Structured Codes for Distributed Matrix Multiplication

Derya Malak

***Abstract*—Our work addresses the well-known open problem of distributed computing of bilinear functions of two correlated sources $\mathbf{A}$ and $\mathbf{B}$. In a setting with two nodes, with the first node having access to $\mathbf{A}$ and the second to $\mathbf{B}$, we establish bounds on the optimal sum rate that allows a receiver to compute an important class of non-linear functions, and in particular bilinear functions, including dot products $\langle \mathbf{A}, \mathbf{B} \rangle$, and general matrix products $\mathbf{A}^{\intercal}\mathbf{B}$ over finite fields. The bounds are tight for large field sizes, for which case we can derive the exact fundamental performance limits for all problem dimensions and a large class of sources. Our achievability scheme involves the design of non-linear transformations of $\mathbf{A}$ and $\mathbf{B}$, carefully calibrated to work synergistically with the structured linear encoding scheme by Körner and Marton. The subsequent converses derived here, calibrate the Han-Kobayashi approach and the strong converse of Ahlswede-Gács-Körner to yield relatively tight converses on the sum rate. We exhibit unbounded compression gains over Slepian-Wolf coding, depending on the source correlations. In the end, this work characterizes the fundamental limits of distributed computing for a crucial class of functions, while succinctly capturing the inherent computation structures and source correlations.**

***Index Terms*—Distributed computation, source coding for compression, structured linear coding, distributed dot-product computation, and distributed matrix multiplication.**

I. Introduction

Basic functions like matrix multiplication, currently constitute the bulk of computational load in scientific computing, as they are omnipresent in applications that include convolution [2], large linear transforms, Fourier transforms, quantum computing [3], as well as in applications of machine learning such as linear regression, least squares modeling [2], and many more. The unprecedented intensity of such computational loads often brings to the fore the necessity for massive parallelization techniques, and we are now witnessing the deployment of massive distributed computing systems, geared at tackling complex distributed computing tasks.

It is the case though that to be successfully deployed, distributed computing requires an intense exchange of information among the participating nodes. In most scenarios, including matrix multiplication, it is evident that to meaningfully parallelize across multiple workers, one must maintain a reduced communication load, which is now considered as a main bottleneck of parallel processing. The need for minimizing this load is clear and evident, and this is a need that has motivated several of the noteworthy parallel computing techniques, such as in [4]–[10] to mention just a fraction, many of which have been designed and tested with success.

Motivated by the above, we here study the communication cost of distributed computing, and we do so for a prevalent class of non-linear functions, namely of bilinear functions. As suggested above, this setting finds itself at the core of various technological fields in edge and cloud computing [11] and machine learning [12], and again, as suggested, this is a setting that entails considerable communication overheads as well as an intriguingly intertwined relationship between communication and computational parallelization. This bottleneck has been studied in seminal works such as [13]–[17], focusing often on the linear function case. Our work focuses on the classical problem of distributed computing of bilinear functions of two correlated sources, placing emphasis on dot and matrix products, while also capturing an important element of modern large data; the strong structural correlations of this data that serves as computing input. In a context similar to [13], [14], [18] where the receiver wishes to compute bilinear functions of the distributed sources, our aim is to establish bounds on the optimal sum rate (the minimum amount of information) that allows a receiver to compute these functions.

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

A. Main Contributions of this Work

- **New structured source codes for distributed computing of dot and matrix products.** We devise an encoding framework for computing the product of two distributed correlated source variables $\mathbf{A}$ and $\mathbf{B}$ (vectors or matrices), over finite fields. To this end, our achievability scheme involves the novel design of *non-linear transformations* for long sequences of $\mathbf{A}$ and $\mathbf{B}$ drawn i.i.d. across realizations, according to some joint probability distribution. These transformations are then carefully calibrated to *work synergistically with the structured linear encoding scheme by Körner and Marton* [13] as well as the *more general scheme by Ahlswede and Han* [19] for computing a class of bilinear functions, all with a vanishing error probability.

  Our achievability results include constructions for distributed computing of dot products (Corollary 1), matrix products that are symmetric (Propositions 1 and 2, and

The author is with the Commun. Systems Dept., EURECOM, Biot Sophia Antipolis, 06904 France (derya.malak@eurecom.fr).

A preliminary version of this work was presented in part at the 2024 Int. Symp. Inf. Theory, Athens, Greece [1].

This research was partially supported by European Research Council ERC-StG Project SENSIBILITÉ under Grant 101077361, by the Huawei France-Funded Chair Toward Future Wireless Networks, and by the program "PEPR Networks of the Future" of France 2030. Manuscript last revised: May 8, 2026.

Theorem 2), and general square matrix products (Proposition 3, and Theorem 3). We also explore a hybrid encoding scheme (Proposition 4), as well as consider recursive and nested applications of the dot product (Propositions 5-7) for distributed matrix multiplication. Our schemes are flexible, allowing the receiver to recover $\mathbf{A}^\intercal\mathbf{B}$ without imposing structural constraints on $\mathbf{A}$ and $\mathbf{B}$.

- **Achievable compression gains.** Contrasting the sum rates of our structured source codes (see Propositions 1-4 and Theorems 2 and 3) with the state-of-the-art codes (e.g., [18], [20], [21]) reveals significant gains when computing dot products and matrix products (of distributed sources $\mathbf{A}$ and $\mathbf{B}$). Our schemes carefully harness the structure of the source data, and the corresponding sum-rate gains are naturally more pronounced in the presence of stronger correlations (see Examples 1-3, and see also Figure 4 which relates to Propositions 1 and 4).
- **Converse results.** Leveraging the Ahlswede-Gács-Körner approach in [22] allows us here to derive a strong converse for the square matrix product setting, as $q \to \infty$ (Theorem 4). Furthermore, calibrating the Han-Kobayashi approach in [14] yields a relatively tight lower bound on the sum rate for all $q \geq 2$ (Theorem 5). In addition to the main converse bounds, for independently and uniformly drawn source matrices $\mathbf{A}$ and $\mathbf{B}$ when $q \to \infty$, we also derive a matching strong converse to the achievability Theorem 3 for the case of square matrix products (Corollary 5). Finally, for the cases of symmetric and square matrix products over $\mathbb{F}_2$, we upper bound the optimality gaps of our design (see Propositions 8 and 9 respectively).
- Our schemes have some additional properties, like balancing the computation load across the nodes, as well as offering some security advantages. We briefly discuss these in Section V.

B. Related Work and Connections of Our Work to the State of the Art

Below, we detail two main approaches used in distributed coding for computation: unstructured coding-based approaches (also known as random binning, relying on hashing functions), and structured coding-based approaches that are more directly geared towards leveraging the structure of the computation task.

a) Unstructured coding for computing: Most of the early approaches use the idea of *random binning* for lossless source coding. Focusing on the case of two statistically dependent, finite alphabet source variables X_1 and X_2 separately observed by two transmitters, the seminal work by Slepian and Wolf [18] provided an unstructured coding technique for the asymptotically lossless compression of i.i.d. sequences $X_1^n = \{X_{1i}\}_{i=1}^n$ and $X_2^n = \{X_{2i}\}_{i=1}^n$, and established the well-known result that $R_{\text{SW}}^{\Sigma} = H_q(X_1, X_2)$ is the optimal rate for jointly recovering (X_1^n, X_2^n). In some cases though, this sum rate can be significantly reduced when the network's task is to compute a function of the sources rather than to communicate the sources themselves. Taking a step towards distributed computing, Yamamoto derived the minimum rate (guaranteeing vanishing probability of error) at which a source has to compress X_1^n for distributed computing of a general function $f(X_1^n, X_2^n)$ with side information X_2^n at the receiver (cf. [23]). Additional related work can also be found in [24], where the authors exploit characteristic graphs to derive lower bounds on perfect hashing. Drawing on [24], the graph-theoretic approaches in [25]–[27], as well as those in [28], have addressed various computing scenarios.

b) Structured coding for computing: Structured coding approaches, on the other hand, deviate from the random coding approach, and instead entail correlated binning of sources and the use of algebraic codes. Körner and Marton (cf. [13]) devised a *structured linear encoding* strategy for distributed computing the modulo-two sum $X_1 \oplus_2 X_2$ of two i.i.d. binary source sequences (X_1^n, X_2^n), where in particular (X_{1i}, X_{2i}) has the same joint distribution as (X_1, X_2), for all $i \in [n]$. Their technique, based on the method of Elias [29], constructs linear codes that achieve an asymptotically vanishing probability of error at a derived minimum sum rate of $2H(X_1 \oplus_2 X_2)$, when the joint distribution of X_1 and X_2 is symmetric (see also Definition 2). Furthermore, the interesting work in [17] provided subspace-based lossless linear computation schemes using nested codes, where these schemes have been shown to generalize those in [13], as well as have been proven to be sum-rate optimal for a class of source distributions. Additionally, for the binary modulo-two sum problem, Ahlswede and Han also derived — by combining the source coding technique from [30] with the method of Elias [29] — an achievable rate region for general binary sources [19, Theorem 10], which contains the regions derived in [18] and [13], and which is generally larger than the convex hull of both.

Furthermore, Han and Kobayashi generalized the structured encoding strategy in [13] to the setting of computing $X_1 \oplus_q X_2$ given q-ary sources X_1 and X_2, with $\oplus_q$ denoting addition over $\mathbb{F}_q$ [14]. Motivated by the problem of compressing non-additive functions, in [14], the authors also identified key function features that differentiate the Slepian-Wolf and Körner-Marton regions, providing conditions under which computing a general bivariate function $f(X_1^n, X_2^n) = \{f(X_{1i}, X_{2i})\}_{i=1}^n$ requires a rate lower than R_{SW}^{Σ}. Along related lines, Ahlswede and Csiszár demonstrated that computing most binary-valued or componentwise functions of distributed source sequences (X_1^n, X_2^n) necessitates transmission rates from separate encoders that are nearly as high as those needed for full reconstruction of (X_1^n, X_2^n) [31]. Furthermore, this same work also revealed that for a class of functions — like for example, computing the joint type (joint composition) of X_1^n and X_2^n, or computing their Hamming distance or its parity — then determining $f(X_1^n, X_2^n)$ in the knowledge of X_2^n, typically required the encoder of X_1 to use a rate comparable to that needed for fully reproducing X_1^n itself. Additionally, it was also revealed that, given a distortion criterion, the problem of an exact characterization of the achievable rate region for $f(X_1^n, X_2^n)$ — excluding componentwise functions — may be as hard a problem as determining the achievable rate region for reproducing X_1^n and X_2^n. In distributed computation of non-linear functions of (X_1, X_2), identifying an injective mapping from the target function to a representation $X_1 \oplus_q X_2$, defined

over a sufficiently large prime field $\mathbb{F}_q$ [20], [21], [32], known as *function embedding*, followed by *structured binning*, may yield rate savings over [18].

As our interest lies in distributed matrix multiplication, we proceed to provide some of the state of art related to this broad problem.

c) Distributed matrix multiplication and codes: Coded matrix multiplication recasts matrix multiplication tasks into a computation channel. Numerous strategies have been developed to enhance distributed coded matrix multiplication to reduce download costs and mitigate stragglers, such as Short-Dot [33], Polynomial [4], MatDot [34], and PolyDot codes [5], [35], [36], all over finite fields. These approaches split source matrices into submatrices via linear transformations and transform matrix multiplication into inner or outer product computations. Distributing subtasks across worker nodes enables efficient, linearly separable processing of rows and columns of matrices. For example, MatDot [34] and PolyDot codes [5] reduce communication costs and improve security, while Polynomial codes [4], as well as generalizations using algebraic function fields [37], namely finite field extensions of fields of rational functions [38], are commonly used to mitigate stragglers. Recent research focuses on achieving even greater reductions in communication costs (e.g., [39]–[43]). However, these approaches are not sum-rate optimal, even in the absence of stragglers. Furthermore, in the absence of stragglers, for Polynomial coding approaches, the user can directly recover the source matrices from subtasks.

Our work builds on the foundational principles of structured codes to develop distributed matrix multiplication techniques over finite fields. We demonstrate significant compression savings for matrix product computations compared to [18], achieved by applying structured encoding to carefully designed non-linear mappings of the distributed source matrices. In addition to improved performance, our approach operates over smaller field sizes than those required in [20], [21], [32]. This use of structured coding idea in the context of distributed matrix multiplication will prove pivotal in capturing source correlations and computation structures jointly, thus well capturing the matrix multiplication problem.

C. Organization

Section II formalizes the distributed matrix multiplication setting. Section III presents our achievable coding schemes and corresponding rates for computing dot products, symmetric, and square matrix products. It also explores recursive and nested implementations of dot products for computing general matrix products. Section IV details our converse bounds and characterizes the optimality gaps of our schemes, while Section V concludes the paper. Throughout the paper, we present various examples to assist the reader in better understanding the results.

Notation. We use regular type for random variables and boldface for vectors and matrices over the finite field $\mathbb{F}_q$. Logarithms base 2 and $q > 2$ are denoted by $\log$ and $\log_q$, respectively, and we write $\exp(\cdot) = q^{(\cdot)}$. We use $\oplus_q$ and $\ominus_q$ to denote addition and subtraction over the finite field $\mathbb{F}_q$, respectively, where $x \ominus_q y \triangleq x \oplus_q (-y)$. When q is prime, $\oplus_q$ coincides with modulo-q addition; for $q = p^m$, $m > 1$, it denotes the standard field addition in $\mathbb{F}_{p^m}$. For a random variable X with PMF P_X, its entropy in binary and q-ary units is $H(X)$ and $H_q(X) = H(X)/\log_2 q$, respectively. Likewise, for (X_1, X_2) with joint PMF P_{X_1,X_2}, the joint and conditional entropies are $H_q(X_1, X_2)$ and $H_q(X_1 \mid X_2)$, respectively. The acronym i.i.d. stands for independent and identically distributed, and $X_1 \perp\!\!\!\perp X_2$ is used to describe statistical independence between X_1 and X_2. $\mathbb{P}(A)$ is the probability of an event A. For a binomial variable $X \sim \mathrm{Bin}(l, p)$ with $l \in \mathbb{N}$ and $p \in [0, 1]$, the complementary cumulative distribution function is given by $\bar{F}(m; l, p) = \sum_{i=m}^{l} \binom{l}{i} p^i (1-p)^{l-i}$. When $l = 1$, then X is a Bernoulli variable, denoted $X \sim \mathrm{Bern}(p)$, and $h(p)$ denotes the binary entropy function.

We denote by $[l]$ the set $\{1, \ldots, l\}$, for $l \in \mathbb{Z}^+$, and by $[l_1,\ l_2]$ the set $\{l_1, \ldots, l_2\}$ for $l_1,\ l_2 \in \mathbb{Z}^+$ such that $l_1 \le l_2$. Given a random matrix $\mathbf{X} = (x_{ij})_{i\in[m],\ j\in[l]} \in \mathbb{F}_q^{m\times l}$, its i-th row, j-th column, and transpose are given by $\mathbf{X}(i, :)$, $\mathbf{X}(:, j)$, $\mathbf{X}^\intercal$, respectively. Alternatively, $\mathbf{x} = (x_i)_{i\in[m]} \in \mathbb{F}_q^{m\times 1}$ (or $\mathbb{F}_q^m$) and $\mathbf{x} = \big((x_j)_{j\in[l]}\big)^\intercal \in \mathbb{F}_q^{1\times l}$ denote column and row vectors, respectively. For a given $\mathbf{x} \in \mathbb{F}_q^{1\times l}$, for $i,\ j \in \mathbb{Z}^+$, and $i \le j$, then $\mathbf{x}(i : j) \triangleq [x_i,\ x_{i+1},\ \ldots,\ x_j]$, and similarly for a column vector. The vertical concatenation of $\mathbf{A} \in \mathbb{F}_q^{m_1\times l}$ and $\mathbf{B} \in \mathbb{F}_q^{m_2\times l}$ is denoted by $[\mathbf{A}; \mathbf{B}] \in \mathbb{F}_q^{(m_1+m_2)\times l}$. The notations $\mathbf{1}_{m\times l}$ and $\mathbf{0}_{m\times l}$ denote $m \times l$ matrices of all ones and all zeros, respectively. We write $X^n \triangleq \{X_i\}_{i=1}^n = (X_1, X_2, \ldots, X_n)$, and use both $[X_1; X_2; \ldots; X_n]$ and $(X_1, X_2, \ldots, X_n)$ to denote a sequence of i.i.d. realizations of X; the intended meaning will be clear from context. We extend this notation to matrices by defining $\mathbf{X}^n$, with $\mathbf{X}^n(i, j)$ representing the length-n sequence of realizations of the (i, j)-th component $\mathbf{X}(i, j) \in \mathbb{F}_q$.

II. System Model and Problem Statement

We consider a distributed scenario involving two sources with separate encoders, and a receiver. The distributed sources separately observe realizations of statistically dependent (i.e., correlated) matrix variables $\mathbf{A} = (a_{ij})_{i\in[m],\ j\in[l]} \in \mathbb{F}_q^{m\times l}$ and $\mathbf{B} = (b_{ij})_{i\in[m],\ j\in[l]} \in \mathbb{F}_q^{m\times l}$, respectively. In other words, Source 1 observes $\mathbf{A}$ and Source 2 observes $\mathbf{B}$, respectively. The receiver aims to compute $\boldsymbol{\mathcal{D}} = (d_{ij})_{i,\ j\in[l]} = \mathbf{A}^\intercal\mathbf{B} \in \mathbb{F}_q^{l\times l}$.

We take a non-real-time approach that relies on accumulating length-n sequences of potentially correlated source matrix realizations. Specifically, the distributed sources are block-encoded with blocklength n. We assume statistically dependent finite alphabet two source sequences $\mathbf{A}^n = (\mathbf{A}(1), \mathbf{A}(2), \ldots, \mathbf{A}(n))$ and $\mathbf{B}^n = (\mathbf{B}(1), \mathbf{B}(2), \ldots, \mathbf{B}(n))$ corresponding to length-n i.i.d. realizations of $\mathbf{A}$ and $\mathbf{B}$, respectively. We have the following additional assumption.

Assumption 1 (Memoryless and i.i.d. observations)**.** *The sequence of pairs $\{(\mathbf{A}(i), \mathbf{B}(i))\}_{i=1}^n$ is i.i.d. according to some joint distribution $P_{\mathbf{A},\mathbf{B}}$, i.e.,*

$$P_{\mathbf{A}^n,\mathbf{B}^n}(\mathbf{A}^n, \mathbf{B}^n) = \prod_{i=1}^{n} P_{\mathbf{A},\mathbf{B}}(\mathbf{A}(i), \mathbf{B}(i)) \ .$$

Two distributed sources separately observe length-n memoryless and i.i.d. realizations (Assumption 1) $\mathbf{A}^n = (\mathbf{A}(1), \mathbf{A}(2), \ldots, \mathbf{A}(n))$ and $\mathbf{B}^n = (\mathbf{B}(1), \mathbf{B}(2), \ldots, \mathbf{B}(n))$, respectively, and then encode their respective realizations independently. The two sources can perform general, and possibly non-linear, componentwise functions $g_1(\mathbf{A}^n) = \{g_1(\mathbf{A}(i))\}_{i=1}^n = \mathbf{X}_1^n$ and $g_2(\mathbf{B}^n) = \{g_2(\mathbf{B}(i))\}_{i=1}^n = \mathbf{X}_2^n$, where each component $\mathbf{X}_1(j)$ and $\mathbf{X}_2(j)$ lies in $\mathbb{F}_q$. As we will see later, in our setting, the encoders will apply the *structured linear coding* technique of Körner-Marton in [13] to the non-linear transformations $\mathbf{X}_1^n \in \boldsymbol{\mathcal{X}}_1^n$ and $\mathbf{X}_2^n \in \boldsymbol{\mathcal{X}}_2^n$, respectively. The separate encoders devise mappings $f_1 : \boldsymbol{\mathcal{X}}_1^n \to \mathcal{R}_{f_1}$ and $f_2 : \boldsymbol{\mathcal{X}}_2^n \to \mathcal{R}_{f_2}$, with ranges $\mathcal{R}_{f_1}$ and $\mathcal{R}_{f_2}$, respectively. The encoder outputs are then transmitted over *noiseless channels* to the common receiver. As we will also see, the receiver will add the received codewords,

$$\mathbf{Z}^n = \mathbf{X}_1^n \oplus_q \mathbf{X}_2^n \tag{1}$$

before proceeding to recover the sequence $\boldsymbol{\mathcal{D}}^n = (\boldsymbol{\mathcal{D}}(1), \boldsymbol{\mathcal{D}}(2), \ldots, \boldsymbol{\mathcal{D}}(n))$ of desired matrix products, with $\boldsymbol{\mathcal{D}}(i) = \mathbf{A}(i)^\intercal \mathbf{B}(i) \in \mathbb{F}_q^{l \times l}$ for all $i \in [n]$, with a small probability of error, as we will clarify later on (see Proposition 1 and its proof).

Definition 1 (An (n, ϵ)-coding scheme [13])**.** *The pair of functions (f_1, f_2) is called an (n, ϵ)-coding scheme if there exists a function $\phi : \mathcal{R}_{f_1} \times \mathcal{R}_{f_2} \to \boldsymbol{\mathcal{Z}}^n$ such that letting*

$$\hat{\mathbf{Z}}^n \triangleq \phi(f_1(\mathbf{X}_1^n), f_2(\mathbf{X}_2^n)) \tag{2}$$

achieves an arbitrarily small probability of error $\mathbb{P}(\hat{\mathbf{Z}}^n \neq \mathbf{Z}^n) < \epsilon$.

In this work, we devise (n, ϵ)-coding schemes (see Section III) that approximate $\boldsymbol{\mathcal{D}}^n$, with accuracy $1 - \epsilon$. The corresponding two-component source network with memoryless and i.i.d. observations is called the *general matrix multiplication source network*, and is shown in Figure 1.

A special case of our setting is when the encoder sequences X_1^n and X_2^n are formed by i.i.d. realizations of distributed sources $X_1 \in \mathbb{F}_2$ and $X_2 \in \mathbb{F}_2$, respectively, where (X_{1i}, X_{2i}) has the same joint distribution as (X_1, X_2). For this case, Körner-Marton in [13] have designed (n, ϵ)-coding schemes for computing $X_1 \oplus_2 X_2$ of a doubly symmetric binary source (DSBS) pair (X_1, X_2), and for any $\epsilon \in (0, 1)$ at rates $R_1 = R_2 = H(X_1 \oplus_2 X_2)$, which achieve the optimum.

Definition 2 (Doubly symmetric binary source (DSBS))**.** *A DSBS with disagreement probability $p \in \big(0, 1/2\big)$ is denoted by $(X_1, X_2) \sim \mathrm{DSBS}(p)$, where its PMF is given as $\mathbb{P}(X_1 = X_2 = 0) = \mathbb{P}(X_1 = X_2 = 1) = (1-p)/2$ and $\mathbb{P}(X_1 = 0,\ X_2 = 1) = \mathbb{P}(X_1 = 1,\ X_2 = 0) = p/2$, where both $X_1 \sim \mathrm{Bern}(1/2)$ and $X_2 \sim \mathrm{Bern}(1/2)$, and where X_2 is the output of a binary symmetric channel (BSC) with a crossover probability p, denoted as $\mathrm{BSC}(p)$, for a given input X_1 [44].*

Our objective, as detailed next, is to determine the *region of achievable rates* for the general matrix multiplication source network to achieve asymptotically[1] lossless compression. We denote the achievable encoding rates by the pair (R_1, R_2). Even when the receiver knows $\mathbf{A}$, $R_2 \geq H_q(\mathbf{A}^\intercal \mathbf{B} \,|\, \mathbf{A})$ is still necessary. Similarly, we must have $R_1 \geq H_q(\mathbf{A}^\intercal \mathbf{B} \,|\, \mathbf{B})$. Hence, the sum rate must satisfy $R_1 + R_2 \geq H_q(\mathbf{A}^\intercal \mathbf{B} \,|\, \mathbf{A}) + H_q(\mathbf{A}^\intercal \mathbf{B} \,|\, \mathbf{B})$, whereas this lower bound may not be achievable with separate encoders in general. In [18], Slepian and Wolf have provided the necessary and sufficient lossless coding rate for distributed compression of $\mathbf{A}^n$ and $\mathbf{B}^n$:

$$\begin{aligned} R_1 &\geq H_q(\mathbf{A} \mid \mathbf{B}) \ , \\ R_2 &\geq H_q(\mathbf{B} \mid \mathbf{A}) \ , \\ R_1 + R_2 &\geq R_{\mathrm{SW}}^{\Sigma}(\mathbf{A}, \mathbf{B}) \triangleq H_q(\mathbf{A}, \mathbf{B}) \ . \end{aligned} \tag{3}$$

In their seminal work [14], Han and Kobayashi derived the necessary and sufficient conditions (restated in Lemma 6) under which any achievable rate $(R_1,\ R_2)$ for the distributed computation of an arbitrary function $f(\mathbf{A}, \mathbf{B})$ of correlated variables $\mathbf{A}$ and $\mathbf{B}$, with entropy $R_{\mathrm{f}} \triangleq H_q(f(\mathbf{A}, \mathbf{B})) \leq R_{\mathrm{SW}}^{\Sigma}$, coincides with the region in (3). These conditions are not always satisfied in our setting, as we demonstrate below. Consequently, we identify regimes where the resulting sum rate $R_{\mathrm{KM}}^{\Sigma} = R_1 + R_2$ is strictly below the Slepian-Wolf bound R_{SW}^{Σ} described by (3), enabling the receiver to recover the matrix product $\boldsymbol{\mathcal{D}} = \mathbf{A}^\intercal \mathbf{B}$ without being able to decode $(\mathbf{A}, \mathbf{B})$ in their entirety. To that end, we denote the gain of our scheme by $\eta \triangleq R_{\mathrm{SW}}^{\Sigma} / R_{\mathrm{KM}}^{\Sigma}$.

III. Achievability

This section details achievable schemes for *structured distributed matrix multiplication*. The first scheme in Proposition 1 considers the symmetric case for $q > 2$ and odd, and embeds the *dot-product computation* problem. The second scheme, given in Theorem 2, improves upon the sum rate of the first. For the general (non-symmetric) case, we propose two schemes. The first (Proposition 3) targets the regime $q > 2$ and odd, while the second (Theorem 3), extends the design to any $q \geq 2$ and achieves a lower sum rate. The achievability results hold for arbitrary source distributions, following, as we will see, the same reasoning as in [13, Theorem 1].

A. Structured Codes for Distributed Matrix Multiplication: the Symmetric Case

Symmetric matrices (e.g., adjacency, Hessian, and covariance) are fundamental, particularly in machine learning and signal processing. Given distributed sources $\mathbf{A} \in \mathbb{F}_q^{m \times l}$ and $\mathbf{B} \in \mathbb{F}_q^{m \times l}$, with entries from $\mathbb{F}_q$ with m even and $l \geq 1$ for $q > 2$ and odd, we next consider distributed computing of $\boldsymbol{\mathcal{D}} = \mathbf{A}^\intercal \mathbf{B} \in \mathbb{F}_q^{l \times l}$, which is a symmetric[2] matrix, i.e.,

[1] Achievability results for distributed matrix multiplication at practical blocklengths (e.g., [45], [46]) can be obtained using approximate or lossy ($\epsilon > 0$) computation techniques leveraging Kolmogorov complexity [47, Ch. 14].

[2] Symmetry can be enforced through random symmetric transformations, such as constructing $\mathbf{B}$ as a linear transformation of $\mathbf{A}$ such that $\mathbf{B} = \mathbf{A}\mathbf{M}$ where $\mathbf{M} \in \mathbb{F}_q^{l \times l}$ is symmetric, or constructing $\mathbf{A}$ and $\mathbf{B}$ as linear transformations $\mathbf{A} = \mathbf{M}_1\mathbf{Q}$ and $\mathbf{B} = \mathbf{M}_2\mathbf{Q}$ of $\mathbf{Q} \in \mathbb{F}_q^{k \times l}$ where the product $\mathbf{M}_1^\intercal \mathbf{M}_2$ is symmetric, by matrix averaging $\frac{1}{2}(\mathbf{A}^\intercal \mathbf{B} \oplus_q \mathbf{B}^\intercal \mathbf{A})$ using the Toeplitz decomposition over a field where 2 is invertible (i.e., $q > 2$ is odd), or by structurally designing the matrices to enforce symmetry.

