SORBONNE UNIVERSITÉ
CRÉATEURS DE FUTURS DEPUIS 1257

EURECOM
Sophia Antipolis

DOCTORAL THESIS
SORBONNE UNIVERSITY

Scientific Field:

Information and Communication Sciences and Technologies

Doctoral School:

Computer Science, Telecommunications, and Electronics of Paris

Presented by

Mohammad Reza Deylam Salehi

To obtain the degree of

Doctor of Philosophy

Thesis Title:

Fundamental Limits of Distributed Non-Linear Function Computation in Several Multi-User Network Models

Defense date 05/12/2025

Reviewers:

Prof. Maël Le Treust, CNRS, Inria, Rennes, France

Prof. Michéle Wigger, Télécom Paris, Paris, France

Examiners:

Prof. Photios A. Stavrou, EURECOM, Sophia Antipolis, France

Prof. Giovanni Neglia, Sophia Antipolis, Inria, France

Prof. Dirk Slock, EURECOM, Sophia Antipolis, France – *Jury President*

Thesis Supervisors:

Prof. Derya Malak, EURECOM, Sophia Antipolis, France

Prof. David Gesbert, EURECOM, Sophia Antipolis, France

THÈSE DE DOCTORAT
SORBONNE UNIVERSITÉ

Spécialité:

Sciences et technologies de l'information et de la communication

École doctorale :

Informatique, télécommunications et électronique de Paris

Présentée par

DEYLAM SALEHI Mohammad Reza

Pour obtenir le grade de

Doctorat

Sujet de la thèse :

Limites fondamentales du calcul distribué de fonctions non-linéaires dans certains réseaux multi-utilisateurs

Soutenue le 05/12/2025

Rapporteurs :

Prof. LE TREUST Maël, CNRS, Inria, Rennes, France

Prof. WIGGER Michéle, Télécom Paris, Paris, France

Examinateurs:

Prof. STAVROU Photios A., EURECOM, Sophia Antipolis, France

Prof. NEGLIA Giovanni, Sophia Antipolis, Inria, France

Prof. SLOCK Dirk, EURECOM, Sophia Antipolis, France – *Président du jury*

Directeur de thèse:

Prof. MALAK Derya, EURECOM, Sophia Antipolis, France – *Co-encadrant*

Prof. GESBERT David, EURECOM, Sophia Antipolis, France – *Directeur*

To my family,
and my country, Iran.

بنی آدم اعضایِ یکدیگرند
که در آفرینش زیک گوهرند

چو عضوی به درد آورد روزگار
دگر عضوها را نماند قرار

تو کز محنتِ دیگران بی غمی
نشاید که نامت نهند آدمی

Human beings are members of a whole, in the creation of one essence and soul.
If one member is afflicted with pain, other members uneasy will remain.
If you have no sympathy for human pain, the name of the human being you cannot retain.

–Saadi Shirazi (12th century)

Abstract

In the era of massive data generation and complex distributed systems, handling the computation costs over networks has become a central challenge. As modern applications in cloud computing, machine learning, and data analytics shift toward distributed architectures, the focus has moved from traditional source coding to functional compression—computing functions of data rather than recovering the data itself.

This thesis develops achievable schemes for distributed function computation over communication networks. Using characteristic graphs, which capture the structural dependencies between data sources and functions, we optimize communication costs across three settings:

Graph-Theoretic Limits of Distributed Computation: We study the distributed computation of arbitrary functions over correlated sources using characteristic graphs and their n-fold OR products. Our framework derives rate bounds based on chromatic entropy for cyclic, d-regular, and general graphs. For cycle graphs, we establish exact results, while for d-regular and general graphs, we develop new bounds on their graph entropies by leveraging spectral properties, including regions that approximate the eigenvalues of the characteristic graph adjacency matrix. These results reveal how the graph structure and spectrum relate to compressibility and communication efficiency.

Multi-Server Multi-Function Distributed Computation: Extending the characteristic graph framework to multi-server, multi-function settings, we analyze scenarios where a user requests complex functions from distributed servers. Our achievability bounds capture the interplay between function structure, source correlations, and data placement. For linearly and non-linearly separable functions, under cyclic placement, our schemes significantly reduce communication costs compared to existing methods, offering new insights into distributed task allocation and network-aware computation.

Non-Linear Function Computation Broadcast: We study a noiseless broadcast model in which a central node (master) enables K users to compute distinct, potentially non-linear functions using their individual side information. Leveraging a novel graph-based coding scheme, we capture the structural dependencies among datasets, user demands, and side information. We derive tight achievability and converse bounds that generalize classical results on the K-user linear computation broadcast and the 2-user non-linear computation broadcast problems, and show strict improvements over the ex-

isting schemes, using characteristic graphs and variable length coding. For example, we demonstrate lower rates versus the K-user linear computation broadcast problem [1].

This thesis leverages graph-theoretic tools for distributed source and distributed functional compression, offering rigorous foundations and practical schemes for computation over networks. The organization of this dissertation is as follows. Chapter 1 introduces the distributed computation frameworks addressed in this work and outlines our contributions. Chapter 2 develops a graph-theoretic approach for functional compression. Chapter 3 applies the characteristic graph framework to a multi-function, multi-server setting. Chapter 4, leveraging characteristic graphs, explores the fundamental limits of broadcast function computation. Finally, Chapter 5 concludes the results presented in this dissertation and details the key future directions.

Résumé (En Français)

À l'ère de la génération massive de données et des systèmes distribués complexes, la gestion des coûts de calcul sur les réseaux est devenue un défi central. Alors que les applications modernes en informatique en nuage, en apprentissage automatique et en analyse de données s'orientent vers des architectures distribuées, l'attention s'est déplacée du codage source traditionnel vers la compression fonctionnelle — c'est-à-dire le calcul de fonctions des données plutôt que la récupération des données elles-mêmes.

Cette thèse développe des schémas atteignables pour le calcul distribué de fonctions sur des réseaux de communication. En utilisant des graphes caractéristiques, qui capturent les dépendances structurelles entre les sources de données et les fonctions, nous optimisons les coûts de communication dans trois contextes:

Limites théoriques des graphes pour le calcul distribué: Nous étudions le calcul distribué de fonctions arbitraires sur des sources corrélées en utilisant les graphes caractéristiques et leurs produits OR n-fois. Notre cadre théorique établit des bornes de débit basées sur l'entropie chromatique pour les graphes cycliques, d-réguliers et généraux. Pour les graphes cycles, nous obtenons des résultats exacts, tandis que pour les graphes d-réguliers et généraux, nous développons de nouvelles bornes en exploitant les propriétés spectrales et des outils tels que le théorème du cercle de Gershgorin. Ces résultats révèlent comment la structure et le spectre du graphe influencent la compressibilité et l'efficacité de la communication.

Calcul Distribué Multi-Serveurs et Multi-Fonctions: En étendant le cadre des graphes caractéristiques aux contextes multi-serveurs et multi-fonctions, nous analysons des scénarios où un utilisateur demande des fonctions complexes à des serveurs distribués. Nos bornes d'atteignabilité capturent l'interaction entre la structure des fonctions, les corrélations entre sources et le placement des données. Pour les fonctions linéairement et multi-linéairement séparables avec un placement cyclique, nos schémas réduisent considérablement les coûts de communication par rapport aux méthodes existantes, apportant de nouvelles perspectives sur l'allocation distribuée des tâches et le calcul tenant compte du réseau.

Diffusion pour le Calcul de Fonctions Non Linéaires: Nous étudions un modèle de diffusion (broadcast) sans interférences dans lequel un nœud central (maître) permet à K utilisateurs de calculer des fonctions distinctes, potentiellement non linéaires, en

utilisant leurs informations secondaires individuelles. En exploitant un nouveau schéma de codage basé sur les graphes, nous capturons les dépendances structurelles entre les ensembles de données, les demandes des utilisateurs et les informations secondaires. Nous établissons des bornes serrées pour l'atteignabilité et la converse, généralisant ainsi les résultats classiques sur les problèmes de diffusion de calcul linéaire à K utilisateurs et de calcul non linéaire à 2 utilisateurs. Nous montrons des améliorations strictes par rapport aux schémas existants, en utilisant les graphes caractéristiques et le codage à longueur variable. Par exemple, nous démontrons des taux plus faibles que ceux du problème de diffusion de calcul linéaire à K utilisateurs [1].

Cette thèse exploite des outils graphes-théoriques pour la compression distribuée de sources et la compression fonctionnelle distribuée, offrant des fondements rigoureux ainsi que des schémas pratiques pour le calcul sur réseaux. L'organisation de cette dissertation est la suivante. Le chapitre 1 présente les cadres de calcul distribué étudiés dans ce travail et expose nos contributions. Le chapitre 2 développe une approche graphes-théorique de la compression fonctionnelle. Le chapitre 3 applique le cadre des graphes caractéristiques à un contexte multi-fonction et multi-serveur. Le chapitre 4, en s'appuyant sur les graphes caractéristiques, explore les limites fondamentales du calcul de fonctions en diffusion. Enfin, le chapitre 5 conclut sur les résultats présentés dans cette dissertation et détaille les principales orientations futures.

Acknowledgements

First and foremost, I would like to thank my advisor, Prof. Malak, for guidance and support throughout this research. I am also grateful to my friends for their help along my Ph.D. journey. Many thanks to my master's supervisor, Dr. Tavakoli, for guiding me on the information theory track. I would also like to thank my office mates from one of the most friendly and supportive offices I have worked in so far.

Above all, I thank my parents and family for their unwavering support, my nephew, Karan, whom I have missed seeing grow up, and in loving memory of my grandfather, who passed away while I was busy writing this thesis.

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

ADC	Analog to Digital Converter
AI	Artificial Intelligence
Bern	Bernoulli
CDC	Coded Distributed Computing
CSI	Channel State Information
DFT	Discrete Fourier Transform
GCT	Gershgorin Circle Theorem
Genie	Genie Lower Bound
i.i.d.	Independent and Identically Distributed
IS	Independent Set
IoT	Internet-of-Things
LCC	Lagrange Coded Computing
LHS	Left-Hand-Side
MAC	Multiple Access Channel
MAN	Maddah-Ali Niesen Placement
MIS	Maximal Independent Set
PMF	Probability Mass Function
RHS	Right-Hand-Side
RIS	Reconfigurable Intelligent Surfaces
SW	Slepian-Wolf Encoding
VoD	Video on Demand

Notations

$\mathbb{Z}^+$	Positive integers excluding 0
$\mathbb{N}$	Natural numbers excluding 0
$\mathbb{C}$	The complex field
$\mathbb{R}$	The real field
$\mathbb{F}_q$	The finite field with q elements
$\otimes$	Kronecker product
$\det(\mathbf{A})$	The determinant of matrix $\mathbf{A}$
$\lambda_k(\mathbf{A})$	The k-th eigenvalue of matrix $\mathbf{A}$
$\sigma(\mathbf{A})$	The set of all eigenvalues of matrix $\mathbf{A}$
L_p-norm	$\|x\|_p = \left(\|x_1\|^p + \|x_2\|^p + \cdots + \|x_n\|^p\right)^{1/p}$ where $x \in \mathbb{R}^n$ and $p \geq 1$
L_∞-norm	$\|x\|_\infty = \max\{\|x_1\|, \|x_2\|, \ldots, \|x_n\|\}$ where $x \in \mathbb{R}^n$
$\wedge$	The binary AND operation
$\vee$	The binary OR operation
$\mathbf{I_V}$	The identity matrix with size $V \times V$
$\mathbf{J}_V$	The all-ones matrix with size $V \times V$
$\mathbf{Z}_V$	The all-zeros matrix with size $V \times V$
$[n]$	$\{1, 2, \ldots, n\}$, where $n \in \mathbb{Z}^+$
$[a : b]$	$\{a, a+1, \ldots, b\}$ for $a, b \in \mathbb{Z}^+$, and $a < b$
$\log_q$	Logarithm in base q, log for $q = 2$
X	A discrete random variable
$\mathcal{X}$	A finite alphabet
$P_X(x)$	PMF of random variable X
$\mathbb{P}(\cdot)$	Probability of a realization or an event
$1_{x \in A}$	The indicator function, which takes the value 1 if $x \in A$ and 0 otherwise
$X \sim \text{Bern}(p)$	A binary random variable with a probability p of one of two values
$X \sim B(n, \epsilon)$	A binomial random variable with n trials and success probability ϵ
$X \sim \text{Unif}(\mathbb{F}_q)$	A random variable uniformly distributed over $\mathbb{F}_q$
X_i^n	A sequence of n i.i.d. random variables over $\mathbb{F}_q$
$H_q(X)$	Entropy of X over $\mathbb{F}_q$

$H(\cdot)$	Entropy of X over $\mathbb{F}_2$
$h(p)$	Binary entropy function, i.e., the entropy of $X \sim \text{Bern}(p)$
$H_q(X, Y)$	The joint entropy of random variables X and Y
$H_q(X \mid Y)$	The conditional entropy of random variable X given Y
$H_B(B(n, \epsilon))$	Entropy of a binomial random variable for $n \in \mathbb{N}$ and $\epsilon \in [0, 1]$
$A_\epsilon^{(n)}$	Typical set of n i.i.d. observations of a random variable with an upper bound on empirical deviation $\epsilon > 0$
$\mathscr{C}$	A random codebook
G_{X_1}	A characteristic (or confusion) graph built by a source X_1
$\mathcal{C}_{G_{X_i}}$	A valid coloring of G_{X_i}
$H_{G_{X_i}}(X_i)$	Characteristic graph entropy of X_i
$\mathbf{A}_f$	The adjacency matrix of a characteristic graph G
$G_{d,V}$	A d-regular graph with V vertices
$Dia(G)$	The diameter of G
$\Gamma(G)$	The set of all maximal independent sets (MISs) of graph G
U	An MIS in $\Gamma(G)$
K_c	Number of demanded functions by the user (Chapter 3)
N_r	Recovery threshold (Chapter 3)
M	Worker's cache size (Chapter 3)
$X_{\mathcal{S}}$	A subset of variables with indices $i \in \mathcal{S}$, i.e. $X_{\mathcal{S}} = \{X_i : \ i \in \mathcal{S}\}$
$G_{\cup_{i=1}^K f_i}$	The union graph or broadcast graph, given user demands f_i for $i \in [K]$
$\mathcal{S}_i$	Side information at user i for $i \in [K]$
G_{f_i}	The characteristic graph that the master node builds for computing f_i given $\mathcal{S}_i$ (Chapter 4)
B	Broadcast message in the computation broadcast model

Chapter 1

Introduction

Distributed computation is a core paradigm in modern data processing, where multiple nodes collaborate to compute functions over data dispersed across a network (see Figure 1.1). This paradigm appears in diverse settings, including distributed learning [2–4], over-the-air computation (AirComp) in wireless systems [5, 6], coded computing for large-scale data processing [7, 8], broadcast services over distributed databases [9], and edge computing in Internet-of-Things (IoT) environments [10]. Emerging applications — such as real-time analytics over reconfigurable intelligent surfaces (RIS) [11] and federated artificial intelligence (AI) at the network edges [12] — require frequent exchange of data among distributed nodes, which further increases the need for computation models that reduce communication in the distributed processing of data.

Distributed computation systems face key challenges related to communication cost and limited computational resources. As of early 2025, global mobile data traffic has reached approximately 172 EB per month, with projections exceeding 430 EB by 2030 [13]. Nokia estimates total network traffic will range from 2400 to 3100 EB per month by 2030 [14]. This growth in data volume highlights the need to perform computation within the network to reduce transmission overhead. In large-scale systems, transmitting or reconstructing complete datasets is often impractical. For example, in Video-on-Demand (VoD) tasks, servers must support recommendation, transcoding (i.e., converting a media file from one format, bitrate, or resolution to another), and caching under storage and processing constraints [15–17]. Communication cost is a critical bottleneck in various wireless network scenarios [18, 19], multi-agent learning setups [20, 21], and it is essential to maintain the coordination and processing of large-scale data with minimal exchange among network nodes. To that end, this dissertation focuses on balancing the costs of computation and communication in large-scale systems.

Several frameworks have been designed to address the challenges in distributed settings. Cluster-based platforms such as MapReduce [22], Spark [23], and Hadoop [24] divide data and computation across servers, coordinating task scheduling, fault recovery, and data shuffling through a centralized or distributed control layer. Other distributed dataflow systems, such as Dryad [25] and Apache Flink [26], extend these principles

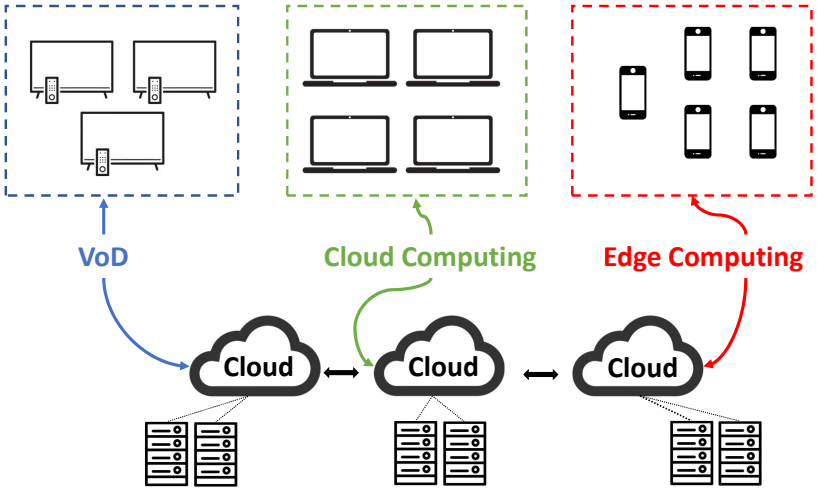

Figure 1.1: An illustration of a distributed computing system, where users have varying needs from different servers or the data stored on these servers.

to streaming and iterative computations, enabling continuous processing of high-volume data. Edge-computing platforms integrate lightweight orchestration to bring computation closer to data sources, reducing latency and bandwidth usage in IoT and AI inference workloads [27]. These frameworks focus primarily on resource management, execution pipelines, and developer interfaces, without necessarily capturing the fundamental limits of communication and computation costs.

From an information-theoretic perspective, different (source and channel) coding strategies are designed to capture function structure for computation and reduce communication costs in distributed settings. Coded Distributed Computing (CDC) [7] achieves a reduction in the number of messages exchanged by designing data placement and task assignment patterns that create coding opportunities during the shuffle phase[1]. Polynomial codes and Lagrange Coded Computing (LCC) [8, 28, 29] introduce structured redundancy in computation to reduce communication costs while tolerating stragglers and, in some cases, enhancing privacy. Index coding [30–32] and lattice coding [5, 33] exploit algebraic structure and placement of data to optimize the communication cost. Network coding [34, 35] and distributed source coding [36, 37] leverage network topology or statistical correlation for compression to exploit the fundamental limits of compressibility. Similarly,

[1]In MapReduce, computation is split into a *map* phase that produces intermediate key–value pairs, a *shuffle* phase that exchanges these pairs across the network by key, and a *reduce* phase that aggregates values with the same key.

sketching-based techniques [38, 39] and task-oriented quantization [40–42] produce compact data representations for approximate computation.

Beyond source compression, certain approaches align coding with the functional or algebraic structure of the computation task [43, 44]. An important class of functions in distributed computation is captured by linearly separable functions. This class embeds various special scenarios, including linear functions such as convolution, discrete Fourier transform, and gradient computation, and inner-product-based representations of matrix multiplication, allowing distributed processing of large-scale data. For distributed linearly separable function computation, recent works have focused on multi-function coding schemes [45, 46] and have explored the fundamental limits of coded computation with side information [47], while for distributed matrix multiplication, studies have addressed various aspects in [48–58]. A broader class of functions, known as nomographic functions [59, 60], can capture non-linear functions of linearly separable transformations. These functions have been used in the context of over-the-air computing for communication-efficient aggregation (where the inherent nature of the additive multiple access channel (MAC) is exploited for aggregation), see e.g., [59–62]. From an algebraic perspective, Han and Kobayashi [63] have shown for a broad class of functions (e.g., modulo-sum, parity, and indicator functions) that exploiting the structure of the function for source compression can achieve lower communication costs versus distributed source compression. A structured linear encoding scheme was devised by Körner and Marton in [64] for computing the modulo 2 sum of two distributed sources, which was proven to achieve an optimal communication cost. The ideas in [64] have been extended to broader function classes, see e.g., [65] for general binary sources, [66] for optimal weighted-sum rates of the modulo-sum, and [67, 68] for distributed computation of general nonlinear functions over finite fields. Körner and Marton's linear codes have been extended by Lalitha *et al.* for computing linear subspaces of distributed sources [69], and by Malak for distributed matrix multiplication [70, 71], while achieving communication savings over standard compression schemes.

Understanding the limits of distributed coding and compression methods requires theoretical models that capture the structure of functions, source correlations, and network topologies. Understanding the limits of distributed coding and compression methods requires theoretical models that capture the structure of functions, source correlations, and network topologies. Graph-theoretic models characterize communication requirements for computing functions of correlated sources. The notion of *characteristic graphs* and *graph entropy*, introduced in [72, 73], has been used in [74, 75] to obtain rate lower bounds for both lossless and lossy computation, with extensions to zero-error computation with side information [73, 76] and multi-terminal settings [77, 78]. Communication complexity theory [79–81] complements this direction by quantifying the minimum interaction needed for function computation. These frameworks connect fundamental limits to practical coding strategies, guiding the design of schemes that exploit structure in functions, source distributions, and network topologies. In particular, Alon and Orlitsky [75] established the

relation between characteristic graphs, chromatic entropy, and graph entropy in a single-source setting, while Orlitsky and Roche [74] extended this framework to distributed coding for computing a deterministic function $f(X_1, X_2)$ of two correlated sources, thereby providing the direct graph-theoretic foundation for the coding-for-computing problem studied in this thesis.

Compression strategies in distributed computation have generally been categorized into zero-error, lossless, and lossy paradigms. Zero-error coding has been largely studied under strict reconstruction requirements [73], while lossless compression has been characterized through classical results such as Slepian–Wolf coding [82] and its graph-based extensions [74, 77, 83]. In parallel, lossy compression has been extensively investigated in both finite-state and rate–distortion formulations [84–92], establishing performance limits under distortion constraints. These works have highlighted the fundamental trade-offs between fidelity, rate, and complexity. In contrast, this thesis has focused on the asymptotically lossless regime, where vanishing error probabilities allow the use of graph-theoretic tools to capture computation–communication trade-offs.

Motivation: A core challenge in distributed function computation is to characterize the structural properties of distributed functions in a way that enables achievable encoding and communication across nodes. Classical source coding, pioneered by Shannon [93] and extended to distributed systems by Slepian and Wolf [82], aims to eliminate redundancy while ensuring accurate data reconstruction. Over time, distinct paradigms have emerged — lossless, lossy [84, 85], and zero-error [73] compression — each offering trade-offs between accuracy, complexity, and communication cost. In many distributed settings, however, the receiver is interested in computing a function of the data rather than reconstructing it in full. This has led to the development of functional compression, where *characteristic graphs* [74, 75] model the distinguishability induced by the function, and graph-theoretic tools such as graph entropy and chromatic number characterize the corresponding compression limits [77, 78, 83]. Characteristic graphs are powerful structures that jointly capture both the structural properties of the function and the statistical correlations in the source data. However, much of the existing literature restricts attention to simple graph families or linear functions.

This thesis broadens the scope by tackling arbitrary, non-separable, and more generally non-linear functions, employing the framework of characteristic graphs. These graphs allow for asymptotically lossless compression of sources for distributed computing of functions. To that end, this thesis develops new graph-theoretic, coloring-based, and entropy-based bounds to handle general graph structures by exploiting graph entropy, n-fold OR products of graphs, and chromatic numbers. Our approach characterizes the fundamental limits of communication for distributed function computation for a broad class of functions and data statistics.

This thesis then extends this framework to the *multi-server multi-function distributed computation* setting, where a user aims to compute several (potentially non-linear) functions from data distributed across multiple servers. Previous work has largely focused on

linear functions or limited placement models. In contrast, in this thesis, we derive upper bounds on communication cost for general functions and data statistics using characteristic graphs. Our analysis reveals structural advantages in special cases, such as cyclic data placements and multi-linear functions, demonstrating the performance of graph-based methods for multi-function computation.

Finally, this thesis addresses the *function computation broadcast* problem, in which a central node (or a master) broadcasts to multiple users — each with side information — to enable computation of different functions. Drawing on recent results in broadcast computation and the capacity-achieving bounds in [1, 94] for linear function demands and in [95] for complementary demands, we introduce a novel graph-theoretic master-users model that accommodates arbitrary demand structures and establishes, to the best of our knowledge, the first general bounds on broadcast rates for non-linear function computation.

The contributions of this thesis are threefold, which we describe next.

1.1 Graph-Theoretic Limits of Distributed Computation

In Chapter 2, we devise achievable schemes for distributed functional compression. To that end, we consider two finite alphabet correlated sources, X_1 and X_2, observed at separate encoders. Given i.i.d. and memoryless source realizations $\mathbf{X}_1^n = X_{11}, X_{12}, \ldots, X_{1n}$ and $\mathbf{X}_2^n = X_{21}, X_{22}, \ldots, X_{2n}$ according to the distributions P_{X_1} and P_{X_2}, respectively, where the joint distribution satisfies P_{X_1,X_2}, a receiver wishes to compute an arbitrary (possibly non-linear) component-wise function $f(\mathbf{X}_1^n, \mathbf{X}_2^n) = (f(X_{11}, X_{21}), f(X_{12}, X_{22}) \ldots, f(X_{1n}, X_{2n}))$ with vanishing error probability as $n \to \infty$, without the joint reconstruction of $\mathbf{X}_1^n$ and $\mathbf{X}_2^n$. In this model, which is summarized in Figure 1.2, each encoder maps its length-n source sequence to a codeword (represented by a color) and transmits it to the common receiver. The design objective is to minimize the communication cost, i.e., the total number of bits sent by both encoders (normalized per symbol), while ensuring that the pair of indices is sufficient to determine $f(X_1, X_2)$.

In this setting, to approximate fundamental limits of distributed functional compression, each encoder builds a *characteristic graph* and applies a valid coloring[2]. The characteristic graph built by encoder 1, denoted G_{X_1}, is defined on the alphabet $\mathcal{X}_1$ representing the possible source realizations. In G_{X_1}, two nodes are adjacent if there exists a realization $x_2 \in \mathcal{X}_2$ such that $f(X_1, x_2)$ may differ across a subset of X_1, and such instances must be distinguished. A similar graph G_{X_2} is constructed by encoder 2. For length n source sequences, the corresponding objects are the *n-fold OR product* graphs, denoted $G_{X_1}^n$ and $G_{X_2}^n$, whose vertex sets are $\mathcal{X}_1^n$ and $\mathcal{X}_2^n$, respectively. In this problem, the achievable rate

[2] A (vertex) coloring of a graph is called valid if any two adjacent vertices are assigned distinct colors.

of each encoder is characterized by the *graph entropy*, which is given as [75, Theorem 5]:

$$H_{G_{X_i}}(X_i) = \lim_{n\to\infty} \frac{1}{n} H^{\chi}_{G^n_{X_i}}(\mathbf{X}_i) \ , \quad i \in \{1,2\} \ , \tag{1.1}$$

where $H^{\chi}_{G^n_{X_i}}$ denotes the *chromatic entropy*[3] of $G^n_{X_i}$. In general, computing or tightly bounding (1.1) is generally intractable, especially for irregular or non-symmetric graph structures, because it requires evaluating chromatic numbers of OR product graphs, which is at best an NP-complete problem [97].

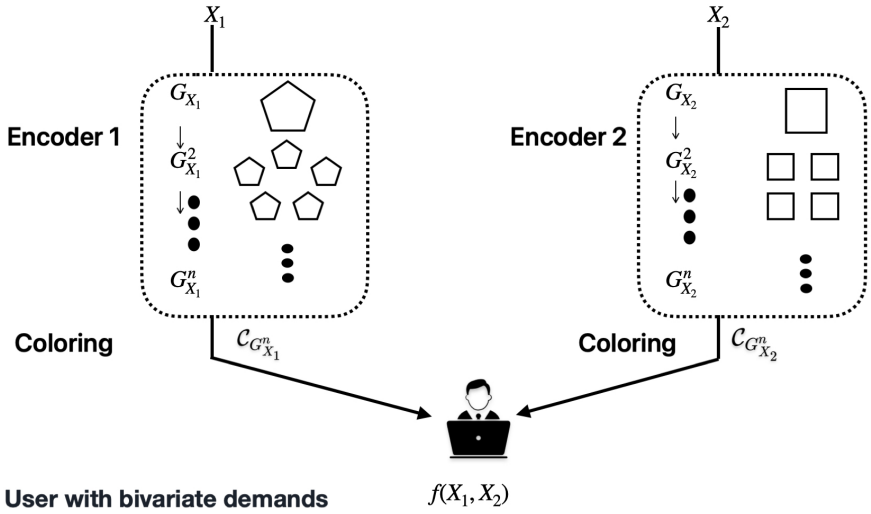

Figure 1.2: An example functional compression scheme with two encoders where each builds a characteristic graph G_{X_i} and its n-fold product $G^n_{X_i}$, and applies color encoding on $G^n_{X_i}$. The user reconstructs the demand using a look-up table.

1.1.1 Main Contributions

We design distributed compression schemes using characteristic graphs for three different graph topologies: cycle graphs, d-regular graphs, and general graphs. Across these classes, our analysis combines explicit vertex-degree calculations in G^n, constructive polynomial-time coloring schemes, spectral analysis of adjacency matrices of G^n ($\mathbf{A}^n_f$), and eigenvalue[4] bounds.

For **cycle graphs**, denoted by C_i, which are 2-regular connected graphs where each vertex is adjacent to two neighbor vertices in a closed loop, we obtain exact degree expressions in C^n_i and characterize the chromatic numbers for even cycles (cf. Proposition 2.2). For odd cycles C_{2k+1}, we develop a recursive coloring scheme that achieves a provable

[3]For a graph $G(\mathcal{V}, \mathcal{E})$, the chromatic entropy is defined as the minimum entropy over all valid coloring distributions of G, where a valid coloring is denoted as $\mathcal{C}_G$ [96].

[4]The k-th eigenvalue of a $V \times V$ matrix $\mathbf{A}$ is denoted by $\lambda_k(\mathbf{A})$, where $k \in [V]$.

gain over greedy coloring methods. (cf. Proposition 2.3), which is quantified as

$$\eta_n = \frac{\big(\chi(C_{2k+1})\big)^n}{\chi(C^n_{2k+1})} \geq 1.2^n \ , \tag{1.2}$$

where $\eta = \lim_{n\to\infty} \eta_n = \infty$. Hence, this recursive scheme yields unbounded coloring gains as blocklength grows, providing a constructive method for significantly reducing achievable rates compared to naive colorings. Cycle graphs can capture demands arising from modular and parity-type function computations.

For **d-regular graphs**, denoted by $G_{d,V}$, which are graphs of V vertices where each vertex has degree d, we characterize both the degrees and chromatic numbers of $G^n_{d,V}$. We also derive explicit lower bounds on their expansion rate (cf. Proposition 2.10) as

$$E_\theta(G^n_{d,V}) \geq \frac{(d \cdot \frac{V^n-1}{V-1})^2}{\Lambda^2(G^n_{d,V}) + \Big((d \cdot \frac{V^n-1}{V-1})^2 - \Lambda^2(G^n_{d,V})\Big) \cdot \frac{|Y|}{V^n}} \ , \tag{1.3}$$

where $\Lambda(G^n_{d,V}) = \max\big(\lambda_2(\mathbf{A}^n_f), |\lambda_{V^n}(\mathbf{A}^n_f)|\big)$, where $\mathbf{A}^n_f$ is the adjacency matrix, $\lambda_2(\cdot)$ is the second-largest eigenvalue, and $\lambda_{V^n}(\cdot)$ is the smallest eigenvalue of $G^n_{d,V}$. The expansion bound in (1.3) reflects the graph's connectivity. A larger expansion rate corresponds to a larger chromatic number, a higher graph entropy, and thus a higher functional compression rate (see (1.1)). Thus, the regularity and spectral properties of d-regular graphs directly govern the communication cost in distributed function computation.

For **general graphs**, denoted by $G = G(\mathcal{V}, \mathcal{E})$ with V vertices and E edges, we derive entropy bounds and spectral bounds that do not rely on structural assumptions. First, for odd-valued V, we establish an achievable upper bound on the graph entropy of X_1 (cf. Proposition 2.12):

$$H_G(X_1) \leq \frac{1}{n} H\Big(\zeta_n \cdot \tfrac{\alpha(G^n)}{V^n}, \zeta_{n-1} \cdot \tfrac{\alpha(G^{n-1})}{V^n}, \ldots, \zeta_0 \cdot \tfrac{1}{V^n}\Big) \ , \tag{1.4}$$

where ζ_t, $t \in [n] \cup \{0\}$, is the count of maximum independent sets of G^t of size[5] $\alpha(G^t)$. Subsequently, we derive a lower bound on graph entropy (cf. Corollary 2.9) as follows:

$$\log_2\left(\frac{2V}{V-1}\right) \leq H_G(X_1) \ .$$

In addition, we express the eigenvalues of the adjacency matrix of G^n using the Gershgorin circle theorem (GCT)[6] (cf. Theorem 2.2):

$$\lambda_k(\mathbf{A}^n_f) = \lambda_k(\mathbf{A}^n_{Gr}) + \lambda_k(\mathbf{A}^n_{f^c}), \quad k \in [V^n] \ , \tag{1.5}$$

[5]The independence number of G, denoted by $\alpha(G)$, represents the cardinality of the maximum independent set, i.e., the size of the largest independent set, of G.

[6]The Gershgorin circle theorem states that the eigenvalues of a square matrix with entries in the complex field lie within specific regions (discs) [98].

where in (1.5) $\mathbf{A}_{Gr}^n$ is a block diagonal matrix with diagonal blocks formed from $\mathbf{A}_f^{n-1}$, and $\mathbf{A}_{f^c}^n = \mathbf{A}_f^n - \mathbf{A}_{Gr}^n$ captures the off-diagonal elements of $\mathbf{A}_{Gr}^n$. This spectral method applies to diverse graph classes, such as sparse, irregular, or dense graphs, and provides computable bounds when direct coloring is intractable. Using (1.5), we derive bounds on chromatic numbers and hence graph entropy.

Overall, these contributions, which we will detail in Chapter 2, highlight the role of a graph's structure — cyclic symmetry, regular expansion, or general spectral properties — in determining the achievable rates in distributed functional compression.

To illustrate these results in practice, consider three representative function classes:

- *Modular and parity-type functions:* For such function classes, the characteristic graphs are cyclic, and since only function equivalence classes need to be determined rather than all source symbols, the minimum communication rate is strictly less than the rate required for joint reconstruction of distributed sources (see [82]).
- *Linear transforms with uniform dependencies:* Each output depends uniformly on many or all inputs (e.g., DFT, dense matrix–vector multiplication). These induce near-regular graph structures, and the expansion bounds show how mixing across coordinates enables color reuse over blocks, reducing the sum rate (see Chapter 2.3.6).
- *Non-linear threshold or indicator functions:* These classes of functions often yield sparse or irregular graphs. The spectral framework, detailed at length in Chapter 2.4.4, relates the largest eigenvalue of A_f^n to $\chi(G^n)$, and $H_G(X_1)$, to approximate the fundamental limits of distributed functional compression.

1.1.2 Summary

By focusing on cycles, d-regular graphs, and general graphs under spectral constraints, this dissertation provides connections between the *graph structure* (degree, expansion, eigenvalues) to *rate*. In particular, exact or near-exact descriptions for cycles, explicit expansion bounds for regular graphs, and spectral GCT-based bounds for general graphs are provided, while existing analyses have typically treated restricted graph families [74, 75], or have produced inequalities using n-fold OR products that are not tight for general graphs [77, 83, 99], and only coarse estimates for such quantities were known in the literature. This clarifies the role of exploiting graph regularity, symmetry, or sparsity in achieving significant rate savings for distributed function computation via distributed source compression.

In Chapter 2, we give an exact characterization for even cycles, tight chromatic number bounds for cycles, expansion-based bounds for d-regular graphs, and spectral bounds applicable to general graphs. For specific characteristic graph families (e.g., even and odd cycles), the approach is constructive; for instance, the converse bound is given by the capacity of C_5 [100]. However, an exact characterization of chromatic numbers and graph entropies is generally NP-hard; using our eigenvalue-based bounds, we present achievable approaches for handling irregular characteristic graphs.

The results presented in Chapter 2 provide the theoretical basis for functional compression with two sources. Building on this foundation, Chapter 3 generalizes the distributed functional compression setting to multiple servers and multiple function demands, where placement strategies and function structure introduce additional layers of complexity.

1.2 Multi-Server Multi-Function Distributed Computation

In Chapter 3, we study a distributed system where data is distributed across multiple servers, each with limited storage capacity (i.e., cache size M). A master node, which has full access to all datasets $(X_1, X_2, \ldots, X_N)$, determines the dataset assignments (disjoint or cyclic) across servers. A single user requests the computation of several (namely K_c) general, possibly non-linear, functions $\{F_j(X_1, X_2, \ldots, X_N)\}_{j\in[K_c]}$ of data stored across all servers (see Figure 1.3). The user should be able to compute the requested functions even if only a subset of servers (N_r out of N) transmits successfully. This setting captures real-world environments such as cloud or edge networks, where datasets are split across servers and functions must be served reliably under storage and communication constraints. Our objective here is to minimize the total communication cost while ensuring the delivery of functions.

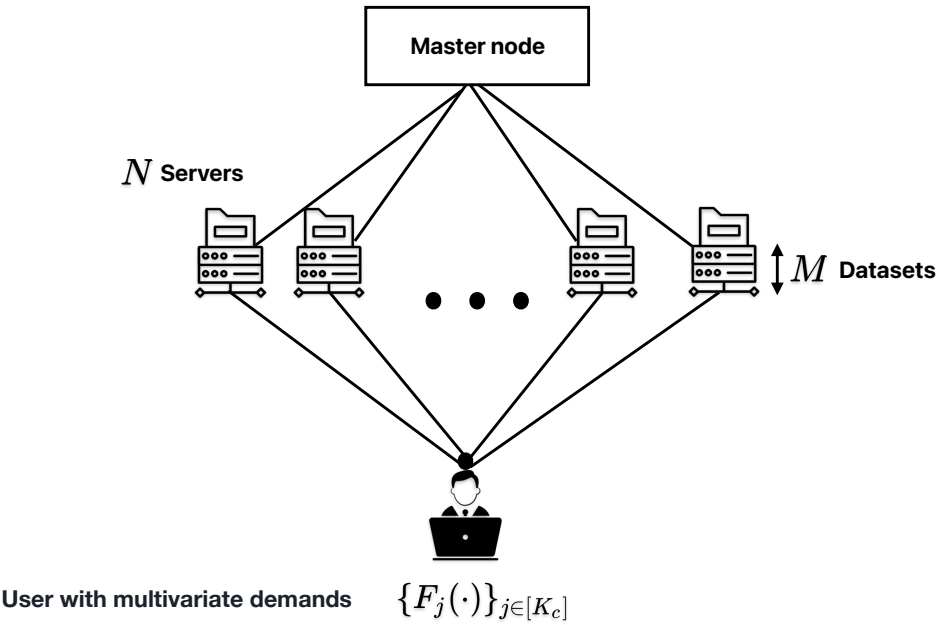

Figure 1.3: A distributed functional compression model, involving a master node that assigns datasets to distributed servers, which compute users' demands, and a user with multivariate functions of distributed datasets

.

The main challenges in this setting are summarized as follows.

- **Data placement:** The dataset assignment policy of the master node to servers

determines how much information is duplicated across servers and, consequently, the minimum number of bits that must be communicated. Placement serves as a fundamental design parameter that shapes characteristic graphs and achievable rates. The use of union graphs ensures that multiple functions can be served simultaneously without duplication, while statistical skew and correlation in the data provide further opportunities for reducing communication costs.

- **Statistical properties:** Leveraging intra-server and inter-server dataset correlations reduces the communication cost in distributed multi-server computation.
- **Multiple functions:** Computing several functions simultaneously is not equivalent to computing them individually; naive approaches that do not take into account the structure of demanded functions can yield redundant transmissions.
- **Straggler mitigation:** The system must allow recovery from any subset of servers (N_r), which imposes combinatorial constraints on data distribution and encoding.

1.2.1 Main Contributions

We employ *characteristic graphs* to capture the user demands and each server's stored data. This approach enables each server to identify data values that must be distinguished in order to ensure correct computation. When multiple functions are requested, these demands can be combined into a *union graph* to ensure that a single encoding function can capture all demands simultaneously. Utilizing such graphs, we derive bounds on the communication cost that reflect the effects of data placement, data correlations, and function structures.

In the case of **cyclic placement**, where datasets are cyclically assigned to servers and subsets of servers have dataset overlaps, we generalize the distributed gradient coding approach [101], which is devised for linearly separable computations, to non-linear function demands. Specifically, we devise novel achievability results for non-linearly separable demands by exploiting the dataset redundancy and the function structures. Specifically, with our approach, the achievable sum rate for the union of function demands satisfies

$$R_{ach} \leq \sum_{i=1}^{N_r} \min_{Z_i = g_i(X_i) \,:\, g_i \in \mathscr{C}_i} H_{G^{\cup}_{X_i}}(X_i) \ , \tag{1.6}$$

where $G^{\cup}_{X_i}$ represents the union graph built by server i to capture all requested functions. Each server devises a coloring codebook $\mathscr{C}_i$ to map data X_i into a compressed symbol $Z_i = g_i(X_i)$, ensuring that the receiver can reconstruct all demanded functions using the transmissions from all servers.

For **linearly separable functions**, such as sums, inner products, or matrix–vector multiplications, algebraic dependencies provide further opportunities for compression. In

this case, the achievable sum rate bound takes the form

$$R_{ach} \leq \begin{cases} \min\{K_c, \Xi\} N_r, & 1 \leq K_c \leq \Xi \cdot N_r, \\ \min\{K_c, K\}, & \Xi \cdot N_r < K_c, \end{cases}$$

where $\Xi = \frac{K}{N} \in \mathbb{Z}^+$ and K_c is the number of linearly separable functions. This bound demonstrates how algebraic structure reduces the communication load by enabling servers to exploit linear dependencies across sub-functions of distributed datasets.

For the more general class of **multi-linear functions**, such as products, we show that correlations or skewed input distributions can further reduce communication compared to baseline strategies. In particular, the achievable sum rate is bounded as

$$R_{ach} \leq \frac{1 - (\epsilon_M)^{N^*}}{1 - \epsilon_M} \cdot h(\epsilon_M) + (\epsilon_M)^{N^*} \cdot 1_{\Xi_N > 0} \cdot h(\epsilon_{\xi_N}) \ , \tag{1.7}$$

where ϵ_M denotes the probability that the product of M sub-functions equals one, N^* is the minimum number of servers required for recovery, and Ξ_N determines whether an additional server must contribute. If such a server is required, it computes a smaller product of size ξ_N, with corresponding probability ϵ_{ξ_N}. This formulation shows explicitly how the rate depends on input statistics and recovery thresholds, clarifying when non-uniformity in the data distribution can be exploited for gains.

To illustrate these ideas, representative examples for distributed computation of linear and bilinear demands for $K_c \geq 1$ are presented in Chapter 3 under the cyclic placement assumption. For a class of computation scenarios, we demonstrate rate reductions by taking into account the dataset statistics and function structures, using the union graph technique. Hence, our approach can eliminate duplicated transmissions that would arise from strategies that are not sensitive to dataset correlations or the dependencies across demanded functions.

1.2.2 Summary

Coded distributed computing and coded caching have shown that intelligent data placement and coded shuffling can reduce communication for linear aggregation tasks. However, most existing designs have been limited to (i) reconstruction or purely linear objectives, (ii) single-function service, or (iii) statistics-agnostic sources. By contrast, the present work takes a different route: it employs characteristic graphs to encode *function-specific* distinguishability at each server, serves *multiple functions jointly* through union graphs, and explicitly incorporates *data statistics and correlations*. This leads to sum-rate bounds that not only specialize to known linear scenarios but also extend seamlessly to non-linear and skewed settings, while revealing the role of placement (e.g., cyclic placement) and redundancy-based schemes to mitigate stragglers.

The general achievability bound presented in the main theorem of Chapter 3 has

been held for arbitrary (possibly skewed) data statistics and general function classes. For cyclic placement and linearly separable functions, the bounds have simplified and matched optimal schemes. For multi-linear (or bilinear) functions, the advantages have become pronounced in the presence of statistical skew. The examples in Chapter 3 demonstrate gains over statistics-agnostic baselines that do not consider dataset correlations and that serve functions independently.

While the multi-server distributed model highlights the impact of placement, data correlations and multiple functions for a single user, distributed systems often serve many users simultaneously. This motivates the broadcast model studied in the next chapter, where a master must satisfy diverse function demands across multiple users with side information.

1.3 Non-Linear Function Computation Broadcast

In Chapter 4 of this thesis, we study a broadcast setting in which a master holds data and must send a common message (or set of messages, denoted by B) to multiple users to satisfy their computational demands (see Figure 1.4). Each user $i \in [K]$ has side information $\mathcal{S}_i$ — possibly subsets of the data or even functions of it — and demands function f_i. The demanded functions may be linear or non-linear, and the side information can vary widely across users. The challenge is that a single broadcast message B must simultaneously enable all users to compute their functions correctly. Unlike point-to-point settings, where communication can be tailored to each receiver, in the computation broadcast setup, the master must design a message that jointly meets all users' distinguishability requirements.

1.3.1 Main Contributions

We start by constructing a characteristic graph for each user, capturing which data values must be distinguished given its side information. We then form a *union characteristic graph*, which captures and aggregates the distinguishability requirements of all users. This construction ensures that a single broadcast message suffices for all users, avoiding duplicated encodings tailored to individual demands. The communication problem thus reduces to finding an encoding of this union graph, where the achievable communication rate closely approximates the optimal rate.

To establish the fundamental limits, we derive both achievability and converse bounds. On the achievability side, we design coding schemes by finding maximal independent set (MIS) distributions to assign codewords (colorings) of the union graph. This ensures that a single carefully constructed broadcast message B suffices for all users. In the broadcast setting with $\mathbf{X}_{[N]} = \{X_1, X_2, \ldots, X_N\}$ and K users having side information $\mathcal{S}_i$, we establish the following bound (cf. Theorem 4.1):

$$R_{ach} = \min_{\mathbf{X}_{[N]} \in U \in \Gamma(G_{\cup_{i'=1}^{K} f_{i'}})} \max_{i \in [K]} I(\mathbf{X}_{[N]}; U \mid \mathcal{S}_i) \ , \tag{1.8}$$

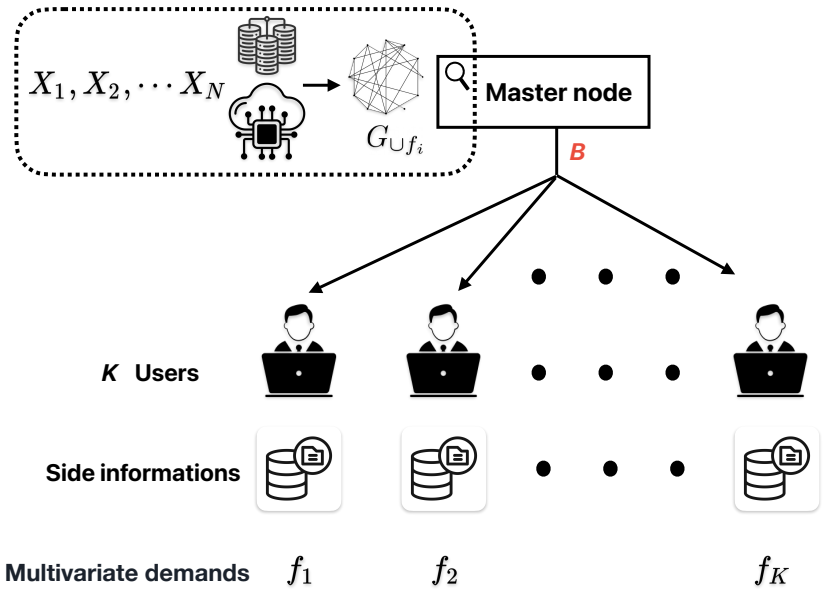

Figure 1.4: Computation broadcast model, involving a master node that has access to all datasets and a set of users, each with limited side information $\mathcal{S}_i$ and with a desired multivariate function $f_i(X_1, X_2, \ldots, X_N)$. Leveraging its computation capability, the master node constructs a characteristic graph for the union of demands of all users to devise a common broadcast message.

where $G_{\cup_{i=1}^K f_i}$ is the corresponding broadcast graph, $\Gamma(G_{\cup_{i=1}^K f_i})$ is the set of MISs of $G_{\cup_{i=1}^K f_i}$, and U denotes a chosen MIS.

