spark based distributed deep learning framework for big data applications. In Proceedings of the International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, Uzbekistan, 2–4 November 2016; pp. 1–5.

12. Orgerie, A.C.; Assuncao, M.D.d.; Lefevre, L. A survey on techniques for improving the energy efficiency of large-scale distributed systems. *ACM Comput. Surv.* **2014**, *46*, 1–31. [[CrossRef](#)]
13. Keralapura, R.; Cormode, G.; Ramamirtham, J. Communication-Efficient Distributed Monitoring of Thresholded Counts. In Proceedings of the ACM SIGMOD International Conference on Management of Data, New York, NY, USA, 27–29 June 2006; pp. 289–300.
14. Li, W.; Chen, Z.; Wang, Z.; Jafar, S.A.; Jafarkhani, H. Flexible constructions for distributed matrix multiplication. In Proceedings of the IEEE International Symposium on Information Theory (ISIT), Melbourne, Australia, 12–20 July 2021; pp. 1576–1581.
15. Liu, Y.; Yu, F.R.; Li, X.; Ji, H.; Leung, V.C. Distributed resource allocation and computation offloading in fog and cloud networks with non-orthogonal multiple access. *IEEE Trans. Veh. Tech.* **2018**, *67*, 12137–12151. [[CrossRef](#)]
16. Noormohammadpour, M.; Raghavendra, C.S. Datacenter traffic control: Understanding techniques and tradeoffs. *IEEE Commun. Surv. Tutor.* **2017**, *20*, 1492–1525. [[CrossRef](#)]
17. Shivaratri, N.; Krueger, P.; Singhal, M. Load distributing for locally distributed systems. *Computer* **1992**, *25*, 33–44. [[CrossRef](#)]
18. Bestavros, A. Demand-based document dissemination to reduce traffic and balance load in distributed information systems. In Proceedings of the IEEE Symposium on Parallel and Distributed Processing, San Antonio, TX, USA, 25–28 October 1995; pp. 338–345.
19. Reisizadeh, A.; Prakash, S.; Pedarsani, R.; Avestimehr, A.S. Tree Gradient Coding. In Proceedings of the IEEE International Symposium on Information Theory (ISIT), Paris, France, 7–12 July 2019; pp. 2808–2812.
20. Ozfatura, E.; Gündüz, D.; Ulukus, S. Gradient Coding with Clustering and Multi-Message Communication. In Proceedings of the IEEE Data Science Workshop, Minneapolis, MN, USA, 2–7 June 2019; pp. 42–46.
21. Tandon, R.; Lei, Q.; Dimakis, A.G.; Karampatziakis, N. Gradient Coding: Avoiding Stragglers in Distributed Learning. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 31 July–3 August 2017.
22. Ye, M.; Abbe, E. Communication-computation efficient gradient coding. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 5610–5619.
23. Halbawi, W.; Azizan, N.; Salehi, F.; Hassibi, B. Improving Distributed Gradient Descent Using Reed-Solomon Codes. In Proceedings of the IEEE International Symposium on Information Theory (ISIT), Vail, CO, USA, 17–22 June 2018; pp. 2027–2031.
24. Maddah-Ali, M.A.; Niesen, U. Fundamental limits of caching. In Proceedings of the IEEE International Symposium on Information Theory (ISIT), Istanbul, Türkiye, 7–12 July 2013; pp. 1077–1081.
25. Karamchandani, N.; Niesen, U.; Maddah-Ali, M.A.; Diggavi, S.N. Hierarchical coded caching. *IEEE Trans. Info Theory* **2016**, *62*, 3212–3229. [[CrossRef](#)]
26. Li, S.; Supittayapornpong, S.; Maddah-Ali, M.A.; Avestimehr, S. Coded TeraSort. In Proceedings of the IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Lake Buena Vista, FL, USA, 29 May–2 June 2017.