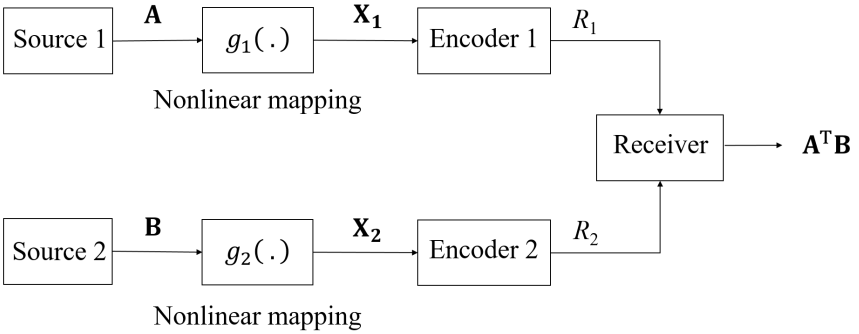

Fig. 1: A general matrix multiplication source network for distributed computation of $\boldsymbol{\mathcal{D}} = \mathbf{A}^\intercal\mathbf{B}$.

$\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{D}}^\intercal$.

Proposition 1. (Distributed computation of symmetric matrix products.) *For the matrix multiplication source network, in the symmetric case, for $q > 2$ and odd, and for any $\epsilon \in (0,1)$,*

$$R^{\Sigma}_{\mathrm{KM,sym}}(\mathbf{A},\mathbf{B}) = 2H_q(\mathbf{U},\,\mathbf{V},\,\mathbf{W}) \tag{4}$$

denotes the achievable sum rate, where $\mathbf{A}$ and $\mathbf{B}$ have the following representations:

$$\mathbf{A} = [\mathbf{A}_1;\mathbf{A}_2] \in \mathbb{F}_q^{m\times l}\ , \quad \textit{and} \quad \mathbf{B} = [\mathbf{B}_1;\mathbf{B}_2] \in \mathbb{F}_q^{m\times l} \tag{5}$$

of even m, with matrix partitions $\mathbf{A}_1,\mathbf{A}_2,\mathbf{B}_1,\mathbf{B}_2 \in \mathbb{F}_q^{m/2\times l}$, and $\mathbf{U} \in \mathbb{F}_q^{m/2\times l}$, $\mathbf{V} \in \mathbb{F}_q^{m/2\times l}$, and $\mathbf{W} \in \mathbb{F}_q^{l\times l}$ are matrix variables, and they satisfy the following relations:

$$\begin{aligned}\mathbf{U} &= \mathbf{A}_2 \oplus_q \mathbf{B}_1 \in \mathbb{F}_q^{m/2\times l}\ ,\\ \mathbf{V} &= \mathbf{A}_1 \oplus_q \mathbf{B}_2 \in \mathbb{F}_q^{m/2\times l}\ ,\\ \mathbf{W} &= \mathbf{A}_2^\intercal\mathbf{A}_1 \oplus_q \mathbf{B}_1^\intercal\mathbf{B}_2 \in \mathbb{F}_q^{l\times l}\ .\end{aligned} \tag{6}$$

Proof. **Outline of the proof.** The proof of Proposition 1 proceeds as follows. We first introduce the non-linear source transformations and explain how their additions in $\mathbb{F}_q$ can be recovered via structured linear encoding. The argument draws on Elias's lemma (Lemma 1), the Körner-Marton theorem for modulo-two sum computation (Theorem 1), and its extension by Han and Kobayashi to standard field addition in $\mathbb{F}_q$ (Lemmas 2 and 3), using an (n,ϵ,δ)-coding scheme (Definition 3). We also employ Lemmas 4 and 5 to extend the setting to q-ary vector variables. We now begin the proof.

Encoding: Sources devise the respective non-linear mappings

$$\begin{aligned}\mathbf{X}_1' &= g_1(\mathbf{A}) = [\mathbf{A}_2;\mathbf{A}_1;\mathbf{A}_2^\intercal\mathbf{A}_1]\ ,\\ \mathbf{X}_2' &= g_2(\mathbf{B}) = [\mathbf{B}_1;\mathbf{B}_2;\mathbf{B}_1^\intercal\mathbf{B}_2]\ .\end{aligned} \tag{7}$$

Vertically concatenating the columns of $\mathbf{X}_1' \in \mathbb{F}_q^{(m+l)\times l}$ and $\mathbf{X}_2' \in \mathbb{F}_q^{(m+l)\times l}$ in (7) we obtain

$$\begin{aligned}\mathbf{X}_1 &= [\mathbf{X}_1'(:,1);\mathbf{X}_1'(:,2);\ldots;\mathbf{X}_1'(:,l)]\ ,\\ \mathbf{X}_2 &= [\mathbf{X}_2'(:,1);\mathbf{X}_2'(:,2);\ldots;\mathbf{X}_2'(:,l)]\ .\end{aligned} \tag{8}$$

Let $\mathbf{Z} = \mathbf{X}_1 \oplus_2 \mathbf{X}_2 \in \mathbb{F}_q^{(m+l)l}$. Exploiting the componentwise and non-linear mappings g_1 and g_2 in (7) to the length-n realizations $\mathbf{A}^n$ and $\mathbf{B}^n$, the sources compute ${\mathbf{X}_1'}^n = g_1(\mathbf{A}^n) \in \mathbb{F}_2^{(m+l)l\times n}$ and ${\mathbf{X}_2'}^n = g_2(\mathbf{B}^n) \in \mathbb{F}_2^{(m+l)l\times n}$. Using (8), $\mathbf{X}_1^n \in \mathbb{F}_q^{(m+l)l\times n}$ and $\mathbf{X}_2^n \in \mathbb{F}_q^{(m+l)l\times n}$ denote the vertical concatenations of ${\mathbf{X}_1'}^n$ and ${\mathbf{X}_2'}^n$, respectively. The separate encoders devise mappings $f_1 : \boldsymbol{\mathcal{X}}_1^n \to \mathcal{R}_{f_1}$ and $f_2 : \boldsymbol{\mathcal{X}}_2^n \to \mathcal{R}_{f_2}$, respectively, which we will detail below. Let $\mathbf{Z}(j) \in \mathbb{F}_q$ be the j-th symbol of $\mathbf{Z}$, and $\mathbf{Z}^n(j) \in \mathbb{F}_q^n$, for $j \in [(m+l)l]$, be its length-n i.i.d. realization.

Decoding: We next detail below how to recover $\mathbf{Z}^n$, which is subsequently used to reconstruct $\boldsymbol{\mathcal{D}}^n$. To do so, we begin by establishing several sufficient conditions below (Lemmas 1-4). Specifically, our encoding scheme requires a well-known fact of Elias [48], which is that the capacity of BSCs can be attained by linear codes, as stated next.

Lemma 1. (Elias's lemma [48].) *Let $\{Z_i\}_{i=1}^{\infty}$ be an i.i.d. binary sequence. For fixed $\epsilon > 0$ and sufficiently large n, there exist a binary matrix[3] $\boldsymbol{\mathcal{C}} \in \mathbb{F}_2^{\kappa\times n}$ and a function $\psi : \mathbb{F}_2^{\kappa} \to \mathbb{F}_2^n$ such that*

$$\kappa < n(H(Z)+\epsilon)\ , \quad \mathbb{P}(\psi(\boldsymbol{\mathcal{C}}Z^n) \neq Z^n) < \epsilon\ . \tag{9}$$

Proof of Lemma 1. This result, originally published in [29], has been detailed in [48, Section 6.2]. Its proof builds upon *coset codes*. We restate the proof here to illustrate how linear codes yield tight error bounds. An (n,κ) coset code is a code with 2^κ codewords of blocklength $n > \kappa$ in which the mapping from message $u^\kappa \in \mathbb{F}_2^{1\times\kappa}$ to codeword z^n is given by $z^n = u^\kappa\boldsymbol{\mathcal{C}} \oplus_2 v^n \in \mathbb{F}_2^{1\times n}$, where $\boldsymbol{\mathcal{C}} \in \mathbb{F}_2^{\kappa\times n}$ is fixed but arbitrary binary encoding matrix and $v^n \in \mathbb{F}_2^{1\times n}$ is a fixed but arbitrary sequence. The codewords of a coset code are thus formed from the codewords of a corresponding *parity-check code*, $\hat{z}^n = u^\kappa\boldsymbol{\mathcal{C}}$, by adding v^n to each codeword. For a BSC, with a transmitted sequence z^n and a noise sequence w^n, the received word is $y^n = z^n \oplus_2 w^n$. After subtracting the fixed sequence v^n from y^n before decoding, we have $\hat{y}^n = y^n \oplus_2 v^n = \hat{z}^n \oplus_2 w^n$.

Since $w^n \perp\!\!\!\perp z^n$ for a BSC, a *maximum likelihood decoder* will correctly decode the same set of noise sequences as for the associated parity-check code [48, Section 6.2]. An (n,κ) parity-check code is specified by a $(n-\kappa)\times n$ binary parity-check matrix $\mathbf{H}$, and it contains all vectors $\hat{z}^n$ whose syndrome

[3]From [19, Appendix IV, Proof of Th. 10, p. 411], a random linear mapping $\boldsymbol{\mathcal{C}} \in \mathbb{F}_2^{\kappa\times n}$, whose components are all chosen independently and uniformly from $\mathbb{F}_2$, yields an (n,ϵ)-coding scheme. The case where $q > 2$ is considered in [14, Lemma 4].

$s^{n-\kappa} \triangleq \mathbf{H}\hat{z}^n$ is equal to zero, namely the set $\{\hat{z}^n \in \{0,1\}^n : \mathbf{H}\hat{z}^n = 0\}$ [48, Section 6.1]. Given some general syndrome $s^{n-\kappa} \in \{0,1\}^{n-\kappa}$, a coset is the set of all vectors z^n satisfying $\mathbf{H}z^n = s^{n-\kappa}$. For any syndrome $s^{n-\kappa}$, *maximum likelihood decoding* can be achieved by calculating $s^{n-k} = \mathbf{H}y^n$, finding the minimum weight sequence $\Psi(s^{n-\kappa})$ that satisfies $s^{n-k} = \mathbf{H}\Psi(s^{n-\kappa})$, and decoding to the codeword $\hat{z}^n = \Psi(s^{n-\kappa}) \oplus_2 y^n$ [48, Theorem 6.1.1].

To complete the proof of Elias's lemma, we need the *coding theorem for parity-check codes* in [48, Theorem 6.2.1]. We present the main steps; the full proof is omitted here for brevity and is available in [48, pp. 206-207]. To this end, employ an ensemble of (n,k) coset codes where each component of $\boldsymbol{\mathcal{C}}$ and v^n is selected from $\mathbb{F}_2$, independently and with equal probability. Then, with *maximum likelihood decoding*, the probability of error for each message for this ensemble of codes used on a BSC, where $Z^n \in \mathbb{F}_2^n$ is viewed as its input sequence, satisfies [48]

$$\mathbb{P}(\psi(\boldsymbol{\mathcal{C}}Z^n)) \neq Z^n) \leq \exp(-nE_r(R)) \ , \tag{10}$$

where $E_r(R) = \ln(2) - 2\ln(\sqrt{p_s} + \sqrt{1-p_s}) - R$ is the *random coding exponent*, as a function of the BSC crossover-probability $p_s = \mathbb{P}(s=1)$, and the transmission rate R. Because $\{Z_i\}_{i=1}^{\infty}$ is stationary, the minimum expected codeword length per symbol is bounded as [47, Theorem 5.4.2]

$$R = (\kappa \ln(2))/n < H(Z) + 1/n \ . \tag{11}$$

Note that $\exp(-nE_r(R)) \overset{(a)}{=} 2^{-n} \cdot (\sqrt{p_s} + \sqrt{1-p_s})^{2n} \cdot 2^{\kappa} \overset{(b)}{<} 2^{\kappa - n}$, where (a) uses the definition of $E_r(R)$ with R from (11), and (b) the triangle inequality. Setting $2^{\kappa-n} = \epsilon$ implies $\mathbb{P}(\psi(\boldsymbol{\mathcal{C}}Z^n) \neq Z^n) < \epsilon$ via (10). For large n, specifically $n > 1/\epsilon$, we have $R < H(Z) + \epsilon$ by (11). This concludes the proof of Lemma 1. □

We continue with the proof of the proposition, and recall that Körner-Marton, in their seminal work [13], applied Lemma 1 to the modulo-two adder source network, given a symmetric source distribution with $(X_1, X_2) \sim \mathrm{DSBS}(p)$. We now restate their main theorem (the *direct part*), which serves as a building block for our achievability schemes and will subsequently be used to prove the proposition.

Theorem 1. (Distributed computation of the modulo-two sum of binary sources [13, Theorem 1].) *For the modulo-two adder source network, for $q = 2$, and for any $\epsilon \in (0,1)$,*

$$R_1 \geq H(Z) \ , \quad R_2 \geq H(Z) \tag{12}$$

denotes the set of achievable rates for computing $Z = X_1 \oplus_2 X_2$.

Proof of Theorem 1. We provide a sketch of the proof here to highlight the utility of algebraic codes. Let $\boldsymbol{\mathcal{C}}(\cdot)$ denote the encoding function used for both X_1^n and X_2^n, such that $f_1(X_1^n) \triangleq \boldsymbol{\mathcal{C}}(X_1^n) = \boldsymbol{\mathcal{C}}X_1^n \in \mathbb{F}_2^{\kappa}$, and $f_2(X_2^n) \triangleq \boldsymbol{\mathcal{C}}(X_2^n) = \boldsymbol{\mathcal{C}}X_2^n \in \mathbb{F}_2^{\kappa}$. Next, define a function $\phi : \mathbb{F}_2^{\kappa} \times \mathbb{F}_2^{\kappa} \to \mathbb{F}_2^n$ as $\phi(a^{\kappa}, b^{\kappa}) \triangleq \psi(a^{\kappa} \oplus_2 b^{\kappa})$, where a^{κ} and b^{κ} are binary sequences of length κ, and $a^{\kappa} \oplus_2 b^{\kappa}$ denotes their modulo-two sum. Since the encoding function $\boldsymbol{\mathcal{C}}$ is linear,

$$\mathbb{P}(\phi(\boldsymbol{\mathcal{C}}X_1^n, \boldsymbol{\mathcal{C}}X_2^n) \neq Z^n) = \mathbb{P}(\psi(\boldsymbol{\mathcal{C}}Z^n) \neq Z^n) < \epsilon \ . \tag{13}$$

This demonstrates that $(\boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{C}})$ is an (n,ϵ)-coding scheme, where rates $R_1 < H_q(Z) + \epsilon$ and $R_2 < H_q(Z) + \epsilon$ are achievable [13, Theorem 1]. This concludes the proof of Theorem 1. □

Continuing with the proof of the proposition, we will make use of several generalizations of Elias's result [48] and Körner-Marton's problem [13], which will subsequently allow us to extend Theorem 1 to additions of vector variables over $\mathbb{F}_q$. In addition, it is worth mentioning that the achievability result in [13, Theorem 1] holds for arbitrary source distributions, whereas the matching converse follows from employing the *strong converse to the source coding theorem* with side information [22] and the symmetric distribution of $(X_1, X_2) \sim \mathrm{DSBS}(p)$ (see Definition 2).

We now proceed with the proof of Proposition 1. Building on [14], we extend the (n,ϵ)-coding scheme in Definition 1 to an (n,ϵ,δ)-coding scheme.

Definition 3 (An (n,ϵ,δ)-coding scheme [14])**.** *The pair of source encoders (f_1, f_2) is called an (n,ϵ,δ)-coding scheme if there exists a function $\phi : \mathcal{R}_{f_1} \times \mathcal{R}_{f_2} \to \mathcal{Z}^n$ that satisfies (13), with the decoding function ψ as given in Lemma 2, such that letting*

$$\hat{Z}^n \triangleq \phi(f_1(X_1^n), f_2(X_2^n)) \ , \tag{14}$$

we have $\mathbb{P}(\hat{Z}^n \neq Z^n) < \delta$, where the encoding rates satisfy $R_1 < H_q(Z) + \epsilon$ and $R_2 < H_q(Z) + \epsilon$.

We next consider a generalization of Elias's result (Lemma 1) in [48] to q-ary variables. Its proof follows from a counting argument (cf. Ahlswede-Han [19, p. 411]), and is omitted here.

Lemma 2. (Han-Kobayashi [14, Lemma 4].) *Let $Z \in \mathbb{F}_q$ be any random variable. Set $Z^n = \{Z_i\}_{i=1}^n = [Z_1; Z_2; \ldots; Z_n] \in \mathbb{F}_q^n$ to denote a length-n i.i.d. realization of Z. Then for any $\epsilon > 0$, $\delta > 0$, and sufficiently large n, a $\kappa \times n$ matrix $\boldsymbol{\mathcal{C}} \in \mathbb{F}_q^{\kappa \times n}$ as a linear encoding function, and a decoding function $\psi : \mathbb{F}_q^{\kappa} \to \mathbb{F}_q^n$ from Lemma 1 exist such that*

$$\kappa < n(H_q(Z) + \epsilon) \ , \quad \mathbb{P}(\psi(\boldsymbol{\mathcal{C}}Z^n) \neq Z^n) < \delta \ . \tag{15}$$

Having established Lemma 2, we now proceed with the proof of Proposition 1 by presenting a generalization of Körner-Marton's problem in [13] to additions in $\mathbb{F}_q$.

Lemma 3. (Han-Kobayashi [14, Lemma 5].) *Let $(X_1,\ X_2)$ be any correlated random variables over $\mathcal{X}_1 \subseteq \mathbb{F}_q$ and $\mathcal{X}_2 \subseteq \mathbb{F}_q$, respectively, encoded separately at different sources. Define $Z = f(X_1, X_2) = X_1 \oplus_q X_2$. Then, for the distributed encoding of the function Z, with linear encoding:*

$$R_1 \geq H_q(Z) \ , \quad R_2 \geq H_q(Z) \tag{16}$$

denotes the set of achievable rates for computing $Z = X_1 \oplus_q X_2$.

Proof of Lemma 3. The proof proceeds along the same lines as that of Theorem 1, where the dimension κ of the encoding matrix $\boldsymbol{\mathcal{C}} \in \mathbb{F}_q^{\kappa \times n}$ is chosen according to Lemma 2. □

With Lemma 3 in place, we continue with the proof of Proposition 1. To this end, we first introduce an intermediate

result (Lemma 4), which guarantees the desired error probability in Lemma 2 by constructing a $\kappa' \times n$ encoding matrix with $\kappa' > \kappa$. This lemma serves as a stepping stone toward Lemma 5, where Lemmas 2 and 3 are extended to q-ary vector variables, necessitating a larger encoding dimension.

Lemma 4. *Consider the setting in Lemma 2. Let $\kappa' = \kappa + \Delta$ for some $\Delta > 0$, which enlarges the value of κ guaranteed by Lemma 2. Then, for any $\epsilon,\ \delta > 0$, and sufficiently large n, a $\kappa' \times n$ matrix $\boldsymbol{\mathcal{C}}' \in \mathbb{F}_q^{\kappa' \times n}$ and a decoding function $\psi' : \mathbb{F}_q^{\kappa'} \to \mathbb{F}_q^n$ exist such that*

$$\kappa' < n(H_q(Z) + \epsilon) \ , \quad \mathbb{P}(\psi'(\boldsymbol{\mathcal{C}}' Z^n) \neq Z^n) < \delta \ . \tag{17}$$

Proof of Lemma 4. Choose a random linear mapping $\boldsymbol{\mathcal{C}}$ independently and uniformly from $\mathbb{F}_q^{\kappa \times n}$, with $\kappa = n(H_q(Z)+\epsilon)$, as in Lemma 2. Then, denoting by $T_\varepsilon(Z)$ the set of all ε-typical sequences for a random variable Z, to evaluate the probability of decoding error, we must consider the following events: i) $E_1 : \ Z^n \notin T_\varepsilon(Z)$, and ii) $E_2 : \ f_1(z^n) = f_1(Z^n)$, for some $z^n \neq Z^n$ such that $z^n \in T_\varepsilon(Z)$, to evaluate the probability of decoding error: $P_e = \mathbb{P}(\psi(\boldsymbol{\mathcal{C}} Z^n) \neq Z^n) = \mathbb{P}(E_1 \ , E_2)$. Because the pair (X_1, X_2) uniquely determines the value of Z, we have $\mathbb{P}(E_1) = 0$. Given $\boldsymbol{\mathcal{C}} \in \mathbb{F}_q^{\kappa \times n}$, by counting all the cases satisfying $f_1(z^n) = f_1(Z^n)$ it follows for any $z^n \neq Z^n$ that $\mathbb{P}(f_1(z^n) = f_1(Z^n)) = (q^{n-1}/q^n)^\kappa = q^{-\kappa}$. Hence, exploiting $\kappa = n(H_q(Z) + \epsilon)$, we have

$$\begin{aligned} \mathbb{P}(E_2) &\leq |T_\varepsilon(Z)| \cdot q^{-\kappa} \\ &\leq \exp(n(H_q(Z) + \varepsilon)) \cdot q^{-\kappa} \\ &= \exp(-n(\epsilon - \varepsilon)) \leq \delta \ , \end{aligned} \tag{18}$$

where δ can be made arbitrarily small by choosing ε small and then n large.

We now set $\kappa' = n(H_q(Z) + \epsilon) + \Delta > \kappa$ for some $\Delta > 0$. Choosing $\boldsymbol{\mathcal{C}}'$ independently and uniformly from $\mathbb{F}_q^{\kappa' \times n}$, by counting all the cases satisfying $f_1(z^n) = f_1(Z^n)$, it follows for any $z^n \neq Z^n$ that $\mathbb{P}(f_1(z^n) = f_1(Z^n)) = (q^{n-1}/q^n)^{\kappa'} = q^{-\kappa'}$. Hence, exploiting $\kappa' > \kappa$, we have

$$\begin{aligned} \mathbb{P}(E_2) &\leq |T_\varepsilon(Z)| \cdot q^{-\kappa'} \\ &\leq \exp(n(H_q(Z) + \varepsilon)) \cdot q^{-\kappa'} \\ &= \exp(n(\kappa'/n - \Delta/n - \epsilon + \varepsilon) - \kappa') \\ &= \exp(-n(\epsilon - \varepsilon) - \Delta) < \exp(-n(\epsilon - \varepsilon)) \leq \delta \ . \end{aligned} \tag{19}$$

From (18) and (19) we infer that $\mathbb{P}(E_2)$ satisfies $P_e(\kappa') < P_e(\kappa) < \delta$ whenever $\kappa' > \kappa$. Hence, (17) follows, which concludes the proof of Lemma 4. □

To proceed with the proof of Proposition 1, we next extend Lemma 3 to *q-ary vector variables*.

Lemma 5. *Let $(\mathbf{X}_1,\ \mathbf{X}_2)$ be any correlated random vectors over $\mathbb{F}_q^m$ each, encoded separately at different sources. Then, for the distributed encoding of $\mathbf{Z} = \mathbf{X}_1 \oplus_q \mathbf{X}_2$, with linear encoding:*

$$R_1 \geq H_q(\mathbf{Z}) \ , \quad R_2 \geq H_q(\mathbf{Z}) \ . \tag{20}$$

Proof of Lemma 5. We prove this lemma using Lemma 4. Let $\mathbf{Z} = \mathbf{X}_1 \oplus_q \mathbf{X}_2 \in \mathbb{F}_q^m$ be any random vector. Then for any fixed $\epsilon > 0$, $\delta > 0$, $j \in [m]$, and sufficiently large n, from Lemmas 2 and 3, a random linear encoding matrix $\boldsymbol{\mathcal{C}}_j \in \mathbb{F}_q^{\kappa_j \times n}$, and a decoding function $\psi_j : \mathbb{F}_q^{\kappa_j} \to \mathbb{F}_q^n$ exist such that

$$\kappa_j < n(H(\mathbf{Z}(j)) + \epsilon) \ , \quad \mathbb{P}(\psi_j(\boldsymbol{\mathcal{C}}_j \mathbf{Z}^n(j)) \neq \mathbf{Z}^n(j)) < \delta \ . \tag{21}$$

Thus, $(\boldsymbol{\mathcal{C}}_j, \boldsymbol{\mathcal{C}}_j)$ is an (n, ϵ, δ)-coding scheme (cf. Definition 3). From Lemma 4, letting $\kappa'_j > \kappa_j$, with κ_j given in (21), a $\kappa'_j \times n$ matrix $\boldsymbol{\mathcal{C}}'_j$ exists such that $(\boldsymbol{\mathcal{C}}'_j, \boldsymbol{\mathcal{C}}'_j)$ is also an (n, ϵ, δ)-coding scheme. A sequential decoding of $\{\mathbf{Z}(j)^n\}$ for ordered $j \in [m]$ is possible, with $\kappa_j < n(H_q(\mathbf{Z}(j) \,|\, \{\mathbf{Z}(j')\}_{j'<j}) + \epsilon)$ for a given j, which allows for reconstructing $\mathbf{Z}^n = \{\mathbf{Z}_i\}_{i=1}^n$, a length-$n$ i.i.d. realization of $\mathbf{Z}$. Thus, setting $\kappa \triangleq \sum_{j\in[m]} \kappa_j < n(\sum_{j\in[m]} H_q(\mathbf{Z}(j) \,|\, \{\mathbf{Z}(j')\}_{j'<j}) + \epsilon) = n(H_q(\mathbf{Z}) + m\epsilon)$, and using a linear encoding matrix $\boldsymbol{\mathcal{C}}$ drawn independently and uniformly from $\mathbb{F}_q^{\kappa \times n}$, leads to an (n, ϵ)-coding scheme with the achievable rate region given in (20). This concludes the proof of Lemma 5. □

We are now ready to prove the statement of Proposition 1. To this end, employ Lemma 5 to the non-linear mappings $\mathbf{X}_1 \in \mathbb{F}_q^{(m+l)l}$ and $\mathbf{X}_2 \in \mathbb{F}_q^{(m+l)l}$ devised in (8) for computing the symmetric product $\boldsymbol{\mathcal{D}}$, for $q > 2$ and odd. Exploiting [13] and [14, Lemma 5], the sum rate

$$R^{\Sigma}_{\mathrm{KM,sym}}(\mathbf{A}, \mathbf{B}) = 2H_q(\mathbf{Z}) = 2H_q(\mathbf{U},\, \mathbf{V},\, \mathbf{W}) \tag{22}$$

is achievable for the receiver to recover $\mathbf{Z}^n = \mathbf{X}_1^n \oplus_q \mathbf{X}_2^n \in \mathbb{F}_q^{(m+l)l \times n}$ with vanishing error. Using the decoded sequence $\hat{\mathbf{Z}}^n = \phi(\boldsymbol{\mathcal{C}}\mathbf{X}_1^n, \boldsymbol{\mathcal{C}}\mathbf{X}_2^n) = \psi(\boldsymbol{\mathcal{C}}\mathbf{X}_1^n \oplus_q \boldsymbol{\mathcal{C}}\mathbf{X}_2^n)$, the receiver computes

$$\begin{aligned} &\frac{1}{2}((\mathbf{U}^\intercal \cdot \mathbf{V} \ominus_q \mathbf{W}) \oplus_q (\mathbf{U}^\intercal \cdot \mathbf{V} \ominus_q \mathbf{W})^\intercal) \\ &\qquad \overset{(a)}{=} \frac{1}{2}(\boldsymbol{\mathcal{D}} \oplus_q \boldsymbol{\mathcal{D}}^\intercal) \overset{(b)}{=} \boldsymbol{\mathcal{D}} \ , \end{aligned} \tag{23}$$

where (a) follows from employing $\mathbf{U}^\intercal \cdot \mathbf{V} \ominus_q \mathbf{W} = \mathbf{A}_2^\intercal \mathbf{B}_2 \oplus_q \mathbf{B}_1^\intercal \mathbf{A}_1$, a reordering of the terms, and $\boldsymbol{\mathcal{D}} = \mathbf{A}_1^\intercal \mathbf{B}_1 \oplus_q \mathbf{A}_2^\intercal \mathbf{B}_2$, (b) from employing the Toeplitz decomposition to uniquely write any symmetric matrix $\boldsymbol{\mathcal{D}} = \mathbf{A}^\intercal \mathbf{B} \in \mathbb{F}_q^{l \times l}$ over a field where 2 is invertible (i.e., q is odd) as $\boldsymbol{\mathcal{D}} = (\boldsymbol{\mathcal{D}} \oplus_q \boldsymbol{\mathcal{D}}^\intercal)/2$. Thus, (22) is achievable for computing the symmetric matrix product $\boldsymbol{\mathcal{D}}$. □

In Proposition 1, based on (23), the receiver computes the matrix product $\mathbf{U}^\intercal \cdot \mathbf{V}$, whose inner dimension is half that of $\mathbf{A}^\intercal \mathbf{B}$, *indicating a computational split between the sources and receiver.* Notably, lossless reconstruction of $\mathbf{Z}^n$ does not imply full recovery of $\mathbf{X}_1^n$ and $\mathbf{X}_2^n$, allowing computation of $\boldsymbol{\mathcal{D}} = \mathbf{A}^\intercal \mathbf{B}$ without revealing the sources in their entirety — a feature with security implications. The extension to odd m is straightforward and omitted for brevity.