On the converse side, we show that our framework generalizes the classical Wyner-Ziv model [102], thereby highlighting that the broadcast computation problem can be viewed as a function-oriented extension of multi-terminal source coding. This connection not only bridges graph-theoretic and information-theoretic perspectives but also identifies the precise dependencies that constrain the minimum broadcast rate. Specifically, in the broadcast model with K users, the following lower bound holds (cf. Theorem 4.1):

$$R_{con} = \min_{P_{\Upsilon, \mathbf{X}_{[N]}}(\upsilon|\mathbf{x}_{[N]})} \max_{i \in [K]} I(\mathbf{X}_{[N]}; \Upsilon \mid \mathcal{S}_i) \ , \tag{1.9}$$

where Υ is an auxiliary random variable representing the common broadcast message.

We further strengthen the converse by presenting extended lower bounds via a genie-aided approach (cf. Proposition 4.2):

$$R_{Genie} = \max_{\pi \in \Pi([K])} \left[H_q(f_{\pi(1)} \mid \mathcal{S}_{\pi(1)}) + \sum_{i=2}^{K} H_q(f_{\pi(i)} \mid \{f_{\pi(j)}, \mathcal{S}_{\pi(j)}\}_{\forall j \in \{1,2,\ldots,i-1\}}, \ \mathcal{S}_{\pi(i)}) \right], \tag{1.10}$$

where $\Pi([K])$ is the set of all permutations of the K users. Another converse bound is obtained using the mutual information between function demands and side information

(cf. Proposition 4.1):

$$H_q(f_1', f_2', \ldots, f_K') \ , \tag{1.11}$$

where $f_i' = f_i|\mathcal{S}_i$ for $i \in [K]$ is any function given the side information that satisfies the demand of user i.

These results recover classical settings — such as the two-user case with linear or complementary demands — as special instances of our framework. Our approach yields new achievable bounds that can handle both non-linear demands and vector size side information of the users. In example cases with three users, it exhibits a tight gap between the achievability result in (1.8) and the converse obtained from (1.11) (see Chapter 4.4). Overall, Chapter 4 demonstrates that broadcast computation can be fully characterized by a single union graph that encodes the needs of all users simultaneously. Moreover, allowing users to hold side information in the form of functions (rather than raw data) substantially reduces communication costs, underscoring the power of graph-theoretic approaches for the problem of computation broadcast.

The framework in Chapter 4 is further illustrated through examples. In the case of a 3-user non-linear computation broadcast problem (Example 4.1 in Chapter 4), the union characteristic graph contains all single-user characteristic graphs; coloring it once produces MISs that, together with each user's side information, enable recovery of their demanded functions. This eliminates the need for separate per-user encoding designs while still exploiting shared structure across demands. For a 3-user linear computation broadcast problem, we adopt the compatible-function approach to achieve an optimal rate comparable to [1], and further enhance performance through a vector-based method with variable-length coding (VLC), which reduces the communication cost beyond existing schemes (Example 4.2 in Chapter 4).

1.3.2 Summary

Broadcast computation intersects with index coding, computation broadcast, and multi-terminal source coding. Prior works have largely focused on restricted cases, such as linear functions or complementary demands between two users, whereas the extension to arbitrary functions and heterogeneous side information introduces significant additional complexity. This thesis generalizes to non-linear finite functions and heterogeneous side information, showing that a single *union characteristic graph* captures the simultaneous distinguishability requirements of all users. The achievability via colorings of this union graph recovers classical linear and complementary results as special cases and extends beyond them.

The union graph technique of Chapter 4 yields a general achievability approach, along with near-matching converses established in structured cases (e.g. $G_{\cup_{i=1}^{K} f_i} = G_{f_i}$ for $i \in K$, where K is the total number of users), and in other scenarios within some additive gaps. For general non-linear demands, a gap remains between inner and outer bounds.

Future work is needed to derive tighter converses.

We next close this Chapter by summarizing the main contributions of the thesis and presenting its overall organization.

1.3.3 Summary and Thesis Outline

In this thesis, we investigate the fundamental limits of distributed function computation in several multi-user network models, with a particular focus on non-linear demands. Our approach relies on characteristic graphs, which capture the structural properties of both data and functions, and allow us to exploit the fundamental limits of asymptotically lossless compression for function computation. The study covers three main settings: multi-source functional compression (Chapter 2), multi-server multi-function computation (Chapter 3), and computation broadcast (Chapter 4). For each multi-user setting of interest, we establish achievable rates and bounds for both linear and non-linear functions under various structural constraints, enabling systematic comparison of our schemes across different network topologies with existing coding strategies to identify when structural properties such as regularity, symmetry, or data correlations yield meaningful communication savings and rate reductions.

The organization of this thesis is as follows[7]. Chapter 1 presents the motivation and the existing literature for the dissertation. Chapter 2 elaborates on distributed functional compression for two-source settings with regular and general graphs. Chapter 3 extends the discussion to the multi-server multi-function computation model, developing new bounds and schemes. Chapter 4 focuses on the broadcast computation problem with multiple users and non-linear demands, for which we provide converse and achievability approaches. Together, these chapters provide a comprehensive investigation of the problem of function computation over networks, advancing both theoretical understanding and practical approaches for functional compression using characteristic graphs for distributed settings. Finally, Chapter 5 concludes by summarizing the main contributions, highlighting their applications and possible use cases, and suggesting future research directions.

[7]The general notation used throughout the thesis is provided in the Notations section, with the inclusion of chapter-specific supplementary notation.

Chapter 2

Graph-Theoretic Limits of Distributed Computation

This chapter[1] investigates the distributed computation of arbitrary functions from two correlated sources located at separate nodes. By leveraging the structural properties of functions through characteristic graphs and their OR products, we establish bounds on the optimal communication rate required for asymptotically lossless function computation. The analysis draws on connections between graph regularity, chromatic entropy, and spectral properties, offering new insights into how graph expansion and spectra serve to characterize computation complexities in distributed settings.

2.1 Introduction

Data compression is the process of using fewer bits than the original size of the source, which is given by Shannon's entropy of a source [93] in the case of a point-to-point communication model. In the setting of distributed sources, where the goal is to recover them jointly at a receiver, the Slepian-Wolf theorem gives the fundamental limits of rate for compression [82]. In the case when the receiver wants to compute a deterministic function of distributed sources, a further reduction in compression is possible via accounting for the structure of the function as well as the structure of the joint source distribution [75]. This approach is known as distributed functional compression, in which a function represents an abstraction of a particular task, the sources separately compress their data and send *color encodings* of their data to a common receiver, and the receiver, from the obtained colors, recovers the desired function of the sources. This approach differs from the conventional approach [82], where the receiver jointly decodes source sequences.

[1] The results in this chapter are presented in [103, 104].

2.1.1 Motivation and Literature Review

Let us begin with an example. Consider a college student database with information including the rental records, demographics, and health of individuals. The Ministry of Science wants to offer housing aid to a particular group of students by requiring information solely on the rental contracts and payslips of the students, and without disclosing their personal data, due to privacy and redundancy constraints. In such settings, conventional data compression or joint source compression schemes are suboptimal, as they store and transmit streams of source data regardless of task relevance. Our study targets this gap by investigating a framework for distributed functional compression, where sources separately encode their inputs using graph-based colorings, and a receiver computes the desired function without decoding all of the source variables. The scenario of student housing is an example of realizing functional compression, which avoids compressing and transmitting large volumes of all available data, and is instead tailored to the specifics of the function, i.e., a student's eligibility for getting housing aid or not.

In Shannon's breakthrough work in [93], the function to be recovered at the receiver is the identity function of the source variable, i.e., the source itself. Generalizing the noiseless coding of a discrete information source, given in [93], to distributed compression and joint decoding of two jointly distributed and finite alphabet random variables X_1 and X_2, Slepian-Wolf theorem gives a theoretical lower bound for the lossless coding rate of distributed coding of such sources [82], where the two data sequences of memoryless correlated sources with finite alphabets $\mathbf{X}_1^n$, and $\mathbf{X}_2^n$ are obtained by n repeated independent drawings from a discrete bivariate distribution. Practical implementation schemes for Slepian-Wolf compression have been proposed, including [34, 35, 105]. Although the model superficially resembles Slepian–Wolf, the key distinction is that the decoder does *not* aim to reconstruct X_1, and the optimal rate region therefore differs fundamentally from the Slepian–Wolf region. In functional compression of distributed sources X_1 and X_2, however, the goal is to compress the sources separately while ensuring that a deterministic function $f(X_1, X_2)$ of these sources can be calculated by a user. Prior attempts on functional compression can be categorized into works focusing on lossless and zero-error compression of functions, [63, 64, 73–75, 77, 78, 82, 83, 93, 100, 106–112], and those for which the compression schemes tolerate distortion for lossy reconstruction [37, 102, 113–117]. Among these, the zero-error version of the coding-for-computing problem has been explicitly revisited in the recent work of Charpenay–Le Treust–Roumy [76, 118], which extends the ideas of Witsenhausen [73] to functional side-information design.

Several special cases of distributed compression have been studied. In [107], Ahlswede and Körner have determined the rate region for a distributed compression setting where separate encoders encode the n-th realizations of correlated sources X_1 and X_2 observed by sources one and two, respectively, and a receiver aims to only recover $\mathbf{X}_2^n$. Körner and Marton, in [64], have dealt with the problem of distributed encoding of two binary sources X_1 and X_2 to compute their modulo-two sum, i.e., $f(X_1, X_2) = (X_1 + X_2) \mod 2$, at

the receiver. In [114], Yamamoto has extended the Wyner-Ziv model [102], which has addressed lossy source coding with side information at the decoder, to a setting where the decoder estimates a function $f(X_1, X_2)$ of the source X_1 given side information X_2. In [63], Han and Kobayashi have established an achievable distributed functional reconstruction scheme, which depends on the structure of $f(X_1, X_2)$, and the joint distribution of (X_1, X_2). Building on [114], optimal coding schemes and achievable rate regions have been derived for lossless and lossy compression of source X_1 for distributed computation of $f(X_1, X_2)$ given side information X_2 [74, 83, 119, 120], for distributed compression of sources X_1 and X_2 for the computation of $f(X_1, X_2)$ [77, 121]. More specifically, in [74], Orlitsky and Roche have provided a single letter characterization for general functions of two variables using the notion of *source characteristic graphs* (confusion graphs) introduced by Körner [72], where the vertices are the possible realizations of a source and the edges capture the function structure.

2.1.2 Overview and Contributions

In this chapter, we design a coding framework for the problem of distributed functional compression with two distributed sources having access to X_1 and X_2, respectively, each with a finite alphabet, and a receiver that wants to reconstruct the function $f(X_1, X_2)$ in an asymptotically lossless manner. To capture the structure of the function f in compression, we exploit the notion of *source characteristic graphs*. Given source variable X_1, the limits of color reuse are determined by chromatic numbers $\chi(G_{X_1})$[2] [73], and the fundamental limits of compression rate are characterized by Körner's graph entropy of G_{X_1}, i.e. $H_G(X_1)$ [72]. While prior works have focused on specific function classes or leveraged receivers' side information, we lack a framework that captures general computation tasks and the fundamental compression limits of distributed functional compression. To capture the computation structure abstracted by the function and characterize minimum achievable rates, we devise a graph-theoretic framework. This approach enables us to attain graph entropy for the given function, realized by computing minimum-entropy colorings of the n-fold OR products of the source characteristic graphs. As the vertex set size grows with n, determining minimum entropy colorings and bounding chromatic numbers become increasingly complex — even under structural regularities. Our framework addresses these challenges through spectral and expansion-based techniques for bounding rates for various classes of characteristic graphs. To address different computation scenarios, we examine several characteristic graph topologies, including cyclic graphs, denoted by C_i where i is the number of vertices, their generalizations to d-regular graphs denoted by $G_{d,V}$, and general characteristic graphs denoted by G, as motivated next.

Cycle graphs (or cyclic graphs) appear in many practical scenarios, such as periodic functions, and mod functions, which are widely used in cryptography, computer algebra

[2] In [73], Witsenhausen has considered zero-error compression using characteristic graphs with side information.

and science, and musical arts [122]. In cryptography, Caesar Ciphers, Rivest-Shamir-Adleman (RSA) algorithm [123], Diffie-Hellman [124], as well as Advanced Encryption Standard (AES) [125], International Data Encryption Algorithm (IDEA), are widely used for secure data transmission [126]. The calculation of checksums within serial numbers is another application of interest [127]. For example, ISBNs (International Standard Book Numbers) use mod 11 arithmetic for 10-digit ISBNs or mod 10 for 13-digit ISBNs to detect errors. In addition, International Bank Account Numbers (IBANs) use mod 97 arithmetic to identify mistakes in bank account numbers entered by users. Cyclic characteristic graphs allow for a more efficient reuse of colors compared to acyclic graphs with a higher average degree. Furthermore, cycles and their products — built to capture a source sequence $\mathbf{X}_1^n$ — have good connectivity properties, enabling an exact characterization of their chromatic numbers (and bounding their graph entropies).

d-regular graphs have broad applications ranging from representing network topologies to modeling social networks, coding theory for constructing error-correcting codes [128], random walks and Markov chains in analyzing state transitions [129], spectral graph theory providing insights into graphs' characteristics [130, 131], and fault-tolerant systems [128, 129, 132–135]. In general, d-regular graphs help model frameworks for structured data, such as graph neural networks [136], making them one of the main topics of investigation in this part.

The main contributions of this chapter can be summarized as follows:

- *Cyclic characteristic graphs:* We derive exact expressions for the degree of a vertex $x^n \in [V^n]$ in the n-fold OR product C_i^n (as detailed by Alon and Orlitsky in [75]) of cycles C_i (where $i = V$, see Proposition 2.2), denoted by $deg(x)$, and for the chromatic number of even cycles C_{2k}, denoted by $\chi(C_{2k})$ where $k \in \mathbb{Z}^+$. Then, we devise a polynomial-time[3] achievable coloring scheme for odd cycles C_{2k+1}, leveraging the structure of C_i, and its OR products (see Proposition 2.3). Given C_i, we investigate the largest eigenvalue of its adjacency matrix, and using that, we present bounds on the chromatic number (see Proposition 2.7). We also provide bounds on Körner's graph entropy of C_i (see Proposition 2.5).
- *d-regular characteristic graphs:* We characterize the exact degree of a vertex and the chromatic number of d-regular graphs, denoted by $G_{d,V}$, and their n-fold OR products $G_{d,V}^n$ (see Propositions 2.8 and 2.9). Additionally, given a d-regular graph, the concept of graph expansion helps determine how the corresponding OR products are related. Capturing the structure of the OR products graphs, we then present a lower bound on the expansion rate of $G_{d,V}^n$ (see Proposition 2.10).
- *General characteristic graphs:* Given a general graph, $G(\mathcal{V}, \mathcal{E})$, we calculate the degree of each vertex for its n-fold OR product (see Corollary 2.5). We present upper and lower bounds on the expansion rate (see Corollary 2.8). We then investigate the entropy of general characteristic graphs (see Proposition 2.12). We derive bounds on

[3]Finding a minimum entropy coloring in general graphs is an NP-hard problem [96].

the largest eigenvalue (see Corollary 2.7), the chromatic number (see Corollary 2.6), using the adjacency matrix of the n-fold OR product graph and the famous *Gershgorin Circle Theorem* (GCT), which is a theorem that identifies the range of the eigenvalues for a given square matrix [137]. We use GCT to bound eigenvalues of the adjacency matrix of a given graph n-fold OR product, via exploiting the structure of OR products (see Theorem 2.2, and Corollary 2.10).

By leveraging chromatic numbers and graph entropy bounds, our results, as outlined above, illustrate the connection between graph structures and entropy-based communication cost and provide insights for functional compression in distributed settings.

Notation: Letter X denotes a discrete random variable with distribution $P_X(x)$ over the finite alphabet $\mathcal{X}$, where x is a realization of X, and $\mathbf{X}^n = X_1, X_2, \ldots, X_n$ is an i.i.d. sequence with each $X_i \sim P_X(x)$. Matrices and vectors are denoted by boldface letters, e.g., $\mathbf{A}$. The joint distribution of source variables X_1 and X_2 is denoted by $P_{X_1,X_2}(x_1, x_2)$. Characteristic graphs G_{X_1} and G_{X_2} are used for computing a function $f(X_1, X_2)$. We denote the n-length realization of X_1 as $\mathbf{x}_1^n = x_{11}, \ldots, x_{1n}$, and similarly for X_2. Given $G(\mathcal{V}, \mathcal{E})$, $\chi(G)$ and $\chi_f(G)$ denote its chromatic and fractional chromatic numbers, respectively. The degree of vertex $x_k \in [V]$ is $deg(x_k)$, $d_{\max} = \max_{k\in[V]} deg(x_k)$ is the maximum degree, and $deg_{\text{avg}}(x_k)$ is the average degree overall $x_k \in [V]$ in G. A cycle graph with $i = V$ vertices is $C_i = G(\mathcal{V}, \mathcal{E})$, its n-fold OR product is $C_i^n = G(\mathcal{V}^n, \mathcal{E}^n)$, and $\mathcal{C}_i^j(l)$ denotes the distinct color set of sub-graph $l \in [V]$ in the j-fold OR product. The coloring distribution and color set of G are denoted by $\mathcal{C}_G$ and $\mathcal{C}(G)$, respectively. A d-regular graph on V vertices is $G_{d,V}$, and a complete graph and its n-fold OR product are denoted by K_i and K_i^n. $\mathcal{C}_G$ denotes the PMF of a valid coloring of G. For graph $G(\mathcal{V}, \mathcal{E})$, $\mathbf{J}_V$ and $\mathbf{I}_V$ are all-one and identity matrices of size $V \times V$. The adjacency matrix is $\mathbf{A}_f = (a_{xx'})_{1\le x,x'\le V}$, a symmetric $(0,1)$-matrix with $a_{xx} = 0$; $a_{xx'} = 1$ if x and x' are adjacent, and 0 otherwise. The determinant and trace of $\mathbf{A}_f$ are denoted by $\det(\mathbf{A}_f)$ and $trace(\mathbf{A}_f)$, respectively. The largest and smallest eigenvalues are $\lambda_1(\mathbf{A}_f)$ and $\lambda_V(\mathbf{A}_f)$. The set of all eigenvalues is $\sigma(\mathbf{A}_f)$, and $\vartheta(G^j)$ is the set of distinct eigenvalues of the adjacency matrix $\mathbf{A}_f^j$. Using GCT, the k-th eigenvalue interval is δ_k, for $k \in [V]$.

2.2 Technical Preliminary

This section introduces the fundamental concepts related to graph theory, such as degrees, independent sets, paths, cycles, d-regular graphs, and the notion of graph expansion [72, 138–141]. Furthermore, it discusses the concepts of characteristic graphs, the n-fold OR products of characteristic graphs, and traditional and fractional coloring of graphs [99, 141, 142].

2.2.1 Source Characteristic Graphs and Their OR Products

A graph is represented by $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = [V]$ denotes the set of its vertices, with cardinality $|\mathcal{V}| = V$, and $\mathcal{E}$ is the set of its edges, with cardinality $|\mathcal{E}| = E$.

Definition 2.1 (Degree of a vertex [138]). *Given $G(\mathcal{V}, \mathcal{E})$, the degree of a vertex $x_k \in [V]$ for $k \in [V]$, represented by $deg(x_k)$, is the number of edges it is connected to, i.e., the number of neighbors of $x_k \in [V]$. The average degree across nodes in G is denoted by deg_{avg} as follows:*

$$deg_{\text{avg}} = \frac{\sum_{x_k \in [V]} deg(x_k)}{V} \ . \tag{2.1}$$

We next introduce the concept of an independent set, which plays a critical role in determining a valid coloring of characteristic graphs that we detail in Section 2.2.2.

Definition 2.2 (Independent set, and maximal independent set [139]). *An independent set, IS_G, in $G(\mathcal{V}, \mathcal{E})$ is a subset of vertices of $\mathcal{V}$, such that no two are adjacent. A maximal independent set, $\Gamma(G)$, is an independent set in G that is not a subset of any other independent set of G. A maximum independent set is an IS_G with maximum cardinality, and its size is referred to as the independence number, denoted by $\alpha(G)$.*

In distributed functional compression with M source nodes, each holding $X_k \in \mathcal{X}_k$, $k \in [M]$, a receiver reconstructs $f(X_1, X_2, \ldots, X_M)$. To distinguish function outputs, each source k builds a characteristic graph G_{X_k} with vertex set $\mathcal{X}_k$ and edges defined by the function and joint source PMF.

Definition 2.3 (Source characteristic graphs [72]). *Let X_1 and X_2 be two distributed source variables with a joint distribution $P_{X_1,X_2}(x_1, x_2)$. Source one builds a characteristic graph $G_{X_1} = G(\mathcal{V}, \mathcal{E})$ for distinguishing the outcomes of a function $f(X_1, X_2)$, where $\mathcal{V} = \mathcal{X}_1$, and an edge $(x_1^1, x_1^2) \in \mathcal{E}$ if and only if there exists a $x_2^1 \in \mathcal{X}_2$ such that $\mathbb{P}(x_1^1, x_2^1) \cdot \mathbb{P}(x_1^2, x_2^1) > 0$ and $f(x_1^1, x_2^1) \neq f(x_1^2, x_2^1)$, i.e., these two vertices of G_{X_1} should be distinguished.*

Definition 2.4 (Entropy of a characteristic graph [72]). *Given a source random variable X_1 with characteristic graph $G_{X_1} = G(\mathcal{V}, \mathcal{E})$, the entropy of G_{X_1} is defined as*

$$H_{G_{X_1}}(X_1) = \min_{X_1 \in U_1 \in \Gamma(G_{X_1})} I(X_1; U_1) \ , \tag{2.2}$$

where $\Gamma(G_{X_1})$ represents the set of all MISs of G_{X_1} [75]. The notation $X_1 \in U_1 \in \Gamma(G_{X_1})$ indicates that the minimization is performed 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}.

From (2.2), it follows that $H_{G_{X_1}}(X_1) \leq H(X_1)$, yielding savings over [93]. Next, we introduce a path, which is a sequence of edges connecting vertices within a graph.

Definition 2.5 (Path and Hamiltonian path [143]). *Given an undirected $G(\mathcal{V}, \mathcal{E})$, a path is a sequence of vertices with distinct endpoint vertices, where each pair of consecutive vertices is connected by an edge, and no vertex is repeated. A path that includes every vertex of a graph exactly once is called a Hamiltonian path.*

Next, we discuss a specific measure that pertains to the paths within the graphs.

Definition 2.6 (The diameter of a graph). *The diameter of a graph $G(\mathcal{V}, \mathcal{E})$, denoted by $Dia(G)$, is defined as the shortest path between the most distant nodes in the graph.*

We next define the class of d-regular graphs that embed the special case of cycles.

Definition 2.7 (d-regular graphs [140, 141]). *A d-regular graph $G_{d,V}(\mathcal{V}, \mathcal{E})$ is a graph where each vertex has the same degree d, i.e., $d = deg(x_k)$ for all $x_k \in [V]$. A d-regular graph $G_{d,V}$, where $d, V \in \mathbb{Z}^+$, satisfies $V \geq d + 1$. Furthermore, if d is odd, the total number of vertices V must be even [144]. Cyclic graphs are 2-regular graphs with a Hamiltonian path.*

We next define the expansion rate of regular graphs.

Definition 2.8 (Expansion rate [145]). *Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected graph. For a subset $Y \subseteq [V]$, the external neighborhood of Y is defined as*

$$N_G(Y) = \{u \in [V] \setminus Y : \exists\, v \in Y \text{ such that } (v, u) \in \mathcal{E}\} \ . \tag{2.3}$$

The expansion rate *of G with respect to Y is*

$$E_\theta(Y) = \frac{|N_G(Y)|}{|Y|}. \tag{2.4}$$

The (vertex) expansion of the graph G is then defined as

$$\phi(G) = \min_{\emptyset \neq Y \subseteq [V],\, |Y| \leq |V|/2} E_\theta(Y) \ . \tag{2.5}$$

We say that G is an expander graph *if $\phi(G)$ is bounded below by a positive constant independent of $|V|$*[4].

We next illustrate the concept of characteristic graphs with an example.

Example 2.1. *For distributed compression of $f(X_1, X_2) = (X_1 + X_2) \mod 2$, with two source variables X_1 and X_2 and one receiver, where X_1 and X_2 are uniform over the alphabets $\mathcal{X}_1 = \{0, 1, 2, 3\}$, and $\mathcal{X}_2 = \{0, 1\}$, respectively. For even-valued, i.e., $X_1 \in \{0, 2\}$, the output is $f(X_1, X_2) = X_2$, and for odd values $X_1 \in \{1, 3\}$, we have $f(X_1, X_2) = (X_2 + 1) \mod 2$. In the characteristic graph built for X_1, namely G_{X_1}, we do not need to distinguish the elements of $\{0, 2\}$ from each other, and similarly for the elements of*

[4]Expansion is a function of Y, and we omit it for simplicity of presentation.

$\{1,3\}$. *However, these two sets must be distinguished, which is possible using two distinct colors* B *and* O. *To that end, we assign the elements of* $\{0,2\}$ *and* $\{1,3\}$ *colors* B *and* O, *respectively. Similarly, the outcome* $X_2 = 1$ *is assigned* R, *and* $X_2 = 0$ *is assigned* Y. *We illustrate the coloring of* G_{X_1} *and* G_{X_2} *in Figure 2.1. Transmission of an assigned color pair from distributed sources to a common receiver according to the described rule is sufficient for the receiver to determine the corresponding outcome of* f *via a look-up table [77]. The scheme satisfies the necessary condition since both* G_{X_1} *and* G_{X_2} *require at least 2 colors (1 bit per source). Using fewer colors by assigning the same color to adjacent vertices violates the condition* $f(x_1^1, x_2^1) \neq f(x_1^2, x_2^1)$ *when* $\mathbb{P}(x_1^1, x_2^1) \cdot \mathbb{P}(x_1^2, x_2^1) > 0$.

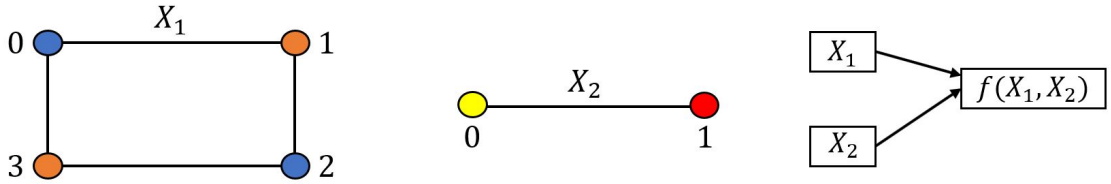

Figure 2.1: Distributed functional compression with two sources and a receiver, where G_{X_1} is cyclic.

To determine the fundamental limits of lossless compression of a source sequence [75], we exploit the notion of *n-fold OR products* of graphs[5], as introduced next.

Given a source variable X, we next detail the construction for the n-fold OR products graphs, by generalizing the rule in Definition 2.3 for building G_X to multiple source instances $\mathbf{X}^n$.

Definition 2.9 (n-fold OR product graph [99, 141, 142])**.** *For* $n > 1$, *the n-fold OR product of* $G_X = G(\mathcal{V}, \mathcal{E})$ *is represented as* $G_X^n = G(\mathcal{V}^n, \mathcal{E}^n)$, *where* $\mathcal{V}^n = \mathcal{X}^n$, *and given two distinct vertices* $\mathbf{x}_1^n = x_{11}^1, \ldots, x_{1n}^1 \in [V^n]$, *and* $\mathbf{x}_2^n = x_{21}^1, \ldots, x_{2n}^1 \in [V^n]$, *it holds that* $(\mathbf{x}_1^n, \mathbf{x}_2^n) \in \mathcal{E}^n$, *when* $\exists$ *at least one* $k \in [n]$ *such that* $(x_{1k}^1, x_{1k}^2) \in \mathcal{E}$.

The n-fold OR product has been used in the asymptotically lossless compression of distributed sources for function computation, see e.g., [74, 77, 78]. Building on [72], the authors in [74] have demonstrated that given two distributed source variables X_1 and X_2, the lowest sum rate needed for distributed computing of $f(X_1, X_2)$ can be achieved by encoding the n-fold OR product graphs $G_{X_1}^n$ and $G_{X_2}^n$, in the limit as n goes to infinity.

We next describe how to iteratively determine n-fold OR product G_X^n, from the $(n-1)$-fold OR product graph G_X^{n-1}.

Definition 2.10 (Sub-graphs of G_X^n)**.** *Given an n-fold OR product graph* G_X^n, *we denote the collection of graphs* $\{G_X^n(l)\}_{l \in [V]}$ *in* G_X^n *as the sub-graphs of* G_X^n, *where each of* $G_X^n(l)$ *represents* $(n-1)$*-fold OR product of* G_X *with itself.*

We next define *'full connection'* (links between sub-graphs).

[5]Here, we focus on OR product graphs for lossless compression of multi-letter schemes [146]. For zero-error source compression, the n-fold AND product of graphs must be used [147]

Definition 2.11 (Full connection between two graphs)**.** *Given $G_1(\mathcal{V}_1, \mathcal{E}_1)$ and $G_2(\mathcal{V}_2, \mathcal{E}_2)$, if for each $x_{1k} \in [V_1]$, and each $x_{2t} \in [V_2]$, there exists an edge (x_{1k}, x_{2t}), i.e., the bipartite graph formed between $[V_1]$ and $[V_2]$ is complete, we describe the two graphs as having a full connection.*

2.2.2 Coloring of Characteristic Graphs

A valid (proper) vertex coloring of G_{X_1} assigns a color to each vertex such that adjacent vertices receive distinct colors, indicating which source realizations need different codes. Non-adjacent vertices may share the same color, known as *traditional graph coloring*. A valid coloring with minimum entropy provides a lower bound on the compression rate for lossless function reconstruction. The minimum number of colors for such coloring is the chromatic number[6], $\chi(G_{X_1})$. *Fractional graph coloring* generalizes this by assigning color sets to each $x_k \in [V]$ so that adjacent vertices have disjoint sets [78, 148]. Given its connection to the minimum graph entropy [72], the fractional chromatic number, which is a lower bound on $\chi(G)$, is also important to study. We next detail how to obtain a valid b-fold coloring out of a available colors.

Definition 2.12 (The fractional chromatic number)**.** *A valid b-fold coloring assigns sets of distinct colors with cardinality b to each vertex such that adjacent vertices receive disjoint sets of b colors. A valid $a : b$ coloring assigns b colors to each vertex from a coloring set of a colors. The notation $\chi_b(G)$ represents the b-fold chromatic number of G, with the smallest a (i.e., total number of colors) such that an $a : b$ coloring exists. The fractional chromatic number of G is*

$$\chi_f(G) = \lim_{b\to\infty} \frac{\chi_b(G)}{b} = \inf_b \frac{\chi_b(G)}{b} \ , \tag{2.6}$$

where $\chi_b(G) \in \mathbb{Z}^+$, and the second equality follows from the subadditivity of $\chi_b(G)$ [148].

Given a graph G, the fractional chromatic number of its n-fold product is computed as [78]:

$$\chi_f(G^n) = (\chi_f(G))^n \ . \tag{2.7}$$

Traditional coloring is a special case of the valid $a : b$ coloring of G^n where $b = 1$ and $a = \chi(G^n)$.

To study a general characteristic graph G, its chromatic number and entropy, we use eigenvalue relations of its adjacency matrix by defining the GCT for square and block matrix representations to handle the coloring of general graphs.

[6] In general, the problem of determining $\chi(G_{X_1})$ is NP-complete [96].

2.2.3 Gershgorin Circle Theorem (GCT)

To investigate a general characteristic graph G and its chromatic number and entropy, we aim to leverage the eigenvalue relationships of its adjacency matrix. Therefore, we define the GCT for square and block matrix representations to address coloring in general characteristic graphs.

Definition 2.13 (GCT for square matrices [137])**.** *Let $\mathbf{A} \in \mathbb{C}^{V\times V}$ be a square matrix with entries a_{kt}, $k,t \in [V]$. For each $k \in [V]$, define the* Gershgorin disc

$$D_k = \Big\{ z \in \mathbb{C} : |z - a_{kk}| \leq \sum_{t\neq k} |a_{kt}| \Big\}. \tag{2.8}$$

Then every eigenvalue of $\mathbf{A}$ lies in at least one of these discs, i.e.,

$$\sigma(\mathbf{A}) \subseteq \bigcup_{k=1}^{V} D_k, \tag{2.9}$$

where $\sigma(\mathbf{A})$ denotes the set of all eigenvalues of $\mathbf{A}$, $\sigma(\mathbf{A}) = \{\lambda_l(\mathbf{A}) : l \in [V]\}$.

Given that $\mathbf{A}_f^n$ is a block matrix, we next define GCT for block matrices.

Definition 2.14 (GCT for block matrices [98, 137])**.** *Let $\mathbf{A} \in \mathbb{R}^{mn\times mn}$ be a symmetric block matrix, partitioned into $n \times n$ blocks $\mathbf{A}_{kt} \in \mathbb{R}^{m\times m}$, for $k,t \in [n]$. For each $k \in [n]$, define the* block Gershgorin radius

$$R_k = \sum_{t\neq k} \|\mathbf{A}_{kt}\|, \tag{2.10}$$

where $\|\cdot\|$ is any matrix norm (e.g., the spectral norm or Frobenius norm)[7]*. For each eigenvalue $\lambda \in \sigma(\mathbf{A}_{kk})$ of the diagonal block $\mathbf{A}_{kk}$, define the corresponding disc*

$$D^b_{k,\lambda} = \{ z \in \mathbb{C} : |z - \lambda| \leq R_k \}. \tag{2.11}$$

Then every eigenvalue of $\mathbf{A}$ lies in at least one of these discs:

$$\sigma(\mathbf{A}) \subseteq \bigcup_{k=1}^{n} \Big(\bigcup_{\lambda\in\sigma(\mathbf{A}_{kk})} D^b_{k,\lambda} \Big) . \tag{2.12}$$

From (2.11), we deduce that the regions (circles) covering the eigenvalues of $\mathbf{A}$ are centered at eigenvalues of $\mathbf{A}_{kk}$, and circles' radius are enlarged by the size of the off-diagonal matrices $\mathbf{A}_{kt}$. Given an n-fold OR product graph with an adjacency matrix $\mathbf{A}_f^n$,

[7] In (2.10), instead of the entry-wise absolute sum, one could equivalently use a matrix norm (e.g., the Frobenius or L_∞ norm) to ensure dimensionally consistent bounds.

where $\mathbf{A}_f^n$ is a $V^n \times V^n$ binary and symmetric matrix, the circles $D_{k,\lambda}^b$ in (2.11) can be simplified to block intervals $\delta_{k,\lambda}^b$.

2.3 Bounds On Cyclic and Regular Graphs

We here detail characteristic graphs that are d-regular and derive lower and upper bounds on their chromatic numbers and graph entropies. Our novel contributions include the characterization of chromatic numbers for n-fold OR products of cycles, as well as a novel coloring scheme for OR products of odd cycles, as detailed in Section 2.3.1. In Section 2.3.2, we bound the entropy of characteristic graphs for cycles. Given a cyclic graph, in Section 2.3.3, we analyze its key properties using the eigenvalues of its adjacency matrix, and in Section 2.3.4, we bound its chromatic number. In Section 2.3.5, we characterize the degrees and chromatic numbers for regular graphs and their OR products. Section 2.3.6 details the expansion rates of OR products of regular graphs, exploiting the connectivity properties of OR products, with implications on the fundamental limits of compressibility of such graphs.

2.3.1 Coloring Cyclic Graphs

Let C_i be a cycle graph with i vertices, that represents the characteristic graph that source X_1 builds for computing f (similarly for source X_2), i.e., $G_{X_1} = C_i$. For an even cycle $i = 2k$, and for an odd cycle, $i = 2k+1$, for some $k \in \mathbb{Z}^+$. We seek to compress G_{X_1} and G_{X_2} to recover the desired function outcome at a receiver in an asymptotically lossless manner. To that end, we will determine the minimum entropy coloring for the n-fold OR product of C_i, denoted by C_i^n (and similarly for G_{X_2}), for the receiver to decode f from the received colors.

We start by determining the degree of each vertex in C_i^j for $j \in [n]$.

Proposition 2.1 (Degree of vertices in C_i^n)**.** *The degrees*[8] *in the n-fold OR product of a cycle graph, C_i^n for $n \geq 2$, are calculated as follows:*

$$deg(x^n) = 2 \cdot \frac{V^n - 1}{V - 1} \ , \quad \forall x^n \in [V^n] \ . \tag{2.13}$$

Proof. See Appendix A.1. □

From Proposition 2.1, we infer that for a given pair of vertices $x_t, x_k \in [V]$ where $t \neq k$, if $deg(x_t) = deg(x_k)$, then for the n-fold OR product, $deg(x_t^n) = deg(x_k^n)$, for $x_t^n, x_k^n \in [V^n]$, i.e., taking the n-fold OR products does not alter the equality of degrees. Therefore, for any d-regular graph (including cycles), we derive the following result about the regularity of its OR products.

[8] In regular graphs, where all vertices have the same degree, we omit the index k of x_k in Propositions 2.1 and 2.8.

Corollary 2.1. *Given a d-regular graph $G_{d,V}$, where $d = deg(x)$, its n-fold OR product with itself for $n \geq 1$, i.e., $G^n_{d,V}$, is also a regular graph, with degree $deg(x^n)$, and total number of edges*

$$E^n = \frac{\left(\sum_{k=1}^{V^n} deg(x^n)\right)}{2}.$$

Even Cycles

Here, we consider even cycles, denoted by C_{2k}, $k \in \mathbb{Z}^+$. The vertices of C_{2k} are sequentially numbered clockwise from 0 to $2k-1$ (e.g., see G_{X_1} in Figure 2.1), and alternatingly colored. Vertices with even indices are assigned one color, while those with odd indices receive another. We next determine the chromatic number of C^n_{2k}, denoted as $\chi(C^n_{2k})$.

Proposition 2.2 (Chromatic number of C^n_{2k})**.** *The chromatic number of C^n_{2k} is given as*

$$\chi(C^n_{2k}) = 2^n \ , \quad k \in \mathbb{Z}^+ \ , \quad n \geq 1 \ . \tag{2.14}$$

Proof. Given C_{2k}, with $\chi(C_{2k}) = 2$, its 2-fold OR product C^2_{2k} consists of $(2k)^2$ vertices and $2k$ sub-graphs, $\{C^2_{2k}(1), C^2_{2k}(2), \ldots, C^2_{2k}(2k)\}$, each containing $2k$ vertices. Since each sub-graph is two-colorable and fully connected to its neighbors, adjacent sub-graphs must use different colors. For instance, $\{C^2_{2k}(1), C^2_{2k}(2)\}$ requires 4 colors in total. However, due to the cyclic structure of OR products, alternating colors from $C^2_{2k}(1)$ can cover $C^2_{2k}(3)$, and similarly for odd-indexed sub-graphs, meaning that $\chi(C^2_{2k}) = 4$. This method can also calculate $\chi(C^n_{2k})$ from $(n-1)$-fold to n-fold OR products. Figure 2.2 shows a valid coloring for C^3_4, where $\chi(C^3_4) = 8$. Similarly, by induction, $\chi(C^n_{2k})$ satisfies (2.14). □

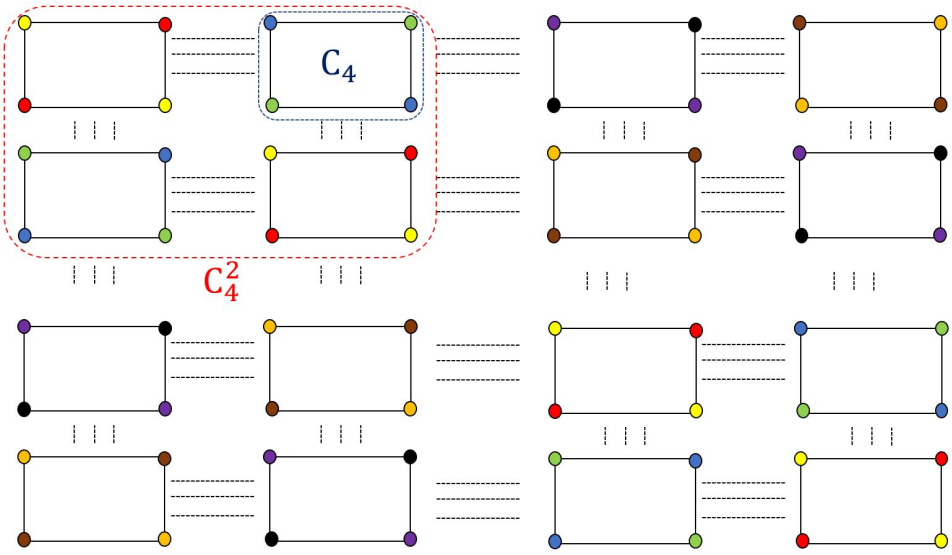

Figure 2.2: A valid coloring of C^3_4 with 8 colors.

Odd Cycles

We here focus on odd cycles, namely C_{2k+1}, $k \in \mathbb{Z}^+$, and their n-fold OR products. In the special case with 3 vertices, C_3 is a complete graph, and a valid coloring requires 3 distinct colors for a receiver to successfully recover the function. Furthermore, for a valid coloring of C_3^n, $\chi(C_3^n) = 3^n$ for $n \geq 1$. For coloring of an odd cycle with the length $i = 2k+1$, for $k \geq 2$, one could reuse the colors. For instance, given C_5, we have $\chi(C_5) = 3$. We next present an achievable scheme for valid colorings of general odd cycles.

Proposition 2.3 (Chromatic numbers of odd cycles)**.** *The chromatic number of C_i^{n+1}, denoted as $\chi(C_i^{n+1})$, can be recursively computed from $\chi(C_i^n)$ as follows:*

$$\chi(C_i^{n+1}) = 2\chi(C_i^n) + \left\lceil \frac{\chi(C_i^n)}{2} \right\rceil \quad , \; i = 2k+1 \text{ and } k \in \mathbb{Z}^{\geq 2}. \tag{2.15}$$

Proof. See Appendix A.2. □

For even cycles, from Proposition 2.2, $\chi(C_i^n) = 2^n$. For odd cycles, in Section 2.3.4, we will establish upper and lower bounds on $\chi(C_i^n)$ using the adjacency matrix of C_i^n, denoted as $\mathbf{A}_f^n$.

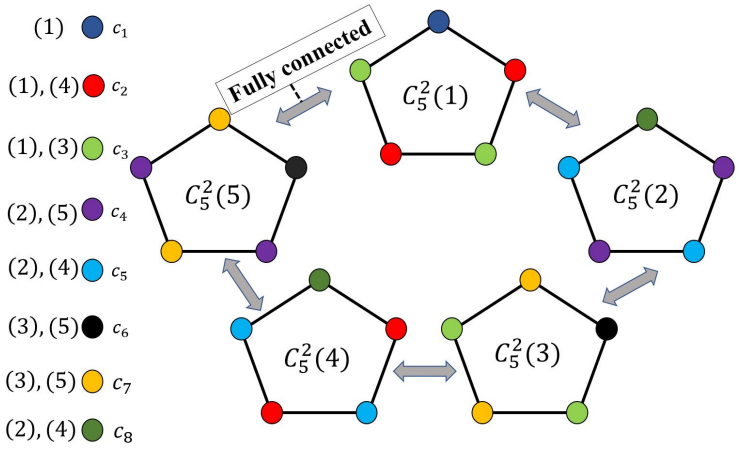

Figure 2.3: The 2-fold OR product of C_5, i.e., C_5^2, and its valid coloring.

We next demonstrate the gain in terms of the required number of colors of our approach in Proposition 2.3 over a greedy algorithm that does not leverage the structure of C_{2k+1}^n in coloring.

Proposition 2.4 (The multiplicative gain of our approach over a greedy coloring algorithm)**.** *The gain of the recursive coloring approach in Proposition 2.3 for C_{2k+1}^n, $k \in \mathbb{Z}^{\geq 2}$, over the greedy algorithm, which calculates $\chi(C_{2k+1})$ and uses $(\chi(C_{2k+1}))^n$ colors for coloring C_{2k+1}^n, is*

$$\eta_n = \frac{\left(\chi(C_{2k+1})\right)^n}{\chi(C_{2k+1}^n)} \geq 1.2^n \ , \tag{2.16}$$

which is exponential and unbounded as $n \to \infty$*, i.e.,* $\eta = \lim_{n\to\infty} \eta_n = \infty$.

Proof. See Appendix A.3. □

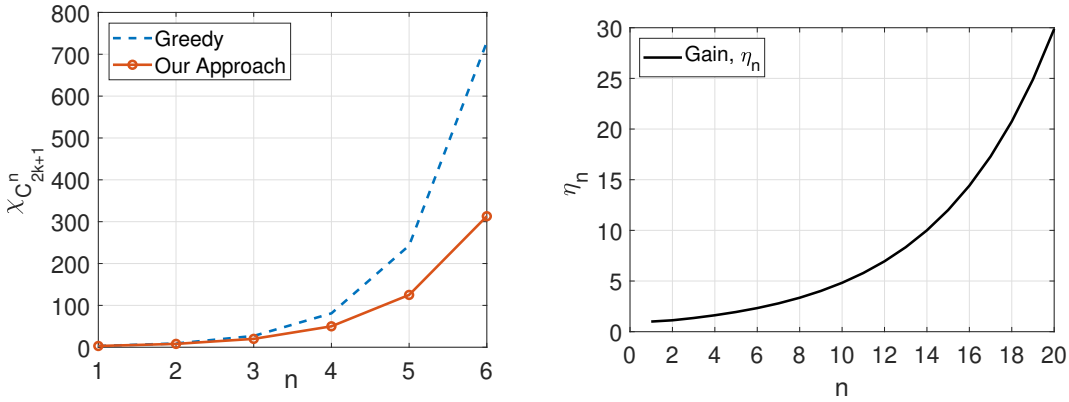

Figure 2.4: (Left) $\left(\chi(C_{2k+1})\right)^n$ (dashed (blue) curve), and $\chi(C^n_{2k+1})$ (solid (orange) curve) for any $k \geq 2$. (Right) The gain, i.e., η_n, of the coloring approach in Proposition 2.3 compared to the Greedy algorithm.

2.3.2 Bounding the Chromatic Entropy of Cycles

Next, we establish an upper bound on the chromatic entropy of C^n_i, for $i \in \mathbb{Z}^+$.