27. Li, S.; Maddah-Ali, M.A.; Yu, Q.; Avestimehr, A.S. A fundamental tradeoff between computation and communication in distributed computing. *IEEE Trans. Inf. Theory* **2017**, *64*, 109–128. [[CrossRef](#)]
28. Yu, Q.; Maddah-Ali, M.A.; Avestimehr, A.S. The exact rate-memory tradeoff for caching with uncoded prefetching. *IEEE Trans. Inf. Theory* **2018**, *64*, 1281–1296. [[CrossRef](#)]
29. Naderializadeh, N.; Maddah-Ali, M.A.; Avestimehr, A.S. Fundamental limits of cache-aided interference management. *IEEE Trans. Inf. Theory* **2017**, *63*, 3092–3107.
30. Subramaniam, A.M.; Heidarzadeh, A.; Narayanan, K.R. Collaborative decoding of polynomial codes for distributed computation. In Proceedings of the IEEE Information Theory Workshop (ITW), Visby, Sweden, 25–28 August 2019; pp. 1–5. [[CrossRef](#)]
31. Dutta, S.; Fahim, M.; Haddadpour, F.; Jeong, H.; Cadambe, V.; Grover, P. On the optimal recovery threshold of coded matrix multiplication. *IEEE Trans. Inf. Theory* **2019**, *66*, 278–301.
32. Yosibash, R.; Zamir, R. Frame codes for distributed coded computation. In Proceedings of the International Symposium on Topics in Coding, Montreal, QC, Canada, 18–21 August 2021; pp. 1–5. [[CrossRef](#)]
33. Dimakis, A.G.; Godfrey, P.B.; Wu, Y.; Wainwright, M.J.; Ramchandran, K. Network coding for distributed storage systems. *IEEE Trans. Inf. Theory* **2010**, *56*, 4539–4551. [[CrossRef](#)]
34. Wan, K.; Sun, H.; Ji, M.; Tuninetti, D.; Caire, G. Cache-aided matrix multiplication retrieval. *IEEE Trans. Inf. Theory* **2022**, *68*, 4301–4319. [[CrossRef](#)]
35. Jia, Z.; Jafar, S.A. On the capacity of secure distributed batch matrix multiplication. *IEEE Trans. Inf. Theory* **2021**, *67*, 7420–7437. [[CrossRef](#)]
36. Soleymani, M.; Mahdavi, H.; Avestimehr, A.S. Analog lagrange coded computing. *IEEE J. Sel. Areas Inf. Theory* **2021**, *2*, 283–295.
37. Yu, Q.; Maddah-Ali, M.A.; Avestimehr, S. Polynomial codes: An optimal design for high-dimensional coded matrix multiplication. In Proceedings of the International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 4403–4413.
38. López, H.H.; Matthews, G.L.; Valvo, D. Secure MatDot codes: A secure, distributed matrix multiplication scheme. In Proceedings of the IEEE Information Theory Workshop (ITW), Mumbai, India, 6–9 November 2022; pp. 149–154. [[CrossRef](#)]
39. Wan, K.; Sun, H.; Ji, M.; Caire, G. Distributed linearly separable computation. *IEEE Trans. Inf. Theory* **2021**, *68*, 1259–1278. [[CrossRef](#)]
40. Zhu, J.; Li, S.; Li, J. Information-theoretically private matrix multiplication from MDS-coded storage. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 1680–1695. [[CrossRef](#)]

41. Das, A.B.; Ramamoorthy, A.; Vaswani, N. Efficient and Robust Distributed Matrix Computations via Convolutional Coding. *IEEE Trans. Inf. Theory* **2021**, *67*, 6266–6282. [[CrossRef](#)]
42. Yu, Q.; Maddah-Ali, M.A.; Avestimehr, A.S. Straggler Mitigation in Distributed Matrix Multiplication: Fundamental Limits and Optimal Coding. *IEEE Trans. Inf. Theory* **2020**, *66*, 1920–1933. [[CrossRef](#)] [[PubMed](#)]
43. Fawzi, A.; Balog, M.; Huang, A.; Hubert, T.; Romera-Paredes, B.; Barekatin, M.; Novikov, A.; R Ruiz, F.J.; Schrittwieser, J.; Swirszcz, G.; et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature* **2022**, *610*, 47–53. [[CrossRef](#)]
44. Aliasgari, M.; Simeone, O.; Kliewer, J. Private and secure distributed matrix multiplication with flexible communication load. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 2722–2734.