We derive a corollary for the *dot-product source network*, where given even-length vectors $\mathbf{A} = (a_i)_{i\in[m]} \in \mathbb{F}_q^m$ and $\mathbf{B} = (b_i)_{i\in[m]} \in \mathbb{F}_q^m$, the receiver computes $d = \langle \mathbf{A}, \mathbf{B} \rangle = \sum_{i=1}^m a_i b_i \in \mathbb{F}_q$ for any $q \geq 2$. This generalizes the result from Proposition 1 originally established for $q > 2$ and odd. This is a special instance where $l = 1$, and thus its proof is omitted.

Corollary 1. (Distributed computation of dot products.) *For the matrix multiplication source network, in the symmetric case where $l = 1$, for any $q \geq 2$ and for any $\epsilon \in (0, 1)$, the sum rate*

$$R^{\Sigma}_{\mathrm{KM,sym}}(\mathbf{A}, \mathbf{B}) = 2H_q(\mathbf{U}, \mathbf{V}, W) \tag{24}$$

is achievable, and the vector variables $\mathbf{U} \in \mathbb{F}_q^{m/2}$ and $\mathbf{V} \in \mathbb{F}_q^{m/2}$, and $W \in \mathbb{F}_q$ satisfy

$$\begin{aligned} \mathbf{U} &= \mathbf{A}_2 \oplus_q \mathbf{B}_1 \ , \\ \mathbf{V} &= \mathbf{A}_1 \oplus_q \mathbf{B}_2 \ , \\ W &= \mathbf{A}_2^{\intercal}\mathbf{A}_1 \oplus_q \mathbf{B}_1^{\intercal}\mathbf{B}_2 \ , \end{aligned} \tag{25}$$

and $\langle \mathbf{A}, \mathbf{B} \rangle = \mathbf{U}^{\intercal}\mathbf{V} \ominus_q W$ for any $q \geq 2$, given the representations of $\mathbf{A}$ and $\mathbf{B}$ as in (5).

Denoting by η_{sym} the gain η in the symmetric case (including dot products and symmetric matrix products), from Corollary 1, the receiver can compute $\langle \mathbf{A}, \mathbf{B} \rangle$ without recovering $(\mathbf{A}, \mathbf{B})$ when $\eta_{\mathrm{sym}} > 1$. We next provide an example under a specific PMF for binary-valued distributed source vectors $(\mathbf{A}, \mathbf{B})$ to show that the achievability result in Corollary 1 does not coincide with (3). In particular, $R^{\Sigma}_{\mathrm{KM,sym}}$ can be substantially smaller than R^{Σ}_{SW}, yielding $\eta_{\mathrm{sym}} > 1$.

Example 1. (Distributed dot product computation over structured source vectors.) *For the matrix multiplication source network, in the symmetric case where $l = 1$, for $q = 2$, consider $\mathbf{A} \in \mathbb{F}_2^m$ and $\mathbf{B} \in \mathbb{F}_2^m$ with the following asymmetric DSBS model:*

$$\begin{aligned} &(a_{\frac{m}{2}+i},\ b_i) \sim \mathrm{DSBS}(p) \ , \\ &(a_i,\ b_{\frac{m}{2}+i}) \sim \mathrm{DSBS}(p) \ \ \textit{are i.i.d. across } i \in [m/2] \ . \end{aligned} \tag{26}$$

Employing the definitions of $\mathbf{X}_1$ and $\mathbf{X}_2$ in (8) and letting $\mathbf{Z} = \mathbf{X}_1 \oplus_2 \mathbf{X}_2 \in \mathbb{F}_q^{m+1}$, from Corollary 1, the achievable sum rate for the receiver to recover $\mathbf{Z} = \big[\mathbf{U}; \mathbf{V}; W\big]$ is given as

$$\begin{aligned} R^{\Sigma}_{\mathrm{KM,sym}}(\mathbf{A}, \mathbf{B}) &= 2H(\mathbf{U}, \mathbf{V}, W) \\ &= 2H(\mathbf{U}) + 2H(\mathbf{V}) \\ &\quad + 2H(\mathbf{A}_2^{\intercal}\mathbf{A}_1 \oplus_2 \mathbf{B}_1^{\intercal}\mathbf{B}_2 \,|\, \mathbf{U}, \mathbf{V}) \\ &\overset{(a)}{=} mh(p) + mh(p) \\ &\quad + 2H(\mathbf{A}_2^{\intercal}\mathbf{A}_1 \oplus_2 \mathbf{B}_1^{\intercal}\mathbf{B}_2 \,|\, \mathbf{U}, \mathbf{V}) \\ &\overset{(b)}{=} 2mh(p) + 2H(\mathbf{U}^{\intercal}\mathbf{A}_1 \oplus_2 \mathbf{A}_2^{\intercal}\mathbf{V} \,|\, \mathbf{U}, \mathbf{V}) \\ &\overset{(c)}{=} 2mh(p) + 2H(\mathbf{Q}^{\intercal}\mathbf{A} \,|\, \mathbf{Q}) \\ &\overset{(d)}{=} 2mh(p) + 2 \sum_{j \in [m]} \binom{m}{j} p^j (1-p)^{m-j} \\ &\qquad \times H\Big(\sum_{i \in \{i_1, i_2, \ldots, i_j\}} a_i \Big) \\ &\overset{(e)}{=} 2mh(p) + 2(1 - (1-p)^m) \ , \end{aligned} \tag{27}$$

where (a) uses $\mathbf{U} = (u_i)_{i \in [m/2]}$ and $\mathbf{V} = (v_i)_{i \in [m/2]}$, defined in (25) with components $u_i, v_i \sim \mathrm{Bern}(p)$, i.i.d. across $i \in [m/2]$; (b) uses $\mathbf{U} = \mathbf{A}_2 \oplus_q \mathbf{B}_1$ and $\mathbf{V} = \mathbf{A}_1 \oplus_q \mathbf{B}_2$ in (25) to rewrite $\mathbf{A}_2^{\intercal}\mathbf{A}_1 \oplus_2 \mathbf{B}_1^{\intercal}\mathbf{B}_2$; (c) follows from $\mathbf{Q} = \big[\mathbf{U}; \mathbf{V}\big] \in \mathbb{F}_2^m$ and observing $\mathbf{A}_2^{\intercal}\mathbf{V} = \mathbf{V}^{\intercal}\mathbf{A}_2 \in \mathbb{F}_2$; and (d) from $H(\mathbf{Q}^{\intercal}\mathbf{A} \,|\, \mathbf{Q}) = H(\mathbf{Q}^{\intercal}\mathbf{A} \,|\, \mathbf{Q}^{\intercal}\mathbf{1}_m)$, given the model in (26), where the nonzero components of $\mathbf{Q}$ are indexed by $\{i_1, i_2, \ldots, i_j\}$ when $\mathbf{Q}^{\intercal}\mathbf{1}_m = j$. Hence,

$$H(\mathbf{Q}^{\intercal}\mathbf{A} \,|\, \mathbf{Q}^{\intercal}\mathbf{1}_m = j) = \begin{cases} 0 \ , & j = 0 \ , \\ H(\mathbf{Q}^{\intercal}\mathbf{A} \,|\, \mathbf{Q}^{\intercal}\mathbf{1}_m = j) \\ = H\Big(\sum\limits_{i \in \{i_1, i_2, \ldots, i_j\}} a_i \Big) \ , & j \geq 1 \ . \end{cases} \tag{28}$$

Finally, step (e) holds because, given $\mathbf{Q} \neq \mathbf{0}_m$, which occurs with probability $1 - (1-p)^m$, the DSBS model implies that the entries of $\mathbf{A}$ are independent and uniformly distributed, i.e., $a_i \overset{i.i.d.}{\sim} \mathrm{Bern}(1/2)$. Hence, incorporating $H\Big(\sum_{i \in \{i_1, i_2, \ldots, i_j\}} a_i\Big) = 1$, $j \geq 1$, we obtain (27).

The encoding rate for the asymptotically lossless compression of $\mathbf{A}$ and $\mathbf{B}$ is given by

$$\begin{aligned} R^{\Sigma}_{\mathrm{SW}}(\mathbf{A}, \mathbf{B}) &\overset{(a)}{=} 2H(\mathbf{A}_1, \mathbf{B}_2) \\ &= 2 \cdot \frac{m}{2}(1 + h(p)) = m(1 + h(p)) \ , \end{aligned} \tag{29}$$

using the Slepian-Wolf theorem [18], where (a) is because $H(\mathbf{A}) = H(\mathbf{A}_1,\ \mathbf{A}_2)$ and $H(\mathbf{B}) = H(\mathbf{B}_1,\ \mathbf{B}_2)$, noting that $\mathbf{A}_1 \perp\!\!\!\perp \mathbf{A}_2$ and $\mathbf{B}_1 \perp\!\!\!\perp \mathbf{B}_2$, for $\mathbf{A}$ and $\mathbf{B}$ with i.i.d.-valued entries.

For this asymmetric DSBS setting, the ratio of R^{Σ}_{SW} in (29) over $R^{\Sigma}_{\mathrm{KM,sym}}$ in (27) is

$$\eta_{\mathrm{sym}} = \frac{R^{\Sigma}_{\mathrm{SW}}(\mathbf{A}, \mathbf{B})}{R^{\Sigma}_{\mathrm{KM,sym}}(\mathbf{A}, \mathbf{B})} = \frac{m(1 + h(p))}{2mh(p) + 2(1 - (1-p)^m)} \ . \tag{30}$$

It is necessary from [14, Lemmas 1-2] (see Conditions 1 and 2 from Lemma 6 in Section IV) that $R_1,\ R_2 \geq mh(p)$, for the joint PMF in (26). Our structured scheme for computing $\mathbf{A}^{\intercal}\mathbf{B}$ incurs $1 - (1-p)^m$ additional bits per source versus this lower bound, approaching 1 as $m \to \infty$. Furthermore, $\lim\limits_{p \to 0} \eta_{\mathrm{sym}} = \infty$, $\lim\limits_{p \to 1} \eta_{\mathrm{sym}} = m/2$, and $\lim\limits_{m \to \infty} \eta_{\mathrm{sym}} = \frac{1 + h(p)}{2h(p)} \geq 1$ matches the gain in [13], which tends to infinity as $p \to 0$ or $p \to 1$.

In Figure 2, we showcase $R^{\Sigma}_{\mathrm{KM,sym}}$ and R^{Σ}_{SW} versus p (in $\log$ scale), using the asymmetric DSBS model in (26) for each pair $\mathbf{A}(:, j)$ and $\mathbf{B}(:, j)$, for $j \in [l]$, with $q = 2$, for distributed computing of symmetric matrices $\boldsymbol{\mathcal{D}} = \mathbf{A}^{\intercal}\mathbf{B}$ for different m with $l = m$, under two further assumptions: (i) $\mathbf{A}_1^{\intercal}\mathbf{B}_1 = \mathbf{B}_1^{\intercal}\mathbf{A}_1$, i.e., $\mathbf{W} = \mathbf{0}_{l \times l}$, and (ii) $\mathbf{A}_2^{\intercal}\mathbf{A}_1 = \mathbf{B}_1^{\intercal}\mathbf{B}_2$. Then, $\mathbf{U}^{\intercal}\mathbf{V} = (\mathbf{A}_2 \oplus_2 \mathbf{B}_1)^{\intercal} \cdot (\mathbf{A}_1 \oplus_2 \mathbf{B}_2) \overset{(a)}{=} \mathbf{A}_2^{\intercal}\mathbf{A}_1 \oplus_2 \mathbf{A}_2^{\intercal}\mathbf{B}_2 \oplus_2 \mathbf{A}_1^{\intercal}\mathbf{B}_1 \oplus_2 \mathbf{B}_1^{\intercal}\mathbf{B}_2 \overset{(b)}{=} \mathbf{A}_1^{\intercal}\mathbf{B}_1 \oplus_2 \mathbf{A}_2^{\intercal}\mathbf{B}_2 = \boldsymbol{\mathcal{D}}$, where (a) and (b) follow from (i) and (ii), respectively. (i)-(ii) ensure η_{sym} to grow exponentially fast, as $p \to 0$ or $p \to 1$. For $q > 2$ and odd, without (i)-(ii), $\mathbf{A}^{\intercal}\mathbf{B}$ can still be recovered (Proposition 1).

Example 1 only captures a restricted class of source vectors, whereas Corollary 1 holds for any possible joint distribution, i.e., any correlation structure between $\mathbf{A}, \mathbf{B} \in \mathbb{F}_q^m$, for $q \geq 2$, where the sum rate required for the partially secure or information-theoretically secure distributed computation of $\langle \mathbf{A}, \mathbf{B} \rangle$ may approach or even exceed R^{Σ}_{SW}. To that end, we

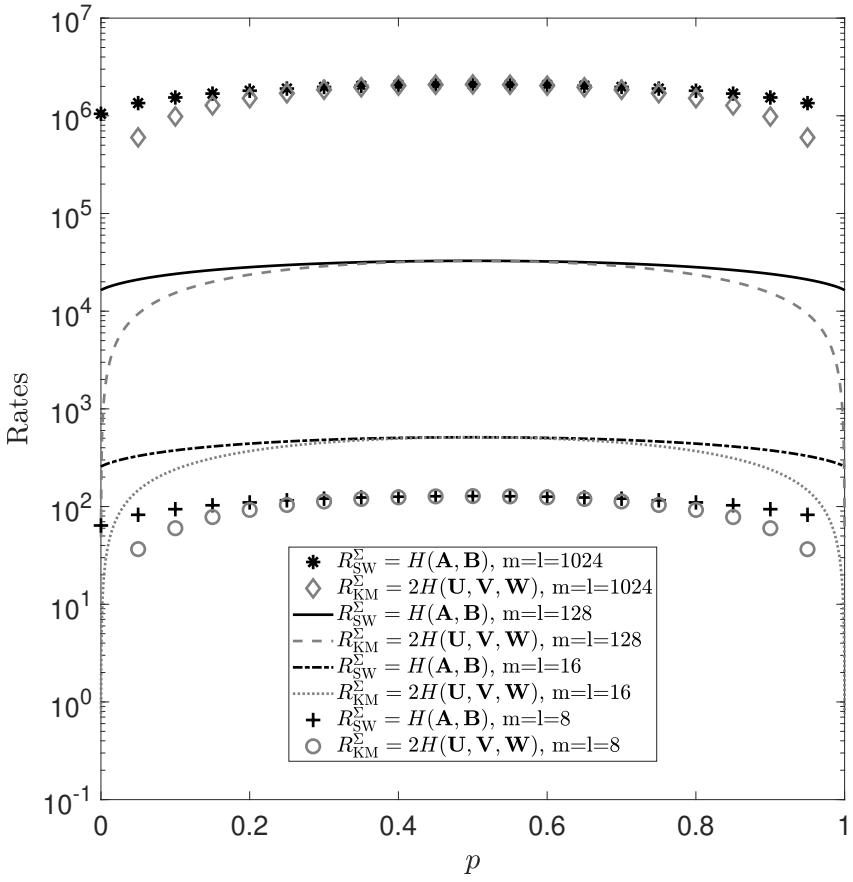

Fig. 2: Rate (in log scale) versus p for the symmetric matrix multiplication network, to compute $\boldsymbol{\mathcal{D}} = \mathbf{A}^{\intercal}\mathbf{B} = \mathbf{B}^{\intercal}\mathbf{A}$, given $\mathbf{A}, \mathbf{B} \in \mathbb{F}_2^{m\times l}$, for different m with $l = m$, where the joint source PMF is given in Example 1 (cf. Corollary 1).

now turn to a potentially more realistic scenario (Example 2), with i.i.d.-valued distributed $\mathbf{A}, \mathbf{B} \in \mathbb{F}_2^m$ with elementwise correlation, and show that the sum rate $R^{\Sigma}_{\mathrm{KM,sym}}$ may still fall below R^{Σ}_{SW} in (3).

Example 2. (Distributed dot product computation over elementwise-correlated vectors.) *For the matrix multiplication source network, in the symmetric case where $l = 1$, for $q = 2$, consider $\mathbf{A} \in \mathbb{F}_2^m$ and $\mathbf{B} \in \mathbb{F}_2^m$, with the following elementwise DSBS model:*

$$(a_i,\ b_i) \sim \mathrm{DSBS}(p) \ \text{ are i.i.d. across } i \in \big[m\big] \ . \tag{31}$$

For a given $p \in \big[0,\ 1/2\big]$, we note that (i) $(a_i,\ b_i) \sim \mathrm{DSBS}(p)$, (ii) $(a_i,\ b_j) \sim \mathrm{DSBS}(1/2)$ for $i \neq j$, and (iii) $(a_i \oplus_2 b_i,\ a_{m/2+i} \oplus_2 b_{m/2+i}) \sim \mathrm{DSBS}(2p(1-p))$ from (i). From Corollary 1,

$$\begin{aligned} R^{\Sigma}_{\mathrm{KM,sym}}(\mathbf{A},\mathbf{B}) &\overset{(a)}{=} 2H(\mathbf{U},\ \mathbf{U}\oplus_2\mathbf{V},\ W) \\ &\overset{(b)}{\leq} 2\cdot\Big(\frac{m}{2} + \frac{m}{2}h(2p(1-p)) + 1\Big) \\ &= m(1+h(2p(1-p))) + 2 \ , \end{aligned} \tag{32}$$

where (a) is because joint entropy is invariant under bijective transformations, and (b) follows from the relations (ii)-(iii) above, which imply $H(\mathbf{U}) = m/2$ and $H(\mathbf{U}\oplus_2\mathbf{V}) = (m/2)\cdot h(2p(1-p))$, respectively, and also from (25), which yields $H(W) \leq 1$ for $W \in \mathbb{F}_2$.

Employing (31), the scheme of Slepian-Wolf in [18] yields the sum rate

$$R^{\Sigma}_{\mathrm{SW}}(\mathbf{A},\mathbf{B}) = m(1+h(p)) \ . \tag{33}$$

For this elementwise DSBS setting, the ratio of R^{Σ}_{SW} in (33) over $R^{\Sigma}_{\mathrm{KM,sym}}$ in (32) is

$$\eta_{\mathrm{sym}} = \frac{R^{\Sigma}_{\mathrm{SW}}(\mathbf{A},\mathbf{B})}{R^{\Sigma}_{\mathrm{KM,sym}}(\mathbf{A},\mathbf{B})} \geq \frac{m(1+h(p))}{m(1+h(2p(1-p))) + 2} \ . \tag{34}$$

It is necessary from [14, Lemmas 1-2] (see Conditions 1 and 2 from Lemma 6 in Section IV) that $R_1,\ R_2 \geq mh(p)$, for the joint PMF in (31), for which the sum rate is upper-bounded by (33). Our structured scheme for computing $\mathbf{A}^{\intercal}\mathbf{B}$ incurs fewer than $(m/2)(1+h(2p(1-p)) - 2h(p)) + 1$ additional bits per source versus this lower bound, approaching $m/2 + 1$ as $p \to 0$ or $p \to 1$, and 1 as $p \to 1/2$. Furthermore, $\lim_{m\to\infty}\eta_{\mathrm{sym}} \geq \frac{1+h(p)}{1+h(2p(1-p))}$, and $\lim_{m\to\infty}\eta_{\mathrm{sym}} \geq 1$ in the limit as $p \to 1/2$. When $p = 1/2$ in (31), resulting in $\mathbf{A}$ and $\mathbf{B}$ being independent, any achievable (R_1, R_2) for $\mathbf{A}^{\intercal}\mathbf{B}$ has to satisfy $R_1 + R_2 \geq 2m = R^{\Sigma}_{\mathrm{SW}}$ from the necessary conditions.

In Example 2, while the elementwise DSBS model of (31) may require the encoders to operate at $R^{\Sigma}_{\mathrm{KM,sym}} > R^{\Sigma}_{\mathrm{SW}}$, the structured encoding scheme ensures that $\mathbf{A}$ and $\mathbf{B}$ are not fully disclosed.

We next derive a necessary condition for the receiver to compute the symmetric matrix product $\boldsymbol{\mathcal{D}} = \mathbf{A}^{\intercal}\mathbf{B} \in \mathbb{F}_q^{l\times l}$ without fully recovering $(\mathbf{A},\ \mathbf{B})$, thus ensuring $\eta_{\mathrm{sym}} > 1$.

Proposition 2. (Necessary condition for achieving $\eta_{\mathrm{sym}} > 1$ in distributed computation of symmetric matrix products.) *For the matrix multiplication source network, in the symmetric case, for $q > 2$ and odd, and for any $\epsilon \in (0,1)$, the condition*

$$H_q(\boldsymbol{\mathcal{D}}) + H_q(\mathbf{U}\ ,\mathbf{V}\,|\,\boldsymbol{\mathcal{D}}) < H_q(\mathbf{A}\,|\,\mathbf{U}\ ,\mathbf{V}\ ,\boldsymbol{\mathcal{D}})\ , \tag{35}$$

with m even and $l \geq 1$, where $\mathbf{U},\mathbf{V} \in \mathbb{F}_q^{m/2\times l}$ are defined in (6), Proposition 1 (and Corollary 1 for $l = 1$, and for any $q \geq 2$) ensures that the sum rate $R^{\Sigma}_{\mathrm{KM,sym}}$ is strictly less than R^{Σ}_{SW} in (3).

Proof. We first consider the dot product $d = \langle\, \mathbf{A},\mathbf{B}\,\rangle$ case from Corollary 1, representing the symmetric setting where $l = 1$, for any $q \geq 2$. We then have from (24)

$$\begin{aligned} R^{\Sigma}_{\mathrm{KM,sym}}(\mathbf{A},\mathbf{B}) &= 2H_q(\mathbf{U},\,\mathbf{V},\,W) = 2H_q(\mathbf{U},\,\mathbf{V},\,d) \\ &= 2H_q(d) + 2H_q(\mathbf{Q}\,|\,d)\ , \end{aligned} \tag{36}$$

$$\begin{aligned} R^{\Sigma}_{\mathrm{SW}}(\mathbf{A},\mathbf{B}) &= H_q(\mathbf{A},\mathbf{B}) \overset{(a)}{=} H_q(\mathbf{A},\mathbf{B},\mathbf{Q},\,d) \\ &\overset{(b)}{=} H_q(\mathbf{Q},\,d) + H_q(\mathbf{A},\mathbf{B}\,|\,\mathbf{U},\,\mathbf{V},d) \\ &\overset{(c)}{=} H_q(d) + H_q(\mathbf{Q}\,|\,d) + H_q(\mathbf{A}\,|\,\mathbf{Q},d)\ , \end{aligned} \tag{37}$$

where (a) holds because both $\mathbf{Q}$ and d are deterministic functions of $(\mathbf{A},\mathbf{B})$; (b) by employing $\mathbf{Q} = \big[\mathbf{U};\mathbf{V}\big]$; and (c) from the condition $H_q(\mathbf{B}\,|\,\mathbf{A},\mathbf{Q}) = 0$. By contrasting (36) with (37), the condition (35) guaranties that $R^{\Sigma}_{\mathrm{KM,sym}} < R^{\Sigma}_{\mathrm{SW}}$.

We then turn to the general symmetric case $\boldsymbol{\mathcal{D}} = \mathbf{A}^{\intercal}\mathbf{B} \in \mathbb{F}_q^{l\times l}$ from Proposition 1 (see (4)), where $l > 1$, and $q > 2$ and odd. Following the same reasoning as in (36) and (37), we obtain:

$$R^{\Sigma}_{\mathrm{KM,sym}}(\mathbf{A},\mathbf{B}) = 2H_q(\boldsymbol{\mathcal{D}}) + 2H_q(\mathbf{Q}\,|\,\boldsymbol{\mathcal{D}})\ , \tag{38}$$

$$R^{\Sigma}_{\mathrm{SW}}(\mathbf{A},\mathbf{B}) = H_q(\boldsymbol{\mathcal{D}}) + H_q(\mathbf{Q}\,|\,\boldsymbol{\mathcal{D}}) + H_q(\mathbf{A}\,|\,\mathbf{Q},\boldsymbol{\mathcal{D}})\ . \tag{39}$$

Thus, by contrasting (38) with (39), the condition (35) guarantees that $R^{\Sigma}_{\mathrm{KM,sym}} < R^{\Sigma}_{\mathrm{SW}}$. □

From Proposition 2, $R^{\Sigma}_{\mathrm{KM,sym}} = 2H_q(\boldsymbol{\mathcal{D}})$ is achievable when $H_q(\mathbf{Q}\,|\,\boldsymbol{\mathcal{D}}) = 0$ in (35). In this case, $H_q(\boldsymbol{\mathcal{D}}) < H_q(\mathbf{A}\,|\,\mathbf{Q},\boldsymbol{\mathcal{D}})$, so $\boldsymbol{\mathcal{D}}$ can be computed at a sum-rate below that required for lossless reconstruction of both $\mathbf{A}$ and $\mathbf{B}$, although partial information about them may still be revealed. For general, potentially non-structured source PMFs, the condition in (35) may fail. In such cases, Proposition 1 may require $R^{\Sigma}_{\mathrm{KM,sym}} > R^{\Sigma}_{\mathrm{SW}}$, while still not necessarily revealing $\mathbf{A}$ and $\mathbf{B}$ completely.

Drawing on Lemmas 1-4 and building on Corollary 1, we next present another achievable region, where given distributed sources $\mathbf{A} \in \mathbb{F}_q^{m\times l}$ and $\mathbf{B} \in \mathbb{F}_q^{m\times l}$, the receiver aims to compute the symmetric matrix $\boldsymbol{\mathcal{D}} = \mathbf{A}^{\intercal}\mathbf{B} \in \mathbb{F}_q^{l\times l}$ for $q > 2$ and odd, achieving an improved $R^{\Sigma}_{\mathrm{KM,sym}}$.