Entropy of an even cycle

From Proposition 2.2, $\chi(C^n_{2k}) = 2^n$. We recall that the chromatic entropy $H^{\chi}_{C^n_{2k}}(\mathbf{X}_1)$, which is the minimum achievable entropy of a valid coloring of C^n_{2k} [72, 75], and the characteristic graph entropy $H_{C_{2k}}(X_1)$ are related as follows [75, Theorem 5]:

$$H_{C_{2k}}(X_1) = \lim_{n\to\infty} \frac{1}{n} H^{\chi}_{C^n_{2k}}(\mathbf{X}_1) \ , \tag{2.17}$$

where

$$H^{\chi}_{C^n_{2k}}(\mathbf{X}_1) = \min_{\mathcal{C}_{C^n_{2k}}} H(\mathcal{C}_{C^n_{2k}}) \ . \tag{2.18}$$

If the distribution of C^n_{2k} is uniform, then $H_{C_{2k}}(X_1)$ is given as follows:

$$H_{C_{2k}}(X_1) = \lim_{n\to\infty} \frac{1}{n} \log_2 2^n = 1 \quad \text{bits} \ . \tag{2.19}$$

Entropy of an odd cycle

We next examine the chromatic and graph entropies of C_{2k+1}. Unlike C_{2k}, the coloring PMF for odd cycles, $P(\mathcal{C}_{C^n_{2k+1}})$, is non-uniform (i.e., $H^{\chi}_{C^n_{2k+1}}(\mathbf{X}_1) < \log_2 \chi(C^n_{2k+1})$), as demonstrated through the examples below.

Example 2.2 (Entropies of C_5 and C_5^2)**.** *Given that X_1 follows a uniform distribution over an alphabet $\mathcal{X}_1$ such that $|\mathcal{X}_1| = 5$ and with a characteristic graph $G_{X_1} = C_5$, the coloring PMF satisfies, $P(\mathcal{C}_{C_5}) = \{\frac{1}{5}, \frac{2}{5}, \frac{2}{5}\}$ with $\chi(C_5) = 3$, and using (2.18) for chromatic entropy, yields $H^{\chi}_{C_5}(X_1) = \min_{\mathcal{C}_{C_5}} H(\mathcal{C}_{C_5}) = 1.52$. For the 2-fold OR product graph C_5^2, where the PMF that satisfies the minimum coloring entropy is $P(\mathcal{C}_{C_5^2}) = \{\frac{4}{25}, \frac{4}{25}, \frac{4}{25}, \frac{4}{25}, \frac{4}{25}, \frac{2}{25}, \frac{2}{25}, \frac{1}{25}\}$ with $\chi(C_5^2) = 8$, similarly using (2.18) we obtain*

$$\frac{1}{2}H(\mathcal{C}_{C_5^2}) = 1.37 < H(\mathcal{C}_{C_5}) = 1.52 \quad \textit{bits} \; . \tag{2.20}$$

While from Example 2.2, we can determine $H^{\chi}_{C_5}(X_1)$ and $H^{\chi}_{C_5^2}(\mathbf{X}_1)$, determining $H^{\chi}_{C_5^n}(\mathbf{X}_1)$, which corresponds to the minimum entropy among all possible valid colorings, becomes complex for large n. Next, given a characteristic graph $G_{X_1} = C_{2k+1}$, we establish an upper bound on $H_{C_{2k+1}}(X_1)$ by employing (2.17) and (2.18) and devising valid colorings for the MISs of C^n_{2k+1}.

Proposition 2.5 (An upper bound on $H_{C_{2k+1}}(X_1)$)**.** *The entropy is upper bounded as follows:*

$$H_{C_{2k+1}}(X_1) \leq \frac{1}{n} H\Big(\zeta_n \cdot \Big(\frac{k^n}{(2k+1)^n}\Big), \zeta_{n-1} \cdot \Big(\frac{k^{(n-1)}}{(2k+1)^n}\Big), \ldots, \zeta_0 \cdot \Big(\frac{1}{(2k+1)^n}\Big)\Big) \; , \tag{2.21}$$

where $\zeta_t \; , \; t \in [n] \cup \{0\}$, represents the number of maximum independent sets with size $\alpha(C^t_{2k+1})$, and ζ_n satisfies

$$\frac{(2k+1)^n \cdot (k^2-1)}{k^{2(n+1)}-1} \cdot k^n < \zeta_n < \frac{(k^{2(n+1)}-1) \cdot (2k+1)^n \cdot k^{(n-1)}}{(2k+1)^{2n} \cdot (k^2-1) \cdot k^{(n-1)} - (k^{2(n+1)}-1)} \; . \tag{2.22}$$

Proof. See Appendix A.4. □

In (2.21), as color reuse increases (i.e., independent sets with high cardinality and high ζ_n), $H_{C_{2k+1}}(X_1)$ decreases. We later generalize Proposition 2.5 to general graphs in Section 2.4.3.

Next, we apply Proposition 2.5 to C_5^3 to derive an upper bound to $H(\mathcal{C}_{C_5^3})$.

Example 2.3 (Bounds on the entropy of C_5^3)**.** *The cardinalities of the MISs for different graph products C_5^j, where $j \in [3] \cup \{0\}$, are given as $\alpha(C_5^0) = 2^0$, $\alpha(C_5^1) = 2^1$, $\alpha(C_5^2) = 2^2$, and $\alpha(C_5^3) = 2^3$, respectively, where $\chi(C_5^3) = 20$, which can be recursively computed using (2.15). Employing Proposition 2.5 yields*

$$\zeta_3 \cdot \frac{8}{125} + \zeta_2 \cdot \frac{4}{125} + \zeta_1 \cdot \frac{2}{125} + \zeta_0 \cdot \frac{1}{125} = 1 \; . \tag{2.23}$$

Employing the ordering $\zeta_3 > \zeta_2 > \zeta_1 > \zeta_0$ in (2.23) leads to the simplification, $64\zeta_0 + 16\zeta_0 + 4\zeta_0 + \zeta_0 \leq 125$. Hence, $\zeta_0 \leq \frac{25}{17}$. Because $\zeta_t \in \mathbb{Z}^+$, $t \in [3] \cup \{0\}$, we have $\zeta_0 = 1$. Employing the same ordering in (2.23) also yields the simplification $8\zeta_3 + 4(\frac{\zeta_3}{2}) + 2(\frac{\zeta_3}{4}) + \frac{\zeta_3}{8} \geq 125$, which yields the condition $\zeta_3 \geq \frac{200}{17}$. Because $\zeta_t \in \mathbb{Z}^+$, (2.23) yields

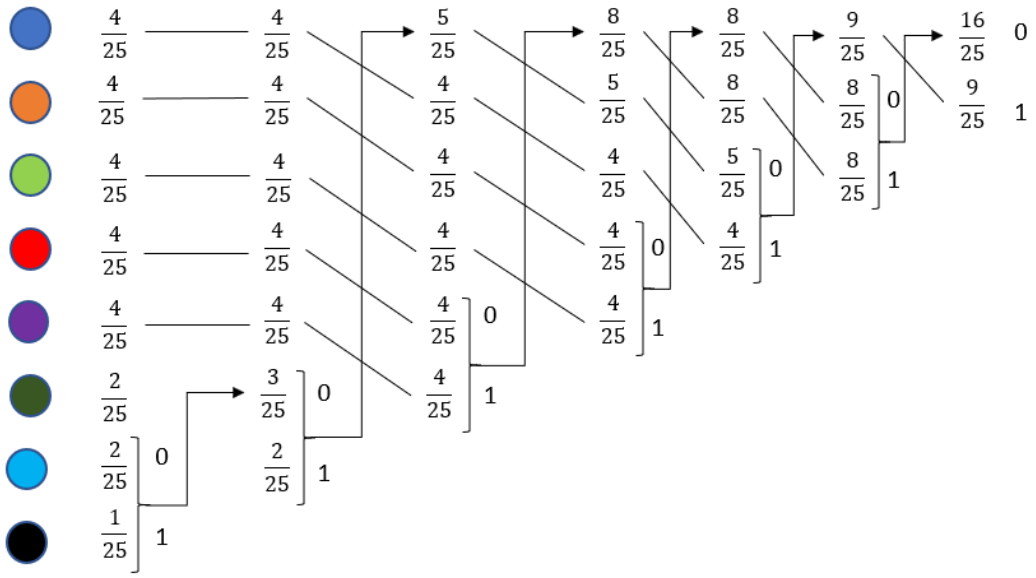

Figure 2.5: Huffman code construction given the coloring PMF for C_5^2.

the upper bound $\zeta_3 \leq 15$. *Employing these lower and upper bounds to (2.21) leads to* $1.37 \leq \frac{1}{3} H_{C_5^3}^{\chi}(\mathbf{X}_1) \leq 1.41$.

Next, we consider another example for C_5^3, where we employ Proposition 2.5 to determine ζ_t's and $P(\mathcal{C}_{C_5^3})$ (similar to Examples 2.2-2.3). We then apply Huffman coding [149] to the coloring random variable to optimize the compression rate of C_5^3.

Example 2.4 (Huffman coding for a given characteristic graph coloring)**.** *Consider the setting of Example 2.2, where* $\chi(C_5) = 3$. *Maximum independent sets of* C_5 *are represented by the colors* Y, B, *and* R *with PMF* $P(\mathcal{C}_{C_5}) = \left\{\frac{1}{5}, \frac{2}{5}, \frac{2}{5}\right\}$. *To achieve* $H_{C_5}^{\chi}(X_1)$ *approximately, a binary Huffman tree is constructed for each color. The assigned codes are* $Y : 1$, $R : 00$, *and* $B : 01$. *Similarly, for* C_5^2, *from (2.15),* $\chi(C_5^2) = 8$. *The color set is denoted as* $\mathcal{C}(C_5^3) = \{c_1, c_2, \ldots, c_8\}$, *with the corresponding Huffman codes:* $c_1 : 11$, $c_2 : 000$, $c_3 : 001$, $c_4 : 010$, $c_5 : 011$, $c_6 : 101$, $c_7 : 1000$, *and* $c_8 : 1001$ *as illustrated in Figure 2.5. For* C_5^3, $\chi(C_5^3) = 20$, *we have the following coloring PMF:*

$$P(\mathcal{C}_{C_5^3}) = \left\{\frac{8}{125}, \ldots, \frac{8}{125}, \frac{4}{125}, \ldots, \frac{4}{125}, \frac{2}{125}, \frac{2}{125}, \frac{1}{125}\right\}, \tag{2.24}$$

where there are $\zeta_3 = 13$ *colors with probability* $\frac{8}{125}$, $\zeta_2 = 4$ *colors with probability* $\frac{4}{125}$, $\zeta_1 = 2$ *colors with probability* $\frac{2}{125}$, *and* $\zeta_0 = 1$ *color with probability* $\frac{1}{125}$, *where the set of* ζ_t*'s are uniquely determined employing* $\chi(C_5^3) = \sum_{t=0}^{3} \zeta_t = 20$ *and* $\zeta_t \geq \zeta_{t-1}$. *Building the binary Huffman encoding tree using* $P(\mathcal{C}_{C_5^3})$ *in (2.24), helps assign codes for encoding* $\mathcal{C}_{C_5^3}$, *that achieves the minimum average code length.*

Next, we examine the relationship between the chromatic numbers of cycles and the eigenvalues of their adjacency matrix.

2.3.3 Eigenvalues of the Adjacency Matrices of C_i

Let $\mathbf{A}_f \in \{0,1\}^{V\times V}$ be the adjacency matrix of $G(\mathcal{V},\mathcal{E})$, where $\mathbf{A}_f(l_1,l_2)=1$ if distinct vertices $l_1,l_2 \in [V]$ must be distinguished, and 0 otherwise. Since there are no self-loops, $\mathbf{A}_f(l_1,l_1)=0$. The adjacency matrix of the 2-fold OR product, $\mathbf{A}_f^2$, is composed of diagonal blocks of $\mathbf{A}_f$, where entries equal to 1 are replaced with all-one matrices $\mathbf{J}_V$, and zeros with all-zero matrices $\mathbf{Z}_V$, representing full or no connectivity between sub-graphs $C_i^2(l)$ for $l \in [V]$,

$$\mathbf{A}_f^2 = \begin{bmatrix} \mathbf{A}_f & \mathbf{J}_V & \mathbf{Z}_V & \dots & \mathbf{Z}_V & \mathbf{J}_V \\ \mathbf{J}_V & \mathbf{A}_f & \mathbf{J}_V & \dots & \mathbf{Z}_V & \mathbf{Z}_V \\ \mathbf{Z}_V & \mathbf{J}_V & \mathbf{A}_f & \mathbf{J}_V & \dots & \mathbf{Z}_V \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{Z}_V & \dots & \mathbf{Z}_V & \mathbf{J}_V & \mathbf{A}_f & \mathbf{J}_V \\ \mathbf{J}_V & \mathbf{Z}_V & \mathbf{Z}_V & \dots & \mathbf{J}_V & \mathbf{A}_f \end{bmatrix}. \tag{2.25}$$

Similarly, to (2.25), we can construct $\mathbf{A}_f^n$ for the n-fold OR product using induction. Given the n-fold OR product of C_i with itself, the adjacency matrix of C_i^n, which we denote by $\mathbf{A}_f^n$, has the following block structure:

$$\mathbf{A}_f^n = \begin{bmatrix} \mathbf{A}_f^{n-1} & \mathbf{J}_{V^{n-1}} & \mathbf{Z}_{V^{n-1}} & \dots & \mathbf{Z}_{V^{n-1}} & \mathbf{J}_{V^{n-1}} \\ \mathbf{J}_{V^{n-1}} & \mathbf{A}_f^{n-1} & \mathbf{J}_{V^{n-1}} & \mathbf{Z}_{V^{n-1}} & \dots & \mathbf{Z}_{V^{n-1}} \\ \mathbf{Z}_{V^{n-1}} & \mathbf{J}_{V^{n-1}} & \mathbf{A}_f^{n-1} & \mathbf{J}_{V^{n-1}} & \mathbf{Z}_{V^{n-1}} & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{Z}_{V^{n-1}} & \dots & \mathbf{Z}_{V^{n-1}} & \mathbf{J}_{V^{n-1}} & \mathbf{A}_f^{n-1} & \mathbf{J}_{V^{n-1}} \\ \mathbf{J}_{V^{n-1}} & \mathbf{Z}_{V^{n-1}} & \mathbf{Z}_{V^{n-1}} & \dots & \mathbf{J}_{V^{n-1}} & \mathbf{A}_f^{n-1} \end{bmatrix}, \tag{2.26}$$

consisting of V row-block matrix partitions of size V^{n-1} each. In every block row of $\mathbf{A}_f^n$, there are exactly two $\mathbf{J}_{V^{n-1}}$ matrices and one $\mathbf{A}_f^{n-1}$, i.e., the adjacency matrix of the $(n-1)$-fold OR product of C_i. We next characterize the eigenvalues of the all-ones matrix $\mathbf{J}_V$ of size $V\times V$, which represents full connectivity between adjacent sub-graphs in the 2-fold OR product (see Definition 2.9). The proof of the following lemma is given in [103, Lemma 1].

Lemma 2.1. *The eigenvalues of the all-ones matrix $\mathbf{J}_V \in 1^{V\times V}$ are 0 and V, with algebraic multiplicities $V-1$ and 1, respectively.*

From Lemma 2.1, we have $\lambda_1(\mathbf{J}_V)=V$. To calculate the eigenvalues of $\mathbf{A}_f^n$, one needs to solve for $\sigma(\mathbf{A}_f^n) \triangleq \{\lambda \in \mathbb{R}:\ \det(\mathbf{A}_f^n - \lambda \mathbf{I}_{V^n})=0\}$, which is the set of all eigenvalues of $\mathbf{A}_f^n$. Let $\{\lambda_k(\mathbf{A}_f),\ k\in[V]\}$ be the set of eigenvalues of $\mathbf{A}_f$, and $\{\nu_k(\mathbf{A}_f^2),\ k\in[V^2]\}$ be the set of eigenvalues of the $V^2\times V^2$ matrix $\mathbf{A}_f^2$. We also let $\{\mathbf{u}_k,\ k\in[V]\}$ and $\{\mathbf{v}_k,\ k\in[V^2]\}$ be the sets of eigenvectors of $\mathbf{A}_f$ and $\mathbf{A}_f^2$, respectively. Solving $\mathbf{A}_f\mathbf{u}=\lambda_k(\mathbf{A}_f)\mathbf{u}$

determines $\lambda_k(\mathbf{A}_f)$ for $k \in [V]$, associated with $\mathbf{u}$. Similarly, for the 2-fold OR product graph, the eigenvalues of $\mathbf{A}_f^2$ are determined by solving

$$\mathbf{A}_f^2\mathbf{v} = \nu(\mathbf{A}_f^2)\mathbf{v} \ , \tag{2.27}$$

where $\mathbf{v} = [\mathbf{v}_1^\intercal, \mathbf{v}_2^\intercal, \ldots, \mathbf{v}_V^\intercal]^\intercal$, and $\mathbf{v}_k$ is a $V \times 1$ vector, for each $k \in [V]$. In other words, for each $\nu(\mathbf{A}_f^2)$, we have a set of V block row equations, each containing V scalar equations. More specifically, $\nu(\mathbf{A}_f^2)$ in (2.27) satisfies the following V block equations:

$$\mathbf{A}_f\mathbf{v}_k + \mathbf{J}_V\mathbf{v}_{k+1} + \mathbf{J}_V\mathbf{v}_{k-1} = \nu(\mathbf{A}_f^2)\mathbf{v}_k \ , \quad k \in [V] \ . \tag{2.28}$$

Using (2.28), we next derive the eigenvalues of $\mathbf{A}_f^n$.

Theorem 2.1 (Distinct eigenvalues of C_i^n)**.** *The adjacency matrix $\mathbf{A}_f^n$ for C_i^n where $n \geq 2$, has the same distinct eigenvalues of $\mathbf{A}_f^{n-1}$ as well as two new distinct eigenvalues.*

Proof. See Appendix A.5. □

Next, in Section 2.3.4, we will examine the relation between the eigenvalues and the chromatic number of C_i^n to establish lower and upper bounds on $\chi(C_i^n)$.

2.3.4 Bounding the Chromatic Number of C_i Using the Eigenvalues of $\mathbf{A}_f$

Given a cycle C_i with an adjacency matrix $\mathbf{A}_f$, where we denote by $\vartheta(C_i)$ the set of its distinct eigenvalues, $\lambda_1(\mathbf{A}_f)$ its largest, and $\lambda_V(\mathbf{A}_f)$ its smallest eigenvalue [150]. Exploiting these, we can derive the following lower and upper bounds on $\chi(C_i)$ as follows [151, 152]:

$$1 - \frac{\lambda_1(\mathbf{A}_f)}{\lambda_V(\mathbf{A}_f)} \leq \chi(C_i) \leq \lfloor \lambda_1(\mathbf{A}_f) \rfloor + 1 \ , \tag{2.29}$$

where two lower bounds on λ_V have been derived in [153] and [154], respectively:

$$\lambda_V(\mathbf{A}_f) \geq -\sqrt{2E \cdot (\frac{(V-1)}{2})} \ , \tag{2.30}$$

$$\lambda_V(\mathbf{A}_f) \geq -\sqrt{\frac{V}{2} \cdot (\frac{(V+1)}{2})} \ . \tag{2.31}$$

We next apply the bound in (2.29) to $\chi(C_i^j)$ corresponding to $\mathbf{A}_f^j$, $j \in [n]$. To that end, let us consider an example where $i = 5$ and $j = 2$. Given C_5, the set of distinct eigenvalues of $\mathbf{A}_f$, using numerical evaluation, are given as $\vartheta(C_5) = \{-1.618, 0.618, 2\}$. Note also that $\vartheta(C_5^2) = \{-6.09, -1.61803,\ 0.61803, 5.09016, 12\}$ for $\mathbf{A}_f^2$. Consider the set $\vartheta(C_5^2)$, where $\lambda_{V^2}(\mathbf{A}_f^2) = -6.09$, for the 2-fold OR product graph C_5^2, an application of (2.30) yields $\lambda_{V^2}(\mathbf{A}_f^2) \geq -60$. Similarly, an application of (2.31) yields that $\lambda_{V^2}(\mathbf{A}_f^2) \geq -12.748$.

Hence, for the 2-fold OR product, one can find that the bound in [154] is tighter for cycles and their n-fold OR products.

We do not have an exact characterization for $\lambda_{V^n}(\mathbf{A}_f^n)$. However, using the lower bounds in (2.30) and (2.31) can help derive a lower bound on $\chi(C_i^n)$. On the other hand, for $\lambda_1(\mathbf{A}_f^n)$, recalling from Proposition 2.1 that all vertices of C_i^n for $n \geq 2$ have equal degrees, and exploiting this feature, we next derive an exact characterization of $\lambda_1(\mathbf{A}_f^n)$, for $n \geq 2$.

Proposition 2.6 (The largest eigenvalue for the adjacency matrix of C_i^n)**.** *The largest eigenvalue of C_i is $\lambda_1(\mathbf{A}_f) = 2$, and the largest eigenvalue of the n-fold OR product C_i^n is determined as*

$$\lambda_1(\mathbf{A}_f^n) = 2 + \sum_{j \in [n-1]} 2V^j \ , \quad n \geq 2 \ . \tag{2.32}$$

Proof. We prove it using Definition 2.1, (2.29), and the proof of Theorem 2.1—which leverages the fact that the eigenvalues of the sum of adjacency matrices equal the sum of their individual eigenvalues. The calculation of eigenvalues for the adjacency matrix of C_i^j is given in (2.28), which links the eigenvalues of $\mathbf{A}_f^j$ with the two $\mathbf{J}_V$ matrices of size $V^{j-1} \times V^{j-1}$ in each block row. From (A.12) and (A.13), the largest eigenvalue for a product graph C_i^n is obtained by adding $\lambda_1(\mathbf{A}_f^{n-1})$ for a sub-graph C_i^{n-1} and twice the largest eigenvalue of $\mathbf{J}_{V^{n-1}}$, which is V^{n-1} (see Lemma 2.1). Thus, $\lambda_1(\mathbf{A}_f) = 2$, and for $n \geq 2$, we achieve (2.32). □

Next, we present new bounds on $\chi(C_i^j)$ by exploiting (2.29), (2.30) and (2.31) which lower bound $\lambda_{V^j}(\mathbf{A}_f^j)$, and Proposition 2.6, which gives the exact value of $\lambda_1(\mathbf{A}_f^j)$.

Proposition 2.7 (Bounding $\chi(C_i^n)$ using eigenvalues of $\mathbf{A}_f^n$)**.** *The chromatic number of the n-fold OR product C_i^n of the cyclic graph C_i, i.e., $\chi(C_i^n)$ is lower and upper bounded as*

$$1 - \frac{2 + \sum_{j=1}^{n-1} 2V^j}{\max\left(-\sqrt{\frac{V^n}{2} \cdot \frac{(V^n+1)}{2}} \ , -\sqrt{2E^n \cdot \frac{(V^n-1)}{2}} \right)} \leq \chi(C_i^n) \leq \sum_{j=1}^{n-1} 2V^j \ + 3 \ . \tag{2.33}$$

Proof. Combining (2.29), which bounds λ_1 and λ_V of $\mathbf{A}_f$, with Proposition 2.6, as well as (2.31) that lower bounds λ_V [154], we have $1 - \frac{2+\sum_{j=1}^{n-1} 2V^j}{\lambda_{V^n}} \leq \chi(C_i^n) \leq \lfloor 2 + \sum_{j=1}^{n-1} 2V^j \rfloor + 1$. We further simplify $\lfloor \cdot \rfloor$, because λ_1 is a positive integer (see Proposition 2.6). Finally, substituting λ_{V^n} with the maximum of the lower bounds in (2.30) and (2.31) leads to (2.33). □

To complement Proposition 2.7, we next derive another bound that depicts the relation between the degree of each node in C_i^n and $\chi(C_i^n)$.

Corollary 2.2 (Bounding $\chi(C_i^n)$ using degrees of C_i^n)**.** *$\chi(C_i^n)$ satisfies the following relation:*

$$1+\frac{deg(x^n)}{\sqrt{2E^n-(V^n-1)\cdot(deg(x^n))+(1+\sum_{i=1}^{n-1}2V^i)\cdot(deg(x^n))}}\leq\chi(C_i^n)\leq\lfloor deg(x^n)\rfloor+1\ . \tag{2.34}$$

Proof. To characterize $deg(x^n)$, we apply (2.13) from Proposition 2.1, and to bound $\chi(C_i^n)$, we use (2.29). Then, we apply the lower bound on λ_{V^n} given in [155], which yields

$$\lambda_V(\mathbf{A}_f)\geq-\sqrt{2E-(V-1)\cdot(\min_{x\in[V]}deg(x))+(\min_{x\in[V]}deg(x)-1)\cdot(\max_{x\in[V]}deg(x))}\ , \tag{2.35}$$

giving the lower bound in (2.34). For the upper bound, we use (2.32), where $deg(x^n)=\lambda_1(\mathbf{A}_f^n)$. □

Next, we consider d-regular graphs, and characterize the vertex degrees and chromatic numbers for the n-fold OR products of d-regular graphs.

2.3.5 From Cycles to d-Regular Graphs

Building on our analysis of C_i^n in Sections 2.3.1-2.3.3, we now focus on d-regular source characteristic graphs. Next, we derive a closed-form expression for the degree of $G_{d,V}^n$.

Proposition 2.8 (Degrees of vertices in $G_{d,V}^n$)**.** *Given a d-regular graph $G_{d,V}=G(\mathcal{V},\mathcal{E})$, the degree of each vertex in the n-fold OR product, denoted by $G_{d,V}^n$, for $n\geq 2$, is expressed as:*

$$deg(x^n)=d\cdot\frac{V^n-1}{V-1}\ ,\quad\forall x^n\in[V^n]\ . \tag{2.36}$$

Proof. The proof follows from employing Definition 2.7 for $G_{d,V}$. For details, see Appendix A.6. □

Using (2.36), for the n-fold OR product of $G_{d,V}$ where V is even, we next determine $\chi(G_{d,V}^n)$.

Proposition 2.9 (The chromatic number of $G_{d,V}^n$)**.** *The chromatic number of the n-fold OR product of $G_{d,V}$ with an even number of vertices, i.e., $V=2k$, $k\in\mathbb{Z}^+$, is determined as:*

$$\chi(G_{d,V}^n)=d^n,\quad n\geq 1\ . \tag{2.37}$$

Proof. See Appendix A.7. □

For example, the 3-regular graph $G_{3,6}$ (see Figure 2.6) has $\chi(G_{3,6})=3$. For $n=2$, there are 6 sub-graphs, namely $\{G_{3,6}^2(l)\}_{l\in[6]}$, where $\chi(G_{3,6}^2(l))=3$ for all $l\in[6]$. Let us

choose a set of vertices in $G^2_{3,6}$ belonging to $\{G^2_{3,6}(5), G^2_{3,6}(6), G^2_{3,6}(1)\}$. We observe that this chosen subset is a complete graph, indicating that $\chi(G^2_{3,6}) \geq 9$. Reusing the same colors for the vertices of the remaining sub-graphs $(G^2_{3,6}(2), G^2_{3,6}(3), G^2_{3,6}(4))$, we deduce that $\chi(G^2_{3,6}) = 3^2 = 9$.

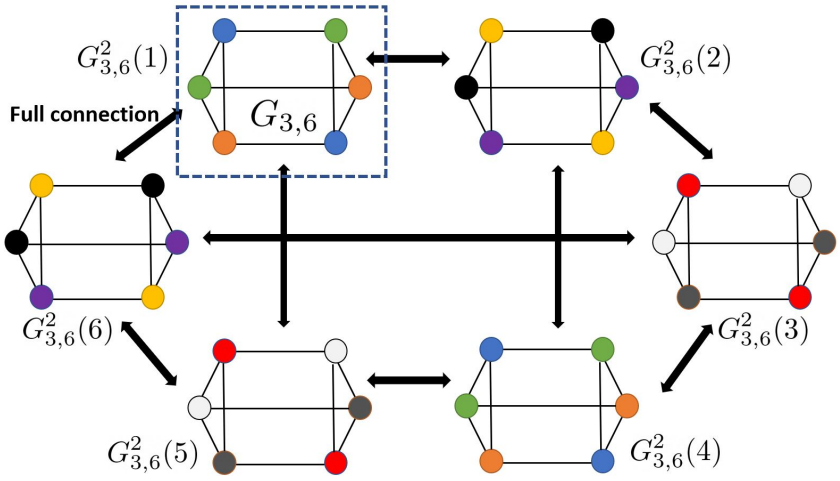

Figure 2.6: A 3-regular graph, $G_{3,6}$, is distinguished by a dashed square and $\chi(G^2_{3,6}) = 9$.

Next, given n-fold OR products of d-regular graphs, $G^n_{d,V}$, we examine their expansion properties and present bounds on their *expansion rates*.

2.3.6 d-Regular Graphs and Graph Expansion

Graph expansion quantifies how well-connected a graph is by measuring how easily subsets of vertices are connected to the remaining vertices of the graph. Graph expansion has applications in fields such as parallel computation [156, 157], complexity theory, and cryptography [158, 159] due to its strong vertex connectivity and robustness properties[9] [131]. In expander graphs, for any pair of distinct vertices $u, v \in [V]$, a path from u to v exists (see Definition 2.5). We next examine expansion rates for several classes of expander graphs, including $G_{d,V}$, C_i, and K_i.

Given $G_{d,V}$, the second largest eigenvalue of its adjacency matrix $\mathbf{A}_f$, namely $\lambda_2(\mathbf{A}_f)$, contributes to the linear expansion of $G_{d,V}$, where the number of edges grows linearly with the total number of vertices. Since the n-fold OR product of $G_{d,V}$ yields a $\deg(x^n)$-regular graph, it exhibits the connectivity properties of expander graphs.

Next, using [160], we establish lower and upper bounds on the expansion rate E_θ for $G^n_{d,V}$.

Proposition 2.10 (A lower bound on expansion rates of $G^n_{d,V}$)**.** *The expansion rate of*

[9]The ability to withstand node or edge removal.

$G^n_{d,V}$, given its adjacency matrix $\mathbf{A}^n_f$, is lower bounded as follows:

$$E_\theta(G^n_{d,V}) \geq \frac{(d \cdot \frac{V^n-1}{V-1})^2}{\Lambda^2(G^n_{d,V}) + \left((d \cdot \frac{V^n-1}{V-1})^2 - \Lambda^2(G^n_{d,V})\right) \cdot \frac{|Y|}{V^n}} \ , \tag{2.38}$$

where

$$\Lambda(G^n_{d,V}) = \max\left(\lambda_2(\mathbf{A}^n_f),\ |\lambda_{V^n}(\mathbf{A}^n_f)|\right) \ , \tag{2.39}$$

with $\lambda_2(\mathbf{A}^n_f)$ and $\lambda_{V^n}(\mathbf{A}^n_f)$ being the second and the smallest eigenvalues of $\mathbf{A}^n_f$, respectively.

Proof. As noted in Section 2.2, for a given $G_{d,V}$ with $\mathbf{A}_f$, $\lambda_1(\mathbf{A}_f) = d$, and the eigenvalues of $\mathbf{A}^n_f$ for the n-fold OR product of any connected $G^n_{d,V}$ are ordered as $\{d \geq \lambda_1 > \lambda_2 > \cdots > \lambda_{V^n}\}$. For any subset Y of $G_{d,V}$, [160] shows that the size of $|N_G(Y)|$ satisfies the following:

$$|N_G(Y)| \geq \frac{d^2 \cdot |Y|}{\Lambda^2(G) + (d^2 - \Lambda^2(G)) \cdot \frac{|Y|}{V}} \ . \tag{2.40}$$

In (2.40), $\Lambda(G) = \max(\lambda_2, |\lambda_V|) \leq d$, equality holds if and only if G is bipartite, or is disconnected (composed of singleton vertices) [131]. Dividing both sides of (2.40) by $|Y|$, letting $E_\theta(G^n) = |N_{G^n}(Y)|/|Y|$, and taking in the total number of vertices, V^n, in (2.40) yields (2.38). □

In Proposition 2.10, given a graph $G_{d,V}$, $\Lambda(G_{d,V})$ captures its connectivity and structural balance. We infer that as $|Y|$ increases, the expansion rate is small, indicating less connectivity between vertices. Given a connected $G_{d,V}$, as evident from (2.4), the maximum expansion rate, denoted by E^{ub}_θ, is achieved when $G^n_{d,V}$ is a complete graph, K^n_i, where $i \in [V^n]$, as follows:

$$E^{ub}_\theta \geq \frac{(V^n-1)^2}{1 + ((V^n-1)^2 - 1) \cdot \frac{|Y|}{V^n}} \ . \tag{2.41}$$

The minimum expansion rate, denoted by E^{lb}_θ, is achieved when $G^n_{d,V} = C^n_i$ because C^n_i has the minimum number of edges to ensure a Hamiltonian path for each $x \in [V]$[10]. This yields

$$E^{lb}_\theta \geq \frac{(2 \cdot \frac{V^n-1}{V-1})^2}{\lambda^2_{V^n}(C^n_i) + ((2 \cdot \frac{V^n-1}{V-1})^2 - \lambda^2_{V^n}(C^n_i)) \cdot \frac{|Y|}{V^n}} \ . \tag{2.42}$$

We next examine the relationship between $\lambda_1(\mathbf{A}_f)$ and $\lambda_2(\mathbf{A}_f)$ for sub-graphs of an expander graph, such as $G_{d,V}$. To this end, we apply [130, Lemma 3], which states that for

[10] Note that cycles are not considered as an expander graph, but their OR product can capture expansion properties.

any subset of vertices, $Y \subseteq [V]$ in $G_{d,V}$, the induced sub-graph[11] S satisfies the following property (see, [130, Lemma 3]):

$$\lambda_1(S) \leq \lambda_2(\mathbf{A}_f) + (d - \lambda_2(\mathbf{A}_f)) \cdot (\frac{|Y|}{V}) \ . \tag{2.43}$$

For a connected 2-regular graph (C_i), adapting (2.43) and selecting vertex sets S and Y such that $Y \cup S = C_i^n$, we establish a simplified relation between $\lambda_1(\mathbf{A}_f^{n-1})$ and $\lambda_2(\mathbf{A}_f^n)$, as follows.

Corollary 2.3 (The spectral relation between $\lambda_1(\mathbf{A}_f^{n-1})$ and $\lambda_2(\mathbf{A}_f^n)$ in 2-regular graphs)**.** *The relationship between $\lambda_2(\mathbf{A}_f^n)$ of C_i^n and $\lambda_1(\mathbf{A}_f^{n-1})$ of C_i^{n-1} is detailed as follows:*

$$\lambda_1(\mathbf{A}_f^{n-1}) \leq \lambda_2(\mathbf{A}_f^n) + \Big(2 \cdot \frac{V^n - 1}{V - 1} - \lambda_2(\mathbf{A}_f^n)\Big) \cdot \frac{1}{V} \ . \tag{2.44}$$

Proof. Given the relation between $\lambda_1(S)$ and $\lambda_2(\mathbf{A}_f)$ for $G_{d,V}$ in (2.43), we consider that $G_{d,V}^n = C_i^n$, and we choose sub-graphs Y and S accordingly, as sub-graphs of C_i^n, where

$$Y \subseteq \bigcup_{t \neq l,\ t \in [V]} C_i^n(t) = \{C_i^n(1), C_i^n(2), \ldots, C_i^n(V)\} \setminus \{C_i^n(l)\} \ . \tag{2.45}$$

Using (2.43), by selecting $S \subset \{C_i^n(1), C_i^n(2), \ldots, C_i^n(V)\}$, namely $C_i^n(l)$, for the LHS of (2.43), in which each sub-graph is isomorphic to C_i^{n-1}, i.e., $S = \big(\bigcup_t C_i^n(t)\big)^{\mathsf{c}}$. Consequently, using (2.45), S, and substituting them in (2.43), we reach the statement in (2.44). □

Corollary 2.3 investigates how the largest eigenvalue of the $(n-1)$-fold sub-graph of a cycle is upper bounded in terms of the second-largest eigenvalue of its n-fold OR product, offering insight into spectral behavior across recursive graph products.

2.3.7 Graph Expansion and Graph Diameter

In this section, due to the connection between graphs' diameter and spectral properties (more specifically, the gap between $\lambda_1(\cdot)$ and $\lambda_2(\cdot)$) as well as graph connectivity, we explore the diameter of a d-regular graph $G_{d,V}$. To this end, we provide an upper bound on the diameter of $G_{d,V}^n$ exploiting [161] and Definition 2.6.

Corollary 2.4 (Graph diameter [161])**.** *Given a d-regular graph $G_{d,V}(\mathcal{V}, \mathcal{E})$, and $\mathbf{A}_f$, the diameter of $G_{d,V}$ is upper bounded as*

$$Dia(G_{d,V}) \leq \left\lceil \frac{\log(V-1)}{\log\left(\frac{\lambda_1(\mathbf{A}_f)}{\lambda_2(\mathbf{A}_f)}\right)} \right\rceil \ , \tag{2.46}$$

[11] An induced sub-graph is formed by selecting a subset of vertices $S \subseteq [V]$ includes all edges in $\mathcal{E}$ whose endpoints lie in S.

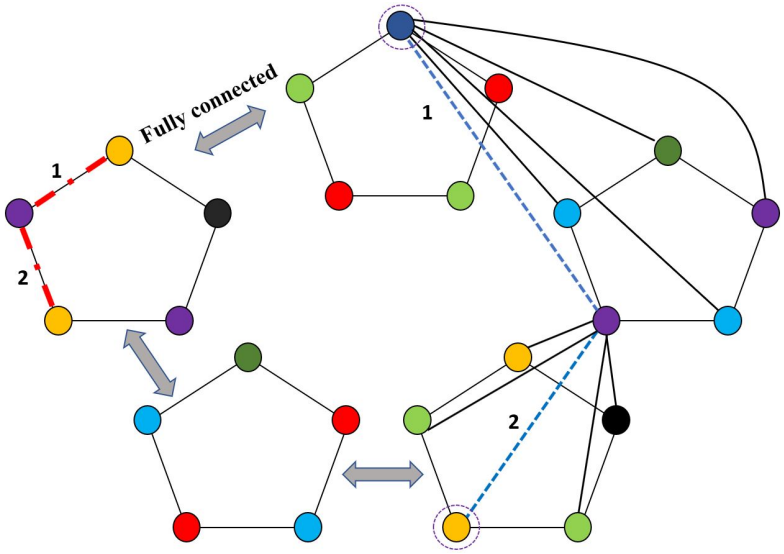

Figure 2.7: The diameter of a C_5 is depicted with a dash-dotted red line, and the diameter of C_5^2 is depicted in a blue dashed line.

where $\lambda_1(\mathbf{A}_f)$ *and* $\lambda_2(\mathbf{A}_f)$ *denote the largest and the second largest eigenvalues of the adjacency matrix* $\mathbf{A}_f$*, respectively.*

Calculating the diameter of a graph G devised for a function is not directly relevant in evaluating $\chi(G)$. For instance, in a complete graph, the chromatic number equals the number of vertices, whereas the diameter is 1. However, using (2.46), we emphasize the effect of the ratio of eigenvalues $(\frac{\lambda_1(\cdot)}{\lambda_2(\cdot)})$ on the diameter, which can depict the function's complexity. Next, we discuss the diameter of the n-fold OR product of d-regular graph $G_{d,V}^n$.

Proposition 2.11 (Diameter of OR product graphs)**.** *The diameter of the n-fold OR product graph of a d-regular graph is equal to the diameter of the first product graph*

$$Dia(G_{d,V}) = Dia(G_{d,V}^n) \ . \tag{2.47}$$

Proof. See Appendix A.8 □

Using (2.46) and (2.47), we observe that the OR power of $G_{d,V}$ does not affect the complexity of functional compression using $G_{d,V}^n$.

Next, we investigate general functions with arbitrary characteristic graphs. Building on Section 2.3, we present bounds on the chromatic number of their n-fold OR products.

2.4 Bounds For General Characteristic Graphs

Here, we focus on general characteristic graphs $G(\mathcal{V}, \mathcal{E})$, their n-fold OR products, G^n. Given a graph G with an adjacency matrix $\mathbf{A}_f$, we first evaluate $deg(x_k)$ for $x_k \in [V]$, and derive bounds on $\lambda_1(\mathbf{A}_f)$ and $\chi(G)$ in Section 2.4.1. We also present bounds on

expansion rates and establish lower and upper bounds on entropies of G^n in Sections 2.4.2 and 2.4.3, respectively. Finally, for the n-fold OR product G^n, we introduce an approach that decomposes $\mathbf{A}_f^n$ to two symmetric block matrices and leverages GCT to investigate the spectrum of G^n in Section 2.4.4.

2.4.1 Degrees and Chromatic Numbers of General Graphs

Given a general graph G, we next derive a recursive relation for $deg(x_k^n)$ for $x_k^n \in [V^n]$, where $deg(x_k^n)$ may vary across vertices, providing a generalization of Propositions 2.1 and 2.8.

Corollary 2.5 (Degrees of vertices in G^n)**.** *Given a general graph $G(\mathcal{V}, \mathcal{E})$, the degrees of vertices of G^n are calculated as follows:*

$$deg(x_k^n) = deg(x_k) + \sum_{j=1}^{n-1} deg(x_k) \cdot V^j \ , \quad \forall x_k \in [V^n] \ . \tag{2.48}$$

Proof. Similarly to Propositions 2.1 and 2.8, we can compute $deg(x_k)$ for $x_k \in [V]$, with the distinction that each x_k may have a different degree. For the 2-fold OR product, each vertex, $x_k^2(l)$ for $l \in V$, connects to $deg(x_k)$ adjacent sub-graphs. Since neighboring nodes differ across x_k, we iteratively compute the degrees of x_k^n separately for each x_k. □

Corollary 2.5 immediately implies that if vertices x_t, $x_k \in [V]$ for $t \neq k$ have equal degrees in G^1, i.e., $deg(x_t) = deg(x_k)$, then $deg(x_t^n) = deg(x_k^n)$ in the n-fold OR product, G^n, $n \geq 2$.

Given a general graph G, we next devise lower and upper bounds on $\chi(G^n)$. To that end, we exploit the block matrix representation of $\mathbf{A}_f^n$ (see (2.26)) and use the maximum number of sub-matrices $\mathbf{J}_{V^{n-1}}$ in the rows of $\mathbf{A}_f^n$.

Corollary 2.6 (Bounds on $\chi(G^n)$ for a general G)**.** *Given a general characteristic graph $G(\mathcal{V}, \mathcal{E})$, the chromatic number of G^n, $\chi(G^n)$, is bounded as follows:*

$$1 - \frac{\lambda_1(\mathbf{A}_f) + d_{\max} \cdot \sum_{j=1}^{n-1} V^j}{\lambda_{V^n}(\mathbf{A}_f^n)} \leq \chi(G^n) \leq \left\lfloor \lambda_1(\mathbf{A}_f) + d_{\max} \cdot \sum_{j=1}^{n-1} V^j \right\rfloor \ , \tag{2.49}$$

Proof. The non-sparsity of an adjacency matrix, i.e., more 1s in its entries, is directly related to its largest eigenvalue. The full connection between sub-graphs in the OR product is represented by $\mathbf{J}_V$, with $\lambda_1(\mathbf{J}_V) = V$. Thus, the row with the most $\mathbf{J}_V$ matrices provides an upper bound for the largest eigenvalue. Additionally, the average number of $\mathbf{J}_{V^{n-1}}$ matrices across block rows of $\mathbf{A}_f^n$, multiplied by V^{n-1} (which represents $\lambda_1(\mathbf{J}_{V^{n-1}})$), approximates $\lambda_1(\mathbf{A}_f^n)$. By modifying (2.29) and using $d_{\max}$ of G to approximate $\lambda_1(\mathbf{A}_f^n)$, we can establish a bound for $\chi(G^n)$.

□

Corollary 2.6 refines the bounds in (2.29), Proposition 2.7 and Corollary 2.2 by leveraging the exact value of $\lambda_1(\mathbf{A}_f^n)$ and accounting for the specific structure of G. Given Corollary 2.6, let us investigate the computational complexity for determining the eigenvalues $\lambda_k(\mathbf{A}_f^n)$ and contrast it with the QR method[12] for calculating $\lambda_k(\mathbf{A}_f^n)$, with a computation complexity of $O(V^{3n})$. However, using Corollary 2.6, the overall complexity remains at $O(V^3)$. This is because computing the eigenvalues of G has a complexity of $O(V^3)$; the summations on the RHS and LHS of (2.49) each have a complexity of $O(n)$, and the maximization step (determining $d_{\max}$) has a complexity of $O(V)$. Hence, the dominant term, i.e., $O(V^3)$, dominates the final complexity.

Next, given a general $G(\mathcal{V}, \mathcal{E})$ with an adjacency matrix $\mathbf{A}_f$, we derive lower and upper bounds on $\lambda_1(\mathbf{A}_f^n)$ using Lemma 2.1, where the lower and upper bounds are functions of the minimum and maximum values of $deg(x_k)$ for $x_k \in [V]$.

Corollary 2.7 (Bounds on $\lambda_1(\mathbf{A}_f^n)$ for a general G)**.** *Given a general graph G, the largest eigenvalue for the adjacency matrix of G^n, denoted by $\lambda_1(\mathbf{A}_f^n)$, is bounded as follows:*

$$\min_{k\in[V]}(deg(x_k)) \cdot \lambda_1(\mathbf{J}_{V^{n-1}}) \leq \lambda_1(\mathbf{A}_f^n) \leq \max_{k\in[V]}(deg(x_k)) \cdot \lambda_1(\mathbf{J}_{V^{n-1}}) \ , \tag{2.50}$$

where we infer that

$$\lambda_1(\mathbf{A}_f^n) \approx deg_{\text{avg}}(x_k) \cdot \lambda_1(\mathbf{J}_{V^{n-1}}) \ . \tag{2.51}$$

Corollary 2.7 illustrates how $deg(x_k)$ determines the spectral properties of G^n, thus the achievable rate in distributed compression.

2.4.2 Bounds on Expansion Rates of General Graphs

Here, given a general graph G, we investigate the expansion of G^n. We exploit (2.41), derived from the characteristics of the n-fold OR products of complete graphs, to obtain the upper bound, E_θ^{ub}, and (2.42), derived from the n-fold OR products of cycles, to obtain the lower bound, i.e., E_θ^{lb}. We next establish lower and upper bounds on $E_\theta(G^n)$.

Corollary 2.8 (Bounds on $E_\theta(G^n)$)**.** *The expansion rate for the n-fold OR product of a general characteristic graph $G(\mathcal{V}, \mathcal{E})$ is lower and upper bounded as follows:*

$$E_\theta^{lb} \leq E_\theta(G^n) \leq E_\theta^{ub} \ , \tag{2.52}$$

where E_θ^{ub}, derived from K_i^n (representing a fully connected characteristic graph), and E_θ^{lb}, from C_i^n (representing a connected graph with the minimum number of edges), for $i = V^n$.

[12]The QR transformation iteratively decomposes a matrix $\mathbf{A}_{(t)}$ where $t \in \mathbb{Z}^+$ denotes the iteration index, into an orthogonal matrix $\mathbf{Q}_{(t)}$ and an upper triangular matrix $\mathbf{R}_{(t)}$, satisfying $\mathbf{A}_{(t)} = \mathbf{Q}_{(t)}\mathbf{R}_{(t)}$. Then, the next iteration is given by $\mathbf{A}_{(t+1)} = \mathbf{R}_{(t)}\mathbf{Q}_{(t)}$. Under typical conditions (e.g., $\mathbf{A}$ is diagonalizable with distinct eigenvalues), it converges to an upper triangular matrix whose diagonal approximates $\lambda_k(\mathbf{A})$. For an $m \times m$ matrix, QR requires a computational complexity of $O(m^3)$ [162].