45. D'Oliveira, R.G.; El Rouayheb, S.; Heinlein, D.; Karpuk, D. Notes on communication and computation in secure distributed matrix multiplication. In Proceedings of the IEEE Conference on Communications and Network Security, Virtual, 29 June–1 July 2020; pp. 1–6. [[CrossRef](#)]
46. Rashmi, K.V.; Shah, N.B.; Kumar, P.V. Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction. *IEEE Trans. Inf. Theory* **2011**, *57*, 5227–5239.
47. Cancès, E.; Friesecke, G. *Density Functional Theory: Modeling, Mathematical Analysis, Computational Methods, and Applications*, 1st ed.; Springer Nature: Berlin/Heidelberg, Germany, 2023. [[CrossRef](#)]
48. Hanna, O.A.; Ezzeldin, Y.H.; Sadjadpour, T.; Fragouli, C.; Diggavi, S. On distributed quantization for classification. *IEEE J. Sel. Areas Inf. Theory* **2020**, *1*, 237–249.
49. Luo, P.; Xiong, H.; Lü, K.; Shi, Z. Distributed classification in peer-to-peer networks. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, CA, USA, 12–15 August 2007; pp. 968–976.
50. Karakus, C.; Sun, Y.; Diggavi, S.; Yin, W. Straggler mitigation in distributed optimization through data encoding. In Proceedings of the International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5434–5442. [[CrossRef](#)]
51. Jia, Z.; Jafar, S.A. Cross subspace alignment codes for coded distributed batch computation. *IEEE Trans. Inf. Theory* **2021**, *67*, 2821–2846.
52. Wang, J.; Jia, Z.; Jafar, S.A. Price of Precision in Coded Distributed Matrix Multiplication: A Dimensional Analysis. In Proceedings of the IEEE Information Theory Workshop (ITW), Kanazawa, Japan, 17–21 October 2021; pp. 1–6.
53. Chang, W.T.; Tandon, R. On the capacity of secure distributed matrix multiplication. In Proceedings of the IEEE Global Communications Conference, Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6.
54. Monagan, M.; Pearce, R. Parallel sparse polynomial multiplication using heaps. In Proceedings of the International Symposium on Symbolic and Algebraic Computation, Seoul, Republic of Korea, 28–31 July 2009; pp. 263–270.
55. Hsu, C.D.; Jeong, H.; Pappas, G.J.; Chaudhari, P. Scalable reinforcement learning policies for multi-agent control. In Proceedings of the IEEE/RISJ International Conference on Intelligent Robots and Systems, Prague, Czech Republic, 27 September 27–1 October 2021; pp. 4785–4791. [[CrossRef](#)]
56. Goldenbaum, M.; Boche, H.; Stańczak, S. Nomographic functions: Efficient computation in clustered Gaussian sensor networks. *IEEE Trans. Wirel. Commun.* **2014**, *14*, 2093–2105. [[CrossRef](#)]
57. Goldenbaum, M.; Boche, H.; Stańczak, S. Harnessing interference for analog function computation in wireless sensor networks. *IEEE Trans. Signal Process.* **2013**, *61*, 4893–4906.