Theorem 2. (Distributed computation of symmetric matrix products.) *For the matrix multiplication source network, in the symmetric case, for $q > 2$ and odd, and for any $\epsilon \in (0,1)$, the sum rate*

$$R^{\Sigma}_{\mathrm{KM,sym,rfd}}(\mathbf{A},\ \mathbf{B}) = 2H_q(\mathbf{Z}) \tag{40}$$

is achievable, where $\mathbf{Z} = \mathbf{X}_1 \oplus_q \mathbf{X}_2 \in \mathbb{F}_q^{(m+l)l}$, with vectors $\mathbf{X}_1,\ \mathbf{X}_2 \in \mathbb{F}_q^{(m+l)l}$ that satisfy

$$\begin{aligned} \mathbf{X}_1 &= \big[\mathbf{X}'_1(:,1);\mathbf{X}'_1(:,2);\ldots;\mathbf{X}'_1(:,l)\big]\ , \\ \mathbf{X}_2 &= \big[\mathbf{X}'_2(:,1);\mathbf{X}'_2(:,2);\ldots;\mathbf{X}'_2(:,l)\big]\ , \end{aligned} \tag{41}$$

with matrices $\mathbf{X}'_1 \in \mathbb{F}_q^{(m+l)\times l}$ and $\mathbf{X}'_2 \in \mathbb{F}_q^{(m+l)\times l}$ derived from the following encoder mappings:

$$\begin{aligned} \mathbf{X}'_1 &= g_1(\mathbf{A}) = \big[\mathbf{A}_2;\mathbf{A}_1;\mathbf{A}_2^{\intercal}\mathbf{A}_1 \oplus_q \mathbf{A}_1^{\intercal}\mathbf{A}_2\big]\ , \\ \mathbf{X}'_2 &= g_2(\mathbf{B}) = \big[\mathbf{B}_1;\mathbf{B}_2;\mathbf{B}_1^{\intercal}\mathbf{B}_2 \oplus_q \mathbf{B}_2^{\intercal}\mathbf{B}_1\big]\ . \end{aligned} \tag{42}$$

Then for any $\epsilon > 0$, $\delta > 0$, and sufficiently large n, a $\kappa \times n$ matrix $\boldsymbol{\mathcal{C}} \in \mathbb{F}_q^{\kappa\times n}$ and decoding functions $\psi_j : \mathbb{F}_q^{\kappa_j} \to \mathbb{F}_q^n$, $j \in [(m+l)l]$ exist such that

$$\kappa = \max\Big\{ \sum_{j\in[ml]} \kappa_j\ ,\ \sum_{j\in[ml+1,\ (m+l)l]} \kappa_j \Big\}\ , \tag{43}$$

$$\kappa_j < n(H_q(\mathbf{Z}(j)) + \epsilon)\ ,\quad j \in [(m+l)l]\ , \tag{44}$$

$$\mathbb{P}(\{\psi_j(\boldsymbol{\mathcal{C}}\mathbf{Z}^n(j)) \neq \mathbf{Z}^n(j)\}_{j\in[(m+l)l]}) < \delta\ , \tag{45}$$

where $\mathbf{Z}(j)$, $j \in [(m+l)l]$, is the j-th component of $\mathbf{Z}$, and $\mathbf{Z}^n(j) = (\mathbf{Z}_1(j),\ldots,\mathbf{Z}_n(j)) \in \mathbb{F}_q^n$.

Proof. **Outline of the proof.** The proof of Theorem 2 proceeds as follows. We first introduce the non-linear source transformations and explain how their additions in $\mathbb{F}_q$ can be recovered via structured linear encoding, drawing on Lemmas 1-5. We now begin the proof.

Encoding: Following Proposition 1, let $\mathbf{Z} = \mathbf{X}_1 \oplus_q \mathbf{X}_2 \in \mathbb{F}_q^{(m+l)l}$. For fixed $\epsilon,\ \delta > 0$ and sufficiently large n, the encoders operate componentwise for each $j \in [(m+l)l]$, using the same matrix $\boldsymbol{\mathcal{C}}_j$, drawn independently and uniformly from $\mathbb{F}_q^{\kappa_j\times n}$ (cf. Ahlswede-Han [19]). They compute $\boldsymbol{\mathcal{C}}_j\mathbf{X}_1^n(j) \in \mathbb{F}_q^{\kappa_j}$ and $\boldsymbol{\mathcal{C}}_j\mathbf{X}_2^n(j) \in \mathbb{F}_q^{\kappa_j}$, where, by Lemmas 2 and 3, $\kappa_j < n(H_q(\mathbf{Z}(j)\,|\,\{\mathbf{Z}(j')\}_{j'<j}) + \epsilon)$. From Lemma 5, we then set $\kappa = \sum_{j\in[m]} \kappa_j < n(H_q(\mathbf{Z}) + m\epsilon)$. Using a linear encoding matrix $\boldsymbol{\mathcal{C}}$ drawn independently and uniformly from $\mathbb{F}_q^{\kappa\times n}$ achieves $R_1 \geq H_q(\mathbf{Z})$ and $R_2 \geq H_q(\mathbf{Z})$.

Decoding: From [19], there exists a decoding function $\psi_j : \mathbb{F}_q^{\kappa_j} \to \mathbb{F}_q^n$ that satisfies:

$$\begin{aligned} \hat{\mathbf{Z}}^n(j) &\triangleq \phi(\boldsymbol{\mathcal{C}}_j\mathbf{X}_1^n(j), \boldsymbol{\mathcal{C}}_j\mathbf{X}_2^n(j)) \\ &\triangleq \psi_j(\boldsymbol{\mathcal{C}}_j(\mathbf{X}_1^n(j) \oplus_q \mathbf{X}_2^n(j))) \end{aligned} \tag{46}$$

such that i) $\kappa_j < n(H(\mathbf{Z}(j)\,|\,\{\mathbf{Z}(j')\}_{j'<j}) + \epsilon)$, and ii) $\mathbb{P}(\psi_j(\boldsymbol{\mathcal{C}}_j\mathbf{Z}^n(j)) \neq \mathbf{Z}^n(j)) < \delta$.

Using the achievability result in [13], and Lemmas 1-2, the achievable sum rate for the receiver to recover the matrix sequence $\mathbf{Z}^n = \mathbf{X}_1^n \oplus_q \mathbf{X}_2^n \in \mathbb{F}_q^{(m+l)l\times n}$ with vanishing error is:

$$R^{\Sigma}_{\mathrm{KM,sym,rfd}}(\mathbf{A},\ \mathbf{B}) = 2H_q(\mathbf{U},\,\mathbf{V},\,\mathbf{W}_S)\ , \tag{47}$$

where the matrix variables $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}_S$ are given as follows:

$$\mathbf{U} = \mathbf{A}_2 \oplus_q \mathbf{B}_1 \in \mathbb{F}_q^{m/2\times l} \ , \quad \mathbf{V} = \mathbf{A}_1 \oplus_q \mathbf{B}_2 \in \mathbb{F}_q^{m/2\times l} \ ,$$
$$\mathbf{W}_S = \mathbf{A}_2^\intercal \mathbf{A}_1 \oplus_q \mathbf{A}_1^\intercal \mathbf{A}_2 \oplus_q \mathbf{B}_1^\intercal \mathbf{B}_2 \oplus_q \mathbf{B}_2^\intercal \mathbf{B}_1 \in \mathbb{F}_q^{l\times l} \ . \tag{48}$$

Using $\hat{\mathbf{Z}}^n$ and exploiting (48), the receiver computes

$$\begin{aligned} &\frac{1}{2}(\mathbf{U}^\intercal \cdot \mathbf{V} \oplus_q \mathbf{V}^\intercal \cdot \mathbf{U} \ominus_q \mathbf{W}_S) \\ &\overset{(a)}{=} \frac{1}{2}((\mathbf{A}_1^\intercal \mathbf{B}_1 \oplus_q \mathbf{A}_2^\intercal \mathbf{B}_2) \oplus_q (\mathbf{A}_1^\intercal \mathbf{B}_1 \oplus_q \mathbf{A}_2^\intercal \mathbf{B}_2)^\intercal) \\ &\overset{(b)}{=} \boldsymbol{\mathcal{D}} \ , \end{aligned} \tag{49}$$

where (a) follows from reordering the terms and using (48), and (b) from the symmetry of $\boldsymbol{\mathcal{D}} = \mathbf{A}^\intercal \mathbf{B} \in \mathbb{F}_q^{l\times l}$, rendering the sum rate in (47) achievable for $q > 2$ and odd.

Concatenating the columns of the matrices $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}_S$ each, defined in (48), denote by

$$\begin{aligned} &\big[\mathbf{U}(:,1); \mathbf{U}(:,2); \ldots; \mathbf{U}(:,l)\big] \\ &= \big[\mathbf{Z}(1); \mathbf{Z}(2); \ldots; \mathbf{Z}(ml/2)\big] \in \mathbb{F}_q^{ml/2} \ , \end{aligned} \tag{50}$$

$$\begin{aligned} &\big[\mathbf{V}(:,1); \mathbf{V}(:,2); \ldots; \mathbf{V}(:,l)\big] \\ &= \big[\mathbf{Z}(ml/2+1); \mathbf{Z}(ml/2+2); \ldots; \mathbf{Z}(ml))\big] \in \mathbb{F}_q^{ml/2} \ , \end{aligned} \tag{51}$$

$$\begin{aligned} &\big[\mathbf{W}_S(:,1); \mathbf{W}_S(:,2); \ldots; \mathbf{W}_S(:,l)\big] \\ &= \big[\mathbf{Z}(ml+1); \mathbf{Z}(ml+2); \ldots; \mathbf{Z}((m+l)l)\big] \in \mathbb{F}_q^{l^2} \ , \end{aligned} \tag{52}$$

where $\mathbf{Z}(j) = \mathbf{A}_2(:,j) \oplus_q \mathbf{B}_1(:,j)$, for $j \in [ml/2]$, $\mathbf{Z}(j) = \mathbf{A}_1(:,j) \oplus_q \mathbf{B}_2(:,j)$, for $j \in [ml/2+1, ml]$, and $\mathbf{Z}(ml+j) = \mathbf{W}_S(j - l\cdot\big(\lceil \frac{j}{l} \rceil - 1\big), \lceil \frac{j}{l} \rceil) \in \mathbb{F}_q$, for $j \in [l^2]$. We note the following. (50) and (51) can be combined into $\mathbf{Z}(j) = g_{1,j}(\mathbf{A}) \oplus_q g_{2,j}(\mathbf{B})$, $j \in [ml]$, for some $g_{1,j}$ and $g_{2,j}$, where $\{\mathbf{Z}(j)\}_{j\in[ml]}$ are mutually independent. Similarly, $\mathbf{Z}(j) = g'_{1,j}(\mathbf{A}) \oplus_q g'_{2,j}(\mathbf{B})$, $j \in [ml+1, \ (m+l)l]$, where $g'_{1,j} \neq g_{1,j}$ and $g'_{2,j} \neq g_{2,j}$, by directly comparing (50), (51), and (52).

From (48), the computation of the linear parts in (50) and (51) requires

$$H_q(\mathbf{Z}(j), \, j \in [ml]) = H_q(\mathbf{U} \ , \mathbf{V}) \leq lm \ , \tag{53}$$

and we also note that the computation of the non-linear part in (52) requires

$$\begin{aligned} &H_q(\mathbf{Z}(j), j \in [ml+1, \ (m+l)l] \,|\, \mathbf{Z}(j), \, j \in [ml]) \\ &\quad \overset{(a)}{=} H_q(\mathbf{W}_S \,|\, \mathbf{U} \ , \mathbf{V}) \\ &\quad \overset{(b)}{=} H_q(\mathbf{U}^\intercal \cdot \mathbf{V} \oplus_q \mathbf{V}^\intercal \cdot \mathbf{U} \ominus_q \mathbf{W}_S \,|\, \mathbf{U} \ , \mathbf{V}) \\ &\quad \overset{(c)}{=} H_q(\mathbf{A}^\intercal \mathbf{B} \,|\, \mathbf{U} \ , \mathbf{V}) \leq l^2 \ , \end{aligned} \tag{54}$$

where (a) is due to (48), and from using definitions in (50), (51), and (52), (b) follows by conditioning, and (c) from employing (49). Hence, for $\mathbf{X}_1 \in \mathbb{F}_q^{(m+l)l}$ and $\mathbf{X}_2 \in \mathbb{F}_q^{(m+l)l}$ given in (41), employing Lemmas 4 and 5, and from (53) and (54), the following rate per source can be achieved for computing the symmetric matrix $\boldsymbol{\mathcal{D}} = \mathbf{A}^\intercal \mathbf{B} \in \mathbb{F}_q^{l\times l}$, for $q > 2$ and odd:

$$\begin{aligned} \frac{\kappa}{n} < \max\{&H_q(\mathbf{Z}(j), \, j \in [ml]) \ , \\ &H_q(\mathbf{Z}(j), \, j \in [ml+1, \ (m+l)l] \,|\, \mathbf{Z}(j), \, j \in [ml])\} + \epsilon \\ = \max\{&H_q(\mathbf{U} \ , \mathbf{V} \,|\, \boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{D}}^\intercal) \ , \\ &H_q(\mathbf{A}^\intercal \mathbf{B} \,|\, \mathbf{U} \ , \mathbf{V}, \ \boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{D}}^\intercal)\} + \epsilon \ , \end{aligned} \tag{55}$$

for a $\kappa \times n$ linear encoding matrix $\boldsymbol{\mathcal{C}} \in \mathbb{F}_q^{\kappa\times n}$ with $\kappa = \max\big\{\sum_{j\in[ml]} \kappa_j \ , \ \sum_{j\in[ml+1, \ (m+l)l]} \kappa_j\big\}$.

From Lemma 4, and [13], [14], and [19], if we choose $\boldsymbol{\mathcal{C}}$ independently and uniformly from $\mathbb{F}_q^{\kappa\times n}$, and κ as in (43), $(\boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{C}})$ forms an (n, ϵ, δ)-coding scheme for decoding $\mathbf{Z}^n$. □

B. Structured Codes for Distributed Matrix Multiplication: the General non-Symmetric Case

Focusing on general $\mathbf{A}, \ \mathbf{B} \in \mathbb{F}_q^{m\times l}$, for $q > 2$ and odd, and $m, \ l > 1$, we next devise distributed encoding schemes for computing $\boldsymbol{\mathcal{D}} = \mathbf{A}^\intercal \mathbf{B} \in \mathbb{F}_q^{l\times l}$. Unlike Proposition 1 and Theorem 2, this setting necessitates new techniques due to the non-symmetric form of $\boldsymbol{\mathcal{D}}$, as we detail below.

Proposition 3. (Distributed computation of square matrix products.) *For the general matrix multiplication source network, for $q > 2$ and odd, and for any $\epsilon \in (0,1)$, the sum rate*

$$\begin{aligned} R_{\mathrm{KM,alt}}^{\Sigma}(\mathbf{A}, \mathbf{B}) = 2H_q(&\{\mathbf{A} \oplus_q \tilde{\mathbf{B}}_j\}_{j=1}^l \, , \\ &\{\mathbf{A}^\intercal \mathbf{A} \oplus_q \tilde{\mathbf{B}}_j^\intercal \tilde{\mathbf{B}}_j\}_{j=1}^l) \end{aligned} \tag{56}$$

is achievable, where $\tilde{\mathbf{B}}_j = \mathbf{B}_j \mathbf{1}_{1\times l} \in \mathbb{F}_q^{m\times l}$, with $\mathbf{B}_j \in \mathbb{F}_q^m$ denoting the j-th column of $\mathbf{B}$.

Proof. We set vector $\boldsymbol{\mathcal{D}}_j$ to be $\boldsymbol{\mathcal{D}}_j = (d_{ij})_{i\in[l]} = \mathbf{A}^\intercal \mathbf{B}_j \in \mathbb{F}_q^l$ for $j \in [l]$. Following the steps of Lemma 1 and the Proofs of Corollary 1 and Proposition 1, the receiver can recover $\{\mathbf{A} \oplus_q \tilde{\mathbf{B}}_j\}_{j=1}^l$, $\{\mathbf{A}^\intercal \mathbf{A} \oplus_q \tilde{\mathbf{B}}_j^\intercal \tilde{\mathbf{B}}_j\}_{j=1}^l$, and then compute the following $l \times l$ matrix:

$$\begin{aligned} &(\mathbf{A} \oplus_q \tilde{\mathbf{B}}_j)^\intercal (\mathbf{A} \oplus_q \tilde{\mathbf{B}}_j) \ominus_q (\mathbf{A}^\intercal \mathbf{A} \oplus_q \tilde{\mathbf{B}}_j^\intercal \tilde{\mathbf{B}}_j) \\ &= \mathbf{1}_{l\times 1} \mathbf{B}_j^\intercal \mathbf{A} \oplus_q \big[\mathbf{A}_1^\intercal; \mathbf{A}_2^\intercal; \ldots, \mathbf{A}_l^\intercal\big] \mathbf{B}_j \mathbf{1}_{1\times l} \\ &= \mathbf{1}_{l\times 1} \big((d_{ij})_{i\in[l]}\big)^\intercal \oplus_q (d_{ij})_{i\in[l]} \mathbf{1}_{1\times l} \\ &= (d_{ij} \oplus_q d_{i'j})_{i, \ i'\in[l]} \in \mathbb{F}_q^{l\times l} \end{aligned} \tag{57}$$

which is a symmetric matrix with l unknowns and $l(l-1)/2 \geq l$ linearly independent equations for $l \geq 2$ and $q > 2$. Hence, $\boldsymbol{\mathcal{D}}_j$, for each $j \in [l]$, as well as $\boldsymbol{\mathcal{D}}$ can be recovered. □

To demonstrate the performance of Proposition 3, we next consider an example.

Example 3. (Computing a non-symmetric matrix product of structured sources.) *For the general matrix multiplication source network, where $l = 2$, for $q = 3$, consider $\mathbf{A}, \ \mathbf{B} \in \mathbb{F}_3^{m\times 2}$, with entries that satisfy $a_{ij} \sim \big(1/2-\zeta, \ 2\zeta, \ 1/2-\zeta\big)$, for $i \in [m]$ and $j \in \{1,2\}$, i.i.d. across i for some $\zeta \in [0, 1/2]$, and the joint PMF of $(a_{i1}, \ b_{i1})$, i.i.d. across $i \in [m]$, is given as*

$$P_{a_{i1}, b_{i1}} = \begin{bmatrix} (\frac{1}{2}-\zeta)(1-p) & (\frac{1}{2}-\zeta)p & 0 \\ 2\zeta p & 0 & 2\zeta(1-p) \\ 0 & (\frac{1}{2}-\zeta)(1-p) & (\frac{1}{2}-\zeta)p \end{bmatrix} . \tag{58}$$

We further assume that $b_{i1} = b_{i2} = -a_{i2}$. Thus, $H_3(\mathbf{A}, \mathbf{B}) = mH_3(a_{i1}, a_{i2}, b_{i1}, b_{i2}) = mH_3(a_{i1}, b_{i1})$.

Therefore, the sum rate for distributed encoding of $(\mathbf{A}, \mathbf{B})$ *is given as [18]*

$$\begin{aligned} R_{\mathrm{SW}}^{\Sigma}(\mathbf{A}, \mathbf{B}) &= m(h(a_{i1}) + h(b_{i1} \,|\, a_{i1})) \\ &= m(h(2\zeta) + (1 - 2\zeta) + h(p)) \ . \end{aligned} \tag{59}$$

Exploiting Proposition 3 to compute $\boldsymbol{\mathcal{D}} = \mathbf{A}^{\intercal}\mathbf{B} \in \mathbb{F}_3^{2\times 2}$*, we can achieve a sum rate of*

$$\begin{aligned} R_{\mathrm{KM,alt}}^{\Sigma}(\mathbf{A}, \mathbf{B}) \leq 2mh\Big(2\big(\frac{1}{2} - \zeta\big)(1-p) + 2\zeta(1-p), \\ 2\big(\frac{1}{2} - \zeta\big)p + 2\zeta p\Big) + 2\log_2(3) \ . \end{aligned} \tag{60}$$

For details on the evaluation of $R_{\mathrm{KM,alt}}^{\Sigma}$*, we refer the reader to [49, Appendix A-J]. These details consist of standard algebraic manipulations and are omitted here.*

In Figure 3, using (58), with $q = 3$ *and* $\zeta = 0.2$*, we display the sum rate* $R_{\mathrm{KM,alt}}^{\Sigma}$ *in (56) and* R_{SW}^{Σ} *versus* p *(in* $\log$ *scale). The gain* η *grows exponentially as* p *approaches* 0 *or* 1*.*

Inspired by the Ahlswede-Han scheme [19], which blends unstructured coding [18] with structured coding [13], we design a new encoding scheme for the distributed computation of $\boldsymbol{\mathcal{D}} = \mathbf{A}^{\intercal}\mathbf{B} \in \mathbb{F}_q^{l\times l}$, for $q \geq 2$ and $m,\ l > 1$, generalizing Proposition 3, originally stated for odd $q > 2$.

Theorem 3. (Distributed computation of square matrix products.) *For the general matrix multiplication source network, for any* $q \geq 2$*, and for any* $\epsilon \in (0, 1)$*, the following sum rate is achievable:*

$$\begin{aligned} R_{\mathrm{AH}}^{\Sigma}(\mathbf{A},\ \mathbf{B}) &= H_q(\mathbf{A}_1, \mathbf{B}_2) \\ &\quad + 2\max\{H_q(\mathbf{A}_2 \oplus_q \mathbf{B}_1 \,|\, \mathbf{A}_1, \mathbf{B}_2), \\ &\quad\ H_q(\mathbf{A}_1^{\intercal}\mathbf{A}_2 \oplus_q \mathbf{B}_1^{\intercal}\mathbf{B}_2 \,|\, \mathbf{A}_1, \mathbf{B}_2, \mathbf{A}_2 \oplus_q \mathbf{B}_1)\} \ . \end{aligned} \tag{61}$$

Proof. We next exploit the achievable rate region of Ahlswede-Han in [19]. Let $\mathbf{S}_1$, $\mathbf{S}_2$ be finite-valued variables such that $\mathbf{S}_1 - \mathbf{A} - \mathbf{B} - \mathbf{S}_2$ forms a Markov chain. For $q = 2$, from [19],

$$\begin{aligned} R_1 &\geq I(\mathbf{S}_1; \mathbf{A} \mid \mathbf{S}_2) + H(\mathbf{A} \oplus_2 \mathbf{B} \mid \mathbf{S}_1,\ \mathbf{S}_2) \ , \\ R_2 &\geq I(\mathbf{S}_2; \mathbf{B} \mid \mathbf{S}_2) + H(\mathbf{A} \oplus_2 \mathbf{B} \mid \mathbf{S}_1,\ \mathbf{S}_2) \ , \\ R_1 + R_2 &\geq R_{\mathrm{AH}}^{\Sigma}(\mathbf{A},\ \mathbf{B}) = I(\mathbf{S}_1,\ \mathbf{S}_2; \mathbf{A},\ \mathbf{B}) \\ &\qquad + 2H(\mathbf{A} \oplus_2 \mathbf{B} \,|\, \mathbf{S}_1,\ \mathbf{S}_2) \ , \end{aligned} \tag{62}$$

which reduces to the rate region of [18] for $\mathbf{S}_1 = \mathbf{A}$, $\mathbf{S}_2 = \mathbf{B}$, and to [13] for $\mathbf{S}_1 = \mathbf{S}_2 = \mathbf{0}_{m\times l}$.

In the general case $q > 2$, the square matrix product $\mathbf{A}^{\intercal}\mathbf{B} \in \mathbb{F}_q^{l\times l}$ can be expressed as

$$\begin{aligned} \mathbf{A}^{\intercal}\mathbf{B} = \mathbf{A}_1^{\intercal}(\mathbf{A}_2 \oplus_q \mathbf{B}_1) \oplus_q (\mathbf{A}_2 \oplus_q \mathbf{B}_1)^{\intercal}\mathbf{B}_2 \\ \ominus_q (\mathbf{A}_1^{\intercal}\mathbf{A}_2 \oplus_q \mathbf{B}_1^{\intercal}\mathbf{B}_2) \ , \end{aligned} \tag{63}$$

where this decomposition — given in (63) — does not rely on the symmetry of the matrix product. Exploiting (63) together with the sum-rate expression R_{AH}^{Σ} in (62), we consider the setting $\mathbf{S}_1 = \mathbf{A}_1$, $\mathbf{S}_2 = \mathbf{B}_2$. In this case, the receiver can recover the quantities $\mathbf{A}_1, \mathbf{B}_2$, $\mathbf{A}_2 \oplus_q \mathbf{B}_1$, and $\mathbf{A}_1^{\intercal}\mathbf{A}_2 \oplus_q \mathbf{B}_1^{\intercal}\mathbf{B}_2$ as specified in (63), thus enabling the recovery of $\mathbf{A}^{\intercal}\mathbf{B}$ at the rate given by (61). The use of $\max\{\cdot, \cdot\}$ in (61) is justified by the same arguments as in (55). □

C. Distributed Computation of Matrix Products using Hybrid Coding

We next introduce a hybrid encoding scheme based on *Körner's characteristic graphs* [24], tailored for scenarios with *linearly separable side information*. The framework in [24] has been used to characterize the communication rate by identifying which source values can be grouped under the same codeword without ambiguity at the receiver, enabling asymptotically lossless function computation, as shown in prior work [24]–[27], [50]–[55]. To motivate our approach, we begin by defining *characteristic graphs* in a point-to-point setting with receiver side information.

Definition 4 (Characteristic graph [25])**.** *Consider a point-to-point model where the source observes* $X_1 \in \mathbb{F}_q$ *and the receiver has side information* $X_2 \in \mathbb{F}_q$*, aiming to compute* $f(X_1, X_2)$*. The characteristic graph* $G_{X_1} = G(\mathcal{V}, \mathcal{E})$*, constructed by the source using* X_1 *with respect to* X_2*,* P_{X_1,X_2}*, and* f*, is defined as follows:*

1) $\mathcal{V}$ *denotes the vertex set such that* $\mathcal{V} = \{x_1^{(k)} \in \mathbb{F}_q\}$*, and*
2) $\mathcal{E}$ *denotes the edge set such that* $\mathcal{E} = \{(x_1^{(1)}, x_1^{(2)}) : x_1^{(1)}, x_1^{(2)} \in \mathcal{V},\ x_1^{(1)} \neq x_1^{(2)},\ \exists x_2 \in \mathbb{F}_q : P_{X_1,X_2}(x_1^{(1)}, x_2^{(1)}) \cdot P_{X_1,X_2}(x_1^{(2)}, x_2^{(2)}) > 0$ *and* $f(x_1^{(1)}, x_2^{(1)}) \neq f(x_1^{(2)}, x_2^{(2)})\}$ *[51].*

Definition 5 (Independent set [56])**.** *An independent set (IS) of a graph* $G(\mathcal{V}, \mathcal{E})$ *is a subset of* $\mathcal{V}$ *with no adjacent pairs. A maximal independent set (MIS) is an IS that cannot be extended by including any additional vertex from* $\mathcal{V}$ *[56], and* $\Gamma(G)$ *denotes the set of all MISs of* G*.*

As established by Orlitsky and Roche [25], the minimum compression rate for computing $f(X_1, X_2)$ given side information X_2 with vanishing error is given by

$$\begin{aligned} H_{G_{X_1}}(X_1 \,|\, X_2) \triangleq \min\{I(X_1; \mathcal{W} \mid X_2) \mid \mathcal{W} - X_1 - X_2 \ , \\ X_1 \in \mathcal{W} \in \Gamma(G_{X_1})\} \ , \end{aligned} \tag{64}$$

which is the *conditional characteristic graph entropy*, where $\mathcal{W} - X_1 - X_2$ indicates a Markov chain, $I(X_1; \mathcal{W} \mid X_2) = H_q(\mathcal{W} \mid X_2) - H_q(\mathcal{W} \mid X_1)$, and $X_1 \in \mathcal{W} \in \Gamma(G_{X_1})$ means that the minimization is over all $P_{\mathcal{W},X_1}(\omega, x_1) > 0$ such that $\mathcal{W}$ is an IS of G_{X_1}.