Proof. See Appendix A.9. □

Recall that the lower bound for $E_\theta(G^n_{d,V})$ in (2.38) is given in terms of $\Lambda(G^n) = \max(\lambda_2(\mathbf{A}^n_f), |\lambda_{V^n}(\mathbf{A}^n_f)|)$. Whereas in Corollary 2.8, we use exact values of $\Lambda(K^n_i)$ and $\Lambda(C^n_i)$ for upper and lower bounds, respectively, with $\mathbf{A}^n_f$ denoting each graph's adjacency matrix. Given G^n, $E_\theta(G^n)$ reflects its connectivity, with higher values leading to limited savings in source compression.

2.4.3 Bounds on Entropies of General Graphs

Here, we derive upper and lower bounds on the graph entropies for general characteristic graphs. For the upper bound, we use a similar achievability approach as in the case of C^n_i (see Proposition 2.5), which relies on coloring the MISs of sub-graphs of G^n, i.e., G^j for $j \in [n]$. For the lower bound, we employ fractional coloring applied to the n-fold OR products of general graphs to establish a bound on $H_G(X_1)$. We next derive an upper bound on $H_G(X_1)$.

Proposition 2.12 (An upper bound on $H_G(X_1)$)**.** *Given a characteristic graph $G(\mathcal{V}, \mathcal{E})$, the entropy of G^n is upper bounded as follows:*

$$H_G(X_1) \leq \frac{1}{n} H\Big(\zeta_n \cdot \frac{\alpha(G^n)}{V^n}, \zeta_{n-1} \cdot \frac{\alpha(G^{n-1})}{V^n}, \ldots, \zeta_0 \cdot \frac{1}{V^n}\Big) \ , \tag{2.53}$$

where ζ_t , $t \in [n] \cup \{0\}$, represents the number of maximum independent sets of G^t with a size of $\alpha(G^t)$.

Proof. See Appendix A.10. □

Despite the upper bound in Proposition 2.12, to the best of our knowledge, with *traditional coloring schemes for $G(\mathcal{V}, \mathcal{E})$* where the total number of vertices is odd, i.e., $V = 2k+1$ for $k \in \mathbb{Z}^{\geq 2}$, there is no established lower bound for $H_G(X_1)$. To that end, in Corollary 2.9, we derive a lower bound on $H_G(X_1)$ by employing the concept of *fractional coloring* (see Definition 2.12).

Corollary 2.9 (A lower bound on $H_G(X_1)$)**.** *The entropy of $H_G(X_1)$ for a general connected characteristic graph $G(\mathcal{V}, \mathcal{E})$ with $V = 2k+1$ and Hamiltonian path, where $k \in \mathbb{Z}^{\geq 2}$, and under uniform distribution of X_1, is lower bounded by*

$$\log_2\left(\frac{2k+1}{k}\right) \leq H_G(X_1) \ . \tag{2.54}$$

Proof. See Appendix A.11. □

From Corollary 2.9, we infer that employing fractional coloring, and (2.7), yields a lower bound on $H_G(X_1)$. For $k = 2$, the lower bound using C_5 for the graph $G(\mathcal{V}, \mathcal{E})$ with $V = 5$ is given by $1.32 \leq H_G(X_1)$, which matches the Shannon capacity of the pentagon [100].

2.4.4 Spectra of General Graphs

Given a general graph G, we analyze the spectrum of $\mathbf{A}_f$ using the concept of GCT, as detailed in Definition 2.13. We then exploit this spectrum to establish bounds on $\chi(G^n)$. For G^n, using the symmetry of $\mathbf{A}_f^n = (\mathbf{A}_f^{n-1} \otimes \mathbf{I}_V + \mathbf{J}_V \otimes \mathbf{A}_f^{n-1}) \in \mathbb{F}_2^{V^n \times V^n}$, where $\otimes$ denotes the Kronecker product, we infer that circle D_k, in which an eigenvalue $\lambda_k(\mathbf{A}_f^n)$ is contained, simplifies to an interval:

$$\delta_k = \{\lambda_k(\mathbf{A}_f^n) \in \mathbb{R} : \left|\lambda_k(\mathbf{A}_f^n) - a_{kk}^n\right| \leq \sum_{t \neq k} |a_{kt}^n|\}, \quad k, t \in [V^n] \ , \tag{2.55}$$

where a_{kk}^n and $\sum_{t:\, t\neq k}^{V^n} |a_{kt}^n|$ denote the diagonal elements and the sums of off-diagonal elements in the k-th row of $\mathbf{A}_f^n$, respectively, then for $k = Vk' + i'$, and $t = Vt' + j'$, the elements are defined as:

$$a_{kt}^n = \left(a_{k't'}^{n-1} \cdot \Delta_{i'j'} + a_{i'j'}^{n-1}\right) \bmod 2 \ , \tag{2.56}$$

where $\Delta_{i'j'}$ denotes the Kronecker delta function (i.e., $\Delta_{i'j'} = 1$ if $i' = j'$, and $\Delta_{i'j'} = 0$ otherwise). The index mappings $k = Vk' + i'$, $t = Vt' + j'$ reflect the structure of the Kronecker product, where $k', t' \in [V^{n-1}]$ index the block position and $i', j' \in [V]$ index the position within each block. The Kronecker delta $\Delta_{i'j'}$ ensures that $a_{k't'}^{n-1}$ contributes only when $i' = j'$, capturing the effect of $\mathbf{A}_f^{n-1} \otimes \mathbf{I}_V$, while $a_{i'j'}^{n-1}$ comes from $\mathbf{J}_V \otimes \mathbf{A}_f^{n-1}$.

Using the recursive definition in (2.56), the entries of $\mathbf{A}_f^n$ can be expressed directly in terms of $\mathbf{A}_f$ as

$$a_{kt}^n = \sum_{m=0}^{n-1} \Big(\Big[\prod_{\substack{r=0 \\ r\neq m}}^{n-1} \Delta_{k_r t_r} \Big] \cdot a_{k_m t_m} \Big) \bmod 2 \ . \tag{2.57}$$

Here, the product $\prod_{r\neq m} \Delta_{k_r t_r}$ enforces equality across all index pairs (k_r, t_r) except at position m, capturing the block dependencies induced by the Kronecker construction. The term $a_{k_m t_m}$ corresponds to the entry of $\mathbf{A}_f$ at the m-th coordinate, while the summation over $m \in [n]$ aggregates contributions from all coordinate directions. This recursive formulation shows that the n-fold adjacency matrix $\mathbf{A}_f^n$ is built systematically from $\mathbf{A}_f$ through coordinate-wise Kronecker composition, thereby linking the local adjacency pattern of G to the global spectral behavior of its n-fold OR product.

We next refine δ_k by exploiting the concept of block GCT, where using Definition 2.14 helps enclose each given $\lambda_k(\mathbf{A}_f^n)$ within an interval denoted by δ_k^b. Interval k is characterized by the diagonal sub-matrices $\mathbf{A}_{kk}$, corresponding to $\mathbf{A}_f^{n-1}$, and the off-diagonal sub-matrices $\mathbf{A}_{kt}$, consisting of $\mathbf{Z}_{V^{n-1}}$ and $\mathbf{J}_{V^{n-1}}$, which represent disconnected and connected components of G^n, respectively. However, the block GCT representation leads to loose bounds on $\lambda_k(\mathbf{A}_f^n)$ and subsequently on $\chi(G^n)$ via (2.29). To tighten these bounds, in Theorem 2.2 and Corollary 2.10, we derive $\lambda_k(\mathbf{A}_f^n)$ by leveraging GCT intervals from

(2.8) and (2.12), and using a decomposition-based approach that splits $\mathbf{A}_f^n$ into two symmetric matrices.

Theorem 2.2 (Computing $\lambda_k(\mathbf{A}_f^n)$ via splitting $\mathbf{A}_f^n$ into two symmetric matrices)**.** *The eigenvalues of $\mathbf{A}_f^n$, denoted by $\lambda_k(\mathbf{A}_f^n)$, are given as follows:*

$$\lambda_k(\mathbf{A}_f^n) = \lambda_k(\mathbf{A}_{Gr}^n) + \lambda_k(\mathbf{A}_{f^c}^n) \ , \quad k \in [V^n], \tag{2.58}$$

where $\mathbf{A}_{Gr}^n$ is a block diagonal matrix with diagonal blocks formed from $\mathbf{A}_f^{n-1}$, and $\mathbf{A}_{f^c}^n = \mathbf{A}_f^n - \mathbf{A}_{Gr}^n$ captures the off-diagonal elements of $\mathbf{A}_{Gr}^n$[13] *.*

Proof. See Appendix A.12. □

Theorem 2.2 demonstrates that by decomposing $\mathbf{A}_f^n$ into $\mathbf{A}_{Gr}^n$ and $\mathbf{A}_{f^c}^n$, we can capture the connections between $\mathbf{A}_f^{n-1}(l)$, $l \in [V]$, corresponding to the sub-graphs of G^n.

Next, we describe an iterative technique to determine $\lambda_k(\mathbf{A}_f^n)$ from $\lambda_k(\mathbf{A}_f)$.

Corollary 2.10. *The eigenvalues of $\mathbf{A}_f^n$ can be iteratively calculated as follows:*

$$\lambda_k(\mathbf{A}_f^n) = \lambda_k(\mathbf{A}_f) + \sum_{j=2}^{n} \lambda_k(\mathbf{A}_{f^c}^j) \ , \quad k \in [V^n] \ . \tag{2.59}$$

Proof. See Appendix A.13. □

In Theorem 2.2 and Corollary 2.10, we demonstrate that exploiting the computation structure through the block representation for $\mathbf{A}_f^n$ helps us achieve a lower complexity compared to [164]. Next in Proposition 2.13, we use Theorem 2.2 to provide a tighter bound for $\chi(G^n)$ compared to the intervals derived from the block GCT representation (see (2.12)).

Proposition 2.13 (Bounds on $\chi(G^n)$ using GCT)**.** *Given a general graph $G(\mathcal{V}, \mathcal{E})$, the chromatic number $\chi(G^n)$ is bounded as follows:*

$$1 - \frac{\lambda_1(\mathbf{A}_{Gr}^n) + \lambda_1(\mathbf{A}_{f^c}^n)}{-\sqrt{(V^n/2) \cdot [(V^n+1)/2]}} \leq \chi(G^n) \leq \lfloor \lambda_1(\mathbf{A}_{Gr}^n) + \lambda_1(\mathbf{A}_{f^c}^n) \rfloor + 1 \ . \tag{2.60}$$

Proof. To prove this result, we use the bounds for $\chi(G^n)$ in (2.29) using the eigenvalues of $\mathbf{A}_f^n$, adjust $\lambda_1(\mathbf{A}_f^n)$ and $\lambda_{V^n}(\mathbf{A}_f^n)$, and by utilizing (2.58) from Theorem 2.2, and (2.31), respectively. □

For the RHS and LHS in (2.60), (2.51) provides a tighter approximation on $\lambda_1(\mathbf{A}_f^n)$ compared to (2.50), which stems from employing [165, Lemma 5], and

$$deg_{\text{avg}} \cdot V^{n-1} \leq \left| \frac{\delta^b(\lambda_1(\mathbf{A}_f^n))}{2} \right| \ .$$

[13] Since $\mathbf{A}_f^n$ are real symmetric, they are diagonalizable. The additive decomposition in (2.58) holds if $\mathbf{A}_{Gr}^n$ and $\mathbf{A}_{f^c}^n$ commute (i.e., admit a common eigen-basis). Without this condition, one may instead apply Weyl's inequalities to bound the eigenvalues; see, e.g., [163] for details.

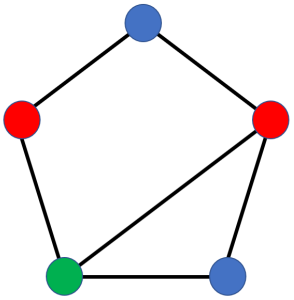

Figure 2.8: Characteristic graph G_{f_1} for Example 2.5.

We next illustrate the utility of Theorem 2.2 and Corollary 2.10 via an example.

Example 2.5. *Consider a characteristic graph G_1 (see Figure 2.8) with an adjacency matrix*

$$\mathbf{A}_{f_1} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}, \tag{2.61}$$

with a set of distinct eigenvalues $\vartheta(G_1) = \{2.4812, 0.6889, 0, -1.1701, -2\}$. Using GCT, we derive five intervals $\{\delta_k\}_{k\in[5]} = \{[-2,2],\ [-2,2],\ [-2,2],\ [-3,3], [-3,3]\}$ for $\mathbf{A}_{f_1}$, one for each eigenvalue, where each δ_k is centered at 0 since trace$(\mathbf{A}_{f_1}^2) = 0$. Two unique intervals with the largest lengths are $\delta_1 = [-2,2]$ and $\delta_2 = [-3,3]$, which are used to determine $\lambda_1(\mathbf{A}_{f_1})$.

$\mathbf{A}_{f_1}^2 =$

$\mathbf{A}_f$	J	Z	Z	J
J	$\mathbf{A}_f$	J	Z	Z
Z	J	$\mathbf{A}_f$	J	Z
Z	Z	J	$\mathbf{A}_f$	J
J	Z	Z	J	$\mathbf{A}_f$

Figure 2.9: The adjacency matrix of 2-fold product graph for (2.61)

From Theorem 2.2 and Corollary 2.10, we have $\lambda_1(\mathbf{A}_{f_1}^2) \in [12, 18]$. Applying the GCT for block matrices (see (2.12) in Section 2.2.3), we obtain $\sigma(\mathbf{A}_{f_1}^2) \in \cup_{k=1}^{5} \delta_k^b$, where $\delta^b = [-18, 18]$ includes all possible eigenvalues but provides a less precise estimate than (2.58). Refining the bounds for $\lambda_1(\mathbf{A}_{f_1}^2)$ using the $\deg_{\text{avg}}(x_k)$ in (2.50) and the upper

bound in (2.60) gives a more precise interval of $[12, 15]$. *This range shows the upper bound is* 3 *units tighter than the maximum degree method.*

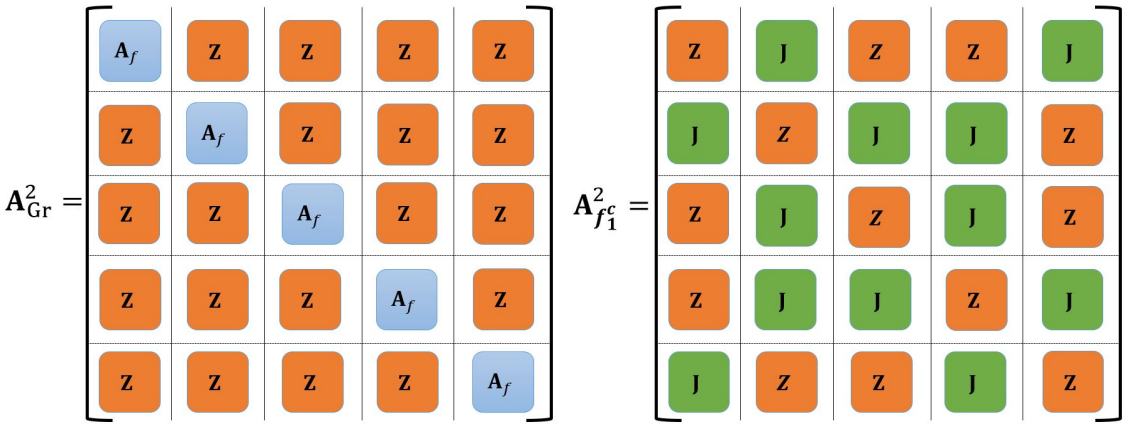

Figure 2.10: Splitting the adjacency matrix $\mathbf{A}^2_{f_1}$ into two symmetric matrices $\mathbf{A}^2_{Gr}$ and $\mathbf{A}^2_{f_1^c}$, where $\mathbf{A}^2_{f_1} = \mathbf{A}^2_{Gr} + \mathbf{A}^2_{f_1^c}$.

2.5 Conclusion

In this chapter, we addressed the problem of distributed functional compression by introducing novel coloring-based encoding schemes for source characteristic graphs. We analyzed various graph topologies — cycles (C_i), d-regular graphs ($G_{d,V}$), and general graphs (G) — and their n-fold OR product realizations (C_i^n, $G_{d,V}^n$, and G^n), exploring the interplay between adjacency matrix eigenvalues and chromatic numbers to develop low-entropy colorings schemes and derive bounds on the compression rate for asymptotically lossless function compression.

For cycles, we derived bounds on the degrees of C_i^n and proposed a recursive coloring scheme for C_{2k+1}^n, which computes valid colorings in polynomial time by leveraging their structural properties. We also investigated the relationship between the spectra of C_i^n and the chromatic numbers $\chi(C_i^n)$ to establish bounds on the minimum entropy colorings.

For d-regular graphs, we analyzed the degrees and eigenvalues for $G_{d,V}^n$. We also investigated the connection between the OR products of $G_{d,V}$ and graph expansion described by the spectral properties of $G_{d,V}$. This enabled us to derive upper and lower bounds on the expansion rates of general characteristic graphs, where the expansion rate reflects the connectivity of sub-graphs, where higher connectivity implies an increase in $H^{\chi}_{G^n}(\mathbf{X}_1)$.

For general characteristic graph topologies G, we derived bounds on the degrees and eigenvalues of G^n using the block matrix representation of the adjacency matrix of G^n, denoted by $\mathbf{A}_f^n$, in order to achieve a reduced computational complexity for deriving the eigenvalues $\lambda_k(\mathbf{A}_f^n)$ for $k \in [V^n]$ compared to the QR method [162, 166]. By leveraging the GCT approach, we provided upper and lower bounds on $\lambda(\mathbf{A}_f^n)$ and compared these with the iterative $\mathbf{A}_f^n$ decomposition method (see Theorem 2.2 and Corollary 2.10), which

exploits the properties of eigenvalues of symmetric matrices to produce tighter bounds for $\lambda(\mathbf{A}_f^n)$.

In conclusion, our results present a unified framework for distributed functional compression linking graph-theoretic structures with information-theoretic bounds. By relating spectral properties to achievable rates, our study offers a coloring-based approach for designing functional compression schemes. We believe that these insights impact both theoretical developments and practical implementations of low-overhead, structure-adaptive coding methods.

Chapter 3

Multi-Server Multi-Function Distributed Computation

This chapter[1] addresses the communication cost in multi-server multi-function distributed computation, considering a broad class of functions and data statistics. We establish upper bounds on the communication cost when a user requests multiple, possibly non-linear, tasks from distributed servers under different data placements. By leveraging structural properties of both the data and the functions, our framework provides general insights into the efficiency of multi-server systems and demonstrates gains under specific placements and function classes.

3.1 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 [167], data-intensive analyses for cloud computing [168], and machine learning [169], as well as applications in various other fields such as computational fluid dynamics [170], high-quality graphics for movie and game rendering [171], and a variety of medical applications [172], to name just a few. In the center of this ever-increasing presence of parallelized computing stands modern parallel processing techniques, such as MapReduce [22, 173, 174] and Spark [23, 175].

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 [49, 176–178]

[1]The results in this chapter are published prior in [110, 111].

in various topologies [179–181]. Central to the efforts to reduce communication costs stand coding techniques such as those found in [8, 55–57, 101, 182–194], including gradient coding [101] and different variants of coded distributed computing that nicely yield gains in reliability, scalability, computation speed, and cost-effectiveness [186]. Similar communication-load aspects are often addressed via polynomial codes [7], which can mitigate stragglers and enhance the recovery threshold, while MatDot codes, devised in [51, 56] 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 [43, 51, 52, 55–58, 195–199], which again, focus on distributed matrix multiplication. For example, focusing on a cyclic dataset placement model, the work in [43] provided useful achievability results, while the authors of [57] have characterized achievability and converse bounds for secure distributed matrix multiplication. Furthermore, the work in [55] found creative methods to exploit the correlation between the entries of the matrix product in order to reduce the cost of communication.

3.1.1 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 sub-functions for the datasets. This multi-server multi-function setting was recently studied by Wan *et al.* in [43] for the class of linearly separable functions, which nicely captures a wide range of real-world tasks [22] such as convolution [196], the discrete Fourier transform [200], 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 a growing need to consider more general classes of functions, including non-linear functions, such as is often the case with sub-functions that produce intermediate values in MapReduce operations [22] or that relate to quantization [201], classification [202], and optimization [54]. 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 [48, 50, 52, 57, 197, 203], with a diverse focus that entails secrecy [48, 52, 203, 204], as well as precision and stragglers [49, 50, 57, 205], to name a few. In addition to matrix multiplication, other important non-linear function classes include sparse polynomial multiplication [206], permutation invariant functions [21] — which often appear in multi-agent settings and have applications in learning, combinatorics, and graph neural networks — as well as nomographic functions [59, 60], which can appear in the context of sensor networks and which have strong connections with interference exploitation and lattice codes,

as nicely revealed in [59, 60].

Our own study in this chapter 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 [43, 207, 208] 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.

3.1.2 Data Correlation and Structure

Crucial in reducing the communication bottleneck of distributed computing is the 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 [82] and Cover [209]. 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 [64], which focused on efficient compression of the modulo 2 sum of two statistically dependent sources, while Lalitha *et al.* [69] 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 [210] generalized the pioneering work of Wyner and Ziv [102] 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 [43, 45, 207, 211–213]. 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 [214], data fusion, and group inferences [215], as well as predictive modeling for artificial intelligence [216] — entail datasets with prominent dependencies and correlations. While various works, such as those by Körner-Marton [64], Han-Kobayashi [63], Yamamoto [210], Alon-Orlitsky [75], 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.

3.1.3 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 [72] and used in data compression [64, 74–76, 78, 217], cryptography [103], image processing [218], and bioinformatics [219]. For example, toward understanding the fundamental limits of distributed functional compression, the work in [72] 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 [75] and the lossy encoding setting in [74, 210] use Körner's graph entropy [72] 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 [74, 75, 78, 103, 217]. The same characteristic graph approach nicely used by Feizi and Médard in [77] 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 the distributed computing of non-linear functions.

In this chapter, 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 [72] 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 [72] 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 sub-functions 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 [43, 82], 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 3.4.2) 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 [43, 207].

In the end, our study 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.

3.1.4 Contributions

In this chapter, 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 [72] 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 [72] to the new multi-server multi-function setting, we are able to reflect the nature of the functions and data statistics 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 sub-functions of the data available in its storage, thereby providing 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 [43, 82], 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 3.4.2) 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 [43, 207].

In the end, our study 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.

Notation: We denote by $H(X) = \mathbb{E}[-\log P_X(x)]$ the Shannon entropy of a random variable X drawn from the 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. Matrices and vectors are denoted by boldface letters, e.g., $\mathbf{A}$. 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 $\mathcal{S}^c = \Omega \backslash \mathcal{S}$ 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 use the convention $\mod \{b, a\} = a$ if a divides b.

3.2 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 (see Figure 3.1).

3.2.1 Datasets, Sub-functions, 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.

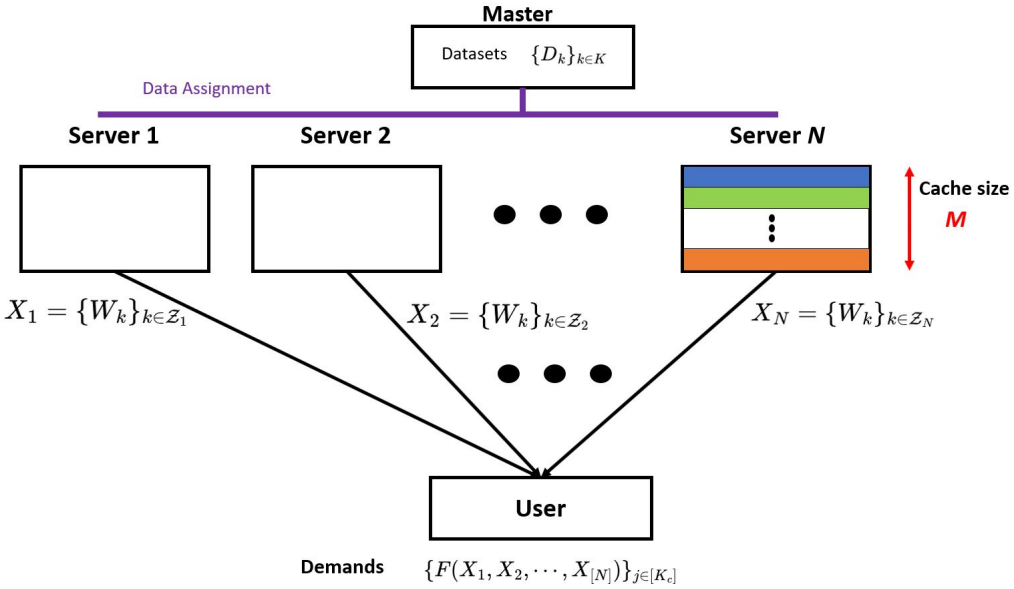

Figure 3.1: A multi-server multi-function distributed setting.

Each server computes a set of sub-functions $W_k = h_k(D_k)$ for $k \in \mathcal{Z}_i \subseteq [K]$, $i \in \Omega$. Datasets $\{D_k\}_{k\in[K]}$ could be dependent[2] 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 sub-functions of the i-th server, $\mathcal{X}_i$ be the alphabet of X_i, and $X_\Omega = (X_1, X_2, \ldots, X_N)$ be the set of sub-functions of all servers. We denote with $\mathbf{W}_k = W_{k1}, W_{k2}, \ldots, W_{kn}$ and $\mathbf{X}_i = X_{i1}, X_{i2}, \ldots, X_{in} \in \mathbb{F}_q^{|\mathcal{Z}_i|\times n}$, the length n sequences of sub-function W_k, and of $W_{\mathcal{Z}_i}$ assigned to server $i \in \Omega$.

3.2.2 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., $\frac{K}{N} \doteq \Xi \in \mathbb{Z}^+$. The dataset placement on N distributed servers is conducted in a circular or cyclic manner in the number of Ξ 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 $|\mathcal{Z}_i| = M = \Xi \cdot (N - N_r + 1)$, and the amount of dataset overlap between the consecutive servers is $\Xi \cdot (N - N_r)$.

Hence, the set of indices assigned to server $i \in \Omega$ is given as follows:

$$\mathcal{Z}_i = \bigcup_{r=0}^{\Xi-1} \{ \mod \{i, N\} + rN, \quad \mod \{i+1, N\} + rN, \ldots, \quad \mod \{i+N-N_r, N\} + rN\}, \tag{3.1}$$

[2]We note that by exploiting the temporal and spatial variation or dependence of data, it is possible to decrease the communication cost.

where $X_i = W_{\mathcal{Z}_i}$, $i \in \Omega$. As a result of (3.1), 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 $i \in \Omega$.

3.2.3 User Demands and Structure of the Computation

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

$$F_j(\mathbf{X}_\Omega) = \{F_j(X_{1l}, X_{2l}, \ldots, X_{Nl})\}_{l=1}^{n}, \quad j \in [K_c] , \tag{3.2}$$

where $F_j(X_{1l}, X_{2l}, \ldots, X_{Nl})$ is the function outcome for the l-th realization $l \in [n]$, given the length n sequence. We note that the representation in (3.2) 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 sub-functions $\{W_k\}_{k\in\mathcal{K}}$ 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:

$$f_j(W_\mathcal{K}) = f_j(h_1(D_1), \ldots, h_K(D_K)) = F_j(\{h_k(D_k)\}_{k\in\mathcal{Z}_1}, \ldots, \{h_k(D_k)\}_{k\in\mathcal{Z}_N}) = F_j(X_\Omega) . \tag{3.3}$$

In the special case of linearly separable functions[3] [43], the demanded functions take the form:

$$\{F_j(X_\Omega)\}_{j\in[K_c]} = \begin{bmatrix} F_1 & F_2 & \ldots & F_{K_c} \end{bmatrix}^\intercal = \mathbf{\Psi}\mathbf{W} , \tag{3.4}$$

where $\mathbf{W} = \begin{bmatrix} W_1 & W_2 & \ldots & W_K \end{bmatrix}^\intercal \in \mathbb{F}_q^{K\times 1}$ is the sub-function vector, and the coefficient matrix $\mathbf{\Psi} = \{\psi_{jk}\} \in \mathbb{F}_q^{K_c\times K}$ 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 sub-functions $\{W_k\}_k$, where $F_j(X_\Omega) = \sum_{k\in[K]} \psi_{jk} \cdot W_k$. We do not restrict $\{F_j(X_\Omega)\}_{j\in[K_c]}$ to linearly separable functions, i.e., it may hold that $\{F_j(X_\Omega)\}_{j\in[K_c]} \neq \mathbf{\Psi}\mathbf{W}$.

3.2.4 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

[3]Special instances of the linearly separable representation of sub-functions $\{W_k\}_k$ given in (3.4) are linear functions of the datasets $\{D_k\}$ and are denoted by $F_j = \sum_k \psi_{jk} D_k$.

demanded functions, the server builds the n-fold OR product $G_{X_i}^n$ of G_{X_i} for compressing $\mathbf{X}_i$ to determine the transmitted information. The minimal possible code rate achievable to distinguish the edges of $G_{X_i}^n$ as $n \to \infty$ is given by the Characteristic graph entropy, $H_{G_{X_i}}(X_i)$. For a primer on key graph-theoretic concepts, characteristic-graph-related definitions, and the fundamental compression limits of characteristic graphs, we refer the reader to [77, 78, 103]. 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 $G_{X_i}^n$, denoted by $\mathcal{C}_{G_{X_i}^n}(\mathbf{X}_i)$. The coloring $\mathcal{C}_{G_{X_i}^n}(\mathbf{X}_i)$ specifies the color classes of $\mathbf{X}_i$ that form ISs 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 $\mathbf{Z}_i = g_i(\mathbf{X}_i) = en(\mathcal{C}_{G_{X_i}^n}(\mathbf{X}_i))$ 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

$$R_i = \frac{T_i}{L} = \frac{H(en(\mathcal{C}_{G_{X_i}^n}(\mathbf{X}_i)))}{n} \geq H_{G_{X_i}}(X_i) \ , \quad i \in \Omega \ , \tag{3.5}$$

where the inequality follows from exploiting the achievability of

$$H_{G_{X_i}}(X_i) = \lim_{n\to\infty} \frac{1}{n} H^{\chi}_{G_{X_i}^n}(\mathbf{X}_i),$$

where $H^{\chi}_{G_{X_i}^n}(\mathbf{X}_i)$ is the chromatic entropy of the graph $G_{X_i}^n$ [72, 75][4].

For the multi-server multi-function distributed setup, using the characteristic-graph-based fundamental limit in (3.5), an achievable sum-rate for asymptotic lossless computation is

$$R_{ach} = \sum_{i\in\Omega} R_i \leq \sum_{i\in\Omega} H_{G_{X_i}}(X_i) \ . \tag{3.6}$$

3.3 Main Results

In this section, we analyze the multi-server multi-function distributed computing framework exploiting the characteristic-graph-based approach in [72]. 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 [43] or the Maddah-Ali and Niesen [28] placements, and (iv) general function classes requested by the user, instead of focusing on a particular function type (see, e.g., [43, 45, 220]).

[4]We refer the reader to Appendix B.1.2 and Chapter 2 for a detailed description of the notions of chromatic and graph entropies (cf. (2.18) and (2.17), respectively).

Subsequently, we delve into specific function computation scenarios. First, we present our main result (Theorem 3.1), which is the most general form that captures (i)-(iv). We then demonstrate (in Proposition 3.1) that the celebrated result of Wan *et al.* [43, Theorem 2] can be obtained as a special case of Theorem 3.1, 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 (3.4). Under a correlated and identically distributed Bernoulli dataset model with a skewness parameter $\epsilon \in (0,1)$ for datasets, we next present in Proposition 3.2 the achievable sum rate for computing Boolean functions. Finally, in Proposition 3.3, 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 ϵ 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 3.1), 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 3.1 is the utilization of a characteristic-graph-based compression technique, as proposed by Körner in [72][5].

Theorem 3.1 (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:*

$$R_{ach} \leq \sum_{i=1}^{N_r} \min_{Z_i=g_i(X_i)\,:\,g_i\in\mathscr{C}_i} H_{G^{\cup}_{X_i}}(X_i)\ , \tag{3.7}$$

where

- $G^{\cup}_{X_i} = \bigcup_{j\in[K_c]} G_{X_i,j}$ *is the union characteristic graph*[6] *that server $i\in\Omega$ builds for computing $\{f_j(W_{\mathcal{K}})\}_{j\in[K_c]}$,*
- $\mathscr{C}_i \ni g_i$ *denotes a codebook of functions that server $i\in\Omega$ uses for computing $\{f_j(W_{\mathcal{K}})\}_{j\in[K_c]}$,*
- *each sub-function 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 $g_i\in\mathscr{C}_i$ denotes the transmitted information from server $i\in\Omega$.*

[5]For a more detailed description of characteristic graphs and their entropies, see Appendix B.1.2.
[6]We refer the reader to see Appendix B.1 for the definition of a union of characteristic graphs.

Proof. See Appendix B.2. □

Theorem 3.1 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 (3.7), the codebook $\mathscr{C}_i$ determines the structure of the union characteristic graph $G^{\cup}_{X_i}$, which, in turn, determines the distribution of Z_i. Therefore, the tightness of the rate upper bound relies essentially on the codebook selection[7]. Because (3.7) is not analytically tractable, in the following, we focus on special instances of Theorem 3.1 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 [43, Theorem 2] is embedded by the characterization provided in Theorem 3.1. We next showcase the achievable sum rate result for linearly separable functions.

Proposition 3.1 (Achievable sum-rate using the characteristic graph approach for linearly separable functions and i.i.d. sub-functions 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 $\frac{K}{N} = \Xi \in \mathbb{Z}^+$, and for a set of K_c linearly separable functions, given as in (3.4), requested by the user, and given i.i.d. uniformly distributed sub-functions over a field of characteristic $q \geq 2$, the characteristic-graph-based compression yields the following bound on the achievable communication rate:*

$$R_{ach} \leq \begin{cases} \min\{K_c, \Xi\} N_r \ , & 1 \leq K_c \leq \Xi \cdot Nr \ , \\ \min\{K_c, K\} \ , & \Xi \cdot Nr < K_c \ . \end{cases} \tag{3.8}$$

Proof. See Appendix B.3. □

We note that Theorem 3.1 results in Proposition 3.1 when three conditions hold: (i) the dataset placement across servers is cyclic following the rule in (3.1), (ii) the sub-functions $W_{\mathcal{K}}$ are i.i.d. and uniform over $\mathbb{F}_q$ (see (B.13) in Appendix B.3), and (iii) the codebook $\mathscr{C}_i$ is restricted to linear combinations of sub-functions $W_{\mathcal{K}}$, which yields that the ISs of $G^{\cup}_{X_i}$ satisfy a set of linear constraints[8] 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 [43, Theorem 2], is valid over a field of characteristic $q > 3$. However,

[7] We also note that it is possible to analyze the computational complexity of building a characteristic graph and computing the bound in (3.7) 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, this chapter focuses primarily on the cost of communication, and we leave the computational complexity analysis to future work.

[8] We detail these linear constraints in Appendix B.3, where the set of linear equations given in (B.14) is used to simplify the entropy $H_{G^{\cup}_{X_i}}(X_i)$ of the union characteristic graph $G^{\cup}_{X_i}$ via the expression given in (B.12) for evaluating the upper bound given in (B.10) 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 Proposition 3.1, the characteristic of $\mathbb{F}_q$ is at least 2, i.e., $q \geq 2$, generalizing [43, Theorem 2] 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 3.1 such that the datasets are correlated and identically distributed, where each sub-function follows a Bernoulli distribution with the same parameter ϵ, 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 3.1, the following proposition (Proposition 3.2) holds for general function types (Boolean) regardless of the data assignment.

Proposition 3.2 (Achievable sum-rate using the characteristic graph approach for general functions and identically distributed sub-functions 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 sub-functions 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:*

$$R_{ach} \leq \sum_{i=1}^{N_r} \min_{Z_i = g_i(X_i)\,:\, g_i \in \mathscr{C}_i} h(Z_i) \ , \tag{3.9}$$

where

- *$\mathscr{C}_i \ni g_i : \{0,1\}^M \to \{0,1\}$ denotes a codebook of Boolean functions that server $i \in \Omega$ uses,*
- *$Z_i = g_i(X_i)$ such that $g_i \in \mathscr{C}_i$ denotes the transmitted information from server $i \in \Omega$,*
- *$G^{\cup}_{X_i}$ has two MISs, namely, $u_0(G^{\cup}_{X_i})$ and $u_1(G^{\cup}_{X_i})$, 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*

$$\mathbb{P}(Z_i = 1) = \mathbb{P}(W_{\mathcal{Z}_i} \in u_1(G^{\cup}_{X_i})) \ , \quad i \in \Omega \ . \tag{3.10}$$

Proof. See Appendix B.4. □

While, admittedly, the above approach (Proposition 3.2) 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 3.1- 3.2, which give the total communication costs for computing linearly separable and Boolean functions, respectively, over $\mathbb{F}_2$, Proposition 3.2, by exploiting the skew and correlations of datasets indexed by $\mathcal{Z}_i$, as well as the functions' structures via the MISs $u_0(G^{\cup}_{X_i})$ and $u_1(G^{\cup}_{X_i})$ 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 3.1, devised with the assumption of i.i.d. and uniformly

distributed sub-functions.

The prior works have focused on devising distributed computation frameworks and exploring their communication costs for specific function classes. For instance, in [64], Körner and Marton have restricted the computation to be the binary sum function, and in [63], 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 [82]. Furthermore, the computation problem has been studied for specific topologies, e.g., the side information setting in [74, 75]. Despite the existing efforts, see, e.g., [63, 64, 74, 75], 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 [82]. On the other hand, the upper bound on the achievable computation scheme presented in Theorem 3.1 can provide savings in the communication cost over [82] for functions including linearly separable functions and beyond. To that end, we exploit Theorem 3.1 to determine an upper bound on the achievable sum-rate for distributed computing of a multi-linear function in the form of

$$f(W_{\mathcal{K}}) = \prod_{k\in[K]} W_k \ . \tag{3.11}$$

Note that (3.11) is used in various scenarios, including distributed machine learning, e.g., to reduce variance in noisy datasets via ensemble learning [221] or weighted averaging [222], sensor network applications to aggregate readings for improved data analysis [223], as well as distributed optimization and financial modeling, where these functions play pivotal roles in establishing global objectives and managing risk and return [224, 225].

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 3.1, our next result, Proposition 3.3, 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 3.3 (Achievable sum-rate using the characteristic graph approach for multi-linear functions and i.i.d. sub-functions 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 $\frac{K}{N} = \Xi \in \mathbb{Z}^+$, and for computing the multi-linear function ($K_c = 1$), given as in (3.11), requested by the user, and given i.i.d. uniformly distributed sub-functions $W_k \sim \text{Bern}(\epsilon)$, $k \in [K]$, for some $\epsilon \in (0, 1)$, the characteristic-graph-based compression yields the following bound on the achievable communication rate:*

$$R_{ach} \leq \frac{1-(\epsilon_M)^{N^*}}{1-\epsilon_M} \cdot h(\epsilon_M) + (\epsilon_M)^{N^*} \cdot 1_{\Xi_N>0} \cdot h(\epsilon_{\xi_N}) \ , \tag{3.12}$$

where

- $\epsilon_M = \epsilon^M$ *denotes the probability that the product of* M *sub-functions, with* $W_k \sim \text{Bern}(\epsilon)$ *being i.i.d. across* $k \in [K]$*, take the value one, i.e.,* $\mathbb{P}\Big(\prod_{k\in\mathcal{S}:|\mathcal{S}|=M} W_k\Big) = \epsilon_M$*,*
- *the variable* $N^* = \Big\lfloor \frac{N}{N-N_r+1} \Big\rfloor$ *denotes the minimum number of servers needed to compute* $f(W_\mathcal{K})$*, given as in (3.11), where each of these servers computes a disjoint product of* M *sub-functions, and*
- *the variable* $\Xi_N = N - N^* \cdot (N - N_r + 1)$ *represents whether an additional server is needed to aid the computation, and if* $\Xi_N \geq 1$*, then* ξ_N *denotes the number of sub-functions to be computed by the additional server, and similarly to the above,* $\mathbb{P}\Big(\prod_{k\in\mathcal{S}:|\mathcal{S}|=\xi_N} W_k\Big) = \epsilon_{\xi_N}$*.*

Proof. See Appendix B.5. □

We next detail two numerical examples (Subsections 3.4.1-3.4.2) to showcase the achievable gains in the total communication cost for Proposition 3.2 and Proposition 3.3, respectively.

3.4 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 3.2, in Section 3.4.1, we focus on computing linearly separable functions, and in Section 3.4.2 (cf. Proposition 3.3), 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 η_{lin} the gain of the sum-rate for the characteristic-graph-based approach given in (3.9) over the sum-rate of the distributed scheme of Wan *et al.* in [43], given in (3.8), and by η_{SW} the gain of the sum-rate in (3.9) over the sum-rate of the fully distributed approach of Slepian-Wolf [82]. 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 [43] via modifying the i.i.d. dataset assumption.

We next study an example scenario (Section 3.4.1) for computing a class of linearly separable functions (3.4) over $\mathbb{F}_2$, where each of the demanded functions takes the form $f_j(W_\mathcal{K}) = \sum_{k\in[K]} \psi_{jk} W_k \mod 2$, $j \in [K_c]$ under a specific correlation model across sub-functions. More specifically, when the sub-functions $W_k \sim \text{Bern}(\epsilon)$ are identically distributed and correlated across $k \in [K]$, and $\Xi \in \mathbb{Z}^+$, we model the correlation across datasets (a) exploiting the joint PMF model in [226, Theorem 1] and (b) for a joint PMF described in Table 3.1. Furthermore, we assume for $K_c > 1$ that $\mathbf{\Psi} = \{\psi_{jk}\} \in \mathbb{F}_2^{K_c \times K}$ is full rank. For the proposed setting, we next demonstrate the achievable gains η_{lin} of our proposed technique versus ϵ for computing (3.4) as a function of skew, ϵ, and correlation,

ρ, of datasets, $K_c \in [N_r] < K$, and other system parameters and showcase the results via Figures 3.2, 3.4-3.6.

Table 3.1: Joint PMF P_{W_2,W_3} of W_2 and W_3 with a crossover parameter p, in Section 3.4.1 (Scenario II).

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

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

We consider the computation of the linearly separable functions given in (3.4) 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 sub-function, where $W_k \sim$Bern(ϵ), $k \in [K]$, using cyclic placement as in (3.1) and incorporating the correlation between the sub-functions, with the correlation coefficient denoted by ρ. We consider three scenarios, as described next:

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

This scenario is similar to the setting in [43], although different from [43], 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 sub-functions of datasets) from distributed servers, as detailed earlier in Section 3.3. We first assume that the sub-functions are not correlated, i.e., $\rho = 0$, and evaluate η_{lin} for $f(W_{\mathcal{K}}) = \sum_{k\in[K]} W_k \mod 2$. 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

$$\begin{aligned}\mathbb{P}\Big(\sum_{k\in\mathcal{S}:\,|\mathcal{S}|=l\le K} W_k \mod 2 = 1\Big) &= \sum_{k\in\mathcal{S}:\,|\mathcal{S}|=l\le K\ ,\,k\,\text{odd}} \mathbb{P}(B(K,\epsilon))=k) \\ &= (1-\epsilon_{l-1})\cdot\epsilon + \epsilon_{l-1}\cdot(1-\epsilon) \\ &\doteq \epsilon_l\ , \quad 1<l\le K,\end{aligned} \tag{3.13}$$

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

Given N_r, we denote by $N^* = \left\lfloor \frac{N}{N-N_r+1} \right\rfloor$ 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 sub-functions, where across these N^* servers, the sets of sub-functions are disjoint. Hence, $\mathbb{P}\Big(\sum_{k\in\mathcal{S}:\,|\mathcal{S}|=M} W_k\Big) = \epsilon_M$. Furthermore, the variable $\Xi_N = N - N^*\cdot(N-N_r+1)$ represents whether additional servers in addition to N^*

servers are needed to aid the computation, and if $\Xi_N \geq 1$, then $\Xi \cdot \Xi_N \doteq \xi_N$ denotes the number of sub-functions to be computed by the set of additional servers, namely, $\mathcal{I}^* \in \Omega$, and similarly to the above, $\mathbb{P}\Big(\sum_{k\in\mathcal{S}:|\mathcal{S}|=\xi_N} W_k\Big) = \epsilon_{\xi_N}$, which is obtained by evaluating ϵ_l at $l = \xi_N$.

Adapting (3.8) for $\mathbb{F}_2$, we obtain the total communication cost $R_{ach}(lin)$ for computing the linearly separable function $f(W_{\mathcal{K}}) = \sum_{k\in[K]} W_k \mod 2$ as

$$R_{ach}(lin) = \sum_{i=1}^{N_r} h\Big(\sum_{k\in\mathcal{Z}_i} W_k\Big) = N_r \cdot h(\epsilon_M) \ . \tag{3.14}$$

Using Proposition 3.2 and (3.13), we derive the sum rate for distributed lossless computing of $f(W_{\mathcal{K}})$ as

$$\sum_{i\in\Omega} R_i \leq N^* \cdot h(\epsilon_M) + 1_{\Xi_N>0} \cdot h(\epsilon_{\xi_N}) \ , \tag{3.15}$$

where the indicator function $1_{\Xi_N>0}$ captures the rate contribution from the additional server, if any. Using (3.15), the gain η_{lin} over the linearly separable solution of [43] is presented as

$$\eta_{lin} = \frac{N_r \cdot h(\epsilon_M)}{N^* \cdot h(\epsilon_M) + 1_{\Xi_N>0} \cdot h(\epsilon_{\xi_N})} \ , \tag{3.16}$$

where $h(\epsilon_{\xi_N})$ represents the rate needed from the set of additional servers $\mathcal{I}^* \in \Omega$, aiding the computation through communicating the sum of the remaining sub-functions in the set $\mathscr{C} \subseteq \mathcal{Z}_{\mathcal{I}^*}$, where the summation for these remaining functions in $\mathscr{C} \subseteq \mathcal{Z}_{\mathcal{I}^*}$ is denoted as $\sum_{k\in\mathscr{C}\subseteq\mathcal{Z}_{\mathcal{I}^*} \ : \ k\notin\bigcup_{i\in\mathcal{N}^*}\mathcal{Z}_i \ , \ |\mathscr{C}|=\xi_N} W_k$, 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 [226] for each W_k, identically distributed with $W_k \sim \text{Bern}(\epsilon)$, and correlation ρ across any two sub-functions. The formulation in [226] yields the following PMF for $f(W_{\mathcal{K}})$:

$$\begin{aligned}\mathbb{P}(f(W_K) = y) = &\binom{K}{y}\epsilon^y(1-\epsilon)^{K-y}(1-\rho)\cdot 1_{y\in A_1} \\ &+ \epsilon^{\frac{y}{K}}(1-\epsilon)^{\frac{K-y}{K}}\rho\cdot 1_{y\in A_2} \ , \ y\in\{0,\ldots,K\},\end{aligned} \tag{3.17}$$

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

We depict the behavior of our gain, η_{lin}, using the same topology $\mathcal{T}(N,K,K_c,M,N_r)$ as in [43], with different system parameters (N,K,M,N_r), under $\rho = 0$ in Figure 3.2-(Left). As we increase both N and K, along with the number of active servers, N_r, the gain, η_{lin}, of the characteristic graph approach increases. This stems from the characteristic graph approach to compute functions $f(W_{\mathcal{K}})$ of $W_{\mathcal{K}}$ using N^* servers. From Figure 3.2-(Right), it is evident that by capturing correlations between the sub-functions, hence, across the servers' caches, η_{lin} grows more rapidly until it reaches the maximum

of (3.16), corresponding to $\eta_{lin} = \frac{N_r}{N^*} = 10$, attributed to full correlation (see, Figure 3.2-(Right)).