58. Huang, W.; Wan, K.; Sun, H.; Ji, M.; Qiu, R.C.; Caire, G. Fundamental Limits of Distributed Linearly Separable Computation under Cyclic Assignment. In Proceedings of the IEEE International Symposium on Information Theory (ISIT'23), Taipei, Taiwan, 25–30 June 2023. [[CrossRef](#)]
59. Wan, K.; Sun, H.; Ji, M.; Caire, G. On Secure Distributed Linearly Separable Computation. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 912–926. [[CrossRef](#)]
60. Slepian, D.; Wolf, J.K. Noiseless coding of correlated information sources. *IEEE Trans. Inf. Theory* **1973**, *19*, 471–480. [[CrossRef](#)]
61. Cover, T. A proof of the data compression theorem of Slepian and Wolf for ergodic sources. *IEEE Trans. Inf. Theory* **1975**, *21*, 226–228. [[CrossRef](#)]
62. Korner, J.; Marton, K. How to encode the modulo-two sum of binary sources. *IEEE Trans. Inf. Theory* **1979**, *25*, 219–221. [[CrossRef](#)]
63. Lalitha, V.; Prakash, N.; Vinodh, K.; Kumar, P.V.; Pradhan, S.S. Linear coding schemes for the distributed computation of subspaces. *IEEE J. Sel. Areas Commun.* **2013**, *31*, 678–690. [[CrossRef](#)]
64. Yamamoto, H. Wyner-Ziv theory for a general function of the correlated sources. *IEEE Trans. Inf. Theory* **1982**, *28*, 803–807. [[CrossRef](#)]
65. Wyner, A.; Ziv, J. The rate-distortion function for source coding with side information at the decoder. *IEEE Trans. Inf. Theory* **1976**, *22*, 1–10. [[CrossRef](#)]
66. Wan, K.; Sun, H.; Ji, M.; Tuninetti, D.; Caire, G. Cache-Aided General Linear Function Retrieval. *Entropy* **2020**, *23*, 25. [[CrossRef](#)]
67. Khalesi, A.; Elia, P. Multi-User Linearly-Separable Distributed Computing. *IEEE Trans. Inf. Theory* **2023**, *69*, 6314–6339. [[CrossRef](#)]
68. Wan, K.; Sun, H.; Ji, M.; Caire, G. On the Tradeoff Between Computation and Communication Costs for Distributed Linearly Separable Computation. *IEEE Trans. Commun.* **2021**, *69*, 7390–7405. [[CrossRef](#)] [[PubMed](#)]
69. Erickson, B.J.; Korfiatis, P.; Akkus, Z.; Kline, T.L. Machine learning for medical imaging. *Radiographics* **2017**, *37*, 505–515. [[CrossRef](#)] [[PubMed](#)]

70. Correa, N.M.; Adali, T.; Li, Y.O.; Calhoun, V.D. Canonical Correlation Analysis for Data Fusion and Group Inferences. *IEEE Signal Process. Mag.* **2010**, *27*, 39–50. [[CrossRef](#)]
71. Kant, G.; Sangwan, K.S. Predictive modeling for power consumption in machining using artificial intelligence techniques. *Procedia CIRP* **2015**, *26*, 403–407. [[CrossRef](#)]
72. Han, T.; Kobayashi, K. A dichotomy of functions  $F(X, Y)$  of correlated sources  $(X, Y)$ . *IEEE Trans. Inf. Theory* **1987**, *33*, 69–76. [[CrossRef](#)]
73. Alon, N.; Orlicsky, A. Source coding and graph entropies. *IEEE Trans. Inf. Theory* **1996**, *42*, 1329–1339. [[CrossRef](#)]
74. Orlicsky, A.; Roche, J.R. Coding for computing. *IEEE Trans. Inf. Theory* **2001**, *47*, 903–917.
75. Körner, J. Coding of an information source having ambiguous alphabet and the entropy of graphs. In Proceedings of the Prague Conference on Information Theory, Prague, Czech Republic, 19–25 September 1973.
76. Malak, D. Fractional Graph Coloring for Functional Compression with Side Information. In Proceedings of the IEEE Information Theory Workshop (ITW), Mumbai, India, 6–9 November 2022.
77. Malak, D. Weighted graph coloring for quantized computing. In Proceedings of the IEEE International Symposium on Information Theory (ISIT), Taipei, Taiwan, 25–30 June 2023; pp. 2290–2295.