Proposition 4. (Hybrid encoding for distributed computation of matrix products.) *For the matrix multiplication source network, for any* $q \geq 2$*, and for any* $\epsilon \in (0, 1)$*, the following sum rate is achievable:*

$$R_{\mathrm{KM-OR}}^{\Sigma}(\mathbf{Y}) = 2H_q(\mathbf{Y}) + H_{G_{\mathbf{A}}}(\mathbf{A} \,|\, \mathbf{Y}) \ , \tag{65}$$

where $\mathbf{Y} = \theta_1(\mathbf{A}) \oplus_q \theta_2(\mathbf{B})$ *for some functions* θ_1 *and* θ_2*, reflecting a linearly separable structure.*

Proof. Let $\mathbf{Y}$ be the side information at the receiver, which is expressed as $\mathbf{Y} = \theta_1(\mathbf{A}) \oplus_q \theta_2(\mathbf{B})$. If $\mathbf{Y} = \mathbf{A} \oplus_q \mathbf{B}$, then $\mathbf{A}^{\intercal}\mathbf{B} = \mathbf{A}^{\intercal}(\mathbf{Y} \ominus_q \mathbf{A})$. Exploiting (64), the minimum compression rate for computing $g(\mathbf{A}, \mathbf{Y})$ given side information $\mathbf{Y}$ is

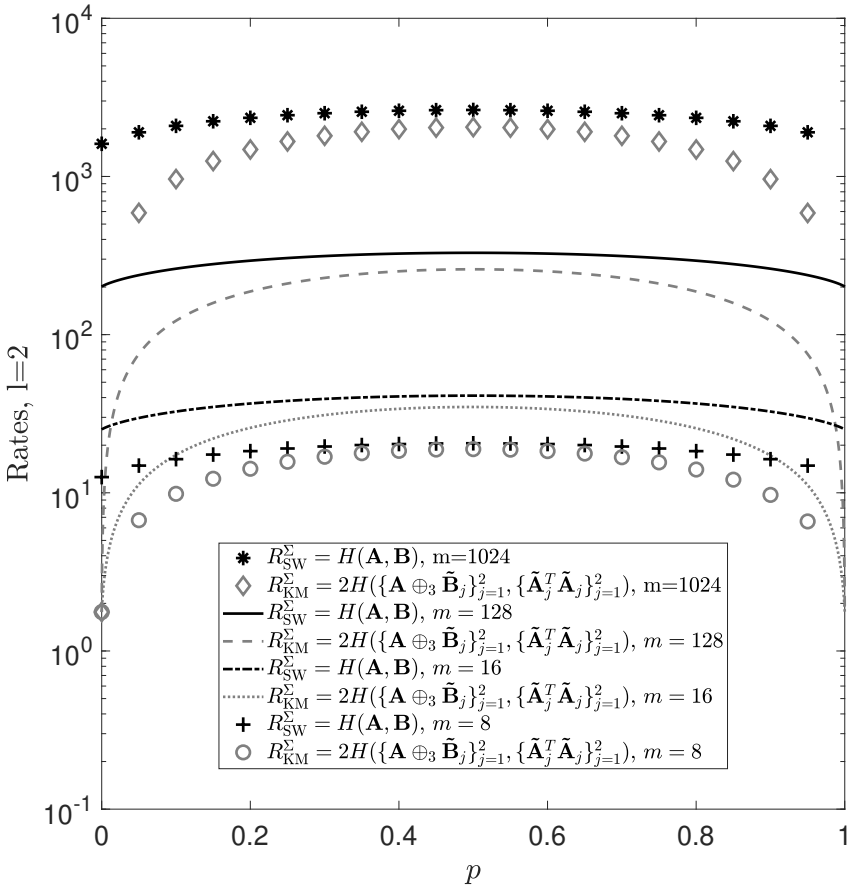

Fig. 3: Rate (in log scale) versus p for the matrix multiplication source network, to compute $\boldsymbol{\mathcal{D}} = \mathbf{A}^\intercal\mathbf{B} \in \mathbb{F}_3^{2\times 2}$, given $\mathbf{A},\mathbf{B} \in \mathbb{F}_3^{m\times l}$, for different m and $l=2$, where the joint source PMF is given in Example 3 (cf. Proposition 3).

equal to $H_{G_{\mathbf{A}}}(\mathbf{A}\,|\,\mathbf{Y})$. To achieve (65), we employ a hybrid coding scheme: first, the structured coding scheme in [13] is used to compute $\mathbf{Y}$; this is followed by the unstructured coding approach of [25] to compute $g(\mathbf{A},\mathbf{Y})$. □

For the matrix multiplication source network, in the symmetric case where $l=1$ and $\mathbf{A}\in\mathbb{F}_2^m$ and $\mathbf{B}\in\mathbb{F}_2^m$, in Figure 4, we contrast the sum-rate performances of various special cases captured by Corollary 1 with a corresponding sum rate $R^{\Sigma}_{\mathrm{KM,sym}}$, and $R^{\Sigma}_{\mathrm{S}} = 2H(\mathbf{A}\oplus_3\mathbf{B})$, which models the sum rate required to compute $\mathbf{A}^\intercal\mathbf{B}$ by embedding the source variables in $\mathbb{F}_3$. This comparison includes the scheme of Slepian-Wolf in [18] with sum rate $R^{\Sigma}_{\mathrm{SW}} = H(\mathbf{A},\mathbf{B})$, and the characteristic graph-based approach with a sum rate $R^{\Sigma}_{\mathrm{KM-OR}} = 2H_q(\mathbf{Y}) + H_{G_{\mathbf{A}}}(\mathbf{A}\,|\,\mathbf{Y})$ where $\mathbf{Y} = \mathbf{A}\oplus_2\mathbf{B}$, and with $R^{\Sigma}_{\mathrm{KM-OR|B}} = H(\mathbf{B}) + H_{G_{\mathbf{A}}}(\mathbf{A}\,|\,\mathbf{B}) \le R^{\Sigma}_{\mathrm{SW}}$, which follows from letting $\mathbf{Y}=\mathbf{B}$ in Proposition 4. We also provide the lower bound given by $R_{\mathrm{f}} = H(\mathbf{A}^\intercal\mathbf{B})$. In Figure 4-(Top-Left), we depict Example 1 with $m=2$, where $(a_1,b_2)\sim\mathrm{DSBS}(p)$, and $(a_2,b_1)\sim\mathrm{DSBS}(p)$. We do not indicate R^{Σ}_{S} and $R^{\Sigma}_{\mathrm{KM-OR}}$, which perform poorly. $R^{\Sigma}_{\mathrm{KM,sym}}$ performs well at low p, and converges to R_{f} as $p\to 0$, and to R^{Σ}_{SW} as $p\to 1$. In Figure 4-(Top-Right), we use $m=1$, where $(a,b)\sim\mathrm{DSBS}(p)$. We indicate R^{Σ}_{S} and $R^{\Sigma}_{\mathrm{KM-OR}}$. At low p values, $R^{\Sigma}_{\mathrm{KM-OR}}$ and R^{Σ}_{SW} converge to R_{f}, whereas R^{Σ}_{S} performs poorly. For large p, structured coding yields low R^{Σ}_{S} and $R^{\Sigma}_{\mathrm{KM-OR}}$. In Figure 4-(Bottom), we depict Example 2 with $m=2$, where $(a_i,b_i)\sim\mathrm{DSBS}(p)$, for each $i\in\{1,2\}$. We also indicate $R^{\Sigma}_{\mathrm{KM-OR}}$. The rate $R^{\Sigma}_{\mathrm{KM,sym}}$ exceeds R^{Σ}_{SW} and is omitted. For any p, $R^{\Sigma}_{\mathrm{S}} < R^{\Sigma}_{\mathrm{SW}}$, and $R^{\Sigma}_{\mathrm{KM-OR}}$ approaches R_{f} for small and large p.

D. Distributed Computation of General Matrix Products via Recursive Dot Products

We next recursively apply the *distributed dot-product computation technique* from Corollary 1 to compute $\boldsymbol{\mathcal{D}} = \mathbf{A}^\intercal\mathbf{B}$, given the general matrix multiplication source network for $q\ge 2$. We then derive the corresponding sum rate, extending beyond the results of Propositions 1 and 3.

Proposition 5. (Distributed computation of square matrix products via recursive application of dot products.) *For the general matrix multiplication source network, for any $q\ge 2$, with even l, and for any $\epsilon\in(0,1)$, the sum rate*

$$R^{\Sigma}_{\mathrm{KM,recursive}}(\mathbf{A},\mathbf{B}) = 2H_q(\{\mathbf{U}_{ij},\,\mathbf{V}_{ij},\,W_{ij}\}_{\,i\le j\in[l]}) \quad (66)$$

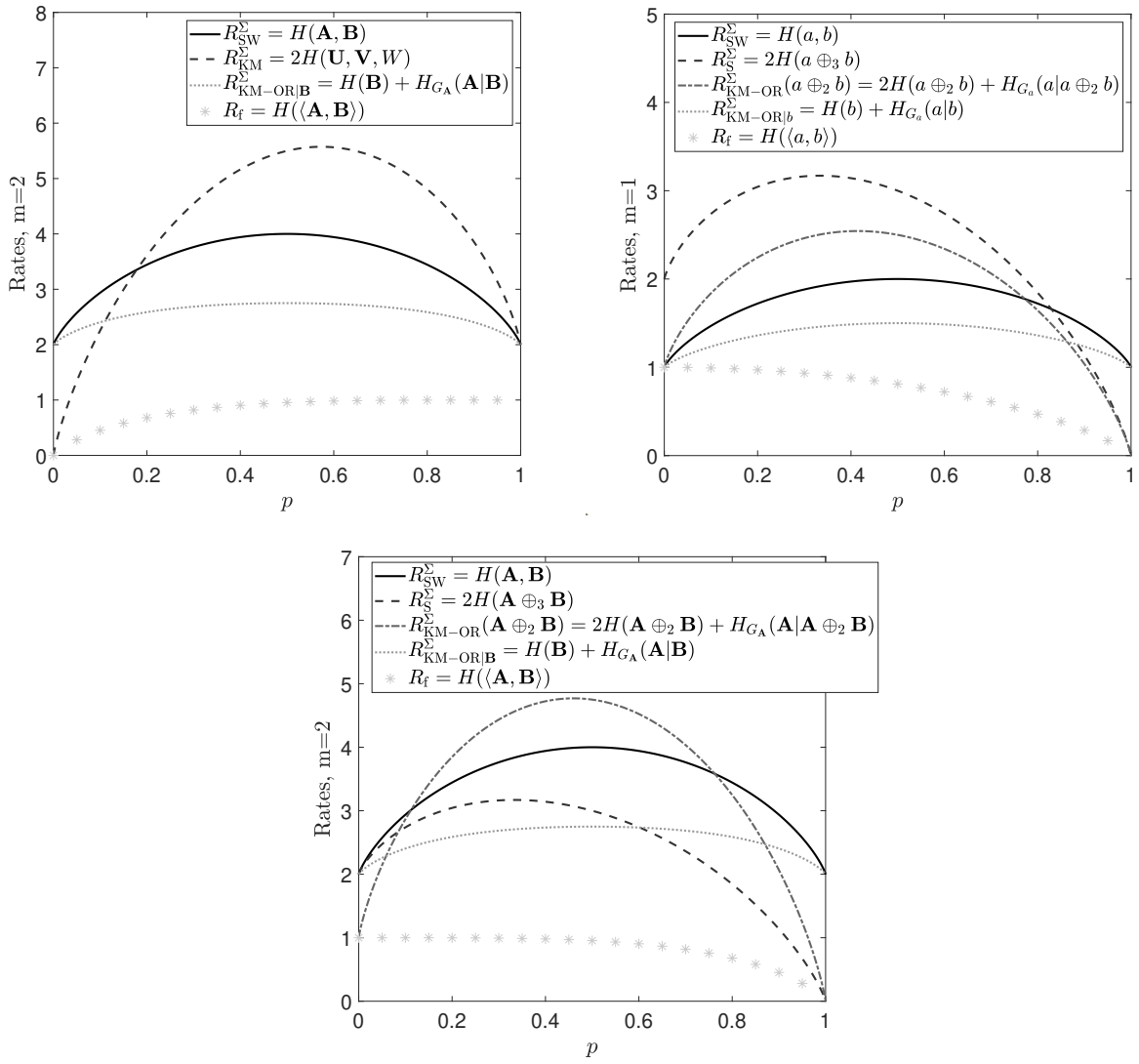

Fig. 4: Rate comparisons for various $P_{\mathbf{A},\mathbf{B}}$. (Top-Left) The asymmetric DSBS model in Example 1 with $m = 2$. (Top-Right) $m = 1$, where $(a, b) \sim \mathrm{DSBS}(p)$. (Bottom) The elementwise DSBS model in Example 2 with $m = 2$.

is achievable, where $\mathbf{A}$ *and* $\mathbf{B}$ *are defined as in (5) and*

$$\mathbf{U}_{ij} = \mathbf{A}_{2i} \oplus_q \mathbf{B}_{1j} \in \mathbb{F}_q^{m/2} \ , \quad \mathbf{V}_{ij} = \mathbf{A}_{1i} \oplus_q \mathbf{B}_{2j} \in \mathbb{F}_q^{m/2} \ ,$$
$$W_{ij} = \mathbf{A}_{2i}^{\intercal}\mathbf{A}_{1i} \oplus_q \mathbf{B}_{1j}^{\intercal}\mathbf{B}_{2j} \in \mathbb{F}_q \ . \tag{67}$$

Proof. Given $\mathbf{A},\ \mathbf{B} \in \mathbb{F}_q^{m\times l}$, and $\boldsymbol{\mathcal{D}} = (d_{ij})_{i,\ j\in[l]} = \mathbf{A}^{\intercal}\mathbf{B} \in \mathbb{F}_q^{l\times l}$, it holds that $d_{ij} = \mathbf{A}_i^{\intercal}\mathbf{B}_j$, for $i,\ j \in [l]$, where $\mathbf{A}_i = \left[\mathbf{A}_{1i}; \mathbf{A}_{2i}\right] \in \mathbb{F}_q^m$ and $\mathbf{B}_j = \left[\mathbf{B}_{1j}; \mathbf{B}_{2j}\right] \in \mathbb{F}_q^m$ represent the i-th and the j-th columns of $\mathbf{A}$ and $\mathbf{B}$, respectively, where $\mathbf{A}_{1i} = (a_{ji})_{j\in[m/2]} \in \mathbb{F}_q^{\frac{m}{2}}$ and $\mathbf{A}_{2i} = (a_{ji})_{j\in[m/2+1,m]} \in \mathbb{F}_q^{\frac{m}{2}}$, similarly for $\mathbf{B}_{1j}$ and $\mathbf{B}_{2j}$. Exploiting Corollary 1 and (67), we observe

$$d_{ij} = \mathbf{U}_{ij}^{\intercal} \cdot \mathbf{V}_{ij} \ominus_q W_{ij} \in \mathbb{F}_q \ , \tag{68}$$

where each component $d_{i'j'}$, for $i' \neq i$ and $j' \neq j$, can be recursively derived using the tuples $\{\mathbf{U}_{ij}, \mathbf{V}_{ij}, W_{ij}\}$, $\{\mathbf{U}_{i'j}, \mathbf{V}_{i'j}, W_{i'j}\}$, and $\{\mathbf{U}_{ij'}, \mathbf{V}_{ij'}, W_{ij'}\}$, which follows from capturing

$$\mathbf{U}_{i'j'} = (\mathbf{U}_{i'j} \oplus_q \mathbf{U}_{ij'}) \ominus_q \mathbf{U}_{ij} \in \mathbb{F}_q^{m/2} \ ,$$
$$\mathbf{V}_{i'j'} = (\mathbf{V}_{i'j} \oplus_q \mathbf{V}_{ij'}) \ominus_q \mathbf{V}_{ij} \in \mathbb{F}_q^{m/2} \ ,$$
$$W_{i'j'} = (W_{i'j} \oplus_q W_{ij'}) \ominus_q W_{ij} \in \mathbb{F}_q \ . \tag{69}$$

Hence, recursively applying Corollary 1 that exploits the structured coding mechanism of Körner-Marton [13], the achievable sum rate for the receiver to recover $\mathbf{A}^{\intercal}\mathbf{B}$ can be determined as $R^{\Sigma}_{\mathrm{KM,recursive}} = 2H_q(\{\mathbf{U}_{ij}, \mathbf{V}_{ij}, W_{ij}\}_{i\leq j\in[l]})$, which gives the achievability result we seek. □

From Proposition 5, the complete computation of $\boldsymbol{\mathcal{D}}$ relies on the set $\{\mathbf{U}_{ij}, \mathbf{V}_{ij}, W_{ij}\}_{i\leq j\in[l]}$. For this setting, we next upper bound the achievable sum rate and the computational complexity of deriving $\boldsymbol{\mathcal{D}}$, measured in terms of the number of multiplications.

Corollary 2. *The achievable sum rate by the encoding scheme outlined in Proposition 5 for the asymptotically lossless computation of* $\boldsymbol{\mathcal{D}} = \mathbf{A}^{\intercal}\mathbf{B} \in \mathbb{F}_q^{l\times l}$ *is upper bounded by*

$$R^{\Sigma}_{\mathrm{KM,recursive}}(\mathbf{A},\mathbf{B}) \leq (m+1)l(l+1) \ . \tag{70}$$

The number of multiplications that the receiver needs to perform to derive $\boldsymbol{\mathcal{D}}$ is

$$\frac{1}{4}ml(l+1)\ . \tag{71}$$

Proof. **Rate.** We employ $R^{\Sigma}_{\mathrm{KM,recursive}}$ from (66), where each tuple $\{\mathbf{U}_{ij},\ \mathbf{V}_{ij},\ W_{ij}\}$, for $i \leq j \in [l]$, in (67) has a total dimension of $2 \cdot m/2 + 1 = m + 1$. Since there are $l(l+1)/2$ such tuples, the total number of bits per source is upper bounded by $(m+1)l(1+l)$. Thus, (70) follows.

Receiver complexity. (71) follows by applying the definitions of $\{\mathbf{U}_{ij}\ ,\mathbf{V}_{ij}\ ,W_{ij}\}$ from (67) and the relation $d_{ij} = \mathbf{U}_{ij}^{\intercal} \cdot \mathbf{V}_{ij} \ominus_q W_{ij}$ in (68) from Proposition 5 for each of the $l(l+1)/2$ tuples, along with the dot product computation cost $m/2$ for $\mathbf{U}_{ij} \in \mathbb{F}_q^{m/2}$ and $\mathbf{V}_{ij} \in \mathbb{F}_q^{m/2}$. □

We next propose a further *recursive application of the dot-product computation technique* in Corollary 1 for distributed computation of a symmetric matrix $\boldsymbol{\mathcal{D}} = \mathbf{A}^{\intercal}\mathbf{B}$, with $q > 2$. The diagonal entries $\{d_{ii}\}_{i\in[l]}$ are calculated first similarly as in Proposition 5. The additional rate required for computing the off-diagonal entries $\{d_{ij}\}_{i<j\in[l]}$ is decided using the symmetry in $\boldsymbol{\mathcal{D}}$.

Proposition 6. (Distributed computation of symmetric matrix products via recursive application of dot products.) *For the matrix multiplication source network, in the symmetric case, for $q > 2$ and odd, and for any $\epsilon \in (0,1)$, the following sum rate is achievable:*

$$\begin{aligned} &R^{\Sigma}_{\mathrm{KM,recursive-sym.}}(\mathbf{A},\ \mathbf{B}) \\ &\quad = 2H_q(\{\mathbf{U}_{ii},\ \mathbf{V}_{ii},\ W_{ii}\}_{i\in[l]}\ ,\{\mathbf{U}_{ij},\ \mathbf{V}_{ij}\}_{\,i<j\in[l]})\ . \end{aligned} \tag{72}$$

Proof. Take two indices $i,\ j \in [l]$ such that $i < j$. When $\boldsymbol{\mathcal{D}} = \mathbf{A}^{\intercal}\mathbf{B}$ is symmetric, it holds that

$$\begin{aligned} d_{ij} &= \mathbf{U}_{ij}^{\intercal} \cdot \mathbf{V}_{ij} \ominus_q W_{ij} \\ &= \mathbf{U}_{ji}^{\intercal} \cdot \mathbf{V}_{ji} \ominus_q W_{ji} = d_{ji} \in \mathbb{F}_q\ . \end{aligned} \tag{73}$$

Exploiting the symmetry in $\boldsymbol{\mathcal{D}}$, and using (69), it holds that

$$\begin{aligned} d_{ji} = &((\mathbf{U}_{ii} \oplus_q \mathbf{U}_{jj}) \ominus_q \mathbf{U}_{ij})^{\intercal} \cdot ((\mathbf{V}_{ii} \oplus_q \mathbf{V}_{jj}) \ominus_q \mathbf{V}_{ij}) \\ &\ominus_q ((W_{ii} \oplus_q W_{jj}) \ominus_q W_{ij}) \in \mathbb{F}_q\ . \end{aligned} \tag{74}$$

The receiver, using the relations (73) and (74), can compute directly the term

$$d_{ij} = \frac{1}{2}(d_{ij} \oplus_q d_{ji})\ , \tag{75}$$

provided that $q > 2$ and odd. Given $\{\mathbf{U}_{ii},\ \mathbf{V}_{ii},\ W_{ii}\}$, the receiver can decode d_{ij} if in addition $\{\mathbf{U}_{ij},\ \mathbf{V}_{ij}\}$ are also given. Exploiting the symmetry in $\boldsymbol{\mathcal{D}}$ and (75), it suffices to determine the diagonal and the upper triangular entries, namely $\{d_{ij}\}_{i\leq j\in[l]}$, to achieve the sum rate in (72). □

From Proposition 6, the computation of $\boldsymbol{\mathcal{D}}$ relies on the diagonal tuples $\{\mathbf{U}_{ii},\ \mathbf{V}_{ii},\ W_{ii}\}_{i\in[l]}$ to determine $\{d_{ii}\}_{i\in[l]}$, and on the off-diagonal tuples $\{\mathbf{U}_{ij},\ \mathbf{V}_{ij}\}_{i<j\in[l]}$, noting that $\boldsymbol{\mathcal{D}}$ is symmetric.

Corollary 3. *The achievable sum rate by the encoding scheme outlined in Proposition 6 for the asymptotically lossless computation of $\boldsymbol{\mathcal{D}} = \mathbf{A}^{\intercal}\mathbf{B} \in \mathbb{F}_q^{l\times l}$ is upper bounded by*

$$R^{\Sigma}_{\mathrm{KM,recursive-sym.}}(\mathbf{A},\ \mathbf{B}) \leq ml(l+1) + 2l\ . \tag{76}$$

The number of multiplications that the receiver needs to perform to derive $\boldsymbol{\mathcal{D}}$ is

$$\frac{1}{2}ml^2\ . \tag{77}$$

Proof. **Rate.** We employ $R^{\Sigma}_{\mathrm{KM,recursive-sym.}}$ from (72), where each tuple $\{\mathbf{U}_{ii},\ \mathbf{V}_{ii},\ W_{ii}\}$, for $i \in [l]$, has a total dimension of $m+1$, and there are l such tuples, and each tuple $\{\mathbf{U}_{ij},\ \mathbf{V}_{ij}\}$, for $i < j \in [l]$, has a total dimension of m, and there are $(l^2-l)/2$ such tuples, the total number of bits per source is upper bounded by $ml(l+1)/2 + l$. This leads directly to (76).

Receiver complexity. Equation (77) follows by noting that each diagonal term d_{ii}, $i \in [l]$, requires a dot product of size $m/2$. Due to the symmetry of $\boldsymbol{\mathcal{D}}$, the remaining $(l^2-l)/2$ off-diagonal terms can be computed using the relation $d_{ij} = \mathbf{U}_{ij}^{\intercal} \cdot \mathbf{V}_{ij} \ominus_q W_{ij} = d_{ji}$ from (73), as well as the alternative expression in (75), each involving two dot products of size $m/2$. □

We next tighten the result in Proposition 6 via a *nested application of the dot-product computation technique* in Corollary 1. In this approach, $\{d_{ii}\}_{i\in[l]}$ are calculated first using the same technique as in Proposition 5, and then the additional rate required for computing $\{d_{ij}\}_{i<j\in[l]}$ is determined as a function of $\{\mathbf{U}_{ij},\ \mathbf{V}_{ij},\ W_{ij}\}_{i,\ j\in[l]}$, as described next.