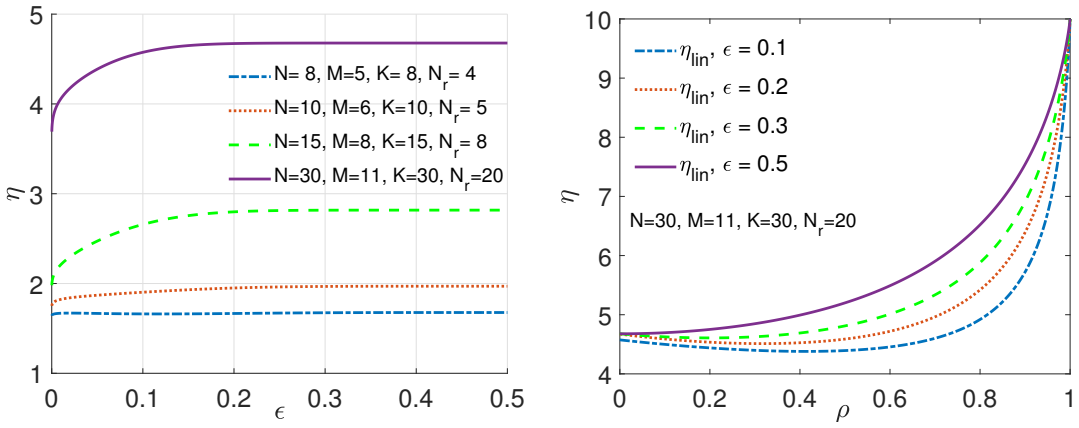

Figure 3.2: The gain η_{lin} of the characteristic graph approach for $K_c = 1$ in Section 3.4.1 (Scenario I). (Left) $\rho = 0$ for various distributed topologies. (Right) The correlation model given as (3.17) for $\mathcal{T}(30, 30, 1, 11, 20)$ with different ϵ values.

What we can also see is that for $\rho = 0$, the gain rises with the increase in ϵ and linearly grows with $\frac{N_r}{N^*}$. As ρ 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 η_{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 ρ goes to 1, over the state of the art [43], as shown in Figure 3.2-(Right) via the purple (solid) curve. Furthermore, one can draw a comparison between the characteristic graph approach and the approach in [82]. Here, we represent the gain as η_{SW}. It is noteworthy that the sum-rate of all servers using the coding approach of Slepian-Wolf [82] is $R_{ach}(SW) = H(W_{\mathcal{K}})$. 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 (3.14) for the same topology of the purple (solid) curve as shown in Figure 3.2-(Right).

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

To gain insights into the behavior of η_{lin}, we consider an example distributed computation model with $K = N = 3$, $N_r = 2$, where the sub-functions 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 $f_1(W_{\mathcal{K}}) = W_2$, and $f_2(W_{\mathcal{K}}) = W_2 + W_3$.

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 $j \in \Omega \backslash i$ 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[9]. The achievable sum rate here takes the form

$$R_{ach}(G) = \min\{H_{G_{X_1}}(X_1) + H_{G_{X_2}}(X_2 \,|\, Z_1), \quad H_{G_{X_2}}(X_2) + H_{G_{X_1}}(X_1 \,|\, Z_2)\} \,. \tag{3.18}$$

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 (B.4) (as detailed in Appendix B.1.4), in Figure 3.3. In (3.18), the first term corresponds to $G_{X_1} = G(\mathcal{V}_{X_1}, \mathcal{E}_{X_1})$, where $\mathcal{V}_{X_1} = \{0,1\}^2$ is built using the support of W_1 and W_2, and the edges $\mathcal{E}_{X_1}$ are built based on the rule that $(x_1^1, x_1^2) \in \mathcal{E}_{X_1}$ if $F(x_1^1, x_2) \neq F(x_1^2, x_2)$ for some $x_2 \in \mathcal{V}_{X_2}$, which, as we see here, requires two colors. Similarly, server 2 constructs $G_{X_2} = G(\mathcal{V}_{X_2}, \mathcal{E}_{X_2})$ given Z_1, where $\mathcal{V}_{X_2} = \{0,1\}^2$ 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[10] $x_2^1 = (w_2^1, w_3^1) \in \mathcal{V}_{X_2}$ and $x_2^2 = (w_2^2, w_3^2) \in \mathcal{V}_{X_2}$ 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 (3.18) captures the impact of $G_{X_2} = G(\mathcal{V}_{X_2}, \mathcal{E}_{X_2})$, 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 (3.18) 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 (3.18). The different N_r achieved by these two configurations is also captured by Figure 3.3.

Alternatively, in the linearly separable approach [43], 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 [43] 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 η_{lin} of the characteristic graph approach over the linearly separable solution of [43] for computing f_1 and f_2 as a function

[9] We can generalize (3.18) 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., $R_{ach}(G)$.

[10] 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 sub-functions W_2 and W_3.

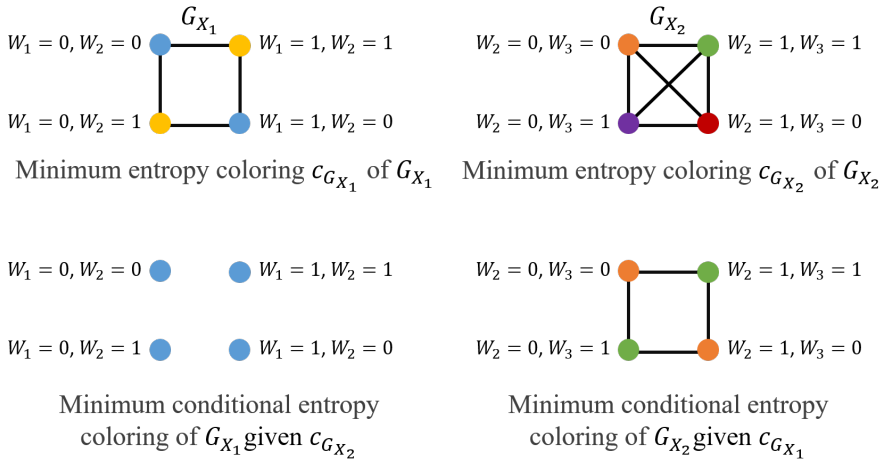

Figure 3.3: Colorings of graphs in Section 3.4.1 (Scenario II). (Top Left-Right) Characteristic graphs G_{X_1} and G_{X_2}, respectively. (Bottom Left-Right) The minimum conditional entropy colorings of G_{X_1} given $\mathcal{C}_{G_{X_2}}$ and G_{X_2} given $\mathcal{C}_{G_{X_1}}$, respectively.

of $\epsilon \in [0,1]$ takes the form

$$\eta_{lin}(\epsilon) = \frac{h(\epsilon) + h(2\epsilon(1-\epsilon))}{2h(\epsilon)} \begin{cases} = 1 & , \ \epsilon = \{\frac{1}{2}\} \ , \\ > 1 & , \ \epsilon \in [0,1] \backslash \{\frac{1}{2}\} \ , \end{cases} \tag{3.19}$$

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, η_{lin} approaches 1.5 as $\epsilon \to \{0,1\}$ (see Figure 3.4).

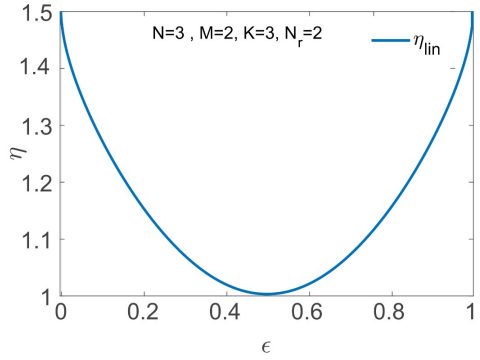

Figure 3.4: η_{lin} in (3.19) versus ϵ, for distributed computing of $f_1 = W_2$ and $f_2 = W_2 + W_3$, where $K_c = 2$, $N_r = 2$, with $\rho = 0$, in Section 3.4.1 (Scenario II).

We next examine the setting where the correlation coefficient ρ is nonzero, using the joint PMF P_{W_2,W_3}, as depicted in Table 3.1, of the required sub-functions (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 [43] is given as

$$\eta_{lin} = \frac{H(W_2) + H(W_2 + W_3)}{H(W_2, W_3)} = \frac{h(\epsilon) + h(2\epsilon p)}{h(\epsilon) + (1-\epsilon)h(\frac{\epsilon p}{1-\epsilon}) + \epsilon h(p)} \ . \tag{3.20}$$

We consider now the correlation model in Table 3.1, where coefficient ρ rises in ϵ for a fixed p. In Figure 3.5-(Left), we illustrate the behavior of η_{lin}, given by (3.20), for computing f_1 and f_2 for $N_r = 2$ as a function of p and ϵ, where for this setting, the correlation coefficient ρ is a decreasing function of p and an increasing function of ϵ. We observe from (3.20) that the gain η_{lin} satisfies $\eta_{lin} \geq 1$ for all $\epsilon \in [0, 1]$, which monotonically increases in p — and hence monotonically decreases in ρ due to the relation $\rho = \frac{1-p}{1-\epsilon}$ — as a function of the deviation of ϵ from $1/2$. For $\epsilon \in (0.5, 1]$, η_{lin} increases in ϵ. 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 ϵ results in η_{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, η_{lin} goes to 1.5 as ϵ tends to zero, which can be derived from (3.20). We here note that the gains are generally smaller than in the previous set of comparisons, as shown in Figure 3.4.

More generally, given a user request consisting of $K_c = 2$ linearly separable functions (i.e., satisfying (3.4)), and after considering (3.20) beyond $N_r = 2$, we see that η_{lin} is at most N_r as ρ approaches one.

We next use the joint PMF model used in obtaining (3.17), 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

$$\eta_{lin} = \frac{h(\epsilon) + H(f_2)}{h(\epsilon) + (1-\epsilon)h(\iota_1) + \epsilon h(\iota_2)} \ , \tag{3.21}$$

where $\iota_1 = (1-\epsilon)(1-\rho) + \rho$, and $\iota_2 = (1-\epsilon)(1-\rho)$. For this model, we illustrate η_{lin} versus ϵ in Figure 3.5-(Right) for different ρ values. Evaluating (3.21), 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 $H(W_2+W_3) = h((1-\epsilon)^2,\ 2\epsilon(1-\epsilon),\ \epsilon^2) = h(2\epsilon(1-\epsilon)) + ((1-\epsilon)^2+\epsilon^2)h\Big(\frac{\epsilon^2}{\epsilon^2+(1-\epsilon)^2}\Big)$ 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 $\eta_{lin} \geq 1.25$.

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 sub-functions are uncorrelated.

We now provide an achievable rate comparison between the approach in [43] and our graph-based approach, as summarized by our Proposition 3.1, which generalizes the result

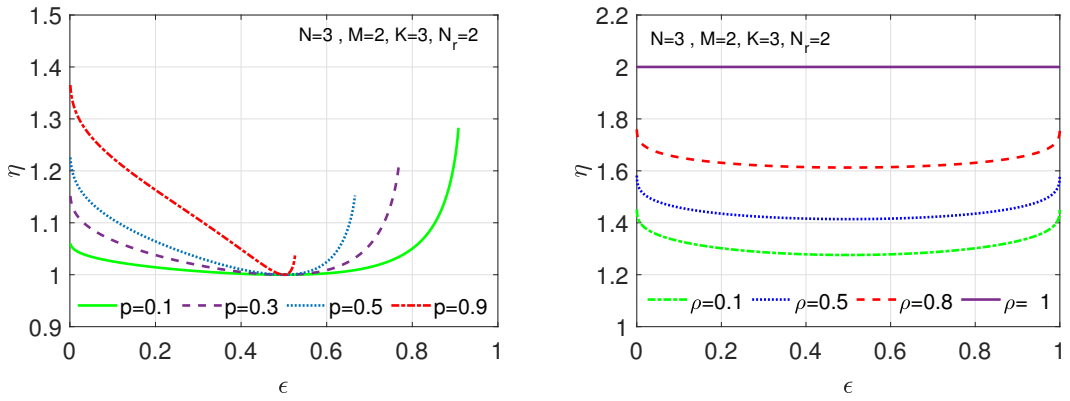

Figure 3.5: η_{lin} versus ϵ, for distributed computing of $f_1 = W_2$ and $f_2 = W_2 + W_3$, where $K_c = 2$, $N_r = 2$, in Section 3.4.1, using different joint PMF models for P_{W_2,W_3} (Scenario II). (Left) η_{lin} in (3.20) for the joint PMF in Table 3.1 for different values of p. (Right) η_{lin} for the joint PMF in (3.17) for different values of ρ.

in [43, Theorem 2] 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 [43] via modifying the i.i.d. dataset assumption and taking into account the skewness incurred within each server in determining the local computations $\sum_{k\in\mathcal{S}:|\mathcal{S}|=M} W_k$ at each server.

For the linearly separable model in (3.4), adapted to account for our setting, exploiting the summation $\sum_{k\in\mathcal{Z}_i} W_k$, and ϵ_M given in (3.15), the communication cost for a general number of K_c with $\rho = 0$ is expressed as

$$R_{ach}(lin) = N_r \cdot h(\epsilon_M) \ . \tag{3.22}$$

In (3.22), as ϵ approaches 0 or 1, then $h(\epsilon_M) \to 0$. Subsequently, the achievable communication cost for the characteristic graph model can be determined as

$$R_{ach}(G) = K_c \cdot N^* \cdot h(\epsilon) \ . \tag{3.23}$$

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[11] for the binary entropy function, which tells us that $h(\mathbb{E}[X]) \geq \mathbb{E}[h(X)]$, we can see that as ϵ approaches 0 or 1, then

$$\lim_{\epsilon\to\{0,1\}} \frac{h(\epsilon_M)}{h(\epsilon)} \leq M \ , \quad M \in \mathbb{Z}^+ \ , \tag{3.24}$$

where the inequality between the left- and right-hand sides becomes loose as a function of

[11] A real-valued function $f : \mathbb{R}^n \to \mathbb{R}$ is Schur concave if $f(x_1, x_2, \ldots, x_n) \leq f(y_1, y_2, \ldots, y_n)$ holds whenever $(x_1, x_2, \ldots, x_n)$ majorizes $(y_1, y_2, \ldots, y_n)$, i.e., $\sum_{i=1}^{k} x_i \geq \sum_{i=1}^{k} y_i$, for all $k \in [n]$ [227].

M. As a result, as ϵ approaches 0 or 1, then $\eta_{lin} \approx M \cdot \frac{N_r}{K_c \cdot N^*}$, which follows from exploiting (3.22), (3.23) and the achievability of the upper bound in (3.24). We illustrate the upper bound on η_{lin} in Figure 3.6 and demonstrate the η_{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 $\Xi = \frac{K}{N} = 1$ and the cache size is $M = N - N_r + 1$, and for $\rho = 0$ and $\epsilon \leq \frac{1}{2}$. We focus only on plotting η_{lin} for $\epsilon \leq \frac{1}{2}$, accounting for the symmetry of the entropy function. Therefore, we only plot for $\epsilon \leq \frac{1}{2}$. The multiplicative coefficient $\frac{N_r}{K_c N^*}$ of η_{lin} determines the growth, which is depicted by the curves.

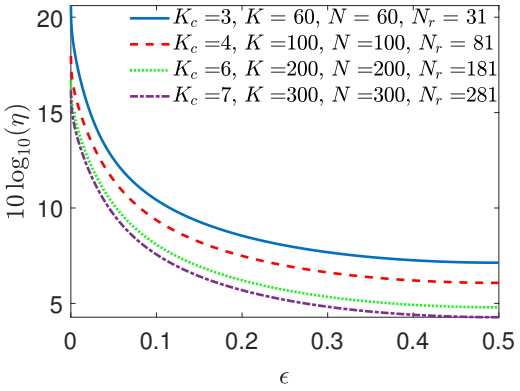

Figure 3.6: η_{lin} in a logarithmic scale versus ϵ for K_c demanded functions for various values of K_c, with $\rho = 0$ for different topologies, as detailed in Section 3.4.1 (Scenario III).

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 (3.24), we see that η_{lin} exponentially grows with term $1 - \epsilon$ for $\epsilon \in [0, 1/2]$[12], and very substantial reduction in the total communication cost is possible as ϵ approaches $\{0, 1\}$, as shown in Figure 3.6 by the blue (solid) curve. The gain over [43, Theorem 2], η_{lin}, for a given topology, changes proportionally to $\frac{N_r}{K_c N^*}$. The gain over [82], η_{SW}, for $\rho = 0$ linearly scales[13] with $\frac{K}{K_c N^*}$. For instance, the gain for the blue (solid) curve in Figure 3.6 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 3.3) are more skewed and have lower entropies than linearly separable functions and, hence, are easier to compute. Therefore, the cost given in (3.23) 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 η_{lin} for more general topologies $\mathcal{T}(N, K, K_c, M, N_r)$ and correlation models beyond (3.17) devised for linearly separable functions, and beyond

[12]Here, we note that the behavior of η_{lin} is symmetric around $\epsilon = 1/2$.

[13]Incorporating the dataset skew to Proposition 3.1 ([43, Theorem 2]), $R_{ach}(lin)$ is simplified to (3.22), which from (3.24) can linearly grow in $M = N - N_r + 1$ at high skew, explaining the inferior performance of Proposition 3.1 over [82] as a function of the skew.

the joint PMF model in Table 3.1, as future work.

Proposition 3.3 illustrates the power of the characteristic graph approach in decreasing the communication cost for distributed computing of multi-linear functions, given as in (3.11), compared to recovering the local computations $\prod_{k\in\mathcal{S}:|\mathcal{S}|=M} W_k$ using [82]. We denote by η_{SW} the gain of the sum-rate for the graph entropy-based approach given in (3.12) — using the conditional entropy-based sum-rate expression in (B.22) — over the sum-rate of the fully distributed scheme of Slepian-Wolf [82] for computing (3.11). For the proposed setting, we next showcase the achievable gains η_{SW} of Proposition 3.3 via an example and showcase the results via Figure 3.7.

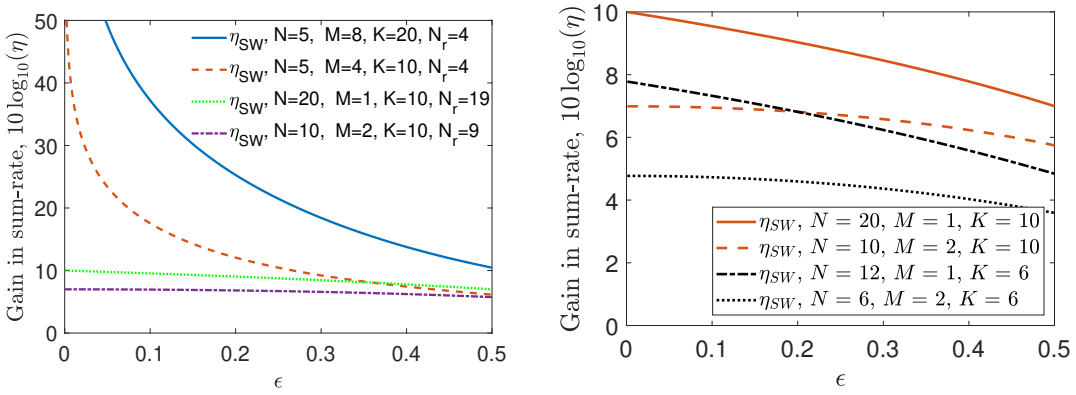

Figure 3.7: Gain $10\log_{10}(\eta_{SW})$ versus ϵ for computing (3.11), where $K_c = 1$, $\rho = 0$, $N_r = N - 1$. (Left) The set of parameters N, K, and M are indicated for each configuration. (Right) $10\log_{10}(\eta_{SW})$ versus ϵ to observe the effect of N for a fixed total cache size MN and fixed K.

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

We study the behaviors of η_{SW} versus the skewness parameter ϵ for computing the multi-linear function given in (3.11) 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, $M = \Xi \cdot (N - N_r + 1)$, 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 3.3 to determine the sum-rate upper bound and illustrate the gains $10\log_{10}(\eta_{SW})$ in decibels versus ϵ in Figure 3.7.

From the numerical results in Figure 3.7 (Left), we observe that the sum-rate gain of the graph entropy-based approach versus the fully distributed approach of [82], η_{SW}, could reach up to more than 10-fold gain in compression rate for uniform and up to 10^6-fold for skewed data. The results for η_{SW} showcase that our proposed scheme can guarantee an exponential rate reduction over [82] as a function of decreasing ϵ. Furthermore, the sum-rate gains scale linearly with the cache size M, which scales with K given $N_r = N - 1$. Note that η_{SW} diminishes with increasing N when M and Ξ are kept fixed. In Figure 3.7 (Right), for $M \ll K$, a fixed total cache size MN, and hence, fixed K, the gain η_{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.

3.5 Conclusions

In this chapter, 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 3.4.1 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 3.4.2, we demonstrate a very substantial reduction in the total communication cost versus the state of the art. Our main results (Theorem 3.1 and Propositions 3.1-3.3) 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 3.1 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 (MAN), detailed in [28] in order to capture the finer-grained granularity of placement that can help tighten the achievable rates (see Appendix B.6). They also involve, beyond the special cases detailed in Propositions 3.1-3.3, 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.

Chapter 4

Non-Linear Function Computation Broadcast

This chapter[1] focuses on the K-user computation broadcast problem over a noiseless broadcast channel, where a master node holding all datasets communicates with distributed users demanding linear or non-linear functions over finite fields. We establish bounds on the optimal broadcast rate by capturing the interaction between computation structures and the user's side information. A graph-based coding framework is developed for achievability, while converse arguments provide tight lower bounds. Examples demonstrate notable rate reductions compared to existing approaches, particularly in the multi-user setting.

4.1 Introduction

In the presence of large-scale data, communication networks are being increasingly deployed for various distributed computation scenarios, including distributed learning [169, 230], over-the-air computing [61, 231], private-key encryption [232, 233], and coded distributed computing [56, 234], to name a few. In such scenarios, meeting the growing and diverse demands of distributed users is exacerbated by the limited communication resources, privacy constraints, and bounded memory and computational capabilities of users. This motivates us to study the problem of computation broadcast, in which source data is communicated to recover functions of data at distributed users, which finds applications in radio broadcasting, digital video broadcasting, and cryptography [235–237].

4.1.1 Motivation and Contributions

Prior works in computation broadcast have focused on linear function demands [1, 94, 238, 239], which allow for exploiting algebraic techniques to encode the source data, or complementary function demands [95]. However, these approaches often depend on the

[1] The results in this chapter are presented in [228, 229].

specific demand structures or the number of users, limiting their general applicability. Furthermore, to the best of our knowledge, no general bound on the broadcast rate has been established for non-linear demands.

This chapter tackles the problem of function computation broadcast, accommodating general function demands (linear, non-separable, and non-linear) over finite fields. We develop a master-users framework using *characteristic graphs* to capture demand structures. The master node has access to all datasets and computation capabilities and transmits information to multiple users. Each user, having a fraction of the master's datasets as side information, computes a general function of the datasets.

Our main contributions are summarized as follows.

- We devise a coding scheme for the K-user computation broadcast that tackles general data distributions, and general non-linear function demands over $\mathbb{F}_q$, denoting a finite field of order $q \geq 2$. This framework, as detailed in Section 4.2, emerges from using characteristic graphs devised by Körner [72] to identify a collection of equivalent source classes. The master builds a broadcast message to enable all users to decode their demanded functions simultaneously in an asymptotically lossless manner.
- This framework reduces communication costs by performing part of the computation at the master node and transmitting functions of datasets as the broadcast message, incorporating the structure of datasets and K user demands (Sections 4.2 and 4.3). For general non-linear demands, our approach outperforms [82]. In a 3-user Boolean computation broadcast model, the achievable rate is 1.5 bits, improving on the 2 bits of [82] (Example 4.1, Section 4.3). Similarly, for a 3-user linear computation broadcast model, we achieve 1.42 bits, outperforming [1] and [82] with rates of 1.5 and 2 bits, respectively (Example 4.2, Section 4.3).
- We derive an achievability scheme and a converse on the communication rate for $K \geq$ 2-user computation broadcast with linear and non-linear demands in Section 4.4.

4.1.2 State of the Art

Communication complexity: Communication complexity, which quantifies the minimum bits required for transmission among multiple parties, was introduced for the distributed computation of Boolean functions [79] and later extended to approximating continuous functions [240], computing rational functions [80], and estimating correlations [81]. Interactive communication models have demonstrated significant communication savings compared to one-way models [241, 242]. While previous studies have explored communication complexity across various multi-party communication models, characterizing this complexity for computation broadcast problems with general function demands remains an open problem.

Compression for computing: Structured codes reduce rates in distributed computation for tasks like modulo-q sum [64, 69], linear functions [5], and matrix multiplica-

tion [71]. Unstructured codes address point-to-point lossy recovery [102], function computation with side information [210], and multi-terminal setups [82, 243–245]. Körner's characteristic graph framework [72] is used to compress sources for distributed computing [63, 74, 75, 77, 103, 110, 111, 246, 247]. Extending this to broadcast models is difficult due to the need to handle function and side information dependencies in message design.

Coded distributed computing: To meet growing computational demands and speed up computing via distributing tasks among multiple servers, techniques like distributed machine learning [248], federated learning [249], and massive parallelization algorithms, e.g., MapReduce [22] and Spark [23], have been proposed. Most strategies exploit the separable nature of functions, e.g., matrix multiplication [57] and gradient descent [101], and design codes to distribute tasks, focusing on straggler mitigation [205], communication complexity [56, 101], and communication-computation tradeoffs [45, 57]. However, they fall short for general non-linear demands.

Computation broadcast: The computation broadcast framework generalizes the index coding problem [32], where users demand source variables instead of their functions, and multiple description coding [250], a special case where all users demand the same distorted source function over $\mathbb{F}_q$ with varying distortion criteria. Linear computation broadcast has been studied for two and more users settings [1, 239]. However, for non-linear computation broadcast, only the case of $K = 2$ users with complementary side information has been explored [95]. Here, we extend [95] to $K \geq 2$ users without structural assumptions on side information.

Notation: We denote random variables and their realizations in regular typeface, and vectors and matrices in boldface, with elements chosen from $\mathbb{F}_q$, where $\mathbb{F}_q$ denotes a finite field of order $q \geq 2$. We denote the binary logarithm, and the logarithm in base $q > 2$, by log and $\log_q$, respectively. Given discrete random variables X and Y over finite alphabets $\mathcal{X}$ and $\mathcal{Y}$, respectively, the notation P_X denotes the probability mass function (PMF) of X, and $P_{X,Y}$ denotes the joint PMF of X and Y. We denote the probability of event A by $\mathbb{P}(A)$. We use $X \sim \text{Unif}(\mathbb{F}_q)$ to denote a uniform random variable over $\mathbb{F}_q$. The Shannon entropies in binary and q-ary units are given by $H(X) = H_2(X)$ and $H_q(X) = \mathbb{E}[-\log_q P_X] = (1/\log q)H(X)$, respectively. $\wedge$ and $\vee$ denote binary AND and OR operations, respectively.

Given a $(K, N, \{\mathcal{S}_i\}_{i\in[K]}, \{f_i\}_{i\in[K]})$ non-linear function computation broadcast model with a sender node having N datasets $\mathbf{X}_{[N]} \triangleq \{X_j\}_{j\in[N]}$, K users, each with side information $\mathcal{S}_i$ and demand f_i, for $i \in [K]$, we denote by G_{f_i} the *characteristic graph* that the sender node builds for computing f_i. We denote by $G_{\cup_{i=1}^K f_i}$ the union graph built by the master node to simultaneously meet the demands of all K users. The permutations between K users are defined by $\Pi([K])$.

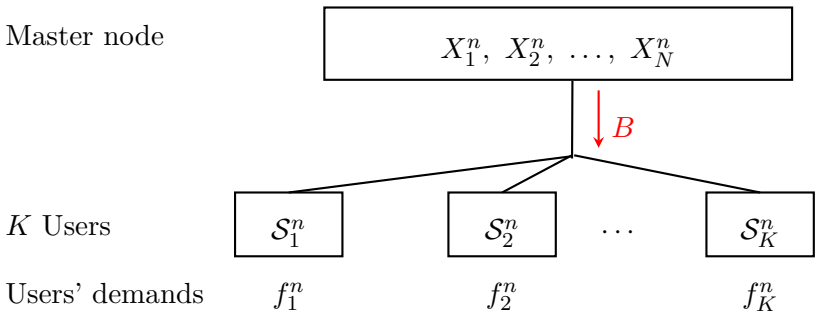

Figure 4.1: A $(K, N, \{\mathcal{S}_i\}_{i\in[K]}, \{f_i\}_{i\in[K]})$ non-linear function computation broadcast model with N datasets $\mathbf{X}_{[N]} = \{X_j\}_{j\in[N]}$, K users, each with side information $\mathcal{S}_i$ and demand f_i, for $i \in [K]$.

4.2 System Model

This section outlines our K-user computation broadcast model, depicted in Figure 4.1. The model consists of a master node, that stores $N \in \mathbb{Z}^+$ datasets, denoted by $\mathbf{X}_{[N]} \triangleq \{X_j\}_{j\in[N]}$, where $X_j \in \mathbb{F}_q$ for $q \geq 2$, and K users. Each of $[K]$ distributed users is equipped with side information, $\mathcal{S}_i \subseteq X_{[N]}$. Users are interested in computing a set of (possibly distinct) functions $\{f_i\}_{i\in[K]}$ using the shorthand notation $f_i = f_i(X_1, \ldots, X_N)$.

The master node, using n i.i.d. realizations of these datasets $X_j^n = \{X_{j,t}\}_{t\in[n]} \in \mathbb{F}_q^n$, broadcasts a message $B = \psi(X_1^n, \ldots, X_N^n)$, designed as a function of the master node's data and the set of demanded functions. The objective of the computation broadcast problem is to minimize the communication rate from the master node, characterized by the entropy of B, denoted by $H_q(B)$, that ensures each user $i \in [K]$ to asymptotically losslessly reconstruct the length n sequence of functions $f_i^n \triangleq \{f_i(X_{1,t}, \ldots, X_{N,t})\}_{t\in[n]}$, using the received message B and the length n realization of $\mathcal{S}_i$, i.e., $\mathcal{S}_i^n = \{\mathcal{S}_{i,t}\}_{t\in[n]}$, i.e., the following condition is met:

$$H_q(f_i^n \mid B, \mathcal{S}_i^n) = 0\ , \quad \forall i \in [K]\ . \tag{4.1}$$

To capture general function demands in the proposed K-user computation broadcast model, we exploit the characteristic graph framework in [72], which has been used to characterize the communication rate for a point-to-point model, where a source holds a random variable X and a receiver has side information Y, and the receiver's goal is to reconstruct a function $f(X, Y)$ in an asymptotically lossless manner [74]. To motivate our approach, we next define characteristic graphs.

Definition 4.1 (**Source characteristic graphs** [74])**.** *Consider a point-to-point communication model with a master node having access to a source random variable $X \in \mathcal{X}$ and a user with side information (finite alphabet) $Y \in \mathcal{Y}$, aiming to compute $f(X, Y)$. The characteristic graph that the master node builds using X for $f(X, Y)$, given Y, is defined as $G_f = G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \mathcal{X}$ denotes the vertex set, and $\mathcal{E}$ the set of edges such that given two distinct vertices $x, x' \in \mathcal{V}$, an edge $(x, x') \in \mathcal{E}$ exists if and only if the*

following conditions hold:

- *i)* $f(x,y) \neq f(x',y)$
- *ii)* $P_{X,Y}(x,y) \cdot P_{X,Y}(x',y) > 0$ *for some* $y \in \mathcal{Y}$.

Characteristic graphs identify which source values can share the same codeword without introducing ambiguity to the user, enabling distributed computation in various topologies, as explored by prior literature, see e.g., [72, 74, 75, 77, 103, 110, 111, 246]. For the point-to-point scenario detailed above, the minimum rate required for a user to compute $f(X,Y)$ given side information Y with vanishing error probability is characterized as [74]

$$\min_{\substack{U-X-Y \\ X \in U \in \Gamma(G_f)}} I(X;U \mid Y) \ , \tag{4.2}$$

where $U - X - Y$ indicates a Markov chain, $I(X;U \mid Y) = H_q(U \mid Y) - H_q(U \mid X,Y)$, and $X \in U \in \Gamma(G_f)$ means that the minimization is over all joint PMFs $P_{U,X}(u,x) > 0$ such that U is an independent set[2] of G_f, and $\Gamma(G_f)$ is the collection of all MISs of G_f.

This chapter extends the characteristic graph framework to address the computation broadcast problem by designing a broadcast message, B, that captures dependencies among the master's datasets $\mathbf{X}_{[N]}$, the structure of f_i, and each user's side information $\mathcal{S}_i$ for $i \in [K]$. This approach serves all users' demands simultaneously without imposing structural constraints (e.g., linearity, separability), unlike prior works [1, 94, 95, 239, 251]. To jointly capture the distinguishability requirements of K users, the framework leverages union graphs, as defined next.

Definition 4.2 (**Union graph given function demands f_1 and f_2** [252])**.** *Given two characteristic graphs $G_{f_1} = G(\mathcal{V},\mathcal{E}_1)$ and $G_{f_2} = G(\mathcal{V},\mathcal{E}_2)$ with the same vertex set $\mathcal{V}$, their union graph, $G_{f_1 \cup f_2}$, is defined as:*

$$G_{f_1 \cup f_2} \triangleq G(\mathcal{V}, \mathcal{E}_1 \cup \mathcal{E}_2) \ , \tag{4.3}$$

where the set of edges is given by the union $\mathcal{E}_1 \cup \mathcal{E}_2$, which captures if a pair of source values needs to be distinguished for at least one of the two functions f_1 and f_2.

Next, using the union graph approach, we will devise an achievability scheme for the computation broadcast problem, which involves the design of the broadcast message B. Generalizing Definition 4.2, we denote by $G_{\cup_{i=1}^K f_i}$ the union graph — referred to as a broadcast graph — built by the master node to simultaneously meet the demands of all K users, where $\Gamma(G_{\cup_{i=1}^K f_i})$ is the set of all MISs of this broadcast graph. From Definition 4.2, an MIS U of $G_{\cup_{i=1}^K f_i}$ satisfies the constraint(4.1).

Assigning a unique codeword to the MISs in $\Gamma(G_{\cup_{i=1}^K f_i})$ ensures that no confusion arises by the assigned codeword in any user $i \in [K]$ in reconstructing f_i. For asymptotically lossless recovery of all demands, the master node builds the n-fold OR product[3] $G^n_{\cup_{i=1}^K f_i}$

[2] We refer the readers to the definitions of the IS and MIS in Chapter 2.2 for detailed explanations.
[3] The n-fold OR products of characteristic graphs are detailed in [75], and Chapters 2-3.

corresponding to the length n realization of $\mathbf{X}_{[N]}$. To that end, the master node uses a binning strategy[4] and builds a mapping $en : \Gamma(G^n_{\cup_{i=1}^K f_i}) \to [0, q^{nR_{ach}} - 1]$, to construct the broadcast message $B = en(U^n)$ using the MISs $U^n \in \Gamma(G^n_{\cup_{i=1}^K f_i})$, such that user $i \in [K]$ computes the non-componentwise function $\hat{f}_i^n \triangleq F_i(U^n, \mathcal{S}_i^n)$, and the condition $\mathbb{P}(\hat{f}_i^n \neq f_i^n) < \epsilon$ is ensured asymptotically for some $\epsilon > 0$, rendering the condition in (4.1) true for B.

Hence, generalizing (4.2) to the broadcast setting, and using Slepian-Wolf coding [82], the broadcast rate, defined by the mutual information between $\mathbf{X}_{[N]}$ and the MISs of the broadcast graph $G_{\cup_{i=1}^K f_i}$, an achievable rate is given by

$$R_B \leq \min_{\mathbf{X}_{[N]} \in U \in \Gamma(G_{\cup_{i'=1}^K f_{i'}})} \max_{i \in [K]} I(\mathbf{X}_{[N]}; U \mid \mathcal{S}_i) \ , \tag{4.4}$$

where the minimization is over all $P_{U,\mathbf{X}_{[N]}}(u, \mathbf{x}_{[N]}) > 0$ such that $U \in \Gamma(G_{\cup_{i=1}^K f_i})$ and (4.1) is ensured[5]. Since U is a function of $\mathbf{X}_{[N]}$, the following substitution holds in (4.4):

$$I(\mathbf{X}_{[N]}; U \mid \mathcal{S}_i) = H_q(U \mid \mathcal{S}_i) \ . \tag{4.5}$$

Thus, we can also present the rate in (4.4) using (4.5). We next demonstrate our coding scheme through examples.

4.3 3-user Computation Broadcast: Examples

Next, we focus on a 3-user computation broadcast model, with $N = 3$ datasets $\mathbf{X}_{[3]}$, where each X_j for $j \in [3]$ is i.i.d. and uniformly distributed over $\mathbb{F}_q$. Each user $i \in [3]$ has side information $\mathcal{S}_i = X_i$ and requests a distinct function $f_i = f_i(X_1, X_2, X_3)$. The former example examines Boolean (non-linear) function demands, while the latter focuses on linear demands.

Example 4.1 (**3-user Boolean computation broadcast**). *We denote user requests by $f_1 = X_1 \wedge X_2 \wedge X_3$, $f_2 = X_1 \vee X_2 \vee X_3$, and $f_3 = X_1 \wedge (X_2 \vee X_3)$, where $X_j \in \mathbb{F}_2$ for $j \in [3]$, $\mathcal{S}_i = X_i$ for $i \in [3]$.*

The characteristic graph $G_{f_1} = (\mathcal{V}, \mathcal{E}_1)$ given $\mathcal{S}_1 = X_1$ is shown in Figure 4.2 (Left), where $\mathcal{V} = \mathbb{F}_2^3$. For $X_1 = 0$, the outcome of $f_1 = 0$ for any (X_2, X_3) pair. Thus, in G_{f_1}, we don't need to distinguish the vertices with $X_1 = 0$. When $X_1 = 1$, $f_1 = 1$ if and only if $X_2 = X_3 = 1$, and otherwise $f_1 = 0$. Thus, $\mathcal{E}_1 = \{(111, 100), (111, 101), (111, 110)\}$. Similarly, we construct G_{f_2} and G_{f_3}, as shown in Figure 4.2 (Middle left) and (Middle right), respectively. Using Definition 4.2, $G_{\cup_{i=1}^3 f_i}$ is shown in Figure 4.2 (Right).

Given $G_{\cup_{i=1}^3 f_i}$ that captures all user demands and side information $\{\mathcal{S}_i\}_{i \in [3]}$, the rate

[4]Theorem 4.1 in Section 4.4 details the binning strategy that achieves R_{ach}.

[5]The right hand side in (4.4) is achievable by assigning colors to $G^n_{\cup_{i=1}^K f_i}$ which achieves the entropy bound of the coloring random variable [72, 77].

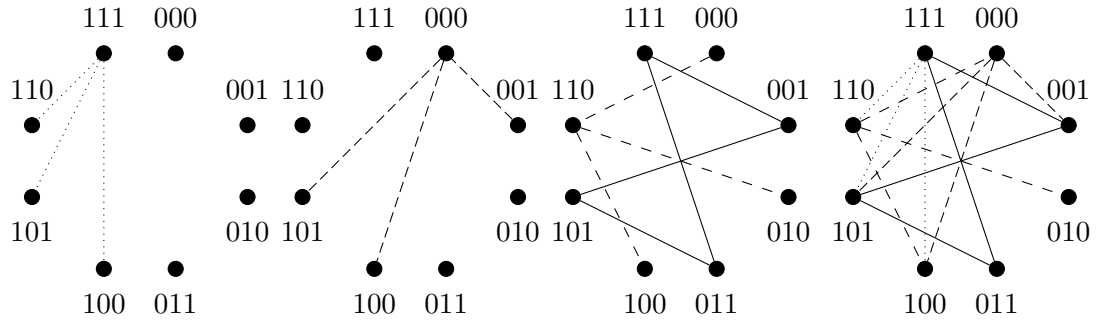

Figure 4.2: Individual characteristic graphs and broadcast graph for Example 4.1: (Left) G_{f_1}, (Middle left) G_{f_2}, and (Middle right) G_{f_3}, based on users' demands f_1, f_2, and f_3, and the side information $\mathcal{S}_i$ for each $i \in [3]$, respectively, and (Right) broadcast graph $G_{\cup_{i=1}^3 f_i}$.

of the broadcast message, employing (4.4) and (4.5), is given by

$$R_B \leq \min_{\mathbf{X}_{[3]} \in U \in \Gamma(G_{\cup_{i=1}^3 f_i})} \max_{i \in [3]} H(U \mid X_i) \ , \tag{4.6}$$

where $G_{\cup_{i=1}^3 f_i}$ *has the following set of MISs:*

$$\Gamma(G_{\cup_{i=1}^3 f_i}) = \left\{ \begin{array}{l} \{000, 010, 011\}, \{001, 110, 011\}, \\ \{001, 010, 011, 100\}, \{101, 110\}, \\ \{010, 100, 101\}, \{000, 010, 111\} \end{array} \right\}. \tag{4.7}$$

The joint PMF $P_{U,\mathbf{X}_{[3]}}$ *that minimizes (4.6) is given in Table 4.1, using which (4.6) yields* $R_B \leq 1.5$ *bits.*

$U \setminus \mathbf{X}_{[3]}$	000	001	010	011	100	101	110	111
$\{000, 010, 111\}$	$\frac{1}{8}$	0	0	0	0	0	0	$\frac{1}{8}$
$\{001, 010, 011, 100\}$	0	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	0	0	0
$\{101, 110\}$	0	0	0	0	0	$\frac{1}{8}$	$\frac{1}{8}$	0

Table 4.1: Joint PMF $P_{U,\mathbf{X}_{[3]}}$ that solves (4.6) for Example 4.1.

Sending colorings of sufficiently large product graphs of $G_{\cup_{i=1}^3 f_i}$ *followed by Slepian-Wolf coding leads to an achievable coding scheme. Hence, from (4.7) and Table 4.1, the minimum entropy coloring is possible using a coloring random variable* $\mathcal{C}_{G_{\cup_{i=1}^3 f_i}} \in \{R, B, G\}$ *that satisfies* $\mathbb{P}(\mathcal{C}_{G_{\cup_{i=1}^3 f_i}} = R) = \mathbb{P}(U = \{000, 111\})$, $\mathbb{P}(\mathcal{C}_{G_{\cup_{i=1}^3 f_i}} = B) = \mathbb{P}(U = \{001, 010, 011, 100\})$, *and* $\mathbb{P}(\mathcal{C}_{G_{\cup_{i=1}^3 f_i}} = G) = \mathbb{P}(U = \{101, 110\})$, *leading the PMF* $\mathbb{P}(R) = \frac{2}{8}$, $\mathbb{P}(B) = \frac{4}{8}$, $\mathbb{P}(G) = \frac{2}{8}$. *Hence, the coloring-based encoding gives an achievable rate of* $R_{ach} \leq H(\mathcal{C}_{G_{\cup_{i=1}^3 f_i}}) = 1.5$ *bits.*

Example 4.1 demonstrates that designing a broadcast message leveraging the structure of all demands and users' side information achieves a rate of $R_{ach} = 1.5$ bits. In contrast,

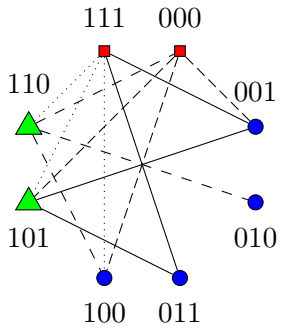

Figure 4.3: The color encoded union graph $G_{\cup_{i=1}^{3} f_i}$ in Example 4.1, where green triangles, red rectangles, and blue circles represent different encoding classes, respectively.

using $\mathcal{S}_i = X_i$ at $i \in [3]$ and applying the Slepian-Wolf theorem [82], sending $(X_1 \oplus_2 X_2, X_2 \oplus_2 X_3)$ achieves the minimum rate for asymptotically lossless reconstruction at each user, as $H(X_1 \oplus_2 X_2, X_2 \oplus_2 X_3) = 2$ bits [82] for all $i \in [3]$, where $\oplus_q$ denotes modulo-q sum. Thus, our graph-based approach achieves a 25% rate reduction versus [82].

Next, we detail a linear computation broadcast example from [1] and show that our approach outperforms that of [1].

Example 4.2 (**3-user linear computation broadcast [1, Example 2]**). *We denote user requests by* $f_1 = X_2 \oplus_3 X_3$, $f_2 = X_1 \oplus_3 X_3$, *and* $f_3 = X_1 \oplus_3 2X_2$, *respectively, where* $X_j \in \mathbb{F}_3$ *for* $j \in [3]$, *and* $\mathcal{S}_i = X_i$ *for* $i \in [3]$.

For the proposed setup, the rate required to encode $G_{\cup_{i=1}^{3} f_i}$, *i.e.,* R_B, *is upper bounded by the solution of (4.4). However, the broadcast graph* $G_{\cup_{i=1}^{K} f_i}$ *has* $|\mathcal{V}| = q^N$ *vertices, rendering the analytical solution for* $\Gamma(G_{\cup_{i=1}^{K} f_i})$ *and hence for (4.4) infeasible for large* q^N, *as compared to Example 4.1 with fewer vertices*[6]. *However, using the sub-additivity properties of graphs, we can divide the solution space and address less complex tasks. To that end, we propose an alternative approach for designing* B *by considering pairs of users and defining compatible functions for each pair, which we detail below.*

Compatibility function-based approach: *To build our achievability scheme, we consider any pair* (i_1, i_2) *of users* $i_1, i_2 \in [3]$ *with respective demands* f_{i_1} *and* f_{i_2}, *and with* $\mathcal{S}_{i_1} = X_{i_1}$ *and* $\mathcal{S}_{i_2} = X_{i_2}$, *respectively. The rate required to encode* $G_{f_{i_1} \cup f_{i_2}}$, *denoted as* $R_{i_1 i_2}$, *must satisfy*

$$R_{i_1 i_2} \leq \min_{\mathbf{X}_{[3]} \in U \in \Gamma(G_{f_{i_1} \cup f_{i_2}})} \max_{i \in \{i_1, i_2\}} \{H_q(U \mid X_i)\} \ ,$$

where instead of explicitly determining $\Gamma(G_{f_{i_1} \cup f_{i_2}})$, *we use a compatibility-based approach and demonstrate its optimality for reconstructing* f_{i_1} *and* f_{i_2}. *A given function* $Z_{i_1 i_2}$ *is called* a compatible function *for pair of users* (i_1, i_2), *if it yields* $\phi_{i_1}(Z_{i_1 i_2}, X_{i_1}) = f_{i_1}$ *and* $\phi_{i_2}(Z_{i_1 i_2}, X_{i_2}) = f_{i_2}$. *For users* $(1, 2)$, *with demands given above, a compatible function is* $Z_{12} = X_1 \oplus_3 X_2 \oplus_3 X_3$. *Similarly, for user pairs* $(1, 3)$, *and* $(2, 3)$, *functions* $Z_{13} =$

[6]The MATLAB code for finding MISs in Example 4.2 is available at [253].