78. Charpenay, N.; Le Treust, M.; Roumy, A. Complementary Graph Entropy, AND Product, and Disjoint Union of Graphs. In Proceedings of the IEEE International Symposium on Information Theory (ISIT), Taipei, Taiwan, 25–30 June 2023; pp. 2446–2451.
79. Deylam Salehi, M.R.; Malak, D. An Achievable Low Complexity Encoding Scheme for Coloring Cyclic Graphs. In Proceedings of the Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, 26–29 September 2023; pp. 1–8.
80. Maugey, T.; Rizkallah, M.; Mahmoudian Bidgoli, N.; Roumy, A.; Guillemot, C. Graph Spectral 3D Image Compression. In *Graph Spectral Image Processing*; Wiley: Hoboken, NJ, USA, 2021; pp. 105–128. [[CrossRef](#)] [[PubMed](#)]
81. Sevilla, J.L.; Segura, V.; Podhorski, A.; Gुरुceaga, E.; Mato, J.M.; Martinez-Cruz, L.A.; Corrales, F.J.; Rubio, A. Correlation between gene expression and GO semantic similarity. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2005**, *2*, 330–338. [[CrossRef](#)]
82. Feizi, S.; Médard, M. On network functional compression. *IEEE Trans. Inf. Theory* **2014**, *60*, 5387–5401. [[CrossRef](#)]
83. Maddah-Ali, M.A.; Niesen, U. Fundamental limits of caching. *IEEE Trans. Inf. Theory* **2014**, *60*, 2856–2867. [[CrossRef](#)]
84. Mosk-Aoyama, D.; Shah, D. Fast Distributed Algorithms for Computing Separable Functions. *IEEE Trans. Info. Theory* **2008**, *54*, 2997–3007. [[CrossRef](#)]
85. Kaur, G. A comparison of two hybrid ensemble techniques for network anomaly detection in spark distributed environment. *J. Inform. Secur. Appl.* **2020**, *55*, 102601.
86. Chen, J.; Li, J.; Huang, R.; Yue, K.; Chen, Z.; Li, W. Federated learning for bearing fault diagnosis with dynamic weighted averaging. In Proceedings of the International Conference on Sensing, Measurement & Data Analytics in the era of Artificial Intelligence, Nanjing, China, 21–23 October 2021; pp. 1–6.
87. Zhao, J.; Govindan, R.; Estrin, D. Computing aggregates for monitoring wireless sensor networks. In Proceedings of the IEEE International Workshop on Sensor Network Protocols and Applications, Anchorage, AK, USA, 1 January 2003; pp. 139–148.
88. Giselsson, P.; Rantzer, A. *Large-Scale and Distributed Optimization: An Introduction*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2018; Volume 2227.
89. Kavadias, S.; Chao, R.O. *Resource Allocation and New Product Development Portfolio Management*, 1st ed.; Elsevier: Amsterdam, The Netherlands; Butterworth-Heinemann: Oxford, UK, 2007; pp. 135–163. [[CrossRef](#)]
90. Diniz, C.A.R.; Tutia, M.H.; Leite, J.G. Bayesian analysis of a correlated binomial model. *Braz. J. Probab. Stat.* **2010**, *24*, 68–77.
91. Boland, P.J.; Proschan, F.; Tong, Y. Some majorization inequalities for functions of exchangeable random variables. *Lect. Not.-Mono. Ser.* **1990**, 85–91. [[CrossRef](#)]
92. Witsenhausen, H. The zero-error side information problem and chromatic numbers (corresp.). *IEEE Trans. Inf. Theory* **1976**, *22*, 592–593. [[CrossRef](#)]
93. Moon, J.W.; Moser, L. On cliques in graphs. *Israel J. Math.* **1965**, *3*, 23–28.
94. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*, 2nd ed.; John Wiley & Sons: New York, NY, USA, 1991.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.