Proposition 7. (Distributed computation of symmetric matrix products via nested application of dot products.) *For the matrix multiplication source network, in the symmetric case, for $q > 2$ and odd, and for any $\epsilon \in (0,1)$, the following sum rate is achievable:*

$$\begin{aligned} &R^{\Sigma}_{\mathrm{KM,nes.-sym.}}(\mathbf{A},\mathbf{B}) = 2H_q\Big(\{\mathbf{U}_{ii},\ \mathbf{V}_{ii},\ W_{ii}\}_{i\in[l]}\ , \\ &\Big\{\mathbf{U}_{ij}(\frac{m}{4}+1:\frac{m}{2}) \oplus_q \mathbf{V}_{ij}(1:\frac{m}{4})\ , \\ &\mathbf{U}_{ij}(1:\frac{m}{4}) \oplus_q \mathbf{V}_{ij}(\frac{m}{4}+1:\frac{m}{2})\ , \\ &\mathbf{U}_{ij}(\frac{m}{4}+1:\frac{m}{2})^{\intercal} \cdot \mathbf{U}_{ij}(1:\frac{m}{4}) \\ &\oplus_q \mathbf{V}_{ij}(1:\frac{m}{4})^{\intercal} \cdot \mathbf{V}_{ij}(\frac{m}{4}+1:\frac{m}{2}) \\ &\ominus_q (\boldsymbol{\alpha}^{\intercal}(\mathbf{U}_{ii},\mathbf{U}_{jj})\mathbf{U}_{ij} + \boldsymbol{\beta}^{\intercal}(\mathbf{V}_{ii},\mathbf{V}_{jj})\mathbf{V}_{ij})\Big\}_{\,i<j\in[l]}\Big)\ , \end{aligned} \tag{78}$$

where $\{\mathbf{U}_{ij},\ \mathbf{V}_{ij},\ W_{ij}\}_{i,\ j\in[l]}$ are defined similarly to (67) in Proposition 5, and $\boldsymbol{\alpha}(\mathbf{U}_{ii},\mathbf{U}_{jj}) \in \mathbb{F}_q^{m/2}$ and $\boldsymbol{\beta}(\mathbf{V}_{ii},\mathbf{V}_{jj}) \in \mathbb{F}_q^{m/2}$ represent coefficient matrices, which are given as follows:

$$\boldsymbol{\alpha}(\mathbf{U}_{ii},\mathbf{U}_{jj}) = \frac{1}{2}\begin{bmatrix} \mathbf{U}_{ii}(\frac{m}{4}+1:\frac{m}{2}) \oplus_q \mathbf{U}_{jj}(\frac{m}{4}+1:\frac{m}{2}) \\ \mathbf{U}_{ii}(1:\frac{m}{4}) \oplus_q \mathbf{U}_{jj}(1:\frac{m}{4}) \end{bmatrix}, \tag{79}$$

$$\boldsymbol{\beta}(\mathbf{V}_{ii},\mathbf{V}_{jj}) = \frac{1}{2}\begin{bmatrix} \mathbf{V}_{ii}(\frac{m}{4}+1:\frac{m}{2}) \oplus_q \mathbf{V}_{jj}(\frac{m}{4}+1:\frac{m}{2}) \\ \mathbf{V}_{ii}(1:\frac{m}{4}) \oplus_q \mathbf{V}_{jj}(1:\frac{m}{4}) \end{bmatrix}. \tag{80}$$

Proof. Given diagonal tuples $\{\mathbf{U}_{ii}, \mathbf{V}_{ii}, W_{ii}\}_{i\in[l]}$, and substituting the expansion from (75), the additional rate needed for the receiver to reconstruct $d_{ij} \in \mathbb{F}_q$, for $i < j \in [l]$, is:

$$\begin{aligned}
&H_q(d_{ij} \,|\, \{\mathbf{U}_{ii}, \mathbf{V}_{ii}, W_{ii}\}_{i\in[l]}) \\
&\quad = H_q((\mathbf{U}_{ii} \oplus_q \mathbf{U}_{jj})^\intercal \cdot (\mathbf{V}_{ii} \oplus_q \mathbf{V}_{jj}) \\
&\quad \ominus_q \mathbf{U}_{ij}^\intercal \cdot (\mathbf{V}_{ii} \oplus_q \mathbf{V}_{jj}) \ominus_q (\mathbf{U}_{ii} \oplus_q \mathbf{U}_{jj})^\intercal \cdot \mathbf{V}_{ij} \\
&\quad \ominus_q (W_{ii} \oplus_q W_{jj}) \\
&\quad \oplus_q 2\mathbf{U}_{ij}^\intercal \cdot \mathbf{V}_{ij} \,|\, \{\mathbf{U}_{ii}, \mathbf{V}_{ii}, W_{ii}\}_{i\in[l]}) \\
&\quad = H_q(\mathbf{U}_{ij}^\intercal \cdot \mathbf{V}_{ij} \ominus_q \frac{1}{2}\mathbf{U}_{ij}^\intercal \cdot (\mathbf{V}_{ii} \oplus_q \mathbf{V}_{jj}) \\
&\quad \ominus_q \frac{1}{2}(\mathbf{U}_{ii} \oplus_q \mathbf{U}_{jj})^\intercal \cdot \mathbf{V}_{ij} \,|\, \{\mathbf{U}_{ii}, \mathbf{V}_{ii}, W_{ii}\}_{i\in[l]}) \\
&\quad = H_q\big(\bar{\mathbf{U}}(ij)^\intercal \cdot \bar{\mathbf{V}}(ij) \,\big|\, \{\mathbf{U}_{ii}, \mathbf{V}_{ii}, W_{ii}\}_{i\in[l]}\big) \ ,
\end{aligned} \tag{81}$$

where the last step follows from the following substitutions:

$$\begin{aligned}
\bar{\mathbf{U}}(ij) &= \mathbf{U}_{ij} \ominus_q \frac{1}{2}(\mathbf{U}_{ii} \oplus_q \mathbf{U}_{jj}) \in \mathbb{F}_q^{m/2} \ , \\
\bar{\mathbf{V}}(ij) &= \mathbf{V}_{ij} \ominus_q \frac{1}{2}(\mathbf{V}_{ii} \oplus_q \mathbf{V}_{jj}) \in \mathbb{F}_q^{m/2} \ .
\end{aligned} \tag{82}$$

Note that $\bar{\mathbf{U}}(ij)^\intercal\bar{\mathbf{V}}(ij) = \langle\, \bar{\mathbf{U}}(ij), \bar{\mathbf{V}}(ij)\,\rangle$. Leveraging Corollary 1 on dot-product computation, and letting $\bar{\mathbf{U}}(ij)^\intercal = \big[\bar{\mathbf{U}}_1(ij)^\intercal \quad \bar{\mathbf{U}}_2(ij)^\intercal\big]$ and $\bar{\mathbf{V}}(ij)^\intercal = \big[\bar{\mathbf{V}}_1(ij)^\intercal \quad \bar{\mathbf{V}}_2(ij)^\intercal\big]$ where $\bar{\mathbf{U}}_1(ij), \bar{\mathbf{U}}_2(ij), \bar{\mathbf{V}}_1(ij), \bar{\mathbf{V}}_2(ij) \in \mathbb{F}_q^{m/4}$, we see from (81) and (82) that the receiver can reconstruct d_{ij} using $\bar{\mathbf{U}}(ij)^\intercal\bar{\mathbf{V}}(ij)$, which can be derived from the following two linear terms:

$$\begin{aligned}
&\bar{\mathbf{U}}_2(ij) \oplus_q \bar{\mathbf{V}}_1(ij) \\
&\quad = \mathbf{U}_{ij}(\frac{m}{4}+1:\frac{m}{2}) \\
&\quad \ominus_q \frac{1}{2}\big(\mathbf{U}_{ii}(\frac{m}{4}+1:\frac{m}{2}) \oplus_q \mathbf{U}_{jj}(\frac{m}{4}+1:\frac{m}{2})\big) \\
&\quad \oplus_q \mathbf{V}_{ij}(1:\frac{m}{4}) \\
&\quad \ominus_q \frac{1}{2}\big(\mathbf{V}_{ii}(1:\frac{m}{4}) \oplus_q \mathbf{V}_{jj}(1:\frac{m}{4})\big) \in \mathbb{F}_q^{m/4} \ ,
\end{aligned} \tag{83}$$

$$\begin{aligned}
&\bar{\mathbf{U}}_1(ij) \oplus_q \bar{\mathbf{V}}_2(ij) \\
&\quad = \mathbf{U}_{ij}(1:\frac{m}{4}) \\
&\quad \ominus_q \frac{1}{2}\big(\mathbf{U}_{ii}(1:\frac{m}{4}) \oplus_q \mathbf{U}_{jj}(1:\frac{m}{4})\big) \\
&\quad \oplus_q \mathbf{V}_{ij}(\frac{m}{4}+1:\frac{m}{2}) \\
&\quad \ominus_q \frac{1}{2}\big(\mathbf{V}_{ii}(\frac{m}{4}+1:\frac{m}{2}) \oplus_q \mathbf{V}_{jj}(\frac{m}{4}+1:\frac{m}{2})\big) \in \mathbb{F}_q^{m/4}
\end{aligned} \tag{84}$$

and the following non-linear term

$$\begin{aligned}
&\bar{\mathbf{U}}_2(ij)^\intercal\bar{\mathbf{U}}_1(ij) \oplus_q \bar{\mathbf{V}}_1(ij)^\intercal\bar{\mathbf{V}}_2(ij) \\
&= \mathbf{U}_{ij}(\frac{m}{4}+1:\frac{m}{2})^\intercal \cdot \mathbf{U}_{ij}(1:\frac{m}{4}) \\
&\ominus_q \mathbf{U}_{ij}(\frac{m}{4}+1:\frac{m}{2})^\intercal \cdot \frac{1}{2}\big(\mathbf{U}_{ii}(1:\frac{m}{4}) \oplus_q \mathbf{U}_{jj}(1:\frac{m}{4})\big) \\
&\ominus_q \frac{1}{2}\big(\mathbf{U}_{ii}(\frac{m}{4}+1:\frac{m}{2}) \oplus_q \mathbf{U}_{jj}(\frac{m}{4}+1:\frac{m}{2})\big)^\intercal \cdot \mathbf{U}_{ij}(1:\frac{m}{4}) \\
&\oplus_q \frac{1}{2}\big(\mathbf{U}_{ii}(\frac{m}{4}+1:\frac{m}{2}) \oplus_q \mathbf{U}_{jj}(\frac{m}{4}+1:\frac{m}{2})\big)^\intercal \\
&\quad \cdot \frac{1}{2}\big(\mathbf{U}_{ii}(1:\frac{m}{4}) \oplus_q \mathbf{U}_{jj}(1:\frac{m}{4})\big) \\
&\oplus_q \mathbf{V}_{ij}(1:\frac{m}{4})^\intercal \cdot \mathbf{V}_{ij}(\frac{m}{4}+1:\frac{m}{2}) \ominus_q \mathbf{V}_{ij}(1:\frac{m}{4})^\intercal \\
&\quad \cdot \frac{1}{2}\big(\mathbf{V}_{ii}(\frac{m}{4}+1:\frac{m}{2}) \oplus_q \mathbf{V}_{jj}(\frac{m}{4}+1:\frac{m}{2})\big) \\
&\ominus_q \frac{1}{2}\big(\mathbf{V}_{ii}(1:\frac{m}{4}) \oplus_q \mathbf{V}_{jj}(1:\frac{m}{4})\big)^\intercal \cdot \mathbf{V}_{ij}(\frac{m}{4}+1:\frac{m}{2}) \\
&\oplus_q \frac{1}{2}\big(\mathbf{V}_{ii}(1:\frac{m}{4}) \oplus_q \mathbf{V}_{jj}(1:\frac{m}{4})\big)^\intercal \\
&\quad \cdot \frac{1}{2}\big(\mathbf{V}_{ii}(\frac{m}{4}+1:\frac{m}{2}) \oplus_q \mathbf{V}_{jj}(\frac{m}{4}+1:\frac{m}{2})\big) \in \mathbb{F}_q \ .
\end{aligned} \tag{85}$$

Exploiting the side information $\{\mathbf{U}_{ii}, \mathbf{V}_{ii}, W_{ii}\}_{i=1}^{l}$, we can significantly simplify (83), (84), and (85), and deduce that the receiver can reconstruct d_{ij} from

$$\mathbf{U}_{ij}(\frac{m}{4}+1:\frac{m}{2}) \oplus_q \mathbf{V}_{ij}(1:\frac{m}{4}) \in \mathbb{F}_q^{m/4} \ , \tag{86i}$$

$$\mathbf{U}_{ij}(1:\frac{m}{4}) \oplus_q \mathbf{V}_{ij}(\frac{m}{4}+1:\frac{m}{2}) \in \mathbb{F}_q^{m/4} \ , \tag{86ii}$$

$$\begin{aligned}
&\mathbf{U}_{ij}(\frac{m}{4}+1:\frac{m}{2})^\intercal \cdot \mathbf{U}_{ij}(1:\frac{m}{4}) \\
&\oplus_q \mathbf{V}_{ij}(1:\frac{m}{4})^\intercal \cdot \mathbf{V}_{ij}(\frac{m}{4}+1:\frac{m}{2}) \\
&\ominus_q (\boldsymbol{\alpha}^\intercal(\mathbf{U}_{ii}, \mathbf{U}_{jj}) \cdot \mathbf{U}_{ij} + \boldsymbol{\beta}^\intercal(\mathbf{V}_{ii}, \mathbf{V}_{jj}) \cdot \mathbf{V}_{ij}) \in \mathbb{F}_q \ ,
\end{aligned} \tag{86iii}$$

where $\boldsymbol{\alpha}(\mathbf{U}_{ii}, \mathbf{U}_{jj}) \in \mathbb{F}_q^{m/2}$ given in (79) and $\boldsymbol{\beta}(\mathbf{V}_{ii}, \mathbf{V}_{jj}) \in \mathbb{F}_q^{m/2}$ given in (80) represent coefficient matrices that are determined as functions of $\mathbf{U}_{ii}, \mathbf{U}_{jj}$ and $\mathbf{V}_{ii}, \mathbf{V}_{jj}$, respectively.

From (81)-(86), which detail the reconstruction of $\{d_{ij}\}_{i<j\in[l]}$, we get the final result. □

Using the sum rate $R^{\Sigma}_{\mathrm{KM,nes.-sym.}}$ given in (78) of Proposition 7, we next derive the rate and complexity of distributed lossless computation of $\boldsymbol{\mathcal{D}} = \mathbf{A}^\intercal\mathbf{B} \in \mathbb{F}_q^{l\times l}$.

Corollary 4. *The achievable sum rate by the encoding scheme outlined in Proposition 7 for the asymptotically lossless computation of $\boldsymbol{\mathcal{D}} = \mathbf{A}^\intercal\mathbf{B} \in \mathbb{F}_q^{l\times l}$ is upper bounded by*

$$R^{\Sigma}_{\mathrm{KM,nes.-sym.}}(\mathbf{A}, \mathbf{B}) \leq \frac{1}{2}ml(l+3) + l(l+1) \ . \tag{87}$$

The number of multiplications that the receiver needs to perform to derive $\boldsymbol{\mathcal{D}}$ is

$$\frac{1}{8}ml(7l-3) \ . \tag{88}$$

Proof. **Rate.** We employ $R^{\Sigma}_{\mathrm{KM,nes.-sym.}}$ from (78), where each $\{\mathbf{U}_{ii}, \mathbf{V}_{ii}, W_{ii}\}$, for $i \in [l]$, has dimension $m+1$, contributing a total of $(m+1)l$ bits. For each off-diagonal tuple with $i < j$, the dimension is $m/2+1$, as seen from (86) in Proposition 7, and there are $l(l-1)/2$ such tuples. Hence, the rate per source is at most $(m+1)l + (m/2+1)l(l-1)/2$, which yields (87).

Receiver complexity. Employing the definitions of $\{\mathbf{U}_{ij}\ , \mathbf{V}_{ij}\ , W_{ij}\}$ from (67) and d_{ij} in (68) (Proposition 5), the total complexity of computing $\{d_{ii}\}_{i\in[l]}$ from $\{\mathbf{U}_{ii}\ , \mathbf{V}_{ii}\ , W_{ii}\}$ is

$$\frac{1}{2}ml \ . \tag{89}$$

Given $\{d_{ii}\}_{i\in[l]}$, to reconstruct $\{d_{ij}\}_{i<j\in[l]}$, we use (81) along with the linear terms in (83)-(84), and the non-linear term in (85), as detailed in Proposition 7. The linear terms incur no multiplicative cost. The non-linear term in (85) involves two dot products — $\bar{\mathbf{U}}_2(ij)^\intercal \cdot \bar{\mathbf{U}}_1(ij)$ and $\bar{\mathbf{V}}_1(ij)^\intercal \cdot \bar{\mathbf{V}}_2(ij)$ — each of complexity $m/4$. Additionally, reconstructing d_{ij} via (86) requires one dot product from (86i)-(86ii), with complexity $m/4$, and four dot products in (86iii), each with complexity $m/4$, yielding a total per-pair cost of $2 \cdot m/4 + m/4 + 4 \cdot m/4 = 7m/4$. Because there are $(l^2 - l)/2$ such tuples, the total complexity of reconstructing $\{d_{ij}\}_{i<j\in[l]}$ is

$$\frac{7}{8} ml(l-1) \ . \tag{90}$$

Combining this with the diagonal complexity from (89) yields the expression in (88). □

Our structured schemes in Propositions 5-7 reduce the computational complexity of distributed matrix multiplication (see Table I). While Strassen-like algorithms [57] and learning-based techniques [42] fall outside our scope, they can be incorporated to further reduce the complexity.

IV. Converses

In this section, for the proposed *matrix multiplication source network*, drawing on Lemmas 1-3 of [14], we first establish necessary conditions on square matrix products, as $q \to \infty$ (Theorem 4) and products of full-rank matrices, with $q \geq 2$ (Theorem 5) on the rate pair $(R_1,\ R_2)$, calibrating the strong converse bound of Ahlswede-Gács-Körner [22] and the Han-Kobayashi approach [14], respectively. To investigate the optimality gaps of our achievability schemes in Section III, we subsequently specialize these conditions to the products of independently and uniformly drawn source matrices from $\mathbb{F}_q^{m\times l}$, as $q \to \infty$ (Corollary 5), then to symmetric matrix products (Proposition 8), and square matrix products (Proposition 9), each with $q = 2$, respectively.

We next provide Lemmas 1-3 of [14], which establish converses on (R_1, R_2) for distributed computing of an arbitrary function $Z = f(X, Y)$ taking values in $\mathcal{Z} \subseteq \mathbb{F}_q$, given separately encoded correlated $(X,\ Y)$ with values in $\mathcal{X} \subseteq \mathbb{F}_q$ and $\mathcal{Y} \subseteq \mathbb{F}_q$, respectively. Let the set $\mathcal{P}$ be defined by $(X, Y) \in \mathcal{P}$ if and only if $P_{X,Y}(k,\ell) = \mathbb{P}(X = k,\ Y = \ell) > 0$, for all $k \in \mathcal{X},\ \ell \in \mathcal{Y}$. For this setting, the *function matrix* denoted by $(f(k,\ell))_{k\in\mathcal{X},\ \ell\in\mathcal{Y}}$ is an $|\mathcal{X}| \times |\mathcal{Y}|$ matrix with the (k,ℓ)-th component being $f(k,\ell)$, where k and ℓ indicate the rows and columns, respectively.

Lemma 6. (Han-Kobayashi [14, Lemmas 1-3].) *Let (X, Y) be any element of $\mathcal{P}$.*

(Condition 1) **[14, Lemma 1].** *If any two distinct rows of the function matrix f are different, then any achievable (R_1, R_2) for f has to satisfy $R_1 \geq H_q(X \,|\, Y)$.*

(Condition 2) **[14, Lemma 2].** *If any two distinct columns of the function matrix f are different, then any achievable (R_1, R_2) for f has to satisfy $R_2 \geq H_q(Y \,|\, X)$.*

(Condition 3) **[14, Lemma 3].** *If, in addition to Conditions 1 and 2, the condition $f(k_1, \ell_1) \neq f(k_2, \ell_2)$ for any (k_1, ℓ_1), (k_2, ℓ_2) with $k_1 \neq k_2$, $\ell_1 \neq \ell_2$ holds, then any achievable (R_1, R_2) for f has to satisfy $R_1 + R_2 \geq H_q(X, Y)$.*

In their seminal work [14], Han and Kobayashi have identified necessary and sufficient conditions — namely, that $f(k,\ell)$, for $k \in \mathcal{X}$, $\ell \in \mathcal{Y}$, satisfies the three properties in Lemma 6 — for the achievable rate region for distributed computing of $Z = f(X, Y)$ to coincide with the Slepian-Wolf region [18] whenever $P_{X,Y}(k,\ell) > 0$ for all $k \in \mathcal{X}$, $\ell \in \mathcal{Y}$ [14, Theorem 1].

To compute the dot product $d = \langle \mathbf{A}, \mathbf{B} \rangle \in \mathbb{F}_q$ in the proposed matrix multiplication source network, Conditions 1-2 hold, while Condition 3 does not. Thus, by [14, Theorem 1], the sum rate $R_1 + R_2$ can fall below $R_{\mathrm{SW}}^{\Sigma}(\mathbf{A},\ \mathbf{B})$. Likewise, for computing $\boldsymbol{\mathcal{D}} = \mathbf{A}^\intercal\mathbf{B} \in \mathbb{F}_q^{l\times l}$, each entry of $\mathbf{A}^\intercal\mathbf{B}$ being a dot product ensures that the same conditions in [14] apply.

We next present a *strong converse* to the *general matrix multiplication source network*, for $\mathbf{A}$ and $\mathbf{B}$, chosen independently and uniformly from $\mathbb{F}_q^{m\times l}$, as $q \to \infty$. We will subsequently derive an additional converse (Theorem 5) for products of full-rank matrices, for any $q \geq 2$.

Theorem 4. (Strong converse for a square matrix product.) *For the general matrix multiplication source network, for random matrices $\mathbf{A}$ and $\mathbf{B}$, chosen independently and uniformly from $\mathbb{F}_q^{m\times l}$, as $q \to \infty$, the set of achievable rates must satisfy*

$$R_1,\ R_2 \geq l \cdot \min\{l, m\} \ . \tag{91}$$

Proof. By the strong converse to the source coding theorem with side information [22], if one variable is available as side information, then we must have

$$R_1 \geq H_q(\mathbf{A}^\intercal\mathbf{B} \,|\, \mathbf{B}) \ , \quad R_2 \geq H_q(\mathbf{A}^\intercal\mathbf{B} \,|\, \mathbf{A}) \ . \tag{92}$$

We next restate [39, Lemma 2], which we exploit to subsequently present the *strong converse*.

Lemma 7. (Entropy of square matrix products [39, Lemma 2].) *Let $\mathbf{A}$ and $\mathbf{B}$ be random matrices drawn independently and uniformly from $\mathbb{F}_q^{m\times l}$. Then,*

$$\lim_{q\to\infty} H_q(\mathbf{A}^\intercal\mathbf{B}) = 2l \cdot \min\{m,\ l\} - \min\{m,\ l\}^2 \ , \tag{93}$$

$$\lim_{q\to\infty} H_q(\mathbf{A}^\intercal\mathbf{B} \,|\, \mathbf{A}) = \lim_{q\to\infty} H_q(\mathbf{A}^\intercal\mathbf{B} \,|\, \mathbf{B}) = l \cdot \min\{l, m\} \ . \tag{94}$$

Proof of Lemma 7. We provide a sketch of the proof by employing Lemmas 8-10 from [39]. As $q \to \infty$, when $\mathbf{A}$ and $\mathbf{B}$ are independent, and when $\mathbf{B}$ is drawn independently and uniformly from $\mathbb{F}_q^{m\times m}$, [39, Lemma 8] yields $R_1 \geq \lim_{q\to\infty} H_q(\mathbf{A}^\intercal\mathbf{B} \,|\, \mathbf{B}) = H_q(\mathbf{A})$. If both $\mathbf{A}$ and $\mathbf{B}$ are drawn independently and uniformly over $\mathbb{F}_q^{m\times l}$, if $m \geq l$, then [39, Lemma 9] yields $R_1 \geq \lim_{q\to\infty} H_q(\mathbf{A}^\intercal\mathbf{B} \,|\, \mathbf{B}) = l^2$, and if $m < l$, then [39, Lemma 10] yields $R_1 \geq \lim_{q\to\infty} H_q(\mathbf{A}^\intercal\mathbf{B} \,|\, \mathbf{B}) = H_q(\mathbf{A}) = lm$. This concludes the proof of Lemma 7. □

	Sum-rate upper bound	Complexity
Recursive dot prods. for general matrix prods.	$(m+1)l(l+1)$ (Eq. (70))	$ml(l+1)/4$ (Eq. (71))
Recursive dot prods. for symmetric matrix prods.	$ml(l+1)+2l$ (Eq. (76))	$ml^2/2$ (Eq. (77))
Nested dot prods. for symmetric matrix prods.	$ml(l+3)/2+l(l+1)$ (Eq. (87))	$ml(7l-3)/8$ (Eq. (88))

TABLE I: Distributed computation of general matrix products via recursive dot products: comparison of sum rates and receiver-side computational complexity for computing $\boldsymbol{\mathcal{D}}$.

We now proceed with the proof of Theorem 4. Adapting the strong converse in Lemma 7 (cf. (94)) to the case where both $\mathbf{A}$ and $\mathbf{B}$ are independently and uniformly distributed over $\mathbb{F}_q^{m\times l}$, in the limit as $q\to\infty$, we obtain (91). □

In Theorem 4, $\mathbf{A}$ and $\mathbf{B}$ drawn independently and uniformly from $\mathbb{F}_q^{m\times l}$, as $q\to\infty$. Hence, in this setup, $R_{\mathrm{SW}}^{\Sigma}=\lim_{q\to\infty} H_q(\mathbf{A},\mathbf{B})=2lm$, and no further compression is possible in direct transmission when the goal is to jointly recover $(\mathbf{A}^n,\mathbf{B}^n)$. However, Condition 3 of Lemma 6 indicates that compression can yield gains in the distributed computation of $\mathbf{A}^{\intercal}\mathbf{B}$, as opposed to directly transmitting the sources. From Theorem 4, structured source coding for computing $\mathbf{A}^{\intercal}\mathbf{B}$, rather than directly compressing $(\mathbf{A},\mathbf{B})$, may yield savings only when $l<m$, where (91) yields $R_1+R_2\geq 2l^2$. We next tighten Theorem 4, under which $\mathbf{A}$ and $\mathbf{B}$ are full rank with probability 1 as $q\to\infty$ [58], for full-rank matrix-product operations with any $q\geq 2$.

Theorem 5. (A tight converse for the product of full-rank matrices.) *For the general matrix multiplication source network, for any $q\geq 2$, with $m,\ l>1$, the achievable rates must satisfy*

$$R_1\geq H_q(\mathbf{A}\mid\mathbf{B})\ ,\quad R_2\geq H_q(\mathbf{B}\mid\mathbf{A})\ , \tag{95}$$

establishing the following lower bound on the achievable sum rate for computing $\mathbf{A}^{\intercal}\mathbf{B}$:

$$R_{\mathrm{HK}}^{\Sigma}(\mathbf{A},\ \mathbf{B})\geq H_q(\mathbf{A}\mid\mathbf{B})+H_q(\mathbf{B}\mid\mathbf{A})\ . \tag{96}$$

Proof. For distributed computing of $\boldsymbol{\mathcal{D}}=\mathbf{A}^{\intercal}\mathbf{B}$, pick k_1 and k_2 be two distinct rows of $f(k,\ell)$ corresponding to source matrices $\mathbf{A}^{(k_1)}$ and $\mathbf{A}^{(k_2)}$. Hence, $f(k_1,\ell)-f(k_2,\ell)=(\mathbf{A}^{(k_1)}-\mathbf{A}^{(k_2)})\mathbf{B}^{(\ell)}$ varies as a function of $\mathbf{B}^{(\ell)}$, rendering Condition 1 true. Similarly, picking ℓ_1 and ℓ_2 be two distinct columns of $f(k,\ell)$ corresponding to $\mathbf{B}^{(\ell_1)}$ and $\mathbf{B}^{(\ell_2)}$, it can be shown that Condition 2 holds true. Hence, from Lemma 6, any achievable rate must satisfy (95). □

Exploiting the relation $H_q(\mathbf{A},\mathbf{B})\geq H_q(\mathbf{A}^{\intercal}\mathbf{B},\mathbf{B})$ and subtracting $H_q(\mathbf{B})$ from both sides, we infer that $H_q(\mathbf{A}\,|\,\mathbf{B})\geq H_q(\mathbf{A}^{\intercal}\mathbf{B}\,|\,\mathbf{B})$. Similarly, $H_q(\mathbf{B}\,|\,\mathbf{A})\geq H_q(\mathbf{A}^{\intercal}\mathbf{B}\,|\,\mathbf{A})$. Thus, given full-rank $\mathbf{A},\mathbf{B}\in\mathbb{F}_q^{m\times l}$, as $q\to\infty$, the rate bounds in Theorem 5 are tighter than those in Theorem 4.

We next demonstrate the tightness of Theorem 3 under the setting described in Lemma 7.

Corollary 5. (Optimality gap for square matrix products in the limit as $q\to\infty$.) *Consider the general matrix multiplication source network, as $q\to\infty$, and for any $\epsilon\in(0,1)$, for random matrices $\mathbf{A}$ and $\mathbf{B}$ drawn independently and uniformly from $\mathbb{F}_q^{m\times l}$. Then, the optimality gap of the scheme in Theorem 3 from the strong converse in Theorem 4 is*

$$\frac{\lim\limits_{q\to\infty} R_{\mathrm{AH}}^{\Sigma}(\mathbf{A},\ \mathbf{B})}{\lim\limits_{q\to\infty}\left(H_q(\mathbf{A}^{\intercal}\mathbf{B}\,|\,\mathbf{B})+H_q(\mathbf{A}^{\intercal}\mathbf{B}\,|\,\mathbf{A})\right)}=1\ . \tag{97}$$

Proof. To prove the optimality gap of (61) in Theorem 3, under the setting described in Lemma 7, from (91) in Theorem 4, we consider two cases, namely $m\geq l$, and $m<l$, respectively.