$X_1 \oplus_3 2X_2 \oplus_3 2X_3$ *and* $Z_{23} = X_1 \oplus_3 2X_2 \oplus_3 X_3$*, respectively, are compatible. For any given* (i_1, i_2)*, the rate achieved by* $Z_{i_1 i_2}$ *satisfies*

$$R_{i_1 i_2} \leq H_3(Z_{i_1 i_2}) = H_3(f_{i_1} \mid X_{i_1}) = 1 \ \ bit, \quad i_1, i_2 \in [3] \ .$$

We split a dataset X_j *into two disjoint sub-variables* X_j^1 *and* X_j^2*, each with a size of* 0.5 *bits, similar to [1, Example 2]. Let* $Z_{i_1 i_2}^l$*, for* $l \in [2]$*, denote the compatible function sub-variables* X_j^l*, for* $l \in [2]$*,* $j \in [3]$*, with* $H_3(Z_{i_1 i_2}^l) = 0.5$ *bits. We consider three broadcast messages as follows: (i)* Z_{12}^1*, (ii)* Z_{23}^2*, and (iii)* $Z_{13}^1 \oplus_3 Z_{13}^2$*. From (i) and (iii),* f_1 *is decoded; using (i) and (ii)* f_2 *is decoded; and finally,* f_3 *is decoded given (ii) and (iii). Thus,* R_B *for broadcasting (i)-(iii), exploiting the independence bound on entropy, satisfies*

$$R_B \leq H_3(Z_{12}^1, Z_{23}^2, Z_{13}^1 \oplus_3 Z_{13}^2) \leq H_3(Z_{12}^1) + H_3(Z_{23}^2) + H_3(Z_{13}^1 \oplus_3 Z_{13}^2) = 1.5 \ \ bits.$$

Given a length n *sequence* X_j^n*, we denote two disjoint splits of* X_j^n *by* $X_j^{\frac{n}{2}} = \{X_{j,1}, \ldots, X_{j,\frac{n}{2}}\}$ *and* $X_j^{\frac{n}{2}+1:n} = \{X_{j,\frac{n}{2}+1}, \ldots, X_{j,n}\}$ *two sub-sequences of length* $\frac{n}{2}$*. Combining the idea of splitting with the above-mentioned compatible function construction enables the asymptotically lossless reconstruction of* $\{f_i^n : i \in [3]\}$ *for sufficiently large* n*.*

Vector-based approach: *We next propose a vector-based achievability scheme, drawing on the notion of characteristic graphs, that involves designing two messages,* B_1 *and* $\mathbf{B}_2$*, derived from* $U \in \Gamma(G_{\cup_{i=1}^3 f_i})$*. When* $X_2 = 0$*, with a probability* $1/3$*, the MISs are determined by the function* $X_1 \oplus_3 X_3$ *and encode these MISs to the message* B_1 *with a rate* $H_3(B_1) = H_3(X_1 \oplus_3 X_3) = 1$*. Similarly, when* $X_2 \in [2]$*, with a probability* $2/3$*, the MISs are determined by* $X_1 \oplus_3 X_2 \oplus_3 X_3$ *and* X_2*. These MISs are encoded to message* $\mathbf{B}_2$ *with a rate of* $H_3(\mathbf{B}_2) = H_3(X_1 \oplus_3 X_2 \oplus_3 X_3, X_2) = 1.63$ *bits. Thus, the rate required to broadcast these messages is*

$$R_M \leq \frac{1}{3} H_3(B_1) + \frac{2}{3} H_3(\mathbf{B}_2) = 1.42 \ bits \ , \tag{4.8}$$

compared to the achievable rate of 1.5 *bits in [1, Example 2].*

Example 4.2 presents two approaches for linear function demands: a compatibility function-based technique (similar to [1]) and a vector-based technique. The performance of the former approach is similar to the linear coding scheme of [1, Example 2] with a rate of 1.5 bits. The latter showed that by carefully designing the broadcast message based on the nature of the demands and the datasets, we can achieve $R_B \leq 1.42$ bits, indicating an improvement over the compatibility function-based approach that completely relies on the linear encoding of the demands. This shows that a vector-based approach, refining the characteristic graph-based broadcast message design, can surpass the state of the art. However, achieving the minimum rate heavily depends on the structure of the demanded

function and lacks a generalizable form.

The next section presents our achievable and converse rates for the K-user computation broadcast model.

4.4 K-User Computation Broadcast: Achievability and Converse Bounds

This section presents an upper bound R_{ach} and a lower bound R_{con} on the communication rate for the K-user computation broadcast model, as shown in Figure 4.1.

Theorem 4.1. *Given a K-user computation broadcast model, with N datasets $\mathbf{X}_{[N]}$, and a broadcast graph $G_{\cup_{i=1}^{K} f_i}$ the communication rate for the broadcast message B satisfies:*

$$R_{ach} = \min_{\mathbf{X}_{[N]} \in U \in \Gamma(G_{\cup_{i'=1}^{K} f_{i'}})} \max_{i \in [K]} I(\mathbf{X}_{[N]}; U \mid \mathcal{S}_i) \ , \tag{4.9}$$

$$R_{con} = \min_{P_{\Upsilon, \mathbf{X}_{[N]}}(\upsilon | \mathbf{x}_{[N]})} \max_{i \in [K]} I(\mathbf{X}_{[N]}; \Upsilon \mid \mathcal{S}_i) \ , \tag{4.10}$$

where $G_{\cup_{i=1}^{K} f_i}$ is the corresponding broadcast graph, and $\Gamma(G_{\cup_{i=1}^{K} f_i})$ is the set of MISs of $G_{\cup_{i=1}^{K} f_i}$. Furthermore, we denote by U^* the MIS that achieves R_{ach}:

$$U^* = \arg \min_{\mathbf{X}_{[N]} \in U \in \Gamma(G_{\cup_{i'=1}^{K} f_{i'}})} \max_{i \in [K]} I(\mathbf{X}_{[N]}; U \mid \mathcal{S}_i) \ . \tag{4.11}$$

Proof. See Appendix C.1. □

We next introduce a converse bound, different from R_{con} in Theorem 4.1, derived using the joint PMF of each of the demanded functions and the available side information.

Proposition 4.1. *Given a $(K, N, \{\mathcal{S}_i\}_{i \in [K]}, \{f_i\}_{i \in [K]})$ non-linear function computation broadcast model, a lower bound on the communication rate required to satisfy all users' demands, given side information $\mathcal{S}_i$ is*

$$H_q(f'_1, f'_2, \ldots, f'_K) \ , \tag{4.12}$$

where $f'_i = f_i | \mathcal{S}_i$ is a conditional random variable capturing the demand f_i of user $i \in [K]$ given $\mathcal{S}_i$, where $f'_i \in \mathbb{F}_q$.

Proof. See Appendix C.2. □

For comparison, we generalize the genie-aided lower bound in [254] to our model, as detailed next.

We next exploit the genie-aided approach introduced in [254] and later used in [9] to provide a lower bound on the sum rate for a $(2, 2, \{X_1, X_2\}, \{f_1, f_2\})$ non-linear function computation broadcast model. For this model, consider user 1 with side information $\mathcal{S}_1 =$

X_1 that needs $H_q(f_1 \mid X_1)$ bits to recover f_1. Once user 1 recovers f_1, a genie provides all information known to user 1, i.e., $(f_1, \mathcal{S}_1)$, to user 2. User 2, with side information $\mathcal{S}_2$ and the additional information provided by the genie, then requires $H_q(f_2|\mathcal{S}_1, \mathcal{S}_2, f_1)$ additional bits to recover their requested function f_2. Hence, the sender node needs to broadcast extra $H_q(f_2 \mid X_1, X_2, f_1)$ bits for user 2 to recover function f_2. Swapping the users and applying the same method, it is clear that without the help of a genie, we require a higher broadcast rate. As a result, a lower bound on the required sum rate is given as [254]:

$$R_{Genie} \geq H_q(f_1 \mid X_1) + H_q(f_2 \mid X_2, X_1, f_1) \ . \tag{4.13}$$

We next generalize the genie-aided lower bound on the sum rate devised for the $K = 2$-user case using $N = 2$ datasets in [254], [94] to our K-user model and compare it to the lower bound in Proposition 4.1. To that end, let $\Pi([K])$ denote the set of all permutations of the set of user indices $[K]$. The lower bound on the sum rate is given by the maximum over all such rates across possible permutations. This is formalized by the following result.

Proposition 4.2. *Let $(K, N, \{\mathcal{S}_i\}_{i\in[K]}, \{f_i\}_{i\in[K]})$ be a non-linear function computation broadcast model. For a given user permutation $\pi \in \Pi([K])$, we assume that a genie reveals to each user $\pi(i)$, for $i \geq 2$ the following:*

- *the side information sets $\{\mathcal{S}_{\pi(j)}\}_{\forall j\in\{1,2,\ldots,i-1\}}$, and*
- *the demanded functions $\{f_{\pi(j)}\}_{\forall j\in\{1,2,\ldots,i-1\}}$ decoded by users $\pi(1), \ldots, \pi(i-1)$.*

Then, a lower bound on the broadcast rate is given by:

$$R_{Genie} = \max_{\pi\in\Pi([K])} \left[H_q(f_{\pi(1)} \mid \mathcal{S}_{\pi(1)}) + \sum_{i=2}^{K} H_q\big(f_{\pi(i)} \mid \{f_{\pi(j)}, \mathcal{S}_{\pi(j)}\}_{\forall j\in\{1,2,\ldots,i-1\}},\ \mathcal{S}_{\pi(i)}\big) \right]. \tag{4.14}$$

Proof. Given a user permutation π, the broadcast rate lower bound in (4.14) is achieved using a genie-aided scheme, where each user $\pi(i) \in [K]$ decodes $f_{\pi(i)}$ using their own side information $\mathcal{S}_{\pi(i)}$ and the data $\bigcup_{j\leq i} \mathcal{S}_{\pi(j)}$ of all preceding users, according to the permutation π.

With this genie-aided scheme, for a given $i \in [K]$, user $\pi(1)$ needs $H_q(f_{\pi(1)} \mid \mathcal{S}_{\pi(1)})$ bits, and user $\pi(i)$, for $i \geq 2$, needs $H_q\big(f_{\pi(i)} \mid \{f_{\pi(j)}, \mathcal{S}_{\pi(j)}\}_{\forall j\in\{1,2,\ldots,i-1\}},\ \mathcal{S}_{\pi(i)}\big)$ bits to recover their requested function, respectively. Hence, for the given permutation, the sender needs a total broadcast rate of

$$H_q(f_{\pi(1)} \mid \mathcal{S}_{\pi(1)}) + \sum_{i=2}^{K} H_q\Big(f_{\pi(i)} \mid \{f_{\pi(j)}, \mathcal{S}_{\pi(j)}\}_{\forall j\in\{1,2,\ldots,i-1\}},\ \mathcal{S}_{\pi(i)}\Big) \ \text{ bits.} \tag{4.15}$$

The final result follows from considering all possible permutations $\Pi([K])$. □

While the approach in Proposition 4.1 satisfies the demands of K users simultaneously by generating a broadcast message without allowing any interaction between the K users,

Proposition 4.2 incorporates such interactions via extending the genie-aided approach devised for two users in [254], to the K user setting. To that end, we employ a successive refinement-based method for recovering the demands, which requires users to share side information as detailed in Proposition 4.2 that yields the rate lower bound in (4.14).

To demonstrate the applicability of the lower bound in (4.14), consider the 3-user Boolean computation broadcast problem in Example 4.1. The lower bound, using (4.14), is given as:

$$\begin{aligned} R_{Genie} \geq & H(f_3|X_3) + H(f_1|X_1, X_3, f_3) \\ & + H(f_2|X_2, X_1, X_3, f_1, f_3) \\ = & 0.905 + 0.25 + 0 = 1.155 \text{ bits} . \end{aligned} \tag{4.16}$$

The application of the lower bound (4.12) in Proposition 4.1 to the 3-user Boolean computation model of Example 4.1 yields 1.45 bits, while the achievable rate, from (4.6), is $R_{ach} = 1.5$ bits, indicating a tight gap between the bounds. This comparison shows that (4.12) in Proposition 4.1 yields a tighter lower bound than $R_{Genie} \geq 1.155$ bits in (4.14) in Proposition 4.2, which is a generalization of the genie-aided lower bound in [254], for serving this type of non-linear demands.

4.5 Conclusion

In this chapter, we examined the K-user computation broadcast model with general data distributions and general function demands over finite fields. Using the characteristic graph framework, we proposed a novel broadcast coding scheme that reduces the communication rate while accommodating both linear and non-linear demands compared to the state of the art. We provided examples of the 3-user model under different strategies (different demand, different cache sizes, skewed data). Moreover, we presented achievable and converse rates for the general K-user computation broadcast problem, while more research is needed to quantify the gap between the bounds.

Chapter 5

Conclusions and Future Directions

In this thesis, we addressed three core problems in several distributed computation settings, from the perspective of communication costs: (i) lossless functional compression with two or more distributed sources for different classes of characteristic graphs, (ii) a multi-server multi-function distributed computation setting where a single user demands several multivariate (potentially non-linear) functions of distributed datasets, and (iii) a computation broadcast problem, where a master node serves multiple users with possibly non-linear function demands, each holding a subset of the datasets as side information. All problems were studied in the asymptotically lossless regime using the characteristic graph framework, which captures distinguishability requirements imposed by the function structure and the dataset statistics. In the following, we summarize the contributions and the future research directions for each technical chapter of this thesis.

5.1 Summary of Contributions, and Future Research Directions for Chapter 2

In the first technical chapter of the thesis, we presented preliminaries on characteristic graphs and lossless function compression, laying the theoretical foundation for the rest of the thesis. We then introduced new coloring-based encoding schemes for source characteristic graphs, analyzing topologies such as cycles (C_i), d-regular graphs ($G_{d,V}$), and general graphs (G). For each graph class, we examined their n-fold OR product extensions — C_i^n, $G_{d,V}^n$, and G^n — and analyzed their chromatic numbers and spectral properties. For cycles, we developed a polynomial-time recursive coloring method for C_{2k+1}^n and derived bounds on $\chi(C_i^n)$ via spectral analysis. For d-regular and general graphs, we studied graph expansion properties and derived eigenvalue bounds of the adjacency matrix by using the GCT method [98]. These tools have enabled the derivation of tight bounds on $H_{G^n}^{\chi}(\mathbf{X}_1)$ and the design of polynomial-time coloring strategies for asymptotically lossless functional compression. Together, these results show *how the structure of the functions, captured via characteristic graphs and their spectral properties, fundamentally controls the*

limits of distributed functional compression.

Future research can extend distributed functional compression by exploring alternative properties of characteristic graphs beyond chromatic number and entropy, such as graph decomposition and graph complements, to derive new converses. Another promising direction is the scenario of asymptotically lossy functional compression, where rate-distortion trade-offs can be studied using characteristic graphs. A potential direction is to extend our coloring approach to random graphs and randomly generated function demands. Such models are relevant when network topologies or computation tasks vary according to a distribution. In this setting, tools from probabilistic graph theory, such as spectral gap[1] estimates and random walk mixing times[2], may provide tractable approximations for achievable rates in stochastic settings. Finally, directed graphs, which arise in several contexts — knowledge graphs in LLM modeling and asymmetric adjacency-matrix models used in blockchain privacy applications — may be of interest for further investigation.

5.2 Summary of Contributions, and Future Research Directions for Chapter 3

In this chapter, we designed a distributed computation framework for general function classes in multi-server, single-user networks where the user demands multivariate functions of datasets distributed across servers. Using Körner's characteristic graph entropy, we developed upper bounds on the communication cost for distributed computation, taking into account dataset correlations, and the structure of demanded functions (e.g., linear, bilinear, or non-linear). Through experiments involving cyclic dataset placement over a field ($\mathbb{F}_2$), we demonstrated that our framework can achieve low communication costs for both linearly separable and multi-linear functions. *The main theoretical results confirmed that exploiting dataset skew, correlations, and the structure of the user's demanded function significantly reduces the required communication rate.*

Several promising directions follow from our work on the multi-server multi-function setting. A first step is to tighten the achievability bound of Theorem 3.1 and to derive new converses. Another important direction is to generalize our uncoded placement model to coded placement strategies (see e.g., [28]) across servers, to improve the rate–memory tradeoff. Extending the analysis to multi-shot and temporally correlated sources is also critical, since real systems often process streams of datasets with high temporal correlations, such as sensor logs or time-varying federated datasets.

Beyond the specific cases analyzed in Propositions 3.1–3.3, the methodology can be generalized to other distributed computation scenarios. For instance, over-the-air com-

[1] The spectral gap of a graph is the difference between the largest and second-largest eigenvalues of its adjacency (or transition) matrix.

[2] The mixing time of a random walk is the number of steps required for its distribution to become close to the stationary distribution.

putation in wireless networks illustrates how function-specific aggregation interacts with channel constraints. Similarly, cluster and cloud computing frameworks highlight the role of dataset placement and communication bottlenecks. Coded computing, in turn, provides fault tolerance under adversarial stragglers. Finally, distributed gradient descent offers another application, as it reflects linearly separable multi-function computation. Embedding such task-aware compression into large-scale distributed systems further strengthens the connection between information-theoretic analysis and the design of practical computing and learning architectures.

5.3 Summary of Contributions, and Future Research Directions for Chapter 4

In the third part of this thesis, we examined the K-user computation broadcast problem over finite fields, consisting of a sender node with access to N datasets, and K user nodes where each user demands a (possibly non-linear) multivariate function of the datasets and has access to a subset of datasets as side information. We assume that the master node has the computational capabilities to devise a common broadcast message to simultaneously meet the user demands. Using the characteristic graph framework, we proposed a novel broadcast coding scheme that outperforms existing methods for both linear function demands, which have been the primary focus of prior works [1, 94, 239], and non-linear function demands, which were previously studied only in restricted two-user or complementary-demand settings [1, 95]. *Our framework extends these results to general, possibly non-linear computations under general dataset distributions, establishing new achievable bounds for function computation broadcast settings. In addition, the presented converse bounds demonstrate the tightness of the achievability results in several interesting non-linear computation broadcast scenarios.*

Future directions for the computation broadcast setting include designing intelligent dataset placement strategies at the user sites, where the side information may itself be a function of the datasets and users have heterogeneous storage capacities. Another direction is to relate the computation broadcast model to the Gray-Wyner[3] setting [244]. In particular, one can investigate the impact of splitting information into a *common part*, i.e., the broadcast message sent to all users, and *private parts*, i.e., individual messages tailored to the specific demands of each user, on the overall communication cost and its optimization, generalizing the tradeoff space for the computation broadcast model. Further directions include extending index coding techniques [255] to the computation broadcast setting, e.g., through MinRank-based methods, which can help achieve optimal communication costs. Beyond these, a natural continuation is to derive tighter bounds between our achievability and converse results, which would refine our knowledge of the

[3]Gray-Wyner channel models the problem of jointly compressing a pair of correlated sources into a common message and two private messages for efficient distribution [244].

fundamental limits of computation broadcast.

Summary of this Dissertation

In summary, this thesis has exploited a graph-theoretic source compression framework for the analysis of distributed function computation across several multi-user network models. By linking the structural properties of characteristic graphs to fundamental communication limits, it has offered both novel theoretical insights and practical tools for the design of achievable coding strategies — such as coloring-based schemes for cyclic, regular, and general characteristic graphs, entropy-based bounds for compressing general graphs, and broadcast coding methods for multi-user computation. These findings, together with the outlined future directions, are expected to advance distributed computation systems by reducing communication costs using source coding to compress functions, employing characteristic graphs to capture the structures of demanded computations in multi-server networks as well as in computation broadcast networks.

5.4 Publications

1. M.R. Deylam Salehi, and D. Malak, "An achievable low complexity encoding scheme for coloring cyclic graphs," in Proc., IEEE Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, Sep. 2023, pp. 1-8. (**IEEE**)
2. D. Malak, M.R. Deylam Salehi, B. Serbetci, and P. Elia, "Multi-Server Multi-Function Distributed Computation." Entropy, vol. 26, no. 6, p. 448, May. 2024. (**Entropy**)
3. D. Malak, M.R. Deylam Salehi, B. Serbetci, and P. Elia, "Multi-Functional Distributed Computing." in Proc., IEEE Conference on Communication, Control, and Computing (Allerton), Urbana, IL, USA, Sep. 2024, pp. 1-8. (**IEEE**)
4. M.R. Deylam Salehi, V. K. Kizhakke Purakkal, and D. Malak, "Non-Linear Function Computation Broadcast." in Proc., IEEE International Symposium on Information Theory (ISIT), Ann Arbor, MI, USA, Jun. 2025, pp. 1-6. (**arXiv**, **IEEE**)
5. M.R. Deylam Salehi, and D. Malak, "Graph-Theoretic Limits of Distributed Computation: Entropy, Eigenvalues, and Chromatic Numbers." Entropy, vol. 27, no. 7, p. 757, Jul. 2025. (**Entropy**)

Poster Presentations

1. A. Tanha, M.R. Deylam Salehi, and D. Malak, "Structured coded matrix multiplication.", IEEE Communication Theory Workshop (CTW), Venezia, Italy, 4–7 May. 2025. (**Link**)
2. M.R. Deylam Salehi, A. Tanha, and D. Malak, "Structured polynomial codes.", *Recent Results Session in IEEE International Symposium on Information Theory (ISIT)*, Athens, Greece, 7–12 Jul. 2024. (**Link**)
3. M.R. Deylam Salehi, and D. Malak, "Coloring of cyclic graphs.", *6GWFF 2023, 6G Wireless Foundations Forum*, Sophia Antipolis, France, 10–11 Jul. 2023. (**Link**)

Appendices

Appendix A

Appendix for Chapter 2

A.1 Proof of Proposition 2.1

Consider $C_i = G(\mathcal{V}, \mathcal{E})$, where for each $x \in [V]$, $deg(x) = 2$ (see Definition 2.7). The 2-fold OR product is denoted by C_i^2, and it has V sub-graphs $\{C_i^2(1), C_i^2(2), \ldots, C_i^2(V)\}$. For a given sub-graph, say $C_i^2(l)$, $l \in [V]$, any $x^2(l) \in C_i^2(l)$, where $x^2(l)$ denotes a vertex in the l-th replica of C_i^2, is connected to any vertex of the adjacent sub-graphs, namely $\in \{C_i^2((l-1) \mod (V)), C_i^2((l+1) \mod (V))\}$. Therefore, each vertex in $C_i^2(l)$ has V edges to $C_i^2((l-1) \mod (V))$ and $C_i^2((l+1) \mod (V))$ each. Accounting for the set of edges between adjacent sub-graphs, and each vertex's degree in the sub-graph itself, the degree of each vertex in C_i^2 is $deg(x^2) = 2 + 2V$. For $n = 3$, this results in

$$deg(x^3) = 2 + 2V + 2V^2 \ . \tag{A.1}$$

For a given $C_i^{n-1}(l)$, each vertex $x^{n-1}(l)$ is connected to all vertices in the adjacent sub-graphs, $C_i^{n-1}((l-1) \mod (V))$ and $C_i^{n-1}((l+1) \mod (V))$. Similarly, the degree for the $(n-1)$-fold OR product, by considering the previous OR product's degree and using induction, is going to be

$$deg(x^{n-1}) = 2 + 2V + 2V^2 + \cdots + 2V^{n-2} = 2 + \sum_{j=1}^{n-2} 2V^j \ . \tag{A.2}$$

Therefore, for building C_i^n, using the same procedure, there are V sub-graphs $\{C_i^n(1)$, $C_i^n(2)$ $, \ldots, C_i^n(V)\}$. Similar to C_i^2, the sub-graph $C_i^n(l)$, $l \in [V]$, is fully connected to the adjacent sub-graphs $\{C_i^n((l-1) \mod (V)), C_i^n((l+1) \mod (V))\}$. Therefore, using induction and the geometric series $\sum_{j=1}^{n-1} V^j$, we can determine $deg(x^n)$ in closed form using (2.13).

A.2 Proof of Proposition 2.3

Proof. We focus on odd cycles C_i, $i = 2k + 1$, $k \geq 2$. The key novelty of our approach is that, instead of coloring the entire OR-product graph at once, we introduce a *sub-graph–based coloring scheme* that assigns *sets of colors* to sub-graphs. This viewpoint naturally lends itself to a coding interpretation and enables efficient constructions that are applicable not only asymptotically but also for *finite blocklength* functional compression.

In particular, each n-fold OR product C_i^n is decomposed into i sub-graphs $\{C_i^n(l)\}_{l\in[i]}$, where each sub-graph is isomorphic to C_i^{n-1}. Rather than coloring C_i^n directly, we color each sub-graph using a *coloring set* $\mathcal{C}_i^n(l)$, with the constraint that adjacent sub-graphs must be assigned disjoint color sets. This guarantees a valid coloring of the entire graph while minimizing the total number of colors.

We illustrate the construction using C_5. Since $\chi(C_5) = 3$, a valid coloring of C_5 requires three colors, denoted by $\{c_1, c_2, c_3\}$. For the 2-fold OR product C_5^2, a greedy approach would require $(\chi(C_5))^2 = 9$ colors. However, by assigning colors at the sub-graph level, we achieve $\chi(C_5^2) = 8$, as shown in Figure 2.3, saving one color.

Specifically, C_5^2 consists of 5 sub-graphs, each isomorphic to C_5. The first two sub-graphs must be assigned disjoint color sets, since they are fully connected. Thus, we assign

$$\mathcal{C}_5^2(1) = \{c_1, c_2, c_3\}, \qquad \mathcal{C}_5^2(2) = \{c_4, c_5, c_6\}.$$

Because there is no edge between sub-graphs $C_5^2(1)$ and $C_5^2(3)$, we can reuse colors from $\mathcal{C}_5^2(1)$ when coloring $C_5^2(3)$, introducing only two new colors. Proceeding cyclically, we obtain

$$\begin{aligned}
&\mathcal{C}_5^2(1) = \{c_1, c_2, c_3\}, \quad \mathcal{C}_5^2(2) = \{c_4, c_5, c_6\},\\
&\mathcal{C}_5^2(3) = \{c_7, c_8, c_1\}, \quad \mathcal{C}_5^2(4) = \{c_2, c_3, c_4\},\\
&\mathcal{C}_5^2(5) = \{c_5, c_6, c_7\},
\end{aligned}$$

which uses only 8 colors in total.

The same principle extends naturally to higher-order products. For C_5^3, each sub-graph $C_5^3(l)$ is isomorphic to C_5^2 and thus requires $|\mathcal{C}_5^3(l)| = 8$ colors. Assigning disjoint color sets to adjacent sub-graphs and reusing colors whenever possible yields $\chi(C_5^3) = 20$, with coloring sets

$$\begin{aligned}
\mathcal{C}_5^3(1) &= \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\},\\
\mathcal{C}_5^3(2) &= \{c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{16}\},\\
\mathcal{C}_5^3(3) &= \{c_{17}, c_{18}, c_{19}, c_{20}, c_1, c_2, c_3, c_4\},\\
\mathcal{C}_5^3(4) &= \{c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}\},\\
\mathcal{C}_5^3(5) &= \{c_{13}, c_{14}, c_{15}, c_{16}, c_{17}, c_{18}, c_{19}, c_{20}\}.
\end{aligned}$$

This construction highlights the core idea: the *number of sub-graphs remains fixed* and equal to i, while the size of each coloring set grows according to the chromatic number of the previous OR product. This structure enables controlled color reuse and leads to a strictly smaller number of colors than the greedy scheme.

By induction, this yields

$$\chi(C_5) = 3, \quad \chi(C_5^2) = 8, \quad \chi(C_5^3) = 20, \quad \chi(C_5^4) = 50, \quad \chi(C_5^5) = 125, \quad \chi(C_5^6) = 313,$$

and, in general, for odd cycles C_i, $i = 2k+1$, $k \geq 2$,

$$\chi(C_i^{n+1}) = 2\chi(C_i^n) + \left\lceil \frac{\chi(C_i^n)}{2} \right\rceil,$$

which completes the proof. □

A.3 Proof of Proposition 2.4

From (2.15) in Proposition 2.3, for C_{2k+1}^n with $k \geq 2$, and by applying lower and upper bounds to the ceiling function, we infer that

$$2\chi(C_{2k+1}^{n-1}) + \frac{\chi(C_{2k+1}^{n-1})}{2} \leq \chi(C_{2k+1}^n) \leq 2\chi(C_{2k+1}^{n-1}) + \frac{\chi(C_{2k+1}^{n-1})}{2} + 1\,. \tag{A.3}$$

Exploiting the number of colors needed for coloring C_{2k+1}^n using greedy algorithm, which is given as $\big(\chi(C_{2k+1})\big)^n = 3^n$, and (A.3), the gain, η_n, lies in the following interval:

$$\frac{3^n}{2\chi(C_{2k+1}^{n-1}) + \frac{\chi(C_{2k+1}^{n-1})}{2} + 1} \leq \eta_n \leq \frac{3^n}{2\chi(C_{2k+1}^{n-1}) + \frac{\chi(C_{2k+1}^{n-1})}{2}}\ . \tag{A.4}$$

For $n = 1$, we have $\chi(C_{2k+1}) = \chi_{\text{Greedy}}(C_{2k+1}) = 3 = \chi_1$. For $n \geq 2$, $\chi_{\text{Greedy}}(C_{2k+1}^n) = 3^{n-1}\chi_1$, whereas our cyclic approach yields

$$\chi(C_{2k+1}^n) \geq \Big(\frac{5}{2}\Big)^{n-1}\chi_1\ , \quad n \geq 2\,. \tag{A.5}$$

Substituting $\chi(C_{2k+1}^{n-1}) \geq \left(\frac{5}{2}\right)^{n-2}\chi_1$ in the denominator of the upper bound in (A.4), where the ratio of the nominator to the denominator leads to $\eta_n \leq \frac{3^{n-1}}{(\frac{5}{2})^{n-1}}$. Furthermore, using (A.5), the denominator of the lower bound in (A.4) is lower bounded as:

$$2\chi(C_{2k+1}^{n-1}) + \frac{\chi(C_{2k+1}^{n-1})}{2} + 1 \geq 2\Big(\frac{5}{2}\Big)^{n-2}\chi_1 + \frac{1}{2}\Big(\frac{5}{2}\Big)^{n-2}\chi_1 + 1 = \Big(\frac{5}{2}\Big)^{n-1}\chi_1 + 1\ , \tag{A.6}$$

and from (A.6), $\eta_n \geq \frac{3^{n-1}\chi_1}{\left(\frac{5}{2}\right)^{n-1}\chi_1+1}$, and as $n \to \infty$, the lower and upper bounds match.

A.4 Proof of Proposition 2.5

We build our achievable scheme by applying the recursive coloring approach for $\chi(C^n_{2k+1})$ in (2.15) and decomposing C^n_{2k+1} into sub-graphs that cover all vertices of C^n_{2k+1}. This decomposition involves two types of sets: (i) non-overlapping maximum independent sets of each product graph C^t_{2k+1} for $t \in [n]$ by noting that the size of a maximum independent set[1] is $\alpha(C^t_{2k+1}) = k^t$, and (ii) the remaining singleton set because $V^t = (2k+1)^t$ is odd for all $t \in [n]$.

We next derive a coloring PMF for the proposed decomposition of C^n_{2k+1}, given by

$$P(\mathcal{C}_{C^n_{2k+1}}) = P(\zeta_0 \cdot \frac{1}{V^n} \ , \ \zeta_1 \cdot \frac{k}{V^n} \ , \dots \ , \ \zeta_n \cdot \frac{k^n}{V^n}) \ ,$$

with coefficients $\{\zeta_t \ , \ t \in \{0\} \cup [n]\}$ that represent the weights assigned to color each set and satisfy the following constraints:

$$\begin{aligned} &(a) \ \Big(\sum_{t=1}^{n} \zeta_t \cdot \frac{k^t}{V^n} \Big) + \zeta_0 \cdot \frac{1}{V^n} = 1, \\ &(b) \ \zeta_t \geq k \cdot \zeta_{t-1} \ , \\ &(c) \ \zeta_0 = 1 \ , \end{aligned} \tag{A.7}$$

where (a) indicates that the coloring assignment forms a valid PMF. Constraint (b) minimizes the total number of colors required. In constraint (c), $\alpha(C^0_{2k+1}) = k^0 = 1$ represents the additional color needed for the singleton set. Incorporating constraints in (A.7), we have

$$\zeta_n \cdot \frac{k^n}{(2k+1)^n} + \zeta_{n-1} \cdot \frac{k^{n-1}}{(2k+1)^n} + \dots + \zeta_0 \cdot \frac{1}{(2k+1)^n} \geq 1 \ . \tag{A.8}$$

Using (A.8), and the properties of geometric series, we obtain

$$1 = \sum_{t=0}^{n} \frac{\zeta_t k^t}{(2k+1)^n} < \frac{\zeta_n}{(2k+1)^n} \cdot \sum_{t=0}^{n} \frac{k^t}{k^{n-t}} = \frac{\zeta_n}{(2k+1)^n} \cdot \frac{1}{k^n} \cdot \frac{(k^2)^{(n+1)} - 1}{k^2 - 1} \ , \tag{A.9}$$

where the final equality follows from computing the summation, which leads to

$$\zeta_n > \frac{(2k+1)^n k^n (k^2-1)}{k^{2(n+1)} - 1} \ .$$

Next, to derive the upper bound on ζ_n, we assign $\zeta_1 = 0$ and maximize ζ_n. From (A.9), we have

$$1 = \sum_{t=0}^{n} \frac{\zeta_t k^t}{(2k+1)^n} > \frac{\zeta_n}{(2k+1)^n} \cdot \sum_{t=0}^{n} \frac{k^t}{k^{n-t}} \ , \tag{A.10}$$

[1] Given the full connection of sub-graphs in the t-fold OR product C^t_{2k+1}, we deduce that $\alpha(C^t_{2k+1}) = k^t$ for all $t \in [n]$ [256].

where reordering (A.10), we have the following upper bound on ζ_n:

$$\zeta_n < \frac{(k^{2(n+1)}-1)\cdot(2k+1)^n \cdot k^{(n-1)}}{(2k+1)^{2n}\cdot(k^2-1)\cdot k^{(n-1)} - (k^{2(n+1)}-1)} \ . \tag{A.11}$$

Similarly, using (A.11) we can bound ζ_t where $t < n$, providing an upper bound on $H_{C_{2k+1}}(X_1)$, which concludes our proof.

A.5 Proof of Theorem 2.1

Assume $\mathbf{A}_f$ has V eigenvalues. To find the eigenvalues of $\mathbf{A}_f^2$, one must solve (2.28), derived from evaluating (2.27), yielding V equations per block row and V^2 equations in total. Hence, there are V^2 equations, from which the remaining $(V-1)\times V$ equations are just replicas of the eigenvalues of any given block row, due to cyclic symmetry. In (2.28), the terms with indices $k+1$ and $k-1$ are in modulo V. Two additional equations are necessary to calculate $\lambda_k(\mathbf{A}_f^j)$ where $k \in [V^j]$, as the power j increases from $n-1$ to n. Both $\mathbf{A}_f$ and $\mathbf{J}_V$ are diagonalizable, i.e., there exists an invertible matrix $\mathbf{P}$ such that $\mathbf{A}_f = \mathbf{P}^{-1}\mathbf{H}_{\mathbf{A}_f}\mathbf{P}$ and $\mathbf{J}_V = \mathbf{P}^{-1}\mathbf{H}_{\mathbf{J}_V}\mathbf{P}$, where $\mathbf{H}_{\mathbf{A}_f}$ and $\mathbf{H}_{\mathbf{J}_V}$ are diagonal matrices. Hence, the sum $\mathbf{A}_f + \mathbf{J}_V$ satisfies

$$\mathbf{A}_f + \mathbf{J}_V = \mathbf{P}^{-1}\mathbf{H}_{\mathbf{A}_f}\mathbf{P} + \mathbf{P}^{-1}\mathbf{H}_{\mathbf{J}_V}\mathbf{P} = \mathbf{P}^{-1}(\mathbf{H}_{\mathbf{A}_f} + \mathbf{H}_{\mathbf{J}_V})\mathbf{P} \ .$$

The eigenvalues of $\mathbf{A}_f + \mathbf{J}_V$ can now be computed using the diagonal of $\mathbf{H}_{\mathbf{A}_f} + \mathbf{H}_{\mathbf{J}_V}$, where $\lambda_k(\mathbf{A}_f)$ and $\lambda_k(\mathbf{J}_V)$ represent the k-th eigenvalues of $\mathbf{A}_f$ and $\mathbf{J}_V$ matrices for $k \in [V]$, respectively.

$$\lambda_k(\mathbf{A}_f + \mathbf{J}_V) = \lambda_k(\mathbf{A}_f) + \lambda_k(\mathbf{J}_V), \quad k \in [V] \ . \tag{A.12}$$

From Lemma 2.1, the number of distinct eigenvalues of $\mathbf{A}_f^2$ differs from the eigenvalues of $\mathbf{A}_f$ at most by two, and similarly for the eigenvalues of $\mathbf{A}_f^j$, $j \in \mathbb{Z}^{+\geq 2}$ derived from $\mathbf{A}_f^{j-1}$ of the $(j-1)$-fold OR product graph. From (2.26), the sub-matrices $\mathbf{A}_f^{j-1}$ in $\mathbf{A}_f^j$ take the following form

$$\mathbf{A}_f^{j-1}\mathbf{v}_k + \mathbf{J}_{V^{j-1}}\mathbf{v}_{k+1} + \mathbf{J}_{V^{j-1}}\mathbf{v}_{k-1} = \nu\mathbf{v}_k, \quad k \in [V] \ , \tag{A.13}$$

where $j \in \mathbb{Z}^{+\geq 2}$, and the column vector $\mathbf{v}_k$ has dimensions $V^{j-1}\times 1$. For the n-fold OR product, $\mathbf{A}_f^n$, is constructed by combining $\mathbf{A}_f^{n-1}$, $\mathbf{J}_{V^{n-1}}$ from the $(n-1)$-fold OR product, and $\mathbf{Z}_{V^{n-1}}$ matrices, as in (2.26). The first block row of $\mathbf{A}_f^n$ includes $\mathbf{A}_f^{n-1}$, two $\mathbf{J}_{V^{n-1}}$ matrices representing full connections to adjacent sub-graphs, and $\mathbf{Z}_{V^{n-1}}$ matrices indicating no connections, all of which affect eigenvalue computation. Thus, C_i^n has two more distinct eigenvalues than C_i^{n-1}. As a result, using Lemma 2.1, we can prove Theorem 2.1.

A.6 Proof of Proposition 2.8

Given $G_{d,V} = G(\mathcal{V}, \mathcal{E})$, its 2-fold OR product is denoted by $G^2_{d,V}$, consisting of V sub-graphs, $G^2_{d,V} = \{G^2_{d,V}(1), G^2_{d,V}(2), \ldots, G^2_{d,V}(V)\}$. For a given sub-graph $G^2_{d,V}(l)$, where $l \in [V]$, any vertex $x^2(l) \in \mathcal{V}^2$ is connected to all vertices in the following d adjacent sub-graphs:

$$\{G^2_{d,V}((l - \lfloor \frac{d}{2} \rfloor) \bmod (V)), \ldots, G^2_{d,V}((l-1) \bmod (V)), G^2_{d,V}((l+1 \bmod (V)), \\ \ldots, G^2_{d,V}((l + \lceil \frac{d}{2} \rceil) \bmod (V))\} \ . \tag{A.14}$$

Thus, each vertex in $G^2_{d,V}(l)$ has $d \times V$ edges to these adjacent sub-graphs. For $n = 2$, $deg(x^2) = d + dV$, accounting for both edges within sub-graphs and edges to adjacent sub-graphs. For $n = 3$, accounting for edges to adjacent sub-graphs and those within each sub-graph (i.e., $deg(x^2)$) yields $deg(x^3) = d + dV + dV^2$. For a given $G^{n-1}_{d,V}(l)$, each vertex $x^{n-1}(l)$ is connected to all vertices in the adjacent sub-graphs

$$\{G^{n-1}_{d,V}((l - \lfloor \frac{d}{2} \rfloor) \bmod (V)), \ldots, G^{n-1}_{d,V}((l-1) \bmod (V)), G^{n-1}_{d,V}((l+1) \bmod (V)), \\ \ldots, G^{n-1}_{d,V}((l + \lceil \frac{d}{2} \rceil) \bmod (V))\} \ . \tag{A.15}$$

Similarly, for the $(n-1)$-fold OR product, $deg(x^{n-1})$ is calculated by induction, starting from the degree in the previous product, which yields

$$deg(x^{n-1}) = d + dV + dV^2 + \cdots + dV^{n-2} = d \cdot \sum_{j=0}^{n-2} V^j \ . \tag{A.16}$$

Therefore, for $G^n_{d,V}$, there are V sub-graphs, where $G^n_{d,V}(l)$, $l \in [V]$, is fully connected to the adjacent sub-graphs, and for $x^n \in [V^n]$,

$$deg(x^n) = d + d \cdot \sum_{j=1}^{n-1} V^j \ , \tag{A.17}$$

which leads to the closed-form expression by (2.36).

A.7 Proof of Proposition 2.9

For any graph $G(\mathcal{V}, \mathcal{E})$ where $\max_{k \in [V]} deg(x_k) \geq 3$ and no complete sub-graph exists of size $\max_{k \in [V]} deg(x_k) + 1$, G can be colored with at most $\max_{k \in [V]} deg(x_k)$ colors, making it $\max_{k \in [V]} deg(x_k)$-colorable [257]. For any $G_{d,V}$, the chromatic number satisfies $\chi(G_{d,V}) \geq \omega(G_{d,V})$, where $\omega(G_{d,V})$ represents the clique number, i.e., the size of the largest complete sub-graph (clique) in $G_{d,V}$ [258].

Upper and lower bounds on $\chi(G_{d,V})$ hold for $d, V \in \mathbb{Z}^+$ with $V \geq d + 1$. From

Proposition 2.8, $G^n_{d,V}$ is a regular graph. In $G^2_{d,V}$, a clique of size d exists among d adjacent sub-graphs, leading to $\chi(G^2_{d,V}) = d^2$, as at least d colors are needed per sub-graph. Similarly, in the j-fold OR product, each sub-graph connects to d others, forming a d^j clique, resulting in $\chi(G^j_{d,V}) = d^j$.

From the induction steps, for the n-fold OR product, we deduce that $\chi(G^n_{d,V}) = d^n$.

A.8 Proof of Proposition 2.11

According to Definition 2.6, the diameter of a graph is the length of the shortest path between its most distant nodes. Consider the second OR product of a d-regular graph, specifically the 2-regular cycle C^2_5. As shown in Figure 2.7, the diameter of C_5 is 2. In the OR product construction, each node is replaced by a sub-graph in the next product, and every edge between nodes induces a complete bipartite connection between the corresponding sub-graphs. In Figure 2.3, there are two complete connections between the first and third sub-graphs, corresponding to the farthest distance in the original graph. Since these connections preserve distances without increasing them, the diameter of C^2_5 remains equal to that of C_5.

From the properties of $G^n_{d,V}$ (see Chapter 2.3.5), the edges between adjacent nodes in $G^n_{d,V}$ induce complete connections between the corresponding subgraphs in higher powers (e.g., $G^{n-1}_{d,V}(i)$ and $G^{n-1}_{d,V}(i+1)$). Thus, computing the second OR power of $G^{n-1}_{d,V}$ preserves the structural connectivity of the sub-graphs in $G^n_{d,V}$. Consequently, the distance between the two most distant nodes of $G^n_{d,V}$ remains unchanged (see Figure 2.3). The complete connectivity among sub-graphs covers all possible paths, thereby preserving the diameter.

A.9 Proof of Corollary 2.8

For the upper bound in (2.52), note that for K_i where $i = V$, the largest eigenvalue is $V - 1$ with multiplicity 1, and -1 has multiplicity $V - 1$ [259]. Thus,

$$\Lambda(K^n_i) = \max(\lambda_2(\mathbf{A}^n_f), |\lambda_{V^n}(\mathbf{A}^n_f)|) = 1 \ , \tag{A.18}$$

and using (2.38), we obtain the RHS of (2.52). Subsequently, for the lower bound in (2.52), we adjust (2.38) for C^n_i, and determine $E_\theta(C^n_i)$. For computing $\Lambda(C^n_i)$, we note that C^n_i is a d-regular graph as specified in Proposition 2.1. Therefore, the eigenvalues satisfy $\{\lambda_1 > \lambda_2 > \cdots > \lambda_{V^n}\}$. Noting that the diagonal entries of $\mathbf{A}^n_f$ are zero allows us to deduce that $trace(\mathbf{A}^n_f) = 0$. Given that for C^n_i, $\lambda_1(\mathbf{A}^n_f)$ is positive and has the largest cardinality, $\lambda_{V^n}(\mathbf{A}^n_f)$ must be approximately equal in magnitude (negative value) and greater in value than $\lambda_2(\mathbf{A}^n_f)$ so that the trace becomes zero, as numerically demonstrated for C^n_i in Chapter 2.3.4, which indicates the cardinality of λ_k follows the order $\{d \geq |\lambda_1| >$

$|\lambda_{V^n}| > |\lambda_2| > \dots\}$. Therefore, by replacing $\Lambda(C_i^n)$ where

$$\max(\lambda_2(\mathbf{A}_f^n), |\lambda_{V^n}(\mathbf{A}_f^n)|) = |\lambda_{V^n}(\mathbf{A}_f^n)| \ , \tag{A.19}$$

in the denominator of (2.38), we reach (2.52).

A.10 Proof of Proposition 2.12

For the n-fold OR product graph G^n, $\alpha(G^n)$ is calculated as follows:

$$\alpha(G^n) = (\alpha(G))^n \ , \tag{A.20}$$

which follows by induction. For $n = 2$, each sub-graph in G^2 is isomorphic to G with an independence number $\alpha(G)$. Hence, the size of the maximum independent set in G^2 is $\alpha(G)^2$. Similarly, for the 3-fold OR product, the sub-graphs are isomorphic to G^2, which implies $\alpha(G^3) = \alpha(G)^3$. This pattern extends to higher-order products [256, Chapter 5], leading to (A.20).

Next, using (A.20) and (2.2), we derive an upper bound on $H_G(X_1)$ for a given G^n. Exploiting the graph decomposition technique in the Proof of Proposition 2.5 (see Appendix A.4), we devise an achievable coloring for G^n, given by

$$P(\mathcal{C}_{G^n}) = P(\zeta_0 \cdot \frac{1}{V^n} \ , \ \zeta_1 \cdot \frac{\alpha(G)}{V^n} \ , \dots \ , \ \zeta_n \cdot \frac{\alpha(G)^n}{V^n}) \ , \tag{A.21}$$

with coefficients $\{\zeta_t \ , \ t \in \{0\} \cup [n]\}$ that represent the weights of the decomposition and satisfy the following constraints:

$$\begin{aligned} &(a) \ \left(\sum_{t=1}^{n} \zeta_t \cdot \frac{\alpha(G)^t}{V^n}\right) + \zeta_0 \cdot \frac{\alpha(G^0)}{V^n} = 1, \\ &(b) \ \zeta_t \geq \alpha(G) \cdot \zeta_{t-1} \ , \\ &(c) \ \zeta_0 = 1 \ . \end{aligned} \tag{A.22}$$

In (A.22), the constraint (a) ensures that the coloring assignment forms a valid PMF. Constraints (b) and (c) are identical to those in the proof of Proposition 2.5. Employing (A.22) yields

$$H\Big(\zeta_n \cdot (\frac{\alpha(G^n)}{V^n}), \zeta_{n-1} \cdot \frac{\alpha(G^{n-1})}{V^n}, \dots, \zeta_0 \cdot \frac{1}{V^n}\Big) \ , \tag{A.23}$$

where (A.23) can be optimized over $\{\zeta_t \ , \ t \in \{0\} \cup [n]\}$ to minimize $H(\mathcal{C}_{G^n})$. Subsequently, to derive $H_{G^n}^{\chi}(\mathbf{X}_1)$, we use (2.18) and normalize the entropy calculated in (A.23) for the n-fold OR product by a factor of $\frac{1}{n}$. Finally, by using (A.20) along with (A.23), which upper bounds $H_G(X_1)$, we reach the statement of the proposition.

A.11 Proof of Corollary 2.9

For any connected $G(\mathcal{V},\mathcal{E})$ with $V = 2k+1$, we have $\chi(C_{2k+1}) \leq \chi(G)$, so $\chi(C_{2k+1}^n)$ gives a lower bound on $\chi(G^n)$. Therefore, we first calculate $\chi_f(C_{2k+1}^n)$, then use it to derive a lower bound on $H_{C_{2k+1}}(X_1)$ and subsequently on $H_G(X_1)$. Using $a:b$ coloring for C_{2k+1}, we show that at least $2b+1$ colors are needed and claim that the infimum of $\chi_f(C_{2k+1}^n)$ occurs at $b=k$. We prove $\chi_b(C_{2k+1}) = 2b+1$, by contradiction: Assume that $\chi_b(C_{2k+1}) \leq 2b$ and assign each vertex $i \in [2k+1]$ a coloring set $\mathcal{C}_i$ of size b. For $i \in [2k]$, the coloring sets are defined as

$$\mathcal{C}_1 = [b], \quad \mathcal{C}_2 = [b+1, 2b], \ \ldots, \mathcal{C}_{2k-1} = [b], \mathcal{C}_{2k} = [b+1, 2b] \ .$$

Using $2b$ colors recursively over the vertices, disjoint color sets are assigned to each node up to the last node x_{2k+1}. However, reusing the previous $2b$ colors for $\mathcal{C}_{2k+1}$ is impossible, leading to a contradiction of the initial assumption. Next, from (2.6), we show that $\frac{2k+1}{k} \leq \chi_f(C_{2k+1})$ holds. From [258], a lower bound on $\chi_f(G)$ is given by $\chi_f(G) \geq \frac{V}{\alpha(G)}$, which by applying $G = C_{2k+1}$, proves the assumption that $b = k$.

To compute $H^{\chi}_{C^n_{2k+1}}(\mathbf{X}_1)$, we use the bound $H^f_G(X_1) \leq H_G(X_1)$ from [78, Lemma 1], and leveraging their coloring PMF in [78, Proposition 2], yielding as $n \to \infty$:

$$H^f_G(X_1) = \lim_{n\to\infty} \frac{1}{n} \inf_b \frac{1}{b} \min_{\mathcal{C}^f_{C^n_i}} H(\mathcal{C}^f_{C^n_i}) \ . \tag{A.24}$$

Given a uniform $\mathcal{C}^f_{C^n_i}$, the entropy is simplified and calculated as $H^{\chi_f}_{C^n_{2k+1}}(\mathbf{X}_1) = \log_2 |\chi_f(C^n_{2k+1})|$. From (2.6), the graph entropy of C^n_{2k+1}, incorporating $\frac{1}{n}$ normalization for the n-th realization, is:

$$\frac{1}{n} H^{\chi}_{C^n_{2k+1}}(\mathbf{X}_1) = \frac{1}{n} \cdot \log_2 \left(\frac{2b+1}{b}\right)^n = \log_2 \left(\frac{2b+1}{b}\right) \ , \tag{A.25}$$

where the infimum of b is k in (A.25), this captures $H_{C_{2k+1}}(X_1)$ and validating (2.54) for any G.

A.12 Proof of Theorem 2.2

By Lemma 2.1, the eigenvalues of the sum of two symmetric matrices equal the sums of their respective eigenvalues. To calculate $\lambda_k(\mathbf{A}^n_f)$ for $k \in [V^n]$, we express $\mathbf{A}^n_f$ as the sum of two symmetric matrices and determine the eigenvalues by summing those of the individual matrices. For $n=1$, the eigenvalues $\lambda_k(\mathbf{A}_f)$ can be calculated using GCT or QR decomposition methods. For $n=2$, the adjacency matrix $\mathbf{A}^2_f$ is partitioned into $\mathbf{A}^2_f = \mathbf{A}^2_{Gr} + \mathbf{A}^2_{f^{\mathsf{c}}}$, where $\mathbf{A}^2_{Gr}$ is a diagonal block with $\mathbf{A}_f$ on the diagonals and $\mathbf{Z}_V$ elsewhere. The term $\mathbf{A}^2_{f^{\mathsf{c}}} = \mathbf{A}^2_f - \mathbf{A}^2_{Gr}$ is constructed with all-zero matrices $\mathbf{Z}_V$ on the diagonal and $\mathbf{J}_V$ or $\mathbf{Z}_V$ on the off-diagonal blocks based on the sub-graphs connections.

This decomposition allows:

$$\lambda_k(\mathbf{A}_f^2) = \lambda_k(\mathbf{A}_{Gr}^2) + \lambda_k(\mathbf{A}_{f^\mathsf{c}}^2), \quad k \in [V^2]. \tag{A.26}$$

Assume that the decomposition holds for $n = j$,

$$\mathbf{A}_f^j = \mathbf{A}_{Gr}^j + \mathbf{A}_{f^\mathsf{c}}^j \tag{A.27}$$

and $\lambda_k(\mathbf{A}_f^j) = \lambda_k(\mathbf{A}_{Gr}^j) + \lambda_k(\mathbf{A}_{f^\mathsf{c}}^j)$. For $n = j+1$, decompose $\mathbf{A}_f^{j+1}$ as: $\mathbf{A}_f^{j+1} = \mathbf{A}_{Gr}^{j+1} + \mathbf{A}_{f^\mathsf{c}}^{j+1}$. Here, $\mathbf{A}_{Gr}^{j+1}$ is block diagonal with $\mathbf{A}_f^j$ on the diagonal blocks, and $\mathbf{A}_{f^\mathsf{c}}^{j+1}$ contains off-diagonal terms derived from $\mathbf{J}_{V^{j+1}}$ and $\mathbf{Z}_{V^{j+1}}$. Thus, by induction, $\lambda_k(\mathbf{A}_f^{n-1})$ is computed using δ_k for using GCT according to (2.55), and the decomposition and eigenvalue sum hold for all $n \geq 2$ as shown in (2.58).

A.13 Proof of Corollary 2.10

Consider a block diagonal matrix (e.g., $\mathbf{A}_{Gr}^j \in \mathbb{F}_2^{V^j \times V^j}$, $j \in [n]$), where each diagonal block is identical to $\mathbf{A}_f^{j-1} \in \mathbb{F}_2^{V^{j-1} \times V^{j-1}}$, and all off-diagonal entries are zero. The eigenvalues of $\mathbf{A}_{Gr}^j$, $j \in [n]$ is equal to $\lambda_k(\mathbf{A}_f^{j-1})$, $k \in [V^{j-1}]$, but the algebraic multiplicity of each eigenvalue is scaled by the number of times the block $\mathbf{A}_f^{j-1}$ appears on the diagonal. Thus, for the 2-fold OR product, we have $\lambda_k(\mathbf{A}_f^2) = \lambda_k(\mathbf{A}_{Gr}^2) + \lambda_k(\mathbf{A}_{f^\mathsf{c}}^2)$. Furthermore, for $n = 3$, we can substitute $\lambda_k(\mathbf{A}_{Gr}^3)$ with $\lambda_k(\mathbf{A}_{Gr}^2) + \lambda_k(\mathbf{A}_{f^\mathsf{c}}^2)$, as follows

$$\lambda_k(\mathbf{A}_f^3) = \lambda_k(\mathbf{A}_{Gr}^2) + \lambda_k(\mathbf{A}_{f^\mathsf{c}}^2) + \lambda_k(\mathbf{A}_{f^\mathsf{c}}^3)\ . \tag{A.28}$$

Similarly, for j-fold OR product, we have

$$\lambda_k(\mathbf{A}_f^j) = \lambda_k(\mathbf{A}_{Gr}^2) + \lambda_k(\mathbf{A}_{f^\mathsf{c}}^2) + \cdots + \lambda_k(\mathbf{A}_{f^\mathsf{c}}^{j-1}) + \lambda_k(\mathbf{A}_{f^\mathsf{c}}^j)\ , \tag{A.29}$$

and for $n = j+1$, the relation is detailed as

$$\lambda_k(\mathbf{A}_f^{j+1}) = \lambda_k(\mathbf{A}_{Gr}^2) + \lambda_k(\mathbf{A}_{f^\mathsf{c}}^2) \cdots + \lambda_k(\mathbf{A}_{f^\mathsf{c}}^{j+1})\ . \tag{A.30}$$

Similarly, for the n-fold OR product $\lambda_k(\mathbf{A}_f^n)$ is calculated as follows:

$$\lambda_k(\mathbf{A}_f^n) = \lambda_k(\mathbf{A}_{Gr}^2) + \lambda_k(\mathbf{A}_{f^\mathsf{c}}^2) + \cdots + \lambda_k(\mathbf{A}_{f^\mathsf{c}}^{n-1}) + \lambda_k(\mathbf{A}_{f^\mathsf{c}}^n),\ k \in [V^n]\ . \tag{A.31}$$

By substituting $\lambda_k(\mathbf{A}_{Gr}^2)$ in (A.31) with $\lambda_k(\mathbf{A}_f)$, we reach the statement of the corollary in (2.59).

Chapter B

Appendix for Chapter 3

B.1 Technical Preliminaries

This appendix summarizes the theoretical background required for Chapter 3. Complete derivations and proofs are provided in Chapter 2. Here, we recall the main definitions and include a worked example for clarity.

B.1.1 Distributed Source Compression and Communication Cost

Given statistically dependent, finite-alphabet, i.i.d. random sequences $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N$ 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 [82]:

$$\sum_{i \in \mathcal{S}} R_i \geq H(X_{\mathcal{S}} \mid X_{\mathcal{S}^c}), \quad \forall \mathcal{S} \subseteq \Omega. \tag{B.1}$$

This serves as the baseline for the distributed functional compression results in Chapter 3.

B.1.2 Characteristic Graphs and Functional Compression

For completeness, we briefly restate the concept of characteristic graphs introduced in Chapter 2. Given a function $f(X_1, X_2)$, the *characteristic graph* G_{X_1} connects two values $x_1^1, x_1^2 \in \mathcal{X}_1$ whenever there exists x_2 such that $f(x_1^1, x_2) \neq f(x_1^2, x_2)$ and both (x_1^1, x_2) and (x_1^2, x_2) have nonzero probability. A valid coloring of G_{X_1} assigns different colors to adjacent vertices, and the chromatic entropy of this coloring determines the achievable compression rate.

The related *graph entropy* and *conditional graph entropy* [72, 74] are defined as

$$H_{G_{X_1}}(X_1) = \min_{X_1 \in U_1 \in \Gamma(G_{X_1})} I(X_1; U_1), \tag{B.2}$$

$$H_{G_{X_1}}(X_1 \mid X_2) = \min_{\substack{U_1 - X_1 - X_2 \\ X_1 \in U_1 \in \Gamma(G_{X_1})}} I(X_1; U_1 \mid X_2), \tag{B.3}$$

where $\Gamma(G_{X_1})$ denotes the set of maximal independent sets of G_{X_1}. These quantities characterize the minimum rates required for distributed function computation

B.1.3 Example: Characteristic Graph Entropy of Ternary Variables

We revisit a canonical example from [74] to illustrate the relation between graph entropy and conditional graph entropy for ternary variables.

Example B.1 (Characteristic Graph Entropy of Ternary Variables, Examples 1-2 in [74])). *Let $X_1 \in \{1,2,3\}$ be uniform, and consider G_{X_1} with a single edge $\mathcal{E}_{X_1} = \{(1,3)\}$. The set of maximal independent sets (MISs) is $\Gamma(G_{X_1}) = \{\{1,2\},\{2,3\}\}$. Following [74], minimizing $I(X_1;U_1)$ over all $P(U_1|X_1)$ yields $H_{G_{X_1}}(X_1) = \frac{2}{3}$.*

Now let (X_1, X_2) be jointly uniform over $\{(x_1,x_2) : x_1, x_2 \in \{1,2,3\}, x_1 \neq x_2\}$. Then $H(X_1|X_2) = 1$, and minimizing $I(X_1;U_1|X_2)$ gives the conditional graph entropy

$$H_{G_{X_1}}(X_1|X_2) = \frac{2}{3} h\left(\frac{1}{4}\right) .$$

Example B.1 demonstrates how side information reduces the communication cost in distributed functional compression. Further examples, the chromatic entropy definition, and asymptotic results connecting $H^{\chi}_{G^n}(X_1)$ and $H_{G_{X_1}}(X_1)$ are presented in Chapter 2.

B.1.4 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 sub-functions. In our proposed framework, for the distributed computing of these functions, we leverage characteristic graphs that can capture the structure of sub-functions. 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 $G_{X_i,j} = G(\mathcal{V}_{X_i}, \mathcal{E}_{X_i,j})$ 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 [77], using multi-functional characteristic graphs[1]. To that end, server $i \in \Omega$ creates a union of graphs on the same set of vertices $\mathcal{V}_{X_i}$ with a set of edges $\mathcal{E}^{\cup}_{X_i}$, which satisfies

$$G^{\cup}_{X_i} = \bigcup_{j\in[K_c]} G_{X_i,j} = G(\mathcal{V}_{X_i}, \mathcal{E}^{\cup}_{X_i}) , \quad \mathcal{E}^{\cup}_{X_i} = \bigcup_{j\in[K_c]} \mathcal{E}_{X_i,j} . \tag{B.4}$$

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 $F_j(x_i^1, x^1_{\Omega\setminus_i}) \neq$

[1] A multi-functional characteristic graph is an OR function of individual graphs [77, Definition 45].

$F_j(x_i^2, x_{\Omega\setminus i}^1)$, 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 (2.18) and (2.17).

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 $\mathcal{X}_i$, MISs of $G_{X_i}^{\cup}$ in (B.4) 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 B.1.1).

Distributed Functional Compression

The fundamental limit of functional compression has been given by Körner [72]. Given $\mathbf{X}_i \in \mathbb{F}_q^{|\mathcal{Z}_i| \times n}$ for server $i \in \Omega$, the encoding function en specifies MISs given by the valid colorings $\mathcal{C}_{G_{X_i}^n}(\mathbf{X}_i)$. Let the number of symbols in $\mathbf{Z}_i = g_i(\mathbf{X}_i) = en(\mathcal{C}_{G_{X_i}^n}(\mathbf{X}_i))$ be T_i for server $i \in \Omega$. Hence, the communication cost of server i, as $n \to \infty$ is given by (3.5).

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

$$R_{ach} = \sum_{i\in\mathcal{S}} R_i \geq H_{G_{X_\mathcal{S}}}(X_\mathcal{S} \,|\, Z_{\mathcal{S}^c}), \quad \mathcal{S} \subseteq \Omega \ , \tag{B.5}$$

where $H_{G_{X_\mathcal{S}}}(X_\mathcal{S})$ is the joint graph entropy of $\mathcal{S} \subseteq \Omega$, and it is defined as [77, Definition 30]:

$$H_{G_{X_\mathcal{S}}}(X_\mathcal{S}) = \lim_{n\to\infty} \min_{\{\mathcal{C}_{G_{X_i}^n}\}_{i\in\mathcal{S}}} \frac{1}{n} H(\mathcal{C}_{G_{X_i}^n}(\mathbf{X}_i),\, i \in \mathcal{S}) \ , \tag{B.6}$$

where $\mathcal{C}_{G_{X_i}^n}(\mathbf{X}_i)$ is the coloring of the n-fold product graph $G_{X_i}^n$ that $i \in \Omega$ builds for computing $f(\mathbf{X}_\Omega)$ [77]. Similarly, exploiting [77, Definition 31], the *conditional* graph entropy of the servers is given as

$$H_{G_{X_\mathcal{S}}}(X_\mathcal{S} \,|\, Z_{\mathcal{S}^c}) = \lim_{n\to\infty} \min_{\{\mathcal{C}_{G_{X_i}^n}\}_{i\in\Omega}} \frac{1}{n} H(\mathcal{C}_{G_{X_i}^n}(\mathbf{X}_i),\, i \in \mathcal{S} \,|\, en(\mathcal{C}_{G_{X_i}^n}(\mathbf{X}_i)),\, i \in \mathcal{S}^c) \ . \tag{B.7}$$

Using (B.4), we jointly capture the structures of the set of demanded functions. Hence, this enables us to provide a refined communication cost model in (3.5) versus the characterizations as a function of K_c, see, e.g., [43, 207, 213].

B.2 Proof of Theorem 3.1

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[2] $G_{X_i,j}$ 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 $G^{\cup}_{X_i} = G(\mathcal{V}_{X_i}, \mathcal{E}_{X_i}) = \bigcup_{j\in[K_c]} G_{X_i,j}$ the *union characteristic graph*, given as in (B.4). In the description of $G^{\cup}_{X_i}$, the set $\mathcal{V}_{X_i}$ is the support set of X_i, i.e., $\mathcal{V}_{X_i} = \mathcal{X}_i$, and $\mathcal{E}_{X_i}$ is the union of edges, i.e., $\mathcal{E}_{X_i} = \bigcup_{j\in[K_c]} \mathcal{E}_{X_i,j}$, where $\mathcal{E}_{X_i,j}$ 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 $\mathscr{C}_i$ such that $\mathscr{C}_i \ni g_i$, 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 $g_i \in \mathscr{C}_i$ 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 $\mathcal{C}_{G_{X_i}}(X_i)$. We denote by $\mathbf{Z}_i = g_i(\mathbf{X}_i) = en(\mathcal{C}_{G^n_{X_i}}(\mathbf{X}_i))$ 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:

$$Z_i = g_i(X_i) \ , \quad i \in \Omega \ . \tag{B.8}$$

Combining the notions of the union graph in (B.4) and the encodings of the individual servers given in (B.8), 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,

$$R_i \geq \min_{Z_i=g_i(X_i)\,:\,g_i\in\mathscr{C}_i} H_{G^{\cup}_{X_i}}(X_i) \ , \tag{B.9}$$

where equality is achievable in (B.9). Because the user can recover the desired functions using any set of N_r servers, the achievable sum rate is upper bounded by

$$R_{ach} \leq \sum_{i=1}^{N_r} \min_{Z_i=g_i(X_i)\,:\,g_i\in\mathscr{C}_i} H_{G^{\cup}_{X_i}}(X_i) \ . \tag{B.10}$$

B.3 Proof of Proposition 3.1

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 (3.4). Given

[2]The characteristic-graph-based approach is valid provided that each sub-function 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 $G^{\cup}_{X_i}$, $i \in \Omega$ (or $G_{X_i,j}$, $j \in [Kc]$ each) has more than one vertex.

the recovery threshold N_r, it holds that

$$R_{ach} \leq \sum_{i=1}^{N_r} \min_{Z_i = g_i(X_i) \,:\, g_i \in \mathscr{C}_i} H_{G^{\cup}_{X_i}}(X_i) = \sum_{i=1}^{N_r} \min_{Z_i \,:\, g_i \in \mathscr{C}_i} \; \min_{X_i \in U_i \in \Gamma(G^{\cup}_{X_i})} I(X_i; U_i), \tag{B.11}$$

$$= \sum_{i=1}^{N_r} \Big[H(W_{(i-1)\Xi+1}^{(i-1)\Xi+M}) - H\Big(W_{(i-1)\Xi+1}^{(i-1)\Xi+M} \,\Big|\, Z_i\Big)\Big], \tag{B.12}$$

$$= \sum_{i=1}^{N_r} \Big[M - \Big(M - H(Z_i)\Big)\Big] = \sum_{i=1}^{N_r} H(Z_i) \;, \tag{B.13}$$

where in (B.11), we used the identity $H_{G^{\cup}_{X_i}}(X_i) = \min_{X_i \in U_i \in \Gamma(G^{\cup}_{X_i})} I(X_i; U_i)$. Furthermore, if the codebook $\mathscr{C}_i$ is restricted to linear combinations of sub-functions, Z_i is given by the following set of linear equations:

$$Z_i = g_i(X_i) = \Big\{ \sum_{k=(i-1)\Xi+1}^{(i-1)\Xi+M} \zeta_k^{(l)} W_k \;,\; l \in [K_c] \Big\} \;. \tag{B.14}$$

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 (3.4), in a distributed manner. Hence, each independent set $U_i \in \Gamma(G^{\cup}_{X_i})$, with $\Gamma(G^{\cup}_{X_i})$ denoting the set of MISs of X_i, of X_i is captured by the linear functions of $\{W_k\}_{k \in [(i-1)\Xi+1:(i-1)\Xi+M]}$, i.e., each $U_i \in \Gamma(G^{\cup}_{X_i})$ is determined by (B.14). Hence, the user can recover the requested functions by linearly combining the transmissions of the N_r servers:

$$f_j(W_{\mathcal{K}}) = \sum_{i=1}^{N_r} \beta_{ji} Z_i = \sum_{i=1}^{N_r} \beta_{ji} g_i(X_i) = \sum_{k=1}^{K} \psi_{jk} W_k \;, \quad j \in [K_c] \;. \tag{B.15}$$

In (B.12), 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 $\Xi = \frac{K}{N}$, it holds under cyclic placement that

$$X_i = W_{(i-1)\Xi+1}^{(i-1)\Xi+M} = W_{(i-1)\Xi+1}, W_{(i-1)\Xi+2}, \ldots, W_{(i-1)\Xi+M} \;, \tag{B.16}$$

and $\zeta_k^{(l)}$ are the coefficients for computing function $l \in [K_c]$. In (B.13), 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

$$H\Big(W_{(i-1)\Xi+1}^{(i-1)\Xi+M} \,\Big|\, Z_i\Big) = H\Big(W_{(i-1)\Xi+1}^{(i-1)\Xi+M} \,,\, Z_i\Big) - H(Z_i) \overset{(a)}{=} H\Big(W_{(i-1)\Xi+1}^{(i-1)\Xi+M}\Big) - H(Z_i) \;, \tag{B.17}$$

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

$$\sum_{i=1}^{N_r} \min_{Z_i \,:\, g_i \in \mathscr{C}_i} H_{G^{\cup}_{X_i}}(X_i) \leq K_c N_r \ . \tag{B.18}$$

Exploiting the cyclic placement model, we can tighten the bound in (B.18). Note that server $i = 1$ can help recover M sub-functions (at most, i.e., M transmissions needed to recover M sub-functions), and each of the servers $i \in [2 : N_r]$ can help recover an additional Ξ sub-functions (at most, i.e., Ξ transmissions are needed to recover Ξ sub-functions). Hence, the set of servers $[N_r]$ suffices to provide $M + (Nr - 1) \cdot \Xi = N \cdot \Xi = K$ sub-functions 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 $1 \leq K_c < \Xi$, it is sufficient for each server to transmit K_c linearly independent combinations of their sub-functions. This leads to resolving $K_c N_r$ linear combinations of K sub-functions from N_r servers that are sufficient to derive the demanded K_c linear functions. Because $K_c N_r < \Xi \cdot N_r$, there are $K - K_c N_r > \Xi \cdot (N - N_r) = M - \Xi$ unresolved linear combinations of K sub-functions.

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

(iii) When $\Xi \cdot N_r < K_c \leq K$, each server needs to transmit at a rate $\frac{K_c}{N_r}$ where $\frac{K_c}{N_r} > \Xi$ and $\frac{K_c}{N_r} \leq \frac{K}{N_r} = \Xi \cdot (\frac{N_r + N - N_r}{N_r}) = \Xi + \Xi \cdot (\frac{N - N_r}{N_r})$, 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 $\Xi \cdot (N - N_r)$ linear combinations, and $\Xi \cdot (\frac{N - N_r}{N_r})$ 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 (B.15) 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:

$$\sum_{i=1}^{N_r} \min_{Z_i \,:\, g_i \in \mathscr{C}_i} H_{G^{\cup}_{X_i}}(X_i) = \begin{cases} K_c N_r \ , & 1 \leq K_c < \Xi \ , \\ \Xi \cdot N_r \ , & \Xi \leq K_c \leq \Xi \cdot Nr \ , \\ K_c \ , & \Xi \cdot Nr < K_c \leq K \ , \\ K \ , & K < K_c \ , \end{cases} \tag{B.19}$$

where it is easy to note that (B.19) matches the communication cost in [43, Theorem 2]. The i.i.d. distribution assumption for W_k ensures that this result holds for any $q \geq 2$.

B.4 Proof of Proposition 3.2

Similarly as in Theorem 3.1, we let $G_{X_i}^{\cup} = \bigcup_{j\in[K_c]} G_{X_i,j}$ 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 $W_{\mathcal{Z}_i} = W_{1+(i-1)\Xi}, W_{1+(i-1)\Xi+1}, \ldots, W_{M+(i-1)\Xi}$, 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 $u_0(G_{X_i}^{\cup})$ and $u_1(G_{X_i}^{\cup})$, 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

$$f(w_{\mathcal{Z}_i}, w_{\mathcal{Z}_{\mathcal{S}}\backslash\mathcal{Z}_i}) = 1 \ , \quad \forall w_{\mathcal{Z}_{\mathcal{S}}\backslash\mathcal{Z}_i} \in \{0,1\}^{|\mathcal{K}\backslash\mathcal{Z}_i|} \ , \tag{B.20}$$

where notation $w_{\mathcal{Z}_{\mathcal{S}}\backslash\mathcal{Z}_i}$ denotes the dataset values for the set of datasets stored in the subset of servers $\mathcal{S}\backslash i$. Note, in general, that $K_n(\mathcal{S}) = |\mathcal{Z}_{\mathcal{S}}| = \left|\bigcup_{i\in\mathcal{S}} \mathcal{Z}_i\right|$. In the case of cyclic placement based on (3.1), out of the set of all datasets $\mathcal{K}$, there are Ξ datasets that belong exclusively to server $i \in \Omega$. In this case, $|\mathcal{K}\backslash\mathcal{Z}_i| = K - \Xi$.

Note that (B.20) captures the independent set $u_1(G_{X_i}^{\cup}) \ni w_{\mathcal{Z}_i}$. Equivalently, the set of dataset values $W_{\mathcal{Z}_i}$ that lands in $u_1(G_{X_i}^{\cup})$ of $G_{X_i}^{\cup}$ yields $Z_i = 1$. The transmitted information takes the value $Z_i = 1$ with a probability

$$\mathbb{P}(Z_i = 1) = \mathbb{P}(W_{\mathcal{Z}_i} \in u_1(G_{X_i}^{\cup})) \ , \quad i \in \Omega \ , \tag{B.21}$$

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

B.5 Proof of Proposition 3.3

Recall that $W_k \sim \text{Bern}(\epsilon)$ are i.i.d. across $k \in [K]$, and each server has a capacity $M = \Xi \cdot (N - N_r + 1)$. This means that given the number of datasets K, each server can compute the product of $\Xi\cdot(N-N_r+1)$ sub-functions and, hence, the minimum number of servers to evaluate the multi-linear function $f(W_{\mathcal{K}}) = \prod_{k\in[K]} W_k$ is $N^* = \left\lfloor \frac{N}{N-N_r+1} \right\rfloor$ such that given its capacity $M = |\mathcal{Z}_i|$, each server can compute the product of a disjoint set of M sub-functions, 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 $G_{X_1} = G(\mathcal{V}_{X_1}, \mathcal{E}_{X_1})$ for X_1, with respect to variables $X_\Omega\backslash X_1 = X_2, \ldots, X_N$ and $f(W_{\mathcal{K}})$, and similarly for other servers to characterize the sum-rate for the computation by evaluating the entropy of each graph.

To evaluate the first term in (3.12), we choose a total of N^* servers with a disjoint set of sub-functions. 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

$$\sum_{i\in\mathcal{N}^*} R_i \overset{(a)}{\leq} H_{G_{X_{i_1}}}(X_{i_1}) + H_{G_{X_{i_2}}}(X_{i_2}\,|\,Z_{i_1}) + \cdots + H_{G_{X_{i_{N^*}}}}(X_{i_{N^*}}\,|\,Z_{i_1}, Z_{i_2}, \ldots, Z_{i_{N^*-1}})$$
$$\overset{(b)}{=} h(\epsilon_M) + \epsilon_M h(\epsilon_M) + (\epsilon_M)^2 h(\epsilon_M) + \cdots + (\epsilon_M)^{N^*-1} h(\epsilon_M)$$
$$\overset{(c)}{=} \frac{1-(\epsilon_M)^{N^*}}{1-\epsilon_M} \cdot h(\epsilon_M), \tag{B.22}$$

where (a) follows from assuming $\mathcal{S} = \{i_1, i_2, \ldots, i_{N^*}\}$ 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[3], i.e., $\sum_{l=0}^{N^*-1}(\epsilon_M)^l = \frac{1-(\epsilon_M)^{N^*}}{1-\epsilon_M}$.

In the case of $\Xi_N = N - N^* \cdot (N - N_r + 1) > 0$, the product of K sub-functions cannot be determined by N^* servers, and we need additional servers $\mathcal{I}^* \in \Omega$ to aid the computation and determine the outcome of $f(W_\mathcal{K})$ by computing the product of the remaining ξ_N sub-functions. In other words, if $\Xi_N > 0$ and $\prod_{i\in\mathcal{S}}\prod_{k\in\mathcal{Z}_i} W_k = 1$, the (N^*+1)-th server determines the outcome of $f(W_\mathcal{K})$ by computing the product of sub-functions $W_k \sim \text{Bern}(\epsilon)$, $k \in \big[N - \xi_N + 1 : N\big]$ that cannot be captured by the previous N^* servers. Hence, the additional rate, given by the second term in (3.12), is given by the product of the term

$$(\epsilon_M)^{N^*} = \mathbb{P}\Big(\prod_{i\in\mathcal{S}}\prod_{k\in\mathcal{Z}_i} W_k = 1\Big)\,, \tag{B.23}$$

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

B.6 Multi-Server Multi-Function Setting Via Maddah-Ali Niesen Placement

This model builds on the cyclic placement model described in Chapter 3. In this setting, each sub-function W_k, $k \in [K]$ is further split into $\binom{N}{|\tau|}$ disjoint subsets of equal size. Hence, W_k can be represented as

$$W_k = (W_{k,\tau} : \tau \subset [N]\,, |\,\tau| = N\gamma)\,. \tag{B.24}$$

We next consider two examples using MAN placement as follows.

[3]While Proposition 3.3 uses the conditional graph entropies, the statements of Theorem 3.1 and Proposition 3.1, and Proposition 3.2 do not take into account the notion of conditional graph entropies. However, as indicated in Chapter 3.4.1 for computing linearly separable functions, and in Chapter 3.4.2 for computing multi-linear functions, respectively, we used the conditional entropy-based sum rate in (B.22) to evaluate and illustrate the achievable gains over [43, 82].

Example B.2 (Affine function under Maddah-Ali Niesen placement model for $N = 3$, $K = 3$, $N_r = 2$, $K_c = 1$, $M = 2$)**.** *Under this setting, we aim to compute* $f(W_{\mathcal{K}}) = W_1 + W_2 + W_3$*, where we assume* $H(W_k) = 1$*,* $k \in [K]$*. Inferring* $\gamma = 2/3$ *and* $|\tau| = N\gamma = 2$*, the subsets of each sub-function are labeled according to (B.24) as*

$$W_k = (W_{k,\{1,2\}},\, W_{k,\{2,3\}},\, W_{k,\{1,3\}})\ . \tag{B.25}$$

We further have $f(W_{\mathcal{K}}) = \Big(\sum_{k=1}^{3} W_{k,\{1,2\}},\, \sum_{k=1}^{3} W_{k,\{2,3\}},\, \sum_{k=1}^{3} W_{k,\{1,3\}}\Big)$.

Using the placement scheme described by (B.26), X_i*,* $i \in \Omega$ *satisfies:*

$$\begin{aligned} X_1 &= \{W_{k,\{1,2\}},\, W_{k,\{1,3\}}\}_{k\in[3]}\ , \\ X_2 &= \{W_{k,\{1,2\}},\, W_{k,\{2,3\}}\}_{k\in[3]}\ , \\ X_3 &= \{W_{k,\{1,3\}},\, W_{k,\{2,3\}}\}_{k\in[3]}\ . \end{aligned}$$

Exploiting (B.5) and using symmetry, it is sufficient to determine the rate region for computing $f(\mathbf{W}_{\mathcal{K}}) = \mathbf{W}_1 + \mathbf{W}_2 + \mathbf{W}_3$ *in a distributed lossless fashion for the subsequent set of constraints:*

$$\begin{aligned} R_1 &\geq H_{G_{X_1}}(X_1 \,|\, X_2,\, X_3) = 0 \\ \sum_{i=1}^{2} R_i &\geq H_{G_{X_1},G_{X_2}}(X_1,\, X_2 \,|\, X_3) = H\Big(\sum_{k=1}^{3} W_{k,\{1,2\}}\Big) = \frac{1}{3} \\ \sum_{i=1}^{3} R_i &\geq H_{G_{X_1}}(X_1) + H_{G_{X_2}}(X_2 \,|\, X_1) + H_{G_{X_3}}(X_3 \,|\, X_1,\, X_2) \\ &= H\Big(\sum_{k=1}^{3} W_{k,\{1,2\}},\, \sum_{k=1}^{3} W_{k,\{1,3\}}\Big) + H\Big(\sum_{k=1}^{3} W_{k,\{2,3\}}\Big) = \frac{2}{3} + \frac{1}{3} = 1\ , \end{aligned}$$

where the second and third inequalities follow from (B.25) that yields $H(W_{k,\tau}) = \frac{1}{3}$*, given that* $H(W_k) = 1$ *and each* $W_{k,\tau}$ *is a disjoint subset of* W_k*, and* $H_{G_{X_3}}(X_3 \,|\, X_1,\, X_2) = 0$.

We observe that the sum rate needed for computing $f(W_{\mathcal{K}}) = W_1 + W_2 + W_3$ is halved under Maddah-Ali Niesen placement versus cyclic placement.

For each $k \in [K]$, subset $W_{k,\tau}$ is placed in the cache of server i if $i \in \tau$. Thus, each server stores a total of $K\binom{N-1}{N\gamma-1}$ subsets. Since the size of each subset is $1/\binom{N}{|\tau|}$, each server requires a memory size $M = K\binom{N-1}{N\gamma-1}/\binom{N}{N\gamma} = K\gamma$ to store a fraction of γ of the entire dataset, where $\gamma > 1 - N_r/N$. For this setting, the server assignments are as follows [28]:

$$X_i = \{W_{k,\tau} : \tau \ni i,\ \tau \subset [N],\ |\tau| = \gamma N,\ k \in [K]\}\,, \quad i \in \Omega\ . \tag{B.26}$$

Example B.3 (K = 3-multi-linear function under Maddah-Ali Niesen placement model)**.** *Given a topology,* $\mathcal{T}(3, 3, 1, 2, 2)$*, we investigate the sum rate needed for computing the*

multi-linear function

$$f(W_{\mathcal{K}}) = \prod_{k=1}^{3} W_k \ .$$

The dataset assignment phase is similar to the procedure described in Example B.2. By deducing $\gamma = 2/3$ *and* $|\tau| = N\gamma = 2$*, we assign labels to the subsets of each sub-function according to (B.24). Thus,*

$$W_k = (W_{k,\{1,2\}},\, W_{k,\{2,3\}},\, W_{k,\{1,3\}}) \ . \tag{B.27}$$

Then, using the symmetry of placement across servers, it is adequate to establish the rate region for the function using the following set of constraints:

$$R_1 \leq H_{G_{X_1}}(X_1 \,|\, X_2, X_3) = 0 \ , \tag{B.28}$$

$$\begin{aligned}
\sum_{i=1}^{2} R_i &\leq H_{G_{X_1},G_{X_2}}(X_1,\, X_2 \,|\, X_3) = H\left(\prod_{k=1}^{k=3} W_{k,\{1,2\}}\right) \ , \\
\sum_{i=1}^{3} R_i &\leq H_{G_{X_1}}(X_1) + H_{G_{X_2}}(X_2 \,|\, X_1) + H_{G_{X_3}}(X_3 \,|\, X_1,\, X_2) \\
&= H\left(\prod_{k=1}^{3} W_{k,\{1,2\}},\, \prod_{k=1}^{3} W_{k,\{1,3\}}\right) + H\left(\prod_{k=1}^{3} W_{k,\{2,3\}}\right) + 0 \\
&\overset{(a)}{=} 3H\left(\prod_{k=1}^{3} W_{k,\{1,2\}}\right) \ ,
\end{aligned} \tag{B.29}$$

where in (a) *we exploited that* $W_{k,\tau}$ *are i.i.d., and the recovery threshold is* $N_r = 2$*. If instead, we were to compute the same demanded function under the cyclic placement assumption with the same topology, the rate region becomes*

$$R_1 \leq H_{G_{X_1}}(X_1 \,|\, X_2, X_3) = 0 \ , \tag{B.30}$$

$$\begin{aligned}
\sum_{i=1}^{2} R_i &\leq H_{G_{X_1},G_{X_2}}(X_1,\, X_2 \,|\, X_3) = H\left(\prod_{k=1}^{2} W_k\right) \ , \\
\sum_{i=1}^{3} R_i &\leq H_{G_{X_1}}(X_1) + H_{G_{X_2}}(X_2 \,|\, X_1) + H_{G_{X_3}}(X_3 \,|\, X_1,\, X_2) \\
&= H\left(\prod_{k=1}^{2} W_k\right) + H\left(W_3\right) + 0 \ .
\end{aligned} \tag{B.31}$$

To ensure a fair comparison of both models, we can adapt the cyclic placement PMF to match the MAN placement distribution. More specifically, we can consider that in the cyclic placement model the probability $\mathbb{P}\left(\prod_{k=1}^{2} W_k = 0\right)$ *equals the probability that the product of corresponding fractions of subfunction* $W_{k,\tau}$ *in the MAN setting equals zero:*

$$\mathbb{P}\left(\prod_{k=1}^{2} W_k = 0\right) = \mathbb{P}\left(\prod_{k=1}^{2} W_{k,\{1,2\}} = 0, \prod_{k=1}^{2} W_{k,\{1,3\}} = 0, \prod_{k=1}^{2} W_{k,\{2,3\}} = 0\right) \ . \tag{B.32}$$

Using this adapted PMF in (B.32) for cyclic placement, the sum rate for cyclic placement given in (B.30) can be rewritten as follows:

$$\begin{aligned}\sum_{i=1}^{3} R_i \leq & H\left(\prod_{k=1}^{2} W_{k,\{1,2\}}, \prod_{k=1}^{2} W_{k,\{1,3\}}, \prod_{k=1}^{2} W_{k,\{2,3\}}\right) \\ & + \left(1 - \mathbb{P}\left(\prod_{k=1}^{2} W_k = 0\right)\right) H\left(W_{3,\{1,2\}}, W_{3,\{1,3\}}, W_{3,\{2,3\}}\right) \ .\end{aligned} \tag{B.33}$$

The first term in the RHS of (B.33) represents the rate required from X_1, and the second term corresponds to the rate at which X_2 transmits. For the given characteristic graph-based approach, the sum rate given by (B.33), hence the communication cost, of the cyclic placement approach adapted according to (B.32), can be proven to be higher compared to the sum rate using MAN data placement in (B.29).

Chapter C

Appendix for Chapter 4

C.1 Proof of Theorem 4.1

To derive R_{ach}, we here use the compress-bin strategy [260, Theorem 11.3] and the encoding scheme from [261], given receivers' side informations, which follows from the Wyner–Ziv model [102]. Consider length n i.i.d realizations of all sources, i.e., $\mathbf{X}_{[N]}^n = \{X_{1,t}, \ldots, X_{N,t}\}_{t\in[n]}$, drawn from $P_{\mathbf{X}_{[N]}}$. Define $\hat{f}_i^n = F_i(U^n, \mathcal{S}_i^n)$, denoting an estimate of f_i^n at user $i \in [K]$, with U^n denoting an MIS of $G^n_{\cup_{i=1}^K f_i}$ built using the length n realizations $\mathbf{X}_{[N]}^n$ and $\{\mathcal{S}_i^n\}_{i\in[K]}$.

Codebook generation The master node generates $q^{nR'}$ i.i.d sequences $U^n(l)$, where $l \in [0, q^{nR'} - 1]$. For $R' > R_{ach}$, the index set $[0, q^{nR'} - 1]$ is uniformly partitioned into $q^{nR_{ach}}$ bins, $\mathcal{B}(m) = [(m-1)q^{n(R'-R_{ach})} : mq^{n(R'-R_{ach})} - 1]$, for $m \in [1, q^{nR_{ach}}]$. The codebook, denoted as $\mathscr{C} = \{m \in [1, q^{nR_{ach}}]\}$, is revealed to all users.

Encoding Given $\mathbf{X}_{[N]}^n$, the master node finds an index $l^* \in [1 : q^{nR'} - 1]$ such that $(U^n(l^*), \mathbf{X}_{[N]}^n) \in A_{\epsilon'}^{(n)}$, where $A_{\epsilon'}^{(n)}$ represents ϵ'-typical sequences $(U^n, \mathbf{X}_{[N]}^n)$ for some $\epsilon' > 0$. If there are multiple such indices, the master randomly selects one of them. If no such index exists, the master selects $l^* = 0$. The master node broadcasts m^* such that $l^* \in \mathcal{B}(m^*)$. Thus, (l^*, m^*) represents the selected indices at the master.

Decoding Let $\epsilon > \epsilon'$. User $i \in [K]$ receives m^*, and finds the unique index $\hat{l}_i \in \mathcal{B}(m^*)$, such that $(U^n(\hat{l}_i), \mathcal{S}_i^n) \in A_\epsilon^{(n)}$. User $i \in [K]$ then computes $F_i(U^n(\hat{l}_i), \mathcal{S}_i^n)$. It declares an error if there is no such $\hat{l}_i$.

Analysis of expected error Let $U^n(\hat{l}_i)$ be the message decoded by user $i \in [K]$. An error event at user $i \in [K]$ is denoted as $\mathcal{E}r_i = \{(U^n(\hat{l}_i), \mathbf{X}_{[N]}^n) \notin A_\epsilon^{(n)}\} = \{\hat{l}_i \neq l^*\}$, which can be categorized into three events:

$$\mathcal{E}r_{i1} = \{(U^n(l), \mathbf{X}_{[N]}^n) \notin A_{\epsilon'}^{(n)}, \text{ for all } l \in [0 : q^{nR'} - 1]\} ,$$

$$\mathcal{E}r_{i2} = \{(U^n(l^*), \mathbf{X}^n_{[N]}) \notin A^{(n)}_\epsilon\} \ ,$$
$$\mathcal{E}r_{i3} = \{(U^n(l), \mathcal{S}^n_i) \in A^{(n)}_\epsilon, \text{ for some } l \in \mathcal{B}(m^*),\ l \neq l^*\} \ .$$

As $n \to \infty$, by covering lemma (cf. [260, Lemma 3.3]), $\mathbb{P}(\mathcal{E}r_{i1}) \to 0$ if $R' > I(\mathbf{X}_{[N]}; U) + \delta(\epsilon')$, where $\delta(\epsilon') > 0$ tends to zero as $\epsilon' \to 0$, by the conditional typically lemma [260, Section 2.5] $\mathbb{P}(\mathcal{E}r_{i2} \cap \mathcal{E}r^c_{i1}) \to 0$, and by packing lemma [260, Lemma 3.1], $\mathbb{P}(\mathcal{E}r_{i3}) \to 0$ if $R' - R_{ach} < I(U; \mathcal{S}_i) - \delta(\epsilon)$. Using these lemmas yields $R_{ach} > I(\mathbf{X}_{[N]}; U \mid \mathcal{S}_i)$ for any $i \in [K]$. Hence, it must hold that

$$R_{ach} \geq \max_{i \in [K]} I(\mathbf{X}_{[N]}; U \mid \mathcal{S}_i) \ , \tag{C.1}$$

where similar techniques exist for lossy source coding with side information, see e.g., the achievability scheme of [261].

Next, we derive the lower bound R_{con}. Given the encoded message B from $\mathbf{X}^n_{[N]}$, user $i \in [K]$ computes $\hat{f}^n_i$ upon receiving B and $\mathcal{S}^n_i$. The t-th realization $f_{i,t}$, for $t \in [n]$, is defined as $f_{i,t} = g_{i,t}(\Upsilon_t, \mathcal{S}_{i,t})$ for some function $g_{i,t}$ where Υ_t is an auxiliary variable defined as $\Upsilon_t = (B, \mathcal{S}^{t-1}_i)$. Letting $t = n$, we obtain

$$\begin{aligned} nR_{ach} \geq & H_q(B) \overset{(a)}{\geq} H_q(B \mid \mathcal{S}^n_i) \overset{(b)}{=} I(\mathbf{X}^n_{[N]}; B \mid \mathcal{S}^n_i) \ , \\ & \overset{(c)}{=} \sum_{t=1}^n I(\mathbf{X}_{[N],t}; B \mid \mathcal{S}^n_i, \mathbf{X}^{t-1}_{[N]}) \ , \\ & \overset{(d)}{=} \sum_{t=1}^n I(\mathbf{X}_{[N],t}; B, \mathbf{X}^{t-1}_{[N]}, \mathcal{S}^{t-1}_i \mid \mathcal{S}_{i,t}) \\ & \overset{(e)}{=} \sum_{t=1}^n I(\mathbf{X}_{[N],t}; \Upsilon_t \mid \mathcal{S}_{i,t}) \ , \end{aligned}$$

where (a) follows from conditioning, (b) from noting that B is function of $\mathbf{X}^n_{[N]}$, i.e., $H_q(B \mid \mathcal{S}^n_i, \mathbf{X}^n_{[N]}) = 0$, (c) from the chain rule of mutual information, (d) is due to the independence of $\mathbf{X}_{[N],t}$ from $(\mathbf{X}^{t-1}_{[N]}, \mathcal{S}^{t-1}_i, \mathcal{S}^n_{i,t+1})$, and (e) from setting the auxiliary variable $\Upsilon_t = (B, \mathbf{X}^{t-1}_{[N]}, \mathcal{S}^{t-1}_i)$.

C.2 Proof of proposition 4.1

Assume that the master generates a broadcast message B that satisfies (4.1) for all users. Since each user $i \in [K]$ can calculate f'_i from the received message, we have

$$H_q(f'_1, f'_2, \ldots, f'_K \mid B) = 0 \ . \tag{C.2}$$

Next, consider the mutual information between B and f'_i,

$$I(f'_1, \ldots, f'_K; B) = H_q(f'_1, \ldots, f'_K) - H_q(f'_1, \ldots, f'_K \mid B),$$

$$\stackrel{(a)}{=} H_q(f'_1, \ldots, f'_K) \ , \tag{C.3}$$

where (a) follows from (C.2). By expanding (C.3) we have

$$I(f'_1, \ldots, f'_K; B) = H_q(B) - H_q(B \mid f'_1, \ldots, f'_K) \ . \tag{C.4}$$

From (C.3) and (C.4),

$$H_q(f'_1, \ldots, f'_K) \leq H_q(B) \leq R \ . \tag{C.5}$$

Bibliography

[1] Y. Yao and S. A. Jafar, "The Capacity of 3 User Linear Computation Broadcast," *IEEE Trans. Inf. Theory*, vol. 70, no. 6, pp. 4414–4438, Apr. 2024.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, Apr 2017, pp. 1273–1282.

[3] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated Learning: Strategies for Improving Communication Efficiency," *arXiv preprint arXiv:1610.05492*, Oct. 2018.

[4] G. Neglia, G. Calbi, D. Towsley, and G. Vardoyan, "The Role of Network Topology for Distributed Machine Learning," in *Proc., IEEE INFOCOM*. Paris, France: IEEE, Apr. 2019, pp. 2350–2358.

[5] B. Nazer and M. Gastpar, "Computation Over Multiple-Access Channels," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3498–3516, Sep. 2007.

[6] H. Hellström, S. Razavikia, V. Fodor, and C. Fischione, "Optimal Receive Filter Design for Misaligned Over-the-Air Computation," in *Proc., Globecom Wkshs.* Kuala Lumpur, Malaysia: IEEE, Dec. 2023, pp. 1529–1535.

[7] Q. Yu, M. Maddah-Ali, and S. Avestimehr, "Polynomial Codes: an Optimal Design for High-Dimensional Coded Matrix Multiplication," in *Proc., Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 4403–4413.

[8] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The Exact Rate-Memory Tradeoff for Caching with Uncoded Prefetching," *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 1281–96, Feb. 2018.

[9] H. Sun and S. A. Jafar, "The Capacity of Private Computation," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3880–3897, Jun. 2019.