Case 1 ($m\geq l$). For the setting of Lemma 7, we evaluate $\mathbf{A}^{\intercal}\mathbf{B}$ using an inner product-based characterization exploiting the following row-block representation:

$$\mathbf{A}=\left[\mathbf{A}_F;\mathbf{A}_\Delta\right]\in\mathbb{F}_q^{m\times l}\ ,\quad \mathbf{B}=\left[\mathbf{B}_F;\mathbf{B}_\Delta\right]\in\mathbb{F}_q^{m\times l}\ , \tag{98}$$

where the probability of a matrix drawn uniformly from $\mathbb{F}_q^{l\times l}$ being singular is equal to $1-\prod_{i=1}^{l}(1-q^{-i})$, which goes to 0 as $q\to\infty$ [58]. Hence, the matrices $\mathbf{A}_F\in\mathbb{F}_q^{l\times l}$ and $\mathbf{B}_F\in\mathbb{F}_q^{l\times l}$ are full-rank, while the matrices $\mathbf{A}_\Delta\in\mathbb{F}_q^{(m-l)\times l}$ and $\mathbf{B}_\Delta\in\mathbb{F}_q^{(m-l)\times l}$ can be directly derived from $\mathbf{A}_F$ and $\mathbf{B}_F$, respectively. More specifically, $\mathbf{A}_\Delta\in\mathbb{F}_q^{(m-l)\times l}$ and $\mathbf{B}_\Delta\in\mathbb{F}_q^{(m-l)\times l}$ can be given as

$$\begin{aligned}\mathbf{A}_\Delta&=\mathbf{G}_1\mathbf{A}_F\in\mathbb{F}_q^{(m-l)\times l}\ ,\\ \mathbf{B}_\Delta&=\mathbf{G}_2\mathbf{B}_F\in\mathbb{F}_q^{(m-l)\times l}\ ,\end{aligned} \tag{99}$$

deterministic mappings $\mathbf{G}_1\in\mathbb{F}_q^{(m-l)\times l}$ and $\mathbf{G}_2\in\mathbb{F}_q^{(m-l)\times l}$ known to source one and source two, respectively. Exploiting (99), we can rewrite the desired matrix product $\mathbf{A}^{\intercal}\mathbf{B}$ as

$$\begin{aligned}\mathbf{A}^{\intercal}\mathbf{B}&=\mathbf{A}_F^{\intercal}\mathbf{B}_F\oplus_q\mathbf{A}_\Delta^{\intercal}\mathbf{B}_\Delta\\ &=\mathbf{A}_F^{\intercal}\mathbf{B}_F\oplus_q\mathbf{A}_F^{\intercal}\mathbf{G}_1^{\intercal}\mathbf{G}_2\mathbf{B}_F\ .\end{aligned} \tag{100}$$

To devise the optimality gap under the setting of Lemma 7, we evaluate R_{AH}^{Σ} in (61) for computing the term $\mathbf{A}_F^{\intercal}\mathbf{B}_F$ in (100), substituting $\mathbf{A}$ and $\mathbf{B}$ by $\mathbf{A}_F=\left[\mathbf{A}_{F1};\mathbf{A}_{F2}\right]$ and $\mathbf{B}_F=\left[\mathbf{B}_{F1};\mathbf{B}_{F2}\right]$, respectively, with $\mathbf{A}_F$ and $\mathbf{B}_F$ drawn independently and uniformly from $\mathbb{F}_q^{l\times l}$:

$$\begin{aligned}R_{\mathrm{AH}}^{\Sigma}(\mathbf{A}_F,\ \mathbf{B}_F)&=H_q(\mathbf{A}_{F1},\mathbf{B}_{F2})\\ &\quad+2\max\{H_q(\mathbf{A}_{F2}\oplus_q\mathbf{B}_{F1}\,|\,\mathbf{A}_{F1},\mathbf{B}_{F2}),\\ &\qquad H_q(\mathbf{A}_{F1}^{\intercal}\mathbf{A}_{F2}\oplus_q\mathbf{B}_{F1}^{\intercal}\mathbf{B}_{F2}\,|\,\mathbf{A}_{F1},\\ &\qquad\qquad\mathbf{B}_{F2},\mathbf{A}_{F2}\oplus_q\mathbf{B}_{F1})\}\\ &\leq 2\cdot\frac{l^2}{2}+2\max\left\{\frac{l^2}{2},\ H_q(\mathbf{A}_{F2})\right\}\\ &=2\cdot\frac{l^2}{2}+2\max\left\{\frac{l^2}{2},\ \frac{l^2}{2}\right\}=2l^2\ ,\end{aligned} \tag{101}$$

where the calculation steps follow the same reasoning as in (105), and the notation $R_{\mathrm{AH}}^{\Sigma}(\mathbf{A}_F,\ \mathbf{B}_F)$ emphasizes that the structured coding is performed for the pair $(\mathbf{A}_F,\ \mathbf{B}_F)$ versus

$(\mathbf{A}, \mathbf{B})$.

To establish the optimality gap, it remains to show the achievability of $\mathbf{A}^\intercal\mathbf{B}$, employing the relation in (100). Exploiting the structured encoding scheme of (101) to recover $\mathbf{A}_F^\intercal\mathbf{B}_F$, we denote the side information available to the receiver by $\mathrm{SI}_F \triangleq \{\mathbf{A}_{F1},\ \mathbf{B}_{F2},\ \mathbf{U}_F = \mathbf{A}_{F2} \oplus_q \mathbf{B}_{F1},\ \mathbf{A}_{F1}^\intercal\mathbf{A}_{F2} \oplus_q \mathbf{B}_{F1}^\intercal\mathbf{B}_{F2}\}$. As a result, the necessary and sufficient rate for the receiver to recover the matrices $\mathbf{A}_\Delta,\ \mathbf{B}_\Delta$, denoted by $R_{\mathrm{SW}}^{\Sigma}(\mathbf{A}_\Delta,\ \mathbf{B}_\Delta \,|\, \mathrm{SI}_F)$, is given as

$$
\begin{aligned}
&R_{\mathrm{SW}}^{\Sigma}(\mathbf{A}_\Delta,\ \mathbf{B}_\Delta \,|\, \mathrm{SI}_F) \triangleq H_q(\mathbf{A}_\Delta,\ \mathbf{B}_\Delta \,|\, \mathrm{SI}_F)\\
&= H_q(\mathbf{G}_1\mathbf{A}_F,\ \mathbf{G}_2\mathbf{B}_F \,|\, \mathbf{A}_{F1},\ \mathbf{B}_{F2},\ \mathbf{A}_{F2} \oplus_q \mathbf{B}_{F1},\\
&\qquad\qquad \mathbf{A}_{F1}^\intercal\mathbf{A}_{F2} \oplus_q \mathbf{B}_{F1}^\intercal\mathbf{B}_{F2})\\
&\overset{(a)}{=} H_q(\mathbf{G}_{11}\mathbf{A}_{F1} \oplus_q \mathbf{G}_{12}\mathbf{A}_{F2},\ \mathbf{G}_{21}\mathbf{B}_{F1} \oplus_q \mathbf{G}_{22}\mathbf{B}_{F2} \,|\, \mathbf{A}_{F1},\\
&\qquad\qquad \mathbf{B}_{F2},\ \mathbf{U}_F,\ \mathbf{A}_{F1}^\intercal\mathbf{A}_{F2} \oplus_q \mathbf{B}_{F1}^\intercal\mathbf{B}_{F2})\\
&= H_q(\mathbf{G}_{12}\mathbf{A}_{F2},\ \mathbf{G}_{21}\mathbf{B}_{F1} \,|\, \mathbf{A}_{F1},\ \mathbf{B}_{F2},\ \mathbf{U}_F,\\
&\qquad\qquad \mathbf{A}_{F1}^\intercal\mathbf{A}_{F2} \oplus_q \mathbf{B}_{F1}^\intercal\mathbf{B}_{F2})\\
&= H_q(\mathbf{G}_{12}\mathbf{A}_{F2},\ \mathbf{G}_{21}(\mathbf{U}_F \ominus_q \mathbf{A}_{F2}) \,|\, \mathbf{A}_{F1},\ \mathbf{B}_{F2},\ \mathbf{U}_F,\\
&\qquad\qquad \mathbf{A}_{F1}^\intercal\mathbf{A}_{F2} \oplus_q (\mathbf{U}_F \ominus_q \mathbf{A}_{F2})^\intercal\mathbf{B}_{F2})\\
&\overset{(b)}{\le} H_q(\mathbf{G}_{12}\mathbf{A}_{F2},\ \mathbf{G}_{21}\mathbf{A}_{F2} \,|\, \mathbf{A}_{F1},\ \mathbf{B}_{F2},\\
&\qquad\qquad \mathbf{A}_{F1}^\intercal\mathbf{A}_{F2} \ominus_q \mathbf{A}_{F2}^\intercal\mathbf{B}_{F2})\\
&\overset{(c)}{=} 0\ ,
\end{aligned}
\tag{102}
$$

where (a) follows from letting $\mathbf{G}_1 = \begin{bmatrix}\mathbf{G}_{11} & \mathbf{G}_{12}\end{bmatrix}$, and $\mathbf{G}_2 = \begin{bmatrix}\mathbf{G}_{21} & \mathbf{G}_{22}\end{bmatrix}$, where the submatrices satisfy $\mathbf{G}_{11}, \mathbf{G}_{12}, \mathbf{G}_{21}, \mathbf{G}_{22} \in \mathbb{F}_q^{(m-l)\times\frac{l}{2}}$. Step (b) follows from eliminating $\mathbf{U}_F$ from the set of conditional random variables, and it is clear that $R_{\mathrm{SW}}^{\Sigma}(\mathbf{A}_\Delta,\ \mathbf{B}_\Delta \,|\, \mathrm{SI}_F\backslash\{\mathbf{U}_F\}) \le H_q(\mathbf{A}_{F2}) = l^2/2$. In step (c), we note that given the set of matrices $\mathbf{A}_{F1} = (a_{ij}^{F1}) \in \mathbb{F}_q^{\frac{l}{2}\times l}$, $\mathbf{B}_{F2} = (b_{ij}^{F2}) \in \mathbb{F}_q^{\frac{l}{2}\times l}$, and $\mathbf{A}_{F1}^\intercal\mathbf{A}_{F2} \ominus_q \mathbf{A}_{F2}^\intercal\mathbf{B}_{F2} = \Big(\sum_{k\in[l/2]} a_{ki}^{F1}a_{kj}^{F2} \ominus_q \sum_{k\in[l/2]} a_{ki}^{F2}b_{kj}^{F2}\Big) \in \mathbb{F}_q^{l\times l}$ available as side information, in the limit as q tends to infinity, with probability 1, the receiver has l^2 linearly independent equations in $\mathbf{A}_{F2} = (a_{ij}^{F2}) \in \mathbb{F}_q^{\frac{l}{2}\times l}$ with $l^2/2$ unknowns. Thus, as $q \to \infty$, given $\mathbf{A}_{F1}$ and $\mathbf{B}_{F2}$, and $\mathbf{A}_{F1}$, drawn independently and uniformly from $\mathbb{F}_q^{\frac{l}{2}\times l}$, the receiver can solve for $\mathbf{A}_{F2}$.

From (101) and (102), we infer that $\mathbf{A}^\intercal\mathbf{B}$ in the case $m \ge l$ can be recovered at a sum rate

$$
R_{\mathrm{AH}}^{\Sigma}(\mathbf{A}_F,\ \mathbf{B}_F) + R_{\mathrm{SW}}^{\Sigma}(\mathbf{A}_\Delta,\ \mathbf{B}_\Delta \,|\, \mathrm{SI}_F) \le 2l^2 + 0 = 2l^2\ . \tag{103}
$$

From Lemma 7, $H_q(\mathbf{A}^\intercal\mathbf{B}) = l^2 \le lm$ when $m \ge l$. Thus, employing (103) leads to

$$
\begin{aligned}
R_1 + R_2 &\le R_{\mathrm{AH}}^{\Sigma}(\mathbf{A}_F,\ \mathbf{B}_F) + R_{\mathrm{SW}}^{\Sigma}(\mathbf{A}_\Delta,\ \mathbf{B}_\Delta \,|\, \mathrm{SI}_F)\\
&\le 2H_q(\mathbf{A}^\intercal\mathbf{B}) = 2l^2\\
&\le 2lm = H_q(\mathbf{A}, \mathbf{B}) = R_{\mathrm{SW}}^{\Sigma}(\mathbf{A}, \mathbf{B})\ .
\end{aligned}
\tag{104}
$$

Case 2 ($m < l$). For the setting of Lemma 7, we evaluate $\mathbf{A}^\intercal\mathbf{B}$ exploiting (61), which gives

$$
\begin{aligned}
R_{\mathrm{AH}}^{\Sigma}(\mathbf{A},\ \mathbf{B}) &\overset{(a)}{=} 2\cdot\frac{lm}{2} + 2\max\Big\{\frac{lm}{2}\ ,\\
&\qquad H_q(\mathbf{A}_1^\intercal\mathbf{A}_2 \oplus_q (\mathbf{U} \ominus_q \mathbf{A}_2)^\intercal\mathbf{B}_2 \,|\, \mathbf{A}_1, \mathbf{B}_2,\\
&\qquad\qquad \mathbf{U} = \mathbf{A}_2 \oplus_q \mathbf{B}_1)\Big\}\\
&\overset{(b)}{\le} lm + 2\max\Big\{\frac{lm}{2}\ ,\ H_q(\mathbf{A}_2)\Big\}\\
&= 2lm = R_{\mathrm{SW}}^{\Sigma}(\mathbf{A},\ \mathbf{B})\ ,
\end{aligned}
\tag{105}
$$

where (a) follows from incorporating $\mathbf{U} = \mathbf{A}_2 \oplus_q \mathbf{B}_1$, (b) from noting that $H_q(\mathbf{A}_1^\intercal\mathbf{A}_2 \oplus_q (\mathbf{U} \ominus_q \mathbf{A}_2)^\intercal\mathbf{B}_2 \,|\, \mathbf{A}_1, \mathbf{B}_2, \mathbf{U} = \mathbf{A}_2 \oplus_q \mathbf{B}_1) \le H_q(\mathbf{A}_2) = lm/2$, given side information $\mathbf{A}_1, \mathbf{B}_2, \mathbf{U}$.

From Lemma 7, when $m < l$, it holds that $H_q(\mathbf{A}^\intercal\mathbf{B}) = 2lm - m^2 \ge lm$. Hence, employing (105) leads to the following sum-rate bound:

$$
\begin{aligned}
R_1 + R_2 \le R_{\mathrm{AH}}^{\Sigma}(\mathbf{A},\ \mathbf{B}) &\le 2lm\\
&= R_{\mathrm{SW}}^{\Sigma}(\mathbf{A},\ \mathbf{B}) \le 2H_q(\mathbf{A}^\intercal\mathbf{B})\\
&= 4lm - 2m^2\ .
\end{aligned}
\tag{106}
$$

The optimality gap follows directly by comparing (104) and (106) with (94) of Lemma 7. □

Theorem 3 considers the regime $q \ge 2$ where $\mathbf{A}$ and $\mathbf{B}$ are drawn independently and uniformly from $\mathbb{F}_q^{m\times l}$. Theorem 4 proves the optimality of Theorem 3 by providing the matching converse (see (97) in Corollary 5) for $\mathbf{A}$ and $\mathbf{B}$ chosen independently and uniformly from $\mathbb{F}_q^{m\times l}$, in the asymptotic setting of $q \to \infty$. For finite q or for more general source distributions, Lemma 7 provides an upper bound to R_{f} through (93), and similarly to $H_q(\mathbf{A}^\intercal\mathbf{B} \,|\, \mathbf{A})$ through (94) (and analogously for $H_q(\mathbf{A}^\intercal\mathbf{B} \,|\, \mathbf{B})$, by modifying the conditioning variable), respectively.

We next investigate the optimality gap for *binary symmetric matrix products* and *binary square matrix products*, in Propositions 8 and 9, respectively.

Proposition 8. (Optimality gap for binary symmetric matrix products.) *Consider the matrix multiplication source network, in the symmetric case, for random matrices $\mathbf{A}$ and $\mathbf{B}$ drawn independently and uniformly from $\mathbb{F}_2^{m\times l}$ for $m,\ l > 1$ such that $m \ge l$, where*

$$
(a_{ij},\ b_{ij}) \sim \mathrm{DSBS}(p) \quad \textit{are i.i.d. across } i \in [m] \textit{ and } j \in [l]\ . \tag{107}
$$

Then, the optimality gap between the hybrid encoding scheme in Proposition 4 and the sum-rate lower bound $R_{\mathrm{HK}}^{\Sigma}(\mathbf{A},\ \mathbf{B})$ in Theorem 5 is upper bounded as

$$
\begin{aligned}
&\frac{R_{\mathrm{KM-OR}}^{\Sigma}(\mathbf{A} \oplus_2 \mathbf{B})}{R_{\mathrm{HK}}^{\Sigma}(\mathbf{A},\ \mathbf{B})}\\
&\le \frac{2mh(p) + m - (l-1)/2 \cdot \bar{F}(l(l-1)/2 + 1; ml, p)}{(2m - l + 1)h(p)}\ ,
\end{aligned}
\tag{108}
$$

where the right-hand side of (108) is upper bounded by $2 + 1/h(p)$ as $m \to \infty$ for $m \ge l$, and by $1 + 1/(2h(p))$ as $m \to \infty$ for $l = 2$, demonstrating the tightness of (65) where $\mathbf{Y} = \mathbf{A} \oplus_2 \mathbf{B}$.

Proof. The symmetric matrix product $\mathbf{A}^\intercal\mathbf{B}$ satisfies Conditions 1 and 2 in Lemma 6. To prove the optimality gap in the symmetric case, for $q = 2$ and $m,\ l > 1$ such that $m \ge l$, we thus exploit the tight converse bounds in (95) of Theorem 5.

Hence, the minimum rate required from source one, for the distributed computation of $\boldsymbol{\mathcal{D}} = \mathbf{A}^\intercal\mathbf{B} \in \mathbb{F}_2^{l\times l}$ must satisfy

$$\begin{aligned}
R_1 &\overset{(a)}{\geq} H(\mathbf{A}\,|\,\mathbf{B},\ \boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{D}}^\intercal) \overset{(b)}{=} H(\mathbf{Y}\,|\,\mathbf{B},\ \mathbf{Y}^\intercal\mathbf{B} = \mathbf{B}^\intercal\mathbf{Y}) \\
&\overset{(c)}{=} \sum_{j\in[l]} H\Big(\{y_{kj}\}_{k\in[m]}\,\Big|\,\{y_{ki}\}_{k\in[m],\ i\in[j-1]}, \{b_{ki}\}_{k\in[m],\ i\in[l]}, \\
&\qquad \Big\{\sum_{k\in[m]} y_{ki}b_{kj} = \sum_{k\in[m]} b_{ki}y_{kj}\Big\}_{i,\ j\in[l]}\Big) \\
&\overset{(d)}{=} \sum_{j\in[l]} H\Big(\{y_{kj}\}_{k\in[m]}\,\Big|\,\{y_{ki}\}_{k\in[m],\ i\in[j-1]}, \{b_{ki}\}_{k\in[m],\ i\in[l]}, \\
&\qquad \Big\{\sum_{k\in\mathcal{K}_j} y_{ki} = \sum_{k\in\mathcal{K}_i} y_{kj}\Big\}_{i,\ j\in[l]}\Big) \\
&\overset{(e)}{\geq} l(m-(l-1)/2)\cdot h(p)\ ,
\end{aligned} \tag{109}$$

and similarly for source two, where (a) follows from exploiting the symmetry $\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{D}}^\intercal \in \mathbb{F}_2^{l\times l}$, and (b) from the elementwise DSBS model in (107), where $\mathbf{Y} = \mathbf{A}\oplus_2\mathbf{B} \in \mathbb{F}_q^{m\times l}$ such that $y_{ki} \sim \mathrm{Bern}(p)$ for all $k\in[m]$, $i\in[l]$, and the fact that given $\mathbf{B}$, the relation $\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{D}}^\intercal$ is equivalent to $\mathbf{Y}^\intercal\mathbf{B} = \mathbf{B}^\intercal\mathbf{Y}$. Step (c) follows from employing the chain rule for entropy, and (d) from setting $\mathcal{K}_i = \{k:\ b_{ki} = 1,\ k\in[m]\}$ for a given value of $i\in[l]$. Step (e) follows from noting the following. For the diagonal entries of $\boldsymbol{\mathcal{D}}$, the relation $\sum_{k\in[m]} y_{ki}b_{kj} = \sum_{k\in[m]} b_{ki}y_{kj} = \sum_{k\in\mathcal{K}_i} y_{ki}$ for $\{i=j\in[l]\}$ holds by definition and therefore provides no reduction in the entropy of $\{y_{kj}\}_{k\in[m]}$. For the non-diagonal entries of $\boldsymbol{\mathcal{D}}$, the corresponding linear side-information terms $\sum_{k\in[m]} y_{ki}b_{kj} = \sum_{k\in[m]} b_{ki}y_{kj}$ for $\{i\neq j\in[l]\}$ yield at most $l(l-1)/2$ linearly independent equations in $\mathbf{Y}$ when $m \gg l$. Consequently, the entropy of $\mathbf{Y}$ is reduced from $mlh(p)$ by up to $(l(l-1)/2)h(p)$. Thus, (109) yields the following tight fundamental limit on R_{HK}^{Σ}:

$$R_{\mathrm{HK}}^{\Sigma}(\mathbf{A},\ \mathbf{B}) \geq l(2m-l+1)h(p)\ . \tag{110}$$

From Proposition 4, employing (65), and letting $\mathbf{Y} = (y_{ij})_{i\in[m],\ j\in[l]} = \mathbf{A}\oplus_2\mathbf{B}$ where $\mathbf{A} \perp\!\!\!\perp \mathbf{Y}$ due to (107), the following sum rate is achievable for the receiver to successfully recover $\boldsymbol{\mathcal{D}}$:

$$\begin{aligned}
R_{\mathrm{KM-OR}}^{\Sigma}(\mathbf{Y}) &\overset{(a)}{=} 2H(\mathbf{Y}\ |\ \mathbf{A}^\intercal\mathbf{B} = \mathbf{B}^\intercal\mathbf{A}) \\
&\quad + H_{G_{\mathbf{A}}}(\mathbf{A}\,|\,\mathbf{Y},\ \mathbf{A}^\intercal\mathbf{Y} = \mathbf{Y}^\intercal\mathbf{A}) \\
&\overset{(b)}{\leq} 2mlh(p) + ml - l(l-1)/2 \\
&\quad \times \bar{F}(l(l-1)/2+1; ml, p)\ ,
\end{aligned} \tag{111}$$

where (a) is derived from rewriting $\mathbf{A}^\intercal\mathbf{B} = \mathbf{B}^\intercal\mathbf{A}$ by using $\mathbf{B} = \mathbf{A}\oplus_2\mathbf{Y}$, and (b) by employing (107), which yields $y_{ij}\sim\mathrm{Bern}(p)$ for all $i\in[m]$, $j\in[l]$, and observing that $\mathbf{A}^\intercal\mathbf{Y} = \mathbf{Y}^\intercal\mathbf{A}$ yields at most $l(l-1)/2$ linearly independent equations in $\mathbf{A}$, which holds when $\mathbf{Y}$ has more than $l(l-1)/2$ nonzero entries, and which occurs with probability $\bar{F}(l(l-1)/2+1; ml, p) = \sum_{j\in[l(l-1)/2+1, ml]} \binom{ml}{j} p^j(1-p)^{ml-j}$. Thus, employing (110) and (111) yields (108). □

Proposition 9. (Optimality gap for binary square matrix products.) *For distributed computing of square matrix products, under the elementwise DSBS model of (107), it holds that*

$$\frac{R_{\mathrm{AH,rfd}}^{\Sigma}(\mathbf{A},\ \mathbf{B})}{R_{\mathrm{HK}}^{\Sigma}(\mathbf{A},\ \mathbf{B})} \leq \begin{cases} \frac{1+3h(p)}{4h(p)}\ , & h(p)\geq\frac{1}{2}\ , \\ \frac{5}{8h(p)}\ , & h(p)<\frac{1}{2}\ , \end{cases} \tag{112}$$

where $R_{\mathrm{AH,rfd}}^{\Sigma}$ provides a refinement over (61) in Theorem 3, as we detail below, and the upper bound on the right-hand side of (112) tends to 1 as $p\to 1/2$.

Proof. The square matrix product $\mathbf{A}^\intercal\mathbf{B}$ satisfies Conditions 1 and 2 in Lemma 6. To prove the optimality gap in the square case, for $q=2$, we thus exploit the tight converse bounds in (95) of Theorem 5. Hence, the minimum sum rate for the distributed computation of a matrix product $\boldsymbol{\mathcal{D}} = \mathbf{A}^\intercal\mathbf{B} \in \mathbb{F}_2^{l\times l}$ must satisfy

$$\begin{aligned}
R_{\mathrm{HK}}^{\Sigma}(\mathbf{A},\ \mathbf{B}) &\geq H(\mathbf{B}\,|\ \mathbf{A}) + H(\mathbf{A}\,|\ \mathbf{B}) \\
&= 2mlh(p) = 2H(\mathbf{A}\oplus_2\mathbf{B})\ ,
\end{aligned} \tag{113}$$

which follows from the elementwise DSBS(p) model, noting that $\mathbf{A}\oplus_2\mathbf{B}$ is independent of $\mathbf{A}$ and $\mathbf{B}$, and employing $H(\mathbf{A}\mid\mathbf{B}) = H(\mathbf{B}\mid\mathbf{A}) = mlh(p)$.

From Theorem 3, employing (61), using the elementwise DSBS model for the matrices $\mathbf{A},\ \mathbf{B}\in\mathbb{F}_2^{m\times l}$, where $(a_{ij},\ b_{ij})\sim\mathrm{DSBS}(p)$, $i\in[m]$, $j\in[l]$, and letting $\mathbf{Y}_1 = \mathbf{A}_1\oplus_2\mathbf{B}_1\in\mathbb{F}_2^{m/2\times l}$, and $\mathbf{Y}_2 = \mathbf{A}_2\oplus_2\mathbf{B}_2\in\mathbb{F}_2^{m/2\times l}$, we rewrite the achievable sum rate for computing $\boldsymbol{\mathcal{D}}$ as

$$\begin{aligned}
&R_{\mathrm{AH}}^{\Sigma}(\mathbf{A},\ \mathbf{B}) = H(\mathbf{A}_1,\mathbf{B}_2) \\
&+ 2\max\Big\{H((\mathbf{B}_2\oplus_2\mathbf{Y}_2)\oplus_2(\mathbf{A}_1\oplus_2\mathbf{Y}_1)\,|\,\mathbf{A}_1,\mathbf{B}_2), \\
&\qquad H(\mathbf{A}_1^\intercal(\mathbf{B}_2\oplus_2\mathbf{Y}_2) \\
&\qquad\quad \oplus_2(\mathbf{A}_1\oplus_2\mathbf{Y}_1)^\intercal\mathbf{B}_2\,|\,\mathbf{A}_1,\mathbf{B}_2,\mathbf{A}_2\oplus_2\mathbf{B}_1)\Big\} \\
&\overset{(a)}{=} 2\cdot\frac{ml}{2} + 2\max\{H(\mathbf{Y}_1\oplus_2\mathbf{Y}_2)\ , \\
&\qquad H(\mathbf{A}_1^\intercal\mathbf{Y}_2\oplus_2\mathbf{Y}_1^\intercal\mathbf{B}_2\,|\,\mathbf{A}_1,\mathbf{B}_2,\mathbf{Y}_1\oplus_2\mathbf{Y}_2)\} \\
&\overset{(b)}{=} ml + 2\max\Big\{\frac{ml}{2}\cdot h(2p(1-p))\ , \\
&\qquad H(\mathbf{A}_1^\intercal(\mathbf{Y}_s\oplus_2\mathbf{Y}_1)\oplus_2\mathbf{Y}_1^\intercal\mathbf{B}_2\,|\,\mathbf{A}_1,\mathbf{B}_2,\mathbf{Y}_s)\Big\} \\
&= ml + \max\{ml\cdot h(2p(1-p))\ , \\
&\qquad 2H(\mathbf{A}_1^\intercal\mathbf{Y}_1\oplus_2\mathbf{Y}_1^\intercal\mathbf{B}_2\,|\,\mathbf{A}_1,\mathbf{B}_2,\mathbf{Y}_s)\} \\
&\overset{(c)}{=} ml + \max\{ml\cdot h(2p(1-p))\ ,\ 2H(\mathbf{Y}_1)\} \\
&= ml(1+h(2p(1-p)))\ ,
\end{aligned} \tag{114}$$

where (a) follows from $\mathbf{Y}_1\oplus_2\mathbf{Y}_2 = (\mathbf{A}_1\oplus_2\mathbf{B}_2)\oplus_2(\mathbf{A}_2\oplus_2\mathbf{B}_1)$, with $\mathbf{A}_1$ and $\mathbf{B}_2$ given, (b) by letting $\mathbf{Y}_s = \mathbf{Y}_1\oplus_2\mathbf{Y}_2$, under the elementwise DSBS$(2p(1-p))$ model for $\mathbf{Y}_s$, which follows directly from (107), and (c) by noting that $H(\mathbf{A}_1^\intercal\mathbf{Y}_1\oplus_2\mathbf{Y}_1^\intercal\mathbf{B}_2\,|\,\mathbf{A}_1,\mathbf{B}_2,\mathbf{Y}_s) \leq H(\mathbf{Y}_1) = (ml/2)\cdot h(p) \overset{(d)}{\leq} (ml/2)\cdot h(2p(1-p))$, where (d) is due to the Schur concavity of $h(\cdot)$. Hence, $\max\{ml\cdot h(2p(1-p))\ ,\ 2H(\mathbf{A}_1^\intercal\mathbf{Y}_1\oplus_2\mathbf{Y}_1^\intercal\mathbf{B}_2\,|\,\mathbf{A}_1,\mathbf{B}_2,\mathbf{Y}_s)\} = ml\cdot h(2p(1-p))$. Note that (114) remains valid even when $R_{\mathrm{f}} = H(\boldsymbol{\mathcal{D}}) > mlh(2p(1-p))$, and it further implies that

$$R_{\mathrm{AH}}^{\Sigma}(\mathbf{A},\ \mathbf{B}) = ml(1+h(2p(1-p)))$$

$$\overset{(a)}{\geq} R^{\Sigma}_{\mathrm{SW}}(\mathbf{A},\ \mathbf{B}) = H(\mathbf{A}\ ,\mathbf{B}) = ml(1+h(p))\ , \tag{115}$$

where (a) follows from the Schur concavity of $h(\cdot)$. The relation (115) indicates that the computation of $\boldsymbol{\mathcal{D}}$ is ensured at a higher rate compared to [18], while our scheme that achieves the sum rate $R^{\Sigma}_{\mathrm{AH}}(\mathbf{A},\ \mathbf{B})$ in (114) does not guarantee the recovery of $(\mathbf{A}, \mathbf{B})$.