[10] H. El-Sayed, S. Sankar, M. Prasad, D. Puthal, A. Gupta, M. Mohanty, and C.-T. Lin, "Edge of things: The big picture on the integration of edge, IoT and the cloud in a distributed computing environment," *IEEE Access*, vol. 6, pp. 1706–1717, Dec. 2017.

[11] M. Di Renzo, K. Ntontin, J. Song, F. H. Danufane, X. Qian, F. Lazarakis, J. De Rosny, D.-T. Phan-Huy, O. Simeone, and R. Zhang, "Reconfigurable intelligent surfaces vs. relaying: Differences, similarities, and performance comparison," *IEEE Open J. Commun. Society*, vol. 1, pp. 798–807, Jun. 2020.

[12] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," *IEEE communications surveys & tutorials*, vol. 22, no. 3, pp. 2031–2063, Apr. 2020.

[13] Ericsson, "Ericsson mobility report – data forecasts," https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/mobile-traffic-forecast, 2025, accessed: August 2025.

[14] Nokia, "Nokia network 2030 traffic forecast," https://www.lightreading.com/broadband/nokia-gets-its-head-round-latest-network-tech-trends, 2024, accessed: August 2025.

[15] H. Kim and I.-C. Park, "High-performance and low-power memory-interface architecture for video processing applications," *IEEE Trans. Circ. and Sys. Video Technology*, vol. 11, no. 11, pp. 1160–1170, Nov. 2001.

[16] E. Kushilevitz and N. Nisan, "Communication complexity," *Press, Cambridge, UK*, vol. 44, pp. 331–360, Jan 1997.

[17] H. Zhao, D. Slock, and P. Elia, "NOMA-Aided Aggregated Coded Caching," in *Proc., Wireless Commun. and Net. Conf. (WCNC)*. Milan, Italy: IEEE, Mar. 2025, pp. 1–6.

[18] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, Jan. 2002.

[19] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc., ACM Int. Wksh. Wire. Sens. networks and applications*, Sep. 2002, pp. 88–97.

[20] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," *Advances in neural information processing systems*, vol. 29, Dec. 2016.

[21] C. D. Hsu, H. Jeong, G. J. Pappas, and P. Chaudhari, "Scalable reinforcement learning policies for multi-agent control," in *Proc., IEEE Int. Conf. Intell. Robots Syst.*, Prague, Czech, Sep. 2021, pp. 4785–4791.

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

[23] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proc., USENIX HotTop Cloud Comp. Wksh.*, Boston, MA, USA, Jun. 2010.

[24] A. K. Karun and K. Chitharanjan, "A review on Hadoop—HDFS infrastructure extensions," in *Proc. IEEE Info. & Commun. Techs.*, Thuckalay, India, Apr. 2013, pp. 132–137.

[25] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly, "Dryad: distributed data-parallel programs from sequential building blocks," in *Proc., ACM SIGOPS/EuroSys on Comp. Sys.*, Mar. 2007, pp. 59–72.

[26] P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, and K. Tzoumas, "Apache flink: Stream and batch processing in a single engine," *The Bulletin of the Technical Committee on Data Engineering*, vol. 38, no. 4, 2015.

[27] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of things journal*, vol. 3, no. 5, pp. 637–646, Jun. 2016.

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

[29] Q. Yu, "Coded computing: A transformative framework for resilient secure private and communication efficient large scale distributed computing," Ph.D. dissertation, Ph. D. dissertation, Dept. Elect. Comput. Eng., University of Southern California, Los Angeles, CA, USA, 2020.

[30] M. Tahmasbi, A. Shahrasbi, and A. Gohari, "Critical graphs in index coding," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 2, pp. 225–235, Dec. 2014.

[31] Z. Bar-Yossef, Y. Birk, T. Jayram, and T. Kol, "Index coding with side information," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1479 – 1494, Mar. 2011.

[32] F. Arbabjolfaei and Y.-H. Kim, "Fundamentals of index coding," *Found. and Trends. Com. and Info. Theo.*, vol. 14, no. 3-4, pp. 163–346, Oct. 2018.

[33] R. Zamir, *Lattice coding for signals and networks: A structured coding approach to quantization, modulation, and multiuser information theory.* Cambridge University Press, Aug. 2014.

[34] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[35] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Sep. 2006.

[36] R. Zamir and T. Berger, "Multiterminal source coding with high resolution," *IEEE Trans. Inf. Theory*, vol. 45, no. 1, pp. 106–117, Aug. 2002.

[37] T. Berger and R. W. Yeung, "Multiterminal source encoding with one distortion criterion," *IEEE Trans. Inf. Theory*, vol. 35, no. 2, pp. 228–236, Mar. 1989.

[38] N. Alon, Y. Matias, and M. Szegedy, "The space complexity of approximating the frequency moments," in *Proc., ACM Symp. Theory of Comput.*, Philadelphia, PA, USA, Jul. 1996, pp. 20–29.

[39] M. Shrivastava, B. Isik, Q. Li, S. Koyejo, and A. Banerjee, "Sketching for Distributed Deep Learning: A Sharper Analysis," *Adv. in Neural Info. Processing Systems*, vol. 37, pp. 6417–6447, Dec. 2024.

[40] N. Shlezinger, Y. C. Eldar, and M. R. Rodrigues, "Hardware-Limited Task-Based Quantization," *arXiv preprint arXiv:1807.08305*, Jul. 2018.

[41] S. Salamatian, N. Shlezinger, Y. C. Eldar, and M. Médard, "Task-based quantization for recovering quadratic functions using principal inertia components," in *Proc., IEEE Int. Symp. Inf. Theory (ISIT)*. IEEE, Jul. 2019, pp. 390–394.

[42] O. Ordentlich and Y. Polyanskiy, "Optimal quantization for matrix multiplication," *arXiv preprint arXiv:2410.13780*, Oct. 2024.

[43] K. Wan, H. Sun, M. Ji, and G. Caire, "Distributed linearly separable computation," *IEEE Trans. Inf. Theory*, vol. 68, no. 2, pp. 1259–1278, Nov. 2021.

[44] K. K. K. Namboodiri, E. Peter, D. Malak, and P. Elia, "Fundamental Limits of Distributed Computing for Linearly Separable Functions," *arXiv preprint arXiv:2509.23447*, Sep. 2025.

[45] A. Khalesi and P. Elia, "Multi-User Linearly-Separable Distributed Computing," *IEEE Trans. Inf. Theory*, vol. 69, pp. 6314–6339, Jun. 2023.

[46] ——, "Perfect Multi-User Distributed Computing," in *Proc., IEEE Int. Symp. Inf. Theory (ISIT)*. IEEE, Jul. 2024, pp. 1349–1354.

[47] K. Wan, D. Tuninetti, M. Ji, G. Caire, and P. Piantanida, "Fundamental limits of decentralized data shuffling," *IEEE Trans. Inf. Theory*, vol. 66, no. 6, pp. 3616–3637, Jan. 2020.

[48] W.-T. Chang and R. Tandon, "On the capacity of secure distributed matrix multiplication," in *Proc., IEEE Global Commun. Conf.*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6.

[49] W. Li, Z. Chen, Z. Wang, S. A. Jafar, and H. Jafarkhani, "Flexible Constructions for Distributed Matrix Multiplication," in *Proc., IEEE Int. Symp. Inf. Theory (ISIT)*, Melbourne, Australia, Jul. 2021, pp. 1576–1581.

[50] J. Wang, Z. Jia, and S. A. Jafar, "Price of precision in coded distributed matrix multiplication: A dimensional analysis," in *Proc., IEEE Inf. Theory Wksh (ITW)*, Kanazawa, Japan, Oct. 2021, pp. 1–6.

[51] H. H. López, G. L. Matthews, and D. Valvo, "Secure MatDot codes: a secure, distributed matrix multiplication scheme," in *Proc., IEEE Inf. Theory Wksh (ITW)*, Mumbai, India, Nov. 2022, pp. 149–154.

[52] R. G. D'Oliveira, S. El Rouayheb, D. Heinlein, and D. Karpuk, "Notes on communication and computation in secure distributed matrix multiplication," in *Proc., IEEE Conf. Commun. and Netw. Secur.*, virtual, Jun. 2020, pp. 1–6.

[53] T. Jahani-Nezhad and M. A. Maddah-Ali, "CodedSketch: A Coding Scheme for Distributed Computation of Approximated Matrix Multiplication," *IEEE Trans. Inf. Theory*, vol. 67, no. 6, pp. 4185–4196, Mar. 2021.

[54] C. Karakus, Y. Sun, S. Diggavi, and W. Yin, "Straggler mitigation in distributed optimization through data encoding," in *Proc., Adv. Neural Inform. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 5434–5442.

[55] K. Wan, H. Sun, M. Ji, D. Tuninetti, and G. Caire, "Cache-Aided Matrix Multiplication Retrieval," *IEEE Trans. Inf. Theory*, vol. 68, no. 7, pp. 4301–4319, Mar. 2022.

[56] S. Dutta, M. Fahim, F. Haddadpour, H. Jeong, V. Cadambe, and P. Grover, "On the Optimal Recovery Threshold of Coded Matrix Multiplication," *IEEE Trans. Inf. Theory*, vol. 66, no. 1, pp. 278–301, Jul. 2019.

[57] Z. Jia and S. A. Jafar, "On the capacity of secure distributed batch matrix multiplication," *IEEE Trans. Inf. Theory*, vol. 67, no. 11, pp. 20–37, Sep. 2021.

[58] M. Aliasgari, O. Simeone, and J. Kliewer, "Private and secure distributed matrix multiplication with flexible communication load," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 2722–2734, Feb. 2020.

[59] M. Goldenbaum, H. Boche, and S. Stańczak, "Nomographic functions: Efficient computation in clustered Gaussian sensor networks," *IEEE Trans. Wirel. Communications*, vol. 14, no. 4, pp. 2093–2105, Dec. 2014.

[60] ——, "Harnessing interference for analog function computation in wireless sensor networks," *IEEE Trans. Sig. Processing*, vol. 61, no. 20, pp. 4893–4906, Jul. 2013.

[61] A. Şahin and R. Yang, "A survey on over-the-air computation," *IEEE Comm. Surveys & Tutorials*, vol. 25, no. 3, pp. 1877–1908, Apr. 2023.

[62] Z. Wang, Y. Zhao, Y. Zhou, Y. Shi, C. Jiang, and K. B. Letaief, "Over-the-Air Computation for 6G: Foundations, Technologies, and Applications," *IEEE Internet of Things Journal*, vol. 11, no. 14, pp. 24 634–24 658, May. 2024.

[63] T. Han and K. Kobayashi, "A dichotomy of functions F(X, Y) of correlated sources (X, Y)," *IEEE Trans. Inf. Theory*, vol. 33, no. 1, pp. 69–76, Jan. 1987.

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

[65] R. Ahlswede and T. Han, "On source coding with side information via a multiple-access channel and related problems in multi-user information theory," *IEEE Trans. Inf. Theory*, vol. 29, no. 3, pp. 396–412, May 1983.

[66] C. Nair and Y. N. Wang, "On optimal weighted-sum rates for the modulo sum problem," in *Proc., IEEE Int. Symp. Inf. Theory (ISIT)*. Los Angeles, CA, USA: IEEE, Jun. 2020, pp. 2416–2420.

[67] D. Krithivasan and S. S. Pradhan, "Distributed source coding using Abelian group codes: A new achievable rate-distortion region," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1495–1519, Feb. 2011.

[68] S. S. Pradhan, A. Padakandla, and F. Shirani, "An Algebraic and Probabilistic Framework for Network Information Theory," *Foundations and Trends® in Communications and Information Theory*, vol. 18, no. 2, pp. 173–379, Dec. 2020.

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

[70] D. Malak, "Distributed Structured Matrix Multiplication," in *Proc., IEEE Int. Symp. Inf. Theory (ISIT)*, Athens, Greece, Jul. 2024.

[71] ——, "Structured Codes for Distributed Matrix Multiplication," *arXiv preprint arXiv:2501.00371*, Jan. 2025.

[72] J. Körner, "Coding of an information source having ambiguous alphabet and the entropy of graphs." in *Proc., Academia Conf. Info. Theory*, Prague,Czech, Sep. 1971, pp. 411–425.

[73] H. Witsenhausen, "The zero-error side information problem and chromatic numbers (corresp.)," *IEEE Trans. Inf. Theory*, vol. 22, no. 5, pp. 592–593, Sep. 1976.

[74] A. Orlitsky and J. R. Roche, "Coding for computing," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, p. 903–917, Mar. 2001.

[75] N. Alon and A. Orlitsky, "Source coding and graph entropies," *IEEE Trans. Inf. Theory*, vol. 42, no. 5, pp. 1329–1339, Sep. 1996.

[76] N. Charpenay, M. Le Treust, and A. Roumy, "Complementary Graph Entropy, AND Product, and Disjoint Union of Graphs," in *Proc., IEEE Int. Symp. Inf. Theory (ISIT)*, Taipei, Taiwan, Jun. 2023, pp. 2446–2451.

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

[78] D. Malak, "Fractional graph coloring for functional compression with side information," in *Proc., IEEE Inf. Theory Wksh. (ITW)*, Mumbai, India, Nov. 2022, pp. 750–755.

[79] A. C.-C. Yao, "Some complexity questions related to distributive computing (preliminary report)," in *Proc., ACM Symp. Theory Comput.*, Atlanta, GA, Apr. 1979, pp. 209–213.

[80] Z.-Q. Luo, "Communication complexity of some problems in distributed computation," Ph.D. dissertation, MIT, Aug. 1989.

[81] U. Hadar, J. Liu, Y. Polyanskiy, and O. Shayevitz, "Communication complexity of estimating correlations," in *Proc., ACM Symp. Theory of Comput.*, Phoenix, AZ, USA, Jun. 2019, pp. 792–803.

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

[83] V. Doshi, D. Shah, M. Médard, and M. Effros, "Functional Compression Through Graph Coloring," *IEEE Trans. Inf. Theory*, vol. 56, Aug. 2010.

[84] J. Ziv, "On Universal Data Compression—An Intuitive Overview," *J. Visual Communication and Image Representation*, vol. 5, no. 4, pp. 317–321, Dec. 1994.

[85] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2325–2383, Oct. 1998.

[86] N. Merhav, "Lossy Compression of Individual Sequences Revisited: Fundamental Limits of Finite-State Encoders," *Entropy*, vol. 26, no. 2, p. 116, Jan. 2024.

[87] R. Timo, S. S. Bidokhti, M. Wigger, and B. C. Geiger, "A rate-distortion approach to caching," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1957–1976, Oct. 2017.

[88] A. Lapidoth, A. Malär, and M. Wigger, "Constrained source-coding with side information," *IEEE Trans. Inf. Theory*, vol. 60, no. 6, pp. 3218–3237, Apr. 2014.

[89] P. A. Stavrou and M. Kountouris, "The role of fidelity in goal-oriented semantic communication: A rate distortion approach," *IEEE Trans. Communications*, vol. 71, no. 7, pp. 3918–3931, May. 2023.

[90] G. Serra, P. A. Stavrou, and M. Kountouris, "Computation of rate-distortion-perception function under f-divergence perception constraints," in *Proc., IEEE Int. Symp. Inf. Theory (ISIT)*, Taipei, Taiwan, Jun. 2023, pp. 531–536.

[91] ——, "On the computation of the gaussian rate–distortion–perception function," *IEEE J. Sel. Areas Info. Theory*, vol. 5, pp. 314–330, Mar. 2024.

[92] ——, "On the rate-distortion-perception function for gaussian processes," *arXiv preprint arXiv:2501.06363*, Jan. 2025.

[93] C. E. Shannon, "A mathematical theory of communication," *Bell Sys. Tech. Journal*, vol. 27, no. 3, pp. 379–423, Jul 1948.

[94] H. Sun and S. A. Jafar, "On the capacity of computation broadcast," *IEEE Trans. Inf. Theory*, vol. 66, no. 6, pp. 3417–3434, Dec. 2019.

[95] J. Ravi and B. K. Dey, "Broadcast function computation with complementary side information," in *Proc., IEEE Global Commun. Conf. (Globecom) Wkshps*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[96] J. Cardinal, S. Fiorini, and G. Joret, "Tight results on minimum entropy set cover," *Algorithmica*, vol. 51, no. 1, pp. 49–60, May. 2008.

[97] J. L. González-Velarde and M. Laguna, "Tabu search with simple ejection chains for coloring graphs," *Annals of Operations Research*, vol. 117, no. 1, pp. 165–174, Nov. 2002.

[98] H. N. Salas, "Gershgorin's theorem for matrices of operators," *Lin. Alg. & Apps.*, vol. 291, no. 1-3, pp. 15–36, Apr. 1999.

[99] N. Alon, "Graph powers," *Contemporary Combinatorics*, vol. 10, pp. 11–28, 2002.

[100] C. Shannon, "The zero error capacity of a noisy channel," *IRE Trans. Inf. Theory*, vol. 2, no. 3, pp. 8–19, Sep. 1956.

[101] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," in *Proc., Int. Conf. Machine Learn.* Sydney, Australia: PMLR, Jul. 2017.

[102] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. 22, no. 1, pp. 1–10, Jan 1976.

[103] M. R. Deylam Salehi and D. Malak, "An achievable low complexity encoding scheme for coloring cyclic graphs," in *Proc., Allerton Conf. Commun., Control, and Comput.*, Sep. 2023, pp. 1–8.

[104] ——, "Graph-Theoretic Limits of Distributed Computation: Entropy, Eigenvalues, and Chromatic Numbers," *Entropy*, Jul. 2025.

[105] M. Pawlak, E. Rafajlowicz, and A. Krzyzak, "Postfiltering versus prefiltering for signal recovery from noisy samples," *IEEE Trans. Inf. Theory*, vol. 49, no. 12, pp. 3195–3212, Dec. 2003.

[106] V. Doshi, D. Shah, M. Medard, and S. Jaggi, "Distributed functional compression through graph coloring," in *Proc., IEEE Data Compr. Conf.*, Snowbird, UT, Mar. 2007, pp. 93–102.

[107] R. Ahlswede and J. Korner, "Source coding with side information and a converse for degraded broadcast channels," *IEEE Trans. Inf. Theory*, vol. 21, no. 6, pp. 629–637, Nov. 1975.

[108] T. P. Coleman, A. H. Lee, M. Médard, and M. Effros, "Low-complexity approaches to Slepian-Wolf near-lossless distributed data compression," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3546–3561, Aug. 2006.

[109] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," *IEEE Trans. Inf. Theory*, vol. 49, no. 3, pp. 626–643, Mar. 2003.

[110] D. Malak, M. R. Deylam Salehi, B. Serbetci, and P. Elia, "Multi-server multi-function distributed computation," *Entropy*, vol. 26, no. 6, p. 448, Jun. 2024.

[111] ——, "Multi-Functional Distributed Computation," in *Proc., Allerton Conf. Commun., Control, and Comput.*, Urbana, IL, USA, Sep. 2024, pp. 1–8.

[112] A. Tanha and D. Malak, "The influence of placement on transmission in distributed computing of boolean functions," in *Proc., IEEE Int. Wksh. Signal Proces. Advances in Wireless Commun.*, Lucca, Italy, Sep. 2024.

[113] H. Feng, M. Effros, and S. Savari, "Functional source coding for networks with receiver side information," in *Proc., Allerton Conf. Commun., Control, and Comput.*, Monticello, IL, Sep. 2004, pp. 1419–1427.

[114] H. Yamamoto, "Wyner-Ziv theory for a general function of the correlated sources (corresp.)," *IEEE Trans. Inf. Theory*, vol. 28, no. 5, pp. 803–807, Sep. 1982.

[115] J. Barros and S. D. Servetto, "On the rate-distortion region for separate encoding of correlated sources," in *Proc., IEEE Int. Symp. Inf. Theory (ISIT)*, Yokohama, Japon, Jun. 2003, p. 171.

[116] A. B. Wagner, S. Tavildar, and P. Viswanath, "Rate region of the quadratic gaussian two-encoder source-coding problem," *IEEE Trans. Inf. Theory*, vol. 54, no. 5, pp. 1938–1961, Apr. 2008.

[117] D. Rebollo-Monedero, J. Forne, and J. Domingo-Ferrer, "From t-closeness-like privacy to postrandomization via information theory," *IEEE Trans. Know. and Data Engineering*, vol. 22, no. 11, pp. 1623–1636, 2009.

[118] N. Charpenay, M. Le Treust, and A. Roumy, "Side information design in zero-error coding for computing," *Entropy*, vol. 26, no. 4, p. 338, Apr. 2024.

[119] F. Shirani and S. S. Pradhan, "A new achievable rate-distortion region for distributed source coding," *IEEE Trans. Inf. Theory*, vol. 67, no. 7, pp. 4485–4503, May. 2021.

[120] D. Yuan, T. Guo, B. Bai, and W. Han, "Lossy Computing with Side Information via Multi-Hypergraphs," in *Proc., IEEE Inf. Theory Wksh (ITW)*, Nov. 2022, pp. 344–349.

[121] M. Sefidgaran and A. Tchamkerten, "Computing a function of correlated sources: A rate region," in *Proc., IEEE Int. Symp. Inf. Theory (ISIT)*, St. Petersburg, Russia, Jul. 2011, pp. 1856–1860.

[122] A. J. Pettofrezzo and D. R. Byrkit, *Elements of number theory.* Prentice-Hall, 1970.

[123] C. Paar and J. Pelzl, *Understanding cryptography: textbook for students.* Springer Sci. & Busi. Media, 2009.

[124] W. Diffie and M. E. Hellman, "Multiuser cryptographic techniques," in *Proc., National Comp.*, New York, NY, USA, Jun 1976, pp. 109–112.

[125] D. Selent, "Advanced encryption standard," *Rivier Academic Journal*, vol. 6, no. 2, pp. 1–14, Sep. 2010.

[126] S. Basu, "International data encryption algorithm (idea)–a typical illustration," *J. Glob. Research Comp. Sci.*, vol. 2, no. 7, pp. 116–118, Aug. 2011.

[127] G. Knill, "Applications: International standard book numbers," *The Math. Teachers*, vol. 74, no. 1, pp. 47–48, Jan. 1981.

[128] J. Friedman and J.-P. Tillich, "Generalized Alon–Boppana Theorems and Error-Correcting Codes," *SIAM J. Disc. Mathematics*, vol. 19, no. 3, pp. 700–718, May 2005.

[129] N. Berestycki, E. Lubetzky, Y. Peres, and A. Sly, "Random walks on the random graph," *The Annals of Probability*, vol. 46, no. 1, pp. 456–490, Oct. 2018.

[130] N. Kahale, "Better expansion for ramanujan graphs," in *Proc., IEEE Symp. of Found. Comp. Sci.*, Oct. 1991, pp. 398–404.

[131] ——, "On the second eigenvalue and linear expansion of regular graphs," in *Proc., Annu. Symp. Found. of Comp. Science*, vol. 33, Cambridge, MA, USA, Oct. 1992, pp. 296–296.

[132] P. Felber, P. Kropf, E. Schiller, and S. Serbu, "Survey on load balancing in peer-to-peer distributed hash tables," *IEEE Com. Surv. & Tut.*, vol. 16, no. 1, pp. 473–492, 2013.

[133] A. Ganesh, L. Massoulié, and D. Towsley, "The effect of network topology on the spread of epidemics," in *Proc., IEEE Annu. Joint Conf. Comput. Commun. Society*, vol. 2, Miami, FL, USA, Mar. 2005, pp. 1455–1466.

[134] A. Tentes, "Expander graphs, randomness extractors and error correcting codes," Ph.D. dissertation, University of Patras, May 2009.

[135] G. Szabó and G. Fath, "Evolutionary games on graphs," *Physics Reports*, vol. 446, no. 4, pp. 97–216, 2007.

[136] J. You, J. M. Gomes-Selman, R. Ying, and J. Leskovec, "Identity-aware graph neural networks," in *Proc., the AAAI conf. on AI*, vol. 35, no. 12, May. 2021, pp. 737–745.

[137] C. Tretter, *Spectral theory of block operator matrices and applications.* World Scientific, Oct. 2008.

[138] V. Havel, "A remark on the existence of finite graphs," *Casopis Pro Pestovani Matematiky*, vol. 80, pp. 477–480, 1955.

[139] R. Beigel, "Finding maximum independent sets in sparse and general graphs," in *Proc., ACM-SIAM Symp. on Discrete Algorithms (SODA)*, vol. 99, Baltimore, MD, USA, Jan. 1999, pp. 856–857.

[140] W.-K. Chen, *Graph theory and its engineering applications.* World Scientific, 1997, vol. 5.

[141] J. F. Nagle, "On ordering and identifying undirected linear graphs," *J. Math. Phys.*, vol. 7, no. 9, pp. 1588–1592, Sep. 1966.

[142] A. Nilli, "On the second eigenvalue of a graph," *Disc. Mathematics*, vol. 91, no. 2, pp. 207–210, Aug. 1991.

[143] J.-C. Bermond, "Hamiltonian graphs," *Sel. Topics Graph Theo.*, pp. 127–167, 1979.

[144] M. Axenovich, "Lecture notes graph theory," *Karlsruher Institut für Technologie*, 2014.

[145] J. Bang-Jensen and G. Z. Gutin, *Digraphs: theory, algorithms and applications*. Springer Sci. & Bus. Media, Dec. 2008.

[146] J. Körner and A. Orlitsky, "Zero-error information theory," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2207–2229, Oct. 1998.

[147] P. Koulgi, E. Tuncel, S. L. Regunathan, and K. Rose, "On zero-error source coding with decoder side information," *IEEE Trans. Inf. Theory*, vol. 49, no. 1, pp. 99–111, Jan. 2003.

[148] E. R. Scheinerman and D. H. Ullman, *Fractional graph theory: a rational approach to the theory of graphs*. Courier Corporation, Sep. 2011.

[149] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proc., IRE*, vol. 40, no. 9, pp. 1098–1101, Sep. 1952.

[150] B. Aspvall and J. R. Gilbert, "Graph coloring using eigenvalue decomposition," *J. SIAM Alg. Disc. Metheods*, vol. 5, no. 4, pp. 526–538, Dec. 1984.

[151] A. J. Hoffman and L. Howes, "On eigenvalues and colorings of graphs, ii," *Annals NY. Acad. Sciences*, vol. 175, no. 1, pp. 238–242, Jul. 1970.

[152] H. S. Wilf, "The eigenvalues of a graph and its chromatic number," *J. Lond. Math. Society*, vol. 1, no. 1, pp. 330–332, Jan. 1967.

[153] R. C. Brigham and R. D. Dutton, "Bounds on graph spectra," *J. Comb. Theory-S.B*, vol. 37, no. 3, pp. 228–234, Dec. 1984.

[154] Y. Hong, "Bounds of eigenvalues of a graph," *Acta Math. App. Sinica*, vol. 4, no. 2, pp. 165–168, May. 1988.

[155] K. C. Das and P. Kumar, "Some new bounds on the spectral radius of graphs," *Disc. Mathematics*, vol. 281, no. 1-3, pp. 149–161, Apr. 2004.

[156] M. Ajtai, J. Komlós, and E. Szemerédi, "Sorting in c log n Parallel Steps," *Combinatorica*, vol. 3, pp. 1–19, 1983.

[157] S. Arora, T. Leighton, and B. Maggs, "On-line algorithms for path selection in a nonblocking network," in *Proc., ACM Symp. Theory of Comput.*, 1990, pp. 149–158.

[158] M. Bellare, O. Goldreich, and S. Goldwasser, "Randomness in interactive proofs," *Comp. Complexity*, vol. 3, pp. 319–354, 1993.

[159] M. Ajtai, J. Komlós, and E. Szemerédi, "Deterministic simulation in logspace," in *Proc., ACM Symp. Theory of Comput.*, 1987, pp. 132–140.

[160] R. M. Tanner, "Explicit concentrators from generalized n-gons," *SIAM J. Alg. Disc. Methods*, no. 3, pp. 287–293, 1984.

[161] F. R. Chung, "Diameters and eigenvalues," *Journal of the American Mathematical Society*, vol. 2, no. 2, pp. 187–196, 1989.

[162] D. S. Watkins, "Understanding the QR algorithm," *SIAM review*, vol. 24, no. 4, pp. 427–440, 1982.

[163] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, Oct. 2012.

[164] C. Echeverría, J. Liesen, and R. Nabben, "Block diagonal dominance of matrices revisited: bounds for the norms of inverses and eigenvalue inclusion sets," *Lin. Alg. and Apps.*, vol. 553, pp. 365–383, Sept. 2018.

[165] K. Mehlhorn, H. Sun, P. Kolev, and L. Zanetti, *Great Ideas in Theoretical Computer Science*. Max Planck inst., 2014.

[166] J. G. Francis, "The QR transformation a unitary analogue to the LR transformation," *The Compu. Journal*, vol. 4, no. 3, pp. 265–271, Jan. 1961.

[167] C. Yang, H. Wu, Q. Huang, Z. Li, and J. Li, "Using spatial principles to optimize distributed computing for enabling the physical science discoveries," *Proc., Natl. Acad. Sci. U.S.A.*, vol. 108, no. 14, pp. 5498–5503, Apr. 2011.

[168] J. Shamsi, M. A. Khojaye, and M. A. Qasmi, "Data-intensive cloud computing: Requirements, expectations, challenges, and solutions," *J. Grid Computing*, vol. 11, no. 2, pp. 281–310, Jun. 2013.

[169] H. Yang, T. Ding, and X. Yuan, "Federated learning with lossy distributed source coding: Analysis and optimization," *IEEE Trans. Commun.*, vol. 71, no. 8, pp. 4561–4576, May 2023.

[170] G. Gan, "Evaluation of room air distribution systems using computational fluid dynamics," *Energy Build.*, vol. 23, no. 2, pp. 83–93, Dec. 1995.

[171] Y. Gao, L. Wang, and J. Zhou, "Cost-efficient and quality of experience-aware provisioning of virtual machines for multiplayer cloud gaming in geographically distributed data centers," *IEEE Access*, vol. 7, pp. 142 574–142 585, Sep. 2019.

[172] C. Lushbough and V. Brendel, "An overview of the BioExtract Server: a distributed, Web-based system for genomic analysis," in *Proc., Adv. Comput. Biology.* New York, NY: Springer, Jan. 2010, pp. 361–369.

[173] K. Grolinger, M. Hayes, W. A. Higashino, A. L'Heureux, D. S. Allison, and M. A. Capretz, "Challenges for MapReduce in Big Data," in *Proc., IEEE World Congress Serv.*, Anchorage, AK, USA, Jun. 2014, pp. 182–189.

[174] M. A. Al-Khasawneh, S. M. Shamsuddin, S. Hasan, and A. A. Bakar, "MapReduce a comprehensive review," in *Proc., Inter. Smart Comput. Elec. Enter.*, Shah Alam, Malaysia, Jul. 2018, pp. 1–6.

[175] A. Khumoyun, Y. Cui, and L. Hanku, "Spark based distributed deep learning framework for big data applications," in *Proc., Int. Inform. Sci. Commun. Technologies*, Tashkent, Uzbekistan, Nov. 2016, pp. 1–5.

[176] A.-C. Orgerie, M. D. d. Assuncao, and L. Lefevre, "A survey on techniques for improving the energy efficiency of large-scale distributed systems," *ACM Comput. Surveys*, vol. 46, no. 4, pp. 1–31, Mar. 2014.

[177] R. Keralapura, G. Cormode, and J. Ramamirtham, "Communication-efficient distributed monitoring of thresholded counts," in *Proc., ACM SIGMOD Int. Conf. Management of Data*, New York, NY, USA, Jun. 2006, p. 289–300.

[178] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. Leung, "Distributed resource allocation and computation offloading in fog and cloud networks with non-orthogonal multiple access," *IEEE Trans. Veh. Technology*, vol. 67, no. 12, pp. 12 137–12 151, Sep. 2018.

[179] M. Noormohammadpour and C. S. Raghavendra, "Datacenter traffic control: Understanding techniques and tradeoffs," *IEEE Commun. Surv. and Tutorials*, vol. 20, no. 2, pp. 1492–1525, Dec. 2017.

[180] N. Shivaratri, P. Krueger, and M. Singhal, "Load distributing for locally distributed systems," *Computer*, vol. 25, no. 12, pp. 33–44, Dec. 1992.

[181] A. Bestavros, "Demand-based document dissemination to reduce traffic and balance load in distributed information systems," in *Proc., IEEE Symp. Parallel Distrib. Process.*, San Antonio, Texas, USA, Oct. 1995, pp. 338–345.

[182] A. Reisizadeh, S. Prakash, R. Pedarsani, and A. S. Avestimehr, "Tree gradient coding," in *Proc., IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 2019, pp. 2808–2812.

[183] E. Ozfatura, D. Gündüz, and S. Ulukus, "Gradient Coding with Clustering and Multi-Message Communication," in *Proc., IEEE Data Science Wksh.*, Minneapolis, MN, USA, Jun. 2019, pp. 42–46.

[184] M. Ye and E. Abbe, "Communication-computation efficient gradient coding," in *Proc., Int. Conf. Machine Learning.* Stockholm, Sweden: PMLR, Jul. 2018, pp. 5610–5619.

[185] W. Halbawi, N. Azizan, F. Salehi, and B. Hassibi, "Improving Distributed Gradient Descent Using Reed-Solomon Codes," in *Proc., IEEE Int. Symp. Inf. Theory (ISIT)*, Vail, Colorado, USA, Jun. 2018, pp. 2027–2031.

[186] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," in *Proc., IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Türkiye, July 2013, pp. 1077–1081.

[187] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. N. Diggavi, "Hierarchical coded caching," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3212–3229, Jun. 2016.

[188] S. Li, S. Supittayapornpong, M. A. Maddah-Ali, and S. Avestimehr, "Coded terasort," in *Proc., IEEE Int. Parallel Distrib. Process. Symp. Wksh.*, Lake Buena Vista, FL, USA, May 2017.

[189] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, "A fundamental tradeoff between computation and communication in distributed computing," *IEEE Trans. Inf. Theory*, vol. 64, no. 1, pp. 109–128, Sep. 2017.

[190] N. Naderializadeh, M. A. Maddah-Ali, and A. S. Avestimehr, "Fundamental limits of cache-aided interference management," *IEEE Trans. Inf. Theory*, vol. 63, no. 5, pp. 3092–107, May 2017.

[191] A. M. Subramaniam, A. Heidarzadeh, and K. R. Narayanan, "Collaborative decoding of polynomial codes for distributed computation," in *Proc., IEEE Inf. Theory Wksh (ITW)*, Visby, Sweden, Aug. 2019, pp. 1–5.

[192] R. Yosibash and R. Zamir, "Frame codes for distributed coded computation," in *Proc., Int. Symp. Topics Coding*, Montreal, QC, Canada, Aug. 2021, pp. 1–5.

[193] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4539–51, Sep. 2010.

[194] M. Soleymani, H. Mahdavifar, and A. S. Avestimehr, "Analog lagrange coded computing," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 1, pp. 283–295, Feb. 2021.

[195] J. Zhu, S. Li, and J. Li, "Information-theoretically private matrix multiplication from MDS-coded storage," *IEEE Trans. Inf. Forens. and Secur.*, vol. 18, pp. 1680–1695, Feb. 2023.

[196] A. B. Das, A. Ramamoorthy, and N. Vaswani, "Efficient and Robust Distributed Matrix Computations via Convolutional Coding," *IEEE Trans. Inf. Theory*, vol. 67, no. 9, pp. 6266–6282, Jul. 2021.

[197] C. Karakus, Y. Sun, S. Diggavi, and W. Yin, "Straggler mitigation in distributed optimization through data encoding," *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1920–1933, Jan. 2020.

[198] A. Fawzi, M. Balog, A. Huang, T. Hubert, B. Romera-Paredes, M. Barekatain, A. Novikov, F. J. R Ruiz, J. Schrittwieser, and G. Swirszcz, "Discovering faster

matrix multiplication algorithms with reinforcement learning," *Nature*, vol. 610, no. 7930, pp. 47–53, Oct. 2022.

[199] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5227–5239, Jul. 2011.

[200] E. Cancès and G. Friesecke, *Density Functional Theory: Modeling, Mathematical Analysis, Computational Methods, and Applications*, 1st ed. Springer Nature, Jul. 2023.

[201] O. A. Hanna, Y. H. Ezzeldin, T. Sadjadpour, C. Fragouli, and S. Diggavi, "On distributed quantization for classification," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 237–249, Apr. 2020.

[202] P. Luo, H. Xiong, K. Lü, and Z. Shi, "Distributed classification in peer-to-peer networks," in *Proc., ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, San Jose, CA, Aug. 2007, pp. 968–976.

[203] Z. Jia and S. A. Jafar, "Cross subspace alignment codes for coded distributed batch computation," *IEEE Trans. Inf. Theory*, vol. 67, no. 5, pp. 2821–2846, Mar. 2021.

[204] M. R. Deylam Salehi and H. Tavakoli, "Achieving secure communication over wiretap channels using the error exponent of the polar code." *Engineering Letters*, vol. 30, no. 1, Mar. 2022.

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

[206] M. Monagan and R. Pearce, "Parallel sparse polynomial multiplication using heaps," in *Proc., Int. Symp. Symb. Algebr.*, New York, NY, USA, Jun. 2009, pp. 263–270.

[207] W. Huang, K. Wan, H. Sun, M. Ji, R. C. Qiu, and G. Caire, "Fundamental Limits of Distributed Linearly Separable Computation under Cyclic Assignment," in *Proc., IEEE Int. Symp. Inf. Theory (ISIT)*, Taipei, Taiwan, Jun. 2023.

[208] K. Wan, H. Sun, M. Ji, and G. Caire, "On Secure Distributed Linearly Separable Computation," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 3, pp. 912–926, Jan. 2022.

[209] T. Cover, "A proof of the data compression theorem of slepian and wolf for ergodic sources (corresp.)," *IEEE Trans. Inf. Theory*, vol. 21, no. 2, pp. 226–228, Mar. 1975.

[210] H. Yamamoto, "Wyner-Ziv theory for a general function of the correlated sources," *IEEE Trans. Inf. Theory*, vol. 28, pp. 803–807, Sep. 1982.

[211] K. Wan, H. Sun, M. Ji, D. Tuninetti, and G. Caire, "Cache-Aided General Linear Function Retrieval," *Entropy*, vol. 23, no. 1, p. 25, Dec. 2020.

[212] A. Khalesi and M. R. D. Salehi, "Typical solutions of multi-user linearly-decomposable distributed computing," *IEEE Networking Letters*, pp. 1–1, Dec. 2025.

[213] K. Wan, H. Sun, M. Ji, and G. Caire, "On the Tradeoff Between Computation and Communication Costs for Distributed Linearly Separable Computation," *IEEE Trans. Commun.*, vol. 69, no. 11, pp. 7390–7405, Aug. 2021.

[214] B. J. Erickson, P. Korfiatis, Z. Akkus, and T. L. Kline, "Machine learning for medical imaging," *Radiographics*, vol. 37, no. 2, pp. 505–515, Mar. 2017.

[215] N. M. Correa, T. Adali, Y.-O. Li, and V. D. Calhoun, "Canonical Correlation Analysis for Data Fusion and Group Inferences," *IEEE Signal Process. Mag.*, vol. 27, no. 4, pp. 39–50, Jun. 2010.

[216] G. Kant and K. S. Sangwan, "Predictive modeling for power consumption in machining using artificial intelligence techniques," *Procedia CIRP*, vol. 26, no. 1, pp. 403–407, Jan. 2015.

[217] D. Malak, "Weighted graph coloring for quantized computing," in *Proc., IEEE Int. Symp. Inf. Theory (ISIT)*, Taipei, Taiwan, Jun. 2023, pp. 2290–2295.

[218] T. Maugey, M. Rizkallah, N. Mahmoudian Bidgoli, A. Roumy, and C. Guillemot, "Graph spectral 3D image compression," *Graph. Spec. Image. Proc.*, pp. 105–128, Aug. 2021.

[219] J. L. Sevilla, V. Segura, A. Podhorski, E. Guruceaga, J. M. Mato, L. A. Martinez-Cruz, F. J. Corrales, and A. Rubio, "Correlation between gene expression and GO semantic similarity," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 2, no. 4, pp. 330–338, Nov. 2005.

[220] D. Mosk-Aoyama and D. Shah, "Fast Distributed Algorithms for Computing Separable Functions," *IEEE. Trans. Info. Theory*, vol. 54, no. 7, pp. 2997–3007, Jun. 2008.

[221] G. Kaur, "A comparison of two hybrid ensemble techniques for network anomaly detection in Spark distributed environment," *J. Inform. Secur. Appl.*, vol. 55, no. 102601, Dec. 2020.

[222] J. Chen, J. Li, R. Huang, K. Yue, Z. Chen, and W. Li, "Federated learning for bearing fault diagnosis with dynamic weighted averaging," in *Proc., Int. Conf. Sensing, Meas. & Data Analytics AI*, Nanjing, China, Oct. 2021, pp. 1–6.

[223] J. Zhao, R. Govindan, and D. Estrin, "Computing aggregates for monitoring wireless sensor networks," in *Proc., IEEE Int. Wksh Sens. Net. Protocols Appl.*, Anchorage, AK, USA, May. 2003, pp. 139–148.

[224] P. Giselsson and A. Rantzer, *Large-scale and distributed optimization: An introduction*, 1st ed. Springer, 2018, vol. 2227.

[225] S. Kavadias and R. O. Chao, *Resource allocation and new product development portfolio management*, 1st ed. Elsevier/Butterworth-Heinemann, Oxford, UK, Nov. 2007.

[226] C. A. Diniz, M. H. Tutia, and J. G. Leite, "Bayesian analysis of a correlated binomial model," *Braz. J. Probab. Stat.*, vol. 24, no. 1, p. 68–77, Mar. 2010.

[227] P. J. Boland, F. Proschan, and Y. Tong, "Some majorization inequalities for functions of exchangeable random variables," *Lect. Not.-Mono. Ser.*, pp. 85–91, Jan. 1990.

[228] M. R. Deylam Salehi, V. K. K. Purakkal, and D. Malak, "Non-Linear Function Computation Broadcast," *preprint arXiv:2502.13688*, Feb. 2025.

[229] ——, "Non-linear function computation broadcast," *Proc., IEEE Int. Symp. Inf. Theory (ISIT)*, Jun 2025.

[230] J. B. Predd, S. B. Kulkarni, and H. V. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 56–69, Jul. 2006.

[231] J. Montalban, R. Cabrera, E. Iradier, P. Angueira, Y. Wu, L. Zhang, W. Li, and Z. Hong, "Broadcast core-network: Converging broadcasting with the connected world," *IEEE Trans. on Broadcast.*, vol. 67, no. 3, pp. 558–569, Aug. 2021.

[232] A. Kate and I. Goldberg, "Distributed private-key generators for identity-based cryptography," in *Proc., Springer Int. Conf. Security and Crypto. for Networks*, Sep. 2010, pp. 436–453.

[233] X. Liu, G. Yang, W. Susilo, J. Tonien, X. Liu, and J. Shen, "Privacy-preserving multi-keyword searchable encryption for distributed systems," *IEEE Trans. Paral. and Dis. Systems*, vol. 32, no. 3, pp. 561–574, Sep. 2020.

[234] S. Li, Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Coded distributed computing: Fundamental limits and practical challenges," in *Proc., Asilomar Conf. Sig. Sys. Comp. (ACSSC)*, Nov. 2016, pp. 509–513.

[235] B. S. Chlebus, L. Gçasieniec, A. Östlin, and J. M. Robson, "Deterministic radio broadcasting," in *Proc., Springer Auto. Lang. Prog.*, Geneva, Switzerland, Jul. 2000, pp. 717–729.

[236] U. Reimers, "Digital video broadcasting," *IEEE Communic. Mag.*, vol. 36, no. 6, pp. 104–110, Jun. 1998.

[237] A. Dupin and S. Abelard, "Broadcast Encryption using Sum-Product decomposition of Boolean functions," Cryptology ePrint Archive, Paper 2024/154, 2024.

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

[239] Y. Yao and S. A. Jafar, "On the generic capacity of K-user symmetric linear computation broadcast," *IEEE Trans. Inf. Theory*, vol. 70, no. 5, pp. 3693–3717, Mar. 2024.

[240] J. N. Tsitsiklis and Z.-Q. Luo, "Communication complexity of convex optimization," *J. Complex.*, vol. 3, no. 3, pp. 231–243, Sep. 1987.

[241] N. Ma, P. Ishwar, and P. Gupta, "Interactive source coding for function computation in collocated networks," *IEEE Trans. Inf. Theory*, vol. 58, no. 7, pp. 4289–4305, Apr. 2012.

[242] N. Ma and P. Ishwar, "Some results on distributed source coding for interactive function computation," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 6180–6195, Aug. 2011.

[243] E. Tuncel, "Slepian-wolf coding over broadcast channels," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1469–1482, Apr. 2006.

[244] R. Gray and A. Wyner, "Source coding for a simple network," *J. Bell Sys. Tech.*, vol. 53, no. 9, pp. 1681–1721, Nov. 1974.

[245] C. T. Li and A. El Gamal, "Extended Gray–Wyner system with complementary causal side information," *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5862–5878, Dec. 2017.

[246] D. Malak, "Distributed computing of functions of structured sources with helper side information," in *Proc., IEEE Int. Wksh. Signal Proces. Advances in Wireless Commun.*, Sep. 2023, pp. 276–280.

[247] E. Özyılkan and E. Erkip, "Distributed compression in the era of machine learning: A review of recent advances," in *Proc., Conf. Info. Sci. and Sys. (CISS)*, Princeton, NJ, USA, Apr. 2024, pp. 1–6.

[248] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1514–1529, Aug. 2017.

[249] L. Cui, X. Su, Y. Zhou, J. Liu, and S. Wen, "Toward Optimized Federated Learning With Compressed Communications by Rate Adaption," *IEEE Trans. Networking*, Jan. 2025.

[250] A. Gamal and T. Cover, "Achievable rates for multiple descriptions," *IEEE Trans. Inf. Theory*, vol. 28, no. 6, pp. 851–857, Nov. 1982.

[251] Y. Yao and S. A. Jafar, "On the capacity of vector linear computation over a noiseless quantum multiple access channel with entangled transmitters," *IEEE Trans. Quant. Engineering*, Oct. 2025.

[252] J. Körner, "Fredman–Komlós bounds and information theory," *J. SIAM on Alg. Disc. Methods*, vol. 7, no. 4, pp. 560–570, Oct. 1986.

[253] M. R. Deylam Salehi, "Miss of the broadcast graph in nonlinear function computation broadcast setting," https://github.com/RezaDeylam/MISBroadcastNonlinear, 2025, version 1.0.0.

[254] S. Jafar, "Capacity with causal and noncausal side information: A unified view," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5468–5474, Nov. 2006.

[255] N. Lee, A. G. Dimakis, and R. W. Heath, "Index coding with coded side-information," *IEEE Communications Letters*, vol. 19, no. 3, pp. 319–322, Mar. 2015.

[256] S. Wilfried Imrich, Klavžar, *Product graphs: structure and recognition*. Wiley, Apr. 2000.

[257] L. Lovász, "Three short proofs in graph theory," *J. Combin. Theory, Series B*, vol. 19, no. 3, pp. 269–271, 1975.

[258] ——, "On the ratio of optimal integral and fractional covers," *Disc. Math.*, vol. 13, no. 4, pp. 383–390, Jan. 1975.

[259] L. Qi, L. Miao, W. Zhao, and L. Liu, "Characterization of graphs with an eigenvalue of large multiplicity," *Adv. Math. Phys.*, no. 1, p. 672, 2020.

[260] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge university press, Dec. 2011.

[261] C. Heegard and T. Berger, "Rate distortion when side information may be absent," *IEEE Trans. Inf. Theory*, vol. 31, no. 6, pp. 727–734, Nov 1985.