Motivated by the encoding framework in [13] and its extension to additions over $\mathbb{F}_q$ in [14, Lemma 5], we proceed to examine the achievability of $R^{\Sigma}_{\mathrm{AH}} \leq 2R_{\mathrm{f}}$ for the setting of Theorem 4. To that end, as $q \to \infty$, for $m \geq l$, $2R_{\mathrm{f}} = 2l^2 \leq 2lm = R^{\Sigma}_{\mathrm{SW}}$, and from (115) $R^{\Sigma}_{\mathrm{AH}} \leq 2ml$, meaning $R^{\Sigma}_{\mathrm{AH}} \leq 2R_{\mathrm{f}}$ only when $l = m$. For $m < l$, $2R_{\mathrm{f}} = 4lm - 2m^2 \geq R^{\Sigma}_{\mathrm{SW}} = 2lm \geq R^{\Sigma}_{\mathrm{AH}}$.

Vertically concatenating the columns of the source matrices $\mathbf{A}$ and $\mathbf{B}$, we obtain $\mathbf{X}_1 = \big[\mathbf{A}(:,1); \mathbf{A}(:,2); \ldots; \mathbf{A}(:,l)\big] \in \mathbb{F}_2^{ml}$, and $\mathbf{X}_2 = \big[\mathbf{B}(:,1); \mathbf{B}(:,2); \ldots; \mathbf{B}(:,l)\big] \in \mathbb{F}_2^{ml}$, respectively. Following the steps of Lemma 1 and Proof of Proposition 1, we let $\mathbf{Z} = \mathbf{X}_1 \oplus_2 \mathbf{X}_2 \in \mathbb{F}_2^{ml}$. For fixed $\epsilon > 0$, $\delta > 0$, and for sufficiently large n, we choose a binary encoding matrix $\boldsymbol{\mathcal{C}}$ drawn independently and uniformly from $\mathbb{F}_2^{\kappa_j \times n}$. Then, there exists a decoding function $\psi_j : \mathbb{F}_2^{\kappa_j} \to \mathbb{F}_2^n$, for each $j \in [ml]$, that satisfies [19]:

$$\begin{aligned}\hat{\mathbf{Z}}^n(j) &\triangleq \phi_j(\boldsymbol{\mathcal{C}}\mathbf{X}_1^n(j), \boldsymbol{\mathcal{C}}\mathbf{X}_2^n(j)) \\ &\triangleq \psi_j(\boldsymbol{\mathcal{C}}\mathbf{X}_1^n(j)) \oplus_2 \boldsymbol{\mathcal{C}}\mathbf{X}_2^n(j)))\end{aligned} \tag{116}$$

such that i) $\kappa_j < n(H(\mathbf{Z}(j)\,|\,\{\mathbf{Z}(j')\}_{j'<j}) + \epsilon)$, where $H(\mathbf{Z}(j)) = h(p)$ for all $j \in [ml]$, under (107), and ii) $\mathbb{P}(\psi_j(\boldsymbol{\mathcal{C}}\mathbf{Z}^n(j))) \neq \mathbf{Z}^n(j)) < \delta$. Hence, the application of Lemmas 4 and 5 yields that given a linear encoding matrix $\boldsymbol{\mathcal{C}} \in \mathbb{F}_q^{\kappa \times n}$ with $\kappa = \sum_{j\in[ml]} \kappa_j$, the pair $(\boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{C}})$ is an (n, ϵ, δ)-coding scheme [14], and the following rate per source can be achieved for computing $\mathbf{Z}$:

$$\frac{\kappa}{n} < H(\mathbf{Z}(j),\, j \in [ml]) + \epsilon = mlh(p) + \epsilon\ . \tag{117}$$

Thus, the achievable sum rate for computing $\boldsymbol{\mathcal{D}}$ becomes (providing a refinement over (114)):

$$\begin{aligned}&R^{\Sigma}_{\mathrm{AH,rfd}}(\mathbf{A},\mathbf{B}) \\ &\overset{(a)}{=} \Big(H(\mathbf{A}_1) - \frac{\kappa}{2n}\Big) + 2\max\Big\{H(\mathbf{A} \oplus_2 \mathbf{B}), \\ &\qquad H(\mathbf{A}_1^{\intercal}\mathbf{A}_2 \oplus_2 (\mathbf{A}_1 \oplus_2 \mathbf{Y}_1)^{\intercal}(\mathbf{A}_2 \oplus_2 \mathbf{Y}_2)\,|\,\mathbf{A} \oplus_2 \mathbf{B},\ \mathbf{A}_1)\Big\} \\ &= \Big(\frac{ml}{2} - \frac{\kappa}{2n}\Big) + 2\max\Big\{mlh(p), \\ &\qquad H(\mathbf{A}_1^{\intercal}\mathbf{Y}_2 \oplus_2 \mathbf{Y}_1^{\intercal}\mathbf{A}_2\,|\,\mathbf{Y}_1,\ \mathbf{Y}_2,\ \mathbf{A}_1)\Big\} \\ &\overset{(b)}{\leq} \Big(\frac{ml}{2} - \frac{\kappa}{2n}\Big) + 2\max\Big\{mlh(p), \\ &\qquad H(\mathbf{A}_2\,|\,\mathbf{Y}_1,\ \mathbf{Y}_2,\ \mathbf{A}_1)\Big\} \\ &\overset{(c)}{=} \Big(\frac{ml}{2} - \frac{\kappa}{2n}\Big) + 2\max\Big\{mlh(p),\ \frac{ml}{2}\Big\} \\ &\overset{(d)}{=} \begin{cases}\frac{ml}{2}(1+3h(p))\ , & h(p) \geq \frac{1}{2}\ , \\ \frac{5ml}{4}\ , & h(p) < \frac{1}{2}\ ,\end{cases} \\ &\leq 2ml\end{aligned} \tag{118}$$

where (a) follows from several observations. We begin by linearly encoding $\mathbf{A}$ using a matrix $\boldsymbol{\mathcal{C}} \in \mathbb{F}_2^{\kappa \times n}$, and applying an (n, ϵ, δ)-coding scheme to compute both $\mathbf{Y} = \mathbf{A} \oplus_2 \mathbf{B}$ and $\mathbf{A}_1^{\intercal}\mathbf{A}_2 \oplus_2 \mathbf{B}_1^{\intercal}\mathbf{B}_2$. This results in an encoding rate of κ/n for each of $\mathbf{A}$ and $\mathbf{B}$. The codewords used to compute $\mathbf{Y}$ (with $H(\mathbf{Y}) = mlh(p)$ from (107)) additionally enable recovery of the mixed terms $\mathbf{A}_2 \oplus_2 \mathbf{B}_1$ and $\mathbf{A}_1 \oplus_2 \mathbf{B}_2$. Given access to these intermediate terms, fully recovering $\mathbf{A}_1$ requires an extra rate of $H(\mathbf{A}_1) - \kappa/(2n)$. With this additional information, the receiver can reconstruct $\mathbf{A}_1, \mathbf{B}_2$, $\mathbf{A}_2 \oplus_q \mathbf{B}_1$, and $\mathbf{A}_1^{\intercal}\mathbf{A}_2 \oplus_q \mathbf{B}_1^{\intercal}\mathbf{B}_2$, from which $\mathbf{A}^{\intercal}\mathbf{B}$ can be fully recovered using the identity in (63). Step (b) follows by conditioning, (c) from that $\mathbf{Y} \perp\!\!\!\perp \mathbf{A}$ and $\mathbf{Y} \perp\!\!\!\perp \mathbf{B}$, and $\mathbf{A}_1 \perp\!\!\!\perp \mathbf{A}_2$, and (d) by choosing $\kappa/n \leq ml \cdot \max\{h(p), 1/2\}$ from (117). From (118), it holds that $R^{\Sigma}_{\mathrm{AH,rfd}} \leq R^{\Sigma}_{\mathrm{SW}} = ml(1+h(p))$, where $R^{\Sigma}_{\mathrm{AH,rfd}}$ can be further reduced below (114) by setting $\kappa/n \leq l \cdot \max\{mh(p), l\}$.

Given the elementwise DSBS model for matrices $\mathbf{A},\ \mathbf{B} \in \mathbb{F}_2^{m\times l}$, we have

$$\begin{aligned}R_{\mathrm{f}} &\overset{(a)}{=} H(\mathbf{A}^{\intercal}(\mathbf{A} \oplus_2 \mathbf{Y})) \\ &= H\Big(\big\{\sum_{k\in[m]} a_{ki}(a_{ki} \oplus_2 y_{ki})\big\}_{i\in[l]}, \\ &\qquad \big\{\sum_{k\in[m]} a_{ki}(a_{kj} \oplus_2 y_{kj})\big\}_{i,\ \substack{j\in[l]\\ i\neq j}}\Big) \\ &\overset{(b)}{=} H\Big(\big\{\sum_{k\in\mathcal{K}_i}(1 \oplus_2 y_{ki})\big\}_{i\in[l]}\Big) \\ &+ H\Big(\big\{\sum_{k\in\mathcal{K}_i} a_{kj} \oplus_2 y_{kj}\big\}_{i,\ \substack{j\in[l]\\ i\neq j}}\,|\,\big\{\sum_{k\in\mathcal{K}_i}(1 \oplus_2 y_{ki})\big\}_{i\in[l]}\Big) \\ &\overset{(c)}{=} H\Big(\big\{\sum_{k\in\mathcal{K}_i}(1 \oplus_2 y_{ki})\big\}_{i\in[l]}\Big) \\ &+ H\Big(\big\{\sum_{k\in\mathcal{K}_i} b_{kj}\big\}_{i,\ \substack{j\in[l]\\ i\neq j}}\,|\,\big\{\sum_{k\in\mathcal{K}_i}(1 \oplus_2 y_{ki})\big\}_{i\in[l]}\Big) \\ &\overset{(d)}{=} H\Big(\big\{\sum_{k\in\mathcal{K}_i}(1 \oplus_2 y_{ki})\big\}_{i\in[l]}\Big) + H\Big(\big\{\sum_{k\in\mathcal{K}_i} b_{kj}\big\}_{i,\ \substack{j\in[l]\\ i\neq j}}\Big) \\ &\overset{(e)}{=} \Big(1 - \frac{1}{2^m}\Big)lh((1-p)^{(|\mathcal{K}_i|)}) + \Big(1 - \frac{1}{2^m}\Big)l(l-1) \\ &\overset{(f)}{\geq} \Big(1 - \frac{1}{2^m}\Big)l(h(p) + l - 1)\ ,\end{aligned} \tag{119}$$

where (a) follows from $\mathbf{Y} = \mathbf{A} \oplus_2 \mathbf{B}$, where $\{y_{kj}\}_{k\in[m],\ j\in[l]}$ are i.i.d., and $\{a_{ki}\}_{k\in[m],\ i\in[l]}$ are independently and uniformly distributed, and $\mathbf{A} \perp\!\!\!\perp \mathbf{Y}$, (b) from letting $\mathcal{K}_i = \{k :\ a_{ki} = 1,\ k \in [m]\}$ for a given i, (c) from using $b_{kj} = a_{kj} \oplus_2 y_{kj} \sim \mathrm{Bern}(1/2)$, (d) is because $b_{kj} \perp\!\!\!\perp b_{ki}$ for any $j \neq i$ and $b_{kj} \perp\!\!\!\perp \{y_{kj},\ y_{ki}\}$, and (e) from using $1 \oplus_2 y_{ki} \sim \mathrm{Bern}(1-p)$ and letting $p^{(k)} = p^{(k-1)}(1-p) + (1-p^{(k-1)})p$, for $k \geq 2$, setting $p^{(1)} = p$, by substituting $1-p$ for p, and noting that $\sum_{k\in\mathcal{K}_i}(1 \oplus_2 y_{ki}) \sim \mathrm{Bern}((1-p)^{(|\mathcal{K}_i|)})$ for $|\mathcal{K}_i| \geq 1$, and $\sum_{k\in\mathcal{K}_i} b_{kj} \sim \mathrm{Bern}(1/2)$, based on the uniform and i.i.d. nature of the components of $\mathbf{B}$, provided $|\mathcal{K}_i| \geq 1$. Step (f) follows from the Schur concavity of $h(\cdot)$. Contrasting (119) with (118), the necessary condition for $R^{\Sigma}_{\mathrm{AH,rfd}} \leq 2R_{\mathrm{f}}$ in the special case of $p = 1/2$ is $l \geq m/\big(1 - 2^{-m}\big)$.

Using $R^{\Sigma}_{\mathrm{AH,rfd}}$ in (118) and the converse R^{Σ}_{HK} in (113) yields the optimality gap in (112). $\square$

V. Conclusions

We tackle the well-known open problem of distributed computing of bilinear functions, including dot and matrix products, which form an important class of non-linear functions. Our key contribution is a class of *structured source codes* over finite fields for distributed computing of dot and matrix products. These codes underpin our achievability schemes. To this end, we introduced a non-linear source transformation-based design principle by leveraging the linear coding scheme of Körner-Marton, and for general square matrix products, a careful calibration of the Ahlswede-Han approach demonstrated optimality as $q \to \infty$ for matrices drawn independently and uniformly from $\mathbb{F}_q^{m\times l}$. Our scheme for $q = 2$ is within a constant factor of optimal, under the elementwise DSBS model. Our designs also include a hybrid encoding scheme that combines Körner's characteristic graphs with the Orlitsky-Roche approach, as well as additional constructions based on recursive and nested applications of distributed dot products. The proposed schemes can surpass the performance of the state of the art, providing savings in the communication cost. Our schemes are complemented by calibrated converse bounds derived via the Han-Kobayashi approach and the strong converse theorem of Ahlswede-Gács-Körner, yielding relatively tight sum-rate converses. Furthermore, the exact optimality of our achievability result in Theorem 3 is guaranteed for the distributed computation of products of independently and uniformly drawn source matrices from $\mathbb{F}_q^{m\times l}$ as $q \to \infty$, as stated in Corollary 5.

Further research is required to extend our coding constructions to general source classes. While our results hold asymptotically, practical coding schemes may rely on zero-error adaptations for exact recovery [59], [60] or on one-shot models for finite-length and approximate representations [61], [62]. One can also explore how computational costs, measured in terms of the required number of multiplications, can be reduced by leveraging a class of algorithms like the Strassen-type algorithms, e.g., [42], [57], for matrix multiplication. Our future directions include expanding the proposed design to a wider range of non-linear functions, such as general bilinear maps, including tensor products, chain matrix products, sorting or classification functions, and non-linearly separable functions, and addressing the problems of distributed rank computation, matrix decomposition, and low-rank matrix factorization [63]. An interesting research direction can involve security and privacy-related ramifications. In particular, one may explore adapting our schemes to the coded distributed matrix multiplication framework by devising resilient polynomial codes with security and privacy guarantees while keeping communication and computation overheads minimal. Our design and analysis tools are also relevant to coded distributed computing scenarios over real numbers [64], and to deep neural network and large language model literature, where large-scale matrix multiplication is a key component, and hardware memory is the main bottleneck [65]. Thus, approximate matrix multiplication via quantization becomes the natural solution, and can be achieved using lossy generalizations of the Körner-Marton scheme [66], to allow a receiver to estimate the product, and analyze the approximation error as a function of the quantization rate.

Acknowledgment

The author acknowledges the constructive discussions with Petros Elia and Arun Padakandla, and thanks the anonymous reviewers for their thorough and constructive feedback, which substantially improved the manuscript, and the editors for their careful handling of the manuscript.

References

[1] D. Malak, "Distributed structured matrix multiplication," in *Proc., IEEE Int. Symp. Inf. Theory*, Athens, Greece, Jul. 2024.

[2] G. Strang, *Introduction to Linear Algebra*. Cambridge University Press, 2023.

[3] H. Buhrman, R. Cleve, J. Watrous, and R. De Wolf, "Quantum fingerprinting," *Phys. Rev. Lett.*, vol. 87, no. 16, p. 167902, Sep. 2001.

[4] 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.*, vol. 30, Long Beach, CA, Dec. 2017, pp. 4403–4413.

[5] 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.

[6] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," in *Proc., Int. Conf. on Machine Learning*, Sydney, Australia, Aug. 2017, pp. 3368–3376.

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

[8] W. Halbawi, N. Azizan, F. Salehi, and B. Hassibi, "Improving distributed gradient descent using Reed-Solomon codes," in *Proc., IEEE ISIT*, Vail, CO, Jun. 2018, pp. 2027–2031.

[9] 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.

[10] Q. Yu, S. Li, N. Raviv, S. M. M. Kalan, M. Soltanolkotabi, and S. A. Avestimehr, "Lagrange coded computing: Optimal design for resiliency, security, and privacy," in *Proc., Int. Conf. Artif. Intell. Stat.*, Naha, Okinawa, Japan, Apr. 2019, pp. 1215–1225.

[11] J. Shamsi, M. A. Khojaye, and M. A. Qasmi, "Data-intensive cloud computing: Requirements, expectations, challenges, and solutions," *J. Grid Comput.*, vol. 11, no. 2, pp. 281–310, Jun. 2013.

[12] 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.

[13] 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.

[14] T. S. 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.

[15] B. Nazer and M. Gastpar, "Computation over multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3498–3516, Sep. 2007.

[16] S. H. Lim, C. Feng, A. Pastore, B. Nazer, and M. Gastpar, "Towards an algebraic network information theory: Distributed lossy computation of linear functions," in *Proc., IEEE ISIT*, Paris, France, Jun. 2019, pp. 1827–31.

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

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

[19] 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.

[20] 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.

[21] S. S. Pradhan, A. Padakandla, and F. Shirani, "An algebraic and probabilistic framework for network information theory," *Found. Trends Commun. Inf. Theory*, vol. 18, no. 2, pp. 173–379, Dec. 2020.

[22] R. Ahlswede, P. Gács, and J. Körner, "Bounds on conditional probabilities with applications in multi-user communication," *Z. Wahrsch. verw. Geb.*, vol. 34, no. 2, pp. 157–177, Jun. 1976.

[23] H. Yamamoto, "Wyner-Ziv theory for a general function of the correlated sources," *IEEE Trans. Inf. Theory*, vol. 28, no. 5, pp. 803–7, Sep. 1982.

[24] J. Körner, "Coding of an information source having ambiguous alphabet and the entropy of graphs," in *Proc., 6th Prague Conf. Inf. Theory*, Prague, Czech Republic, Sep. 1971, pp. 411–425.

[25] A. Orlitsky and J. R. Roche, "Coding for computing," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, p. 903–917, Mar. 2001.

[26] D. Malak, "Fractional graph coloring for functional compression with side information," in *Proc., IEEE ITW*, Mumbai, India, Nov. 2022.

[27] 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.

[28] A. Lenz, R. Bitar, A. Wachter-Zeh, and E. Yaakobi, "Function-correcting codes," *IEEE Trans. Inf. Theory*, May 2023.

[29] P. Elias, "Coding for noisy channels," in *IRE Conv. Rec.*, no. Part 4, 1955, p. 37–46.

[30] T. Berger, "Multiterminal source coding," in *The Information Theory Approach to Communications*, G. Longo, Ed. New York, NY, USA: Springer-Verlag, 1978.

[31] R. Ahlswede and I. Csiszár, "To get a bit of information may be as hard as to get full information," *IEEE Trans. Inf. Theory*, vol. 27, no. 4, pp. 398–408, Jul. 1981.

[32] M. A. Sohail, T. A. Atif, and S. S. Pradhan, "Unified approach for computing sum of sources over CQ-MAC," in *Proc., IEEE ISIT*, Espoo, Finland, 2022, pp. 1868–1873.

[33] S. Dutta, V. Cadambe, and P. Grover, ""Short-Dot": Computing large linear transforms distributedly using coded short dot products," in *Proc., Adv. Neural Inf. Process. Syst.*, vol. 29, Barcelona, Spain, Dec. 2016.

[34] H. H. López, G. L. Matthews, and D. Valvo, "Secure MatDot codes: A secure, distributed matrix multiplication scheme," in *Proc., IEEE ITW*, Mumbai, India, Nov. 2022, pp. 149–154.

[35] 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.

[36] H. Yang and J. Lee, "Secure distributed computing with straggling servers using polynomial codes," *IEEE Trans. Inf. Foren. and Secur.*, vol. 14, no. 1, pp. 141–150, Jun. 2018.

[37] A. Fidalgo-Díaz and U. Martínez-Peñas, "Distributed matrix multiplication with straggler tolerance using algebraic function fields," *arXiv preprint arXiv:2401.13573*, Jan. 2024.

[38] G. D. Villa Salvador, *Topics in the theory of algebraic function fields*. Springer, 2006.

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

[40] 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.

[41] 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.

[42] A. Fawzi, M. Balog, A. Huang, T. Hubert, B. Romera-Paredes, M. Barekatain, A. Novikov, F. J. R Ruiz, J. Schrittwieser, G. Swirszcz, D. Silver, D. Hassabis, and P. Kohli, "Discovering faster matrix multiplication algorithms with reinforcement learning," *Nature*, vol. 610, no. 7930, pp. 47–53, Oct. 2022.

[43] 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.

[44] L. Yu, "Gray–Wyner and mutual information regions for doubly symmetric binary sources and Gaussian sources," *IEEE Tran. Inf. Theory*, Jul. 2023.

[45] D. Irony, S. Toledo, and A. Tiskin, "Communication lower bounds for distributed-memory matrix multiplication," *J. Parallel Distrib. Comput.*, vol. 64, no. 9, pp. 1017–1026, Sep. 2004.

[46] F. Shirani and S. S. Pradhan, "Finite block-length gains in distributed source coding," in *Proc., IEEE ISIT*, Honolulu, HI,, Jun. 2014, pp. 1702–1706.

[47] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 2012.

[48] R. G. Gallager, *Information Theory and Reliable Communication*. Wiley: New York, Jan. 1968, vol. 588.

[49] D. Malak, "Structured codes for distributed matrix multiplication," *arXiv preprint arXiv:2501.00371*, Dec. 2024.

[50] N. Alon and A. Orlitsky, "Source coding and graph entropies," *IEEE Trans. Inf. Theory*, vol. 42, pp. 1329–39, Sep. 1996.

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

[52] D. Malak, "Distributed computing of functions of structured sources with helper side information," in *Proc., IEEE Int. Wksh. Signal Proces. Advances in Wireless Commun. (SPAWC)*, Shanghai, China, Sep. 2023.

[53] M. R. D. Salehi and D. Malak, "An achievable low complexity encoding scheme for coloring cyclic graphs," in *Proc., Allerton*, Monticello, IL, Sep. 2023, pp. 1–8.

[54] D. Malak, M. R. Deylam Salehi, B. Serbetci, and P. Elia, "Multi-functional distributed computing," in *Proc., IEEE Allerton*, Urbana-Champaign, IL, 2024, pp. 1–8.

[55] 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.

[56] R. Beigel, "Finding maximum independent sets in sparse and general graphs," in *Proc., ACM-SIAM SODA*, vol. 99, Baltimore, MD, USA, Jan. 1999, pp. 856–857.

[57] V. Strassen *et al.*, "Gaussian elimination is not optimal," *Numerische mathematik*, vol. 13, no. 4, pp. 354–356, Aug. 1969.

[58] W. C. Waterhouse, "How often do determinants over finite fields vanish?" *Discrete Math.*, vol. 65, no. 1, pp. 103–104, May 1987.

[59] J. Körner and A. Orlitsky, "Zero-error information theory," *IEEE Trans. Inf. Theory*, vol. 44, pp. 2207–29, Oct. 1998.

[60] N. Charpenay, M. Le Treust, and A. Roumy, "Optimal zero-error coding for computing under pairwise shared side information," in *Proc., IEEE ITW*, Saint-Malo, France, Apr. 2023, pp. 97–101.

[61] S. Torabi and J. M. Walsh, "Lossy interactive sum modulo two computation of binary sources," 2016. [Online]. Available: https://faculty.coe.drexel.edu/jwalsh/Torabi_ICASSP16.pdf

[62] ——, "Distributed lossy interactive function computation," in *Proc., Allerton*, Monticello, IL, Sep. 2016, pp. 393–400.

[63] A. Khalesi and P. Elia, "Tessellated distributed computing," *IEEE Trans. Inf. Theory*, Apr. 2025.

[64] P. Moradi, H. Akbarinodehi, and M. A. Maddah-Ali, "General coded computing: Adversarial settings," *arXiv preprint arXiv:2502.08058*, Feb. 2025.

[65] O. Ordentlich and Y. Polyanskiy, "Optimal quantization for matrix multiplication," *arXiv preprint, arXiv:2410.13780v3*, pp. arXiv–2410, Oct. 2024.

[66] D. Krithivasan and S. S. Pradhan, "Lattices for distributed source coding: Jointly Gaussian sources and reconstruction of a linear function," *IEEE Trans. Inf. Theory*, vol. 55, no. 12, pp. 5628–5651, Nov. 2009.

Derya Malak is an Assistant Professor (Maître de Conférence) in the Communication Systems Department at Eurecom, France. Previously, she was a tenure track Assistant Professor in the Department of ECSE at RPI between 2019-2021, and a Postdoctoral Associate at MIT between 2017-2019. She received her Ph.D. in ECE at the University of Texas at Austin in 2017, B.S. in Electrical and Electronics Engineering (EEE) with a minor in Physics at Middle East Technical University, in 2010, and M.S. in EEE at Koc University, in 2013. Dr. Malak has held visiting positions in INRIA and LINCS, Paris, and at Northeastern University, and summer internships at Huawei, Plano, TX, and Bell Labs, Murray Hill, NJ. Her expertise is in information theory, communication theory, and networking areas. She has developed novel distributed computation solutions, and wireless caching algorithms by capturing the confluence of storage, communication, and computation aspects.

Dr. Malak was awarded the Graduate School fellowship by UT Austin between 2013-2017. She was selected to participate in the Rising Stars Workshop for Women in EECS, MIT, in 2018. She received the best paper awards in WiOpt 2022 and WiOpt 2023. Her research has been funded by the ANR PEPR, NSF, the Rensselaer-IBM AI Research Collaboration, and the DARPA Dispersive Computing Programs. She is the recipient of the ERC Starting Grant 2023-2028 on computing nonlinear functions over communication networks (SENSIBILITÉ).